
DOCTORAL THESIS

# Efficient and Robust Techniques for Predictive Control of Nonlinear Processes

**Jörn Klaas Gruber**

**Seville, June 2010**


DOCTORAL THESIS

# Efficient and Robust Techniques for Predictive Control of Nonlinear Processes

**Jörn Klaas Gruber**

Thesis submitted for the degree of
Doctor of Philosophy

Dpto. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingenieros
Universidad de Sevilla

Seville, June 2010


DOCTORAL THESIS

# Efficient and Robust Techniques for Predictive Control of Nonlinear Processes

**Author:** Jörn Klaas Gruber
**Supervisors:** Carlos Bordóns Alba
Daniel Rodríguez Ramírez

To María

To my parents

# Acknowledgements

In first place I have to express my deep gratitude to my advisors Carlos and Dani R. who made possible this PhD thesis. Besides all the assistance in research questions in the last years, they have been always good friends and colleagues.

For their important influence on this PhD thesis I would like to thank Dani L. and Teo, both for their indispensable support in areas like stability and convex optimization as well as for many non-scientific discussions. Special thanks to Eduardo for giving me a home in his research group and the warm reception in Seville.

For being good friends who always have been there to help with words and deeds I would like to convey my gratefulness to Gil, Manolo R., Manolo V., Guilherme, Mirko, Antonio, Vicente, Isa and Pablo. My thankfulness to the Predictive Control Group and the Departamento de Sistemas y Automática, especially to Miguel Ángel, Fernando D., David M. and Ignacio. Many thanks to Matthias and Alejandro O. for their valuable help in applying different control strategies to the fuel cell. Furthermore, I would also like to thank Pablo S., Adrián and Filiberto for putting up with me during these years in the same office.

My gratitude to the Departamento de Lenguajes y Computación of the Universidad de Almería for the excellent collaboration. My special thanks to Manolo B., Paco, José Luis, Ramón and Jorge for our friendship beyond greenhouse climate control.

Furthermore, I would like to thank my friends from València and Castelló de la Plana for the great reception, the exciting time together and the introduction to the Spanish language. Special thanks to Nacho for our friendship during all these years. Moreover, I am grateful to my friends from Stuttgart and Elda, especially to Ana and Martin.

My deep gratitude to my in-laws who always have been like my own family. Thanks

to Ángel Manuel, Ángel, Consoly and Juan Pedro for simultaneously being family and good friends. I am deeply grateful to Manoli, for never doubting in giving me a new home and for the ongoing lessons in "real" Spanish. Special thanks to María for giving me every day so many reasons to not regret my decisions.

Thanks to my brother Ingo and my sister-in-law Barbara for their support during the last years. My deepest gratitude to my parents for encouraging me to chose this direction, for their faith in me and their unconditional love.

*Gracias de corazón a todos los que me animaron a seguir mis sueños.*

# Abstract

Model Predictive Control (MPC) has undergone a great development in the last few decades and has become one of the most popular advanced control techniques in the research community and industry. The success of MPC is the result of several factors, amongst the most important ones are the intuitive formulation of the control problem, the possibility to control a great variety of processes, the consideration of constraints and, at least in the case of linear MPC, the easy implementation of the control law. Another advantage for the use in industrial applications is the possibility to use models which can be obtained easily from the considered process, e.g. step or impulse response models.

Even though virtually all processes of practical importance exhibit some degree of nonlinear behavior, the grand majority of MPC techniques have been developed for linear systems. The application of linear MPC to strongly nonlinear dynamic processes can result in a deficient control performance due to plant/model mismatch. In order to obtain a better control performance, the use of Nonlinear Model Predictive Control (NMPC) should be considered. Another problem in process control is the influence of exogenous disturbances or system uncertainties which can destabilize the closed-loop system under certain conditions or, at least, result in an insufficient control performance. Here, Min-Max Model Predictive Control (MMMPC) offers the possibility to prevent this undesired effect and to obtain a more robust control performance by explicitly considering uncertainties and disturbances in the used prediction model.

Unfortunately, both NMPC and MMMPC exhibit some mayor drawbacks for their use in industry. In the case of NMPC, the difficulty to obtain suitable models of the considered process and the required solution of a possibly non-convex optimization problem have to be mentioned. On the other side, the main drawback of MMMPC is the computational burden to compute the control signal, with a computational complexity growing exponentially with the considered prediction horizon. Besides, both control strategies considerably complicate the study of theoretical issues as stability, robustness and optimality. The mentioned problems account for the seldom use of NMPC and

MMMPC in industry where only a few applications have been reported, even when there is evidence that these control techniques usually improve the control performance.

This thesis addresses the study and development of novel MPC techniques based on nonlinear and uncertain models, i.e. the general framework of this thesis covers NMPC and MMMPC. Particular attention has been paid to the applicability of the developed control strategies, based on the idea to reduce the gap between industrial practice and academic research. On the theoretical side, stability issues played an important role in the design of new MPC strategies, including the specification of the necessary and sufficient conditions to obtain closed-loop stability.

The main focus in the development of MPC has been on nonlinear discrete-time Volterra series models and their use in receding horizon control strategies. Based on second order Volterra series models, a computationally efficient iterative algorithm to solve the optimization problem of an unconstrained NMPC strategy has been enhanced to include constraints and a weighting of the control effort. Furthermore, a novel approach based on the convexification of the possibly non-convex optimization problem of NMPC based on second order Volterra series models has been developed. This approach determines a convex hull of the optimization problem and enables the possibility of global minimization.

In addition, discrete-time uncertain linear models have been considered within the framework of robust control as a previous step to an MMMPC based on Volterra series models. Here, the robust stability of a linear MMMPC strategy based on a nonlinear upper bound of the worst case has been proven. Under consideration of additive and persistent disturbances in a second order Volterra series model, a novel nonlinear MMMPC strategy has been developed. An explicit formulation of the worst case cost has been obtained and, as a consequence, the min-max optimization problem is reduced to a mere minimization problem.

Finally, the developed control strategies have been implemented and the practical applicability has been validated in experiments with different benchmark systems. The obtained results showed that the proposed NMPC and MMMPC strategies have an improved control performance in comparison to linear MPC. The successful application of the different control strategies joins the small number of NMPC and MMMPC applications reported in specialized literature.

# Resumen

El control predictivo basado en modelo (MPC, del inglés *Model Predictive Control*) ha experimentado un gran desarrollo en las últimas décadas y se ha convertido en una de las técnicas de control avanzado más populares en la comunidad científica y la industria. Varios factores contribuyen al éxito de MPC, entre los más importantes, la formulación intuitiva del problema de control, la posibilidad de controlar una gran variedad de procesos, la consideración de restricciones y, al menos en el caso de MPC lineal, la fácil implementación de la ley de control. La posibilidad de utilizar modelos que se pueden obtener fácilmente del proceso considerado, p.ej. modelos de respuesta ante impulso o escalón, representa otra ventaja de MPC, especialmente en aplicaciones industriales.

Prácticamente todos los procesos dinámicos de importancia industrial exhiben cierto comportamiento no lineal. Sin embargo, la gran mayoría de técnicas de MPC han sido desarrolladas para sistemas lineales. La aplicación de MPC lineal a procesos con dinámica fuertemente no lineal puede resultar en un rendimiento de control deficiente, debido a la discrepancia entre el sistema y el modelo. Con el fin de obtener un mejor rendimiento de control, se puede considerar el uso de control predictivo no lineal (NMPC, del inglés *Nonlinear Model Predictive Control*). Otro problema en el control de procesos es la influencia de perturbaciones exógenas o incertidumbres que pueden desestabilizar el sistema en bucle cerrado en ciertas ocasiones o, al menos, resultar en un rendimiento de control insuficiente. El control predictivo mín-máx (MMMPC, del inglés *Min-Max Model Predictive Control*), basado en un modelo que considera explícitamente incertidumbres y perturbaciones, da la oportunidad de prevenir este efecto no deseado y obtener un control más robusto.

Desafortunadamente, tanto el NMPC como el MMMPC tienen algunos inconvenientes para su uso en la industria. En el caso de NMPC hay que mencionar la dificultad de obtener modelos apropiados del proceso considerado y la solución necesaria de un problema de optimización posiblemente no convexo. Por otro lado, el principal inconveniente de MMMPC es la carga computacional, resultado del cálculo de la señal

de control, que crece exponencialmente con el horizonte de predicción. Además, ambas estrategias de control complican considerablemente el estudio de cuestiones teóricas como estabilidad, robustez y optimalidad. A pesar de que hay evidencia de que estas estrategias habitualmente mejoran el rendimiento de control, el uso de NMPC y MMMPC en la industria es muy escaso y se reduce a unas pocas aplicaciones debido a los problemas anteriormente mencionados.

Esta tesis aborda el estudio y el desarrollo de novedosas técnicas de MPC basado en modelos no lineales e inciertos, es decir, el marco general cubre el NMPC y el MMMPC. En el desarrollo se ha prestado especial atención a la aplicabilidad de las estrategias de control, con la idea de reducir la brecha entre la práctica industrial y la investigación académica. En el lado teórico, el tema de estabilidad juega un importante papel e incluye la especificación de las condiciones necesarias y suficientes para obtener estabilidad en bucle cerrado.

El objetivo principal es el desarrollo de nuevas estrategias de MPC basado en los modelos de Volterra en tiempo discreto y su posible uso en estrategias de control de horizonte deslizante. El punto de partida ha sido un algoritmo iterativo y computacionalmente eficiente, usado para resolver el problema de optimización de una estrategia de NMPC basado en modelos de Volterra de segundo orden. Se ha mejorado esta estrategia con la consideración de restricciones y la ponderación del esfuerzo de control. Por otra parte, para el NMPC basado en modelos de Volterra de segundo orden se ha desarrollado un novedoso enfoque basado en la convexificación del problema inicial y posiblemente no convexo. Este enfoque determina una envoltura convexa del problema de optimización y da la posibilidad de una minimización global.

Además, se han considerado modelos inciertos lineales dentro del marco del control robusto como paso previo al desarrollo de una estrategia de MMMPC basado en modelos de Volterra. En este caso, se ha demostrado la estabilidad robusta de una estrategia de MMMPC lineal basado en una cota superior no lineal del peor caso. Con el fin de conseguir un control más robusto, se ha desarrollado una novedosa estrategia de MMMPC no lineal basado en modelos de Volterra de segundo orden con perturbaciones aditivas y persistentes. Debido al carácter no autorregresivo del modelo, se ha obtenido una formulación explícita del coste del peor caso y, en consecuencia, el problema de optimización mín-máx se reduce a un mero problema de minimización.

Por último, se han implementado las estrategias de control desarrolladas y se ha validado la aplicabilidad práctica en experimentos con diferentes sistemas usados como banco de pruebas. En comparación con un MPC lineal, las diferentes estrategias de NMPC y MMMPC mostraron un rendimiento de control superior y dieron mejores resultados. El empleo exitoso de las diferentes estrategias se une al bajo número de aplicaciones del NMPC y el MMMPC existentes en la literatura especializada.

# Contents

# Glossary

### Notation

| | |
|---|---|
| $\mathbf{A}$ | boldface upper case letters denote operators mapping input to output. |
| $A(\cdot)$ | italic upper case letters denote polynomials. |
| $A$ | italic upper case letters denote matrices. |
| $\boldsymbol{a}$ | boldface italic lower case letters denote vectors. |
| $a$ | italic lower case letters denote scalars. |
| $\mathbf{a}(\cdot)$ | boldface lower case letters denote functions generating a column vector. |
| $\mathrm{a}(\cdot)$ | lower case letters denote functions generating a scalar. |

### Symbols

| | |
|---|---|
| $q^{-1}$ | backward shift operator |
| $a^*$ | optimal value of $a$ |
| $a^f$ | feasible value of $a$ |
| $\|\boldsymbol{a}\|_Q^2$ | $\boldsymbol{a}^T Q \boldsymbol{a}$ |
| $\|\boldsymbol{a}\|_\infty$ | infinity norm of $\boldsymbol{a}$ |
| $\|\boldsymbol{a}\|_1$ | 1-norm of $\boldsymbol{a}$ |
| $\tilde{a}$ | approximated value of $a$ |
| $\boldsymbol{a}^T$ | transpose of $\boldsymbol{a}$ |
| $a^{(j)}$ | value of $a$ in the $j$-th iteration |
| $\hat{a}$ | expected value of $a$ |
| $\hat{a}(k+i\|k)$ | expected value of $a(k+i)$ with available information at instant $k$ |
| $\mathrm{vert}\{\boldsymbol{\mathcal{A}}\}$ | set of vertices of $\boldsymbol{\mathcal{A}}$ |
| $\mathrm{trace}(A)$ | trace of the matrix $A$ |
| $A^{[j]}$ | $j$-th row of the matrix $A$ |
| $A^{[jk]}$ | element $jk$ of the matrix $A$ |

| | |
|---|---|
| $I_n$ | identity matrix in $\mathbb{R}^{n\times n}$ |
| $I$ | identity matrix of appropriate dimensions |
| $\mathbf{0}_{m\times n}$ | matrix in $\mathbb{R}^{m\times n}$ with all entries equal to zero |
| $\mathbf{0}$ | matrix of appropriate dimensions with all entries equal to zero |

## Variables

| | |
|---|---|
| $A$ | system matrix |
| $\boldsymbol{b}_c$ | vector of the set of constraints |
| $B$ | input matrix |
| $B$ | matrix to calculate quadratic terms of Volterra series prediction model |
| $\boldsymbol{c}$ | constant vector of Volterra series prediction model |
| $d$ | estimation error |
| $\boldsymbol{d}$ | vector containing the estimation error |
| $D$ | matrix of linear model to include influence of disturbance |
| $e$ | error |
| $\mathbf{f}(\boldsymbol{u})$ | vector of Volterra series prediction model with future-future and future-past cross terms in $\boldsymbol{u}$ |
| $\boldsymbol{g}$ | vector of Volterra series prediction model with past-past terms in $\boldsymbol{u}_p$ |
| $G$ | matrix of Volterra series prediction model to include influence of $\boldsymbol{u}$ |
| $h_0$ | constant offset of Volterra series model |
| $h_1$ | linear term parameter of Volterra series model |
| $h_2$ | nonlinear term parameter of Volterra series model |
| $H$ | matrix of Volterra series prediction model to include influence of $\boldsymbol{u}_p$ |
| $J(\cdot)$ | cost function (performance index) |
| $K$ | feedback matrix |
| $L_c$ | matrix of the set of constraints |
| $L_u$ | transformation matrix |
| $N$ | prediction horizon |
| $N_t$ | truncation order |
| $N_u$ | control horizon |
| $Q$ | weighting matrix |
| $r$ | reference signal |
| $\boldsymbol{r}$ | reference trajectory |
| $R$ | weighting matrix |
| $t_c$ | computation time |

| | |
|---|---|
| $t_m$ | sampling time |
| $u$ | input signal |
| $u_r$ | steady-state input signal |
| $\boldsymbol{u}$ | future input sequence |
| $\boldsymbol{u}_l$ | vector with first element corresponding to last applied control signal, other entries equal to zero |
| $\boldsymbol{u}_p$ | vector of past input values |
| $\boldsymbol{u}_r$ | vector containing the steady-state input signal |
| $\boldsymbol{u}_0$ | initial input sequence |
| $\Delta u$ | input increment |
| $\boldsymbol{\Delta u}$ | sequence of control increments |
| $\boldsymbol{\Delta u}_0$ | initial input increment sequence |
| $v$ | control input in semi-feedback case |
| $W$ | matrix of Volterra series prediction model to include influence of disturbance |
| $y$ | system output |
| $\boldsymbol{y}$ | output prediction along the prediction horizon |
| $\lambda$ | weighting factor for the control effort |
| $\theta$ | disturbance |
| $\boldsymbol{\theta}$ | disturbance sequence |

**Acronyms**

| | |
|---|---|
| ARMAX | Autoregressive Moving Average Model with Exogenous Inputs |
| CARIMA | Controlled Autoregressive Integrated Moving Average |
| CPU | Central Processing Unit |
| DMC | Dynamic Matrix Control |
| FIR | Finite Impulse Response |
| GPC | Generalized Predictive Control |
| ISS | Input-to-State Stability |
| LMI | Linear Matrix Inequality |
| LTI | Linear Time-Invariant |
| MAC | Model Algorithmic Control |
| MISO | Multiple-Input Single-Output |
| MMMPC | Min-Max Model Predictive Control |
| MPC | Model Predictive Control |

| | |
|---|---|
| MPHC | Model Predictive Heuristic Control |
| MSE | Mean Square Error |
| NIMC | Nonlinear Internal Model Control |
| NLP | Nonlinear Programming |
| NMPC | Nonlinear Model Predictive Control |
| OLE | Object Linking and Embedding |
| OPC | OLE for Process Control |
| PAC | Programmable Automation Controller |
| PCI | Peripheral Component Interconnect |
| PEM | Polymer Electrolyte Membrane, also Proton Exchange Membrane |
| PRBS | Pseudo-Random Binary Sequence |
| PRMS | Pseudo-Random Multilevel Sequence |
| QCQP | Quadratically Constrained Quadratic Programming |
| QP | Quadratic Programming |
| SISO | Single-Input Single-Output |
| SQP | Sequential Quadratic Programming |
| SSE | Sum of Square Errors |

# Chapter 1

# Introduction

## 1.1 Motivation

Many modern control techniques are based on mathematical models which approximate the dynamic evolution of the system to be controlled. The mathematical model is used by the controller to compute the control action satisfying a certain criterion. Consequently, the quality of the mathematical model used to approximate the system dynamics has a decisive influence on the control performance. In Model Predictive Control (MPC) the computation of the control action is based on the evolution of the system predicted by means of the mathematical model. Nowadays, MPC represents one of the most common advanced control techniques applied to industrial processes [23, 75]. The widespread employment of MPC techniques is the result of several factors, amongst others the intuitive formulation of the control problem, the possibility to control a great variety of processes, the consideration of constraints in the computation of the input signal and, in the case of MPC based on linear models, the easy implementation of the resulting control law. Another advantage is the possibility to use mathematical models which can be obtained easily from the process, e.g. step response models in Dynamic Matrix Control (DMC) [31] or impulse response models in Model Predictive Heuristic Control[1] (MPHC) [111, 112]. Different studies [104, 130, 13] showed that MPC represents one of most commonly used advanced control techniques and underline the popularity of MPC in industry.

Virtually all dynamic processes of practical importance exhibit some degree of nonlinear behavior [97]. Nevertheless, the grand majority of control techniques has been

[1]The Model Predictive Heuristic Control is also known as Model Algorithmic Control (MAC).

developed for systems with linear dynamics. The linear control techniques usually perform well when the considered process only possesses a weak nonlinear dynamic. Besides, process control in the neighborhood of a nominal operating point results in many cases in a good control performance. The predominant use of linear control techniques is a consequence of the oftentimes good results and the easy implementation of these techniques. Furthermore, linear models can be obtained with common control engineering methods, e.g. step or impulse response (which can be used in DMC or MAC, respectively) of a system in a certain operating point or employment of well known linear identification methods to obtain a transfer function from experimental data. Additionally, for linear control techniques a large variety of theoretical results is available, discussing themes as stability, optimality and robustness [23, 75, 85, 88].

However, in many cases the performance of a linear controller applied to a nonlinear dynamic process is not very efficient. This poor control performance is usually a consequence of strong nonlinearities in the regarded process, especially when a variable operating point is used. With a linear model determined for a certain operating point, the used model is not capable to approximate the process with sufficient quality in a wider range of operation. Besides, some processes exhibit strong nonlinearities even in the vicinity of an operating point and have a negative effect on the behavior of the closed-loop system. The disregard of the nonlinear dynamics of a process frequently results in an unacceptable control performance and, in the worst case, to a destabilization of the controlled system. Furthermore, it has to be mentioned that some processes are operated continuously in a transient mode, e.g. batch processes which are never in steady-state operation, or experience operating modes far away from steady state at least during some periods, e.g. startup and shutdown. In the mentioned cases, with the objective of a better control performance, the use of control techniques based on nonlinear models should be considered. However, from the idea of using nonlinear models in process control several new questions arise. One of the most important issues when dealing with nonlinear models for control purposes is the choice of a suitable model structure. This election not only defines implicitly the capability to approximate the process nonlinearities, i.e. not all model structures are suitable to approximate certain nonlinearities, but also influences in a high degree the control technique used to compute the control action.

Naturally, the main advantage of nonlinear models with respect to linear models is the possibility to deal with the nonlinear dynamics of the process. For the approximation of nonlinear dynamics, the user can choose from a huge variety of different nonlinear models. It has to be emphasized that nonlinear models do not represent a homogenous group and cannot be classified as easy as linear models. The difference between two nonlinear models can be bigger than the difference between a nonlinear and a linear model. Furthermore, many nonlinear models for control purposes are much

more difficult to obtain than linear models (e.g. step or impulse response models), either from input/output data correlation or with first principles considering mass and energy conservation laws [23]. Another problem when dealing with nonlinear models is the possible identification of non-existent dynamics and noise characteristics as a consequence of an over-parametrization or a higher degree of freedom of the chosen nonlinear model structure. Additionally, the interpretation and understanding of nonlinear models is frequently more difficult than in the case of linear models and often requires a profound knowledge of the regarded process.

The basic concept of MPC offers the possibility to use nonlinear models to predict the future evolution of the system. Hence, MPC in combination with nonlinear models, called Nonlinear Model Predictive Control (NMPC), can be employed to compute the control action. Although the idea behind MPC does not exclude nonlinear models, the use of this kind of models entails several problems. From a practical point of view, the consideration of nonlinear models in a quadratic cost function leads to a possibly non-convex optimization problem with several minimums. While linear MPC requires in every sampling period the solution of a convex problem, usually carried out with Quadratic Programming (QP), NMPC requires (at least a partial) solution with the help of nonlinear programming (NLP) [17]. The difficulty of the optimization problem results in an important increase in the computation time, limiting the use of NMPC in many cases to slow processes or the consideration of small horizons. Besides, from a more theoretical viewpoint, the possible non-convexity of the optimization problem considerably complicates the study of stability and robustness. Furthermore, as already mentioned, nonlinear models can differ in a high degree between each other, requiring frequently custom-tailored approaches both for the solution of the problem and for the study of the theoretical aspects. The mentioned problems account for the seldom use of NMPC in industry where only a few NMPC applications have been reported, including the areas of refining, chemicals, polymers as well as air and gas processing [104]. The computational burden, resulting from non-convex optimization problems with the possibility of several minimums, converts the application of NMPC to a difficult task with many unresolved issues as stability, robustness and others.

The changeover from linear to nonlinear models allows to approximate processes in a wider range of operation and to consider a richer dynamic behavior of the system. But even with complex nonlinear models it is difficult to capture the complex dynamics of a process. Furthermore, both linear and nonlinear models ignore the possible influence of unmodeled disturbances on the future evolution of the system. Even though stability of MPC or NMPC can be proven, the existence of uncertainties or disturbances can destabilize the closed-loop system under certain conditions or, at least, result in an insufficient control performance [23, 118]. In order to prevent this effect, uncertainties or disturbances can be considered explicitly in the formulation of the model used to

predict the future evolution of the system, being parameter uncertainties and bounded exogenous disturbances two of the most common descriptions used to account for model uncertainty [133]. However, the use of an uncertain model for prediction purposes does not result in an unique future trajectory, but in the generation of a family of trajectories. A key issue when dealing with uncertain models, both linear and nonlinear, is the choice of a suitable realization of the uncertainty or disturbance. This choice has to assure that the dynamics of the process can be approximated sufficiently well by the chosen uncertain model structure, otherwise none of the predictions will correspond to the future evolution of the system.

Analogously to the nonlinear models, a wide range of uncertain models can be considered in MPC to predict the evolution of the considered process. However, the extension of MPC to compute the input action based on the future system evolution predicted by an uncertain model is not trivial and leads both to computational and theoretical difficulties. In the case of bounded uncertainties the resulting family of trajectories is also bounded. This bound represents the worst case with respect to the uncertainty and the minimization of the associated cost by a suitable choice of the input action results in a more robust control. The minimization of the worst case cost in order to compute the control action is known as Min-Max Model Predictive Control (MMMPC) [25]. The main drawback of this approach is the computational burden that takes to compute the control signal, with a computational complexity growing exponentially with the considered prediction horizon. As a result, the number of applications of MMMPC is very small, even when there is evidence that the min-max approach usually performs better than standard MPC in processes with uncertain dynamics. On the theoretical side, the considered uncertainty complicates the study of stability, representing a field with many open issues.

This thesis addresses the study and development of novel MPC techniques based on nonlinear and uncertain models, i.e. the general framework covers both NMPC and MMMPC. Due to the above mentioned problems when employing nonlinear or uncertain models, particular attention will be paid to the applicability of the resulting MPC techniques. On the more theoretical side, stability issues will play an important role in the design of new MPC techniques. The concurrent achievement of stability and applicability is a challenging task since in many cases the necessary conditions to assure stability counteract applicability. In the case of MMMPC, the main focus will be on the reduction of the computational burden associated to the maximization problem, i.e. the determination of the worst case, in order to allow the use of longer prediction horizons. For the design of NMPC techniques, the attention will be on the solution of a possibly non-convex cost function.

It is clear that the election of a certain model structure has a decisive effect on the later development of novel MPC strategies. In this thesis, the principally used model structure is a discrete-time nonlinear Volterra series model [132], with a special emphasis on the second order type. The decision to use Volterra series models for the approximation of complex system dynamics lies in the particular model structure which represents the natural extension of a linear convolution model with the nonlinearity considered in an additive term. This structure, facilitating the separation of linear and nonlinear terms, can be exploited in the development of MPC techniques based on Volterra series models. The second model, a linear model in state-space representation, is one of the most common models used in control engineering. Although this model has not the ability to approximate nonlinear dynamics, the explicit consideration of uncertainties can be used to achieve a good control performance in presence of process nonlinearities and disturbances. The use of the mentioned linear model in this thesis can be considered as a necessary approach to extend the concept of MMMPC to MPC based on Volterra series models.

## 1.2 State of the art

Volterra series models are used in a wide range of areas to approximate the dynamics of nonlinear processes, amongst others in biomedical applications [58, 78, 87, 10], acoustics [128, 60], electronics [110, 136] and process control [38, 4, 35], but especially in signal processing for nonlinear filter design [94, 56]. An extensive bibliographic list on nonlinear system identification in signal processing, including Volterra series models, can be found in [42]. The interest to use Volterra series models to approximate the nonlinear dynamic behavior in the different areas has several reasons, probably the most important ones are:

- Volterra series models, although being nonlinear models, are linear in the parameters. As a consequence, parameter estimation for Volterra series models can be carried out with identification techniques usually used for linear models, e.g. least squares methods.

- The parameters of Volterra series models can be estimated from experimental input-output data and hence, a deep knowledge of the process to be approximated is not required.

- Volterra series models can be used to model a great variety of different nonlinear dynamics, e.g. non-minimum-phase systems.

Besides the above mentioned reasons, Volterra series models have some characteristics especially interesting for control purposes:

- Volterra series models are the logical extension of finite impulse response (FIR) models which have been used widely in linear MPC. Being a generalization of FIR models, Volterra series models show a similar qualitative behavior.
- The nonlinearity of Volterra series models is considered in an additional term. This particular structure allows the separation of linear and nonlinear terms and can be exploited in the design of control techniques.

One of the main drawbacks of Volterra series models is the elevated number of parameters to describe the nonlinear dynamics of a process which results in the requirement of large data sets for identification purposes. Another important issue is the impossibility to use Volterra series models to approximate unstable processes, limiting their use to the approximation of stable systems.

It is obvious that the mentioned reasons convert Volterra series models in a good candidate for the development of NMPC techniques. The flexibility of this model type allows the modeling of dynamic processes in a wide range of different areas without the necessity of a deep understanding of the regarded system [120, 38]. A large class of nonlinear system models are covered by Volterra series models including bilinear models which have under certain conditions equivalent Volterra model representations [70]. It has been shown in [20] that Volterra series model can be used to approximate arbitrarily well any stable system with fading memory characteristics where a better approximation requires higher degree models with longer memory sequences. In practice, a trade-off between the number of parameters and the quality of approximation has to be found.

The parameters of Volterra series models are frequently identified with experimental input-output data from the process to be approximated. For a proper identification, [96] propose Pseudo-Random Multilevel Sequences (PRMS) which sufficiently excite the regarded process in order to obtain suitable data. In [101] input-sequences for the separate identification of linear and nonlinear term parameters are presented. These input sequences are supposed to be plant-friendly as they possess a reduced number of transitions. A constrained multisine input signal is proposed in [113], leading to suitable data for identification purposes in an acceptable time period while keeping the variation in both input and output signals within user-defined constraints.

In [52, 71] the identification of autoregressive models with a reduced number of model parameters has been proposed. Then, the Volterra series model parameters can

be computed from the identified autoregressive model. Another approach is the reduction of the parametric complexity by using orthonormal basis functions like Laguerre [24] and Kautz functions [32] or generalized orthonormal basis functions [124, 92]. In [70], using Nonlinear Internal Model Control (NIMC), the closed-loop performance is related to the open-loop modeling error. Based on this relation, an optimization method is proposed which allows a control relevant reduction of a Volterra series model.

### 1.2.1 NMPC based on Volterra series models

Although Volterra series models can be used to approximate a wide range of different systems, only a few research articles on NMPC strategies based on Volterra series models have been published. Analogously and in consequence to the scarce research on NMPC using Volterra series models, only a small number of practical applications have been reported.

With respect to the use of Volterra series models in NMPC, [37, 77] proposed a computationally efficient optimization method to calculate the input sequence. The unconstrained optimization is carried out by means of an iterative algorithm based on the separation of the linear and the nonlinear term of a second order Volterra series model. The control strategy consists of a conventional linear controller extended by an auxiliary loop to include the nonlinear dynamics in the optimization process. The NMPC strategy is validated in [37] by means of simulation case studies with a polymerization reaction in a continuous stirred tank reactor (CSTR) and an isothermal reaction based on van de Vusse kinetics. The capability of the proposed NMPC strategy to control Multiple-Input Multiple-Output (MIMO) systems has been shown in [77] with the simulation of a multivariable CSTR. The mentioned NMPC strategy can be considered as a starting point in this thesis for the development of novel NMPC techniques based on Volterra series models. The iterative optimization approach has been modified in [51] to consider autoregressive Volterra series models for the prediction of the future evolution of the system. The modified NMPC was applied in a simulation case study to a Single-Input Single-Output (SISO) two-level tank system and showed the viability of the modified control strategy.

In [50] two similar suboptimal NMPC strategies for second order Volterra series models have been presented. Here, the first strategy is based on the assumption of constant increments in the input signal over the entire control horizon, i.e. the resulting input sequence has a constant slope. The second strategy is a stair-like control where the input signal is assumed to be constant along the control horizon. In both NMPC strategies, depending only on one sole decision variable (constant slope or constant

input signal), the possibly non-convex optimization problem is converted to a forth order polynomial. The resulting one-dimensional optimization problem, which can still be non-convex, can be easily minimized with respect to the decision variable. The cost function is then evaluated for all values of the decision variable associated to non-complex minimums and the value of the decision variable corresponding to the global minimum of the cost is chosen. The reduced optimization problem allows to compute analytically the value of the decision variable which minimizes the cost and, as a consequence, results in a very fast computation of the input sequence. Independently from the chosen optimization method, analytical or numerical, input constraints can be included in the minimization of the cost.

In [36, 35] the unconstrained NMPC proposed in [37, 77] was extended to consider the weighting of the control effort in the cost function. The inclusion of a control effort weighting does not change the computational complexity of the optimization problem and, as a consequence, does not change the efficiency of the iterative optimization. The problem of convexity of the optimization problem under consideration of a weighting of the control effort was studied in [35]. It was shown that convexity of the optimization problem can be assured by a suitable choice of the weighting factor leading to a guaranteed convergence of the iterative algorithm. Furthermore, [35] compares the control performance of the proposed NMPC for processes with dead time when the dead time is considered implicitly in the Volterra series model or a separate dead time compensator is used.

The NMPC for second order Volterra series models [37, 77] was generalized in [66] for its use in combination with higher order Volterra series models. Analogously to the original optimization approach, the optimization problem is solved by an iterative optimization algorithm based on the separation of the linear and nonlinear terms. In difference to the original NMPC strategy [37, 77] where the second order term of the Volterra series model is considered in the auxiliary loop, [66] proposes to include also third and higher order terms in the auxiliary loop. The NMPC strategy was validated in a case study of a nonlinear polymerization process identified by a third order Volterra series models. A comparison with a linear MPC and an NMPC strategy based on a second order Volterra series models showed that the use of a higher order Volterra series improves the control performance when highly nonlinear processes are considered.

Recently, [4] applied an NMPC strategy based on a second order Volterra series model to a detailed simulation model of a crude oil processing facility. The considered process is a MIMO system with 2 inputs and 4 outputs, i.e. the process represents a thin or underactuated system, and exhibits a strong coupling between the different states. The applied NMPC did not achieve a stabilization of the 4 states in the given

references due to the complexity of the considered process and the interconnection of the states. In a second step, only a subsystem with 2 inputs and 2 outputs was considered. The applied NMPC strategy based on second order Volterra series models stabilized the system in the given references in setpoint tracking and disturbance rejection simulations. A traditional PI control applied to the same system failed to adequately handle the coupling of the system states and obtained unacceptable results with important oscillations in the inputs and outputs.

With respect to the stability of NMPC strategies it is important to have in mind that Volterra models represent a generalization of FIR models and can be considered as stable fading memory systems. For MPC based on linear FIR models, several authors have proven closed-loop stability, amongst others [41, 135, 91]. Based on the finite memory of Volterra series models, [85] proposes to use a terminal equality constraint to ensure stability. Other publications, such as [82, 86, 33] point out that general stability of NMPC strategies can be proven under certain conditions using Lyapunov functions. In spite of the mentioned propositions, stability of NMPC based on Volterra series models is an open field with few results.

### 1.2.2 Min-max MPC

Even with complex mathematical models it is difficult to capture the dynamics of a physical process. The possible model mismatch and external disturbances lead to a deficient prediction of the future evolution of the considered system. In order to obtain a more robust control performance, uncertain models can be used in an MMMPC framework, initially proposed by [134]. Different MMMPC strategies can be found, based on the predicted open-loop or closed-loop evolution of the system. The open-loop MMMPC strategy minimizes the worst case without a feedback of the computed predictions [25, 5]. The imposed constraints have to be satisfied for all possible trajectories of the evolution of the system, resulting in a considerably conservative control performance. The complexity of the resulting optimization problem depends exponentially on the used prediction horizon, i.e. on the length of the considered disturbance vector, and represents an NP-hard problem [73]. The open-loop approach was one of the first MPC strategies based on a min-max formulation of the optimization problem.

The conservativeness of the control law can be reduced using a closed-loop formulation of the MMMPC strategy, proposed in [123]. In the closed-loop approach the considered problem is minimized under explicit consideration of a feedback of the predicted evolution of the system [85]. The feedback approach results in an infinite dimensional optimization problem and is obviously far more complex than the optimization pro-

blem corresponding to the open-loop MMMPC. Generally, the input sequence is very difficult to compute and only a few algorithms for linear systems can be found [89, 43]. The high computational burden limits the application of the closed-loop MMMPC strategy to simple processes or low prediction horizons. Note that in this thesis only the open-loop MMMPC strategy will be considered.

Recently it was shown that the open-loop MMMPC control law based on linear models is piecewise affine when a 1-norm based criterion [14, 55] is used in the cost function. Analogously, [106] proved piecewise affinity for quadratic objective functions. This property enables the possibility to build explicit forms of the control law with a reduced complexity [90]. Another common approach is the use of an upper bound of the worst case cost instead of computing it explicitly. The solution to the optimization problem can then be computed by means of Linear Matrix Inequalities [59, 74]. In the case of upper bounds, the difference between the exact solution and the computed upper bound augments the conservativeness of the resulting control law.

For feedback MMMPC, the optimization problem based on convex costs and constraints can be casted as a finite dimensional convex optimization problem [34]. In the special case of a quadratic cost the optimization problem results in a Quadratically Constrained Quadratic Program (QCQP). For the sake of completeness tube-based robust MPC, which allows the computation of the control signal by solving a standard QP problem in every sampling period, has to be mentioned [83, 84, 69]. The concept of tubes was introduced in [16] and is based on the idea of a sequence of sets where every set can be reached from the previous one. Tube-based MPC can be used for robust constraint fulfillment both for linear [27] and nonlinear systems [79, 22].

In [81] sufficient conditions have been presented to design an asymptotically stabilizing MMMPC in case of state-dependent uncertainties which disappear when the system reaches its equilibrium. In the case of persistent uncertainties, Input-to-State Stability (ISS) is a suitable framework for the analysis of the stabilizing properties of MPC [126, 54]. The sufficient conditions for stability of uncertain systems controlled by MMMPC for a general class of bounded uncertainties are presented in [68]. In [62] under consideration of the a priori sufficient conditions for robust stability, a new approach for the design of closed-loop MMMPC schemes for nonlinear systems with guaranteed ISS is presented. Currently, a considerable research effort is devoted to the stability analysis of discrete uncertain systems. For an overview on robust stability issues the reader is referred to [67].

## 1.3 Objectives

The main objective of this thesis is to contribute to the development of novel nonlinear and robust MPC strategies. The main focus will be on discrete-time Volterra series models and their possible use in receding horizon control strategies. In addition, discrete-time uncertain linear models will be considered within the framework of robust MPC as a previous step to an MMMPC based on Volterra series models. Particular attention in the design of new control strategies will be paid to practical applicability, related to the computational burden, and stability issues.

In spite of the importance of nonlinear processes in the industry, robust and nonlinear MPC strategies are rarely used in industrial applications. Furthermore, in the area of advanced MPC techniques a significant gap between the academic research and the industrial practice can be observed. One of the objectives of this thesis is the development of novel MPC techniques which fulfill the requirements of industrial applications. An important reason for the scarce success in the industrial practice can be found in the difficulty to obtain suitable prediction models. Therefore, the models considered in this thesis are easily obtainable from experimental input-output data, a common practice in industry. In the case of Volterra series models, being the logical extension of linear convolution models, the parameters can be identified with linear identification techniques. In the case of uncertain linear models, uncertainties have been considered as a persistent bounded additive term where the bound can be obtained directly from the comparison of the system output and the predicted output.

Another problem when applying advanced MPC techniques is the computational complexity of the resulting optimization problem. In the case of NMPC, the computation of the input action requires the solution of a possibly non-convex problem. Here, the particular structure of Volterra series models, i.e. the separability of linear and nonlinear terms, can be exploited to find computationally efficient algorithms to solve the optimization problem. In MMMPC, the numerical complexity lies in the computation of the worst case and growths exponentially with the prediction horizon. With the practical applicability in mind, new approaches for the determination of the worst case will be studied. Furthermore, the new control strategies should allow an easy implementation and, as a consequence, facilitate their use in industrial applications.

Naturally, stability is a fundamental issue in the design of new MPC techniques. For the practical implementation of MPC techniques, stability is especially relevant when disturbances are considered explicitly. Both for MMMPC and NMPC, the study and proof of stability will be an important objective in this thesis.

Summarizing the above mentioned objectives, this thesis will cover the following themes:

- Development of novel NMPC and MMMPC strategies based on Volterra series models and uncertain linear models
- Study of the stabilizing properties of the control strategies and formulation of the necessary conditions to achieve stability
- Implementation of the different MPC strategies and application to nonlinear industrial processes

## 1.4 Thesis overview

The thesis is organized as follows:

- Chapter 2 gives a general introduction to Volterra series models with an emphasis on non-autoregressive discrete-time second order Volterra series models and their use in MPC. The parameter identification from empirical data is explained and the transformation from an autoregressive to a non-autoregressive representation is shown. For the second order Volterra series model a prediction model for its use in MPC is presented. Then, the usually quadratic cost function of MPC strategies is given and the general optimization problem for MPC based on second order Volterra series models is defined. The last section presents the state-space representation of Volterra series models, important with a view to the necessary stability proofs.
- Chapter 3 gives a detailed description of different benchmark systems used in this thesis for the application of MPC strategies. The benchmarks systems include a pilot plant, a fuel cell and a greenhouse and represent processes with nonlinear dynamics. The nonlinear process dynamics are approximated by second order Volterra series models for their posterior use in MPC strategies. The model parameters are identified from input-output data obtained in experiments.
- Chapter 4 deals with iterative algorithms to solve the optimization problem of an NMPC based on second order Volterra series models. Starting with a basic unconstrained formulation of the iterative algorithm, different modifications are introduced in the cost function and the iterative algorithm in order to consider constraints and a weighting of the control effort. In a second step, some changes will be introduced in the constrained iterative algorithm to guarantee stability.

The constrained iterative algorithms are applied to different benchmark systems and the control performance is illustrated by experimental results.

- In Chapter 5 a convexification approach for NMPC problems based on second order Volterra series models is presented. The presented approach allows the global minimization of the optimization problem under consideration of a suitable weighting of the control effort. After some changes in the convexification approach stability of the resulting NMPC strategy is assured. The practical applicability of the developed NMPC strategy is verified in experiments with a benchmark system.

- Chapter 6 presents two MMMPC strategies for uncertain linear systems. The first strategy uses a nonlinear upper bound of the worst case and reduces considerably the computational complexity of the optimization problem. In the case of the second strategy, a close approximation of the solution of the min-max problem is computed using a QP problem. Both control strategies have a considerably lower computational burden than the original optimization problem and stability can be proven. Finally, the MMMPC strategies are applied to a benchmark system and the obtained results are presented.

- In Chapter 7 an MMMPC strategy based on second order Volterra series models with additive and persistent uncertainty is given. It is shown that the non-autoregressive character of the used model allows the explicit formulation of the worst case. The computational burden to compute a new input sequence is much lower as the min-max optimization problem degenerates to a minimization problem. Introducing some changes in the cost function stability of the MMMPC strategy is proven. The MMMPC strategies based on second order Volterra series models are implemented and applied to a benchmark system. The control performance is illustrated by experimental results.

- Chapter 8 summarizes the contributions and results presented in this thesis and gives possible directions for future research activities.

- The Appendix A gives a detailed definition of the used prediction model based on a second order Volterra series model as well as other mathematical definitions used throughout this thesis. In the Appendix B the Spanish translations of the Chapters 1 and 8 can be found.

## 1.5 List of publications

The following articles have been issued or submitted for publication during the elaboration of this thesis:

**Book chapters:**

1. J. K. Gruber, C. Bordons (2010). Regulation of the air supply in a fuel cell using Model Predictive Control. *Power Plant Applications of Advanced Control Techniques*. ProcessEng Engineering GmbH, Vienna, Austria. Submitted for publication.

**Journal papers:**

1. J. K. Gruber, J. L. Guzmán, F. Rodríguez, C. Bordons, M. Berenguel, J. A. Sánchez (2010). Nonlinear MPC based on a Volterra series model for greenhouse temperature control using natural ventilation. *Control Engineering Practice*. Submitted for publication.

2. J. K. Gruber, A. Oliva, C. Bordons (2010). Nonlinear MPC for the airflow in a PEM fuel cell using a Volterra series model. *Control Engineering Practice*. Submitted for publication.

3. J. K. Gruber, C. Bordons, R. Bars, R. Haber (2009). Nonlinear predictive control of smooth nonlinear systems based on Volterra models. Application to a pilot plant. *International Journal of Robust and Nonlinear Control*. DOI: 10.1002/rnc.1549. Accepted for publication.

4. J. K. Gruber, D. R. Ramírez, T. Alamo, C. Bordons, E. F. Camacho (2009). Control of a pilot plant using QP based min-max predictive control. *Control Engineering Practice* **17**(11): 1358-1366.

5. J. K. Gruber, M. Doll, C. Bordons (2009). Design and experimental validation of a constrained MPC for the air feed of a fuel cell. *Control Engineering Practice* **17**(8): 874-885.

6. D. R. Ramírez, J. K. Gruber, T. Álamo, C. Bordóns, E. F. Camacho (2008). Control Predictivo Mín-Máx de una Planta Piloto (in Spanish). *Revista Iberoamericana de Automática e Informática Industrial* **5**(3): 37-47.

7. J. K. Gruber, C. Bordons (2007). Control predictivo no lineal basado en modelos de Volterra. Aplicación a una planta piloto (in Spanish). *Revista Iberoamericana de Automática e Informática Industrial* **4**(3): 34-45.

**Conference papers:**

1. A. Ferramosca, J. K. Gruber, D. Limon, E. F. Camacho (2010). MPC for tracking of constrained nonlinear systems. Application to a pilot plant. Submitted to *49th IEEE Conference on Decision and Control*, Atlanta, USA.

2. J. K. Gruber, T. Alamo, D. R. Ramírez, C. Bordons (2010). A convex approach for NMPC based on second order Volterra series models. Submitted to *49th IEEE Conference on Decision and Control*, Atlanta, USA.

3. J. K. Gruber, D. R. Ramírez, T. Alamo, C. Bordons (2009). Nonlinear Min-Max Model Predictive Control Based on Volterra Models. Application to a Pilot Plant. In *Proceedings of the 2009 European Control Conference*, Budapest, Hungary.

4. J. K. Gruber, J. L. Guzmán, F. Rodríguez, M. Berenguel, C. Bordóns (2009). Nonlinear Model Predictive Control of Greenhouse Temperature Using a Volterra Model. In *Proceedings of the 2009 European Control Conference*, Budapest, Hungary.

5. J. K. Gruber, C. Bordons, F. Dorado (2008). Nonlinear control of the air feed of a fuel cell. In *Proceedings of the 2008 American Control Conference*, Seattle, USA.

6. J. K. Gruber, D. R. Ramírez, T. Alamo, C. Bordons, E. F. Camacho (2008). Min-Max Model Predictive Control of a pilot plant. In *Proceedings of the 2008 American Control Conference*, Seattle, USA.

7. J. K. Gruber, D. R. Ramírez, T. Alamo, C. Bordons, E. F. Camacho (2008). Min-Max Predictive Control of a Pilot Plant Using a QP Approach. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancún, Mexico.

8. J. K. Gruber, F. Rodríguez, C. Bordóns, J. L. Guzmán, M. Berenguel, E. F. Camacho (2008). A Volterra Model of the Greenhouse Temperature Using Natural Ventilation. In *Proceedings of the 17th IFAC World Congress*, Seoul, Korea.

9. J. K. Gruber, C. Bordons, M. Doll (2008). Control de la Tasa de Exceso de Oxígeno de una Pila de Combustible (in Spanish). In *Actas de las XXIX Jornadas de Automática*, Tarragona, Spain.

10. J. K. Gruber, F. Dorado, C. Bordons (2008). Nonlinear Predictive Control Using Volterra Models. Application to a pilot Plant. In *Proceedings of the Controlo 2008 Conference - 8th Portuguese Conference on Automatic Control*, Vila Real, Portugal.

11. J. K. Gruber, F. Rodríguez, C. Bordóns, J. L. Guzmán, M. Berenguel (2007). Modeling greenhouse temperature using Volterra models. In *Proceedings of the 2007 IFAC Agricontrol Conference*, Osijek, Croatia.

12. J. K. Gruber, C. Bordons, R. Ortega, R. Talj (2007). Diseño de un Controlador PI no Lineal para el Suministro de Aire de una Pila de Combustible (in Spanish). In *Actas de las XXVIII Jornadas de Automática*, Huelva, Spain.

13. F. Dorado, J. K. Gruber, C. Bordons, E. F. Camacho (2007). Volterra Model Based Predictive Control. Application to a PEM Fuel Cell. In *Proceedings of the 14th Nordic Process Control Workshop*, Helsinki, Finland.

14. J. K. Gruber, C. Bordons (2006). Aplicación de MPC no Lineal Basado en Modelos de Volterra a una Planta Piloto (in Spanish). In *Actas de las XXVII Jornadas de Automática*, Almería, Spain.

# Chapter 2

# Volterra series models

The current chapter gives a short introduction to Volterra series models with an emphasis on non-autoregressive discrete-time second order Volterra series models. The different aspects of approximating nonlinear dynamic systems by Volterra series models and their use in MPC strategies will be discussed.

In the first place the basic definitions of general Volterra series models are given and some important subclasses of these models will be presented. Starting with the continuous-time Volterra model, both autoregressive and non-autoregressive discrete-time Volterra series models will be defined. After the definition of the Volterra series model the identification of the model parameters will be presented, including the choice of an input signal sequence for obtaining suitable input-output data for identification purposes as well as the least squares method for parameter estimation. Furthermore, the transformation of Volterra series models from an autoregressive to a non-autoregressive form will be given. The third section defines in detail the nonlinear prediction model based on a second order Volterra series. Then, after a short explanation of the general idea of MPC and the definition of a quadratic cost function, a description of the general optimization problem resulting from MPC based on second order Volterra series models will be given. Furthermore, the state-space formulation of second order Volterra series models, necessary for the stability analysis of the later presented NMPC strategies, will be presented. Finally, the advantages and disadvantages of Volterra series models and their use in MPC will be summarized.

It has to be mentioned that this chapter is not thought to give an general overview about models, identification and MPC strategies, but to describe the Volterra series models and the identification techniques used in this document as well as the definition

of the optimization problem resulting from the combination of quadratic cost functions and prediction models based on second order Volterra models.

## 2.1 Model definitions

One of the main questions when representing a dynamic system by a mathematical model is the posterior usage of the model. In control engineering disciplines mainly two concepts can be distinguished: modeling for simulation and modeling for control. Although both concepts are based on a mathematical abstraction of the system, models for simulation and for control show large differences, especially in its complexity. Models for simulation purposes are generally characterized by a high accuracy, leading to an elevated complexity and nonlinearity. Examples for highly complex simulation models are the ones used for generating weather forecasts, simulating nuclear processes or earthquakes.

On the other side, models for control purposes are usually less complex and accurate and in many cases linear. This is due to the fact that the model will be used directly to compute the control action to be applied to the system. In MPC, a sequence of future control actions is calculated with the mentioned model. With commonly used quadratic cost functions, the computation of the control action gives rise to different optimization problems and, as a consequence, to different methods to resolve these problems, especially in the case of nonlinear models. Therefore, basic MPC techniques commonly use simple representations to model the dynamic behavior of the system to control, e.g. autoregressive moving average model with exogenous inputs (ARMAX) [72] in Generalized Predictive Control (GPC) [28], finite impulse response (FIR) [72] model in Model Predictive Heuristic Control (MPHC) [112] and step response model in Dynamic Matrix Control (DMC) [31]. For a general overview of MPC strategies see [104, 23].

The input-output relationship of a causal single-input single-output (SISO) system is a mapping of the past input values to the present output of the system [26]. This relationship can be expressed for a Linear Time-Invariant (LTI) system by a convolution integral [132, 26]:

$$y(t) = \int_0^\infty h_1(\tau)\, u(t-\tau)\, \mathrm{d}\tau \tag{2.1}$$

where the input signal $u(t)$ is related to the output $y(t)$ and the system dynamics are determined by the impulse response $h_1(t)$. With this representation the output $y(t)$

can can be viewed as a function of the input history for all $\tau \leq t$ [38]. For nonlinear time-invariant SISO systems the relation between the input $u(t)$ and the output $y(t)$ can be expressed as a series of multiple convolution integrals [132, 26]:

$$y(t) = \sum_{n=0}^{\infty} \int_0^{\infty} \overset{n}{\cdots} \int_0^{\infty} h_n(\tau_1, \ldots, \tau_n)\, u(t-\tau_1) \cdots u(t-\tau_n)\, \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_n \tag{2.2}$$

being $h_n(\tau_1, \ldots, \tau_n)$ the $n$-th order Volterra kernel. Without loss of generality, the Volterra kernels can be assumed to have a symmetric character [26]. An $n$-th order Volterra kernel is symmetric satisfying the condition:

$$h_n(t_1, \ldots, t_n) = h_n(t_{\pi(1)}, \ldots, t_{\pi(n)}) \tag{2.3}$$

where $\pi(\cdot)$ denotes any permutation of the integers $1, \ldots, n$ [120]. Note that the kernel $h_0$ denotes a constant offset of the continuous Volterra series model. With all kernels of second or higher order equal to zero, i.e $h_n(\cdot) = 0 \; \forall\, n \geq 2$ the model (2.2) is reduced to an impulse response model for a linear system. With the higher order terms, (2.2) represents a logical extension of a linear convolution model with the nonlinearity being an additional and additive term.

The discrete-time Volterra series model is defined analogously to the continuous time SISO (2.2) models. With the convolution integrals replaced by discrete convolution sums the discrete-time SISO Volterra series model is defined as [38]:

$$y(k) = h_0 + \sum_{n=1}^{\infty} \sum_{i_1=0}^{\infty} \overset{n}{\cdots} \sum_{i_n=0}^{\infty} h_n(i_1, \ldots, i_n) u(k-i_1) \cdots u(k-i_n) \tag{2.4}$$

where $h_0$ is the model offset, $h_1$ are the linear term parameters and $h_n \; \forall\, n \geq 2$ represents the nonlinear term parameters. It has been shown in [20] that any stable fading memory system can be approximated arbitrarily well by finite Volterra series models with a sufficiently high order. Truncating the infinite terms of the model (2.4) leads to the following definition of discrete-time finite Volterra series models:

$$y(k) = h_0 + \sum_{n=1}^{m} \sum_{i_1=0}^{N_1} \overset{n}{\cdots} \sum_{i_n=0}^{N_n} h_n(i_1, \ldots, i_n) u(k-i_1) \cdots u(k-i_n) \tag{2.5}$$

where $m$ denotes the order of the model. Note that the lower limits of the sums can be changed from 0 to 1 in (2.4) without loss of generality for causal systems. As the input value $u(k)$ has no immediate effect on the output $y(k)$, the Volterra kernels satisfy:

$$h_n(i_1, \ldots, i_n) = 0 \quad \text{if} \quad i_1 = 0 \vee i_2 = 0 \vee \ldots \vee i_n = 0 \qquad \forall\, n = 1, \ldots, m \tag{2.6}$$

As a consequence, these parameters can be neglected allowing the mentioned change in the lower limit of the sums. In the specialized literature, both notations with a lower

bound of summation of 0 [38, 98, 50] and 1 [38, 70, 36] can be found. Analogously to the continuous case and without loss of generality, the Volterra kernels can be considered symmetric satisfying the condition:

$$h_n(i_1, \ldots, i_n) = h_n(i_{\pi(1)}, \ldots, i_{\pi(n)}) \tag{2.7}$$

Furthermore, due to the commutativity of scalar multiplication the statement:

$$u(k-i_1) \cdots u(k-i_n) = u(k-i_{\pi(1)}) \cdots u(k-i_{\pi(n)}) \tag{2.8}$$

holds. After changing the lower limits of the sums from 0 to 1 in (2.5) and under consideration of symmetry (2.7) and commutativity (2.8) the discrete-time finite Volterra series model can be written in the following triangular form:

$$y(k) = h_0 + \sum_{n=1}^{m} \sum_{i_1=1}^{N_1} \sum_{i_2=i_1}^{N_2} \overset{n}{\cdots} \sum_{i_n=i_{n-1}}^{N_n} h_n^{tr}(i_1, \ldots, i_n) u(k-i_1) \cdots u(k-i_n) \tag{2.9}$$

with $m$ being the order of the model. The coefficients $h_n^{tr}(\cdot)$ of the triangular form can be determined from the coefficients $h_n(\cdot)$ of the non-triangular model (2.5). With respect to the calculation of the output $y(k)$ or for control purposes, the Volterra kernels can be modified without loss of generality to be upper triangular matrices. Nevertheless, the use of the triangular form of a Volterra series model has a positive effect on the later described parameter identification process due to the lower number of parameters to be identified.

As already mentioned before, in the case of models for control purposes the idea is to use less complex models than the ones used for simulation. In order to calculate the control action in a reasonable time, the complexity of the model has to be appropriate to compute the control signal within the current sampling period. With an increasing computational effort to calculate the input signal with higher order Volterra series models, in many cases the used model is limited to second order Volterra series models [38, 70, 98, 36, 50]. Analogously, for the design of new control approaches only second order Volterra series models are used in this document[1]:

$$y(k) = h_0 + \sum_{i=1}^{N_1} h_1(i) u(k-i) + \sum_{i=1}^{N_2} \sum_{j=i}^{N_2} h_2(i,j) u(k-i) u(k-j) \tag{2.10}$$

The second order Volterra series model for multiple-input single-output (MISO) systems is defined in an analogy to the single-input single-output (SISO) model given in

[1]Note that through the entire document exclusively the triangular form of the Volterra kernels will be used. Hence, for the sake of simplicity of the notation the superscript $tr$ to denote the triangular form has been neglected.

(2.10). With two or more inputs the Volterra series model containing only self kernels is given by:

$$y(k) = h_0 + \sum_{n=1}^{m}\sum_{i=1}^{N_{1,n}} h_{1,n}(i)u_n(k-i) + \sum_{n=1}^{m}\sum_{i=1}^{N_{2,n}}\sum_{j=i}^{N_{2,n}} h_{2,n}(i,j)u_n(k-i)u_n(k-j) \quad (2.11)$$

The definition of the model using exclusively self-kernels means that each input is multiplied only with itself and no cross terms multiplying different inputs are considered (cross-kernels) ([26]). In addition, the autoregressive form of Volterra series models considers the last output values in the computation of the new output value. The SISO case for autoregressive second order Volterra series models is defined by:

$$\begin{aligned} y(k) &= h_0^{ar} + \sum_{i=1}^{N_y^{ar}} h_y^{ar}(i)y(k-i) + \sum_{i=1}^{N_1^{ar}} h_1^{ar}(i)u(k-i) + \\ &\quad \sum_{i=1}^{N_2^{ar}}\sum_{j=i}^{N_2^{ar}} h_2^{ar}(i,j)u(k-i)u(k-j) \end{aligned} \quad (2.12)$$

Analogously, autoregressive MISO second order Volterra series models can be written as:

$$\begin{aligned} y(k) &= h_0^{ar} + \sum_{i=1}^{N_y^{ar}} h_y^{ar}(i)y(k-i) + \sum_{n=1}^{m}\sum_{i=1}^{N_{1,n}^{ar}} h_{1,n}^{ar}(i)u_n(k-i) + \\ &\quad \sum_{n=1}^{m}\sum_{i=1}^{N_{2,n}^{ar}}\sum_{j=i}^{N_{2,n}^{ar}} h_{2,n}^{ar}(i,j)u_n(k-i)u_n(k-j) \end{aligned} \quad (2.13)$$

It has to be mentioned that in this document only the SISO (2.10) and MISO type (2.11) of non-autoregressive second order Volterra series models are considered for control purposes. Bilinear and autoregressive Volterra series models have not been considered as their use in MPC leads to several computational optimization problems and requires a reduction of the computational complexity as done in [50, 52]. In spite of the problems in the optimization process of model based predictive control, autoregressive Volterra series models can be a good choice for the identification of the dynamics of a given system. An identified autoregressive Volterra series model can be transformed easily to a non-autoregressive form in the case of open-loop stable processes. For the transformation from the autoregressive to the non-autoregressive form see Section 2.2.1. For further information on the different types of Volterra series models the reader is referred to [132, 38, 120]. In the following sections the denomination Volterra series model will be used as a synonym for the discrete-time non-autoregressive form of Volterra series models. When addressing the autoregressive form of Volterra series model, the autoregressive character of the model will be mentioned explicitly.

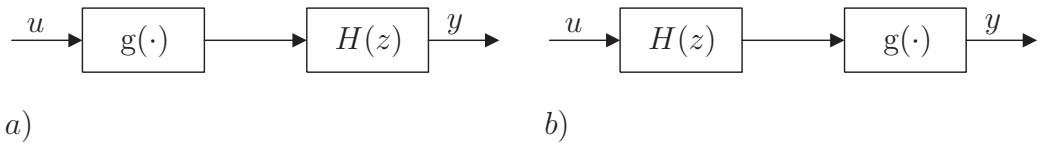


**Figure 2.1**: Block-oriented Volterra series models, a) Hammerstein model, b) Wiener model.

### 2.1.1 Important subclasses of Volterra series models

The class of non-autoregressive Volterra series models contains several subclasses, the most noteworthy are the diagonal Volterra series models, the Hammerstein and the Wiener models [38]. In the following paragraphs the mentioned subclasses will be explained in general and their second order description will be given.

One of the most important subclasses of Volterra series models are the diagonal ones. This type of model is characterized by non-zero parameters on the main diagonals of the Volterra kernels and zero valued parameters on the off-diagonals. Hence, the general Volterra series model (2.9) with all off-diagonal parameters equal to 0 can be written as:

$$y(k) = h_0 + \sum_{n=1}^{m} \sum_{i=1}^{N_n} h_n(i, \overset{n}{\ldots}, i) u(k-i)^n \tag{2.14}$$

It is clear from the definition of the off-diagonal elements and also from the diagonal Volterra series model (2.14) that the number of parameters is reduced with respect to the general model (2.9). In the case of a second order model (2.10), the diagonal Volterra series model is defined by:

$$y(k) = h_0 + \sum_{i=1}^{N_1} h_1(i) u(k-i) + \sum_{i=1}^{N_2} h_2(i,i) u(k-i)^2 \tag{2.15}$$

Another important subclass of Volterra series models are the Hammerstein models, one of the simplest and most popular members of the family of block-oriented models [38]. The Hammerstein model consists of a single static nonlinearity g(·) in series with a linear dynamic model defined by a transfer function $H(z)$, see Fig 2.1. For the general case, the output of the Hammerstein model is defined by:

$$y(k) = h_0 + \sum_{n=1}^{m} \sum_{i=1}^{j} \gamma_n h_1(i) u(k-i)^n \tag{2.16}$$

It can be seen that the Hammerstein model uses the same parameters $h(i)$ for the first and higher order terms where the factor $\gamma_n$ is used to scale the parameters used

in the different terms. Furthermore it can be observed that the given Hammerstein model (2.16) itself is a subclass of the diagonal Volterra series model given in 2.14. The second order Hammerstein model, based on (2.16), can be written as:

$$y(k) = h_0 + \sum_{i=1}^{N_1} h(i)u(k-i) + \gamma \sum_{i=1}^{N_1} h(i)u(k-i)^2 \tag{2.17}$$

The Wiener model, also from the family of the block-oriented models, is another important and extensively used subclass of the general Volterra series model. Consisting of a linear dynamic model $H(z)$ and a single static nonlinearity connected in series (see Fig. 2.1), the Wiener model can be regarded as the dual of the Hammerstein model [38]. The general definition for a $n$-th order Wiener model is given by:

$$y(k) = h_0 + \sum_{n=1}^{m} \sum_{i_1=1}^{j} \sum_{i_2=i_1}^{j} \overset{n}{\cdots} \sum_{i_n=i_{n-1}}^{j} \gamma_n h(i_1) \cdots h(i_n) u(k-i_1) \cdots u(k-i_n) \tag{2.18}$$

From the definition of the Wiener model it can be seen that the higher order parameters are products of the parameters $h(i)$, scaled only by the factor $\gamma_n$. Hence, with (2.10) the second order Wiener model can be written as:

$$y(k) = h_0 + \sum_{i=1}^{N_1} h(i)u(k-i) + \gamma \sum_{i=1}^{N_1} \sum_{j=i}^{N_1} h(i)h(j)u(k-i)u(k-j) \tag{2.19}$$

Being the diagonal Volterra series model, the Hammerstein and the Wiener model subclasses of the general Volterra series models, the mentioned models can be used within the same model predictive control framework as the general Volterra series models. A schematic overview of the mentioned subclasses of Volterra series models is given in Fig. 2.2. It has to be mentioned that one of the major disadvantages of Volterra series models, analogously to step and impulse response models, is the high number of model parameters. Especially for higher order systems the parameter number grows rapidly with an increasing truncation order. To model higher order systems, in many cases one of the above mentioned subclasses of Volterra series models is used instead of the general Volterra series model. The subclasses of diagonal Volterra series models, Hammerstein and Wiener models are characterized by a slower increase of the model parameters in function of the model order and the used truncation order. The number of parameters for the second order Volterra series model and its subclasses is given in Tab. 2.1 for an unique truncation order $N_1 = N_2 = N_t$. Fig. 2.3 compares the number of model parameters for different truncation orders. For details on the mentioned subclasses of Volterra series models see [38].

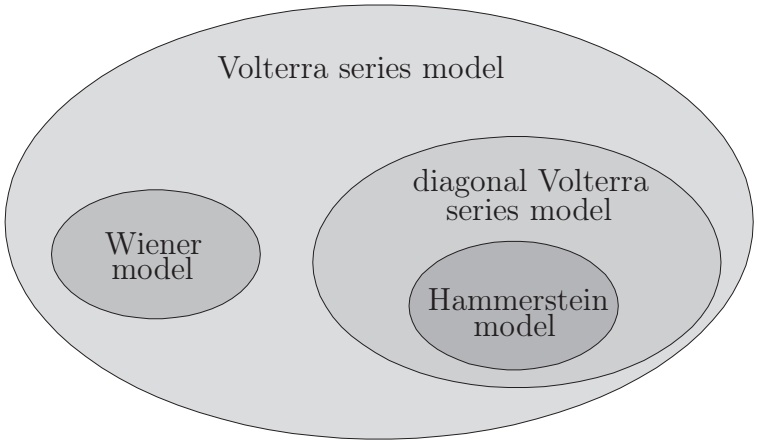


**Figure 2.2**: Schematic overview of some important subclasses of Volterra series models.

| Model | Number of parameters |
|---|---|
| Volterra series model | $1+N_t+\frac{(N_t+1)N_t}{2}$ |
| diagonal Volterra series model | $1+N_t+N_t$ |
| Hammerstein model | $1+N_t+1$ |
| Wiener model | $1+N_t+1$ |

**Table 2.1**: Number of model parameters of the second order Volterra series model and its subclasses for an unique truncation order $N_1 = N_2 = N_t$.

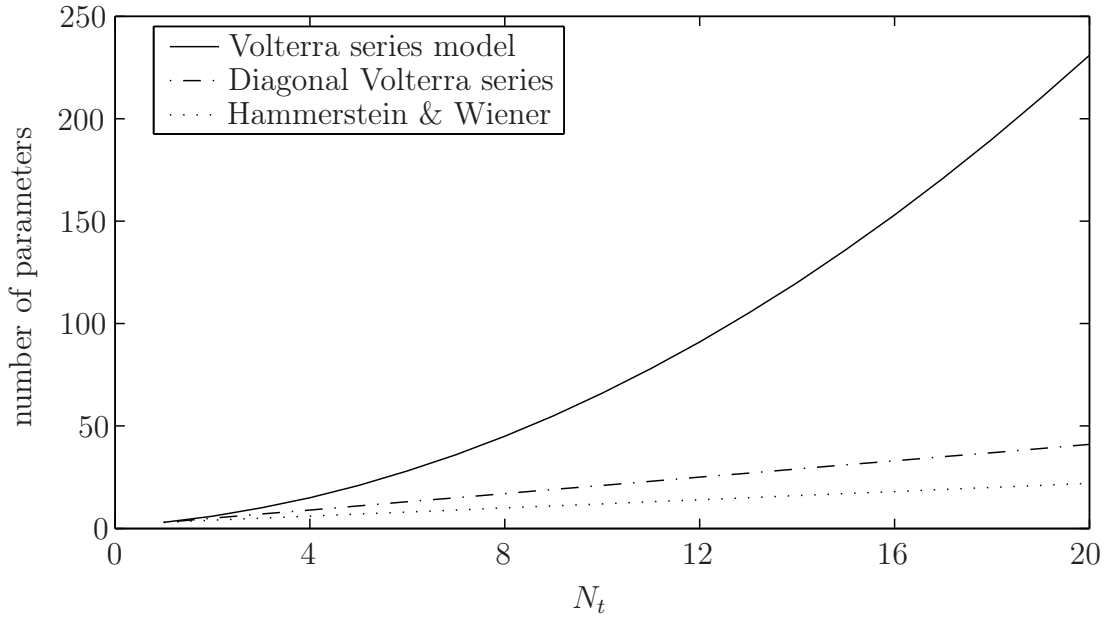


**Figure 2.3**: Graphical representation of the number of model parameters for second order Volterra series models and its subclasses as a function of the truncation order $N_1 = N_2 = N_t$.

In some cases the use of one of the mentioned subclasses is not desired as these models have a reduced ability to describe the dynamic behavior of a system. In order to reduce the number of parameters for the general Volterra series model, different truncation orders for the first and the second order term parameters can be used. In many cases a very short truncation order for the second order term improves considerably the quality of adjustment with respect to a linear system, without the necessity to use the same truncation order as for the linear part of the model.

## 2.2 Model parameter identification

System identification in its broadest sense means to obtain information of a given system from measured data and to build a dynamical model based on the gathered information. In the case of Volterra series models, identification refers to the determination of the model parameters of the Volterra kernels from suitable input-output. For Volterra series models, being a typical black box model, the identification correlates the applied input signal and the measured output signal and determines the numerical values of the parameters, see the general scheme for model parameter identification based on input-output data in Fig. 2.4. In the specialized literature many identification methods can be found, e.g. the simplex method, maximum likelihood estimation [72], instrumental variable methods [125] or neural network identification [6], but probably

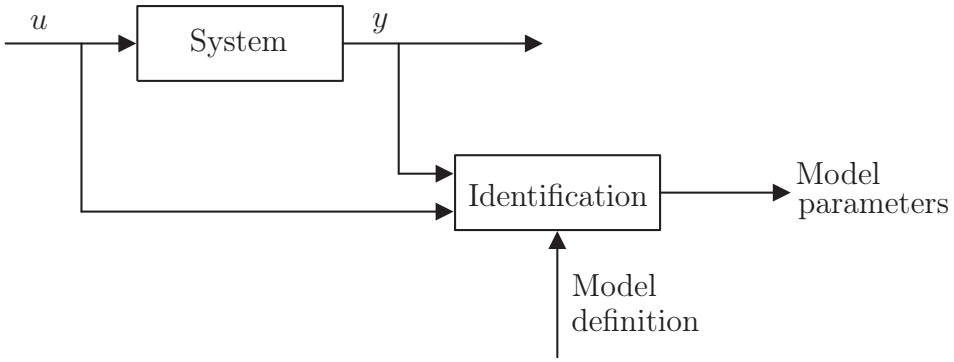


**Figure 2.4**: Diagram of the identification based on input-output data.

the most used one is the least squares method minimizing the quadratic error between the modeled output and the measured output of the system. Although second or higher order Volterra series models are nonlinear, these models are linear in the parameters. This attribute, perhaps one of the main advantages of nonlinear Volterra series models, allows the use of identification techniques generally used for linear models. This section describes the input signal used in this document to obtain suitable input-output data and the least squares method to identify the model parameters. Furthermore, the transformation from the autoregressive Volterra series model to the non-autoregressive form will be explained.

A very important role in model parameter identification plays the choice of the input signal sequence in order to get adequate input-output data. A common input signal to gather input-output data for linear model parameter identification is the *pseudo random binary sequence* (PRBS), a deterministic signal with white noise properties [72]. The PRBS is characterized by only two different signal values with changing period lengths where the lengths of the periods are not random, but generated by shift registers. Analogously, for the parameter identification of a nonlinear model a *pseudo random multilevel sequence* (PRMS) can be used. In PRMS, the signal switches deterministically between a finite number of levels. It has been shown in [95] that an $(n+1)$-level PRMS sufficiently excites a system to be identified by an $n$-th order Volterra series model. Therefore, in order to obtain suitable input-output for the identification of a second order Volterra series model, PRMS with 3 different levels have been used as input sequences in this document, see Fig. 2.5. For details on the exact definition of PRMS the reader is referred to [94, 95], for other possible input sequences, e.g. Gaussian white noise and sine-power sequences, see [38, 120].

As already mentioned, Volterra series models are linear in the parameters allowing the use of well known linear identification techniques. Exploiting the linear character of

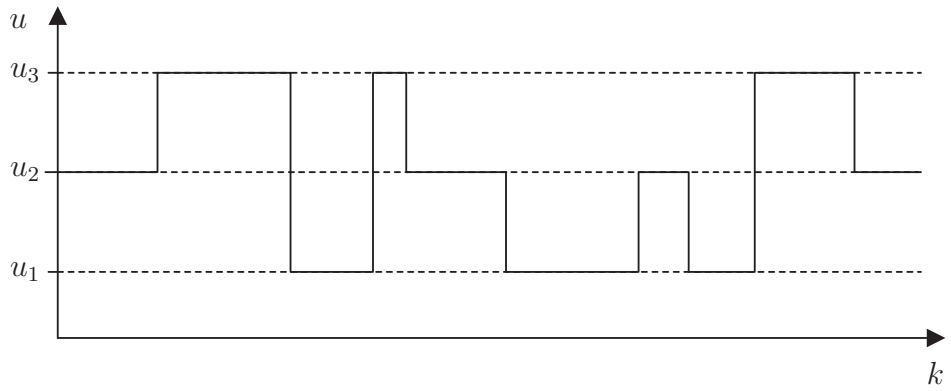


**Figure 2.5**: Diagram of a pseudo random multilevel sequence (PRMS) with 3 levels, suitable to identify a second order Volterra series model.

the parameters, in this document the least squares estimation method has been used for the parameter identification of the Volterra kernels. With suitable input-output data obtained in experiments from a process, denoting $u(k)$ the input signal and $y(k)$ the output signal, the system can be represented generally as a Volterra series model (2.5) in the following way:

$$\begin{aligned} y(k) &= h_0 + h_1(1)u(k-1) + \ldots + h_1(N_1)u(k-N_1) + \\ &\quad h_2(1,1)u(k-1)^2 + \ldots + h_2(N_2,N_2)u(k-N_2)^2 + \ldots \end{aligned} \tag{2.20}$$

With the two following vectors, one containing the unknown model parameters

$$\boldsymbol{\theta} = [h_0 \;\; h_1(1) \;\; \ldots \;\; h_1(N_1) \;\; h_2(1,1) \;\; \ldots \;\; h_2(N_2,N_2) \;\; \ldots]^T \tag{2.21}$$

and the other one the input data used by the model

$$\boldsymbol{\varphi}(k) = \left[1 \;\; u(k-1) \;\; \ldots \;\; u(k-N_1) \;\; u(k-1)^2 \;\; \ldots \;\; u(k-N_2)^2 \;\; \ldots\right]^T \tag{2.22}$$

the system output (2.20) for the current sampling period $k$ can be written as:

$$y(k) = \boldsymbol{\varphi}(k)^T\boldsymbol{\theta} \tag{2.23}$$

Usually the system is not perfectly deterministic but shows some noise in the measurement $y(k)$. Therefore (2.23) is extended to include an additional term $\varepsilon(k)$. With the residual $\varepsilon(k)$ considering noise the system output (2.23) becomes:

$$y(k) = \boldsymbol{\varphi}(k)^T\boldsymbol{\theta} + \varepsilon(k) \tag{2.24}$$

The term $\boldsymbol{\varphi}(k)^T\boldsymbol{\theta}$ can be considered as the prediction $\hat{y}(k|k-1)$ for the output $y(k)$ made at $k-1$. Thus, the error between the model and the system can be defined:

$$\varepsilon(k) = y(k) - \boldsymbol{\varphi}(k)^T\boldsymbol{\theta} \tag{2.25}$$

The least squares optimization criterion gives rise to the following problem where the sum of quadratic errors has to be minimized over the set of input-output data:

$$J(\boldsymbol{\theta}) = \sum_{i=0}^{M} \varepsilon(k+i)^2 = \sum_{i=0}^{M} (y(k+i) - \boldsymbol{\varphi}(k+i)^T \boldsymbol{\theta})^2 \tag{2.26}$$

In order to write the optimization problem in matricial form, the vector $\boldsymbol{\Upsilon}_k$ containing the system output measurements and $\Psi_k$ denoting the matrix containing the vectors $\boldsymbol{\varphi}(\cdot)$ are defined:

$$\boldsymbol{\Upsilon}_k = [y(k) \;\; y(k+1) \;\; \ldots \;\; y(k+M)] \tag{2.27}$$

$$\Psi_k = \left[\boldsymbol{\varphi}(k)^T \;\; \boldsymbol{\varphi}(k+1)^T \;\; \ldots \;\; \boldsymbol{\varphi}(k+M)^T\right] \tag{2.28}$$

With (2.27)-(2.28) the sum of quadratic errors can be written as:

$$J(\boldsymbol{\theta}) = (\boldsymbol{\Upsilon}_k - \Psi_k \boldsymbol{\theta})^T (\boldsymbol{\Upsilon}_k - \Psi_k \boldsymbol{\theta}) \tag{2.29}$$

Finally, by means of the derivative of (2.29) the vector of model parameters becomes:

$$\boldsymbol{\theta}^* = \left(\Psi_k^T \Psi_k\right)^{-1} \Psi_k^T \boldsymbol{\Upsilon}_k \tag{2.30}$$

For a more exhaustive analysis of the least squares method and different identification methods see [119, 72, 52].

It has to be mentioned that the high number of parameters of Volterra series models requires a large set of input-output data to carry out an appropriate identification. As already described in section 2.2, for many systems a reduction of the number of model parameters is possible, allowing smaller data sets to be used during the identification process. Another problem of the high number of parameters in the identification process using the least squares method is the one leading frequently to a singular or nearly singular to be inverted (2.30). The singular matrix is usually the result of a bad conditioned identification problem, avoidable through reformulation of the problem with less parameters to be identified.

Besides the parameter estimation from input-output data of the process to be identified, more systematic methods exist to model the system behavior using directly the equations describing the system dynamic. That is, the parameters of the Volterra series model are derived from another mathematical model, usually a detailed prediction model based on first principles as mass and energy balances. As already explained in section 2.1, these models usually show an elevated complexity in order to fit well the system dynamics, but are difficult to use for control purposes. The analytical parameter estimation can be carried out by means of a Taylor series expansion of the original model with consecutive Carleman linearization [120, 80]. The reader is referred to [38, 35] for details on the described analytic method to determine the model parameters of a Volterra series model.

### 2.2.1 Transformation from autoregressive to non-autoregressive form

In this section the transformation of autoregressive Volterra series models (2.12) to the non-autoregressive representation (2.10) will be described. In general, autoregressive Volterra series models obtain a similar model fit with less parameters than non-autoregressive Volterra series models. Nevertheless, the linear and nonlinear terms of autoregressive Volterra series models can not be separated and the use of this kind of model in MPC results in more complex optimization problems. The transformation is especially interesting when only a small input-output data set for the parameter identification is available. After the transformation to a non-autoregressive representation, the model can be used in the later presented control strategies.

The autoregressive discrete-time second order Volterra series model (2.12) can be written in the following form similar to a transfer function model with a constant offset and an additive and additional nonlinear term [52, 45]:

$$A(q^{-1})y(k) = h_0^{ar} + B_1(q^{-1})u(k) + B_2(q_1^{-1}, q_2^{-1})u(k)^2 \tag{2.31}$$

where $q^{-1}$, $q_1^{-1}$, $q_2^{-1}$ denote backward shift operators and the polynomials $A(q^{-1})$, $B_1(q^{-1})$ and $B_2(q_1^{-1}, q_2^{-1})$ are given by:

$$\begin{aligned} A(q^{-1}) &= 1 - h_y^{ar}(1)q^{-1} - \ldots - h_y^{ar}(N_y)q^{-N_y} \\ B_1(q^{-1}) &= h_1^{ar}(1)q^{-1} + \ldots + h_1^{ar}(N_1)q^{-N_1} \\ B_2(q_1^{-1}, q_2^{-1}) &= \sum_{i=1}^{N_2}\sum_{j=i}^{N_2} h_2^{ar}(i,j)q_1^{-i}q_2^{-j} \\ h_2^{ar}(i,j)q_1^{-i}q_2^{-j}u(k)^2 &= h_2^{ar}(i,j)u(k-i)u(k-j) \end{aligned} \tag{2.32}$$

The constant coefficient $h_0$ of the non-autoregressive Volterra series model (2.10) can be defined as a division of $h_0^{ar}$ and $A(1)$, where $A(1)$ denotes the substitution of the backward shift operator by $q^{-1} = 1$. With

$$A(1) = 1 - \sum_{i=1}^{N_y} h_y^{ar}(i) \tag{2.33}$$

the offset $h_0$ is defined as:

$$h_0 = \frac{h_0^{ar}}{A(1)} \tag{2.34}$$

The linear term parameters of the non-autoregressive Volterra series model can be obtained directly by polynomial long division of $B_1(q^{-1})$ and $A(q^{-1})$:

$$\frac{h_1^{ar}(1)q^{-1} + \ldots + h_1^{ar}(N_1^{ar})q^{-N_1^{ar}}}{A(q^{-1})} = h_1(1)q^{-1} + \ldots + h_1(N_1)q^{-N_1} + \nu_1 \tag{2.35}$$

where $\nu_1$ denotes the remainder of the linear term. For the calculation of the parameters $h_2(i,j)$ the polynomial $A(q^{-1})$ has to be substituted by $A(q_1^{-1}, q_2^{-1})$, necessary due to the quadratic character of $B_2(q_1^{-1}, q_2^{-1})$. With the new polynomial defined by:

$$A(q_1^{-1}, q_2^{-1}) = 1 - h_y^{ar}(1)q_1^{-1}q_2^{-1} - \ldots - h_y^{ar}(N_y)q_1^{-N_y}q_2^{-N_y} \tag{2.36}$$

the quadratic term parameters $h_2(i,j)$ are calculated, analogously to the linear term parameters, by polynomial long division of $B_2(q_1^{-1}, q_2^{-1})$ and $A(q_1^{-1}, q_2^{-1})$. Then, with $B_2(q_1^{-1}, q_2^{-1})$ being a triangular matrix, the division is carried out for each non-zero diagonal of $B_2(q_1^{-1}, q_2^{-1})$ in the following form:

$$\begin{aligned}
&\frac{h_2^{ar}(1,1)q_1^{-1}q_2^{-1}+\ldots+h_2^{ar}(N_2^{ar},N_2^{ar})q_1^{-N_2^{ar}}q_2^{-N_2^{ar}}}{A(q_1^{-1},q_2^{-1})} = \\
&\qquad h_2(1,1)q_1^{-1}q_2^{-1}+\ldots+h_2(N_2,N_2)q_1^{-N_2}q_2^{-N_2}+\nu_2(1) \\
&\frac{h_2^{ar}(1,2)q_1^{-1}q_2^{-1}+\ldots+h_2^{ar}(N_2^{ar}-1,N_2^{ar})q_1^{-N_2^{ar}+1}q_2^{-N_2^{ar}+1}}{A(q_1^{-1},q_2^{-1})} \cdot q_2^{-1} = \\
&\qquad \left(h_2(1,2)q_1^{-1}q_2^{-1}+\ldots+h_2(N_2-1,N_2)q_1^{-N_2+1}q_2^{-N_2+1}\right) \cdot q_2^{-1}+\nu_2(2) \\
&\qquad\qquad \vdots \\
&\frac{h_2^{ar}(1,N_2^{ar})q_1^{-1}q_2^{-1}}{A(q^{-1})} \cdot q_2^{-N_2^{ar}+1} = \\
&\qquad \left(h_2(1,N_2^{ar})q_1^{-1}q_2^{-1}+\ldots+h_2(\xi,N_2)q_1^{-\xi}q_2^{-\xi}\right) \cdot q_2^{-N_2^{ar}+1}+\nu_2(N_2^{ar})
\end{aligned} \tag{2.37}$$

with $\xi = N_2 - N_2^{ar} + 1$.

Note that increasing the truncation orders $N_1$ and $N_2$ reduces the remainders $\nu_1$ and $\nu_2(i)$ for $i = 1, \ldots, N_2^{ar}$ of the polynomial long divisions and leads to a better approximation. With $N_1 \to \infty$ and $N_2 \to \infty$ a perfect approximation of the autoregressive Volterra series model by the non-autoregressive Volterra series model is reached, i.e. $\nu_1 \to 0$ and $\nu_2(i) \to 0$ for $i = 1, \ldots, N_2^{ar}$.

## 2.3 Prediction model

One of the most important elements of a model-based predictive control algorithm is the used prediction model to estimate the future behavior of the system in function of the applied control signal. The following paragraphs give a detailed definition of the prediction model for the discrete-time second-order Volterra series model shown in section 2.1.

With the second order Volterra series model (2.10) the system output can be predicted easily with:

$$\begin{aligned}\hat{y}(k+1|k) &= h_0 + h_1(1)\,u(k|k) + \ldots + h_1(N_1)\,u(k-N_1+1) + \\ &\quad h_2(1,1)\,u(k|k)^2 + h_2(1,2)\,u(k|k)\,u(k-1) + \ldots + \\ &\quad h_2(N_2,N_2)\,u(k-N_2+1)^2\end{aligned} \tag{2.38}$$

where $\hat{y}(k+1|k)$ denotes the output prediction made at $k$ for $k+1$ and $u(k|k)$ is the current control signal applied to the system. Usually the identified model is only an approximation of the real system and shows some model mismatch when the model does not capture the entire dynamic behavior of the system. As a consequence of the model mismatch it is recommendable to include an additional term in the output prediction containing the current estimation error. The estimation error in $k$ is defined as the difference between the measured output $y(k)$ and the output prediction $\hat{y}(k|k-1)$ made at $k-1$ for $k$:

$$d(k) = y(k) - \hat{y}(k|k-1) \tag{2.39}$$

In order to obtain a simple expression for the prediction model and without loss of generality the variable $N_t$ will be used as a common truncation order for the linear and nonlinear terms with $N_t = \max(N_1, N_2)$. In the case of $N_1 > N_2$, i.e. $N_t = N_1$, the missing second order term parameters are defined as $h_2(i,j) = 0 \;\forall\; i > N_2 \;\forall\; j > N_2$. In the opposite case, i.e. $N_2 > N_1$ and therefore $N_t = N_2$, the linear term parameters are defined as $h_1(i) = 0 \;\forall\; i > N_1$. Adding the prediction error and changing the notation for the truncation order, the one step ahead output prediction based on a second order Volterra series model becomes:

$$\begin{aligned}\hat{y}(k+1|k) &= h_0 + h_1(1)\,u(k|k) + \ldots + h_1(N_t)\,u(k-N_t+1) + \\ &\quad h_2(1,1)\,u(k|k)^2 + h_2(1,2)\,u(k|k)\,u(k-1) + \ldots + \\ &\quad h_2(N_t,N_t)\,u(k-N_t+1)^2 + d(k)\end{aligned} \tag{2.40}$$

Analogously, the predictions $\hat{y}(k+2|k)$, …, $\hat{y}(k+N|k)$ with $N$ denoting the prediction

horizon are defined as:

$$\begin{aligned}
\hat{y}(k+2|k) &= h_0 + h_1(1)\,u(k+1|k) + \ldots + h_1(N_t)\,u(k-N_t+2) + \\
&\quad h_2(1,1)\,u(k+1|k)^2 + h_2(1,2)\,u(k+1|k)\,u(k|k) + \ldots + \\
&\quad h_2(N_t,N_t)\,u(k-N_t+2)^2 + d(k) \\
&\vdots \\
\hat{y}(k+N|k) &= h_0 + h_1(1)\,u(k+N-1|k) + \ldots + h_1(N_t)\,u(k+N-N_t) + \\
&\quad h_2(1,1)\,u(k+N-1|k)^2 + h_2(1,2)\,u(k+N-1|k)\,u(k+N-2|k) + \\
&\quad \ldots + h_2(N_t,N_t)\,u(k+N-N_t)^2 + d(k)
\end{aligned} \tag{2.41}$$

Note that in the expressions for the predictions $\hat{y}(k+2|k)$, ..., $\hat{y}(k+N|k)$ the estimation error $d(k)$ for the current sampling period has been used. Without any knowledge about the future prediction error, the current prediction error $d(k)$ is used in the predictions $\hat{y}(k+2|k)$, ..., $\hat{y}(k+N|k)$.

A common technique in model-based predictive control is the use of a control horizon, defining the number of future control actions, i.e. the length of the input sequence, to be calculated. This means that the control action is calculated only for the given control horizon $N_u$ and not for the entire prediction horizon $N$ with $N_u \leq N$. Once the control horizon is reached it is assumed that the control signal is constant. Therefore, after reaching the control horizon, the input signal is defined as [23]:

$$u(k+i|k) = u(k+N_u-1|k) \;\forall\; N_u \leq i \leq N-1 \tag{2.42}$$

It is clear from the above defined predictions $\hat{y}(k+1|k), \ldots, \hat{y}(k+N|k)$ that the input signal has not to be defined beyond the prediction horizon as the predictions are not based upon the input signals $u(k+i|k)$ for $i > N-1$.

Finally, with all predictions $\hat{y}(k+i|k)$ with $i = 1, \ldots, N$ defined along the prediction horizon $N$, the future behavior of the system modeled by a second order Volterra series model can be written in matricial form [37, 38]:

$$\hat{\boldsymbol{y}} = G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) \tag{2.43}$$

which can be considered as a natural extension of the models used in the most common predictive control strategies based on linear convolution models, e.g. Model Predictive Heuristic Control (MPHC) [111, 112]. The vector $\hat{\boldsymbol{y}} \in \mathbb{R}^N$ is the vector of the predicted system output along the prediction horizon and $\boldsymbol{u} \in \mathbb{R}^{N_u}$ denotes the future input sequence over the control horizon. The term $\boldsymbol{G}\boldsymbol{u}$ with $\boldsymbol{G} \in \mathbb{R}^{N \times N_u}$ represents the linear part depending on the future input sequence. The term $\boldsymbol{c} \in \mathbb{R}^N$ includes all variables not depending on the future input sequence and the vector $\mathbf{f}(\boldsymbol{u}) \in \mathbb{R}^N$ contains the future-future and future-past cross terms depending on the input sequence $\boldsymbol{u}$. The constant term $\boldsymbol{c} \in \mathbb{R}^N$ is defined as [37, 38]:

$$\boldsymbol{c} = H\boldsymbol{u}_p + \boldsymbol{g} + \boldsymbol{h}_0 + \boldsymbol{d} \tag{2.44}$$

where $\boldsymbol{u}_p \in \mathbb{R}^{N_t}$ denotes the past input values applied to the system. The term $H\boldsymbol{u}_p$ with $H \in \mathbb{R}^{N \times N_t}$ represents the linear part depending on the past input values. The vector $\boldsymbol{d} \in \mathbb{R}^N$ with $\boldsymbol{d} = [d(k), \ldots, d(k)]^T$ contains the current estimation error, i.e. the error between the measured output and the predicted system output. The past-past cross terms depending on $\boldsymbol{u}_p$ are considered in $\boldsymbol{g} \in \mathbb{R}^N$ and the constant offset $h_0$ of the Volterra series model (2.10) is considered in $\boldsymbol{h}_0 \in \mathbb{R}^N$ with $\boldsymbol{h}_0 = [h_0, \ldots, h_0]^T$. The detailed definitions of all the vectors and matrices used in the prediction model (2.43)-(2.44) are given in the Appendix A.1. With the complete definition of the prediction model (2.43)-(2.44) based on a second order Volterra series model (2.10), the future evolution of the model output can be predicted for a given prediction horizon $N$ and control horizon $N_u$ and a known input sequence $\boldsymbol{u}$.

Note that $\mathbf{f}(\boldsymbol{u})$ is a nonlinear function depending on the input signal and, as a consequence, the prediction model (2.43)-(2.44) is nonlinear. As the nonlinear component is additive, the prediction model can be seen as the logical nonlinear extension of the prediction models based upon finite impulse response (FIR) or step response models. The problems resulting from the nonlinear prediction model will be discussed in section 2.5.

## 2.4 Objective function for MPC

In general terms, model-based predictive control is based on the computation of an input sequence that makes the controlled system to follow a given reference. The input sequence, which can have different forms for the model based predictive control algorithms in literature [23], is calculated minimizing an objective function based on the identified model.

Model-based predictive controllers, see Fig. 2.6, can be characterized by the following strategy [23]:

1. In every sampling period $k$ the future output values are predicted over the prediction horizon using the prediction model. The predicted output values $\hat{y}(k+i|k)\ \forall\ i = 1, \ldots, N$ depend on the future input signals $u(k+i|k)\ \forall\ i = 0, \ldots N_u - 1$ to be calculated and the values already known at $k$, i.e. the input signals applied in the previous sampling periods and the measured output values up to the current sampling period.

2. The future input signals are calculated by optimizing a given objective function

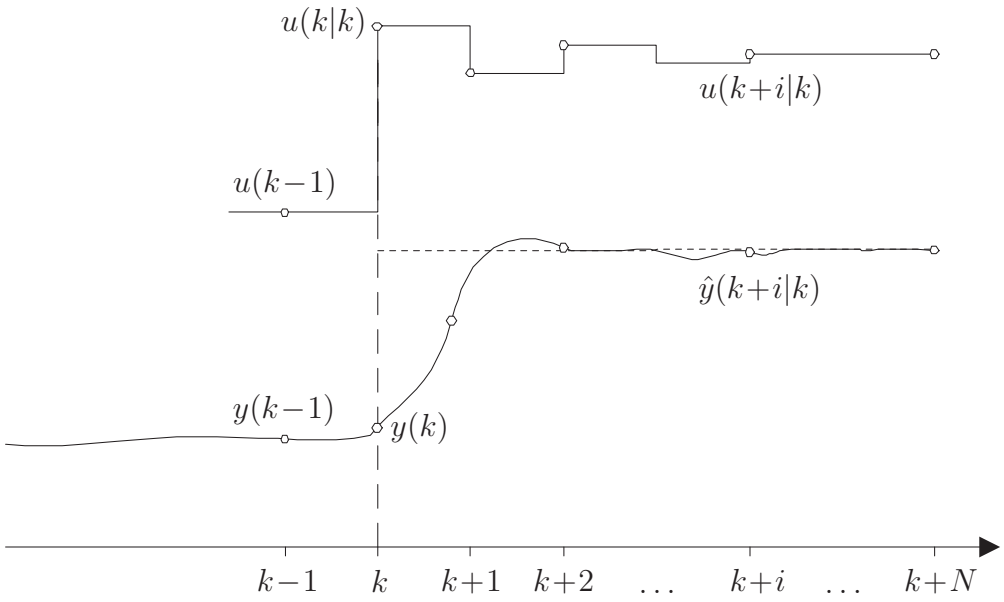


**Figure 2.6**: General model-based predictive control strategy [23].

in order to keep the error between the predicted output and a given reference (desired output value) as small as possible. In model-based predictive control the objective function is usually a quadratic cost function of the errors between the predicted output and the reference over the entire prediction horizon. Depending on the control objectives, the mentioned cost function can include a weighting of the control effort, obtaining this way a smoother control behavior. Furthermore, depending on the used control law, the optimization can include input and output constraints, restricting the input signals and the system output to determined sets of permitted values.

3. From the input sequence $u(k+i|k)\ \forall\ i = 0, \ldots N_u - 1$ only the first element $u(k|k)$ is applied in a receding horizon strategy to the controlled system. The remaining elements are not applied to the system because in the next sampling period $k+1$ new data of the system have been collected ($y(k+1)$) and the optimization of step 1 and 2 is repeated with the updated information. In the following sampling period $k+1$ the input signal $u(k+1|k+1)$ is calculated and applied to the system. Note that $u(k+1|k+1)$ has not necessarily the same value as $u(k+1|k)$ due to model mismatch or a suboptimal calculation of the input signal.

The above described strategy can be implemented as shown in the schematic diagram in Fig. 2.7. A model of the system is used to predict the future output based on the past input and output values as well as on the input sequence to be calculated.

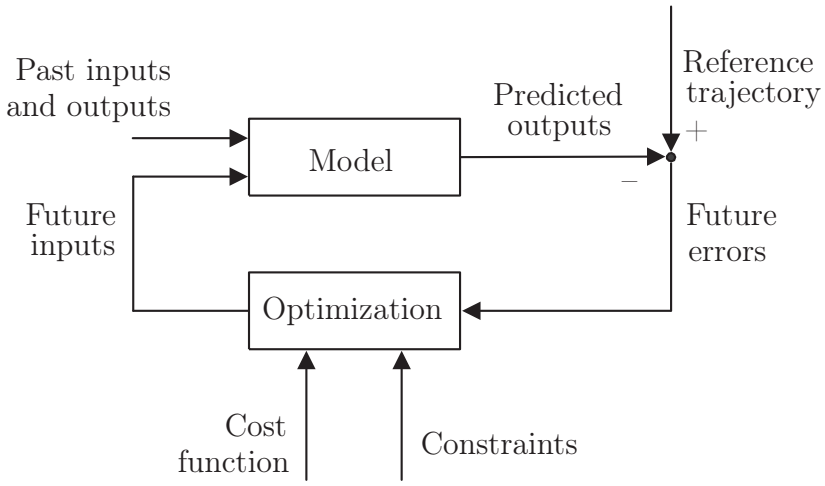


**Figure 2.7**: Schematic diagram of a model-based predictive control.

This input sequence is computed by a numerical solver[2] minimizing the cost function. If constraints have been defined, the computation of the input sequence is carried out under consideration of possible input and output constraints. The fact that the optimization routine uses the system model to calculate the future input sequence makes the prediction model one of the most important components of a model-based predictive control strategy. Another decisive factor is the choice of the objective function, also called cost function or performance index, allowing to adjust the controller to the desired behavior.

The cost function $J(\boldsymbol{u})$ to be minimized by a model-based predictive control strategy is usually given by the quadratic function [75, 23][3]:

$$J(\boldsymbol{u}) = \sum_{i=1}^{N} \delta(i) \left(\hat{y}(k+i|k) - r(k+i)\right)^2 + \sum_{i=0}^{N_u-1} \lambda(i)\, \Delta u(k+i|k)^2 \tag{2.45}$$

being $r(k+i)$ the reference $i$ steps ahead. The quadratic error between the predicted output $\hat{y}(k+i|k)$ and the desired reference value $r(k+i)$ is weighted by the parameter $\delta(j)$. The control effort $\Delta u(k+i|k)$ is penalized with the weighting factor $\lambda(j)$. The cost function considers the output error over the entire prediction horizon and the control effort along the control horizon. The weighting factors must satisfy the conditions $\delta(i) \geq 0$ for $i = 1, \ldots, N$ and $\lambda(i) \geq 0$ for $i = 0, \ldots, N_u - 1$.

[2]In unconstrained linear MPC, e.g. Model Algorithmic Control or Dynamic Matrix Control, an explicit solution to the minimization problem can be found.

[3]Note that the given performance index is suitable only for SISO models. For the cost function based on models with several inputs or outputs, the reader is referred to [23]

In this work, two special cases of the general cost function (2.45) have been used. The first one assumes $\delta(i) = 1$ for $i = 1, \ldots, N$ and $\lambda(i) = 0$ for $i = 0, \ldots, N_u - 1$, resulting in the following expression:

$$\begin{aligned} J(\boldsymbol{u}) &= \sum_{i=1}^{N} \left(\hat{y}(k+i|k) - r(k+i)\right)^2 \\ &= \left(\hat{\boldsymbol{y}} - \boldsymbol{r}\right)^T \left(\hat{\boldsymbol{y}} - \boldsymbol{r}\right) \end{aligned} \tag{2.46}$$

where $\boldsymbol{r} \in \mathbb{R}^N$ denotes the reference trajectory. An exact definition of the reference trajectory vector $\boldsymbol{r}$ is given in the Appendix A.2. The minimization of this cost function usually results in an aggressive control behavior as the control effort is not considered. For the second cost function the assumption $\delta(i) = 1$ for $i = 1, \ldots, N$ and $\lambda(i) = \lambda$ for $i = 0, \ldots, N_u - 1$ with $\lambda > 0$ has been made. Thus, the cost function considering a constant penalization of the control effort over the entire control horizon can be written as:

$$\begin{aligned} J(\boldsymbol{u}) &= \sum_{i=1}^{N} \left(\hat{y}(k+i|k) - r(k+i)\right)^2 + \sum_{i=0}^{N_u-1} \lambda\, \Delta u(k+i|k)^2 \\ &= \left(\hat{\boldsymbol{y}} - \boldsymbol{r}\right)^T \left(\hat{\boldsymbol{y}} - \boldsymbol{r}\right) + \lambda \boldsymbol{\Delta u}^T \boldsymbol{\Delta u} \end{aligned} \tag{2.47}$$

with $\boldsymbol{\Delta u} \in \mathbb{R}^{N_u}$ representing the future input increment sequence. For the definition of $\boldsymbol{\Delta u} \in \mathbb{R}^{N_u}$ see the Appendix A.2.

## 2.5 MPC using Volterra series models

In order to calculate an input sequence for the system, the prediction model (2.43)-(2.44) based on a second order Volterra series model is used in the cost functions defined in section 2.4. The following paragraphs show the quadratic cost functions for second order Volterra series and the encountered problems. Furthermore, the input sequence to be calculated will be defined in a theoretical manner.

Using the prediction model (2.43)-(2.44) in combination with the cost function $J(\boldsymbol{u})$ (2.46) without considering a penalization of the control effort, the performance index can be written as:

$$J(\boldsymbol{u}) = \left(G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r}\right)^T \left(G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r}\right) \tag{2.48}$$

In order to include a penalization of the control effort, the prediction model (2.43)-(2.44) based on the second order Volterra series model can be used in the cost function

(2.47). The resulting performance index

$$J(\boldsymbol{u}, \boldsymbol{\Delta u}) = (G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r})^T (G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r}) + \lambda \boldsymbol{\Delta u}^T \boldsymbol{\Delta u} \tag{2.49}$$

is a function of the input sequence $\boldsymbol{u}$ and the input increments $\boldsymbol{\Delta u}$. For optimization purposes, the cost function should depend only on $\boldsymbol{u}$ or only on $\boldsymbol{\Delta u}$. In this thesis, the cost function depending only on $\boldsymbol{u}$ has been chosen[4]. Therefore, the term containing the penalization of the control effort is transformed to depend on $\boldsymbol{u}$ instead of $\boldsymbol{\Delta u}$. In a general way, the control increments $\boldsymbol{\Delta u}$ can be written as:

$$\begin{aligned} \boldsymbol{\Delta u} &= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & -1 & 1 \end{bmatrix}}_{L_u} \cdot \underbrace{\begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N_u-1|k) \end{bmatrix}}_{\boldsymbol{u}} - \underbrace{\begin{bmatrix} u(k-1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\boldsymbol{u}_l} \\ &= L_u \boldsymbol{u} - \boldsymbol{u}_l \end{aligned} \tag{2.50}$$

where the column vector $\boldsymbol{u}_l \in R^{N_u}$ contains the last applied control signal $u(k-1)$ as first element and zeros as other elements. With (2.50) the term containing the penalization of the control effort can be written as:

$$\lambda \boldsymbol{\Delta u}^T \boldsymbol{\Delta u} = \lambda \boldsymbol{u}^T L_u^T L_u \boldsymbol{u}^T - 2\lambda \boldsymbol{u}_l^T L_u \boldsymbol{u} + \lambda \boldsymbol{u}_l^T \boldsymbol{u}_l \tag{2.51}$$

Finally, using the transformed penalization term (2.51) the cost function (2.49) can be expressed exclusively as a function of $\boldsymbol{u}$:

$$\begin{aligned} J(\boldsymbol{u}) &= (G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r})^T (G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r}) + \\ &\quad \lambda \boldsymbol{u}^T L_u^T L_u \boldsymbol{u}^T - 2\lambda \boldsymbol{u}_l^T L_u \boldsymbol{u} + \lambda \boldsymbol{u}_l^T \boldsymbol{u}_l \end{aligned} \tag{2.52}$$

With the cost function defined for the second order Volterra series model, the input sequence can be calculated by minimizing the cost function without penalization of the control effort (2.48) or by minimizing the cost function considering a weighting function for the control effort (2.52):

$$\begin{aligned} \boldsymbol{u}^* = \arg \min_{\boldsymbol{u}} \; & J(\boldsymbol{u}) \\ \text{s.t. } & \mathbf{q}(\boldsymbol{u}) \leq \boldsymbol{b}_c \end{aligned} \tag{2.53}$$

where the input and output constraints are considered by the expression $\mathbf{q}(\boldsymbol{u}) \leq \boldsymbol{b}_c$ with $\mathbf{q}(\boldsymbol{u}) : \mathbb{R}^{N_u} \mapsto \mathbb{R}^{n_c}$ and $\boldsymbol{b}_c \in \mathbb{R}^{n_c}$ being $n_c$ the number of constraints. In the case

[4]Naturally, both forms are valid for MPC strategies. Nevertheless, the transformation of the cost function to depend only $\boldsymbol{\Delta u}$ is more complex as the nonlinear term $\mathbf{f}(\boldsymbol{u})$ in (2.49) has to be rewritten as a function of $\boldsymbol{\Delta u}$.

of output constraints, the function $\mathbf{q}(\boldsymbol{u})$ is nonlinear due to the nonlinear prediction model. If only linear constraints are used, i.e. output constraints are not considered, (2.53) can be rewritten as:

$$\begin{aligned} \boldsymbol{u}^* &= \arg \min_{\boldsymbol{u}} J(\boldsymbol{u}) \\ &\quad \text{s.t. } L_c \boldsymbol{u} \leq \boldsymbol{b}_c \end{aligned} \tag{2.54}$$

The use of the prediction model (2.43)-(2.44) based on a second order Volterra series model (2.10) in model-based predictive control with a quadratic cost function from section 2.4 results in a fourth order polynomial depending on the input sequence $\boldsymbol{u}$. These polynomials are not necessarily convex in $\boldsymbol{u}$ and lead in the non-convex case to optimization problems due to several minimums of the cost function. One of the possibilities is the computation of the input sequence with numerical techniques, but without guarantee to globally minimize the cost function and, as a consequence, to obtain the optimal input sequence. Furthermore, the numerical optimization normally leads to a high computational effort and gives rise to control problems for fast processes. Another way to calculate the input sequence is the exploitation of convex optimization methods, i.e. transforming the non-convex problem to a convex one [21].

## 2.6 Volterra series models in state space

Generally, non-autoregressive discrete-time Volterra series models as the ones presented in Section 2.1 can be described in discrete state-space representation. The state-space representation has a special importance for the later given stability proofs for the different model-based predictive control strategies. In the following paragraphs the discrete state-space representation for second order Volterra series models (2.10) will be given.

In a first step, the past input values $u(k-i)$ with $i = 1, \cdots, N_t$ of the non-autoregressive second order Volterra series model (2.10) can be viewed as system states, e.g. the model states are defined by:

$$x_i(k) = u(k-i) \quad \text{for} \quad i = 1, \ldots, N_t \tag{2.55}$$

where $x_i(k)$ is the $i$-th element of the state vector $\boldsymbol{x}(k) \in \mathbb{R}^{N_t}$. It can be seen easily

that the definition of the states (2.55) can be rewritten in the form:

$$\begin{array}{rcl} x_1(k) & = & u(k-1) \\ x_2(k) & = & x_1(k-1) \\ x_3(k) & = & x_2(k-1) \\ \vdots & \vdots & \vdots \\ x_{N_t}(k) & = & x_{N_t-1}(k-1) \end{array} \tag{2.56}$$

where the first state $x_i(k)$ represents the last applied input signal and the remaining states depend on the states from the previous instant. With the states $x_i(k)$ for $i = 1, \ldots, N_t$ the model output of the second order Volterra series model (2.10) can be expressed by:

$$y(k) = \mathrm{f}(\boldsymbol{x}(k)) = \sum_{i=1}^{N_t} h_1(i)\, x_i(k) + \sum_{i=1}^{N_t} \sum_{j=i}^{N_t} h_2(i,j)\, x_i(k)\, x_j(k) \tag{2.57}$$

Note the similarity of (2.57) and (2.10) where the past input values $u(k-i)$ for $i = 1, \ldots, N_t$ have been substituted by the previously defined states $x_i(k)$ for $i = 1, \ldots, N_t$. Now it can be seen easily that (2.10) can be expressed as a nonlinear state space model defined by:

$$\begin{array}{rcl} \boldsymbol{x}(k+1) & = & A\boldsymbol{x}(k) + Bu(k) \\ y(k) & = & \mathrm{f}(\boldsymbol{x}(k)) \end{array} \tag{2.58}$$

where the state matrix $A \in \mathbb{R}^{N_t \times N_t}$ and the input matrix $B \in \mathbb{R}^{N_t}$ are given by (2.56) as:

$$A = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 & 0 \\ 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 & 0 \\ 0 & 0 & \ldots & 0 & 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{2.59}$$

Note that the constant offset $h_0$ given in the second order Volterra series model (2.10) has been removed in the state-space representation (2.58) by means of a linear transformation $\check{y}(k) = y(k) - h_0$. For the sake of simplicity of the notation and without loss of generality the output of the state space model without considering the offset has been denoted $y(k)$. For first order Volterra series models, i.e. finite impulse response models, the output can be defined as $y(k) = C^T \boldsymbol{x}(k)$. Then, the output matrix contains the model parameters $C = [h_1(1)\ \ h_1(2)\ \ \ldots h_1(N_t)]$.

It is clear that the second order Volterra series model in state space representation (2.58) can be used in control strategies to predict the future output evolution for a

given input. Based directly on (2.58), a prediction model can be defined as:

$$\begin{aligned} \boldsymbol{x}(k+i+1|k) &= A\boldsymbol{x}(k+i|k) + Bu(k+i|k) \\ y(k+i|k) &= \mathrm{f}(\boldsymbol{x}(k+i|k)) \end{aligned} \tag{2.60}$$

where $\boldsymbol{x}(k+i+1|k)$ represents the state vector for $k+i+1$ predicted at $k$ and $\boldsymbol{x}(k|k) = \boldsymbol{x}(k)$ denotes the initial state for the prediction. The prediction model of a second order Volterra series model in state-space representation (2.60) considering the estimation error as a result of model mismatch can be written as:

$$\begin{aligned} \boldsymbol{x}(k+i+1|k) &= A\boldsymbol{x}(k+i|k) + Bu(k+i|k) \\ y(k+i|k) &= \mathrm{f}(\boldsymbol{x}(k+i|k)) + d(k) \end{aligned} \tag{2.61}$$

## 2.7 Conclusions of the chapter

In this chapter the basic structure of Volterra series models has been presented, with a special emphasis on the second order type used frequently in this document. Being linear in the parameters, the identification of the parameters of a Volterra series model can be carried out easily with the least squares method using input-output data from the process to be modeled. In order to obtain appropriate input-output data for identification the pseudo random multilevel sequence, a signal with white-noise-like properties which sufficiently excites the system, has been presented. For the second order Volterra series model a prediction model for posterior use in model-based predictive control has been defined. Furthermore, the cost functions with and without penalization of the control effort were presented. Based on these cost functions, the general optimization problem of model predictive control based on a second order Volterra series model was defined. Finally, it was shown that second order Volterra models can be expressed easily in state-space representation which is necessary for the stability analysis of model-based predictive control strategies.

As mentioned in section 2.1, Volterra series models can approximate arbitrarily well any fading memory system. This property allows the modeling of a wide range of stable processes by Volterra series models, but eliminates the possibility to model unstable systems. One of the major advantages of Volterra series models is the linearity in the model parameters which allows the usage of linear parameter identification techniques for nonlinear Volterra series model. A possible drawback is the high number of parameters of the used model requiring large sets of data for the identification process. If only a small data set is available, an autoregressive Volterra series model with a reduced number of parameters can be identified. If necessary, autoregressive Volterra series can be easily transformed to a non-autoregressive form and used in MPC

strategies based on non-autoregressive Volterra series models. Furthermore, second order Volterra series models in combination with the usually quadratic cost functions of model-based predictive control lead to possibly non-convex forth order polynomials, representing an optimization problem when computing the future input sequence.

# Chapter 3

# Benchmark systems

This chapter gives a detailed description of the real processes used as benchmark systems for the application of different model-based predictive control strategies. For their use in model-based control strategies the plants will be approximated by second order Volterra series models and the model parameters will be identified from input-output data obtained from experiments with the different processes.

The main benchmark system used in this document is a lab-scale pilot plant emulating an exothermic chemical reaction in continuous operation. In a first step a short introduction to the operating mode of the laboratory process will be given and a simulation model based on first principles will be presented. Finally, for control purposes the nonlinear temperature dynamics of the pilot plant will be approximated by a second order Volterra series model. The second benchmark is a commercial proton exchange membrane fuel cell, characterized by the fast dynamic of the electrochemical reaction. Analogously to the pilot plant, a description of the process and the oxygen consumption will be given. After defining the control objective, a Volterra series model of the fuel cell dynamics will be identified. The third benchmark system is an experimental greenhouse which requires a regulated inside temperature to ensure correct crop growth. The greenhouse dynamics will be approximated by a second order Volterra series model under consideration of several disturbances which influence the mentioned temperature in the greenhouse. The chosen benchmark systems are representative for a wide range of processes with different time scales and degrees of nonlinearity and therefore show that the results of this thesis have a good applicability to real processes.

The main focus of this chapter lies on the approximation of the benchmarks systems' dynamics by second order Volterra series models. For the three benchmark systems,

the dynamics of the variable to be controlled will be identified from experimental input-output data by means of the least squares estimation method.

## 3.1 Pilot plant

One of the benchmark systems used to apply different model-based predictive control strategies is a lab-scale pilot plant which allows the simulation of processes with variables commonly found in industrial processes as temperatures and flow rates. The system is equipped with different sensors for measurement and several actuators for process control. In this document the mentioned pilot plant is used to emulate an exothermic chemical reaction based on temperature changes. With the help of a mathematical model the emulation of the chemical reaction is carried out.

The plant is located in the laboratory of the Department of Systems Engineering and Automation (University of Seville) and has been studied before by several authors [129, 29, 30] and used as a benchmark for control purposes [108, 107, 99, 100]. The following sections explain the operating modes of the pilot plant and give a description of the control and monitoring system. Afterwards, the mathematical model of the exothermic chemical reaction will be given and the used method for emulation will be explained. Finally, the identification of the process' dynamics by means of a second order Volterra series model will be shown.

### 3.1.1 Process description

The pilot plant shown in Fig. 3.1 emulates a continuous stirred tank reactor (CSTR) and contains several means to cool or to heat the fluid in the reactor. The main elements of the system are the tank reactor, an electric resistance, a cooling jacket, a valve to manipulate the flow rate through the cooling jacket as well as a water tank. The general plant structure with the mentioned elements is given in the schematic diagram in Fig. 3.2.

The pilot plant is operated with water, both in the reactor and the cooling jacket. The reactor has a capacity of approximately 25 l and is equipped with an electric resistance in its interior. The electric resistance has a maximum power of 14.4 kW and can be used to supply caloric energy to the water in the tank. On the other hand the cooling jacket is used to reduce the caloric energy of the tank reactor. The heat

**Figure 3.1**: Pilot plant used to study different model-based predictive control strategies.

dissipation can be regulated by the aperture $v_8$ of the valve manipulating the flow rate $F_j$ through the cooling jacket (see Fig. 3.2). The cooling fluid, water, circulating through the cooling jacket is taken from a tank with a capacity of 1 $m^3$. After circulating through the jacket the cooling fluid returns to the tank. To maintain the temperature of the cold water in a certain interval the tank has an auxiliary cooler controlled by a thermostat which maintains the temperature $T_{T2}$ between 18 °C and 19 °C.

The sensors and actuators of the plant are connected to a Schneider M340 programmable automation controller (PAC). The M340 is connected by Ethernet to a personal computer that runs the Unity Pro software package. The control algorithms proposed in the following chapters have been implemented directly in Matlab/Simulink and the communication with Unity Pro is done using the OPC protocol (OLE for Process Control). Hence, both the Unity Pro environment and the controller implemented

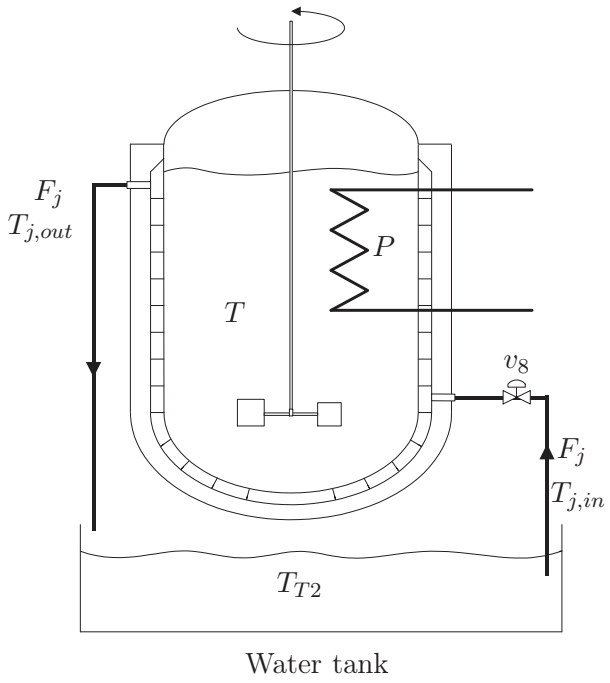


**Figure 3.2**: Diagram of the pilot plant with its main elements.

in Matlab/Simulink run on the same personal computer based on a Pentium 4 processor with 3 GHz using Windows XP as operating system.

### 3.1.2 Emulation of a chemical reaction

The pilot plant (see Fig. 3.1) is used to emulate an exothermic chemical reaction as in [121]. For the emulation, the energy generated by the chemical reaction is calculated with a mathematical model of the reaction and supplied by the electric resistance. The use of a resistance to emulate an exothermic reaction has the advantage that no chemical reaction takes place in the reactor, which allows a safer and cheaper use of the process, while real industrial instrumentation and equipment are used.

The emulated chemical reaction considered in this documents represents a refinement process where a reactant $A$ is transformed to a substance $B$ generating caloric energy. The reaction is generally defined by $A \rightarrow B$ and was used previously in [63] and [44]. The reactant is supplied to the reactor by the emulated feed $F_{f,in}$ to keep the chemical reaction active. Before entering the reactor, the feed passes through a heat exchanger in order to reduce the temperature difference between the feed and the reactor content. The emulated outflow $F_{f,out}$ is used to keep the volume of the reactor content constant. As a consequence, as feed and outflow have the same flow rate and

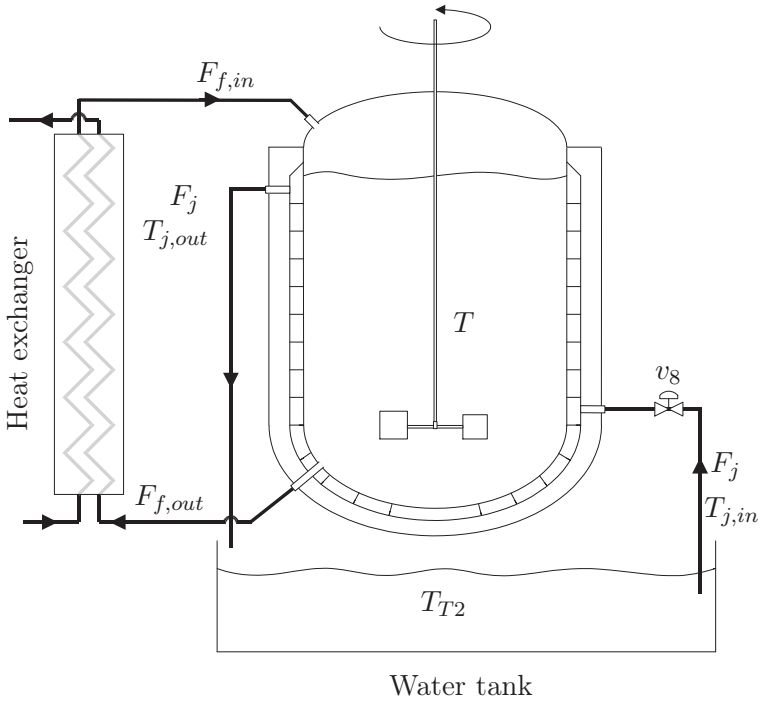


**Figure 3.3**: Diagram of the CSTR emulated by the pilot plant showing the emulated feed $F_{f,in}$ and outflow $F_{f,out}$.

nearly the same temperature, the two flows hardly provoke temperature changes in the interior of the reactor. Figure 3.3 shows the diagram of the pilot plant with the emulated feed $F_{f,in}$ and outflow $F_{f,out}$.

Considering identical flow rates for the feed and the outflow, i.e $F_f = F_{f,in} = F_{f,out}$, the reactor volume $V$ and mass $M$ are constant. The temperature changes of the reactor content can then be defined as [63, 44]:

$$\frac{\mathrm{d}T}{\mathrm{d}t} = -\frac{F_j}{V}(T_{j,out} - T_{j,in}) + \frac{(-\Delta H)\cdot V}{MC_p}\, k_0\, e^{-E/(RT)} C_A^2 \tag{3.1}$$

where the first term models the heat dissipation by the cooling jacket and the second term denotes the generated heat by the exothermic chemical reaction. The variables $F_j$, $T_{j,in}$ y $T_{j,out}$ represent the flow rate through the cooling jacket and the temperature of the cooling fluid entering and leaving the cooling jacket, respectively. $C_A$ is the concentration of the reactant in the reactor content. It has been assumed that the feed neither supplies nor removes caloric energy from the reactor as the feed passes through a heat exchanger and enters the reactor nearly with the temperature of the reactor content. The reactant concentration $C_A$ in the plant reactor is calculated by [63, 44]:

$$\frac{\mathrm{d}C_A}{\mathrm{d}t} = \frac{F_f}{V}(C_{A,in} - C_A) - k_0\, e^{-E/(RT)} C_A^2 \tag{3.2}$$

| Parameter | Value | Unit |
|---|---|---|
| $k_0$ | $1.2650 \cdot 10^{17}$ | $\frac{\mathrm{l}}{\mathrm{mol \cdot s}}$ |
| $C_p$ | 4.18 | $\frac{\mathrm{kJ}}{\mathrm{K \cdot kg}}$ |
| $\Delta H$ | $-105.57$ | $\frac{\mathrm{kJ}}{\mathrm{mol}}$ |
| $E/R$ | 13550 | K |

| Variable | Value | Unit |
|---|---|---|
| $V$ | 25 | l |
| $M$ | 25 | kg |
| $C_{A,in}$ | 1.2 | $\frac{\mathrm{mol}}{\mathrm{l}}$ |
| $F_f$ | 0.05 | $\frac{\mathrm{l}}{\mathrm{s}}$ |
| $T_{j,in}$ | 291.15 | K |

**Table 3.1**: Model parameters and constant variables of the emulated exothermic chemical reaction.

where the first term represents changes in the reactant concentration due to the feed and the outflow. The second term considers the reduction of the concentration as a result of the reactant consumption by the chemical reaction. $C_{A,in}$ denotes the reactant concentration in the feed. The model parameters and the variables used with constant values are shown in Table 3.1. Figure 3.4 shows the temperature $T$ and the concentration $C_A$ of the reactor in steady state as a function of the flow rate $F_j$ through the cooling jacket calculated with the mathematical model (3.1)-(3.2). For the emulation of the exothermic process the electric resistance supplying caloric energy to the reactor is used. Therefore the concentration $C_A$ is calculated with (3.2) using the measured tank temperature $T$. With the calculated concentration $C_A$ and the measured temperature $T$ the temperature gradient (3.1) can be calculated. Hence, with the known parameters for the reactor mass ($M$) and the specific heat capacity of water ($C_p$) the necessary power to obtain the calculated temperature gradient is defined as:

$$P = C_p M \frac{\mathrm{d}T}{\mathrm{d}t} \tag{3.3}$$

Finally, the electric resistance operated with the necessary power $P$ provides the caloric energy that is supplied to the reactor in order to emulate the chemical reaction. Hence, the chemical reaction can be emulated simulating the differential equations (3.1)-(3.2) and supplying the reaction heat (3.3) by means of the electric resistance.

In order to emphasize the nonlinear character of the process the empirical model for the heat exchange in the cooling jacket and the static nonlinearity relating the flow rate $F_j$ with the valve opening $v_8$ will be given. The heat exchange in the cooling jacket due to the temperature difference is given by the following experimental model [45]:

$$F_j \cdot (T_{j,out} - T_{j,in}) = \frac{T - \alpha}{\beta}(1 - e^{-\gamma F_j}) \tag{3.4}$$

with $\alpha = 291.13\,\mathrm{K}$, $\beta = 19.15\,\mathrm{s/l}$ and $\gamma = 26.14\,\mathrm{s/l}$. Furthermore, the relationship between the flow rate $F_j$ through the cooling jacket and the valve opening $v_8$ is identified

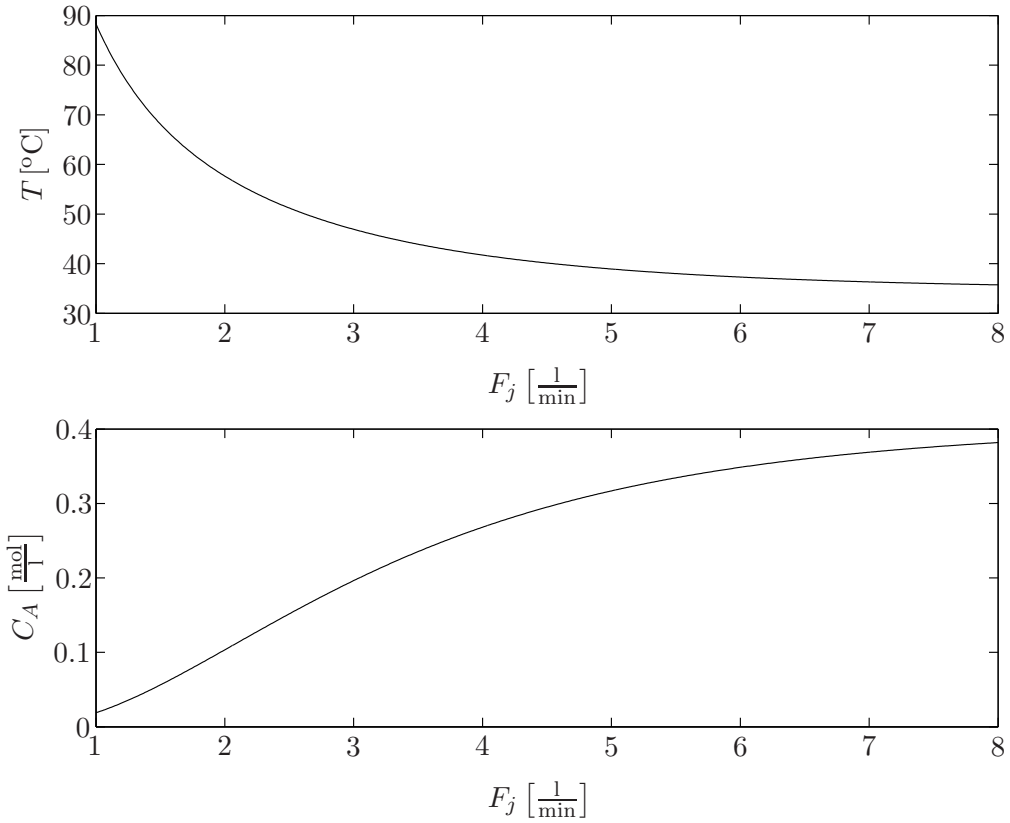


**Figure 3.4**: Temperature $T$ and concentration $C_A$ of the reactor in steady state as a function of the flow rate $F_j$ through the cooling jacket calculated with the equations of the temperature (3.1) and the concentration (3.2).

from experimental data and can be approximated by:

$$F_j = \delta_0 + \delta_1 v_8 + \delta_2 v_8^2 + \delta_3 v_8^3 + \delta_4 v_8^4 + \delta_0 v_8^5 \tag{3.5}$$

with $\delta_0 = 4.84 \cdot 10^{-4}$, $\delta_1 = 3.43 \cdot 10^{-4}$, $\delta_2 = 2.69 \cdot 10^{-4}$, $\delta_3 = 1.29 \cdot 10^{-5}$, $\delta_4 = -1.46 \cdot 10^{-8}$ and $\delta_5 = -6.36 \cdot 10^{-10}$. Note that (3.4) and (3.5) have not been used to emulate the chemical reaction, but can be used for a simulation of the entire process. The nonlinear static relation between the opening $v_8$ and the flow rate $F_j$, based on (3.5), is shown in Fig. 3.5.

The chemical reaction is nonlinear in the dynamics of the temperature and the concentration due to the exponential function and the quadratic terms of the concentration in the model equations (3.1) and (3.2). The relation between the opening of the valve $v_8$ and the flow rate $F_j$ through the cooling jacket, given in (3.5), adds some static nonlinearity to the model. Furthermore, the heat exchanger also shows a static nonlinear behavior, converting the process (3.1)-(3.5) in a highly nonlinear system.

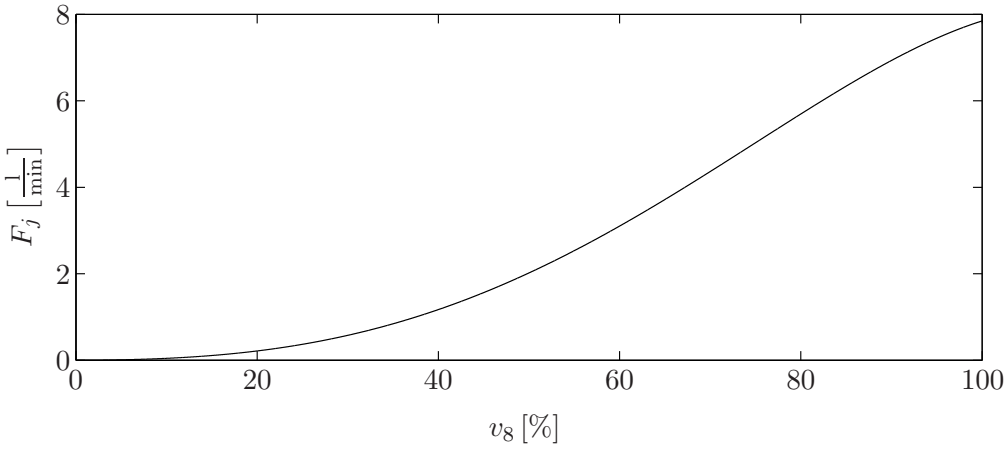


**Figure 3.5**: Static relation between the flow rate $F_j$ and the opening of valve $v_8$.

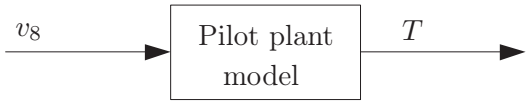


**Figure 3.6**: Block diagram of the considered model of the pilot plant with the model input (valve opening $v_8$) and output (temperature $T$).

### 3.1.3 Identification of a Volterra series model

For control purposes the pilot plant emulating the exothermic chemical reaction will be approximated by a second order Volterra series model with the valve opening $v_8$ being the model input and the temperature $T$ being the output (see Fig. 3.6). The identification requires, as already mentioned in Section 2.2, adequate experimental input-output data for the parameter estimation. Therefore a pseudo random multilevel sequence has been applied to the valve $v_8$ of the plant in order to obtain the necessary data. As showen by [94, 95] the parameter estimation for a second order Volterra series model requires an input sequence with at least 3 different levels. For the pilot plant emulating an exothermic chemical reaction a PRMS with input sequence values of $v_8 = \{40, 60, 80\}\,\%$ and periods between 20 and 60 minutes has been chosen. The resulting output of the system, the temperature $T$, varies between 38 °C and 79 °C, covering a large interval of possible operating points, see Fig. 3.7.

With the obtained experimental input-output data the parameter identification was carried out by means of the least squares estimation method. During the identification process different sampling times and truncation orders for the chosen model were tested. Finally a sampling time of $t_s = 60\,\mathrm{s}$ was chosen and the truncation orders of $N_1 = 60$ and $N_2 = 30$ were determined. In order to reduce the number of parameters

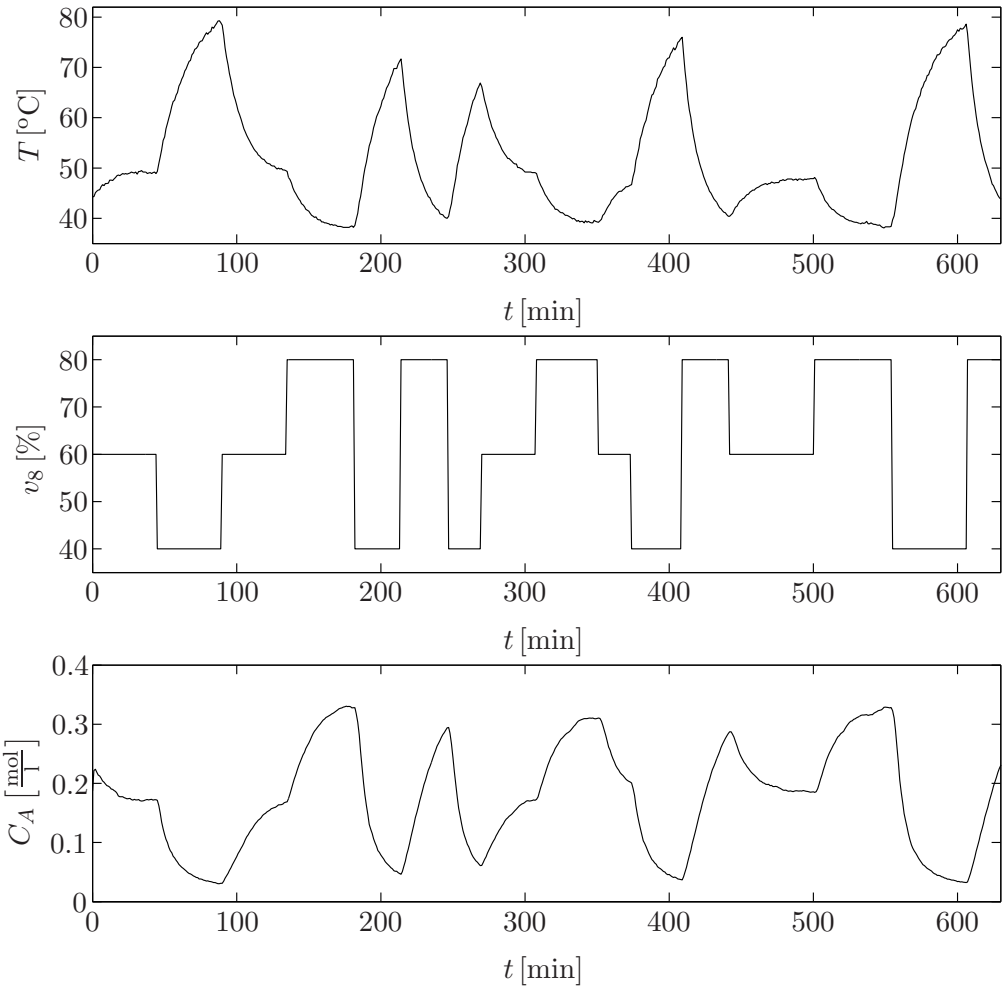


**Figure 3.7**: Experimental data set with temperature $T$, aperture of the valve $v_8$ and emulated concentration $C_A$ used for the identification of a Volterra series model of the pilot plant with an exothermic chemical reaction.

to be identified, a diagonal second order Volterra series model has been chosen to approximate the process dynamics. The use of a general, non-diagonal second order Volterra series model would require a larger data set for the parameter identification. The finally identified linear and nonlinear term parameters are shown in Fig. 3.8. It can be observed that both the linear and nonlinear term parameters tend to 0 along their truncation orders and, as a consequence, support the chosen values for $N_1$ and $N_2$.

In order to validate the identified second order Volterra series model another experimental data set (see Fig. 3.9) obtained from the process was used. During the experiment step-like changes in the input signal were applied to the process. For the input sequence the same levels as for the PRMS (see Fig. 3.7) have been used, i.e.

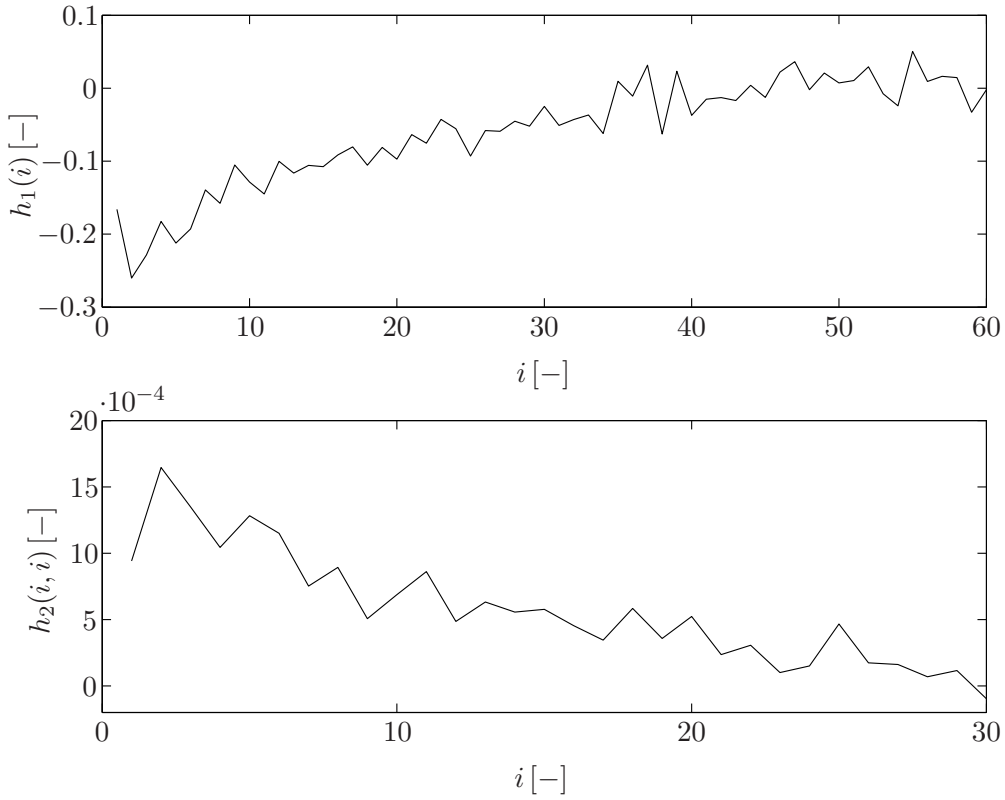


**Figure 3.8**: Identified parameters $h_1$ and $h_2$ for the diagonal second order Volterra series model approximating the pilot plant emulating an exothermic chemical reaction.

$v_8 = \{40, 60, 80\}$ %. In contrast to the first experiment, the time between two changes in the input value was set to a constant value of 69 minutes. It can be observed that the second experiment covers the same temperature interval like the first experiment as a result of the identical input values. A comparison of the measured process output and the output of the identified model can be seen in Fig. 3.10, both for the identification and the validation. For the identification the model shows a good fit with a small error for high output values. It can be observed that the identified model is a good approximation of the nonlinear process dynamics capable to reproduce the nonlinear gain and the dynamic behavior of the pilot plant emulating a chemical reaction. With respect to the data set used for validation, the identified model shows its ability to approximate the given process with the same error for high output values as already seen in the identification. In general terms, the validation underlines the goodness of the identified model and approves the chosen sampling time $t_s$ and truncation orders $N_1$ and $N_2$.

The quality of the identification and the validation was checked by means of the

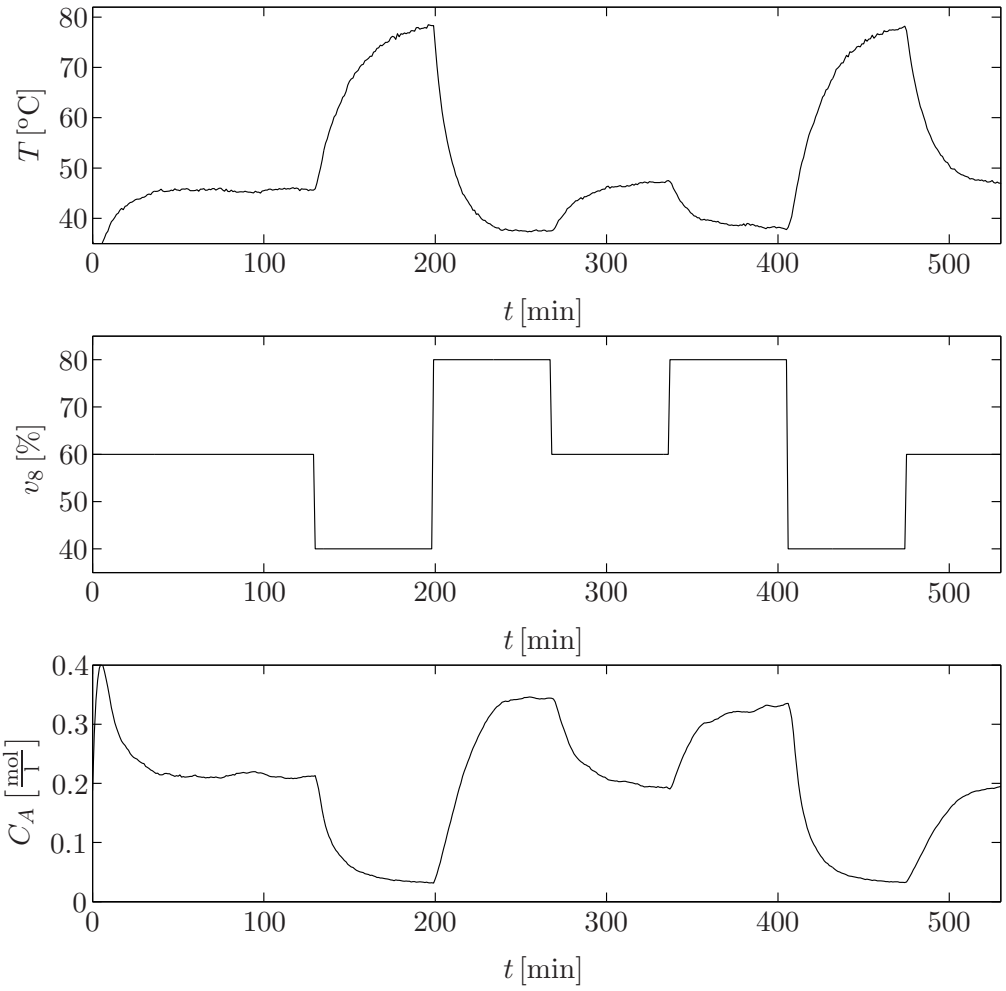


**Figure 3.9**: Experimental data set with temperature $T$, aperture of the valve $v_8$ and emulated concentration $C_A$ used for the validation of the identified Volterra series model of the pilot plant with an exothermic chemical reaction.

Mean Square Error (MSE) defined by:

$$\varepsilon = \frac{\sum_{k=1}^{M} (y(k) - \hat{y}(k))^2}{M} \tag{3.6}$$

where $y(k)$ denotes the measured process output, $\hat{y}(k)$ is the simulated model output and $M$ is the size of the data set, i.e. number of used samples. The obtained MSE for the identification was $\varepsilon_{id} = 1.304$ whereas the MSE for the validation was $\varepsilon_{val} = 1.425$. With the identified and validated model the static relation between input and output, i.e. the model output for a constant input signal, can be obtained. The static output

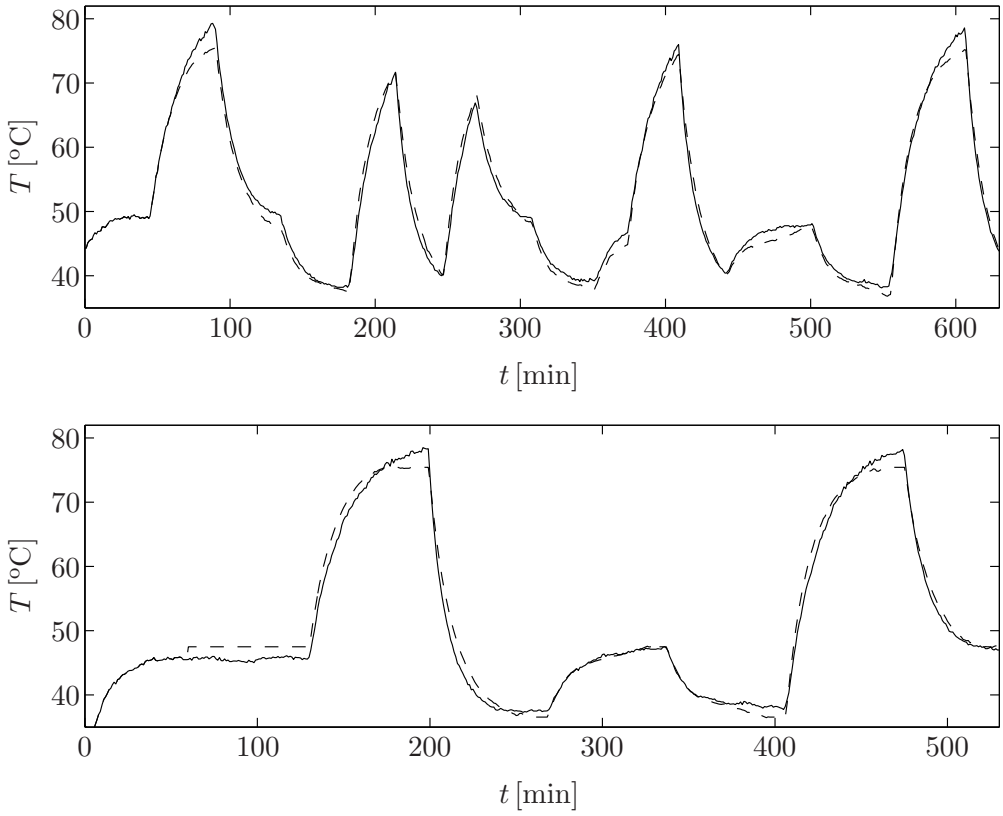


**Figure 3.10**: Comparison of the measured process output (solid line) and the output of the identified second order Volterra series model (dashed line) both for the identification and the validation.

for the used second order diagonal Volterra series model can be calculated by:

$$\hat{y}_{stat} = h_0 + u_{stat} \cdot \sum_{i=1}^{N_1} h_1(i) + u_{stat}^2 \cdot \sum_{i=1}^{N_2} h_2(i,i) \tag{3.7}$$

The static characteristic for the interval of input values used during the two experiments, i.e $v_8 = \{40, 80\}$ %, is given in Fig. 3.11. It can be observed that static output ranges from approximately 37 °C to 79 °C, corresponding to the values encountered in the experimental results. Finally, the reasonable shape of the parameters (see Fig. 3.8), the graphical results of the identification and validation (see 3.10) as well as the mean square errors for the identification and validation show the goodness of the identified model and emphasize the capability of the chosen model to approximate the nonlinear dynamics of the pilot plant emulating an exothermic chemical process.

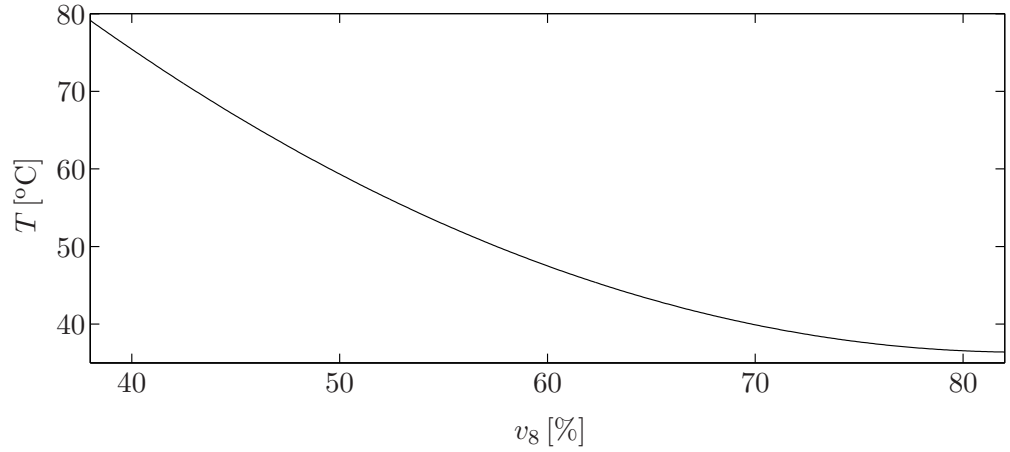


**Figure 3.11**: Static characteristic of the identified second order Volterra series model as a function of the input value.

## 3.2 Fuel cell

The second system used as a benchmark for control purposes is a fuel cell (Ballard's Nexa Power Module [11]), the first commercially available fuel cell module (see Fig. 3.12). Fuel cells are electrochemical devices that generate electric energy from reactants continually, this while fuel and oxidant are provided. Although they were invented more than a century ago, they have received a great deal of attention in the last decades as good candidates for clean electricity generation both in stationary and automotive applications. There are many unresolved issues regarding materials, manufacturing and maintenance, automatic control being one of the most important ones. There are many types of fuel cells, this section being focussed on a Polymer Electrolyte Membrane (PEM) fuel cell, which runs at low temperature and shows fast dynamical response, high power density, small size, low corrosion and high efficiency, which makes it suitable for mobile applications [114, 127]. It is clear that good performance of these devices is closely related to the kind of control that is used, so a study of new control alternatives is justified.

The used fuel cell module can be considered as a benchmark system since it is widely used by many research groups and it is representative of state-of-the-art PEM technology. The mentioned fuel cell module is located in the laboratory of the Department of Systems Engineering and Automation (University of Seville) and has been used by several researchers to test different control strategies [8, 109, 40]. The following sections give a general description of the fuel cell system and the oxygen excess ratio as controlled variable and show the identification of the process dynamics by means of a second order Volterra series model.

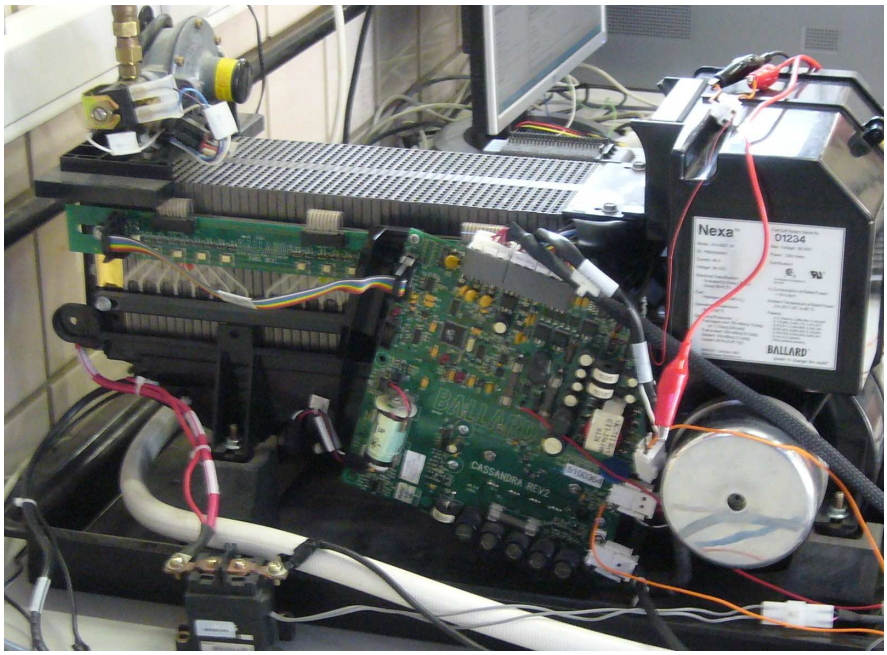


**Figure 3.12**: The Ballard Nexa Power Module [11] used for control purposes.

### 3.2.1 Process description

The fuel cell system used in this document is a commercial PEM fuel cell using hydrogen and oxygen from ambient air as fuel and oxidation medium, respectively. The hydrogen, stored in a tank, enters the stack on the anode side with a constant pressure. On the other side, the compressor aspirates environmental air and supplies it to the stack on the cathode side (see Fig. 3.13). In the anode the hydrogen dissociates into protons and electrons of which the protons are conducted through the membrane towards the cathode. Due to the electric isolation of the membrane the electrons cannot pass through the membrane and are forced through an external circuit generating electric energy. In the cathode, the oxygen molecules, the electrons and the protons react and form water as a residual [103]:

$$O_2 + 4H^+ + 4e^- \Rightarrow 2H_2O \tag{3.8}$$

Hydrogen and oxygen must be continuously supplied to the fuel cell in order to maintain the electrochemical reaction. The main control variable in the used PEM fuel cell power system is the air flow which is supplied by a compressor. Hydrogen is fed through a fast opening valve to track a desired ratio of the air flow. The ratio between the oxygen consumed due to the electrochemical reaction and the oxygen flow generated by the compressor is denoted as oxygen excess ratio (for details on the oxygen excess ratio see Section 3.2.2). This ratio must fulfil the stoichiometric relation

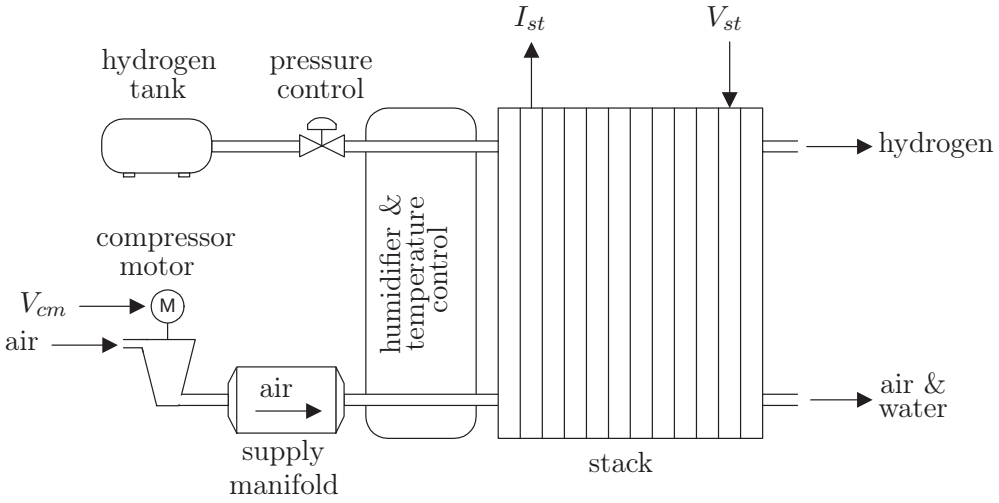


**Figure 3.13**: General scheme of the fuel cell containing the stack and the auxiliary components.

required to produce the demanded current, otherwise a phenomenon called oxygen starvation occurs. This phenomenon implies a fast stack degradation and low power generation, and the only way to finish it is by stopping the reactants flow and the current demand [18]. Different studies are reported about this undesired phenomenon, and in the literature controlling the oxygen excess ratio is proposed in order to prevent it [102, 131]. The fast system dynamics are a challenge in fuel cell control as they require the development of control strategies with low computational complexities in order to calculate a new input signal within one sampling period.

The used fuel cell stack is composed of 46 cells connected in series with a voltage of 26 V at rated power and a rated current of 46 A. The air supply is auto-humidified before reaching the cells and the fuel cell stack is air-cooled by an integrated small fan [11]. The hydrogen feeding is carried out in dead-end mode, i.e. with a closed anode outlet, supplying hydrogen at the exact rate at which it is consumed [12]. The anode-cathode pressure ratio is controlled to avoid membrane stress or damage, and therefore the hydrogen and oxygen mass flows are correlated. In order to simulate a variable current demand the fuel cell is connected to a load bank (see Fig. 3.14). Furthermore a real-time system is used for the data acquisition and for control purposes.

### 3.2.2 Oxygen excess ratio

Performance and safety issues of fuel cells are closely related to the regulation of the oxygen excess ratio $\lambda_{O_2}$, defined as the ratio between the oxygen $W_{O_2,ca,in}$ entering the

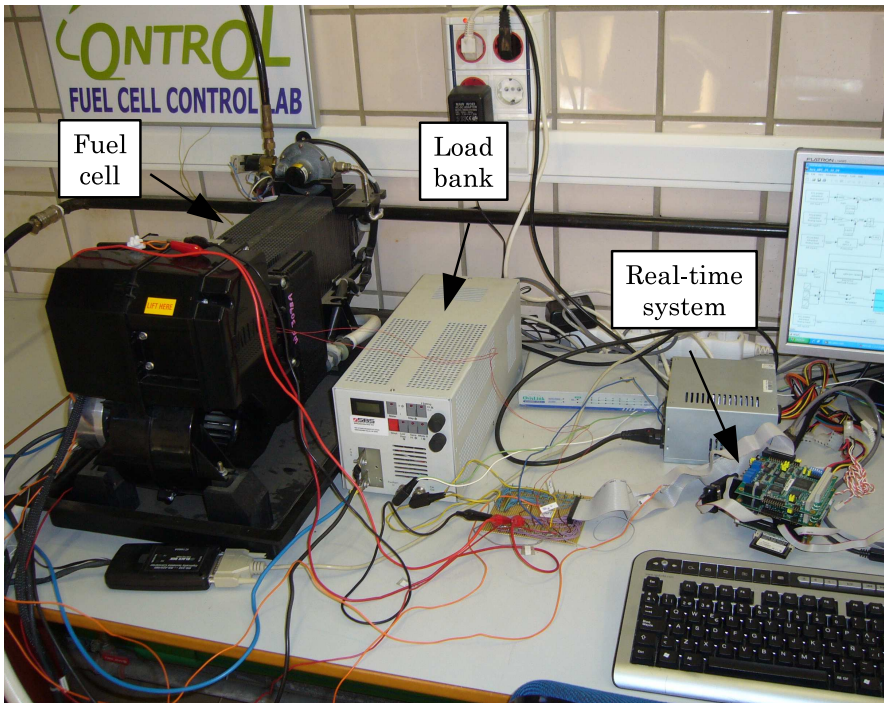


**Figure 3.14**: Experimental setup including the fuel cell, the load bank and the real-time system.

cathode and the oxygen $W_{O_2,reacted}$ reacting in the fuel cell stack. The oxygen excess ratio $\lambda_{O_2}$ is considered a performance variable of the system and its regulation is an important issue since this parameter has a decisive influence on the safety and longevity of the fuel cell.

The oxygen excess ratio has been defined in [103] as:

$$\lambda_{O_2} = \frac{W_{O_2,ca,in}}{W_{O_2,reacted}} \tag{3.9}$$

where the oxygen consumption rate $W_{O_2,reacted}$ is proportional to the stack current $I_{st}$ and given by [61]:

$$W_{O_2,reacted} = M_{O_2}\frac{nI_{st}}{4F} \tag{3.10}$$

with $n$ the number of cells of the used fuel cell stack, $M_{O_2}$ the molar mass of oxygen and $F$ the Faraday constant. The oxygen mass flow rate $W_{O_2,ca,in}$ entering the cathode channel depends on the mass flow rate of dry air $W_{a,ca,in}$ at the cathode inlet:

$$W_{O_2,ca,in} = x_{O_2,ca,in}W_{a,ca,in} \tag{3.11}$$

The oxygen mass fraction $x_{O_2,ca,in}$ can be calculated by:

$$x_{O_2,ca,in} = \frac{y_{O_2,ca,in}M_{O_2}}{y_{O_2,ca,in}M_{O_2} + (1 - y_{O_2,ca,in})M_{N_2}} \tag{3.12}$$

where $M_{N_2}$ denotes the molar mass of nitrogen and $y_{O_2,ca,in}$ represents the oxygen mole fraction. The mass flow rate of dry air at the cathode inlet is defined as:

$$W_{a,ca,in} = \frac{1}{1+\omega_{ca,in}} W_{ca,in} \tag{3.13}$$

with the humidity ratio:

$$\omega_{ca,in} = \frac{M_v}{M_{a,ca,in}} \frac{p_{v,ca,in}}{p_{a,ca,in}} \tag{3.14}$$

being $M_v$ the molar mass of vapor and $M_{a,ca,in}$ the molar mass of air at the cathode inlet. The molar mass $M_{a,ca,in}$ is defined generally by:

$$M_{a,ca,in} = y_{O_2,ca,in} M_{O_2} + (1 - y_{O_2,ca,in}) M_{N_2} \tag{3.15}$$

The vapor pressure $p_{v,ca,in}$ and the dry air pressure $p_{a,ca,in}$ used to calculate the humidity ratio $\omega_{ca,in}$ are defined as:

$$p_{v,ca,in} = \phi_{ca,in} p_{sat}(T_{ca,in}) \tag{3.16}$$

$$p_{a,ca,in} = p_{ca,in} - p_{v,ca,in} \tag{3.17}$$

with $\phi_{ca,in}$ denoting the relative humidity of air at the cathode inlet. $p_{sat}(T_{ca,in})$ and $p_{ca,in}$ represent the vapor saturation pressure for the temperature $T_{ca,in}$ and the pressure at the cathode inlet, respectively.

With the sensor voltage $V_{ca,in}$, measuring the air speed at the cathode inlet, the air flow rate $W_{ca,in}$ used in (3.13) can be calculated by [45]:

$$W_{ca,in} = \psi_1 V_{ca,in} + \psi_2 {V_{ca,in}}^2 + \psi_3 {V_{ca,in}}^3 \tag{3.18}$$

with $\psi_1 = 2.654$, $\psi_2 = 1.481$ and $\psi_3 = 1.862$. For the cathode inlet pressure $p_{ca,in}$ a simplified model was used due to the impossibility to measure the pressure directly. This model, developed in [45], expresses the pressure $p_{ca,in}$ as a function of the air flow rate $W_{ca,in}$ at the cathode inlet and the stack current $I_{st}$:

$$p_{ca,in} = \varphi_1 + \varphi_2 W_{ca,in} + \varphi_3 I_{st} \tag{3.19}$$

with $\varphi_1 = 1.0033$, $\varphi_2 = 2.1 \cdot 10^{-4}$ and $\varphi_3 = -475.7 \cdot 10^{-6}$. Furthermore it is assumed that the relative humidity of the air at the cathode inlet is $\phi_{ca,in} = 1$ and that the temperature $T_{ca,in} = 298.15\,\mathrm{K}$ corresponds to the one of ambient air. In order to calculate the oxygen excess ratio the values of $I_{st}$ and $V_{ca,in}$ are required. Therefore these two values will be measured by sensors with the help of a real-time system (see section 3.2.3). Then, with (3.9)-(3.19), the assumptions about the relative humidity $\phi_{ca,in}$ and temperature $T_{ca,in}$ as well as the measurements $I_{st}$ and $V_{ca,in}$, the oxygen excess ratio $\lambda_{O_2}$ can be easily calculated. The parameters used to calculate the oxygen excess ratio $\lambda_{O_2}$ (3.9) are given in Table 3.2.

| Parameter | Value | Unit |
|---|---|---|
| $n$ | 46 | – |
| $F$ | 96485 | $\frac{\text{C}}{\text{mol}}$ |
| $M_{O_2}$ | 32 | $\frac{\text{g}}{\text{mol}}$ |
| $M_{N_2}$ | 28 | $\frac{\text{g}}{\text{mol}}$ |

| Parameters | Value | Unit |
|---|---|---|
| $M_v$ | 18 | $\frac{\text{g}}{\text{mol}}$ |
| $y_{O_2,ca,in}$ | 0.21 | – |
| $T_{ca,in}$ | 298.15 | K |
| $\varphi_{ca,in}$ | 1 | – |

**Table 3.2**: Model parameters used to calculate the oxygen excess ratio $\lambda_{O_2}$ (3.9) [103].

As already mentioned, the objective of the control strategy is the regulation of the oxygen excess ratio (3.9) which depends on the oxygen consumption rate $W_{O_2,reacted}$ and the oxygen mass flow rate $W_{O_2,ca,in}$. It is clear that the oxygen consumption rate $W_{O_2,reacted}$ (3.10) cannot be influenced as it depends only on the constants $M_{O_2}$, $n$ and $F$ and the measurable, but not manipulable, disturbance $I_{st}$. As a consequence, the regulation of the oxygen excess ratio $\lambda_{O_2}$ has to be carried out by manipulating the oxygen mass flow rate $W_{O_2,ca,in}$ (3.11). The mass flow rate $W_{O_2,ca,in}$ is a function of the mass flow rate of dry air $W_{a,ca,in}$ (3.13) which itself is a function of the air flow rate $W_{ca,in}$ at the cathode inlet. It is clear that the air flow rate $W_{ca,in}$ depends directly on the velocity of the compressor and, as a consequence, on the voltage $V_{cm}$ applied to the compressor motor. Hence, the oxygen mass flow rate $W_{O_2,ca,in}$ (3.11) can be manipulated by means of the compressor motor voltage $V_{cm}$. Therefore the compressor motor voltage $V_{cm}$ will be used as input signal in order to regulate the oxygen excess ratio $\lambda_{O_2}$.

The oxygen starvation phenomenon occurs when the oxygen partial pressure falls below a critical level at any location at the cathode [102]. Tests showed that the oxygen starvation phenomenon can cause damages to the electro-catalyst of the fuel cell, as well as reducing its performance [131]. To avoid this phenomenon, the oxygen excess ratio always has to satisfy $\lambda_{O_2} > 1$. For safety reasons [102] propose to regulate the oxygen excess ratio to $\lambda_{O2} = 2$. This value contains some security margin in order to reduce the probability of starvation and simultaneously provides a good performance. A variable profile for the oxygen excess ratio is proposed in [9], leading to a high efficiency for each value of the load. In this case deviations from the desired value imply lower efficiency, and negative deviations increase the starvation phenomenon probability. Apart from avoiding the oxygen starvation phenomenon the air supply control for a fuel cell should maximize the net output power. The net output power $P_{net}$ can be defined generally as the difference between the power generated by the fuel cell stack $P_{FCS}$ and the power $P_{aux}$ used by the compressor, air fan and other auxiliary equipment, i.e $P_{net} = P_{FCS} - P_{aux}$. For the determination of a suitable oxygen excess

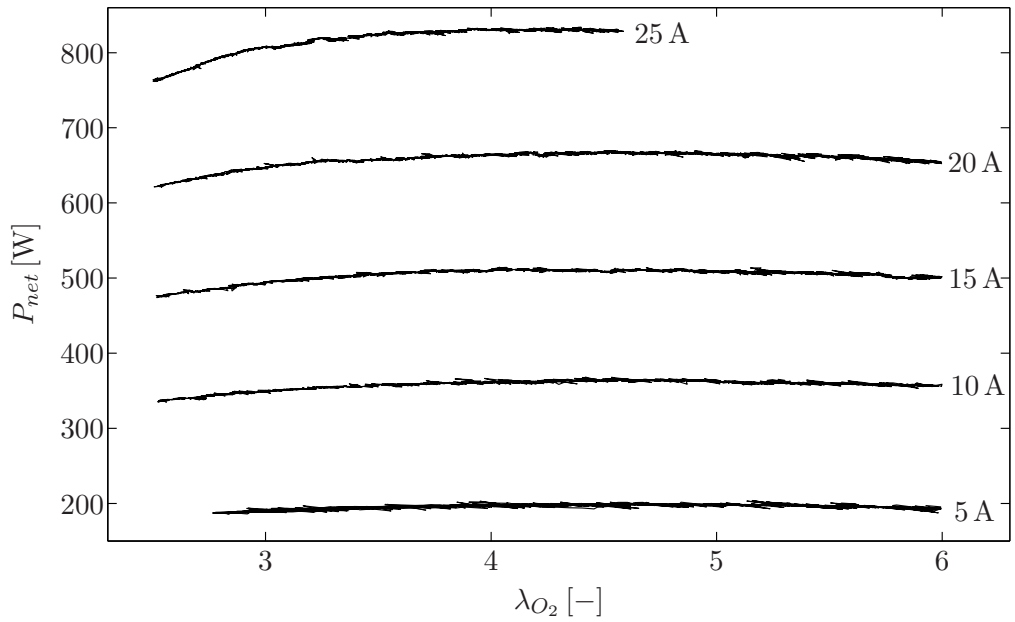


**Figure 3.15**: Net power of the fuel cell module in steady state for different values of the stack current $I_{st}$ and the oxygen excess ratio $\lambda_{O_2}$.

ratio several experiments have been carried out in order to generate a map relating the net output power $P_{net}$ and the oxygen excess ratio $\lambda_{O_2}$ for different stack currents, see Fig. 3.15. In the results it can be observed that the net output power increases for a constant stack current with an increasing oxygen excess ratio until reaching its maximum. Once reached the maximum a further increase of the oxygen excess ratio has a negligible effect on the net output power. For higher values of the oxygen excess ratio the fuel cell is operated in a more efficient region of the electrochemical reaction. However, higher values of the oxygen excess ratio also lead to significant auxiliary loads and reduce the net output power. As a trade-off between a high net output power and a sufficiently big safety margin a regulation of the oxygen excess ratio to a value of $\lambda_{O_2}^{ref} = 4$ has been chosen as control objective.

### 3.2.3 Real-time environment

For control, monitoring and measurement, a real-time system based on the embedded computer standard PC/104 (see Fig. 3.16) has been used. The communication with the fuel cell module is enabled by two Advantech PCM-3718HO multifunction cards, each one offering 16 analog inputs and 1 analog output. For computational purposes an Advantech PCM-3370 CPU (central processing unit) card, based on a 650 MHz Pentium III processor with 256 MB memory, has been used. The three cards, all in the

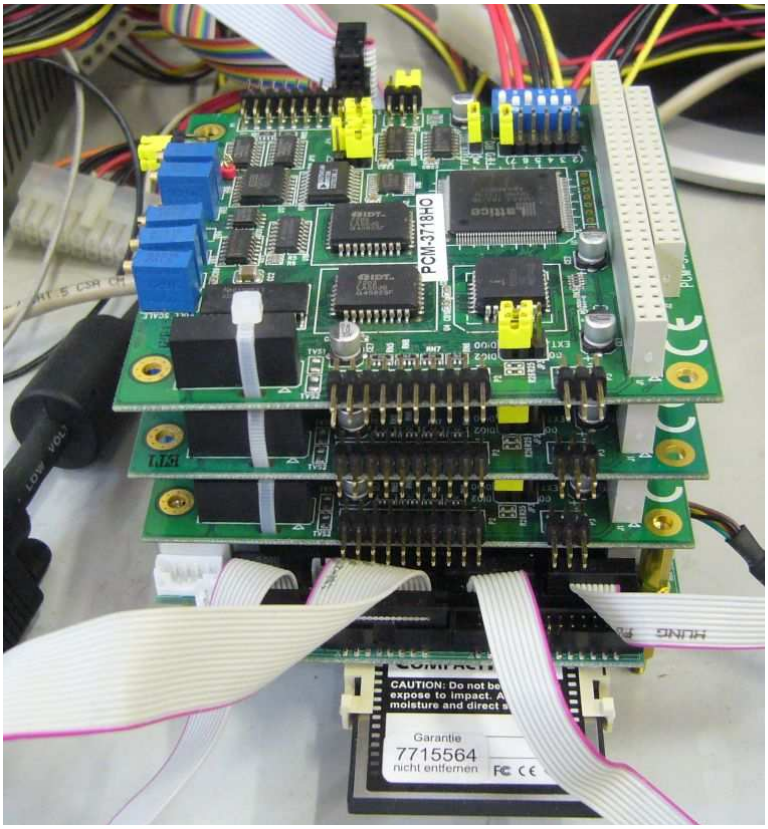


**Figure 3.16**: The embedded computer based on the PC/104 standard used for control, monitoring and measurement purposes.

PC/104 form factor, are interconnected by a PCI (peripheral component interconnect) bus. The real-time system has been used in the configuration shown in Fig. 3.17 with 4 analog inputs and 2 analog outputs. The inputs of the real-time system represent the measurements of the stack voltage $V_{st}$, the stack current $I_{st}$, the sensor voltage $V_{ca,in}$ at the cathode inlet and the load current $I_{load}$. The stack voltage $V_{st}$, the stack current $I_{st}$ and the sensor voltage $V_{ca,in}$ are measured with sensors already placed by the fuel cell manufacturer whereas the load current $I_{load}$, considered as a disturbance, is measured with a sensor at the electronic load bank. One of the analog outputs of the real-time system corresponds to the compressor motor voltage $V_{cm}$, considered as the input signal for the fuel cell control. Note that in the original configuration of the fuel cell module the air supply is controlled by the manufacturer's built-in controller. In this thesis the built-in controller is overridden, as done in [46], in order to validate the later presented control strategies in experiments. The second analog output of the real-time system is the voltage $V_{lb}$ of the load bank which allows the simulation of predefined profiles for the load current $I_{load}$.

The data acquisition, signal processing and fuel cell control with the real-time sys-

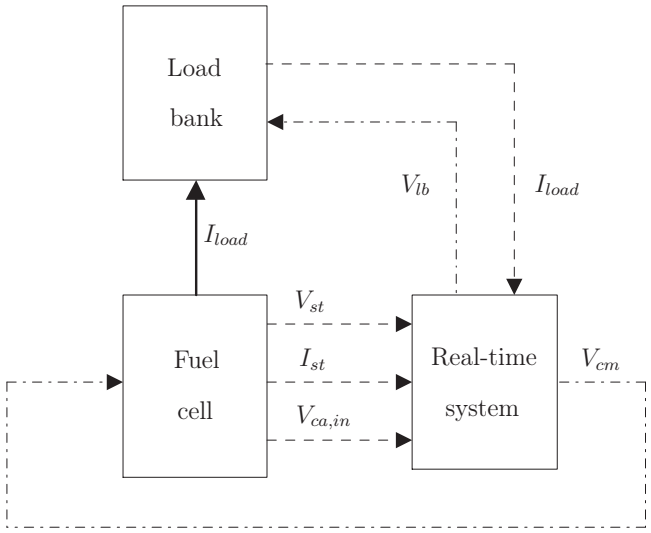


**Figure 3.17**: Used configuration with the information flows (dashed lines for analog inputs and dash-dotted lines for analog outputs) of the real-time system as well as the load current (solid line) demanded by the load bank.

tem has been implemented as a Simulink model. This Simulink model includes the computation of the oxygen excess ratio $\lambda_{O_2}$ (see section 3.2.2) as well as the later presented control strategies to calculate the input signal (compressor motor voltage $V_{cm}$). For its use on the on the embedded computer the Simulink model has to be compiled with the Real-Time Workshop toolbox. The compiled Simulink model already includes a basic operating system and can be used directly on the real-time system without necessity to use an external operating system. The real-time system allows to use a sampling time in the order of a few milliseconds, depending mainly on the computational complexity of the implemented controller.

Note that the fuel cell module delivered by the manufacturer includes its own software for measuring and collecting data of internal states over a serial port. The mentioned software gives access to a wide range of measurements, but with a relatively high sampling time of 200 ms. Furthermore, the measured data from Ballard's software package can be written in a file, but they cannot be accessed in real-time for further purposes.

### 3.2.4 Identification of a Volterra series model

The mathematical model (3.9)-(3.19) cannot be employed directly as a prediction model but it can be used to define a general structure of an adequate prediction model. With

(3.9)-(3.17) the calculation of the oxygen excess ratio $\lambda_{O_2}$ can be expressed in short form as:

$$\lambda_{O_2} = \frac{W_{ca,in}}{I_{st}} \cdot \theta(p_{ca,in},\, T_{ca,in},\;\; \phi_{ca,in},\, n,\, F,\, y_{O_2,ca,in},\, M_{O_2},\, M_{N_2},\, M_v) \tag{3.20}$$

consisting of a division of the air flow rate $W_{ca,in}$ at the cathode inlet and the stack current $I_{st}$ and an additional function containing the remaining variables and parameters. The additional function $\theta(\cdot)$ depends on the pressure $p_{ca,in}$, the temperature $T_{ca,in}$, the relative humidity $\phi_{ca,in}$, the number $n$ of cells of the fuel cell stack, the Faraday constant $F$, the oxygen mole fraction $y_{O_2,ca,in}$ and the molar masses $M_{O_2}$, $M_{N_2}$ and $M_v$ of oxygen, nitrogen and vapor, respectively. The air flow rate $W_{ca,in}$ depends mainly on the applied compressor motor voltage $V_{cm}$ as already mentioned in section 3.2.2. Due to the inertia of the mechanical components of the compressor and its motor the air flow rate $W_{ca,in}$ not only depends on the current value of the compressor motor voltage but also on the compressor motor voltage applied in previous instants, i.e. the air flow rate $W_{ca,in}$ has a dynamic behavior with respect to the applied compressor motor voltage $V_{cm}$. In contrast, the stack current $I_{st}$, considered as a disturbance, has an immediate effect on the oxygen excess ratio $\lambda_{O_2}$.

With the input signal $u$ (compressor motor voltage $V_{cm}$), the disturbance $w$ (stack current $I_{st}$) and the output $y$ (oxygen excess ratio $\lambda_{O_2}$) the second order Volterra series model:

$$y(k) = y_0 + \frac{1}{w(k)} \left( \sum_{i=1}^{N_1} h_1(i)\, u(k-i) + \sum_{i=1}^{N_2} \sum_{j=i}^{N_2} h_2(i,j) u(k-i) u(k-j) \right) \tag{3.21}$$

will be used to approximate (3.20). It can be seen that the chosen model structure considers, analogously to (3.20), the immediate effect of the disturbance on the output. Furthermore, with the last $N_1$ respectively $N_2$ elements of the applied input signal as well as the quadratic term in (3.21) a nonlinear dynamic behavior of the oxygen excess ratio with respect to the compressor motor voltage has been considered.

After the definition of the model structure (3.21) several experiments with the fuel cell module were carried out in order to obtain suitable data for the identification of the model parameters. During the open loop experiments changes in the stack current and the compressor motor voltage were applied, see Fig. 3.19. The stack currents and the compressor motor voltages were chosen so that the resulting oxygen excess ratio lies in an interval of $\lambda_{O_2} = [3,\, 7.2]$, containing the control objective defined in section 3.2.2. In order to obtain suitable data for the parameter identification a pseudo random binary sequence (PRBS) was chosen for the stack current and the compressor motor voltage. As can be seen in Fig. 3.19 a PRBS for the compressor motor voltage was applied to the system while the stack current was held constant during 20 seconds

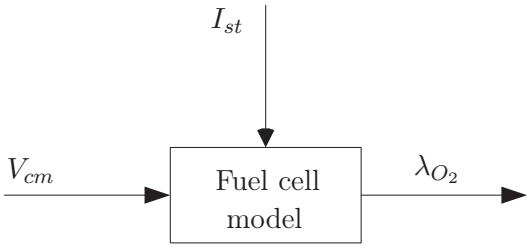


**Figure 3.18**: Block diagram of the considered model of the fuel cell with the model input (compressor motor voltage $V_{cm}$), disturbance (stack current $I_{st}$) and output (oxygen excess ratio $\lambda_{O_2}$).

before changing to a different current stack level. Afterwards, the compressor motor voltage was held constant during 20 seconds before changing to a new value while a PRBS was used for the stack current.

As the chosen Volterra series model (3.21) is linear in the parameters the identification was carried out with the least squares method using the obtained experimental data shown in Fig. 3.19. During the identification process it was observed that the differently delayed second order cross terms ($h_2(i,j)$ for $i \neq j$) had little influence on the model dynamics. Therefore, only the equally delayed cross terms were considered in the chosen model (3.21) and the number of parameters to be identified was reduced considerably. Finally, the model was identified with a sampling time of $t_s = 5\,\text{ms}$ and a truncation order of $N_1 = N_2 = N_t = 16$, i.e the second order Volterra series relating the stack current and the compressor motor voltage with the oxygen excess ratio can be written as:

$$y(k) = y_0 + \frac{1}{w(k)} \left( \sum_{i=1}^{16} h_1(i)\, u(k-i) + h_2(i,i) u(k-i)^2 \right) \tag{3.22}$$

The model offset was identified with $y_0 = 0.072$ and the parameters $h_1(i)$ and $h_2(i,i)$ for $i = 1, \ldots, 16$ are shown in Fig. 3.20. With a truncation order of $N_t = 16$ and the sampling time $t_s = 5\,\text{ms}$ the compressor motor voltage $V_{cm}$ influences the oxygen excess ratio $\lambda_{O_2}$ over an interval of 80 milliseconds. Fig. 3.21 shows a comparison of the oxygen excess ratio $\lambda_{O_2}$ and the output of the identified model (3.22) at different operation points. It can be seen from the figure that the oxygen excess ratio $\lambda_{O_2}$ adopts instantaneously a new value after a change in the stack current $I_{st}$ whereas a variation in the compressor motor voltage $V_{cm}$ shows a dynamic adoption of a new value for the oxygen excess ratio $\lambda_{O_2}$. The reaction to changes in the stack current $I_{st}$ and the compressor motor voltage $V_{cm}$ justifies the chosen model structure. Note that the disturbance in the chosen model structure (3.22) can be considered as a nonlinear

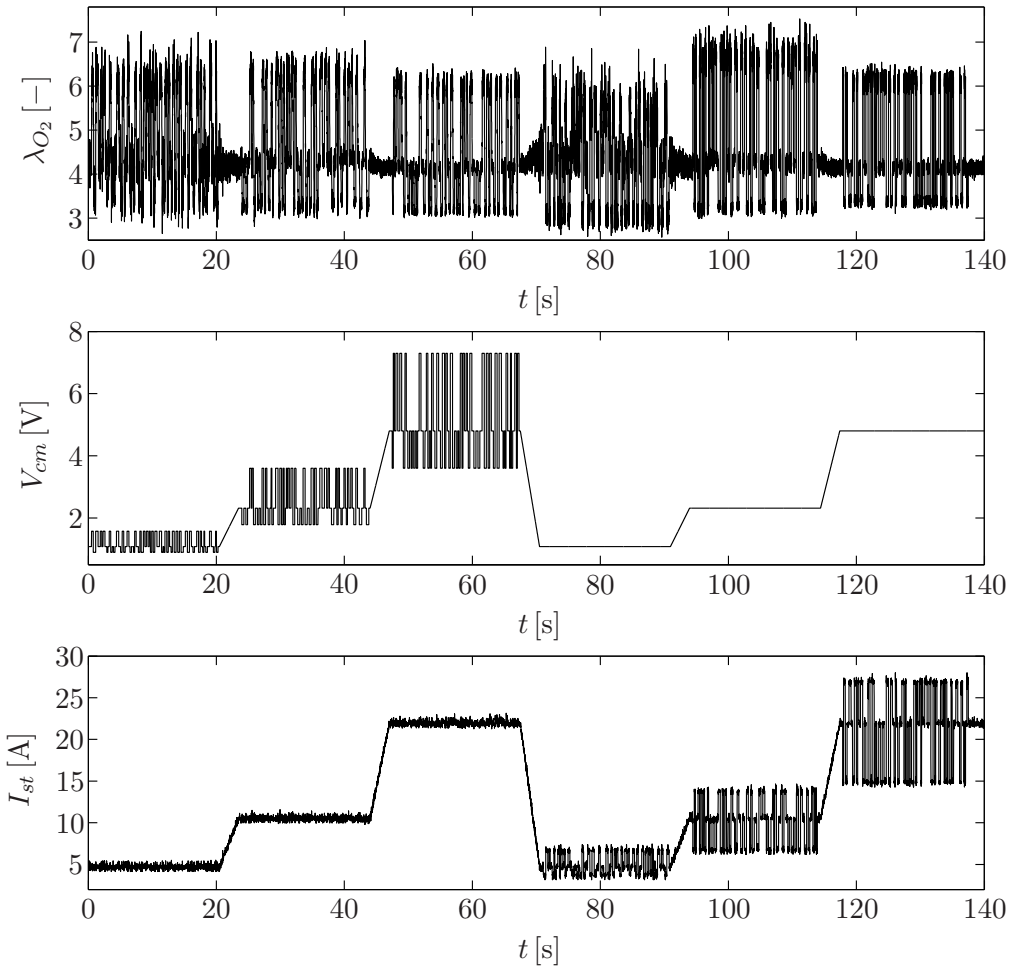


**Figure 3.19**: Experimental results of the fuel cell used for the model identification. Top: oxygen excess ratio $\lambda_{O_2}$, middle: stack current $I_{st}$, bottom: compressor motor voltage $V_{cm}$.

scaling with feedforward effect. As a consequence of this scaling the matrices $G$ and $H$ of the prediction model (2.43)-(2.44) and the matrix $B$ used to compute the vectors $\boldsymbol{f}$ (A.9) and $\boldsymbol{g}$ (A.15) have to be modified. For details on the necessary modifications see the Appendix A.3.

The identified model was validated with a second set of experimental data obtained from the fuel cell module. The used set was generated in an open loop experiment with a PRBS in the compressor motor voltage $V_{cm}$ while the stack current $I_{st}$ was held constant and with a PRBS in the stack current $I_{st}$ for a constant compressor motor voltage $V_{cm}$. The levels for the compressor motor voltage $V_{cm}$ and the stack current $I_{st}$ were different with respect to the experiment used for the parameter identification and resulted in an oxygen excess ratio of $\lambda_{O_2} = [2.7, 8]$. A comparison between the measured oxygen excess ratio $\lambda_{O_2}$ and the output $y$ of the identified model (3.22) can be seen in Fig. 3.22 for some operation points. The results of the validation show that

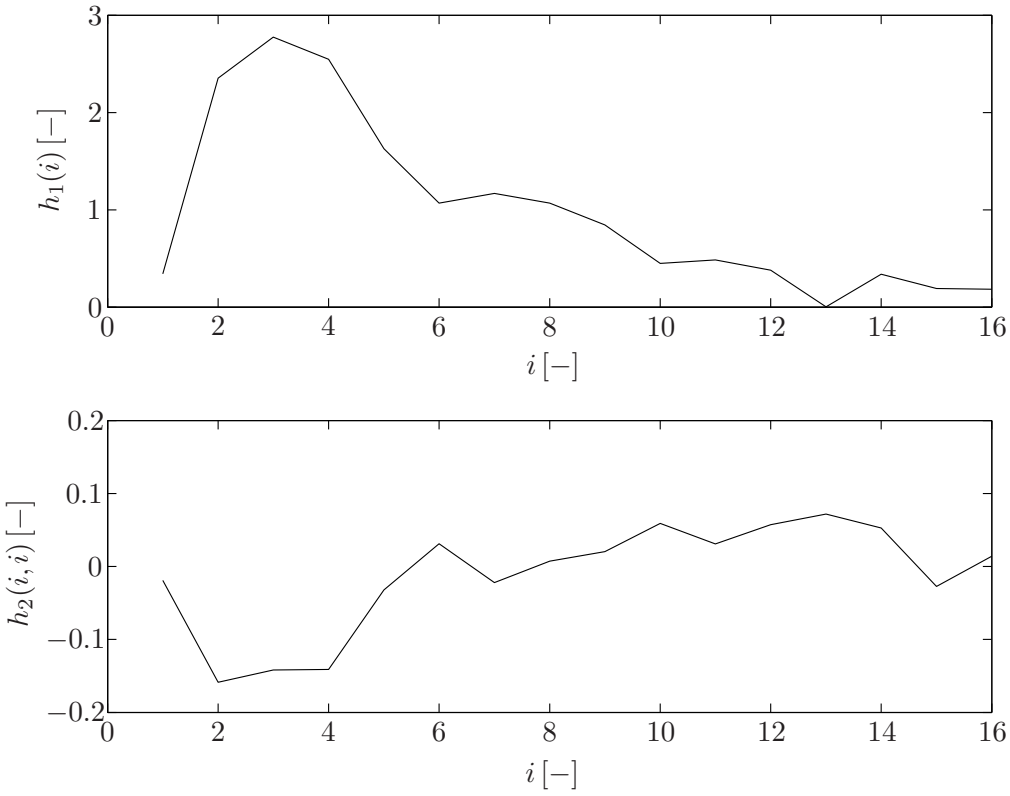


**Figure 3.20**: Identified parameters $h_1(i)$ and $h_2(i,i)$ for $i = 1, \ldots, 16$ for the chosen second order Volterra series model (3.22).

the identified model approximates the dynamical behavior of the system without an offset in steady state. Both the model identification as well as the validation justify the use of only one element of the stack current $I_{st}$ in (3.22). Furthermore, the truncation order of $N_t = 16$ is long enough to model the dynamics related to the compressor motor voltage $V_{cm}$. Finally, the mean square error (3.6) between model output and measured value during identification and validation of the prediction model (3.22) for different operation points was calculated and can be seen in Tab. 3.3. The good fit of the predicted model output in Fig. 3.21 and Fig. 3.22 as well as the computed mean square errors for the identification and validation justify the use of the identified model (3.22).

## 3.3 Greenhouse

Crop growth is mainly influenced by surrounding environmental climatic variables and by the amount of water and fertilizers supplied by irrigation. The proper handling of

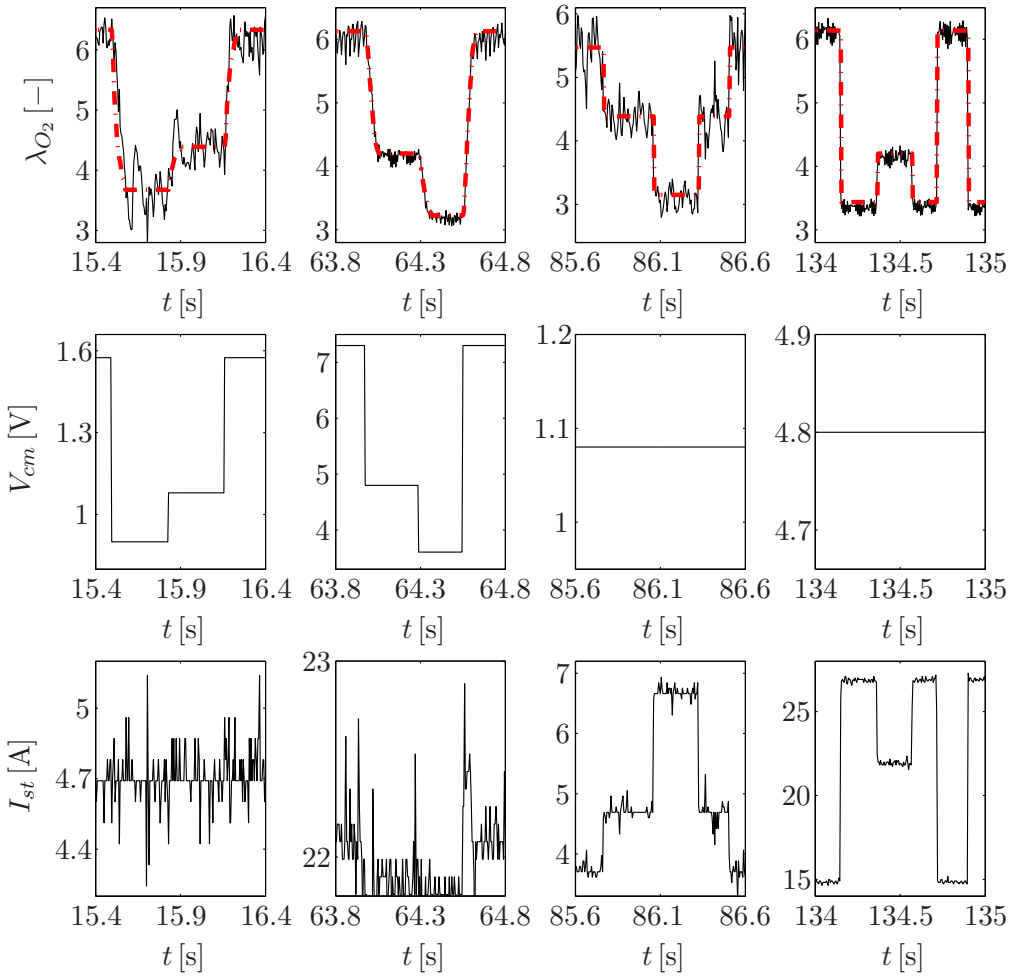


**Figure 3.21**: Results of the model identification. Top: oxygen excess ratio $\lambda_{O_2}$ (solid line) and output of the identified model (3.22) (dash-dotted line), middle: compressor motor voltage $V_{cm}$, bottom: stack current $I_{st}$.

these variables allows the control of crop growth. Therefore, greenhouses are ideal for cultivation as they constitute closed environments in which climatic and fertirrigation variables can be controlled to allow an optimal growth and development of the crop. A greenhouse, representing a light-transmissive construction, augments the temperature in the interior as a consequence of the greenhouse effect and protects from external weather conditions as well as plagues and diseases. Greenhouses are primarily used for agricultural production, but also for research projects as the inside climate can be controlled.

The experimental greenhouse used in this document is located in Las Palmerillas Experimental Station (El Ejido, Almería, Spain) and has been extensively used by different researchers, both for modeling [115, 116, 49] and control application [15, 7,

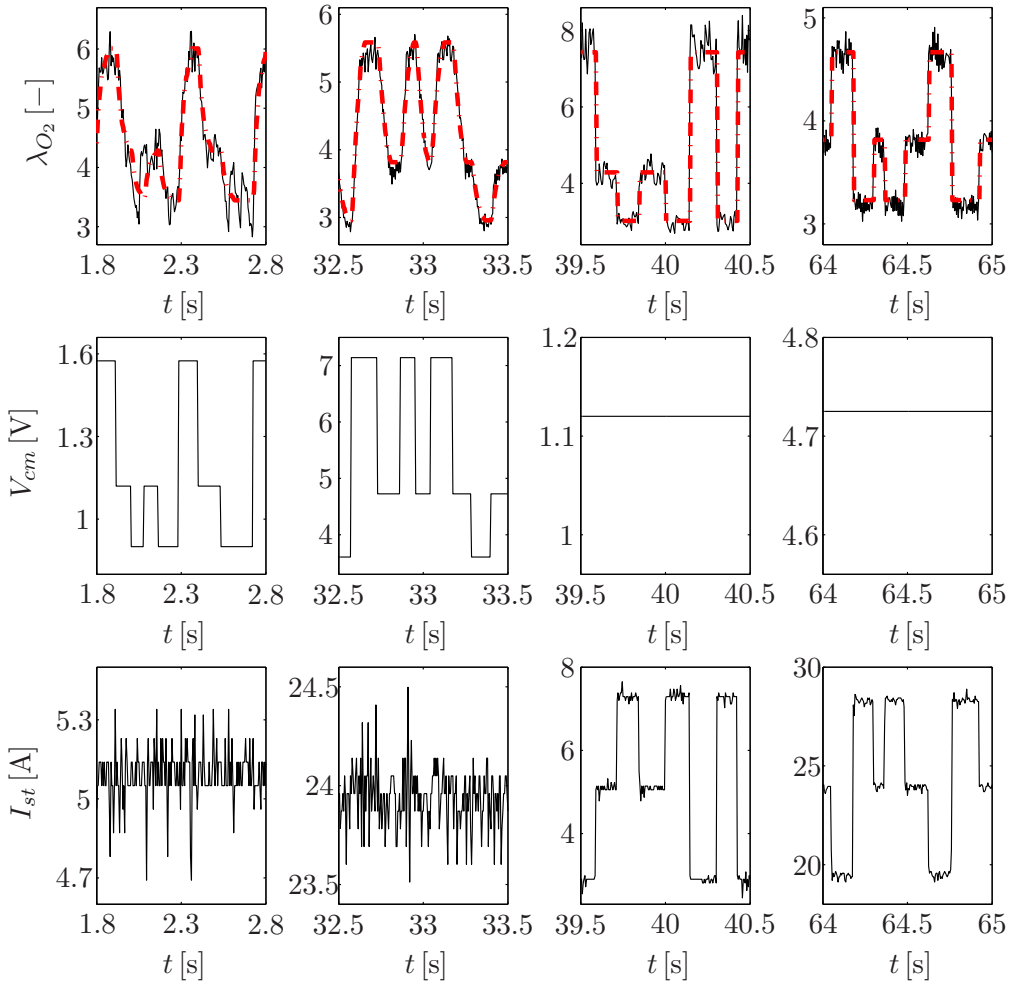


**Figure 3.22**: Results of the model validation. Top: oxygen excess ratio $\lambda_{O_2}$ (solid line) and output of the identified model (3.22) (dash-dotted line), middle: compressor motor voltage $V_{cm}$, bottom: stack current $I_{st}$.

117]. In the following sections a description of the used greenhouse and the data acquisition system will be given and the approximation of the dynamic behavior of the greenhouse using a second order Volterra series model will be shown.

### 3.3.1 Process description

The used greenhouse is an experimental system equipped with sensors and actuators for data acquisition and control purposes (see Fig. 3.23). It is a typical "parral" (flat symmetric roof) greenhouse of this zone with a soil surface of 876 $m^2$and a height of 4.4 m. The greenhouse is oriented in East-West direction and the covering material is a

| Oxygen excess ratio | Identification | Validation |
|---|---|---|
| MSE, overall | 0.0848 | 0.1084 |
| MSE, $\lambda_{O_2} < 3.5$ | 0.0673 | 0.1261 |
| MSE, $3.5 \leq \lambda_{O_2} \leq 5.5$ | 0.0793 | 0.0789 |
| MSE, $\lambda_{O_2} > 5.5$ | 0.1096 | 0.2452 |

**Table 3.3**: Mean square error between model output and measured value during identification and validation of the prediction model (3.22) for different operation points.

200 $\mu$m thick polyethylene film. Its structure is constructed with cross-sectioned type-I rows whose posts are installed in concrete shoes each 2 meters with an inclination of 65 °, with the upper bound united by horizontal cross-sectioned type-L elements. The sidewalls are formed by cables and vertical bars fixed to the ground and united to the horizontal elements. The upper ends of the vertical bars and inclined posts are connected to tightened cables which are part of a 2 m × 2 m mesh. The upper part of the greenhouse is supported by vertical posts with distributed concrete shoes with a distance of 8 m × 2 m between each other.

In the central part of the cover, between the posts placed every 8 m, a small depression or valley can be found. In these valleys the height of the tightened wires is reduced by the vertical bars fixed to the ground every 2 m. The peak altitude of each section, formed between the posts, is 4.4 m and the height from the ground to the valley is about 3.6 m [117]. The cover material is a 0.18 mm thick thermal polyethylene film placed between two meshes of galvanized wire united to the previously mentioned tightened wires. The sidewalls and roof windows are protected by a fine mesh and a plastic foil in order to avoid the entrance of insects into the greenhouse.

The natural ventilation system consists of roof and lateral windows (see Fig. 3.24). The lateral windows are placed in the North and South side of the greenhouse and can be opened with the help of an electric motor. Opening the lateral windows, the plastic foil of the windows is rolled up on an axis connected to the electric motor. The roof windows, also governed by an electric motor, open towards the West with a maximum angle of 45 °. Note that the experimental greenhouse also possesses a heating as well as a artificial irrigation and fertilization system. As the irrigation and fertilization system and the heating have not been used in this work, only the natural ventilation system is described. For a more exhaustive description of the experimental greenhouse see [117].

The experimental greenhouse is equipped with a data acquisition system to collect

**Figure 3.23**: Exterior view of the used experimental greenhouse and photos of the control and monitoring system as well as some of the installed sensors in the interior of the greenhouse.

measurements from the sensors installed in the interior and exterior of the greenhouse. The data acquisition system is based on an industrial Ethernet using Compact FieldPoints (National Instruments) which satisfy the standards for automatization in production environments. During the development of the control system, particular attention has been paid to the devices to be installed due to the specific climate conditions in greenhouses which include relative humidities close to the saturation and high temperatures. The industrial specifications of Compact FieldPoints guarantee a correct operation for temperatures over 70 °C. The modular structure of systems based on Compact FieldPoints allows an easy integration of the devices in the software package LabView (National Instruments) and facilitates the measurement and control of industrial processes.

In the exterior of the greenhouse the following variables are measured: air temperature, relative humidity, global radiation, photosynthetically active radiation, rain detection, wind speed and direction as well as temperature in the outer face of the greenhouse cover. The sensors to measure the cover temperature are located on the East side (two sensors) and the West side (two sensors) of the greenhouse, the rest of the sensors are located above the greenhouse (see upper left photo in Fig. 3.23). In the greenhouse the following variables are measured where the numbers in parentheses

**Figure 3.24**: View on the lateral (left) and roof (right) windows of the experimental greenhouse.

**Figure 3.25**: View on the lateral (left) and roof (right) windows of the experimental greenhouse.

indicate the number of used sensors for the corresponding variables: ambient temperature (2), relative humidity (2), ground temperature in a depth of 1 cm and 40 cm (1 for each depth), temperature of foil covering the ground (2), temperature of the inner greenhouse cover (4), temperature of plant leafs (4), photosynthetically active radiation (1), global radiation (1), carbon dioxide concentration (1) and wind speed (1). The sensors for the ambient temperature, relative humidity and carbon dioxide are located to 2.5 m above the ground, whereas the photosynthetically active radiation and global radiation are measured in a height of 3.5 m. The sensor of the wind speed is installed in a height of 2.8 m. The sensors for the cover temperature are collocated in the east and west sidewalls. Fig. 3.25 shows some of the sensors installed in the interior of the greenhouse. The data acquisition is carried out with an uniform sampling time of $t_s = 1$ min for all sensors. The data acquisition is carried out with a control and monitoring system implemented in the LabView environment and running on a personal computer. The different climate control strategies are implemented in Matlab, giving rise to a great flexibility in the design of control techniques.

The main purpose to use a greenhouse in agricultural production is the climate control in order to optimize the ambient conditions for crop growth and yield. Besides the temperature and the relative humidity in the greenhouse, many other factors as the radiation and the concentration of carbon dioxide influence directly or indirectly the crop growth. In mild climates as the one in the greenhouse location (South-East Spain), the greatest problem in greenhouse climate control is cooling. For economical reasons the used standard tool is the natural ventilation, manipulable by the aperture of the roof and lateral windows. During daytime, the ventilation allows a natural cooling of the greenhouse interior in order to avoid harmful temperatures for the crop. At night, the windows are used to ensure a greenhouse temperature above the exterior temperature. With the windows partially or completely closed during the night, the relative humidity in the greenhouse rises towards the morning hours. As a high humidity elevates the risk of plant diseases and parasites, a good ventilation has to be ensured during the morning hours. Hence, the principal control objective is the temperature regulation with a short interval of good ventilation during the morning in order to reduce the relative humidity in the greenhouse.

### 3.3.2 Identification of a Volterra series model

As already mentioned the main objective is the regulation of the greenhouse temperature in order to achieve suitable climate conditions for crop growth. In many cases, especially in areas with hot summers and mild winters, the temperature regulation is carried out by natural ventilation, which itself can be manipulated through the aperture of the lateral and roof windows of the greenhouse. Although detailed physical greenhouse models can be found in the literature [19, 57, 117], these models cannot be used as control-oriented models in predictive control strategies due to its high complexity. As a consequence, a prediction model relating the aperture of the windows and the main disturbances with the greenhouse temperature has to be developed.

Generally, greenhouse temperature modeling is a challenging task due to its nonlinear dynamic behavior with respect to changes in the aperture of the windows. Furthermore, the greenhouse temperature is strongly influenced by the external environmental conditions (e.g. solar radiation and outside temperature) which can be considered as measurable disturbances. Due to the possibility to consider nonlinear dynamics and to include disturbances, a second order Volterra series model has been chosen for the general model structure. From different studies [47, 45] it is known that the greenhouse temperature is mainly influenced by the aperture of the roof and lateral windows, the outside temperature, the outside wind speed, the soil surface temperature and the outside global radiation. As a consequence, the main variables considered for modeling

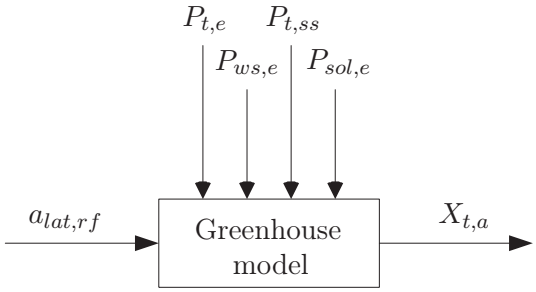


**Figure 3.26**: Block diagram of the considered model of the greenhouse with the model input (aperture of the roof and lateral windows $a_{lat,rf}$), disturbances (outside temperature $P_{t,e}$, outside wind speed $P_{ws,e}$, soil surface temperature $P_{t,ss}$, outside global solar radiation $P_{sol,e}$) and output (greenhouse temperature $X_{t,a}$).

purposes are (see Fig. 3.26)[1]:

- Output: greenhouse temperature $X_{t,a}$
- Input: aperture of the roof and lateral windows $a_{lat,rf}$
- Disturbances: outside temperature $P_{t,e}$, outside wind speed $P_{ws,e}$, soil surface temperature $P_{t,ss}$, outside global solar radiation $P_{sol,e}$

Defining the input as $u = a_{lat,rf}$, the output as $y = X_{t,a}$ and the disturbances as $w_1 = P_{t,e}$, $w_2 = P_{ws,e}$, $w_3 = P_{sol,e}$ and $w_4 = P_{t,ss}$ the second order Volterra series model (2.11) considering 1 input and 4 measurable disturbances can be written as:

$$\begin{aligned} y(k) &= h_0 + \sum_{i=1}^{N_1} h_1(i)u(k-i) + \sum_{i=1}^{N_2}\sum_{j=i}^{N_2} h_2(i,j)u(k-i)u_n(k-j) \\ &\quad + \sum_{n=1}^{m}\sum_{i=1}^{N_{1,n}} h_{1,n}(i)w_n(k-i) + \sum_{n=1}^{m}\sum_{i=1}^{N_{2,n}}\sum_{j=i}^{N_{2,n}} h_{2,n}(i,j)w_n(k-i)w_n(k-j) \end{aligned} \tag{3.23}$$

It has to be mentioned that during the identification process additional inputs were tested, e.g. the product $a_{lat,rf}P_{ws,e}$. However, no large improvements in the parameter identification stage were detected when using these additional inputs while the number of parameters did increase. Moreover, the inclusion of these modulated variables will complicate the use of MPC strategies as the products of system input and disturbances lead to a more complex optimization problem.

[1]The Leaf Area Index is another variable affecting the greenhouse dynamics. However, the dynamics of this variable is quite slower in comparison to the considered variables and, as a consequence, has not been included in the greenhouse model.

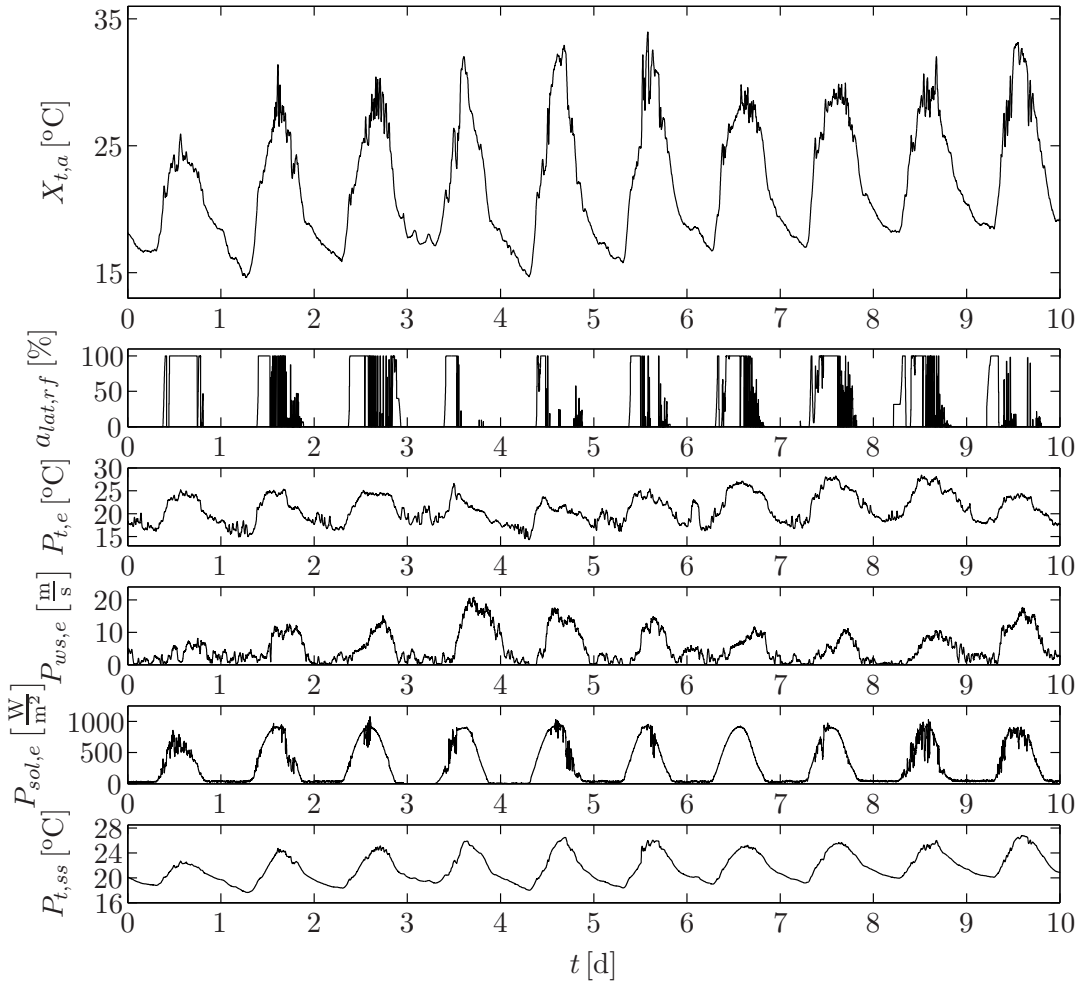


**Figure 3.27**: Data set used for the parameter identification with the output $X_{t,a}$ (greenhouse temperature), the input $a_{lat,rf}$ (aperture of the roof and lateral windows) and the disturbances $P_{t,e}$ (outside temperature), $P_{ws,e}$ (outside wind speed), $P_{t,ss}$ (soil surface temperature) and $P_{sol,e}$ (outside global solar radiation).

In order to identify the parameters of the Volterra model of the greenhouse temperature, experimental data of autumn (from August to February) and spring (from March to June) seasons from 2007 and 2008 have been used. Fig. 3.27 shows an interval of 10 days with the input, the output and the considered disturbances used in the identification. A second set of experimental data of a long season from September 2008 to June 2009 have been used for validation purposes (see Fig. 3.28 for a set of 10 days used in the model validation). It can be seen that the used sets for identification and validation have rich input signals, necessary for the identification of a second order Volterra series model. The sampling time for both sets was 1 minute and the data of the wind speed and the global solar radiation were filtered through a first order filter with a time constant of 5 minutes before using them for calibration purposes.

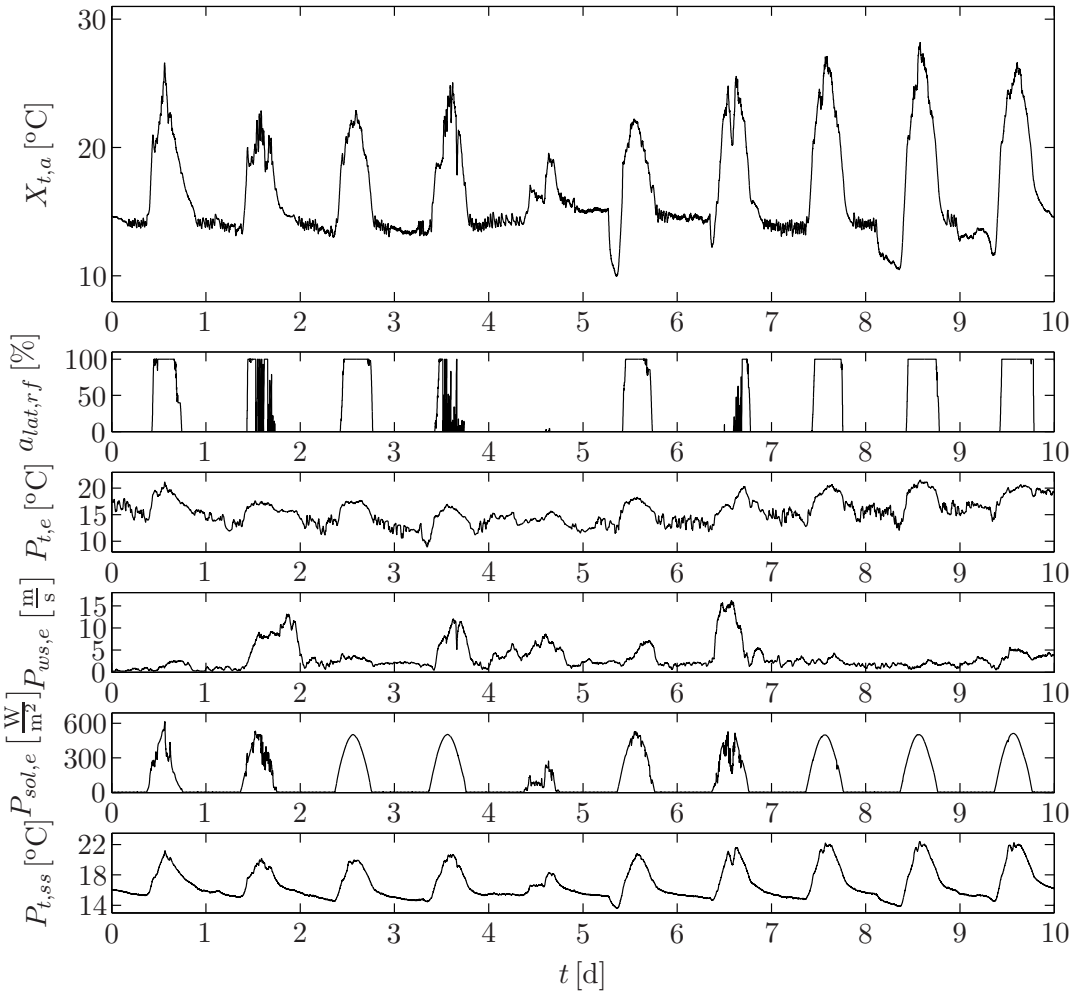


**Figure 3.28**: Data set used for the model validation with the output $X_{t,a}$ (greenhouse temperature), the input $a_{lat,rf}$ (aperture of the roof and lateral windows) and the disturbances $P_{t,e}$ (outside temperature), $P_{ws,e}$ (outside wind speed), $P_{t,ss}$ (soil surface temperature) and $P_{sol,e}$ (outside global solar radiation).

One of the initial problems in the model identification was to select the truncation orders for the input and the disturbances influencing the greenhouse inside temperature. Due to this reason, in a first step a least squares identification with long truncation orders for all considered variables was carried. As the greenhouse is a fading memory system the parameters tend to zero along the horizon and the truncation orders were reduced in the case of parameters being zero or nearly zero. The used sampling time for the model was $t_s = 1\,\text{min}$ and the finally chosen truncation orders for the input and the different disturbances are given in Tab. 3.4.

In a second step, as many parameters identified with the least squares method had noisy profiles, constraints in the parameters have been included using Sequential

| Variable | $1^{\text{st}}$ order | $2^{\text{nd}}$ order |
|---|---|---|
| $a_{lat,rf}$ | 20 | 10 |
| $P_{t,e}$ | 1 | 1 |
| $P_{ws,e}$ | 2 | 2 |
| $P_{sol,e}$ | 5 | 0 |
| $P_{t,ss}$ | 10 | 0 |

**Table 3.4**: Used truncation orders for the input and the different disturbances in the second order Volterra series model approximating the nonlinear dynamics of the considered greenhouse.

Quadratic Programming. The constraints were chosen on the basis of the results obtained in the identification with the least squares method. For the aperture of the roof and lateral ventilations the constraints

$$\begin{array}{rcll} h_1(i+1) & < & h_1(i) & \text{for } i = 1, \ldots, 3 \\ h_1(i+1) & \geq & h_1(i) & \text{for } i = 4, \ldots, 19 \\ h_1(i) & \leq & 0 & \text{for } i = 1, \ldots, 20 \end{array} \tag{3.24}$$

and

$$\begin{array}{rcll} h_2(i+1,i+1) & \leq & h_2(i,i) & \text{for } i = 1, \ldots, 9 \\ h_2(i,j+1) & \leq & h_2(i,j) & \text{for } i = 1, \ldots, 9,\ j = i, \ldots, 9 \\ h_2(i,j) & \geq & 0 & \text{for } i = 1, \ldots, 10,\ j = 1, \ldots, 10, \end{array} \tag{3.25}$$

were imposed in the linear term and nonlinear term parameters. For the solar radiation, the following linear term parameters were used:

$$\begin{array}{rcll} h_{1,3}(i+1) & \leq & h_{1,3}(i) & \text{for } i = 1, \ldots, 4 \\ h_{1,3}(i) & \geq & 0 & \text{for } i = 1, \ldots, 5 \end{array} \tag{3.26}$$

In an analogous way, the constraints

$$\begin{array}{rcll} h_{1,4}(i+1) & \leq & h_{1,4}(i) & \text{for } i = 1, \ldots, 9 \\ h_{1,4}(i) & \geq & 0 & \text{for } i = 1, \ldots, 10 \end{array} \tag{3.27}$$

were considered in the linear term parameters of the soil surface temperature.

With the constraints in the parameters a second order Volterra series model of the greenhouse temperature was identified. A comparison of the greenhouse temperature and the output of the identified model is given in Fig. 3.29, both for the identification process and the validation. As can be seen in the results, the model output shows a good match with the measured greenhouse temperature, both for the identification and the validation data set. The mean square error (3.6) for the identification and

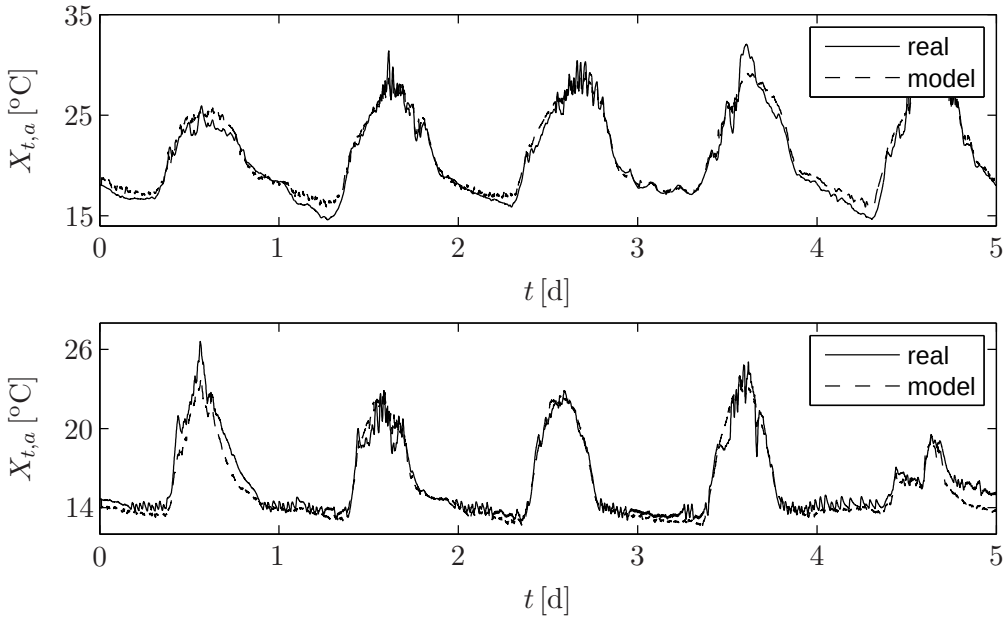


**Figure 3.29**: Identification (top) and validation results (bottom). Comparison of the experimental data (solid line) and the output of the identified second order Volterra series model (dashed line).

validation was $\varepsilon_{id} = 0.933$ and $\varepsilon_{val} = 0.988$, representing good values for a model of the given characteristics. The identified linear term parameters of the aperture of the lateral and roof ventilations $a_{lat,rf}$, outside global solar radiation $P_{sol,e}$ and soil surface temperature $P_{t,ss}$ are shown in Fig. 3.30, while the second order term parameters of the aperture of the lateral and roof ventilations $a_{lat,rf}$ are shown in Fig. 3.31. It can be seen that the linear and nonlinear term parameters identified under consideration of the constraints (3.24)-(3.27) show a coherent and reasonable shape.

## 3.4 Conclusions of the chapter

In this chapter three benchmark systems used in this document for the application of model-based predictive control strategies have been presented. The system dynamics of the different processes have been approximated by second order Volterra series models. For the parameter identification experimentally obtained input-output data have been used and the parameters have been estimated by means of the least squares method.

For the first system, a pilot plant emulating an exothermic chemical reaction, the identified Volterra series model relates the opening of a valve with the temperature in

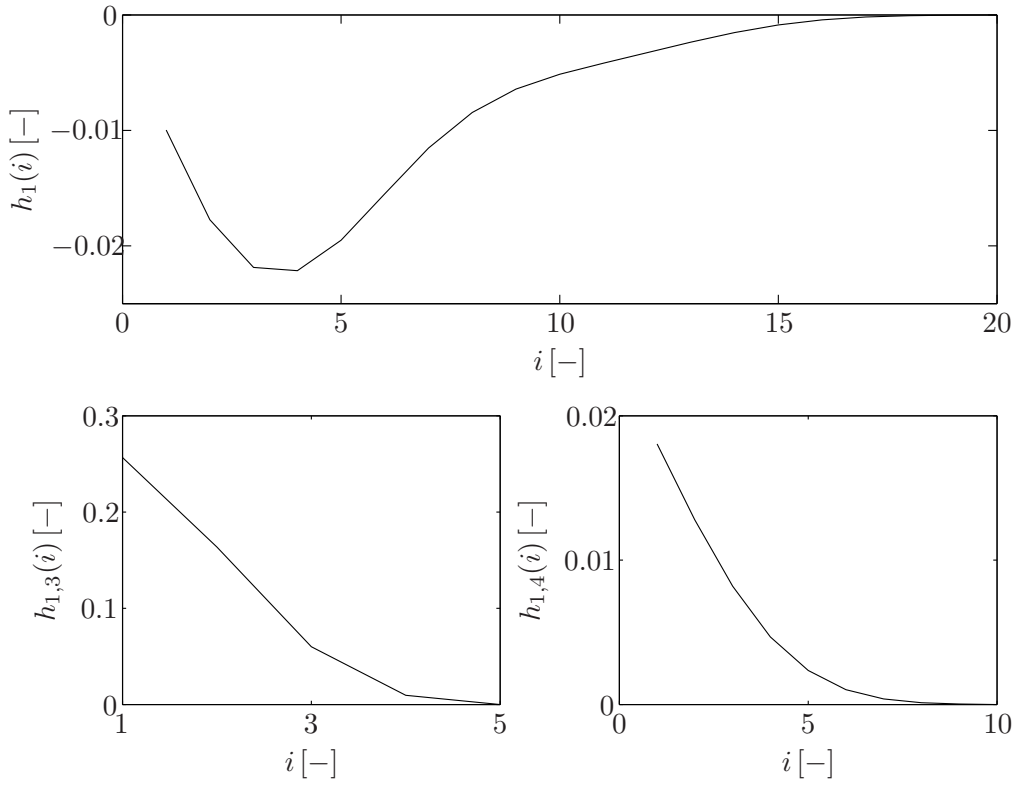


**Figure 3.30**: Identified linear term parameters for the aperture of the roof and lateral ventilations $a_{lat,rf}$ (top), solar radiation $P_{sol,e}$ (bottom left) and soil surface temperature $P_{t,ss}$ (bottom right).

the reactor. This system possesses rather slow dynamics and can be considered as a SISO system. The second system used for benchmark purposes is a commercial fuel cell, characterized by a very fast dynamic behavior. The oxygen excess ratio has been modeled by a second order Volterra series model considering the compressor motor voltage as input variable. Due to the strong nonlinear influence of the current demanded by an external load, the model parameters are scaled by the load current. The identified model can be considered as a SISO model with a feedforward effect based on the scaling of the parameters. The third benchmark system presented in this chapter is an experimental greenhouse used for crop growth under suitable environmental conditions. Although the greenhouse can be considered as a system with only 1 input (windows aperture) several disturbances have a strong influence on the output (greenhouse inside temperature). Therefore the greenhouse temperature has been approximated by a second order Volterra series model considering 1 input and 4 measurable disturbances.

These systems cover a wide range of processes, with varying time scales (from milliseconds to minutes) and different nonlinearities. The consideration of disturbances and multiple inputs make them appropriate to validate the control strategies presented

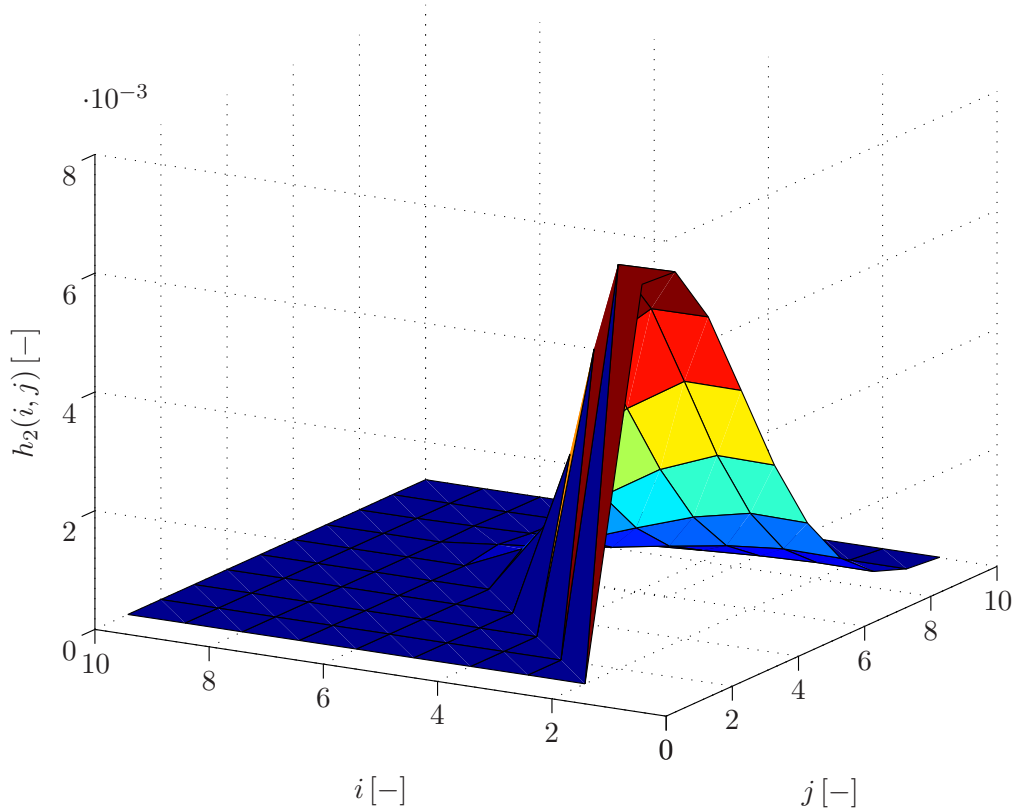


**Figure 3.31**: Identified second order term parameters for the aperture of the roof and lateral ventilations $a_{lat,rf}$.

in the following chapters.

# Chapter 4

# Iterative optimization algorithms

This chapter shows the idea of iterative optimization methods for NMPC strategies based on second order Volterra series models. The unconstrained optimization method, originally published in [37, 77, 38], can be considered in this document as a general starting point for the development of new MPC strategies based on Volterra series models.

In the first place, the general idea of iterative optimization for models which can be separated in a linear and a nonlinear part will be explained. Based on the inversion of the linear part of the model, the unconstrained iterative control strategy [37, 77, 38] minimizes a quadratic cost function based on a second order Volterra series model. Afterwards, the original iterative control strategy will be extended to consider linear input constraints and a weighting of the control effort during the optimization process. Finally some changes will be introduced in the constrained iterative optimization to guarantee stability. The proposed control strategies based on the constrained optimization and the constrained optimization with guaranteed stability will be applied to the benchmark systems and the behavior of the control strategies will be illustrated by experimental results.

## 4.1 General idea of iterative optimization

One of the main problems when dealing with nonlinear models in MPC is the optimization of the cost function in order to compute a new input signal. In linear unconstrained MPC strategies, the inversion of the model allows to determine the optimal new input

sequence, e.g. in Generalized Predictive Control (GPC) [28] and Dynamic Matrix Control (DMC) [31]. Unfortunately, in NMPC the simple inversion of the model is in the majority of the cases not possible and different approaches for the optimization have to be used. The general idea of the iterative optimization is based on the separability of the linear and nonlinear terms of Volterra series models.

Consider the model $\hat{\mathbf{P}}$ approximating a system denoted $\mathbf{P}$. Hence, the predicted output can generally be written as:

$$\hat{y} = \hat{\mathbf{P}}[u] \tag{4.1}$$

where $\hat{\mathbf{P}}$ is an operator mapping the input signal $u$ to the predicted output $\hat{y}$. If the model $\hat{\mathbf{P}}$ is not a perfect approximation of the system $\mathbf{P}$, the estimation error can be defined as the difference between the system output $y$ and the predicted output $\hat{y}$:

$$d = y - \hat{y} \tag{4.2}$$

As a direct consequence, the predicted output $\hat{y}$ can be written as system output $y$ minus the estimation error $d$:

$$\hat{y} = y - d \tag{4.3}$$

Now, substituting in (4.1) the predicted output $\hat{y}$ by (4.3), the system output can be expressed as:

$$y = \hat{\mathbf{P}}[u] + d \tag{4.4}$$

Then, substituting the system output $y$ by the desired reference $r$ and inverting the operator $\hat{\mathbf{P}}$, the necessary input $u$ for the given reference $r$ under consideration of the estimation error $d$ is defined by:

$$u = \hat{\mathbf{P}}^{-1}[r - d] \tag{4.5}$$

The input $u$ (4.5) can be used to control the system represented by the operator $\mathbf{P}$ as shown in Fig. 4.1. Note that the inverse $\hat{\mathbf{P}}^{-1}$ may not exist, especially for badly conditioned or nonlinear models. Furthermore, the direct inversion of the model (4.5) does not allow to consider constraints or a weighting of the control effort in the computation of the input.

Consider now a nonlinear model represented by the operator $\hat{\mathbf{P}}$ and separable in a linear operator $\hat{\mathbf{P}}_l$ and a nonlinear operator $\hat{\mathbf{P}}_{nl}$ as in the case of a second order Volterra series model. Hence, the nonlinear operator $\hat{\mathbf{P}}$ can be expressed as:

$$\hat{\mathbf{P}} = \hat{\mathbf{P}}_l + \hat{\mathbf{P}}_{nl} \tag{4.6}$$

Analogously to (4.1), the output prediction for the model based on the nonlinear operator $\hat{\mathbf{P}}$ can be written with (4.6) as:

$$\hat{y} = \left(\hat{\mathbf{P}}_l + \hat{\mathbf{P}}_{nl}\right)[u] \tag{4.7}$$

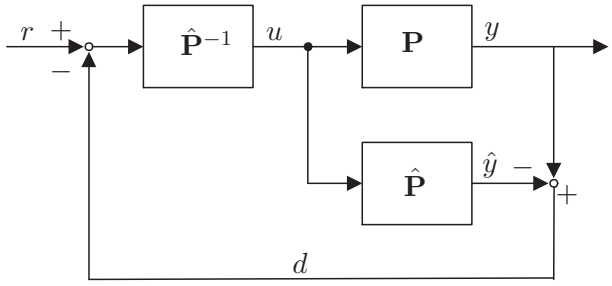


**Figure 4.1**: Schematic diagram of the control strategy based on the inversion of the operator $\hat{\mathbf{P}}$ defined in (4.5).

Using the definition of the predicted output based on the system output and the estimation error as given in (4.3), the output prediction (4.7) can be expressed as:

$$y = \left(\hat{\mathbf{P}}_l + \hat{\mathbf{P}}_{nl}\right)[u] + d \tag{4.8}$$

As already mentioned, the inverse of the operator $\hat{\mathbf{P}} = \hat{\mathbf{P}}_l + \hat{\mathbf{P}}_{nl}$ may not exist. Therefore, under the assumption that the linear operator $\hat{\mathbf{P}}_l$ can be inverted, the nonlinear operator $\hat{\mathbf{P}}$ is rewritten as:

$$\hat{\mathbf{P}} = \hat{\mathbf{P}}_l \left(I + \hat{\mathbf{P}}_l^{-1}\hat{\mathbf{P}}_{nl}\right) \tag{4.9}$$

where $I$ denotes the identity matrix. Then, the inversion of the operator $\hat{\mathbf{P}}$ (4.9) can be carried out by:

$$\hat{\mathbf{P}}^{-1} = \left(I + \hat{\mathbf{P}}_l^{-1}\hat{\mathbf{P}}_{nl}\right)^{-1} \hat{\mathbf{P}}_l^{-1} \tag{4.10}$$

where the nonlinear operator $\hat{\mathbf{P}}_{nl}$ is not inverted explicitly. With $\hat{\mathbf{P}}^{-1}$ having the characteristic form of a closed-loop system, the inversion of $\hat{\mathbf{P}}$ can be represented as shown in Fig. 4.2. Now, using the inversion (4.10) in (4.7) and substituting the system output $y$ by the desired reference $r$, the necessary output for a given reference can be defined as:

$$u = \left(I + \hat{\mathbf{P}}_l^{-1}\hat{\mathbf{P}}_{nl}\right)^{-1} \hat{\mathbf{P}}_l^{-1}[r - d] \tag{4.11}$$

The inversion of the nonlinear operator $\hat{\mathbf{P}}$ can be easily implemented in the form shown in Fig. 4.2. Note that the computation of the input $u$ is based on a closed-loop concept and, as a consequence, represents an iterative optimization approach. With the original nonlinear system, given by the operator $\mathbf{P}$, the entire control strategy can be represented by Fig. 4.3. It can be seen that in the first place the input value $\xi = r - d$ for the controller is computed. With this value the controller calculates the control input $u$ by inversion of the mathematical model. The control input is applied to the system and to the mathematical model. The output $y$ of the system is measured

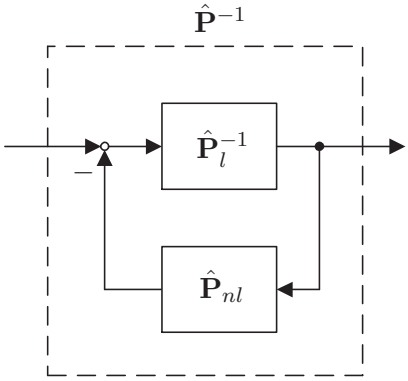


**Figure 4.2**: Graphical representation of the inversion of the nonlinear operator $\hat{\mathbf{P}}$ as defined in (4.10).

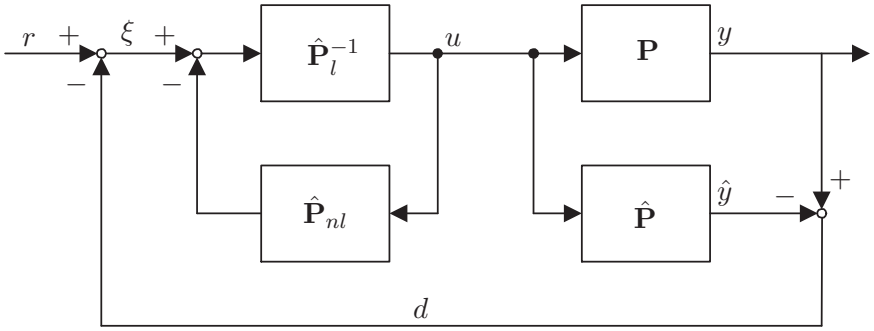


**Figure 4.3**: Schematic diagram of a possible control strategy based on the inversion of the nonlinear operator $\hat{\mathbf{P}}$ (4.10).

and the model output $\hat{y}$ can be calculated. The difference $d$ of these two outputs, the measured ($y$) and the predicted ($\hat{y}$), is used to calculate the input value $\xi = r - d$ for the controller.

Note that the theoretical development of the control strategy presented in this section considers only a control and prediction horizon of 1. For longer horizons the development is carried out in an analogous way. The resulting controller calculates the input sequence $\boldsymbol{u}$ of future input values from the vector $\boldsymbol{\xi}$ defined by

$$\boldsymbol{\xi} = \boldsymbol{r} - \boldsymbol{d} \tag{4.12}$$

where $\boldsymbol{r} = [r, \ldots, r]^T$ with $\boldsymbol{r} \in \mathbb{R}^N$ denotes the reference trajectory, $\boldsymbol{d} = [d, \ldots, d]^T$ with $\boldsymbol{d} \in \mathbb{R}^N$ represents the vector of the estimation error and $N$ is the prediction horizon. From this input sequence only the first value will be applied to the system and the mathematical model. Now, with the measured ($y$) and predicted ($\hat{y}$) output, the one step ahead prediction error $d$ can be calculated. Finally, the vector $\boldsymbol{\xi}$ for

the controller is calculated with the vector $\boldsymbol{d}$ and the vector of the desired reference trajectory $\boldsymbol{r}$.

The following sections explain in a detailed manner the optimization based on the shown iterative approach. In a first step, the original algorithm [77, 38], based on an unconstrained optimization without penalization of the control effort, will be presented. The original approach will be modified to consider linear input constraints and a weighting of the control effort. Afterwards, some changes will be introduced in the control strategy based on the iterative constrained optimization approach in order to guarantee stability.

## 4.2 Unconstrained optimization

With the iterative optimization approach from Section 4.1 the nonlinear prediction model (2.43)-(2.44) based on a second order Volterra series can be used in MPC where the input sequence is computed by minimizing a cost function. Without considering constraints, the optimization problem (compare (2.53) and (2.54)) is defined as:

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} J(\boldsymbol{u}) \tag{4.13}$$

Without penalizing the control effort, the quadratic performance based on a second order Volterra series model (2.43)-(2.44) can be expressed as[1]:

$$\begin{aligned} J(\boldsymbol{u}) &= (\hat{\boldsymbol{y}} - \boldsymbol{r})^T (\hat{\boldsymbol{y}} - \boldsymbol{r}) \\ &= (G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r})^T (G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r}) \end{aligned} \tag{4.14}$$

The optimal input sequence can be obtained by minimizing the cost function (4.14). It is clear from the nonlinearity of the used prediction model that the inversion of the entire model, a common technique in linear MPC, is not possible. Therefore the computation of the input sequence is carried out by means of an iterative approach, similar to the one presented in Section 4.1. This approach exploits the fact that a Volterra series model can be separated in its linear and nonlinear parts. It can be seen from Fig. 4.3 that with the iterative optimization approach a new input sequence is calculated by inversion of the linear operator. As the optimization is carried out in a closed-loop like manner, the nonlinear part is computed using the input sequence calculated by inversion of the linear part of the model. Denoting the new input sequence $\boldsymbol{u}^{(i)}$, the nonlinear term has been computed with the input sequence $\boldsymbol{u}^{(i-1)}$ *calculated in*

[1] In order to give a complete vision of the used technique and to increase the readability of the current section, the definition of the performance index (2.48) is repeated in this section.

*the previous iteration*. Then, the cost function (4.14) to be minimized can be rewritten as:

$$J(\boldsymbol{u}^{(i)}) = \left(G\boldsymbol{u}^{(i)} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right)^T \left(G\boldsymbol{u}^{(i)} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right) \tag{4.15}$$

Now, the performance index is a convex quadratic function in $\boldsymbol{u}^{(i)}$ and its minimum can be easily obtained by means of its derivative in $\boldsymbol{u}^{(i)}$. With the derivative defined as

$$\frac{\partial J(\boldsymbol{u}^{(i)})}{\partial \boldsymbol{u}^{(i)}} = 2G^T G\boldsymbol{u}^{(i)} + 2G^T \left(\boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right) \tag{4.16}$$

the cost function is minimized when the condition

$$\frac{\partial J(\boldsymbol{u}^{(i)})}{\partial \boldsymbol{u}^{(i)}} = 0 \tag{4.17}$$

is satisfied. Resolving (4.16) the input sequence for the $i$-th iteration can be calculated easily as:

$$\boldsymbol{u}^{(i)} = \left(G^T G\right)^{-1} G^T \left(\boldsymbol{r} - \boldsymbol{c} - \mathbf{f}(\boldsymbol{u}^{(i-1)})\right) \tag{4.18}$$

With the input sequence $\boldsymbol{u}^{(i)}$ obtained by solving (4.18) the nonlinear part of the model $\mathbf{f}(\boldsymbol{u}^{(i)})$ can be computed. Based on this nonlinear term, the new input sequence $\boldsymbol{u}^{(i+1)}$ can be calculated. Finally, (4.18) is repetitively solved until some convergence criterion is met, e.g. the difference between the computed input sequences $\boldsymbol{u}^{(i)}$ and $\boldsymbol{u}^{(i-1)}$ falls below a predefined level.

### 4.2.1 Unconstrained control law

The unconstrained iterative optimization (4.18) of a quadratic cost function in combination with a prediction model based on a second order Volterra series model can be implemented in the following way [77, 38]:

- Step 1: Define $i = 0$ and $\boldsymbol{u}^{(0)} = \boldsymbol{u}_0$
- Step 2: Set $i = i + 1$ and calculate $\mathbf{f}(\boldsymbol{u}^{(i-1)})$
- Step 3: Compute $\boldsymbol{u}^{(i)}$ solving the unconstrained optimization problem

$$\boldsymbol{u}^{(i)} = \left(G^T G\right)^{-1} G^T \left(\boldsymbol{r} - \boldsymbol{c} - \mathbf{f}(\boldsymbol{u}^{(i-1)})\right)$$

- Step 4: Check the final accuracy

$$\sum_{j=0}^{N_u-1} \left|u^{(i)}(k+j|k) - u^{(i-1)}(k+j|k)\right| < \delta$$

- Step 5: If the previous condition has been satisfied, define the control input under consideration of saturation in the actuator of the system as:

$$u(k|k) = \begin{cases} u_{min} & \text{if} \quad u^{(i)}(k|k) < u_{min} \\ u^{(i)}(k|k) & \text{elsewise} \\ u_{max} & \text{if} \quad u^{(i)}(k|k) > u_{max} \end{cases} \tag{4.19}$$

  define $u(k+j|k) = u^{(i)}(k+j|k)$ for $j = 1, \ldots, N_u - 1$ and apply $u(k|k)$ to the system. If the previous condition has not been satisfied, return to step 2.

It can be seen in the algorithm that the term $\mathbf{f}(\mathbf{u}^{(i-1)})$ has a constant value during the optimization in step 3. In the case of not satisfying the final accuracy condition the term $\mathbf{f}(\mathbf{u}^{(i-1)})$ is recalculated in step 2 with the control input sequence computed in step 3 of the previous iteration. After calculating a new input sequence the satisfaction of the final accuracy condition is checked in step 4. This condition is satisfied when the sum of the absolute values of the difference between the current input sequence and the input sequence from the previous iteration falls below some constant value $\delta$. A flowchart of the described iterative algorithm to solve the unconstrained optimization problem is given in Fig. 4.4. It has to be mentioned that the presented iterative optimization algorithm cannot guarantee the convergence of the solution if the initial optimization problem (4.13) is non-convex. Nevertheless, with a suitable filtering of the desired reference, convexity can be achieved in the neighborhood of the current operating point. For details on the convergence of the presented algorithm see [39].

The initial vector $\boldsymbol{u}_0$ at $k$ can be chosen by shifting the input sequence calculated in the previous sampling period $k-1$ by one element. Therefore, the optimized input sequence is stored at the end of the optimization routine to be available for its use as an initial input sequence in the next sampling period. Using the shifted input sequence, the initial input sequence in the $k$-th sampling period is defined as:

$$\boldsymbol{u}_0 = \begin{bmatrix} u(k|k-1) \\ u(k+1|k-1) \\ \vdots \\ u(k+N_u-3|k-1) \\ u(k+N_u-2|k-1) \\ u(k+N_u-2|k-1) \end{bmatrix} \tag{4.20}$$

Note that the presented algorithm considers neither a weighting of the control effort nor constraints. Therefore the control strategy will show an aggressive behavior without the possibility to smoothen the system's reaction. Just like in other MPC strategies

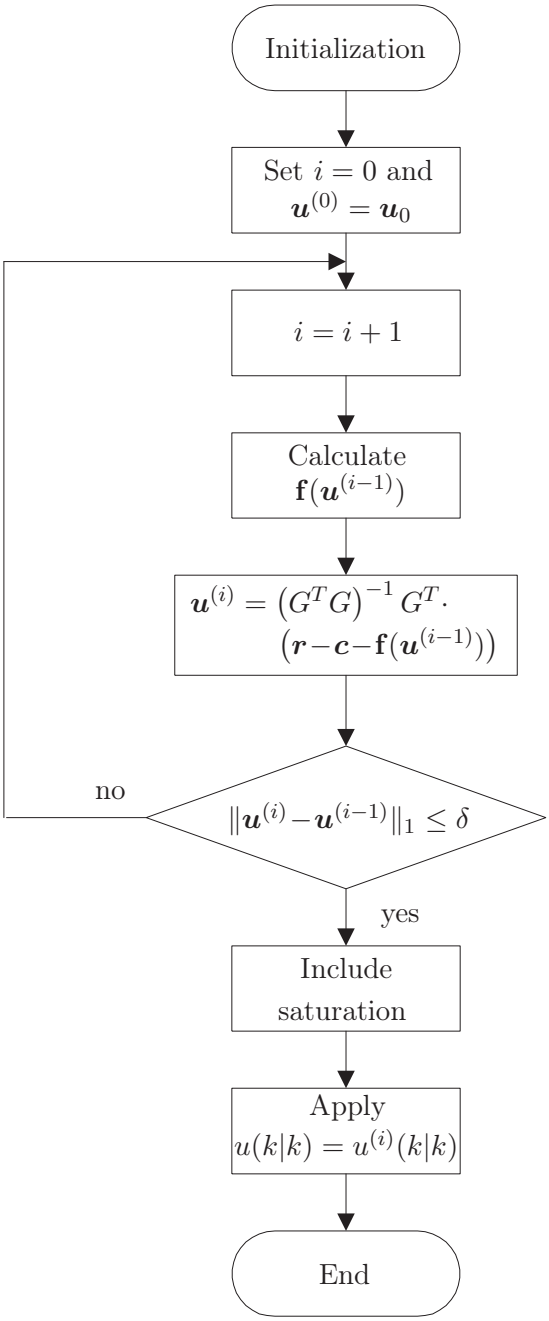


**Figure 4.4**: Flowchart of the iterative algorithm to solve the unconstrained optimization problem.

[23], the use of a saturation for the input action results in a suboptimal solution to the optimization problem. To avoid the mentioned problems the presented control strategy will be extended in the following section in order to include constraints and a weighting function for the input sequence.

## 4.3 Constrained optimization

The inclusion of constraints and a weighting of the control effort in the minimization of the cost function requires some changes of the control strategy presented in Section 4.2. For the consideration of constraints the optimization routine has to be changed, using quadratic programming (QP) instead of the explicit solution.

The general optimization problem considering linear constraints is defined by[2]:

$$\begin{aligned} \boldsymbol{u}^* &= \arg \min_{\boldsymbol{u}} J(\boldsymbol{u}) \\ &\quad \text{s.t. } L_c\boldsymbol{u} \leq \boldsymbol{b}_c \end{aligned} \tag{4.21}$$

and the quadratic cost function with a weighting of the control effort in combination with a prediction model based on a second order Volterra series model is given by:

$$\begin{aligned} J(\boldsymbol{u}) &= (G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r})^T (G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r}) + \\ &\quad \lambda \boldsymbol{u}^T L_u^T L_u \boldsymbol{u}^T - 2\lambda \boldsymbol{u}_l^T L_u \boldsymbol{u} + \lambda \boldsymbol{u}_l^T \boldsymbol{u}_l \end{aligned} \tag{4.22}$$

with $\boldsymbol{u}_l \in R^{N_u}$ being a column vector where the first element corresponds to $u(k-1)$ and the remaining elements are zero, compare (2.50).

With the quadratic cost function considering a weighting of the control effort (4.22) the optimal input sequence is defined as the minimizer of the optimization problem (4.21). Analogously to the unconstrained case in Section 4.2, the used nonlinear prediction model inhibits to find an explicit solution for the constrained optimization problem. Nevertheless, with the possibility to separate the linear and the nonlinear part of the prediction model the optimization can be carried out in an iterative manner based on the general idea of iterative optimization (see Section 4.1). Therefore, the cost function (4.22) will be modified so that the linear terms depend on the input sequence $\boldsymbol{u}^{(i)}$ to be minimized and the nonlinear terms are considered as constants depending on the input sequence $\boldsymbol{u}^{(i-1)}$ calculated in the previous iteration:

$$\begin{aligned} J(\boldsymbol{u}^{(i)}) &= \left(G\boldsymbol{u}^{(i)} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right)^T \left(G\boldsymbol{u}^{(i)} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right) + \\ &\quad \lambda \boldsymbol{u}^{(i)T} L_u^T L_u \boldsymbol{u}^{(i)T} - 2\lambda \boldsymbol{u}_l^T L_u \boldsymbol{u}^{(i)} + \lambda \boldsymbol{u}_l^T \boldsymbol{u}_l \end{aligned} \tag{4.23}$$

[2]In order to give a complete vision of the used technique and to increase the readability of the current section, the definitions of the optimization problem with linear constraints (2.54) and the cost function considering a weighting of the control effort (2.52) are repeated in this section.

Note that the modified performance index (4.23) depends quadratically on $\boldsymbol{u}^{(i)}$ and, as a consequence, represents a convex function. With (4.23) the initial constrained optimization problem (4.21) can be expressed as:

$$\begin{aligned} \boldsymbol{u}^{(i)} &= \arg \min_{\boldsymbol{u}^{(i)}} J(\boldsymbol{u}^{(i)}) \\ &\quad \text{s.t. } L_c \boldsymbol{u}^{(i)} \leq \boldsymbol{b}_c \end{aligned} \tag{4.24}$$

With the performance index being a convex and quadratic function in the input sequence $\boldsymbol{u}^{(i)}$ to be calculated, the constrained optimization problem (4.24) can be solved by means of quadratic programming (QP). Separating the quadratic and linear terms as well as the constants, the cost function (4.23) can be rewritten as:

$$J(\boldsymbol{u}^{(i)}) = \frac{1}{2} \boldsymbol{u}^{(i)T} S \, \boldsymbol{u}^{(i)} + \boldsymbol{p}^{(i)T} \boldsymbol{u}^{(i)} + m^{(i)} \tag{4.25}$$

with

$$\begin{aligned} S &= G^T G + \lambda L_u^T L_u \\ \boldsymbol{p}^{(i)T} &= \left(\boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)} - \boldsymbol{r})\right)^T G - \lambda \boldsymbol{u}_l^T L_u \\ m^{(i)} &= \left(\boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right)^T \left(\boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right) + \lambda \boldsymbol{u}_l^T \boldsymbol{u}_l \end{aligned} \tag{4.26}$$

Now, the input sequence $\boldsymbol{u}^{(i)}$ can be computed by solving the convex quadratic optimization problem (4.24) based on the quadratic performance index (4.25) by means of quadratic programming. With the new input sequence, the nonlinear part $\mathbf{f}(\boldsymbol{u}^{(i)})$ can be recalculated and the input sequence $\boldsymbol{u}^{(i+1)}$ can be determined. This procedure is repeated until some convergence criterion is fulfilled, e.g. the difference between the computed input sequence $\boldsymbol{u}^{(i)}$ and the previous input sequence $\boldsymbol{u}^{(i-1)}$ falls below a certain limit. Finally, the initial constrained optimization problem (4.21) can be solved using the presented iterative approach. With the performance index in the form of (4.25)-(4.26) the minimization can be carried out by quadratic programming.

### 4.3.1 Constrained control law

The constrained minimization problem (4.24) can be solved using a modification of the iterative algorithm for the unconstrained problem (see Section 4.2.1) [77, 38]. Analogously to the unconstrained case, the constrained optimization considering a weighting of the control effort can be implemented in the following way [47]:

- Step 1: Define $i = 0$ and $\boldsymbol{u}^{(0)} = \boldsymbol{u}_0$
- Step 2: Set $i = i + 1$ and calculate $\mathbf{f}(\boldsymbol{u}^{(i-1)})$

- Step 3: Compute $\boldsymbol{u}^{(i)}$ solving the constrained optimization problem (4.24) by minimizing the cost function

$$J(\boldsymbol{u}^{(i)}) = \frac{1}{2}\boldsymbol{u}^{(i)T}\, S\, \boldsymbol{u}^{(i)} + \boldsymbol{p}^{(i)T}\, \boldsymbol{u}^{(i)} + m^{(i)}$$

  with quadratic programming and the parameters $S$, $\boldsymbol{p}^{(i)T}$ and $m^{(i)}$ (4.26).

- Step 4: Check the final accuracy

$$\sum_{j=0}^{N_u-1} \left| u^{(i)}(k+j|k) - u^{(i-1)}(k+j|k) \right| < \delta$$

- Step 5: If the previous condition has been satisfied, define the control signal $u(k|k) = u^{(i)}(k|k)$ and apply $u(k|k)$ to the system. If the previous condition has not been satisfied, return to step 2.

Analogously to the unconstrained case in Section 4.2.1, the nonlinear term $\mathbf{f}(\boldsymbol{u}^{(i-1)})$ is held constant during the optimization in Step 3. If the final accuracy condition is not satisfied, the term $\mathbf{f}(\boldsymbol{u}^{(i)})$ is calculated with the input sequence $\boldsymbol{u}^{(i)}$ before computing a new input sequence $\boldsymbol{u}^{(i+1)}$. For the proposed constrained optimization a flowchart is given in Fig. 4.5. Comparable to the case of unconstrained optimization, the iterative algorithm for constrained optimization cannot guarantee convergence if the initial optimization problems (4.21) is non-convex. Nevertheless, a non-convex iterative optimization problem can be transformed to a convex one by a suitable choice of the weighting factor $\lambda$ [35]. Furthermore, convexity can be achieved by using a filter for the desired reference. For details on the convergence of the iterative optimization algorithm the reader is referred to [39]. The initial vector for the input sequence $\boldsymbol{u}_0$ can be defined by shifting the sequence of the previous sampling period by one:

$$\boldsymbol{u}_0 = \begin{bmatrix} u(k|k-1) \\ u(k+1|k-1) \\ \vdots \\ u(k+N_u-3|k-1) \\ u(k+N_u-2|k-1) \\ u(k+N_u-2|k-1) \end{bmatrix} \tag{4.27}$$

Note that the algorithm for the constrained optimization problem without considering constraints and a weighting of $\lambda = 0$ gives the same results as the algorithm for the unconstrained case. This means that the proposed constrained optimization algorithm includes the algorithm from Section 4.2.1. Furthermore it has to be mentioned that

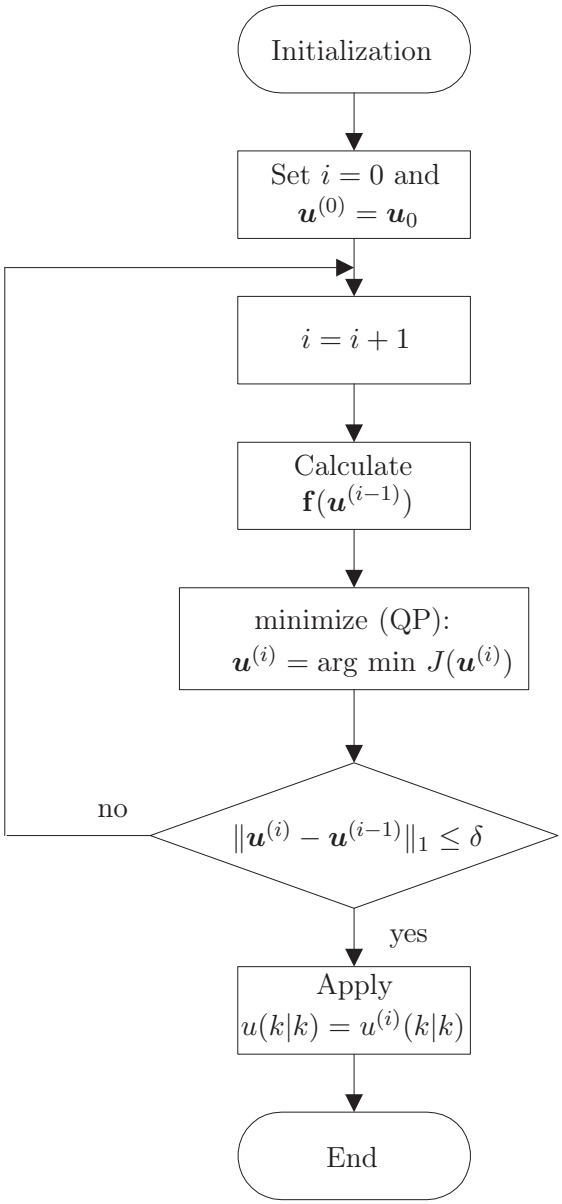


**Figure 4.5**: Flowchart of the iterative algorithm to solve the constrained optimization problem.

output constraints have not been considered in the proposed iterative algorithm. The use of output constraints in combination with the nonlinear prediction model (2.43)-(2.44) leads to the following optimization problem:

$$\begin{aligned} \boldsymbol{u}^{(i)} \quad = \arg \quad & \min_{\boldsymbol{u}^{(i)}} \; J(\boldsymbol{u}^{(\boldsymbol{i})}) \\ & \text{s.t. } \mathbf{q}(\boldsymbol{u}^{(i)}) \leq \boldsymbol{b}_c \end{aligned}$$

It is clear that the use of output constraints for an optimization problem based on a nonlinear model leads to nonlinear constraints which cannot be solved by quadratic programming. Therefore, the optimization under consideration of output constraints has to be carried out by nonlinear programming (NLP) or sequential quadratic programming (SQP). Hence, nonlinear constraints can be considered substituting in Step 3 of the presented optimization algorithm the quadratic programming by nonlinear programming. Nevertheless, the use of nonlinear programming will considerably augment the necessary time to calculate a new input sequence and therefore eliminates one of the biggest advantages of the iterative optimization, the low computational complexity and fast computation of a new input sequence. Instead of including the nonlinear programming in the proposed iterative optimization approach it is recommendable to solve the initial optimization problem (4.21) directly by nonlinear programming.

## 4.4 Constrained optimization with guaranteed stability

In this section the cost function and the constrained optimization problem presented in Section 4.3 will be modified to guarantee stability. Based on a feasible and easy determinable solution for the optimization problem, convergence for the modified MPC strategy can be shown and, as a consequence, input-to-state stability (ISS) [54, 67] can be proven. For the stability proof, the optimization problem has to be rewritten in state-space representation.

The general optimization to guarantee input-to-state stability is defined by:

$$\begin{aligned} \boldsymbol{u}^* \quad = \arg \quad & \min_{\boldsymbol{u}} \; J(\boldsymbol{u}) \\ & \text{s.t. } L_c \boldsymbol{u} \leq \boldsymbol{b}_c \\ & \quad u(k+N_u-1|k) = u_r(k) \\ & \quad N \geq N_t + N_u \end{aligned} \tag{4.28}$$

where the last element $u(k+N_u-1|k)$ of the input sequence has to correspond to the necessary steady-state input $u_r(k)$ for a given reference $r(k)$ (the exact definition of

$u_r(k)$ will be given later in Section 4.4.1.1). Furthermore, the prediction horizon $N$ has to be greater or equal to the sum of the truncation order $N_t$ and the control horizon $N_u$. The quadratic cost function considering a prediction model based on a second order Volterra series model used in the optimization problem (4.28) is given by

$$\begin{aligned} J(\boldsymbol{u}) &= (G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r})^T (G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) - \boldsymbol{r}) + \\ &\quad \lambda (\boldsymbol{u} - \boldsymbol{u}_r)^T (\boldsymbol{u} - \boldsymbol{u}_r) \end{aligned} \tag{4.29}$$

where the vector $\boldsymbol{u}_r \in \mathbb{R}^N$ contains the steady-state input $u_r$:

$$\boldsymbol{u}_r = [u_r(k), \quad u_r(k), \quad \ldots, \quad u_r(k)]^T \tag{4.30}$$

It is clear that the quadratic cost function (4.29) considering a second order Volterra series model results in a possibly non-convex function with several minima. In order to avoid the use of a nonlinear programming technique to solve the optimization problem (4.28), the same iterative optimization approach as in Section 4.3 will be used. Therefore and analogously to the unconstrained (see Section 4.2) and the constrained optimization (see Section 4.3), the cost function (4.29) will be modified so that the nonlinear term $\mathbf{f}(\boldsymbol{u})$ depends on the input sequence $\boldsymbol{u}^{(i-1)}$ calculated in the previous iteration whereas the remaining terms depend on the input sequence $\boldsymbol{u}^{(i)}$ to be computed in the current iteration:

$$\begin{aligned} J(\boldsymbol{u}^{(i)}) &= \left(G\boldsymbol{u}^{(i)} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right)^T \left(G\boldsymbol{u}^{(i)} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right) + \\ &\quad \lambda \left(\boldsymbol{u}^{(i)} - \boldsymbol{u}_r\right)^T \left(\boldsymbol{u}^{(i)} - \boldsymbol{u}_r\right) \end{aligned} \tag{4.31}$$

Then, the cost (4.31) is a quadratic and convex function in $\boldsymbol{u}^{(i)}$ and can be solved easily by quadratic programming. Based on the modified cost function (4.31) the initial constrained optimization problem with guaranteed stability (4.28) can be rewritten as:

$$\begin{aligned} \boldsymbol{u}^{(i)} = \arg \min_{\boldsymbol{u}^{(i)}} \; & J(\boldsymbol{u}^{(i)}) \\ \text{s.t. } & L_c \boldsymbol{u}^{(i)} \leq \boldsymbol{b}_c \\ & u^{(i)}(k + N_u - 1|k) = u_r(k) \\ & N \geq N_t + N_u \end{aligned} \tag{4.32}$$

Now, by separation of the quadratic and linear terms as well as the constants, the convex and quadratic cost function (4.32) can be represented by:

$$J(\boldsymbol{u}^{(i)}) = \frac{1}{2} \boldsymbol{u}^{(i)T} S \, \boldsymbol{u}^{(i)} + \boldsymbol{p}^{(i)T} \boldsymbol{u}^{(i)} + m^{(i)} \tag{4.33}$$

with the parameters:

$$\begin{aligned} S &= G^T G + \lambda I \\ \boldsymbol{p}^{(i)T} &= \left(\boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right)^T G - \lambda \boldsymbol{u}_r^T \\ m^{(i)} &= \left(\boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right)^T \left(\boldsymbol{c} + \mathbf{f}(\boldsymbol{u}^{(i-1)}) - \boldsymbol{r}\right) + \lambda \boldsymbol{u}_r^T \boldsymbol{u}_r \end{aligned} \tag{4.34}$$

Finally, the input sequence $\boldsymbol{u}^{(i)}$ can be computed with quadratic programming solving the optimization problem (4.32) based on the convex performance index (4.33) with the parameters (4.34). Analogously to the unconstrained (Section 4.2) and constrained optimization (Section 4.3) a new input sequence $\boldsymbol{u}^{(i)}$ is calculated with the nonlinear term $\mathbf{f}(\boldsymbol{u}^{(i-1)})$ based on the input sequence from the previous iteration. This procedure is repeated until some convergence criterion is fulfilled, e.g. the difference between the computed input sequence $\boldsymbol{u}^{(i)}$ and the previous input sequence $\boldsymbol{u}^{(i-1)}$ falls below a certain limit. Finally, the initial constrained optimization problem with guaranteed stability (4.28) can be solved by quadratic programming and used as a MPC strategy based on a second order Volterra series model. The iterative optimization problem can be solved by the procedure presented in Section 4.3.1 using the parameters (4.34) and the equality constraint $u^{(i)}(k+N_u-1|k) = u_r(k)$ under consideration of the condition $N \geq N_t + N_u$

### 4.4.1 Robust stability

In this section the robust stability of the proposed MPC strategy based on a second order Volterra series is shown. In a first step, the optimization problem will be rewritten in state-space formulation. Afterwards, a feasible and easy determinable solution for the optimization problem will be given. Finally, based on the feasible solution, convergence for the proposed MPC strategies can be shown and, as a consequence, input-to-state stability (ISS) [54, 67] can be proven. The proof of asymptotic stability for the presented MPC strategies based on the nominal prediction model, i.e. without considering an estimation error, is given in the Appendix A.4. Note that the stability proof is given in a very general form, not only suitable for the iterative optimization but also for other optimization techniques, e.g. sequential quadratic programming.

#### 4.4.1.1 Optimization problem in state-space representation

For the stability proof of the proposed NMPC strategy based on the input sequence (2.61) and a second order Volterra series prediction model (2.61) under consideration of the estimation error, the following general optimization problem is considered:

$$\begin{aligned} \min_{\boldsymbol{u}(k)} \quad & J(\boldsymbol{x}(k), \boldsymbol{u}(k), d(k)) \\ \text{s.t. } & u(k+i|k) \in U, && i = 0, \ldots, N_u - 1 \\ & h(\boldsymbol{x}(k+i|k)) \in U, && i = N_u, \ldots, N-1 \\ & \boldsymbol{x}(k+i|k) \in X, && i = 0, \ldots, N-1 \\ & \boldsymbol{x}(k+N|k) \in \Omega(d(k)) \end{aligned} \tag{4.35}$$

where $J(\boldsymbol{x}(k), \boldsymbol{u}(k), d(k))$ denotes the cost function for a finite prediction horizon $N$ and a finite control horizon $N_u$. The prediction and control horizon have to satisfy the condition $N_u \leq N$. The initial state is denoted $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$ represents the sequence of $N_u$ future input values along the control horizon. The predictions of the states $\boldsymbol{x}(k+i|k)$ for $i = N_u + 1, \ldots, N$, i.e. after reaching the end of the control horizon, are calculated with a local control law $h(\boldsymbol{x}(k+i|k))$ for $i = N_u, \ldots, N-1$. The terminal set

$$\Omega(d(k)) = \{\boldsymbol{x} : \mathrm{f}(\boldsymbol{x}(k+N|k)) + d(k) = r(k)\} \tag{4.36}$$

guarantees that the last element $\boldsymbol{x}(k+N|k)$ of the predicted output meets the desired reference $r(k)$. The prediction of the successor state can be generally written as:

$$\begin{array}{rcll} \boldsymbol{x}(k+i+1|k) & = & \boldsymbol{\phi}(\boldsymbol{x}(k+i|k), u(k+i|k)), & i = 0, \ldots, N_u - 1 \\ \boldsymbol{x}(k+i+1|k) & = & \boldsymbol{\phi}(\boldsymbol{x}(k+i|k), h(\boldsymbol{x}(k+i|k))), & i = N_u, \ldots, N \end{array} \tag{4.37}$$

The sequence $\boldsymbol{u}(k)$ calculated at $k$ minimizing the cost function (4.35) is defined in a general manner as:

$$\boldsymbol{u}(k) = \; [u(k|k), \;\; u(k+1|k), \;\; \ldots, \;\; u(k+N_u-1|k)]^T \tag{4.38}$$

and the cost function $J(\cdot,\cdot,\cdot)$ for a prediction horizon $N$ and a control horizon $N_u$ can be written as:

$$\begin{array}{rcl} J(\boldsymbol{x}(k), \boldsymbol{u}(k), d(k)) & = & \displaystyle\sum_{i=0}^{N_u-1} L(\boldsymbol{x}(k+i|k), u(k+i|k), d(k)) + \\ & & \displaystyle\sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}(k+i|k), d(k)) \end{array} \tag{4.39}$$

where $L(\cdot,\cdot,\cdot)$ represents the stage cost along the control horizon and $L_h(\cdot,\cdot)$ denotes the stage cost based on the local control law[3]. The quadratic stage costs $L(\cdot,\cdot,\cdot)$ and $L_h(\cdot,\cdot)$ used in the cost function are defined in general as:

$$\begin{array}{rcl} L(\boldsymbol{x}(k+i|k), u(k+i|k), d(k)) & = & \|\mathrm{f}(\boldsymbol{x}(k+i|k)) + d(k) - r(k)\|_Q^2 + \\ & & \|u(k+i|k) - u_r(k)\|_R^2 \\ L_h(\boldsymbol{x}(k+i|k), d(k)) & = & \|\mathrm{f}(\boldsymbol{x}(k+i|k)) + d(k) - r(k)\|_Q^2 + \\ & & \|h(\boldsymbol{x}(k+i|k)) - u_r(k)\|_R^2 \end{array} \tag{4.40}$$

where $\mathrm{f} : \mathbb{R}^{N_t} \mapsto \mathbb{R}$ (2.57) is a nonlinear function which maps the state vector to the output (2.61) and $r(k)$ denotes the reference of the system output. For a given reference $r(k)$ the variable $u_r(k)$ represents the necessary steady-state input signal.

[3]For the sake of simplicity the notation $L_h(\boldsymbol{x}(k+i|k), d(k)) = L(\boldsymbol{x}(k+i|k), h(\boldsymbol{x}(k+i|k)), d(k))$ has be chosen.

Based on the output prediction (2.61) and the estimation error, defined by $d(k) = y(k) - y(k|k-1)$ and assumed to be constant over the entire prediction horizon, the output prediction considering the estimation error can be written as

$$y(k+i|k) = \tilde{y}(k+i|k) + d(k) \tag{4.41}$$

where the output of the nominal model is defined as:

$$\tilde{y}(k+i|k) = \mathrm{f}(\boldsymbol{x}(k+i|k)) \tag{4.42}$$

In order to prove stability the last element of the output prediction has to satisfy the condition $y(k+N|k) = r(k)$. Substituting in (4.41) the output prediction for $k+N$ made at $k$ by the desired reference, the predicted nominal output $\tilde{y}(k+N|k)$ can be written as:

$$\tilde{y}(k+N|k) = r(k) - d(k) \tag{4.43}$$

Now, assuming that the nominal output reaches steady state in $k+N$ or before, the necessary steady-state input[4] $u_r(k)$ is based on the predicted nominal output $\tilde{y}(k+N-1|k)$:

$$\begin{array}{rcl} u_r(k) & = & \chi^{-1}(\tilde{y}(k+N|k)) \\ & = & \chi^{-1}(r(k) - d(k)) \end{array} \tag{4.44}$$

where $\chi : \mathbb{R} \mapsto \mathbb{R}$ is the static nonlinear function which maps the steady-state input to the nominal steady-state output. The steady-state input (4.44) is used directly to define the local control law in the initial optimization problem (4.35):

$$h(\boldsymbol{x}(k+i|k)) = u_r(k), \quad i = N_u, \ldots, N-1 \tag{4.45}$$

#### 4.4.1.2 Feasibility of the solution

Consider the optimal sequence:

$$\boldsymbol{u}^*(k) = \; [u^*(k|k), \;\; u^*(k+1|k), \;\; \ldots, \;\; u^*(k+N_u-1|k)]^T \tag{4.46}$$

for the optimization problem (4.35) with the optimal cost $J^*(\boldsymbol{x}(k))$ and the optimal predicted states $\boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots, N$. Furthermore, consider the shifted solution $\boldsymbol{u}^f(k+1)$ for $k+1$ given by:

$$\boldsymbol{u}^f(k+1) = \; [u^f(k+1|k+1), \;\; u^f(k+2|k+1), \;\; \ldots, \;\; u^f(k+N_u|k+1)]^T \tag{4.47}$$

[4]Note that the steady-state input depends on the given reference and the known estimation error. For the sake of simplicity, the notation $u_r(k) = u_r(r(k), d(k))$ has been chosen. Furthermore it has to be mentioned that the reference is constant, i.e. $r(k) = r(k+1)$.

where the elements $u^f(k+i|k+1)$ for $i = 1, \ldots, N_u$ are defined by:

$$u^f(k+i|k+1) = \begin{cases} u^*(k+i|k) & \text{for} \quad i = 1, \ldots, N_u - 1 \\ h(\boldsymbol{x}^f(k+N_u|k+1)) & \text{for} \quad i = N_u \end{cases} \tag{4.48}$$

The use of the shifted solution $\boldsymbol{u}^f(k+1)$ in the optimization problem (4.35) results in the cost $J^f(\boldsymbol{x}(k+1))$ and the predicted states $\boldsymbol{x}^f(k+i|k+1)$ for $i = 2, \ldots, N+1$.

The application of the optimal sequence $\boldsymbol{u}^*(k)$ results in the predicted states $\boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots N_u$. After reaching the end of the control horizon, the local control law $h(\boldsymbol{x}(k+i|k))$, corresponding to the necessary steady-state input $u_r(k)$, is applied to the system. Based on the local control law the states $\boldsymbol{x}^*(k+i|k+1)$ for $i = N_u+1, \ldots N$ are obtained. As the optimal solution $\boldsymbol{u}^*(k)$ and the local control $h(\boldsymbol{x}(k+i|k))$ are based on the conditions given in the optimization problem (4.35), the state vector at $k+N$ satisfies $\boldsymbol{x}^*(k+N|k) \in \Omega(d(k))$.

Due to the fact that the estimation error does not affect the states, the predicted state for $k+1$ made at $k$ satisfies $\boldsymbol{x}^*(k+1|k) = \boldsymbol{x}(k+1)$. With $\boldsymbol{u}^f(k+1)$ being the shifted sequence $\boldsymbol{u}^*(k)$ plus the additional term of the local control law, the states obtained from the application of $\boldsymbol{u}^f(k+1)$ satisfy $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots N_u$. During the remaining prediction horizon the local control law with $h(\boldsymbol{x}(k+i|k+1)) = u_r(k+1)$ is applied to the system. Both the solution $\boldsymbol{u}^f(k+1)$ (based on the shifted optimal solution $\boldsymbol{u}^*(k)$) and the local control law $h(\boldsymbol{x}(k+i|k+1))$ are calculated satisfying the necessary conditions of (4.35). Hence, the statement $\boldsymbol{x}^f(k+N+1|k+1) \in \Omega(d(k+1))$ is true and, as a consequence, the solution $\boldsymbol{u}^f(k+1)$ to the optimization problem is feasible.

#### 4.4.1.3 Convergence

Consider the costs $J^*(\boldsymbol{x}(k))$ at $k$ based on the optimal sequence $\boldsymbol{u}^*(k)$ as well as the cost $J^f(\boldsymbol{x}(k+1))$ at $k+1$ as a result of the feasible solution $\boldsymbol{u}^f(k+1)$. In order to show the convergence it has to be shown that the calculated costs are monotonically decreasing.

The optimal cost $J^*(\boldsymbol{x}(k))$ at $k$ can be written with (4.39) generally as:

$$\begin{aligned} J^*(\boldsymbol{x}(k)) \quad = \quad & \sum_{i=0}^{N_u-1} L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k), d(k)) + \\ & \sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}^*(k+i|k), d(k)) \end{aligned} \tag{4.49}$$

and the cost $J^f(\boldsymbol{x}(k+1))$ at $k+1$ calculated with the feasible solution $\boldsymbol{u}^f(k+1)$ becomes[5]:

$$\begin{aligned} J^f(\boldsymbol{x}(k+1)) &= \sum_{i=1}^{N_u-1} L(\boldsymbol{x}^f(k+i|k+1), u(k+i|k+1), d(k+1)) + \\ & \quad \sum_{i=N_u}^{N} L_h(\boldsymbol{x}^f(k+i|k+1), d(k+1)) \end{aligned} \tag{4.50}$$

Hence, the difference of the two cost functions $\Delta J(k+1) = J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k))$ can be written as:

$$\begin{aligned} \Delta J(k+1) &= L_h(\boldsymbol{x}^f(k+N|k+1), d(k+1)) - L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k)) + \\ & \quad \sum_{i=1}^{N_u-1} \Big( L(\boldsymbol{x}^f(k+i|k+1), u(k+i|k+1), d(k+1)) - \\ & \qquad L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k), d(k)) \Big) + \\ & \quad \sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}^f(k+i|k+1), d(k+1)) - L_h(\boldsymbol{x}^*(k+i|k), d(k)) \end{aligned} \tag{4.51}$$

Now consider a prediction horizon $N \geq N_u + N_t$ and the nilpotent character ($A^{N_t} = 0$) of the state matrix (2.59) of the used prediction model (2.61). With the chosen prediction horizon at $k+1$, the (at least) last $N_t$ input values correspond to the steady-state input $u_r(k+1)$. Therefore, the predicted nominal output reaches steady state at $k+N$ (or earlier). Based on the definition of the steady-state input (4.44) the nominal output is then given by $\tilde{y}(k+N|k+1) = r(k+1) - d(k+1)$. Furthermore, with the nominal output defined as $\tilde{y}(k+i|k) = \mathrm{f}(\boldsymbol{x}(k+i|k))$ (4.42) the statement $\mathrm{f}(\boldsymbol{x}^f(k+N|k+1)) = r(k+1) - d(k+1)$ holds. Hence, with $h(\boldsymbol{x}^f(k+N|k+1)) = u_r(k+1)$ and $\mathrm{f}(\boldsymbol{x}^f(k+N|k+1)) = r(k+1) - d(k+1)$ the stage cost for $k+N$ computed at $k+1$ satisfies $L_h(\boldsymbol{x}^f(k+N|k+1), d(k+1)) = 0$. Finally, the difference in the cost functions can be expressed:

$$\Delta J(k+1) = -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k)) + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \tag{4.52}$$

[5]Note that the upper bound in the first sum has been changed from $N_u$ to $N_u - 1$. With $u^f(k+N_u|k+1) = h(\boldsymbol{x}^f(k+N_u|k+1))$ (4.48) the stage cost for $k+N_u$ calculated at $k+1$ can be included in the second sum by changing the lower bound from $N_u+1$ to $N_u$.

with the terms

$$\begin{aligned}\alpha_1 &= \sum_{i=1}^{N_u-1}\Big(\|\mathrm{f}(\boldsymbol{x}^f(k+i|k+1))+d(k+1)-r(k+1)\|_Q^2 - \\ &\qquad \|\mathrm{f}(\boldsymbol{x}^*(k+i|k))+d(k)-r(k)\|_Q^2\Big)\end{aligned} \tag{4.53}$$

$$\begin{aligned}\alpha_2 &= \sum_{i=1}^{N_u-1}\Big(\|u^f(k+i|k+1)-u_r(k+1)\|_R^2 - \\ &\qquad \|u^*(k+i|k)-u_r(k)\|_R^2\Big)\end{aligned} \tag{4.54}$$

$$\begin{aligned}\alpha_3 &= \sum_{i=N_u}^{N-1}\Big(\|\mathrm{f}(\boldsymbol{x}^f(k+i|k+1))+d(k+1)-r(k+1)\|_Q^2 - \\ &\qquad \|\mathrm{f}(\boldsymbol{x}^*(k+i|k))+d(k)-r(k)\|_Q^2\Big)\end{aligned} \tag{4.55}$$

$$\begin{aligned}\alpha_4 &= \sum_{i=N_u}^{N-1}\Big(\|h(\boldsymbol{x}^f(k+i|k+1))-u_r(k+1)\|_R^2 - \\ &\qquad \|h(\boldsymbol{x}^*(k+i|k))-u_r(k)\|_R^2\Big)\end{aligned} \tag{4.56}$$

**Lemma 4.1** *A quadratic function $g(a) = a^2$ is locally Lipschitz continuous in $a \in [b_1, b_2]$ with $-\infty < b_1 \leq b_2 < \infty$. With this condition a Lipschitz constant $L_q$ can be found such that $\|g(a_1 + a_2) - g(a_1)\| \leq L_q\|a_2\|$.*

**Lemma 4.2** *The inverse $\chi^{-1}$ of the static output nonlinearity of the Volterra model in state space representation is Lipschitz continuous and, as a consequence the condition $\|\chi^{-1}(a_1 + a_2) - \chi^{-1}(a_1)\| \leq L_\chi\|a_2\|$ is satisfied.*

**Lemma 4.3** *The output nonlinearity* f *of the Volterra model in state space representation is Lipschitz continuous and can be bounded by $\|\mathrm{f}(\boldsymbol{a}_1 + \boldsymbol{a}_2) - \mathrm{f}(\boldsymbol{a}_1)\| \leq L_f\|\boldsymbol{a}_2\|$.*

**Lemma 4.4** *The predicted states satisfy $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots, N_u$ as $u^f(k + i|k + 1) = u^*(k + i|k)$ for $i = 1, \ldots, N_u - 1$. The prediction of the states based on the optimal and the feasible solution are defined for $i = N_u + 1, \ldots, N$ by:*

$$\begin{aligned}\boldsymbol{x}^*(k+i|k) &= A\boldsymbol{x}^*(k+i-1|k)+Bu_r(k)\\ \boldsymbol{x}^f(k+i|k+1) &= A\boldsymbol{x}^f(k+i-1|k+1)+Bu_r(k+1)\end{aligned} \tag{4.57}$$

*Then the difference between the states* $\Delta\boldsymbol{x}(k+i) = \boldsymbol{x}^f(k+i|k+1) - \boldsymbol{x}^*(k+i|k)$ *for* $i = N_u+1, \ldots, N$ *can be written as:*

$$\Delta\boldsymbol{x}(k+i) = A\Delta\boldsymbol{x}(k+i-1) + B\Delta u_r \tag{4.58}$$

*where* $\Delta u_r = u_r(k+1) - u_r(k)$ *represents the difference of the steady-state input. With* $\Delta u_r$ *bounded by input constraints the difference of the predicted states is also bounded by*

$$\|\Delta\boldsymbol{x}(k+i)\| \leq c_{\boldsymbol{x}} \|\Delta u_r\| \tag{4.59}$$

*being* $c_{\boldsymbol{x}}$ *a constant.*

**Term $\alpha_1$:** The predicted states based on the optimal and the feasible solution satisfy $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots, N_u - 1$ and for the reference applies $r(k) = r(k+1)$. Furthermore the increment in the estimation error is defined as $\Delta d = d(k+1) - d(k)$. With $z(k+i|k) = \mathrm{f}(\boldsymbol{x}^*(k+i|k)) + d(k) - r(k)$ the term $\alpha_1$ can be expressed as:

$$\alpha_1 = \sum_{i=1}^{N_u-1} \|z(k+i|k) + \Delta d\|_Q^2 - \|z(k+i|k)\|_Q^2 \tag{4.60}$$

Applying Lemma 4.1 to (4.60) the term $\alpha_1$ can be bounded by:

$$\alpha_1 \leq c_1(Q, N_u) \cdot \|\Delta d\| \tag{4.61}$$

being $c_1(\cdot,\cdot)$ a constant depending on the weighting factor $Q$ and the control horizon $N_u$.

□

**Term $\alpha_2$:** The optimal solution and the feasible solution satisfy $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \ldots, N_u - 1$ (4.48). Furthermore the increment in the steady-state input is defined as $\Delta u_r = u_r(k+1) - u_r(k)$. With $z_1(k+i|k) = u^*(k+i|k) - u_r(k)$ the term $\alpha_2$ can be written as:

$$\alpha_2 = \sum_{i=1}^{N_u-1} \|z_1(k+i|k) - \Delta u_r\|_R^2 - \|z_1(k+i|k)\|_R^2 \tag{4.62}$$

In a first step $\alpha_2$ in the form of (4.62) can be bounded with the help of Lemma 4.1 by:

$$\alpha_2 \leq c_2(R, N_u) \cdot \|\Delta u_r\| \tag{4.63}$$

The increment in the necessary steady-state input is defined generally as $\Delta u_r = \chi^{-1}(r(k+1) - d(k+1)) - \chi^{-1}(r(k) - d(k))$ and for the reference applies $r(k) = r(k+1)$.

With $z_2 = r(k) - d(k)$ and $\Delta d = d(k+1) - d(k)$ the increment in the steady-state input can be rewritten as $\Delta u_r = \chi^{-1}(z_2 - \Delta d) - \chi^{-1}(z_2)$. Under consideration of Lemma 4.2 the norm of $\Delta u_r$ can be bounded by $\|\Delta u_r\| \leq L_\chi \|\Delta d\|$. Using this bound in (4.63) the term $\alpha_2$ can be finally bounded by:

$$\alpha_2 \leq c_2(R, L_\chi, N_u) \cdot \|\Delta d\| \tag{4.64}$$

being $c_2(\cdot,\cdot,\cdot)$ a constant depending on the weighting factor $R$, the parameter $L_\chi$ and the control horizon $N_u$.

□

**Term $\alpha_3$:** Consider the predictions $\boldsymbol{x}^*(k+i|k)$ and $\boldsymbol{x}^f(k+i|k+1)$ with the difference $\Delta\boldsymbol{x}(k+i) = \boldsymbol{x}^f(k+i|k+1) - \boldsymbol{x}^*(k+i|k)$ for $i = N_u, \ldots N-1$ and the initial condition $\boldsymbol{x}^f(k+N_u-1|k+1) = \boldsymbol{x}^*(k+N_u-1|k)$. Defining $z_1(k+i|k) = \mathrm{f}(\boldsymbol{x}^*(k+i|k) + \Delta\boldsymbol{x}(k+i)) - \mathrm{f}(\boldsymbol{x}^*(k+i|k)) + \Delta d$ and $z_2(k+i) = \mathrm{f}(\boldsymbol{x}^*(k+i|k)) + d(k) - r(k)$ under consideration of $\Delta d = d(k+1) - d(k)$ and $r(k+1) = r(k)$, the term $\alpha_3$ can be bounded with Lemma 4.1 in the following form:

$$\begin{aligned} \alpha_3 &= \sum_{i=N_u}^{N-1} \|z_1(k+i) + z_2(k+i)\|_Q^2 - \|z_2(k+i)\|_Q^2 \\ &\leq c_3(Q, N, N_u) \cdot \|z_1(k+i)\| \end{aligned} \tag{4.65}$$

Furthermore, with f being Lipschitz continuous (Lemma 4.3) the term $z_1(k+i|k)$ can be bounded with $\|z_1(k+i|k)\| \leq \|\mathrm{f}(\boldsymbol{x}^*(k+i|k) + \Delta\boldsymbol{x}(k+i)) - \mathrm{f}(\boldsymbol{x}^*(k+i|k))\| + \|\Delta d\| \leq L_f \|\Delta\boldsymbol{x}(k+i)\| + \|\Delta d\|$. Hence, (4.65) can be expressed as:

$$\alpha_3 \leq c_3(Q, N, N_u) \cdot (L_f \|\Delta\boldsymbol{x}(k+i)\| + \|\Delta d\|) \tag{4.66}$$

With the help of Lemma 4.4 the difference of the predicted states based on the optimal and the feasible solution can be bounded with $\|\Delta\boldsymbol{x}(k+i)\| \leq \|c_{\boldsymbol{x}}\, \Delta u_r\|$. Using this upper bound, (4.66) can be rewritten as:

$$\alpha_3 \leq c_3(Q, N, N_u) \cdot (L_f \|c_{\boldsymbol{x}}\, \Delta u_r\| + \|\Delta d\|) \tag{4.67}$$

Finally, with Lemma 4.3 and $\|\Delta u_r\| \leq L_\chi \|\Delta d\|$ (see explanation for Term $\alpha_2$) an upper bound of $\alpha_3$ (4.67) is defined by:

$$\alpha_3 \leq c_3(Q, L_\chi, L_f, c_{\boldsymbol{x}}, N, N_u) \cdot \|\Delta d\| \tag{4.68}$$

depending only on the norm of the increment $\|\Delta d\|$ in the estimation error.

□

**Term $\alpha_4$:** It is clear that the statement

$$\alpha_4 = 0 \tag{4.69}$$

is true as the local control law (4.45) corresponds to the necessary steady-state input signal.

□

Finally, after defining upper bounds for $\alpha_1$, $\alpha_2$ and $\alpha_3$ as well as showing that $\alpha_4 = 0$, the difference $\Delta J(k+1) = J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k))$ (4.52) between the costs for the optimal and the feasible solution can be bounded with:

$$J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k)) \leq -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k)) + c_V \cdot \|\Delta d\| \tag{4.70}$$

where $c_V = c_1(Q, N_u) + c_2(R, L_\chi, N_u) + c_3(Q, L_\chi, L_f, c_{\boldsymbol{x}}, N, N_u)$. With the terms $c_V \cdot \|\Delta d\| > 0$ and $-L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k)) \leq 0$ it is ensured that the cost based on the feasible solution cost will decrease as long as $-L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k)) > c_V \cdot \|\Delta d\|$. As a direct consequence, the system is steered into the set:

$$\Psi_d = \{\boldsymbol{x} \colon L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k)) \leq c_V \cdot \|\Delta d\|\} \tag{4.71}$$

from any arbitrary $\boldsymbol{x}$. However, when the state enters the set $\Psi_d$ it is not guaranteed that the cost decreases, giving rise to the possibility that the system remains in the set or evolves out of it. The stage cost always satisfies $-L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k)) \leq 0$, hence (4.70) can be rewritten in the form:

$$J^f(\boldsymbol{x}(k+1)) \leq J^*(\boldsymbol{x}(k)) + c_V \cdot \|\Delta d\| \tag{4.72}$$

Furthermore, the inequality:

$$J^*(\boldsymbol{x}(k)) + c_V \cdot \|\Delta d\| \leq \max_{\boldsymbol{x} \in \Psi_d} J^*(\boldsymbol{x}) + c_V \cdot \|\Delta d\| = \beta_d \tag{4.73}$$

holds for any $\boldsymbol{x}(k) \in \Psi_d$. Then, taking into account (4.72) and (4.73), it follows that:

$$J^f(\boldsymbol{x}(k+1)) \leq \beta_d, \quad \forall\ \boldsymbol{x}(k) \in \Psi_d \tag{4.74}$$

Finally, whenever the state enters into $\Psi_d$, it evolves into the set:

$$\Psi_\beta = \{\boldsymbol{x} \colon J^f(\boldsymbol{x}) \leq \beta_d\} \tag{4.75}$$

Hence, once in the set $\Psi_d$ the state may evolve outside of $\Psi_d$, but it will remain always inside $\Psi_\beta$. Thus, the state is ultimately bounded and the system is stabilized using the feasible solution.

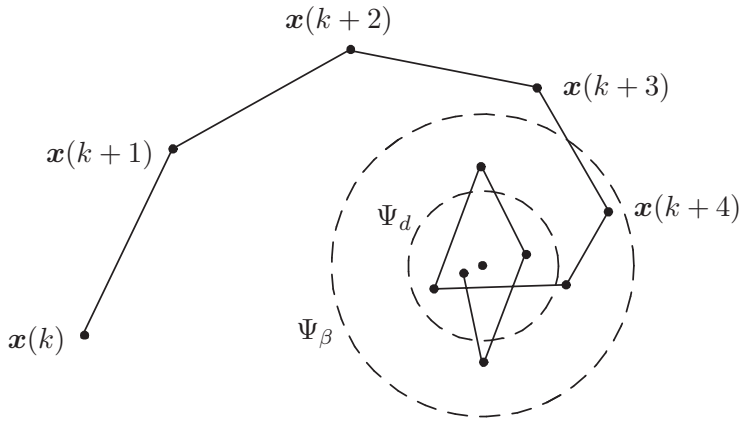


**Figure 4.6**: Evolution of the system state $\boldsymbol{x}(\cdot)$ and the sets $\Psi_d$ and $\Psi_\beta$.

It is obvious that the costs $J^*(\boldsymbol{x}(k+1))$ based on the optimal solution at $k+1$ and $J^f(\boldsymbol{x}(k+1))$ based on the feasible solution satisfy:

$$J^*(\boldsymbol{x}(k+1)) \leq J^f(\boldsymbol{x}(k+1)) \tag{4.76}$$

With (4.74), (4.75) and (4.76) it is straightforward to show that the system controlled by the NMPC strategy based on the optimization problem (4.35) is ultimately bounded by the set $\Psi_\beta$. Hence, with the given proof, the initial optimization problem (4.28) guarantees input-to-state stability. Figure 4.6 shows the possible evolution of the system state $\boldsymbol{x}$ and the sets $\Psi_d$ and $\Psi_\beta$ around the origin (or the desired reference). Once inside the set $\Psi_\beta$, the control strategy maintains the system inside the set.

**Remark 4.5** *Note that the equality constraint used in (4.28) is not necessary to guarantee stability of the proposed NMPC strategy. Nevertheless, considering the equality constraint allows to use the vector and matrices as defined in the Appendix A.1. Otherwise, the prediction model (2.43)-(2.44) has to be reformulated in order to consider the local control law* $h(\boldsymbol{x}(k+i|k)) = u_r$ *for* $i = N_u, \ldots, N-1$.

## 4.5 Experimental results

After the detailed definition of different MPC strategies based on second order Volterra series prediction models, both the constrained controller (see Section 4.3) and the constrained controller with guaranteed stability (see Section 4.4) were applied to the benchmark systems presented in Chapter 3. In the following sections the obtained experimental results will be presented.

### 4.5.1 Constrained optimization

The behavior of the constrained MPC strategy proposed in Section 4.3 was validated in experiments with the three benchmark systems. With the pilot plant different experiments such as setpoint tracking and disturbance rejection were carried out. Afterwards, the proposed controller was applied to the commercial fuel cell and tested in several experiments both with a constant and a variable setpoint. Finally, for the greenhouse the performance of the controller was verified in an experiment stabilizing the temperature in a given setpoint under normal environmental conditions.

#### 4.5.1.1 Pilot plant

In order to carry out different experiments with the pilot plant emulating a chemical reaction (see Section 3.1) the proposed constrained MPC strategy was implemented in the Matlab/Simulink environment with a prediction horizon of $N = 25$, a control horizon of $N_u = 15$, a weighting factor of $\lambda = 5$ and a final accuracy of $\varepsilon = 10^{-3}$. Furthermore, the system input was restricted by the following constraints:

$$\begin{array}{rcccll} 5 & \leq & u(k+i|k) & \leq & 100, & i = 0, \ldots, 14 \\ -20 & \leq & \Delta u(k+i|k) & \leq & 20, & i = 0, \ldots, 14 \end{array} \tag{4.77}$$

In order to analyze the system behavior, several experiments have been carried out to validate the reference tracking and disturbance rejection capabilities of the proposed control strategy.

In the first place a setpoint tracking experiment has been carried out. The chosen values for the reference are different enough to result in control actions in a large interval, see Fig. 4.7. After the reference changes some overshoot (1.7 °C after the first and −1.2 °C after the second setpoint change), justified by the nonlinear process behavior, can be observed. In steady state the controller shows only small changes in the control action, necessary to stabilize the output on the reference in presence of variations in the generated heat and the cold water temperature.

In the second place a setpoint tracking experiment with an error in the model of the exothermic reaction was carried out (see Fig. 4.8). The error introduced in the activation energy $E$ of the chemical reaction (see Section 3.1.2) was held constant during the entire experiment. As the mentioned parameter has a big influence on the reaction, $E$ was increased only by 3 %. The results show some overshoot after the setpoint changes, approximately 2.2 °C after the first step and −3.4 °C after the second one. Nevertheless the proposed controller stabilizes the temperature in the desired

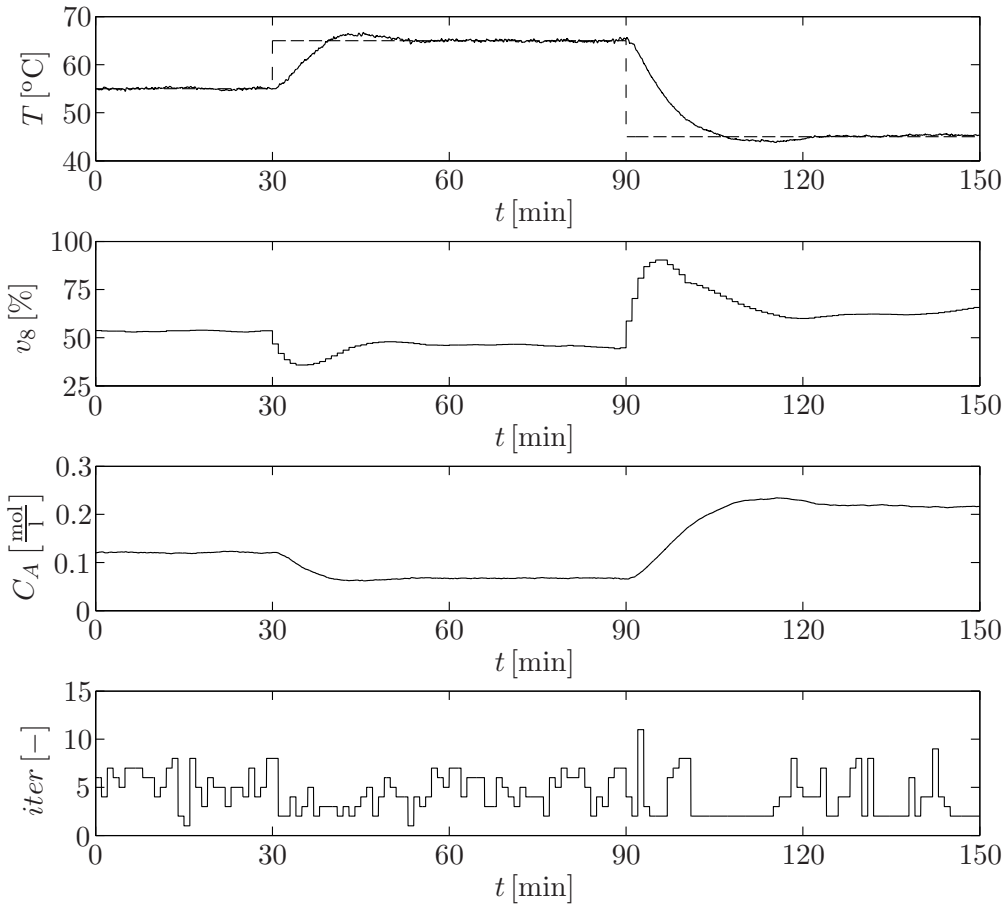


**Figure 4.7**: Experimental results of the pilot plant for setpoint tracking and controlled by the constrained control law. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and necessary iterations *iter* to meet the convergence criterion.

value and no further oscillations can be observed. To see the influence of the modified parameter the results can be compared with the setpoint tracking experiment shown in Fig. 4.7.

In the third experiment the disturbance rejection capabilities of the proposed constrained MPC strategy were proven by means of an additive disturbance in the input of the system. The results of the experiment can be seen in Fig. 4.9 where $u$ denotes the input signal for a given setpoint calculated by the control strategy and $v_8$ represents the effective opening of the valve measured directly in the pilot plant. In normal operation mode and in absence of errors in the integrated valve controller the variables $u$ and $v_8$ have the same values, i.e. $v_8 = u$. In the experiment a discrepancy between $u$ and $v_8$ has been introduced so that it acts as an input disturbance (like an error in the integrated valve controller). With the disturbance $\Delta v_8 = -15\,\%$ the measured opening of the valve has the value $v_8 = u + \Delta v_8$. After the application of the disturbance in

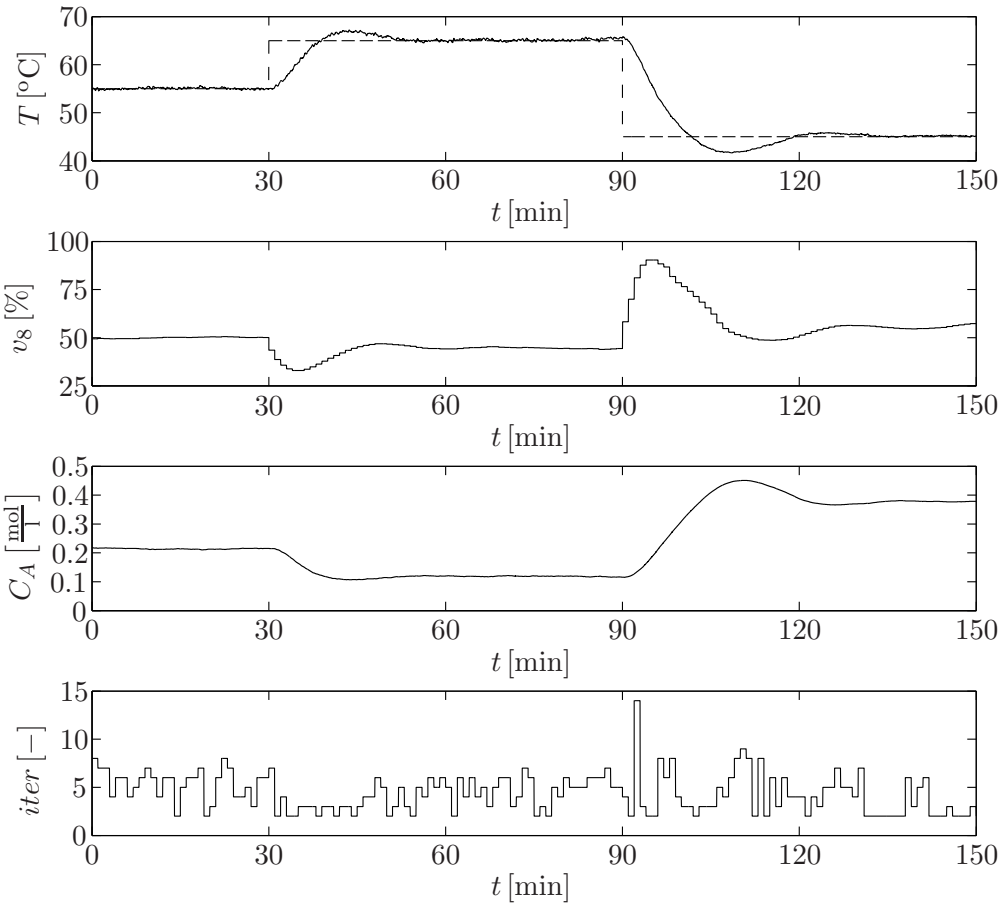


**Figure 4.8**: Experimental results of the pilot plant with a persistent disturbance in the emulated chemical reaction and controlled by the constrained control law. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and necessary iterations *iter* to meet the convergence criterion.

$t = 70\,\text{min}$ and the disappearance in $t = 110\,\text{min}$ the proposed control strategy reacts rapidly to the increasing error in the temperature and shows good disturbance rejection capabilities. Neither the temperature nor the control action show oscillations after the application and the disappearance of the disturbance.

Finally, Fig. 4.10 shows the experimental results applying an additive disturbance in the feed $F_f$. In $t = 60\,\text{min}$ a change in the feed flow rate of $\Delta F_f = -0.02\,\text{l/s}$, which corresponds to an error of 40 %, has been applied. With an increasing error in the measured temperature, the controller reduces the opening of the valve and reaches a compensation of the divergence after 25 minutes. In this experiment a maximal error of $-5\,^{\circ}\text{C}$ in the measured temperature can be observed. In spite of the strong disturbance applied to the system, the proposed control strategy rejects the disturbance without oscillations in the temperature and the control action.

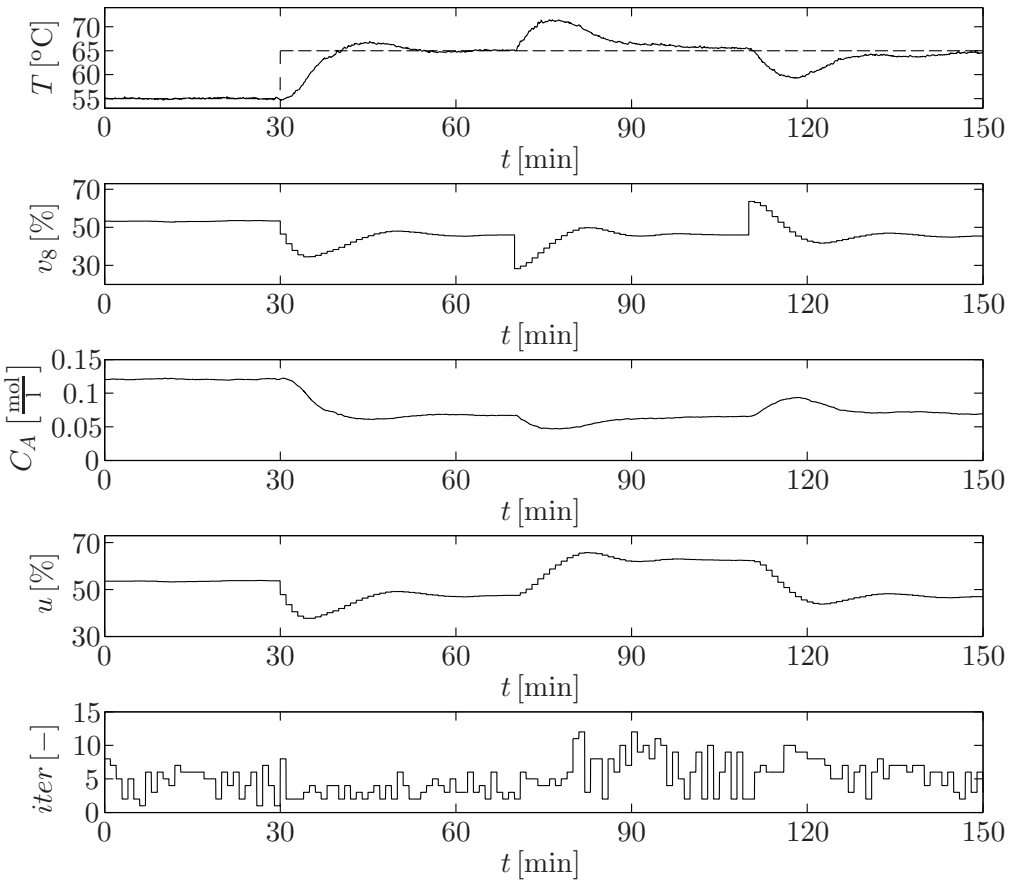


**Figure 4.9**: Experimental results of the pilot plant with a disturbance in the opening of the valve $v_8$ and controlled by the constrained control law. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, input value $u$ calculated by the controller and necessary iterations *iter* to meet the convergence criterion.

It is important to mention that the calculation of the control signal took place without problems within the chosen sampling time of $t_s = 60\,\text{s}$. During the presented experiments the average computation time was $t_c^{avg} = 0.059\,\text{s}$, with a maximum of $t_c^{max} = 0.486\,\text{s}$ and a minimum of $t_c^{min} = 0.021\,\text{s}$. In order to meet the convergence criterion, the proposed optimization needed an average of 4.69 iterations, with a maximum of 14 and a minimum of 1 iteration.

#### 4.5.1.2 Fuel cell

The proposed constrained MPC strategy based on the identified second order Volterra series model with parameter adaptation to changes in the current $I_{load}$ (see Section 3.2.4) was implemented as a Simulink model and afterwards compiled with the Real-

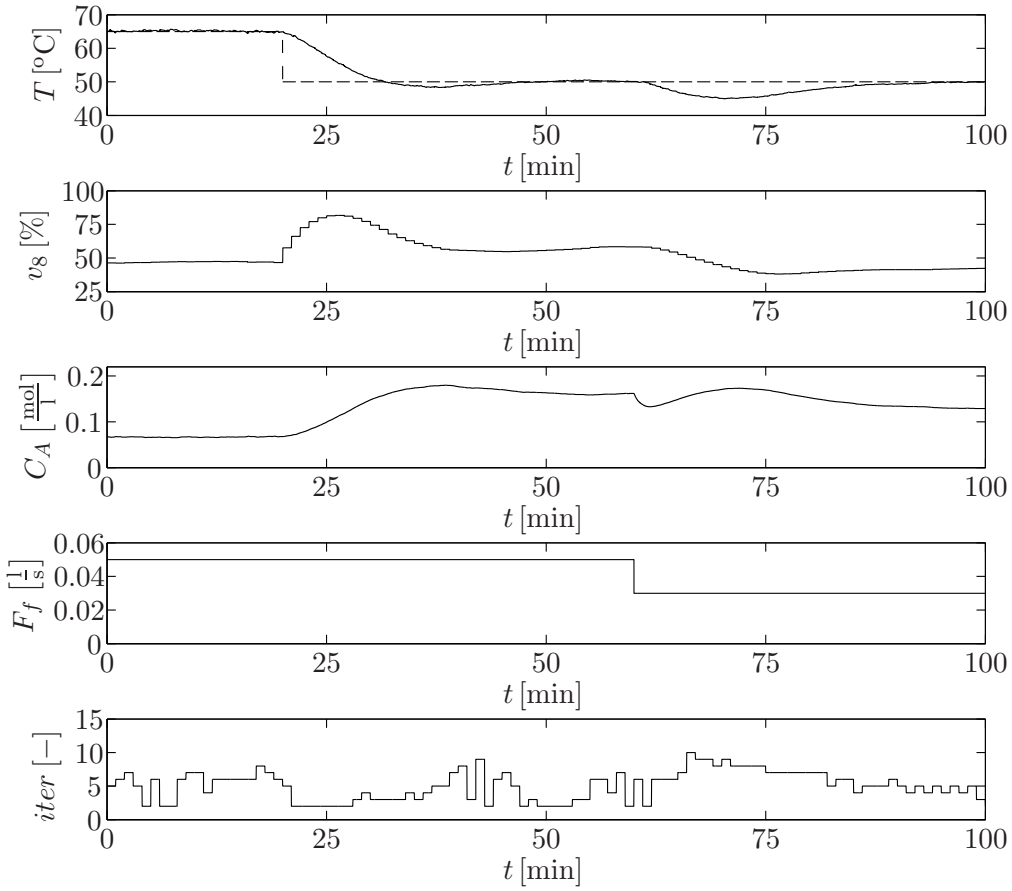


**Figure 4.10**: Experimental results of the pilot plant with a disturbance in the feed $F_f$ and controlled by the constrained control law. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, feed $F_f$ and necessary iterations *iter* to meet the convergence criterion.

Time Workshop toolbox. The compiled Simulink model can then be used directly on the real-time system to regulate the oxygen excess ratio $\lambda_{O_2}$ of the fuel cell module. The quadratic programming was implemented with Lemke's complementary pivoting algorithm [65] and not with the built-in Matlab function as this function is not included in Matlab's real-time framework.

The sampling time for the real-time system and, as a consequence, the control strategy was set to $t_s = 5\,\text{ms}$, corresponding to the used sampling time for the identification process of the Volterra series model. For the prediction and control horizons values of $N = 16$ and $N_u = 5$ were chosen, respectively. The weighting factor used in the control strategy was set to a value of $\lambda = 1$ and a final accuracy of $\varepsilon = 10^{-3}$ was used.

Furthermore, constraints both in the control signal and its increments were considered:

$$\begin{array}{rcccl} 0.8\,\mathrm{V} & \leq & u(k+i|k) & \leq & 7.2\,\mathrm{V}, \quad i=0,\ldots,4 \\ -0.5\,\mathrm{V} & \leq & \Delta u(k+i|k), & & i=0,\ldots,4 \end{array} \tag{4.78}$$

The upper constraint of 7.2 V in the control action corresponds to the maximum input value permitted by the microcontroller of the compressor motor and the lower constraint of 0.8 V has been chosen for safety reasons. The constraint in the increment of the control action had to be used as a faster reduction of the input value leads to an emergency shutdown of the fuel cell module. Furthermore, the number of iterations in the control algorithm presented in Section 4.3.1 was limited to a value of 4. This limitation was necessary in order to guarantee that the real-time system calculates the new control action within the sampling time of $t_s = 5\,\mathrm{ms}$. It is possible that the optimization terminates without reaching the final accuracy due to the limitation in the number of iterations. Note that the limitation in the number of iterations can cause a termination of the optimization without reaching the final accuracy. Nevertheless, the computed input sequence is close to the optimal solution and is applied to the system.

In a first experiment the proposed constrained NMPC strategy was used to regulate the oxygen excess ratio to a desired value of $\lambda_{O_2}^{ref} = 4$, which corresponds to the control objective defined in Section 3.2.2. The results in Fig. 4.11 show that the control strategy regulates the oxygen excess ratio $\lambda_{O_2}$ to the desired value in presence of stepwise changes in the load current $I_{load}$. After compensating the errors in the oxygen excess ratio $\lambda_{O_2}$ due to the changes in the load current $I_{load}$, the input signal $V_{cm}$ hardly varies. The stack voltage $V_{st}$ reaches steady state shortly after a change in the load current $I_{load}$ has been applied and leads to a fast adaption of the net output power $P_{net}$ to the new demand. Furthermore it can be observed that the necessary time $t_c$ to calculate the new input signal has a value between 0.7 ms and 2.5 ms and lies clearly below the sampling time of $t_s = 5\,\mathrm{ms}$. The results show that the applied constrained NMPC strategy controls the oxygen excess ratio $\lambda_{O_2}$ in a wide range of the load current $I_{load}$ and that the fuel cell module supplies the demanded net output power $P_{net}$.

The experiment shown in Fig. 4.11 was repeated with a linear MPC [45] in order to compare the experimental results. The linear MPC was implemented with a sampling time of $t_s = 5\,\mathrm{ms}$ and the same control parameters as the proposed constrained NMPC ($N = 16$, $N_u = 5$, $\lambda = 1$). A detailed comparison of the reaction of both control strategies to changes in the load current $I_{load}$ is given in Fig. 4.12. It can be observed that the proposed constrained NMPC compensates the negative peaks in the oxygen excess ratio $\lambda_{O_2}$ slightly faster due to the more aggressive reaction of the nonlinear control strategy. In the case of the positive peak both controllers need the same time to compensate the error as the control action and its slew rate are limited. Nevertheless, the linear MPC shows a longer overshoot than the NMPC. With the proposed nonlinear

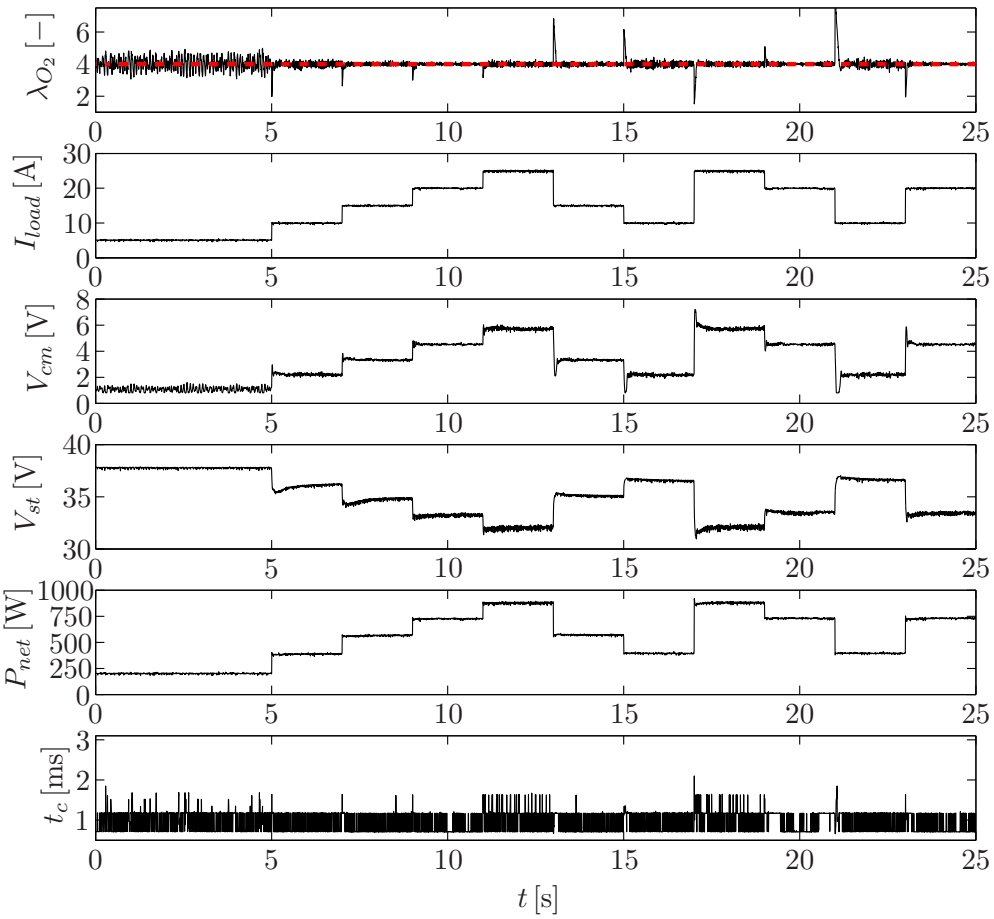


**Figure 4.11**: Experimental results of fuel cell module controlled by the proposed NMPC with stepwise changes in the load current and a constant reference $\lambda_{O_2}^{ref} = 4$ for the oxygen excess ratio. From top to bottom: oxygen excess ratio $\lambda_{O_2}$, load current $I_{load}$, compressor motor voltage $V_{cm}$, stack voltage $V_{st}$, net output power $P_{net}$ and computational time $t_c$.

control strategy the shown negative peaks are compensated in 40 ms and 70 ms and the positive peak in 110 ms. In order to allow a better comparison of the two controllers, the sum of square errors (SSE) is given in Tab. 4.1 for the experiments carried out with the proposed NMPC and the linear MPC. The comparison of the experimental results and the SSEs show a better performance of the constrained NMPC, attributed to the nonlinear dynamics considered in the nonlinear control strategy.

Furthermore experiments with the proposed NMPC and the fuel cell's built-in controller were carried out in order to compare the performance of both controllers in presence of changes in the load current $I_{load}$. In previous experiments [46] it was observed that the built-in controller regulates the oxygen excess ratio $\lambda_{O_2}$ to a variable value depending on the load current $I_{load}$. It is clear that a comparison of the experimental results is only possible if both controllers regulate the oxygen excess ratio $\lambda_{O_2}$

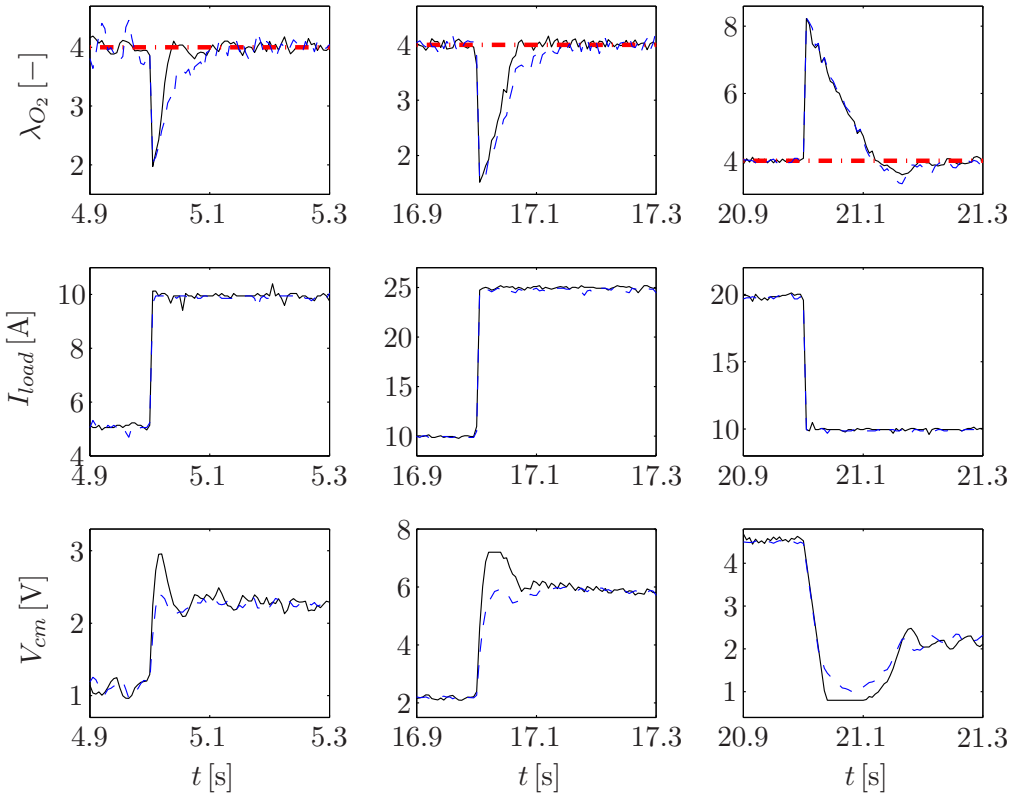


**Figure 4.12**: Experimental comparison of the NMPC (solid line) and a linear MPC (dashed line) with stepwise changes in the load current and a constant reference $\lambda_{O_2}^{ref} = 4$ (dash-dotted line). From top to bottom: oxygen excess ratio $\lambda_{O_2}$, load current $I_{load}$ and compressor motor voltage $V_{cm}$.

| Performance index | NMPC | MPC |
|---|---|---|
| SSE, $t = [0, 25]$ s | 336.67 | 413.63 |
| SSE, $t = [4.9, 5.3]$ s | 12.06 | 19.25 |
| SSE, $t = [16.9, 17.3]$ s | 31.89 | 41.03 |
| SSE, $t = [20.9, 21.3]$ s | 127.76 | 137.03 |

**Table 4.1**: Comparison of the sum of square errors (SSE) for the experiments shown in Figs. 4.11 and 4.12 for the proposed constrained MPC strategy and a linear MPC.

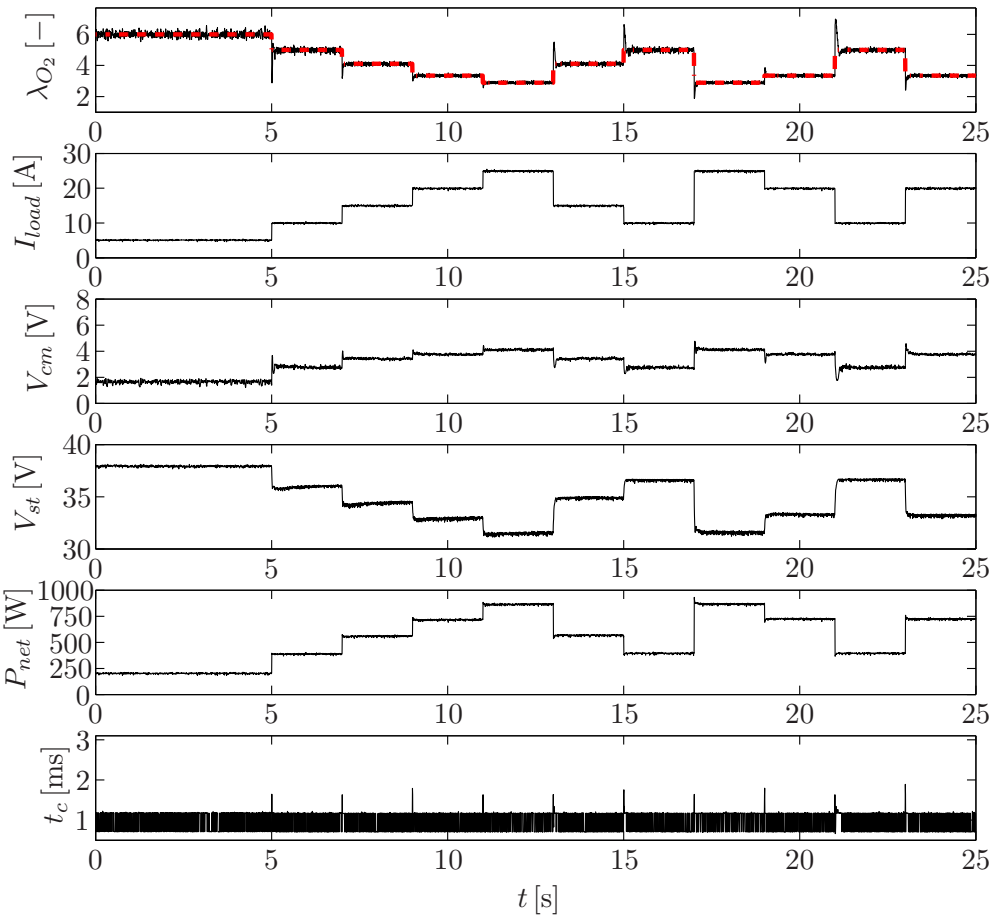


**Figure 4.13**: Experimental results of fuel cell module controlled by the proposed NMPC with stepwise changes in the load current and a varying reference $\lambda_{O_2}^{ref}$ for the oxygen excess ratio. From top to bottom: oxygen excess ratio $\lambda_{O_2}$, load current $I_{load}$, compressor motor voltage $V_{cm}$, stack voltage $V_{st}$, net output power $P_{net}$ and computational time $t_c$.

to the same value. Therefore a reference generator for the NMPC was developed from experimental data obtained from the fuel cell controlled by the built-in controller. The reference generator, a static nonlinearity and implemented as a simple lookup table, calculates the desired value $\lambda_{O_2}^{ref}$ for the oxygen excess ratio based on the measured value of the load current $I_{load}$. Finally, identical experiments were carried out with the proposed NMPC and the built-in controller.

The experimental results of the fuel cell module controlled by the proposed NMPC in combination with a varying setpoint generated by the reference generator are given in Fig. 4.13. The results show that NMPC effectively compensates errors in the oxygen excess ratio $\lambda_{O_2}$ caused by changes in the load current $I_{load}$. It can be observed that the input signal $V_{cm}$ shows only insignificant variations after compensating the divergence between the oxygen excess ratio $\lambda_{O_2}$ and its reference $\lambda_{O_2}^{ref}$. Shortly after a change in

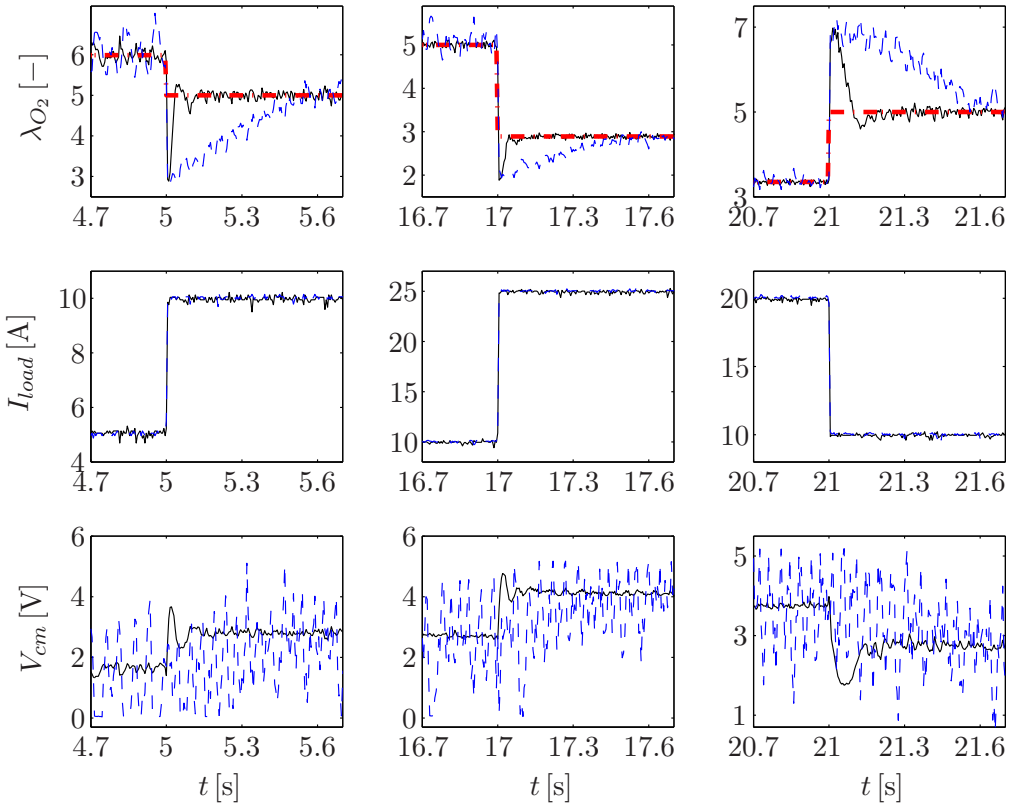


**Figure 4.14**: Experimental comparison of the NMPC (solid line) and the fuel cell's built-in controller (dashed line) with stepwise changes in the load current and a varying reference $\lambda_{O_2}^{ref}$ (dash-dotted line). From top to bottom: oxygen excess ratio $\lambda_{O_2}$, load current $I_{load}$ and compressor motor voltage $V_{cm}$.

the load current $I_{load}$ both the stack current $V_{st}$ and the net output power $P_{net}$ reach steady state. The necessary time $t_c$ to compute a new input signal lies between 0.8 ms and 2.4 ms and therefore clearly below the sampling time of $t_s = 5$ ms. It can be seen from the results that the proposed NMPC can be used to control the oxygen excess ratio $\lambda_{O_2}$ for a varying setpoint and a wide range of operating points.

The experiment carried out with the NMPC to control the oxygen excess ratio at a variable setpoint was repeated with the built-in controller. A detailed comparison of the results of these two controllers is shown in Fig. 4.14. It can be observed in the results that the maximum deviation of the oxygen excess ratio $\lambda_{O_2}$ is the same for both controllers as it is physically impossible to avoid these characteristic peaks. Nevertheless, the NMPC shows a fast reaction to deviations and compensates the two shown negative peaks in approximately 40 ms and the positive peak in 80 ms. In contrast, the built-in controller needs in the three shown cases at least 500 ms to compensate the error in the oxygen excess ratio. With respect to the compressor motor

| Performance index | NMPC | Built-in c. |
|---|---|---|
| SSE, $t = [0, 25]\,\mathrm{s}$ | 128.61 | 850.51 |
| SSE, $t = [4.7, 5.7]\,\mathrm{s}$ | 18.99 | 161.86 |
| SSE, $t = [16.7, 17.7]\,\mathrm{s}$ | 7.16 | 33.45 |
| SSE, $t = [20.7, 21.7]\,\mathrm{s}$ | 37.74 | 225.89 |

**Table 4.2**: Comparison of the sum of square errors (SSE) for the experiments shown in Figs. 4.13 and 4.14 for the NMPC and the built-in controller.

voltage $V_{cm}$, the NMPC reacts very fast to deviations in the oxygen excess ratio and after the compensation of the deviations the input signal shows little variations. Note that the built-in controller uses internally a sampling time of approximately 1 ms. Due to limitations of the used real-time system the input signal $V_{cm}$ computed by the built-in controller was measured with a sampling time of $t_s = 5\,\mathrm{ms}$. For the entire experiment and the sections shown in Fig. 4.14, the sum of square errors (SSE) is given in Tab. 4.2 for the proposed NMPC and the built-in controller. The comparison of the results in Fig. 4.14 and the SSEs show that the proposed controller has a considerably better performance than the original built-in controller with respect to the compensation of errors in the oxygen excess ratio $\lambda_{O_2}$, especially after changes in the load current $I_{load}$.

#### 4.5.1.3 Greenhouse

The proposed constrained NMPC based on the identified Volterra series model (see Section 3.3.2) was implemented in Matlab/Simulink and LabView and validated in experiments with the greenhouse. Both for the prediction horizon and the control horizon values of $N = 9$ and $N_u = 9$ have been chosen. For the final accuracy a value of $\varepsilon = 10^{-3}$ has been used and the sampling time of the controller was set to $t_s = 60\,\mathrm{s}$ (the chosen sampling time corresponds to the sampling time used during the identification process, see Section 3.3.2). The following constraints both in the control signal and its increments were considered:

$$\begin{array}{rcll} 0 & \leq & u(k+i|k) & \leq 100, \qquad i = 0, \ldots, 8 \\ -25 & \leq & \Delta u(k+i|k) & \leq 25, \qquad i = 0, \ldots, 8 \end{array} \tag{4.79}$$

The constraints in the control signal and its increments have been chosen according to the physical limitations of the actuators. The lower and upper bounds of the control signal represent completely closed or opened windows. The constraints in the control increments were necessary as the windows need several minutes to completely open or close.

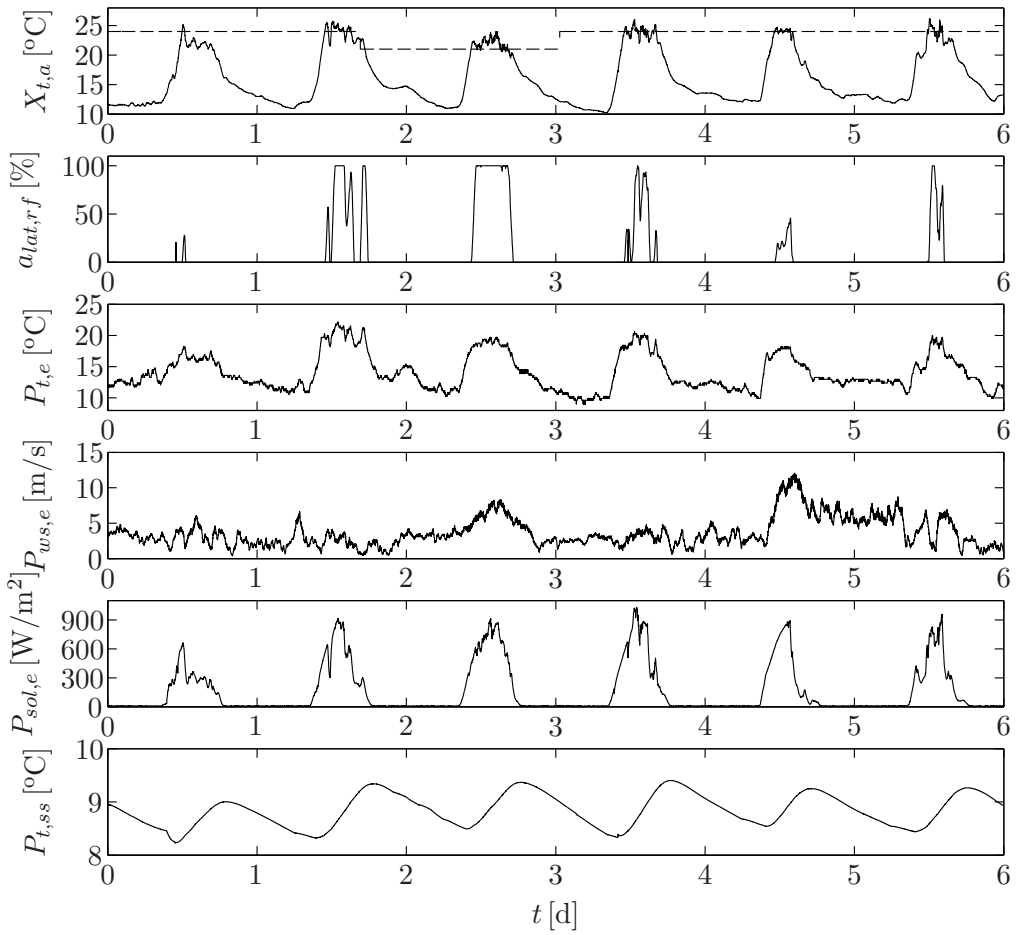


**Figure 4.15**: Experimental results of the proposed controller from January 2010. From top to bottom: output $X_{t,a}$ (greenhouse temperature) and desired reference, input $a_{lat,rf}$ (aperture of the roof and lateral windows) and disturbances $P_{t,e}$ (outside temperature), $P_{ws,e}$ (outside wind speed), $P_{sol,e}$ (outside global solar radiation) and $P_{t,ss}$ (soil surface temperature).

The proposed constrained NMPC was validated under different weather conditions, in Fig. 4.15 some results from January 2010 can be seen. The experimental results show the evolution of the main variables for a period of 6 days. During the shown experiment, a value of 24 °C has been used for the setpoint of the greenhouse temperature $X_{t,a}$; only during the third day a setpoint of 21 °C was chosen. It can be seen that both under clear day and under strong disturbances mainly in solar radiation and wind speed, the system is able to regulate the greenhouse temperature $X_{t,a}$ around the given setpoint while rejecting disturbances.

To give a more detailed view of the controller performance under strong disturbances in outside weather, a zoom of the results of the fifth day is shown in Fig. 4.16. The proposed NMPC stabilizes the controlled temperature $X_{t,a}$ during more than 3 hours

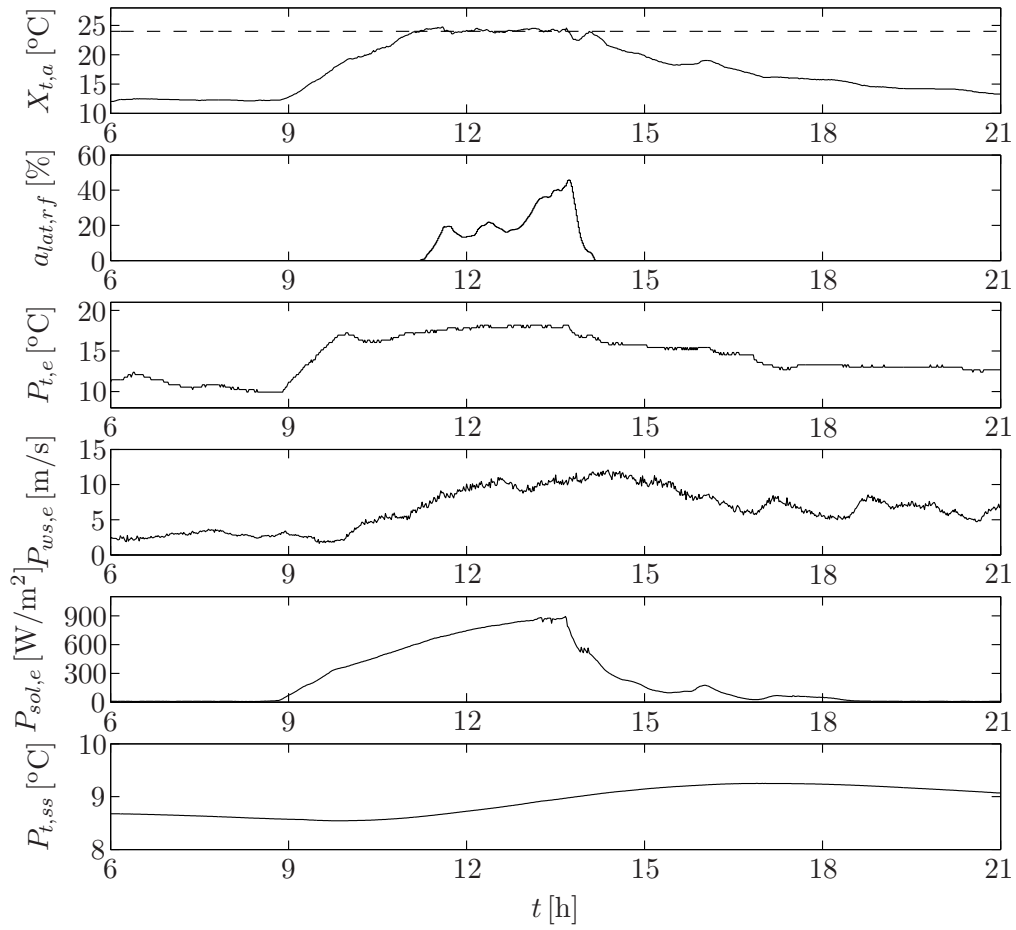


**Figure 4.16**: Detailed experimental results of the proposed controller for one day (day 5 in Fig. 4.15). From top to bottom: output $X_{t,a}$ (greenhouse temperature) and desired reference, input $a_{lat,rf}$ (aperture of the roof and lateral windows) and disturbances $P_{t,e}$ (outside temperature), $P_{ws,e}$ (outside wind speed), $P_{sol,e}$ (outside global solar radiation) and $P_{t,ss}$ (soil surface temperature).

in the given setpoint. During the 3 hours, the temperature is maintained in an interval around the setpoint with a maximum positive deviation of 0.75 °C and a maximum negative deviation of −0.45 °C. The results show a good control performance and underline the possibility to approximate the greenhouse dynamics by means of a second order Volterra series model. Furthermore, the controller demonstrates its ability to reject even strong disturbances.

### 4.5.2 Constrained optimization with guaranteed stability

Analogously to the constrained NMPC, the proposed constrained NMPC with guaranteed stability (see Section 4.4) was validated in experiments. The validation was carried out in different experiments as setpoint tracking and disturbance rejection with the pilot plant emulating a chemical reaction (see Section 3.1).

For the experiments, the proposed control strategy with guaranteed stability was implemented in Matlab/Simulink with a prediction horizon of $N = 80$, a control horizon of $N_u = 15$ and a sampling time of $t_s = 60\,\mathrm{s}$. With the chosen horizons and the truncation order of $N_t = 60$ (see Section 3.1.3), the necessary condition $N \geq N_t + N_u$ is fulfilled. For the weighting factor and the final accuracy values of $\lambda = 5$ and $\varepsilon = 10^{-3}$ were used. The input was restricted to:

$$\begin{aligned} 5 \leq \quad & u(k+i|k) \quad \leq 100, \quad i = 0, \ldots, 14 \\ -20 \leq \quad & \Delta u(k+i|k) \quad \leq 20, \quad i = 0, \ldots, 14 \\ u(k+14|k) = \quad & u_r(k) \end{aligned} \tag{4.80}$$

which corresponds to the constraints defined in (4.77) plus the necessary terminal equality constraint to guarantee stability. Note that the experiments carried out with the pilot plant to analyze the control behavior correspond to the ones shown in Section 4.5.1.1.

The results of a setpoint tracking experiment with the pilot plant emulating an exothermic chemical reaction are given in Fig. 4.17. The results show that the system reaches steady state without any overshoot, both after the first and after the second change in the setpoint. It can be observed that the controller reacts in a first moment very aggressive to the setpoint changes. Later, the controller shows a very smooth behavior and leads the system to the new setpoint. With respect to the results given in Fig. 4.7, the system controlled by the constrained NMPC with guaranteed stability needs the same time to reach steady state.

In the second experiment the disturbance rejection capabilities of the proposed controller were tested, see Fig. 4.18. Therefore an error in the activation energy $E$ of the mathematical model of the underlying exothermic chemical reaction was introduced. The parameter $E$ was increased by 3 % and held constant during the entire experiment. The results show some overshoot after both setpoint changes (approximately $1.3\,^{\circ}\mathrm{C}$ after the first step and $-1.7\,^{\circ}\mathrm{C}$ after the second one). The proposed controller with guaranteed stability needs considerably longer to completely reject the introduced disturbance than the constrained controller (see Fig. 4.8). Nevertheless, the proposed NMPC with guaranteed stability obtains a notably reduction in the overshoot in com-

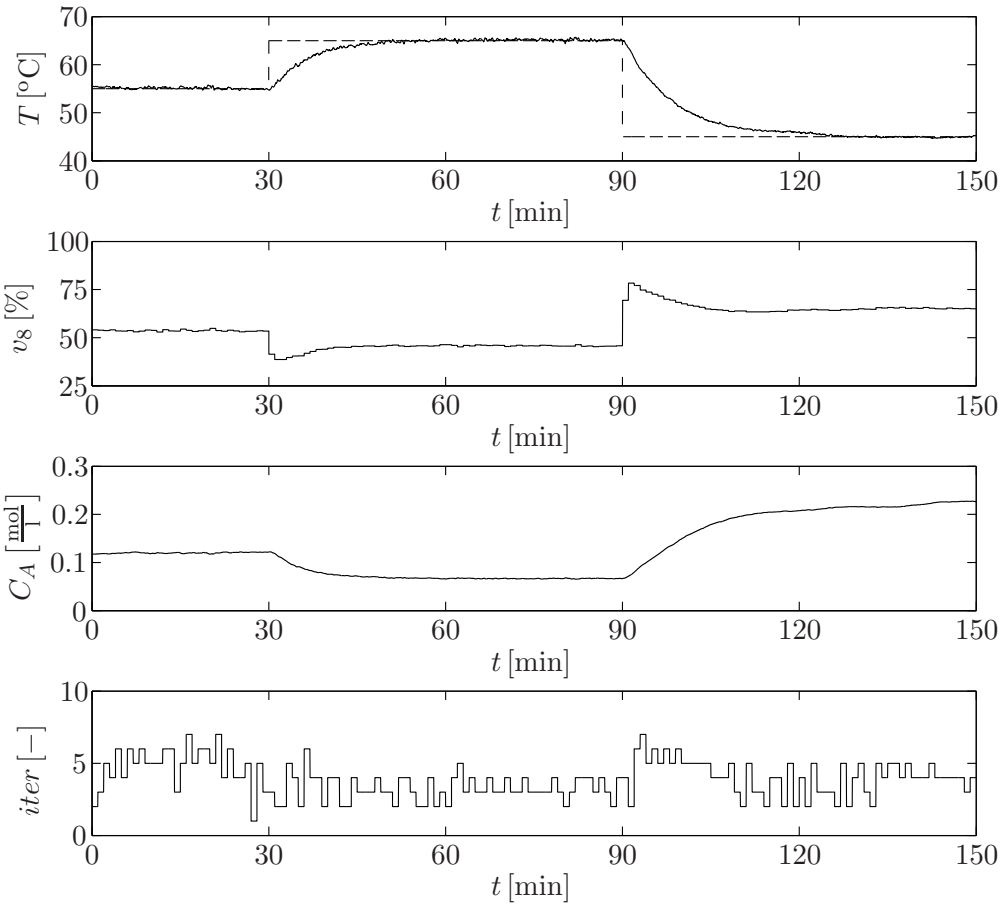


**Figure 4.17**: Reference tracking experiment with the pilot plant applying the constrained control law with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and necessary iterations *iter* to meet the convergence criterion.

parison with the overshoot observed in Fig. 4.8.

In the third experiment with the proposed NMPC with guaranteed stability an additive disturbance in the input of the system was applied. The disturbance $\Delta v_8 = -15\,\%$ was applied in $t = 70\,\text{min}$ and removed in $t = 110\,\text{min}$. In absence of the disturbance the effective opening of the valve is given by $v_8 = u$. During the application of the disturbance in the system input, the valve opening is given by $v_8 = u + \Delta v_8$. In the results shown in Fig. 4.19, it can be observed, that the proposed NMPC with guaranteed stability rejects the applied disturbance and compensates effectively the error in the temperature. In comparison with the results shown in Fig. 4.9, it can be seen that the application of the disturbance to the system controlled by the NMPC with guaranteed stability results in a slightly higher maximum error in the temperature and in a somewhat slower compensation of the divergence. However, the proposed NMPC

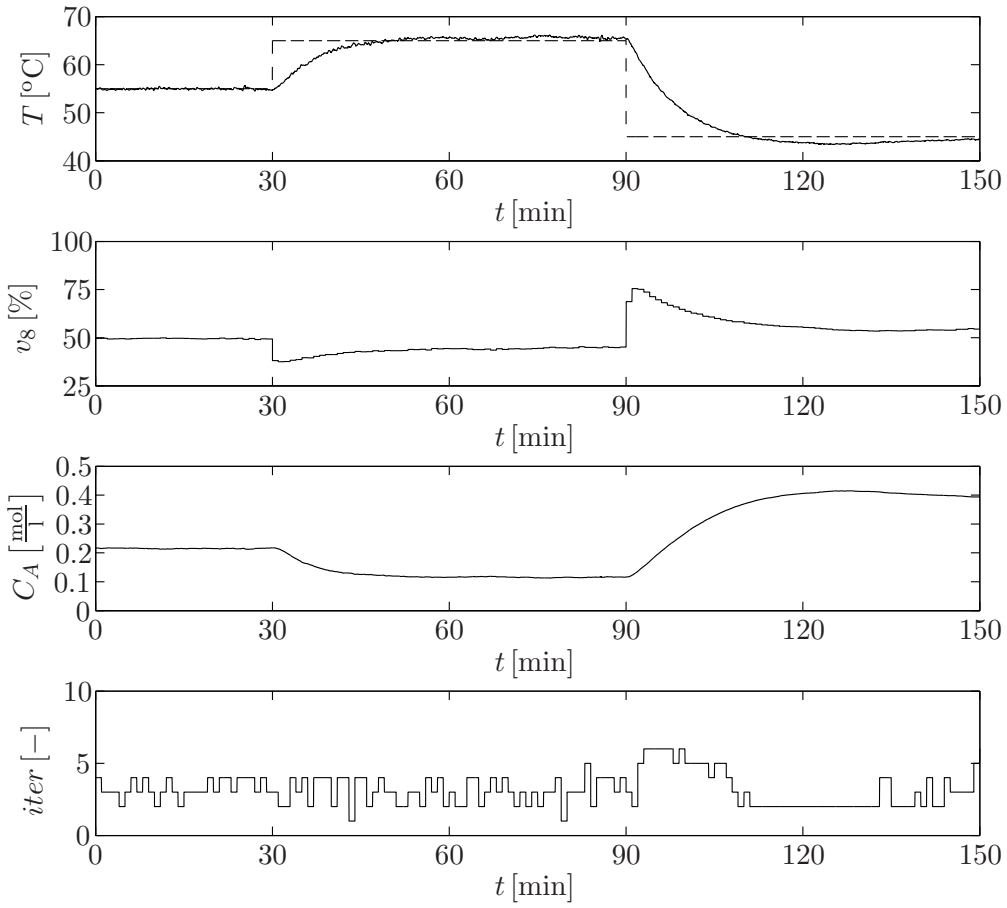


**Figure 4.18**: Experimental results of the pilot plant with a persistent disturbance in the emulated chemical reaction and controlled by the constrained control law with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and necessary iterations *iter* to meet the convergence criterion.

with guaranteed stability shows good disturbance rejection capabilities and leads the temperature to the given setpoint.

In the last experiment, see Fig. 4.20, the reaction of the system to an additive disturbance in the feed $F_f$ was tested. The disturbance $\Delta F_f = -0.02\,\mathrm{l/s}$, which corresponds to an error of $-40\,\%$, was applied in $t = 60\,\mathrm{min}$. After the application of the disturbance the temperature of the pilot plant decreases as a result of the reduced concentration $C_A$ and reaches a maximum error of $-5.7\,^{\circ}\mathrm{C}$. As a result of the increasing error in the temperature, the proposed controller reduces the opening of the valve $v_8$ and, as a consequence, compensates the error. Finally, at the end of the experiment, the temperature $T$ nearly reaches steady state. The proposed controller effectively rejects the disturbance, but clearly slower and with a higher maximum error than the controller used in the results shown in Fig. 4.10

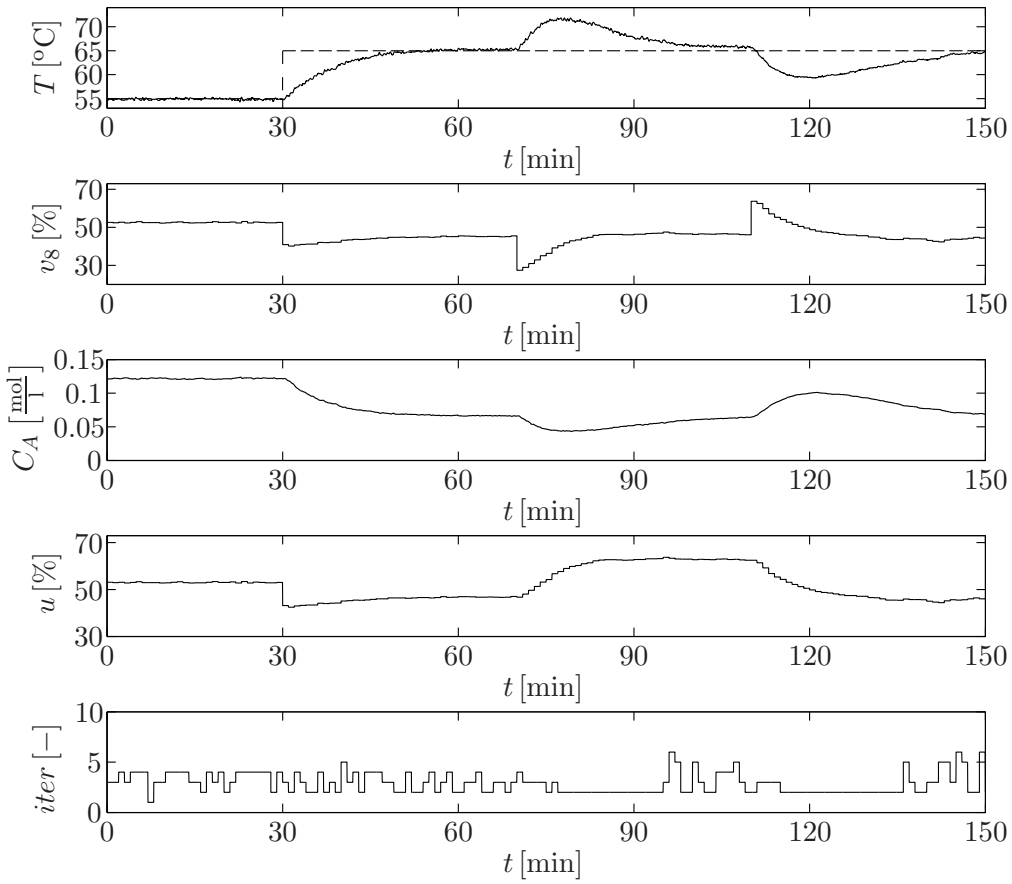


**Figure 4.19**: Experimental results of the pilot plant with a disturbance in the opening of the valve $v_8$ and controlled by the constrained control law with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and necessary iterations *iter* to meet the convergence criterion.

Note that a new input sequence was always calculated within the used sampling time of $t_s = 60\,\text{s}$ during the shown experiments. The average computation time was $t_c^{avg} = 0.076\,\text{s}$, with a minimum of $t_c^{min} = 0.031\,\text{s}$ and a maximum of $t_c^{max} = 0.285\,\text{s}$. With respect to the convergence criterion, the proposed NMPC with guaranteed stability needed an average of 3.27 iterations, with a maximum of 7 and a minimum of 1 iteration.

## 4.6 Conclusions of the chapter

In this chapter, several NMPC strategies based on iterative optimization algorithms have been proposed. The three control strategies are based on quadratic cost functions and second order Volterra series models and allow a fast computation of the optimal

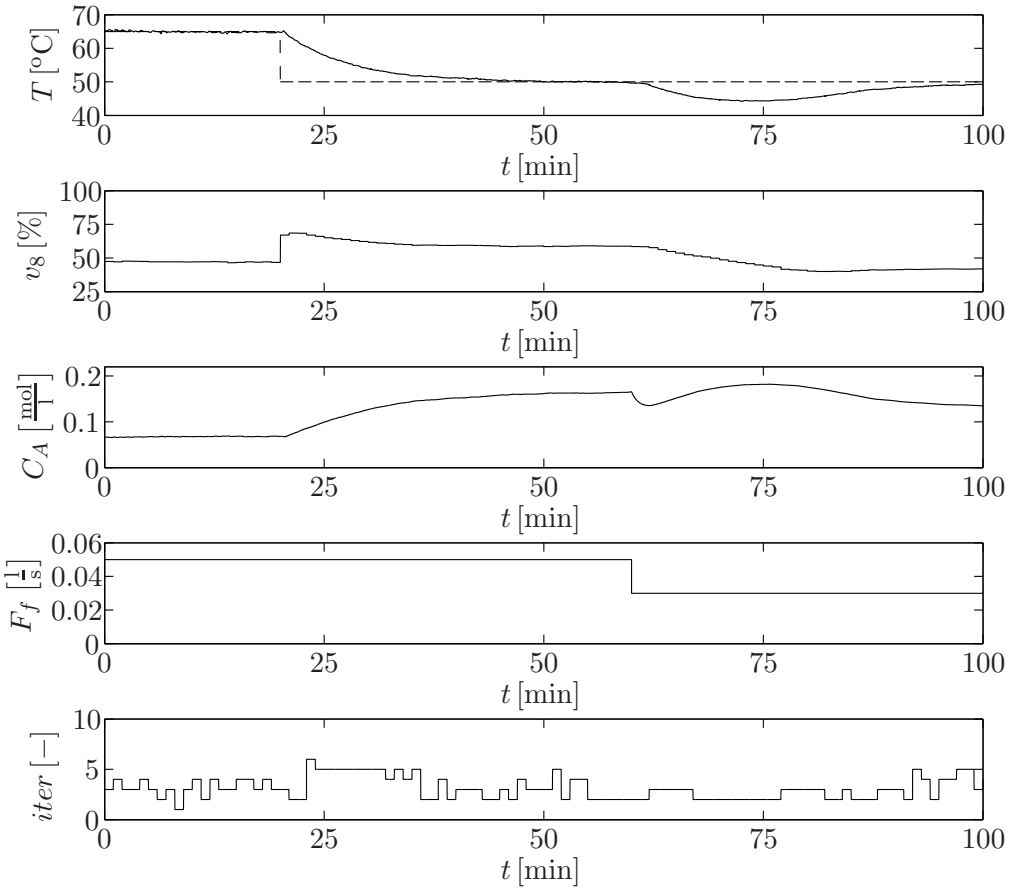


**Figure 4.20**: Experimental results of the pilot plant with a disturbance in the feed $F_f$ and controlled by the constrained control law with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and necessary iterations *iter* to meet the convergence criterion.

input sequence. The unconstrained optimization algorithm in Section 4.2 can be considered as the starting point for the development of new nonlinear control strategies and was originally published in [77, 38]. This algorithm was modified to include linear input constraints and a weighting function for the control effort (Section 4.3). Output constraints have not been considered in the proposed algorithm as the nonlinearity of the used second order Volterra series model results in nonlinear constraints. The combination of the iterative approach and sequential quadratic programming to solve the optimization problem with nonlinear constraints would eliminate the advantage of fast computation. Finally, the constrained NMPC strategy was modified in order to guarantee input-to-state stability (see Section 4.4). The developed control strategies (the constrained control strategy and the constrained control strategy with robust stability) were applied successfully to different benchmark systems and the control behavior was illustrated by means of experimental results.

Note that the three control strategies are based on an iterative optimization approach which does not guarantee convergence if the initial optimization problem is non-convex. In experiments it was observed that the non-convexity leads to optimization problems and a non-converging iterative optimization. Nevertheless, convexity can be obtained for the two constrained control strategies by high values for the weighting factor $\lambda$ of the control effort. In [35] it was shown how to determine the necessary value for $\lambda$ for a control horizon of $N_u = 1$. However, the possible convergence problem represents a serious drawback for the iterative control strategies and requires the development of a control strategy which guarantees a converging optimization.

# Chapter 5

# A convex approach to Volterra based MPC

In this chapter a novel approach to use second order Volterra series models in NMPC will be presented. Quadratic lower bounds of the original cost function are obtained. For those quadratic lower bounds, convexity can be achieved by adding a weighting function for the control effort. Minimizing globally the approximating convex cost functions, a new input sequence can be computed.

In the following section the general idea of convexification for second order functions will be explained. The convexification approach will be used to approximate a cost function based on a second order Volterra series prediction model by convex quadratic functions. Afterwards a control law based on the minimization of the strict convex functions will be presented. For the proposed convexification approach convergence will be proven, one of the main drawbacks in the iterative optimization algorithms presented in Chapter 4. Finally, another control law based on the idea of convexification, with a modification in the cost function and the general optimization problem to guarantee stability, will be presented. Both control strategies will be applied to one of the benchmark systems and the control performance will be illustrated by experimental results.

## 5.1 General idea of convexification

This section shows the general concept to assure strict convexity for a quadratic cost function when the model output is defined by a second order function. Generally, the predicted output error can be defined as:

$$\boldsymbol{e} = \mathbf{p}(\boldsymbol{u}) + \boldsymbol{b} \tag{5.1}$$

with $\mathbf{p}(\boldsymbol{u})$: $\mathbb{R}_u^N \to \mathbb{R}^N$ being a nonlinear second order function in $\boldsymbol{u}$ and $\boldsymbol{b} \in \mathbb{R}^N$ denotes a constant vector. With the predicted output error (5.1) the quadratic cost function without penalization of the control effort (2.46) can be expressed as:

$$J(\boldsymbol{u}) = \boldsymbol{e}^T \boldsymbol{e} \tag{5.2}$$

It is straightforward to show that the cost (5.2) with $\boldsymbol{e}$ being a nonlinear second order function in $\boldsymbol{u}$ may possess more than one minimum. The first order Taylor approximation of (5.2) at $\boldsymbol{e}^{(0)} = \mathbf{p}(\boldsymbol{u}^{(0)}) + \boldsymbol{b}$ is given by:

$$\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}) = \boldsymbol{e}^{(0)T}\boldsymbol{e}^{(0)} + 2\boldsymbol{e}^{(0)T}\left(\boldsymbol{e} - \boldsymbol{e}^{(0)}\right) = -\boldsymbol{e}^{(0)T}\boldsymbol{e}^{(0)} + 2\boldsymbol{e}^{(0)T}\boldsymbol{e} \tag{5.3}$$

Since $J(\boldsymbol{u})$ is convex with respect to $\boldsymbol{e}$, the linearisation $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)})$ satisfies:

$$\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}) \leq J(\boldsymbol{u}), \;\; \forall \; \boldsymbol{e}^{(0)} \tag{5.4}$$

Moreover, since $\tilde{J}(\boldsymbol{u}^{(0)}, \boldsymbol{e}^{(0)}) = J(\boldsymbol{u}^{(0)})$, the statement:

$$J(\boldsymbol{u}) = \sup_{\boldsymbol{e}^{(0)}} \tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}) \tag{5.5}$$

holds. Hence, $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)})$ represents a lower bound of the original cost function (5.2). Now, using the predicted output error (5.1) in (5.3), the approximated cost function can be expressed as a function in $\boldsymbol{u}$:

$$\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}) = 2\boldsymbol{e}^{(0)T}\mathbf{p}(\boldsymbol{u}) + 2\boldsymbol{e}^{(0)T}\boldsymbol{b} - \boldsymbol{e}^{(0)T}\boldsymbol{e}^{(0)} \tag{5.6}$$

It can easily be seen that (5.6) is a second order function, depending only on $\mathbf{p}(\boldsymbol{u})$. In order to check convexity, the approximated cost function $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)})$ is decomposed in its quadratic, linear and constant terms. Therefore the term $2\boldsymbol{e}^{(0)T}\mathbf{p}(\boldsymbol{u})$ is written in the following way:

$$2\boldsymbol{e}^{(0)T}\mathbf{p}(\boldsymbol{u}) = \boldsymbol{u}^T S\, \boldsymbol{u} + 2\boldsymbol{h}^T\boldsymbol{u} + q \tag{5.7}$$

with the $S = S^T \in \mathbb{R}^{N_u \times N_u}$, $\boldsymbol{h} \in \mathbb{R}^{N_u}$ and $q \in \mathbb{R}$ to be defined on basis of the used nonlinear model. Substituting the first term in (5.6) by (5.7), the lower bound of the cost function becomes:

$$\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}) = \boldsymbol{u}^T S\, \boldsymbol{u} + 2\boldsymbol{h}^T\boldsymbol{u} + q + 2\boldsymbol{e}^{(0)T}\boldsymbol{b} - \boldsymbol{e}^{(0)T}\boldsymbol{e}^{(0)} \tag{5.8}$$

It is clear that $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)})$ is strictly convex with respect to $\boldsymbol{u}$ if and only if $S$ is definite positive ($S > 0$). If $S$ is not positive definite, an additional term weighting the control effort can be added to (5.8) to obtain a strict convex cost function. In MPC, this additional term is usually the quadratic function $\lambda \boldsymbol{\Delta u}^T \boldsymbol{\Delta u}$ based on the control increments $\boldsymbol{\Delta u}$ and weighted by the factor $\lambda$. Transforming the weighting term for the control effort as shown in (2.51), the approximated performance index (5.8) including a penalization of the control effort is defined as:

$$\begin{aligned} \tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, \lambda) &= \boldsymbol{u}^T (S + \lambda L_u^T L_u)\boldsymbol{u} + 2(\boldsymbol{h}^T - \lambda \boldsymbol{u}_l^T L_u)\boldsymbol{u} + \\ & \quad q + 2\boldsymbol{e}^{(0)T}\boldsymbol{b} - \boldsymbol{e}^{(0)T}\boldsymbol{e}^{(0)} + \lambda \boldsymbol{u}_l^T \boldsymbol{u}_l \end{aligned} \tag{5.9}$$

where the weighting factor $\lambda$, considered as a design parameter, is chosen in a way that it guarantees strict convexity:

$$S + \lambda L_u^T L_u > 0 \tag{5.10}$$

If $S$ is not positive definite, the condition (5.10) is satisfied and strict convexity of the approximated cost function (5.9) can be assured for the weighting factor:

$$\lambda = -\min\left(\text{eig}\left((L_u^T)^{-1} S L_u^{-1}\right)\right) + \rho \tag{5.11}$$

where $\text{eig}(S)$ denotes the eigenvalues of $S$ and $\rho$ denotes a constant with $\rho > 0$. If $S$ is positive definite, the weighting factor $\lambda = \varphi$ with $\varphi \geq 0$ can be used. Analogously, the original cost function (5.2) extended to include the weighting of the control effort (2.51) is denoted as:

$$J(\boldsymbol{u}, \lambda) = J(\boldsymbol{u}) + \lambda \boldsymbol{u}^T L_u^T L_u \boldsymbol{u}^T - 2\lambda \boldsymbol{u}_l^T L_u \boldsymbol{u} + \lambda \boldsymbol{u}_l^T \boldsymbol{u}_l \tag{5.12}$$

Since $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}) \leq J(\boldsymbol{u})$ (5.4), it follows that:

$$\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, \lambda) \leq J(\boldsymbol{u}, \lambda) \tag{5.13}$$

where the strict inequality holds for $\boldsymbol{u} \neq \boldsymbol{u}^{(0)}$ and the equality is valid in the point of approximation $\boldsymbol{u} = \boldsymbol{u}^{(0)}$. Hence, $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, \lambda)$ can be considered a lower bound of the cost function $J(\boldsymbol{u}, \lambda)$.

Now consider several convex cost functions $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(j-1)}, \lambda^{(j)})$ approximated around $\boldsymbol{e}^{(j-1)}$ for $j = 1, \ldots, n$ where $\lambda^{(j)}$ is the necessary weighting to guarantee convexity for the cost function approximated around $\boldsymbol{e}^{(j-1)}$. With the condition:

$$\lambda^{(j)} \geq \lambda^{(j-1)} \tag{5.14}$$

the weighting factor $\lambda^{(n)}$ assures convexity for all cost functions $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(j-1)}, \lambda^{(n)})$ with $j = 1, \ldots, n$. The maximum of several approximated cost functions based on the weighting factor $\lambda^{(n)}$ is defined by:

$$\tilde{J}_{\Sigma}^{(n)}(\boldsymbol{u}, \lambda^{(n)}) = \max_{j=1,\ldots,n} \left\{ \tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(j-1)}, \lambda^{(n)}) \right\} \tag{5.15}$$

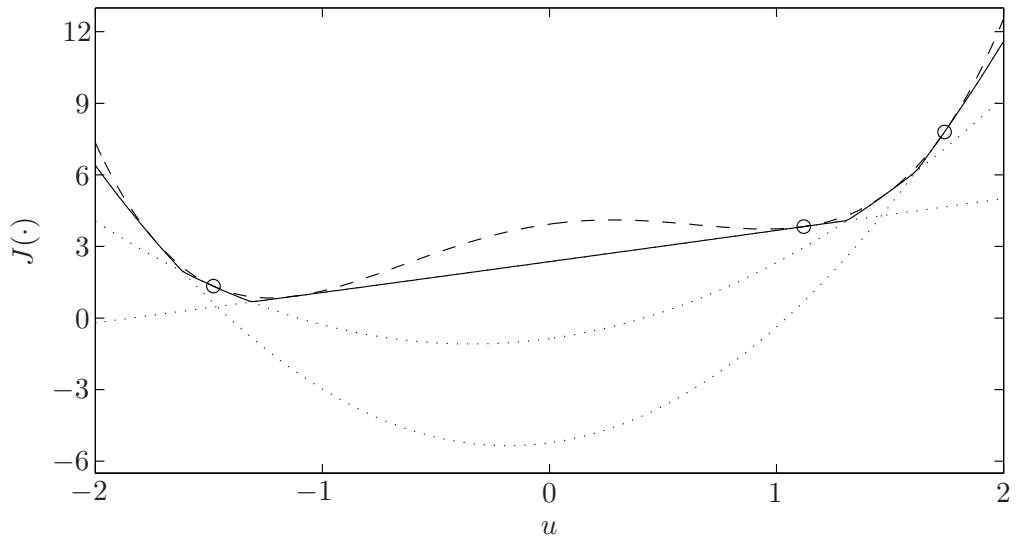


**Figure 5.1**: Example for the approximation by means of convex functions with $J(\boldsymbol{u}, \lambda^{(n)})$ (dashed line), $\tilde{J}_{\Sigma}^{(n)}(\boldsymbol{u}, \lambda^{(n)})$ (solid line) and $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(j-1)}, \lambda^{(n)})$ (dotted line) for $j = 1, \ldots, n$ and $n = 3$ as well as the points of approximation (circles).

Using (5.15) the cost function (5.12) with the weighting factor $\lambda^{(n)}$ can be approximated around different points $\boldsymbol{e}^{(j-1)}$ with $\lambda^{(n)}$ guaranteeing strict convexity for all approximations. The pointwise maximum of these approximated strict convex cost functions is also a strict convex function [21]. With (5.13) it can be shown that $\tilde{J}_{\Sigma}^{(n)}(\boldsymbol{u}, \lambda^{(n)})$ represents a convex hull of the cost function $J(\boldsymbol{u}, \lambda^{(n)})$ [76]. In Fig. 5.1 an example for the approximation of a non-convex function $J(\boldsymbol{u}, \lambda^{(n)})$ by the convex function $\tilde{J}_{\Sigma}^{(n)}(\boldsymbol{u}, \lambda^{(n)})$ is shown. The example has been generated with 3 randomized points of approximation.

## 5.2 Optimization based on convexification

The convexification shown in Section 5.1 can be used in the optimization of NMPC strategies based on second order Volterra series models. Therefore, a quadratic cost function in combination with a second order Volterra series model will be approximated by convex functions. Based on the convexification, an optimization approach for NMPC strategies will be presented. Finally, convergence for the optimization approach will be proven.

For second order Volterra model series, the future output can be expressed in matrix form as shown in (2.43)-(2.44). One of the main drawbacks of this representation is the mixture of linear and nonlinear terms in the variable $\mathbf{f}(\boldsymbol{u})$ as future-future and

future-past cross terms depending on the input sequence $\boldsymbol{u}$ are considered. Therefore the prediction model (2.43)-(2.44) is rewritten in the form:

$$\begin{aligned} \hat{\boldsymbol{y}} &= G\boldsymbol{u} + \mathbf{f}_{nl}(\boldsymbol{u}) + \mathbf{f}_{l}(\boldsymbol{u}) + \boldsymbol{c} && (5.16)\\ \boldsymbol{c} &= H\boldsymbol{u}_p + \boldsymbol{g} + \boldsymbol{h}_0 + \boldsymbol{d} && (5.17) \end{aligned}$$

where the vector $\mathbf{f}_{nl}(\boldsymbol{u}) \in \mathbb{R}^N$ contains the nonlinear future-future cross terms and $\mathbf{f}_{l}(\boldsymbol{u}) \in \mathbb{R}^N$ denotes the vector of linear future-past cross terms. A detailed definition of the vectors $\mathbf{f}_{nl}(\boldsymbol{u})$ and $\mathbf{f}_{l}(\boldsymbol{u})$ is given in the Appendix A.5. The other variables used in (5.16)-(5.17) are defined as shown in Section 2.3 and in the Appendix A.1.

With the modified predicted output for a second order Volterra series model (5.16)-(5.17) the definition (5.1) of the predicted output error for a second order function can be written as:

$$\mathbf{p}(\boldsymbol{u}) + \boldsymbol{b} = G\boldsymbol{u} + \mathbf{f}_{nl}(\boldsymbol{u}) + \mathbf{f}_{l}(\boldsymbol{u}) + \boldsymbol{c} - \boldsymbol{r} \tag{5.18}$$

where $G\boldsymbol{u}$, $\mathbf{f}_{nl}(\boldsymbol{u})$ and $\mathbf{f}_{l}(\boldsymbol{u})$ are the only terms which depend on the input sequence $\boldsymbol{u}$. Hence, $\mathbf{p}(\boldsymbol{u})$ and $\boldsymbol{b}$ for the modified Volterra series prediction model can be defined as:

$$\begin{aligned} \mathbf{p}(\boldsymbol{u}) &= G\boldsymbol{u} + \mathbf{f}_{nl}(\boldsymbol{u}) + \mathbf{f}_{l}(\boldsymbol{u}) && (5.19)\\ \boldsymbol{b} &= \boldsymbol{c} - \boldsymbol{r} && (5.20) \end{aligned}$$

To assure convexity, the term $\mathbf{p}(\boldsymbol{u})$ has to be decomposed as shown in (5.7). With $\mathbf{f}_{nl}(\boldsymbol{u})$ being the only term depending in a quadratic way on $\boldsymbol{u}$, the matrix $S$ is calculated by means of $\mathbf{f}_{nl}(\boldsymbol{u})$. $G\boldsymbol{u}$ and $\mathbf{f}_{l}(\boldsymbol{u})$ depend in a linear form on $\boldsymbol{u}$ and, as a consequence, determine the vector $\boldsymbol{h}$. The constant $q$ is equal to 0 as $\mathbf{p}(\boldsymbol{u})$ does not contain any constant. With these considerations, the following relations for the modified Volterra series prediction model can be defined:

$$\begin{aligned} \boldsymbol{u}^T S\boldsymbol{u} &= 2\boldsymbol{e}^{(0)T}\mathbf{f}_{nl}(\boldsymbol{u}) && (5.21)\\ 2\boldsymbol{h}^T\boldsymbol{u} &= 2\boldsymbol{e}^{(0)T}\left(G\boldsymbol{u} + \mathbf{f}_{l}(\boldsymbol{u})\right) && (5.22)\\ q &= 0 && (5.23) \end{aligned}$$

In order to determine the matrix $S$ and the vector $\boldsymbol{h}$, the term $2\boldsymbol{e}^{(0)T}\mathbf{p}(\boldsymbol{u})$ in (5.7) is defined in matrix form as:

$$\begin{aligned} 2\boldsymbol{e}^{(0)T}\mathbf{p}(\boldsymbol{u}) &= \underbrace{2\,\boldsymbol{e}^{(0)T}}_{\boldsymbol{\Phi}=[\phi_1,\,\phi_2,\,\ldots,\,\phi_N]} \begin{bmatrix} \boldsymbol{u}^T T_1\boldsymbol{u} + 2\boldsymbol{l}_1^T\boldsymbol{u} + q_1 \\ \boldsymbol{u}^T T_2\boldsymbol{u} + 2\boldsymbol{l}_2^T\boldsymbol{u} + q_2 \\ \vdots \\ \boldsymbol{u}^T T_N\boldsymbol{u} + 2\boldsymbol{l}_N^T\boldsymbol{u} + q_N \end{bmatrix} \\ &= \sum_{i=1}^{N} \phi_i \left(\boldsymbol{u}^T T_i\boldsymbol{u} + 2\boldsymbol{l}_i^T\boldsymbol{u} + q_i\right) \end{aligned} \tag{5.24}$$

with $T_i = T_i^T$ for $i, \ldots, N$. Combining (5.7) and (5.24) the parameters $S$, $\boldsymbol{h}^T$ and $q$ are given by[1]:

$$S = \sum_{i=1}^{N} \phi_i T_i, \quad \boldsymbol{h}^T = \sum_{i=1}^{N} \phi_i \boldsymbol{l}_i^T, \quad q = \sum_{i=1}^{N} \phi_i q_i \tag{5.25}$$

To determine the matrix $S$ for the modified prediction model (5.16)-(5.17) based on a second order Volterra series model, the nonlinear term $\mathbf{f}_{nl}(\boldsymbol{u})$ is written in general form as:

$$\mathbf{f}_{nl}(\boldsymbol{u}) = \begin{bmatrix} \mathrm{f}_{nl,1}(\boldsymbol{u}) \\ \mathrm{f}_{nl,2}(\boldsymbol{u}) \\ \vdots \\ \mathrm{f}_{nl,N}(\boldsymbol{u}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}^T B_{nl,1} \boldsymbol{u} \\ \boldsymbol{u}^T B_{nl,2} \boldsymbol{u} \\ \vdots \\ \boldsymbol{u}^T B_{nl,N} \boldsymbol{u} \end{bmatrix} \tag{5.26}$$

with the matrices $B_{nl,i}$ given in the Appendix A.5. With (5.24) and (5.26) it results that $T_i = B_{nl,i}$. Hence, according to (5.25) the matrix $S$ is defined as:

$$S = \sum_{i=1}^{N} \phi_i B_{nl,i} \tag{5.27}$$

In order to determine the vector $\boldsymbol{h}$, the linear term $\mathbf{f}_l(\boldsymbol{u})$ is rewritten as:

$$\mathbf{f}_l(\boldsymbol{u}) = \begin{bmatrix} \mathrm{f}_{l,1}(\boldsymbol{u}) \\ \mathrm{f}_{l,2}(\boldsymbol{u}) \\ \vdots \\ \mathrm{f}_{l,N}(\boldsymbol{u}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_p^T B_{l,1} \\ \boldsymbol{u}_p^T B_{l,2} \\ \vdots \\ \boldsymbol{u}_p^T B_{l,N} \end{bmatrix} \boldsymbol{u} \tag{5.28}$$

with the matrices $B_{l,i}$ given in the Appendix A.5. The matrix $G$ can be expressed as:

$$G = \left[\begin{array}{c} G^{[1]} \\ \hline G^{[2]} \\ \hline \vdots \\ \hline G^{[N]} \end{array}\right] \tag{5.29}$$

where $G^{[i]}$ denotes the $i$-th row of the matrix $G$. Based on (5.28) and (5.29), the term $G\boldsymbol{u} + \mathbf{f}_l(\boldsymbol{u})$ in (5.22) can be given by:

$$G\boldsymbol{u} + \mathbf{f}_l(\boldsymbol{u}) = \begin{bmatrix} G^{[1]} + \boldsymbol{u}_p^T B_{l,1} \\ G^{[2]} + \boldsymbol{u}_p^T B_{l,2} \\ \vdots \\ G^{[N]} + \boldsymbol{u}_p^T B_{l,N} \end{bmatrix} \boldsymbol{u} \tag{5.30}$$

[1]Note that the parameter $q$ has already been defined for a second order Volterra series model by $q = 0$ (5.23). Nevertheless, (5.25) gives the definition of $q$ for a general second order model.

From (5.24) and (5.30) it is clear that $2\,l_i^T = G^{[i]} + \boldsymbol{u}_p^T B_{l,i}$ with $i = 1, \ldots, N$. Based on the definition (5.25), the parameter $\boldsymbol{h}$ can be defined in the form :

$$\boldsymbol{h}^T = \frac{1}{2}\sum_{i=1}^{N} \phi_i \left(G^{[i]} + \boldsymbol{u}_p^T B_{l,i}\right) \tag{5.31}$$

With $S$ (5.27), $\boldsymbol{h}^T$ (5.31) and $q$ (5.23) defined, the original cost function with a weighting of the control effort (5.12) based on a second order Volterra series model can be approximated by a convex function (5.9). By means of the design parameter $\lambda$, strict convexity of the approximated cost function can be guaranteed.

### 5.2.1 Control law based on the convexification

The convexified cost functions are used in an NMPC strategy based on a second order Volterra series model to compute the input sequence. The control law can be implemented as an iterative approach approximating the original cost function by a convex hull. The minimization of the convex hull results in a global optimization of the original cost considering the same weighting of the control effort as the convex cost functions.

In a first step an initial candidate input sequence is used to determine a convexified function approximating the cost based on a second order Volterra series model. Minimizing this convex cost function, a new candidate input sequence can be obtained. The new input sequence is then used to define a second convex function approximating the original cost. The pointwise maximum of the different convex cost functions represents a convex hull for the original cost function[2]. The global minimization of the convex hull results then in a new input sequence. It is clear that the difference between the original cost function and the convex hull is reduced with an increasing number of convexified functions. This procedure is repeated until the final accuracy tolerance is satisfied, i.e. the difference between the original cost and the convex hull falls below a certain level. Constraints in the input sequence and its increments can be included easily by $L_c\boldsymbol{u} \leq \boldsymbol{b}_c$ with $L_c \in \mathbb{R}^{n_c \times N_u}$ and $\boldsymbol{b}_c \in \mathbb{R}^{n_c}$ being $n_c$ the number of constraints.

The iterative procedure for the optimization based on the convexification can be expressed in the following form:

[2]The different cost functions are used in the $j$-th iteration with the maximum weighting factor $\lambda^{(j)}$ to assure convexity, see (5.14). Furthermore, the original cost function considering a weighting of the control effort is also used with the maximum weighting factor $\lambda^{(j)}$.

- Step 1: Define $j = 0$, $\boldsymbol{u}^{(0)} = \boldsymbol{u}_0$ and $\lambda^{(0)} = \varphi$ with $\varphi \geq 0$, calculate $\boldsymbol{b}$.
- Step 2: Set $j = j + 1$, calculate $\boldsymbol{e}^{(j-1)} = \mathbf{p}(\boldsymbol{u}^{(j-1)}) + \boldsymbol{b}$.
- Step 3: Determine $S^{(j-1)}$ (5.27), $\boldsymbol{h}^{(j-1)}$ (5.31) and $q^{(j-1)}$ (5.23) so that:

$$2\boldsymbol{e}^{(j-1)T}\mathbf{p}(\boldsymbol{u}) = \boldsymbol{u}^T S^{(j-1)}\boldsymbol{u} + 2\boldsymbol{h}^{(j-1)T}\boldsymbol{u} + q^{(j-1)} \tag{5.32}$$

- Step 4: Check if $S^{(j-1)} + \lambda^{(j-1)} L_u^T L_u > 0$. If yes, define $\lambda^{(j)} = \lambda^{(j-1)}$. If not, calculate a new weighting factor:

$$\lambda^{(j)} = -\min\left(\text{eig}\left((L_u^T)^{-1} S^{(j-1)} L_u^{-1}\right)\right) + \rho \tag{5.33}$$

  with $\rho > 0$.

- Step 5: $\hat{\boldsymbol{u}}^{(j)}$ is the minimizer of the convex hull, defined as the solution to the optimization problem:

$$\begin{aligned} &\min_{\hat{\boldsymbol{u}}^{(j)}, \alpha^{(j)}} \quad \alpha^{(j)} \\ &\text{s.t.} \quad \hat{\boldsymbol{u}}^{(j)T}\left(S^{(n)} + \lambda^{(j)} L_u^T L_u\right)\hat{\boldsymbol{u}}^{(j)} + 2\left(\boldsymbol{h}^{(n)T} - \lambda^{(j)}\boldsymbol{u}_l^T L_u\right)\hat{\boldsymbol{u}}^{(j)} + \\ &\qquad \ldots + q^{(n)} + 2\boldsymbol{e}^{(n)T}\boldsymbol{b} - \boldsymbol{e}^{(n)T}\boldsymbol{e}^{(n)} + \lambda^{(j)}\boldsymbol{u}_l^T\boldsymbol{u}_l \leq \alpha^{(j)} \\ &\qquad L_c\hat{\boldsymbol{u}}^{(j)} \leq \boldsymbol{b}_c \\ &\qquad n = 0, \ldots, j-1 \end{aligned} \tag{5.34}$$

- Step 6: Check if $\hat{\boldsymbol{u}}^{(j)} = \hat{\boldsymbol{u}}^{(j-1)}$. If yes, define $\boldsymbol{u} = \hat{\boldsymbol{u}}^{(j)}$ and apply $u(k|k)$ to the system. Otherwise check if $J(\hat{\boldsymbol{u}}^{(j)}, \lambda^{(j)}) \leq J(\boldsymbol{u}^{(j-1)}, \lambda^{(j-1)})$ is satisfied. If yes, set $\boldsymbol{u}^{(j)} = \hat{\boldsymbol{u}}^{(j)}$. If not, set $\hat{\boldsymbol{u}}^{(j)} = \boldsymbol{u}^{(j-1)} + \kappa\left(\hat{\boldsymbol{u}}^{(j)} - \boldsymbol{u}^{(j-1)}\right)$ and check the previous condition. Repeat the scaling of $\hat{\boldsymbol{u}}^{(j)}$ until the condition is satisfied.
- Step 7: Check if the final accuracy tolerance $J(\boldsymbol{u}^{(j)}, \lambda^{(j)}) - \alpha^{(j)} \leq \delta$ with $0 < \delta \ll 1$ is satisfied. If the convergence condition is satisfied, define $\boldsymbol{u} = \boldsymbol{u}^{(j)}$ and apply $u(k|k)$ to the system. Otherwise return to Step 2.

The choice of the scaling parameter $\kappa$ plays an important role, as the number of necessary iterations depends on its value. If a scaling of the input sequence is necessary, a very low value for $\kappa$ results in a new input sequence $\boldsymbol{u}^{(j)}$ close to $\boldsymbol{u}^{(j-1)}$. This leads to an elevated number of iterations as the minimization of the approximated cost functions is carried out in small steps. In simulations, the best results with respect to the number of iterations have been obtained with values of $0.8 \leq \kappa \leq 0.95$. Generally, the scaling parameter has to be chosen in the interval $0 < \kappa < 1$.

It has to be mentioned that the initial weighting factor can be chosen equal to $\lambda^{(0)} = 0$. In this case, the proposed approach calculates in every sampling period the smallest weighting factor $\lambda^{(j)}$ to guarantee strict convexity for all approximated cost functions. To obtain a less aggressive control behavior, a positive definite initial weighing factor $\lambda^{(j)} = \varphi$ with $\varphi > 0$ can be used. In this case, the procedure changes the weighting factor only in the case of a non-convex approximation of the cost function.

The input sequence calculated in the previous sampling period $k-1$ can be shifted by one element and used as the initial input candidate sequence $\boldsymbol{u}_0$ at $k$:

$$\boldsymbol{u}_0 = \begin{bmatrix} u(k|k-1) \\ u(k+1|k-1) \\ \vdots \\ u(k+N_u-3|k-1) \\ u(k+N_u-2|k-1) \\ u(k+N_u-2|k-1) \end{bmatrix} \tag{5.35}$$

In order to use the shifted solution of the previous sampling period as initial input candidate sequence, the computed input sequence has to be stored at the end of the optimization routine in order to be available in the next sampling period.

### 5.2.2 Convergence

The optimization procedure proposed in Section 5.2.1 guarantees convergence as the following two conditions are fulfilled:

- **C1:** the minimized approximated cost $\alpha^{(j)}$ increases monotonically in every iteration, i.e. $\alpha^{(j)} > \alpha^{(j-1)}$, if $\hat{\boldsymbol{u}}^{(j)} \neq \hat{\boldsymbol{u}}^{(j-1)}$ is true.
- **C2:** the original cost $J(\boldsymbol{u}^{(j)}, \lambda^{(j)})$ with the computed input sequence $\boldsymbol{u}^{(j)}$ is non-increasing with respect to the cost of the previous iteration, i.e. the costs satisfy $J(\boldsymbol{u}^{(j)}, \lambda^{(j)}) \leq J(\boldsymbol{u}^{(j-1)}, \lambda^{(j-1)})$.

**Proof of C1:** In every iteration a new constraint is added to the optimization problem (5.34). Hence, for $\hat{\boldsymbol{u}}^{(j)} \neq \hat{\boldsymbol{u}}^{(j-1)}$, i.e. the new constraint is active, the minimization results in a higher value of $\alpha^{(j)}$.

□

**Guarantee for C2:** Consider the candidate input sequence $\boldsymbol{u}^{(j-1)}$ computed in the previous iteration and the sequence $\hat{\boldsymbol{u}}^{(j)}$ being the solution of the optimization problem (see Step 5 of the proposed optimization approach) in the $j$-th iteration. If the original cost $J(\hat{\boldsymbol{u}}^{(j)}, \lambda^{(j)})$ based on the solution $\hat{\boldsymbol{u}}^{(j)}$ is larger than the cost $J(\boldsymbol{u}^{(j-1)}, \lambda^{(j-1)})$ based on the candidate input sequence $\boldsymbol{u}^{(j-1)}$, i.e.

$$J(\hat{\boldsymbol{u}}^{(j)}, \lambda^{(j)}) > J(\boldsymbol{u}^{(j-1)}, \lambda^{(j-1)}) \tag{5.36}$$

is true, the solution to the optimization problem is scaled in the form $\hat{\boldsymbol{u}}^{(j)} = \boldsymbol{u}^{(j-1)} + \kappa\,(\hat{\boldsymbol{u}}^{(j)} - \boldsymbol{u}^{(j-1)})$ with $0 < \kappa < 1$. This scaling of the input sequence is repeated until the condition $J(\hat{\boldsymbol{u}}^{(j)}, \lambda^{(j)}) \leq J(\boldsymbol{u}^{(j-1)}, \lambda^{(j-1)})$ is satisfied.

The cost function $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(j-1)}, \lambda^{(j)})$ is an approximation of the original cost function $J(\boldsymbol{u}, \lambda^{(j)})$ around $\boldsymbol{e}^{(j-1)} = \mathbf{p}(\boldsymbol{u}^{(j-1)}) + \boldsymbol{b}$. Therefore, the approximated and the original cost functions have in $\boldsymbol{u}^{(j-1)}$ the same value and the same gradient $\boldsymbol{\xi}(\boldsymbol{u}^{(j-1)})$. It is clear that the statement

$$\tilde{J}(\hat{\boldsymbol{u}}^{(j)}, \boldsymbol{e}^{(j-1)}, \lambda^{(j)}) < \tilde{J}(\boldsymbol{u}^{(j-1)}, \boldsymbol{e}^{(j-1)}, \lambda^{(j)}) \tag{5.37}$$

is true for $\hat{\boldsymbol{u}}^{(j)} \neq \boldsymbol{u}^{(j-1)}$ (otherwise the final accuracy will be satisfied). As a consequence of (5.37), the gradient $\boldsymbol{\xi}(\boldsymbol{u}^{(j-1)})$ is negative in direction of $\hat{\boldsymbol{u}}^{(j)}$. With a decreasing cost of $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(j-1)}, \lambda^{(j)})$ in direction of $\hat{\boldsymbol{u}}^{(j)}$, the original cost $J(\boldsymbol{u}, \lambda^{(j)})$ (with the same gradient in $\boldsymbol{u}^{(j-1)}$) also decreases around $\boldsymbol{e}^{(j-1)} = \mathbf{p}(\boldsymbol{u}^{(j-1)}) + \boldsymbol{b}$ in direction of $\hat{\boldsymbol{u}}^{(j)}$. With the scaling of the solution, the condition:

$$J(\hat{\boldsymbol{u}}^{(j)}, \lambda^{(j)}) \leq J(\boldsymbol{u}^{(j-1)}, \lambda^{(j-1)}) \tag{5.38}$$

will be satisfied at least in a small neighborhood around $\boldsymbol{u}^{(j-1)}$. Then, based on Step 6, the candidate input sequence is defined as $\boldsymbol{u}^{(j)} = \hat{\boldsymbol{u}}^{(j)}$ and the condition C2 with a non-increasing original cost:

$$J(\boldsymbol{u}^{(j)}, \lambda^{(j)}) \leq J(\boldsymbol{u}^{(j-1)}, \lambda^{(j-1)}) \tag{5.39}$$

is true.

□

In Fig. 5.2 a simple example for the used scaling is shown. The non-convex original cost $J(\boldsymbol{u}, \lambda^{(0)})$ is approximated by the convex function $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, \lambda^{(1)})$ around $\boldsymbol{e}^{(0)}$. Minimizing this convex function, the candidate input sequence $\hat{\boldsymbol{u}}^{(1)}$ is computed. It can be observed that the cost $J(\hat{\boldsymbol{u}}^{(1)}, \lambda^{(1)})$ of the non-convex function in $\hat{\boldsymbol{u}}^{(1)}$ is clearly bigger than $J(\boldsymbol{u}^{(0)}, \lambda^{(0)})$, i.e. $J(\hat{\boldsymbol{u}}^{(1)}, \lambda^{(1)}) > J(\boldsymbol{u}^{(0)}, \lambda^{(0)})$. Based on the scaling $\hat{\boldsymbol{u}}^{(1)} = \boldsymbol{u}^{(0)} + \kappa\,(\hat{\boldsymbol{u}}^{(1)} - \boldsymbol{u}^{(0)})$, the input sequence $\boldsymbol{u}^{(1)}$ is determined (see Step 6 of the proposed optimization approach). It can be seen that the new cost satisfies the condition $J(\boldsymbol{u}^{(1)}, \lambda^{(1)}) \leq J(\boldsymbol{u}^{(0)}, \lambda^{(0)})$. With the convex cost function $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, \lambda^{(1)})$ being an

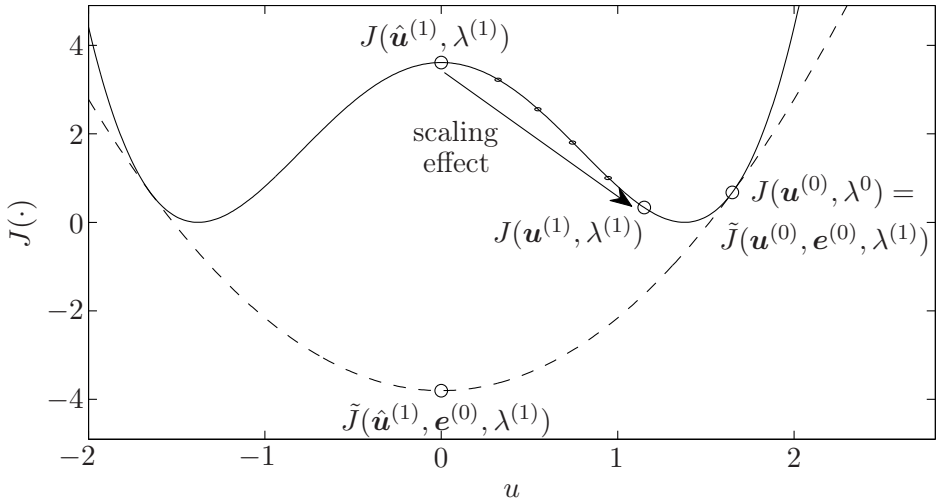


**Figure 5.2**: Example for scaling of the computed candidate input sequence with the original cost function $J(\cdot,\cdot)$ and the convex cost function $\tilde{J}(\cdot,\cdot,\cdot)$.

approximation of $J(\boldsymbol{u}, \lambda^{(0)})$, both functions have the same value and the same gradient in $\boldsymbol{u}^{(0)}$. It can be seen that the value of the convex function $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, \lambda^{(1)})$ decreases for input sequences moving from $\boldsymbol{u}^{(0)}$ to $\hat{\boldsymbol{u}}^{(1)}$, i.e. the convex function has a negative gradient in $\boldsymbol{u}^{(0)}$. With the convex function $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, \lambda^{(1)})$ and the non-convex function $J(\boldsymbol{u}, \lambda^{(0)})$ possessing the same gradient in $\boldsymbol{u}^{(0)}$, an input sequence $\boldsymbol{u}^{(1)}$ satisfying the condition of a non-increasing cost $J(\boldsymbol{u}^{(1)}, \lambda^{(1)}) \leq J(\boldsymbol{u}^{(0)}, \lambda^{(0)})$ can be always found (at least in a small neighborhood around $\boldsymbol{u}^{(0)}$). Note that in the shown example the weighting factor $\lambda^{(1)} = \lambda^{(0)}$ has been used and, as a consequence, $J(\boldsymbol{u}, \lambda^{(0)}) = J(\boldsymbol{u}, \lambda^{(1)})$ holds.

The conditions C1 and C2 guarantee that the difference between the original cost and the approximated cost decreases monotonically in every iteration. Therefore, convergence of the proposed optimization procedure is guaranteed. With the candidate input sequence converging to $\boldsymbol{u}^*$, the statement $\tilde{J}_\Sigma^{(j)}(\boldsymbol{u}^*, \lambda^{(j)}) = J(\boldsymbol{u}^*, \lambda^{(j)})$ holds (compare Proof of C1). Being $\tilde{J}_\Sigma^{(j)}(\boldsymbol{u}^*, \lambda^{(j)})$ the global minimum of the convex hull, $J(\boldsymbol{u}^*, \lambda^{(j)})$ represents the global minimum of the original cost function with the weighting factor $\lambda^{(j)}$. This means that the proposed procedure guarantees convergence and minimizes globally the original cost function with a weighting factor of $\lambda^{(j)}$.

## 5.3 Optimization based on convexification with guaranteed stability

This section explains the necessary modifications in the NMPC strategy based on a second order Volterra series model using a convexification approach in order to prove stability. After the definition of the general optimization problem in state-space representation, a feasible solution for the considered problem will be defined. Based on the feasible solution, it can be shown that the solution computed by the convexification algorithm guarantees input-to-state stability (ISS) [54, 67].

In a first step, the original cost function (5.2) and the approximated cost function (5.8) are extended to include a quadratic weighting term of the input sequence. In contrast to the weighting term $\lambda \boldsymbol{\Delta u}^T \boldsymbol{\Delta u}$ used in Section 5.2, the quadratic term $\lambda \left(\boldsymbol{u} - \boldsymbol{u}_r\right)^T \left(\boldsymbol{u} - \boldsymbol{u}_r\right)$ will be included in the cost functions. Hence, the original cost function becomes:

$$J(\boldsymbol{u}, \lambda) = J(\boldsymbol{u}) + \lambda \boldsymbol{u}^T \boldsymbol{u}^T - 2\lambda \boldsymbol{u}_r^T \boldsymbol{u} + \lambda \boldsymbol{u}_r^T \boldsymbol{u}_r \tag{5.40}$$

and the approximated cost function can be written as:

$$\begin{aligned} \tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, \lambda) \;\; = \;\; & \boldsymbol{u}^T (S + \lambda I) \boldsymbol{u} + 2(\boldsymbol{h}^T - \lambda \boldsymbol{u}_r^T) \boldsymbol{u} \\ & + q + 2\boldsymbol{e}^{(0)T} \boldsymbol{b} - \boldsymbol{e}^{(0)T} \boldsymbol{e}^{(0)} + \lambda \boldsymbol{u}_r^T \boldsymbol{u}_r \end{aligned} \tag{5.41}$$

For the approximated cost function (5.41) strict convexity ($S + \lambda I > 0$) can be assured easily by a suitable choice of the weighting parameter $\lambda$. If $S$ is positive definite, strict convexity is given for every $\lambda \geq 0$. Otherwise, the weighting factor $\lambda$ can be chosen to assure strict convexity in the following form:

$$\lambda = -\min\left(\mathrm{eig}\left(S\right)\right) + \rho \tag{5.42}$$

where $\rho$ represents a constant with $\rho > 0$. Analogously to (5.13), $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)})$ is a lower bound for $J(\boldsymbol{u}, \lambda)$ with

$$\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, \lambda) \leq J(\boldsymbol{u}, \lambda) \tag{5.43}$$

where the equality is true for $\boldsymbol{u} = \boldsymbol{u}^{(0)}$ and the inequality holds for every $\boldsymbol{u} \neq \boldsymbol{u}^{(0)}$. With the condition $\lambda^{(j)} \geq \lambda^{(j-1)}$, the pointwise maximum of several approximated cost functions defined by:

$$\tilde{J}_{\Sigma}^{(n)}(\boldsymbol{u}, \lambda^{(n)}) = \max_{n=1,\ldots,j} \left\{ \tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(j-1)}, \lambda^{(n)}) \right\} \tag{5.44}$$

can be considered as a convex hull of the cost function $J(\boldsymbol{u}, \lambda^{(n)})$ [76].

The shown convexification can now be used to compute the input sequence for an NMPC strategy based on a second order Volterra series prediction model. Using the same prediction model (5.16)-(5.17) as in Section 5.2, the approximated cost function can be expressed with $S$ (5.27), $\boldsymbol{h}^T$ (5.31) and $q$ (5.23) in the form of (5.41). Finally, in order to use the optimization approach proposed in Section 5.2.1, Step 4 and Step 5 have to be adjusted to the modified approximated cost function (5.41). Therefore, the convexity check in Step 4 has to be substituted by:

- Step 4: Check if $S^{(j-1)} + \lambda^{(j-1)} I > 0$. If yes, define $\lambda^{(j)} = \lambda^{(j-1)}$. If not, calculate a new weighting factor:

$$\lambda^{(j)} = -\min\left(\text{eig}\left(S^{(j-1)}\right)\right) + \rho \tag{5.45}$$

  with $\rho > 0$.

and the minimization of the convex hull in Step 5 has to be replaced by:

- Step 5: $\hat{\boldsymbol{u}}^{(j)}$ is the minimizer of the convex hull, defined as the solution to the optimization problem:

$$\begin{aligned} &\min_{\hat{\boldsymbol{u}}^{(j)}, \alpha^{(j)}} \quad \alpha^{(j)} \\ &\text{s.t.} \quad \hat{\boldsymbol{u}}^{(j)T}\left(S^{(n)} + \lambda^{(j)} I\right)\hat{\boldsymbol{u}}^{(j)} + 2\left(\boldsymbol{h}^{(n)T} - \lambda^{(j)}\boldsymbol{u}_r^T\right)\hat{\boldsymbol{u}}^{(j)} + \\ &\qquad \ldots + q^{(n)} + 2\boldsymbol{e}^{(n)T}\boldsymbol{b} - \boldsymbol{e}^{(n)T}\boldsymbol{e}^{(n)} + \lambda^{(j)}\boldsymbol{u}_r^T\boldsymbol{u}_r \leq \alpha^{(j)} \\ &\qquad L_c\hat{\boldsymbol{u}}^{(j)} \leq \boldsymbol{b}_c \\ &\qquad \hat{u}(k + N_u - 1|k) = u_r \\ &\qquad n = 0, \ldots, j-1 \end{aligned} \tag{5.46}$$

Based on the modified approximated cost function (5.41) and the equality constraint $\hat{u}(k + N_u - 1|k) = u_r$ and considering a prediction horizon of $N \geq N_t + N_u$, input-to-state stability can be guaranteed for the proposed modified optimization approach. The stability proof and an explanation for the necessary conditions will be given in the following sections. For the choice of $\boldsymbol{u}_0$, $\varphi$ and $\kappa$, the declarations made in Section 5.2.1 apply. Furthermore, the given proof of convergence in Section 5.2.2 is also valid for the modified optimization.

### 5.3.1 Robust stability proof

In order to prove robust stability of the new NMPC strategy, the considered optimization problem will be formulated in state space representation. In the state space, a simple feasible solution for the optimization problem will be determined. Finally, input-to-state stability can be proven by means of the convergence of the proposed NMPC strategy based on the modified convexification approach. It has to be mentioned that the stability proof is formulated in a general manner for NMPC strategies based on second order Volterra series prediction models considering a variable weighting factor $\lambda$. Hence, the stability proof also holds for other NMPC strategies based on the same characteristics (Volterra series prediction model, consideration of the estimation error, variable weighting factor $\lambda$ and quadratic cost function). The Appendix A.6 shows the proof of asymptotic stability for the proposed control strategy based on the nominal prediction model, i.e. without estimation error.

#### 5.3.1.1 Optimization problem in state-space representation

In general terms, the proposed NMPC strategy uses the convexification approach to minimize a cost function based on second order Volterra series model with the estimation error (2.61). The general optimization problem, for a finite prediction horizon $N$ and a finite control horizon $N_u$ with $N \geq N_u$, considers the cost function:

$$\begin{aligned} J(\boldsymbol{x}(k), \boldsymbol{u}(k), d(k), R^{(j)}) \;=\; & \sum_{i=0}^{N_u-1} L(\boldsymbol{x}(k+i|k), u(k+i|k), d(k), R^{(j)}) + \\ & \sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}(k+i|k), R^{(j)}) \end{aligned} \tag{5.47}$$

to be minimized. The vector $\boldsymbol{x}(k)$ denotes the initial state, $\boldsymbol{u}(k)$ is the sequence of $N_u$ future input values:

$$\boldsymbol{u}(k) = \; [u(k|k), \;\; u(k+1|k), \;\; \ldots, \;\; u(k+N_u-1|k)]^T \tag{5.48}$$

and $d(k)$ is the estimation error at $k$. The variable $R^{(j)} \in \mathbb{R}^{N_u \times N_u}$ denotes the weighting of the control effort calculated in the $j$-th iteration by the algorithm presented in Section 5.2.1 and can be defined as $R^{(j)} = \lambda^{(j)} I$. The term $L(\cdot,\cdot,\cdot,\cdot)$ is the stage cost considering the input $u(k+i|k)$ for $i = 0, \ldots, N_u - 1$ and $L_h(\cdot,\cdot)$ represents the stage cost based on the local control law $h(\boldsymbol{x}(k+i|k))$ for $i = N_u, \ldots, N-1$. The

optimization is subject to:

$$\begin{array}{ll} u(k+i|k) \in U, & i=0,\ldots,N_u-1 \\ h(\boldsymbol{x}(k+i|k)) \in U, & i=N_u,\ldots,N-1 \\ \boldsymbol{x}(k+i|k) \in X, & i=0,\ldots,N-1 \\ \boldsymbol{x}(k+N|k) \in \Omega(d(k)) & \end{array} \tag{5.49}$$

where the predicted output $y(k+N|k) = \mathrm{f}(\boldsymbol{x}(k+N|k)) + d(k)$ meets the desired reference $r(k)$ using the invariant terminal set:

$$\Omega(d(k)) = \{\boldsymbol{x} : \mathrm{f}(\boldsymbol{x}(k+N|k)) + d(k) = r(k)\} \tag{5.50}$$

The function $\mathrm{f} : \mathbb{R}^{N_t} \mapsto \mathbb{R}$, defined previously in (2.57), is a nonlinear function which maps the state vector to the output $y(k+i|k) = \mathrm{f}(\boldsymbol{x}(k+i|k)) + d(k)$ (2.61) and $r(k)$ denotes the reference of the system output. The predictions of the states $\boldsymbol{x}(k+i|k)$ are computed with the future input values and the local control law as:

$$\begin{array}{ll} \boldsymbol{x}(k+i+1|k) = \boldsymbol{\phi}(\boldsymbol{x}(k+i|k), u(k+i|k)), & i=0,\ldots,N_u-1 \\ \boldsymbol{x}(k+i+1|k) = \boldsymbol{\phi}(\boldsymbol{x}(k+i|k), h(\boldsymbol{x}(k+i|k))), & i=N_u,\ldots,N-1 \end{array} \tag{5.51}$$

The quadratic stage costs $L(\cdot,\cdot,\cdot,\cdot)$ and $L_h(\cdot,\cdot)$ used in the cost function (5.47) are defined in general as[3]:

$$\begin{array}{rcl} L(\boldsymbol{x}(k+i|k), u(k+i|k), d(k), R^{(j)}) & = & \|\mathrm{f}(\boldsymbol{x}(k+i|k)) + d(k) - r(k)\|_Q^2 + \\ & & \|u(k+i|k) - u_r(k)\|_{R^{(j)}}^2 \\ L_h(\boldsymbol{x}(k+i|k), R^{(j)}) & = & \|\mathrm{f}(\boldsymbol{x}(k+i|k)) + d(k) - r(k)\|_Q^2 + \\ & & \|h(\boldsymbol{x}(k+i|k)) - u_r(k)\|_{R^{(j)}}^2 \end{array} \tag{5.52}$$

where $u_r(k)$ represents the necessary steady-state input signal for a given reference $r(k)$.

Analogously to the iterative control strategy presented in Section 4.4.1.1 a necessary condition to proof stability is that the output prediction for $k+N$ made at $k$ satisfies $y(k+N|k) = r(k)$. With the nominal output prediction $\tilde{y}(k+i|k)$, the output prediction can be defined with (4.41):

$$y(k+i|k) = \tilde{y}(k+i|k) + d(k) \tag{5.53}$$

Now, substituting in (5.53) the predicted output $y(k+N|k)$ for $k+N$ made at $k$ by the reference $r(k)$, the nominal predicted output can be written as:

$$\tilde{y}(k+N|k) = r(k) - d(k) \tag{5.54}$$

[3]For the sake of simplicity the notation $L_h(\boldsymbol{x}(k+i|k), R^{(j)}) = L(\boldsymbol{x}(k+i|k), h(\boldsymbol{x}(k+i|k)), d(k), R^{(j)})$ has been chosen.

Under the assumption that the nominal predicted output $\tilde{y}(k+N|k)$ reaches steady state in $k+N$ or before, the necessary steady-state input $u_r(k)$ can be defined as[4]:

$$\begin{aligned} u_r(k) &= \chi^{-1}(\tilde{y}(k+N|k)) \\ &= \chi^{-1}(r(k)-d(k)) \end{aligned} \tag{5.55}$$

being $\chi : \mathbb{R} \mapsto \mathbb{R}$ a static nonlinearity mapping the steady-state input to the nominal steady-state output prediction. The steady-state input $u_r(k)$ is also used to define the local control law $h(\boldsymbol{x}(k+i|k))$ used in the stage cost (5.52):

$$h(\boldsymbol{x}(k+i|k)) = u_r(k), \quad i = N_u, \ldots, N-1 \tag{5.56}$$

#### 5.3.1.2 Feasibility of the solution

Consider the solution:

$$\boldsymbol{u}^s(k) = \; [u^s(k|k), \;\; u^s(k+1|k), \;\; \ldots, \;\; u^s(k+N_u-1|k)]^T \tag{5.57}$$

calculated at $k$ with the optimization algorithm presented in Section 5.3. The control sequence $\boldsymbol{u}^s(k)$ and the local control law, defined in (5.56) as the steady-state input $u_r(k)$, lead to the predicted states $\boldsymbol{x}^s(k+i|k)$ for $i = 1, \ldots, N$. Based on the optimal or suboptimal sequence $\boldsymbol{u}^s(k)$ the shifted solution:

$$\boldsymbol{u}^f(k+1) = \; [u^f(k+1|k+1), \;\; u^f(k+2|k+1), \;\; \ldots, \;\; u^f(k+N_u|k+1)]^T \tag{5.58}$$

with the elements:

$$u^f(k+i|k+1) = \begin{cases} u^s(k+i|k) & \text{for } i = 1, \ldots N_u - 1 \\ h(\boldsymbol{x}^f(k+N_u|k+1)) & \text{for } i = N_u \end{cases} \tag{5.59}$$

can be obtained. Note that the input sequence $\boldsymbol{u}^f(k+1)$ corresponds to the shifted sequence $\boldsymbol{u}^s(k)$ plus the additional term of the local control law. With the shifted solution $\boldsymbol{u}^f(k+1)$ and the local control law $h(\boldsymbol{x}^f(k+i|k+1)) = u_r(k+1)$ for $i = N_u+1, \ldots, N$ the predicted states $\boldsymbol{x}^f(k+i|k+1)$ for $i = 2, \ldots, N+1$ are obtained.

Consider the predictions $\boldsymbol{x}^s(k+i|k)$ for $i = 1, \ldots N_u$ obtained with the optimal or suboptimal sequence $\boldsymbol{u}^s(k)$ as well as the predicted states $\boldsymbol{x}^s(k+i|k+1)$ for $i = N_u+1, \ldots N$ based on the local control law $h(\boldsymbol{x}(k+i|k))$. With the vector $\boldsymbol{u}^s(k)$ and the local control law $h(\boldsymbol{x}(k+i|k))$ $i = N_u, \ldots, N-1$ computed on basis of the conditions (5.49), the predicted state vector for $k+N$ satisfies $\boldsymbol{x}^s(k+N|k) \in \Omega(d(k))$.

[4]Note that the steady-state input depends on the given reference $r(k)$ and the known estimation error $d(k)$. For the sake of simplicity, the notation $u_r(k) = u_r(r(k), d(k))$ has been chosen. Furthermore it has to be mentioned that a constant reference has been assumed, i.e. $r(k) = r(k+1)$.

Note that the prediction for $k+1$ made at $k$ satisfies $\boldsymbol{x}^s(k+1|k) = \boldsymbol{x}(k+1)$ as the estimation error considered in the Volterra based prediction model has no influence on the states. Hence, with the initial state $\boldsymbol{x}^f(k+1|k+1) = \boldsymbol{x}^s(k+1|k)$ and $u^f(k+i|k+1) = u^s(k+i|k)$ for $i = 1, \dots N_u - 1$, the predictions made at $k+1$ satisfy $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^s(k+i|k)$ for $i = 2, \dots N_u$. The predictions $\boldsymbol{x}^f(k+i|k+1)$ for $i = N_u+1, \dots, N+1$ are computed with the local control law $h(\boldsymbol{x}(k+i|k+1)) = u_r(k+1)$ for $i = N_u, \dots, N$. With the shifted solution $\boldsymbol{u}^f(k+1)$, based on the sequence $\boldsymbol{u}^s(k)$, and the local control law $h(\boldsymbol{x}(k+i|k+1))$ satisfying the specified conditions of (5.49), the statement $\boldsymbol{x}^f(k+N+1|k+1) \in \Omega(d(k+1))$ is true and, as a consequence, the solution $\boldsymbol{u}^f(k+1)$ to the optimization problem is feasible.

#### 5.3.1.3 Convergence

Consider the cost $J_0^s(\boldsymbol{x}(k)) = J(\boldsymbol{x}(k), \boldsymbol{u}^s(k), d(k), R_0)$ at $k$ based on the optimal or suboptimal solution $\boldsymbol{u}^s(k)$, the local control law $h(\boldsymbol{x}^s(k+i|k))$, the estimation error $d(k)$ and the initial weighting $R_0$ of the convexification approach. Furthermore, consider the cost $J_0^f(\boldsymbol{x}(k+1)) = J(\boldsymbol{x}(k+1), \boldsymbol{u}^f(k+1), d(k+1), R_0)$ at $k+1$ calculated with the feasible solution $\boldsymbol{u}^f(k)$, the local control law $h(\boldsymbol{x}^f(k+i|k+1))$, the estimation error $d(k+1)$ and the initial weighting $R_0$. The convergence of the control strategy based on the proposed optimization algorithm based on convexification and subject to the conditions given in (5.49) can be guaranteed if the costs are monotonically decreasing.

The cost $J_0^s(\boldsymbol{x}(k))$ at $k$ associated to the optimal or suboptimal solution $\boldsymbol{u}^s(k)$ is defined with (5.47) as:

$$\begin{aligned} J_0^s(\boldsymbol{x}(k)) \;=\; & \sum_{i=0}^{N_u-1} L(\boldsymbol{x}^s(k+i|k), u^s(k+i|k), d(k), R_0) + \\ & \sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}^s(k+i|k), R_0) \end{aligned} \tag{5.60}$$

and the cost $J_0^f(\boldsymbol{x}(k+1))$ at $k+1$ has the form:

$$\begin{aligned} J_0^f(\boldsymbol{x}(k+1)) \;&=\; \sum_{i=1}^{N_u} L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1), d(k+1), R_0) + \\ &=\; \sum_{i=N_u+1}^{N} L_h(\boldsymbol{x}^f(k+i|k+1), R_0) \end{aligned} \tag{5.61}$$

With $u^f(k+N_u|k+1) = h(\boldsymbol{x}^f(k+N_u|k+1))$ the stage cost for $k+N_u$ based on the feasible solution can be expressed as $L(\boldsymbol{x}^f(k+N_u|k+1), u^f(k+N_u|k+1), d(k+1), R_0) =$

$L_h(\boldsymbol{x}^f(k+N_u|k+1),R_0)$. Then, the difference $\Delta J_0(k+1) = J_0^f(\boldsymbol{x}(k+1)) - J_0^s(\boldsymbol{x}(k))$ of the costs can be written generally as:

$$\begin{aligned}\Delta J_0(k+1) &= L_h(\boldsymbol{x}^f(k+N|k+1),R_0) - L(\boldsymbol{x}^s(k|k),u^s(k|k),d(k),R_0)\\ &\quad+\sum_{i=1}^{N_u-1}\Big(L(\boldsymbol{x}^f(k+i|k+1),u^f(k+i|k+1),d(k+1),R_0)\\ &\qquad - L(\boldsymbol{x}^s(k+i|k),u^s(k+i|k),d(k),R_0)\Big)\\ &\quad+\sum_{i=N_u}^{N-1}\Big(L_h(\boldsymbol{x}^f(k+i|k+1),R_0) - L_h(\boldsymbol{x}^s(k+i|k+1),R_0)\Big)\end{aligned} \tag{5.62}$$

Consider a prediction horizon $N \geq N_u + N_t$ and the nilpotent character $A^{N_t} = 0$ of the state matrix of the used prediction model (2.61). In $k+1$, the last $N_t$ input values correspond to the local control law $h(\boldsymbol{x}(k+i|k+1)) = u_r(k+1)$ for $i = N_u, \ldots, N$. Due to $A^{N_t} = 0$ the prediction $\boldsymbol{x}^f(k+N|k+1)$ reaches steady state. Furthermore, based on the definition of the steady-state input (5.55) the nominal output for $k+N$ predicted in $k+1$ is given by $\tilde{y}(k+N|k+1) = r(k+1) - d(k+1)$. With the definition (4.42) of the nominal output $\tilde{y}(k+i|k) = \mathrm{f}(\boldsymbol{x}(k+i|k))$ the relation $\mathrm{f}(\boldsymbol{x}(k+N|k+1)) = r(k+1) - d(k+1)$ can be obtained. Finally, with the mentioned relation and the local control law $h(\boldsymbol{x}^f(k+N|k+1)) = u_r(k+1)$ it can be shown that the stage cost for $k+N$ computed at $k+1$ is $L_h(\boldsymbol{x}^f(k+N|k),R_0) = 0$. Then, the difference $\Delta J_0(k+1)$ of the cost functions can be expressed:

$$\Delta J_0(k+1) = -L(\boldsymbol{x}^s(k|k),u^s(k|k),d(k),R_0) + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \tag{5.63}$$

with the terms

$$\begin{aligned}\alpha_1 &= \sum_{i=1}^{N_u-1}\Big(\|\mathrm{f}(\boldsymbol{x}^f(k+i|k+1)) + d(k+1) - r(k+1)\|_Q^2 -\\ &\qquad \|\mathrm{f}(\boldsymbol{x}^s(k+i|k)) + d(k) - r(k)\|_Q^2\Big)\end{aligned} \tag{5.64}$$

$$\begin{aligned}\alpha_2 &= \sum_{i=1}^{N_u-1}\Big(\|u^f(k+i|k+1) - u_r(k+1)\|_{R_0}^2 -\\ &\qquad \|u^s(k+i|k) - u_r(k)\|_{R_0}^2\Big)\end{aligned} \tag{5.65}$$

$$\begin{aligned}\alpha_3 &= \sum_{i=N_u}^{N-1}\Big(\|\mathrm{f}(\boldsymbol{x}^f(k+i|k+1)) + d(k+1) - r(k+1)\|_Q^2 -\\ &\qquad \|\mathrm{f}(\boldsymbol{x}^s(k+i|k)) + d(k) - r(k)\|_Q^2\Big)\end{aligned} \tag{5.66}$$

$$\alpha_4 = \sum_{i=N_u}^{N-1} \Big( \|h(\boldsymbol{x}^f(k+i|k+1)) - u_r(k+1)\|_{R_0}^2 - \|h(\boldsymbol{x}^s(k+i|k)) - u_r(k)\|_{R_0}^2 \Big) \tag{5.67}$$

The Lemmas 4.1-4.4 presented in Chapter 4 will be used to define upper bounds for the terms $\alpha_1$, $\alpha_2$, $\alpha_3$ and $\alpha_4$ used in the difference of the cost functions given in (5.63).

**Term $\boldsymbol{\alpha_1}$:** With $\boldsymbol{x}^s(k+1|k) = \boldsymbol{x}(k+1)$ and the input values of the feasible solution defined as $u^f(k+i|k+1) = u^s(k+i|k)$ for $i = 1, \dots N_u - 1$, the predicted states satisfy $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^s(k+i|k)$ for $i = 1, \dots, N_u - 1$. The reference is assumed to be constant, i.e. $r(k) = r(k+1)$, and the estimation error increment is given generally as $\Delta d = d(k+1) - d(k)$. Defining the auxiliary variable $z(k+i|k) = \mathrm{f}(\boldsymbol{x}^s(k+i|k)) + d(k) - r(k)$ the term $\alpha_1$ can be written as:

$$\alpha_1 = \sum_{i=1}^{N_u-1} \|z(k+i|k) + \Delta d\|_Q^2 - \|z(k+i|k)\|_Q^2 \tag{5.68}$$

Now, using Lemma 4.1 on (5.68) the term $\alpha_1$ can be bounded by:

$$\alpha_1 \leq c_1(Q, N_u) \cdot \|\Delta d\| \tag{5.69}$$

where the upper bound depends on the estimation error increment $\Delta d$ and the constant $c_1(\cdot,\cdot)$.

□

**Term $\boldsymbol{\alpha_2}$:** According to the definition (5.59), the elements of the feasible solution satisfy $u^f(k+i|k+1) = u^s(k+i|k)$ for $i = 1, \dots, N_u - 1$. With the steady-state input increment defined as $\Delta u_r = u_r(k+1) - u_r(k)$ and the auxiliary variable $z_1(k+i|k) = u^s(k+i|k) - u_r(k)$ the term $\alpha_2$ can be expressed as:

$$\alpha_2 = \sum_{i=1}^{N_u-1} \|z_1(k+i|k) - \Delta u_r\|_{R_0}^2 - \|z_1(k+i|k)\|_{R_0}^2 \tag{5.70}$$

Applying Lemma 4.1 to (5.70), the term $\alpha_2$ is bounded by:

$$\alpha_2 \leq c_2(R_0, N_u) \cdot \|\Delta u_r\| \tag{5.71}$$

and depends on a constant $c_2(\cdot,\cdot)$ and the steady-state input increment input $\Delta u_r$. Based on the definition (5.55), the steady-state input increment input can be expressed

as $\Delta u_r = \chi^{-1}(r(k+1) - d(k+1)) - \chi^{-1}(r(k) - d(k))$. With the auxiliary variable $z_2 = r(k) - d(k)$, the constant reference $r(k+1) = r(k)$ and the estimation error increment $\Delta d = d(k+1) - d(k)$, the necessary steady-state input increment can be rewritten as $\Delta u_r = \chi^{-1}(z_2 - \Delta d) - \chi^{-1}(z_2)$. Applying Lemma 4.2 to the norm of the necessary steady-state input increment, the bound $\|\Delta u_r\| \leq L_\chi \|\Delta d\|$ is obtained. Finally, using the expression for $\|\Delta u_r\|$ in (5.71), the term $\alpha_2$ is bounded by:

$$\alpha_2 \leq c_2(R_0, L_\chi, N_u) \cdot \|\Delta d\| \tag{5.72}$$

where the constant $c_2(\cdot,\cdot,\cdot)$ depends on the initial weighting factor $R_0$, the parameter $L_\chi$ and the control horizon $N_u$.

□

**Term $\alpha_3$:** Consider a constant reference $r(k+1) = r(k)$, the estimation error increment $\Delta d = d(k+1) - d(k)$ and the difference between the states for $i = N_u, \ldots, N-1$ given by $\Delta \boldsymbol{x}(k+i) = \boldsymbol{x}^f(k+i|k+1) - \boldsymbol{x}^s(k+i|k)$. With the auxiliary variables $z_1(k+i|k) = \mathrm{f}(\boldsymbol{x}^s(k+i|k)) + d(k) - r(k)$ and $z_2(k+i|k) = \mathrm{f}(\boldsymbol{x}^s(k+i|k) + \Delta \boldsymbol{x}(k+i)) - \mathrm{f}(\boldsymbol{x}^s(k+i|k)) + \Delta d$ the term term $\alpha_3$ can be expressed as:

$$\alpha_3 = \sum_{i=N_u}^{N-1} \|z_1(k+i|k) + z_2(k+i|k)\|_Q^2 - \|z_1(k+i|k)\|_Q^2 \tag{5.73}$$

Then, with Lemma 4.1 the term $\alpha_3$ can be bounded by:

$$\alpha_3 \leq c_3(Q, N, N_u) \cdot \|z_2(k+i|k)\| \tag{5.74}$$

Now, using Lemma 4.3 the auxiliary variable can be bounded by $\|z_2(k+i|k)\| \leq \|\mathrm{f}(\boldsymbol{x}^s(k+i|k)) + \Delta \boldsymbol{x}(k+i) - \mathrm{f}(\boldsymbol{x}^s(k+i|k))\| + \|\Delta d\| \leq L_f \|\Delta \boldsymbol{x}(k+i)\| + \|\Delta d\|$. Then, the upper bound (5.74) of the term $\alpha_3$ can be rewritten as:

$$\alpha_3 \leq c_3(Q, N, N_u) \cdot (L_f \|\Delta \boldsymbol{x}(k+i)\| + \|\Delta d\|) \tag{5.75}$$

Furthermore, applying Lemma 4.4 to (5.75) the upper bound of $\alpha_3$ becomes:

$$\alpha_3 \leq c_3(Q, N, N_u) \cdot (L_f \|c_{\boldsymbol{x}}\, \Delta u_r\| + \|\Delta d\|) \tag{5.76}$$

With Lemma 4.2 the necessary steady-state input increment can be bounded by $\|\Delta u_r\| \leq L_\chi \|\Delta d\|$ (see explanation for Term $\alpha_2$). Hence, (5.76) can be modified and the upper bound of $\alpha_3$ is defined by:

$$\alpha_3 \leq c_3(Q, L_\chi, L_f, c_{\boldsymbol{x}}, N, N_u) \cdot \|\Delta d\| \tag{5.77}$$

depending only on the constant $c_3(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$ and the norm of the increment $\|\Delta d\|$.

□

**Term $\boldsymbol{\alpha_4}$:** The term $\alpha_4$ is given by:

$$\alpha_4 = 0 \tag{5.78}$$

as the local control law (5.56) is defined with the steady-state input.

□

Finally, with the upper bounds for $\alpha_1$ (5.69), $\alpha_2$ (5.72) and $\alpha_3$ (5.77) as well as the term $\alpha_4 = 0$ (5.78), the difference $\Delta J_0(k+1) = J_0^f(\boldsymbol{x}(k+1)) - J_0^s(\boldsymbol{x}(k))$ (5.63) between the costs associated to the feasible solution and the optimal or suboptimal solution is bounded for $N \geq N_u + N_t$ by:

$$J_0^f(\boldsymbol{x}(k+1)) - J_0^s(\boldsymbol{x}(k)) \leq -L(\boldsymbol{x}(k|k)^s, u^s(k|k), d(k), R_0) + c_V \cdot \|\Delta d\| \tag{5.79}$$

with the constant parameter $c_V$ given by:

$$c_V = c_1(Q, N_u) + c_2(R_0, L_\chi, N_u) + c_3(Q, L_\chi, L_f, c_{\boldsymbol{x}}, N, N_u) \tag{5.80}$$

Note that $c_V \cdot \|\Delta d\| > 0$ and $-L(\boldsymbol{x}^s(k|k), u^s(k|k), d(k), R_0) \leq 0$, thus it is ensured that the worst case cost will decrease as long as $-L(\boldsymbol{x}^s(k|k), u^s(k|k), d(k), R_0) > c_V \cdot \|\Delta d\|$. It is clear that the system is steered in the set:

$$\Psi_d = \{\boldsymbol{x} \colon L(\boldsymbol{x}^s(k|k), u^s(k|k), d(k), R_0) \leq c_V \cdot \|\Delta d\|\} \tag{5.81}$$

from any arbitrary $\boldsymbol{x}$. However, when the state enters the set $\Psi_d$ it may remain inside or evolve out of it as the decrease of the cost is not guaranteed inside the set. With the quadratic stage cost always satisfying $-L(\boldsymbol{x}^s(k|k), u^s(k|k), d(k), R_0) \leq 0$ and (5.79) follows that:

$$J_0^f(\boldsymbol{x}(k+1)) \leq J_0^s(\boldsymbol{x}(k)) + c_V \cdot \|\Delta d\| \tag{5.82}$$

For any $\boldsymbol{x}(k) \in \Psi_d$ the inequality:

$$J_0^s(\boldsymbol{x}(k)) + c_V \cdot \|\Delta d\| \leq \max_{\boldsymbol{x} \in \Psi_d} J_0^s(\boldsymbol{x}) + c_V \cdot \|\Delta d\| = \beta_d \tag{5.83}$$

holds. Then, taking into account (5.82) and (5.83) follows that:

$$J_0^f(\boldsymbol{x}(k+1)) \leq \beta_d, \quad \forall\ \boldsymbol{x}(k) \in \Psi_d \tag{5.84}$$

Whenever the state enters into $\Psi_d$, it evolves into the set:

$$\Psi_\beta = \{\boldsymbol{x} \colon J_0^f(\boldsymbol{x}) \leq \beta_d\} \tag{5.85}$$

Hence, once in the set $\Psi_d$ the state may evolve outside of $\Psi_d$, but it will remain inside $\Psi_\beta$. Thus, the state is ultimately bounded and the system is stabilized using the feasible solution.

Now, with the inequality (5.84) and the set (5.85), input-to-state stability for the NMPC based on the convexification algorithm, can be proven. Therefore, consider the optimization approach (see Section 5.3) initialized in $k+1$ with the feasible solution, i.e. $\boldsymbol{u}^{(0)}(k+1) = \boldsymbol{u}^{f}(k+1)$, and the initial weighting matrix $R^{(0)} = R_0 = \varphi^{(0)}\boldsymbol{I}$ where $\varphi^{(0)} \geq 0$. Then, in every iteration a new candidate input sequence $\boldsymbol{u}^{(j)}(k+1)$ and a new weighting matrix $R^{(j)} = \varphi^{(j)}\boldsymbol{I}$ with $\varphi^{(j)} \geq \varphi^{(j-1)}$ are calculated. Furthermore, consider the cost $J^{(j)}(\boldsymbol{x}(k+1)) = J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), d(k+1), R^{(j)})$ associated to the candidate input sequence $\boldsymbol{u}^{(j)}(k+1)$ and the weighting matrix $R^{(j)}$. From the proof of the condition C2 (see Section 5.2.2) it can be shown that the statement:

$$J^{(j)}(\boldsymbol{x}(k+1)) \leq J_0^{f}(\boldsymbol{x}(k+1)) \tag{5.86}$$

holds. Furthermore, with the weighting matrix $R^{(j)} \geq R^{(j-1)}$ it is clear that

$$J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), d(k+1), R^{(0)}) \leq J^{(j)}(\boldsymbol{x}(k+1)) \tag{5.87}$$

is always fulfilled. Combining (5.86) and (5.87) it can be shown that for every $\boldsymbol{u}^{(j)}(k+1)$ the inequality:

$$J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), d(k+1), R^{(0)}) \leq J_0^{f}(\boldsymbol{x}(k+1)) \tag{5.88}$$

holds. Now, from the inequality (5.88), it is clear that the cost $J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), d(k+1), R^{(0)})$ based on the solution $\boldsymbol{u}^{(j)}(k+1)$ computed at the $j$-th iteration is always lower or equal to the cost $J^{f}(\boldsymbol{x}(k+1))$ associated to the feasible solution $\boldsymbol{u}^{f}(k+1)$. With (5.84), (5.85) and (5.88) it is straightforward to show that the system controlled by the NMPC based on the convexification approach is ultimately bounded by the set $\Psi_\beta$. Hence, under consideration of the conditions given in (5.49) and a prediction horizon $N \geq N_u + N_t$, the NMPC strategy based on the optimization problem using the convexification approach for second order Volterra series models (see Section 5.3) possesses input-to-state stability. Once inside the set $\Psi_\beta$, the control strategy maintains the system inside the set.

**Remark 5.1** *Note that the performance index $J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), R^{(j)}, d(k+1))$ decreases monotonically in every iteration (see proof of C2 in Section 5.2.2) and, as a consequence, the cost $J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), d(k+1), R^{(0)})$ also decreases monotonically. Therefore, a higher optimality of the computed solution leads to a higher convergence of the proposed NMPC. Hence, the proposed algorithm can be stopped in an arbitrary iteration guaranteeing always input-to-state stability.*

**Remark 5.2** *Note that with a horizon of $N = N_u + N_t$ not only the output prediction $y(k+N|k+1) = r(k+1)$ reaches steady-state, but also $y(k+N+1|k+1) = r(k+1)$.*

*The same applies to the output prediction for $k+N$ made at $k$, i.e. $y(k+N|k) = r(k)$. Therefore, without an error in the output predictions for $k+N$ made at $k$ and for $k+N+1$ made at $k+1$, a terminal cost in the initial optimization problem (5.47) is not required.*

**Remark 5.3** *Furthermore, it has to be mentioned that the equality constraint $\hat{u}(k+N_u-1|k) = u_r$ in (5.46) is not necessary to assure stability of the proposed control strategy. However, with the mentioned equality constraint, the matrices and vectors defined in the Appendix A.1 can be used without any modification. Without considering the equality constraint, the prediction model (2.43)-(2.44) has to be adjusted to consider the local control law $h(\boldsymbol{x}(k+i|k)) = u_r$ for $i = N_u, \ldots, N-1$.*

## 5.4 Experimental results

The two proposed NMPC strategies (Sections 5.2.1 and 5.3) using a convexification of the cost function based on a second order Volterra series models were tested in experiments. In the following sections the behavior of the proposed control strategies will be illustrated by experimental results.

### 5.4.1 Convexified optimization

The NMPC strategy based on the convexification of the cost function presented in Section 5.2.1 was validated in experiments. The experiments, including setpoint tracking and disturbance rejection, were carried out with the pilot plant emulating an exothermic chemical reaction (see Section 3.1). Analogously to the experiments in Section 4.5.1.1, the proposed control strategy was implemented in the Matlab/Simulink environment. Furthermore, with a prediction horizon of $N = 25$ and a control horizon of $N_u = 15$ the same values as in Section 4.5.1.1 have been used for the proposed control strategy. For the final accuracy a value of $\varepsilon = 10^{-3}$ has been chosen and the constraints:

$$\begin{array}{rcccl} 5 & \leq & u(k+i|k) & \leq & 100, \quad i = 0, \ldots, 14 \\ -20 & \leq & \Delta u(k+i|k) & \leq & 20, \quad i = 0, \ldots, 14 \end{array} \tag{5.89}$$

have been considered. The proposed control strategy was tested with the same reference tracking and disturbance rejection experiments as done before with the iterative control strategies in Sections 4.5.1.1 and 4.5.2.

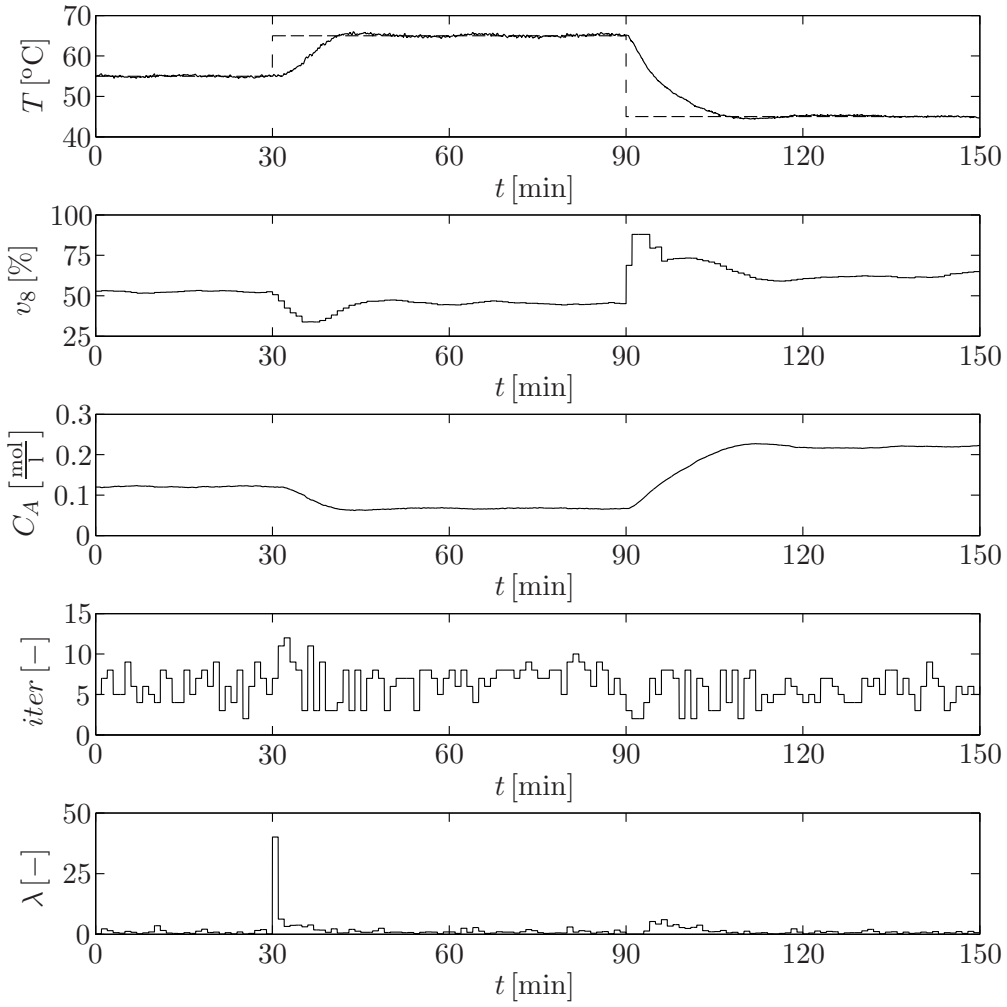


**Figure 5.3**: Setpoint tracking experiment controlled by the proposed NMPC strategy based on the convexification approach. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, necessary iterations *iter* and weighting factor $\lambda$.

The results of the setpoint tracking experiment are shown in Fig. 5.3. Starting with a reference of 55 °C the setpoint is first changed to 65 °C and later to 45 °C. After the first setpoint change in $t = 30$ min an overshoot of approximately 1 °C can be observed, the second setpoint change in $t = 90$ min results in an overshoot of $-0.7$ °C. In both cases, when the system reaches steady state after approximately 25 minutes, the control signal shows only insignificant changes. Generally, the shown results are very similar to the ones obtained with the constrained iterative control strategy (see Fig. 4.7).

In the second experiment, the disturbance rejection capabilities of the proposed controller were proven by means of a constant error in the parameter $E$ of the emulated

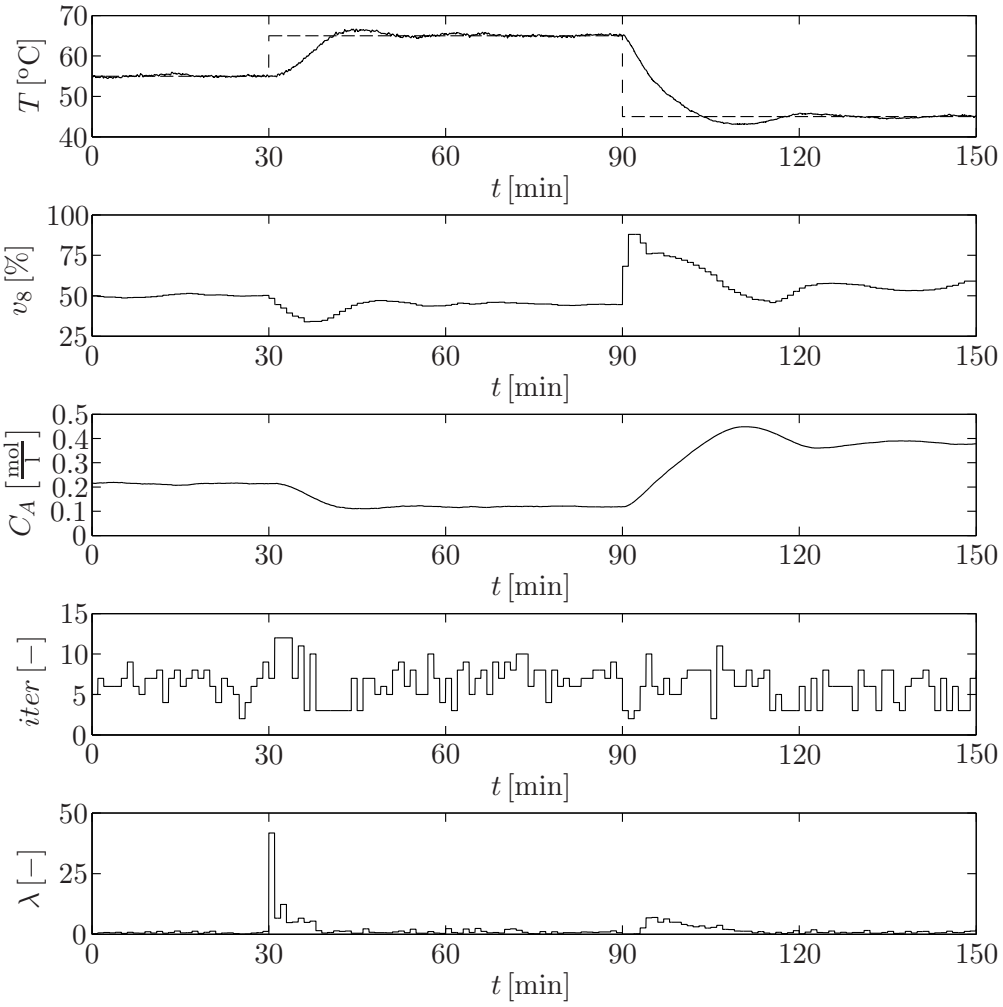


**Figure 5.4**: Disturbance rejection experiment (persistent disturbance in the emulated chemical reaction) controlled by the proposed NMPC strategy based on the convexification approach. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, necessary iterations *iter*, and weighting factor $\lambda$.

exothermic chemical reaction model and by changes in the given setpoint (see Fig. 5.4). As the parameter $E$ has a strong influence on the dynamic behavior of the emulated reaction, the introduced error corresponds only to 3 % of the original value of $E$. After the changes in the setpoint, the system shows an overshoot of approximately 1.6 °C and −2 °C. As can be observed, the overshoot after the second setpoint increased considerably with respect to the results shown in Fig. 5.3. In spite of the introduced error in the parameter $E$, the proposed control strategy stabilizes the system around the given reference and shows only negligible changes in the control action for the system in steady state.

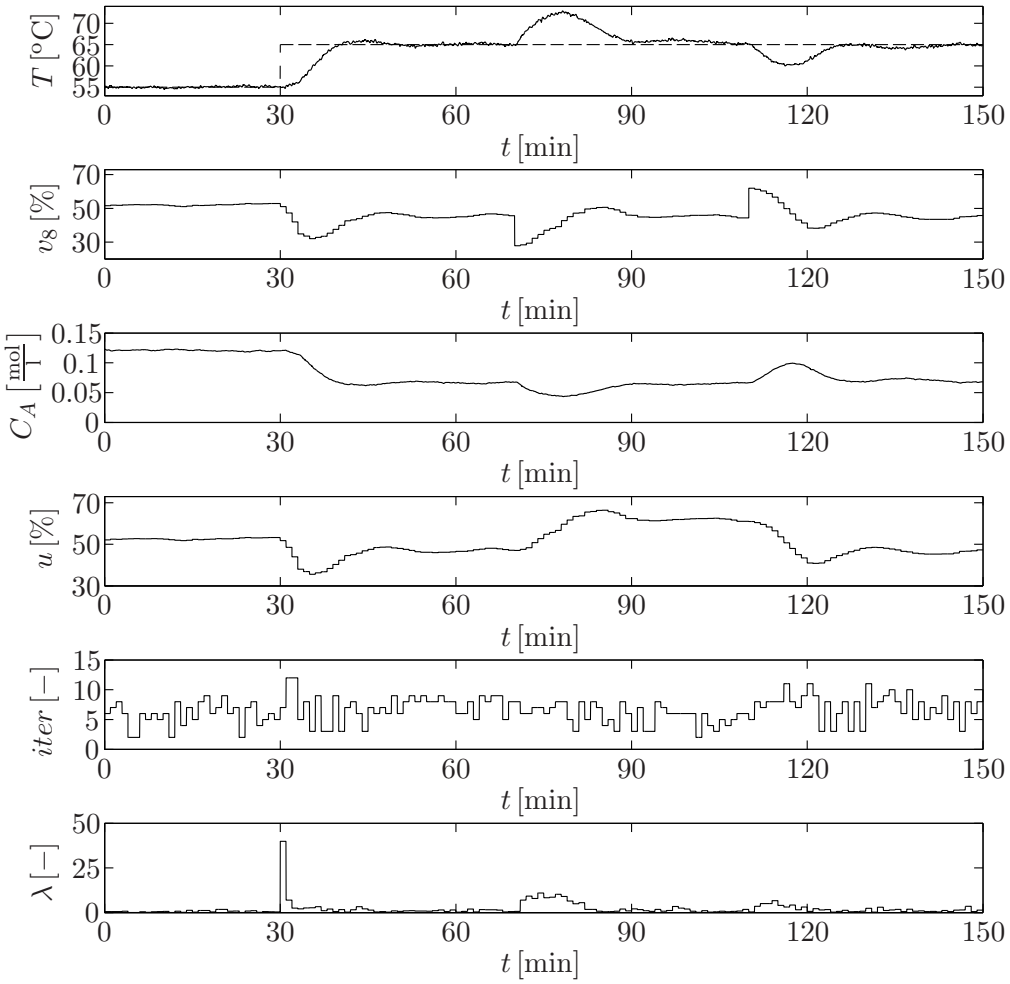


**Figure 5.5**: Disturbance rejection experiment (disturbance in the valve opening $v_8$) controlled by the proposed NMPC strategy based on the convexification approach. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, input value $u$ calculated by the controller, necessary iterations *iter* and weighting factor $\lambda$.

The third experiment with the pilot plant controlled by the proposed NMPC strategy was carried out with an additive disturbance in the system input (see Fig. 5.5). The mentioned disturbance has a value of $\Delta v_8 = -15\,\%$ and was applied in the period from $t = 70\,\text{min}$ until $t = 110\,\text{min}$. During the application of the disturbance, the effective valve opening is given by $v_8 = u + \Delta v_8$ whereas, in absence of the disturbance, the valve opening is defined by $v_8 = u$. It can be observed that the controller, after the application of the disturbance, efficiently compensates the error in the valve opening and the in temperature. After the sudden disappearance of the disturbance, the controller compensates the error in the output temperature and the system reaches steady state at the end of the experiment. In spite of the amplitude of the chosen disturbance, the proposed NMPC strategy rejects the disturbance efficiently and leads the system

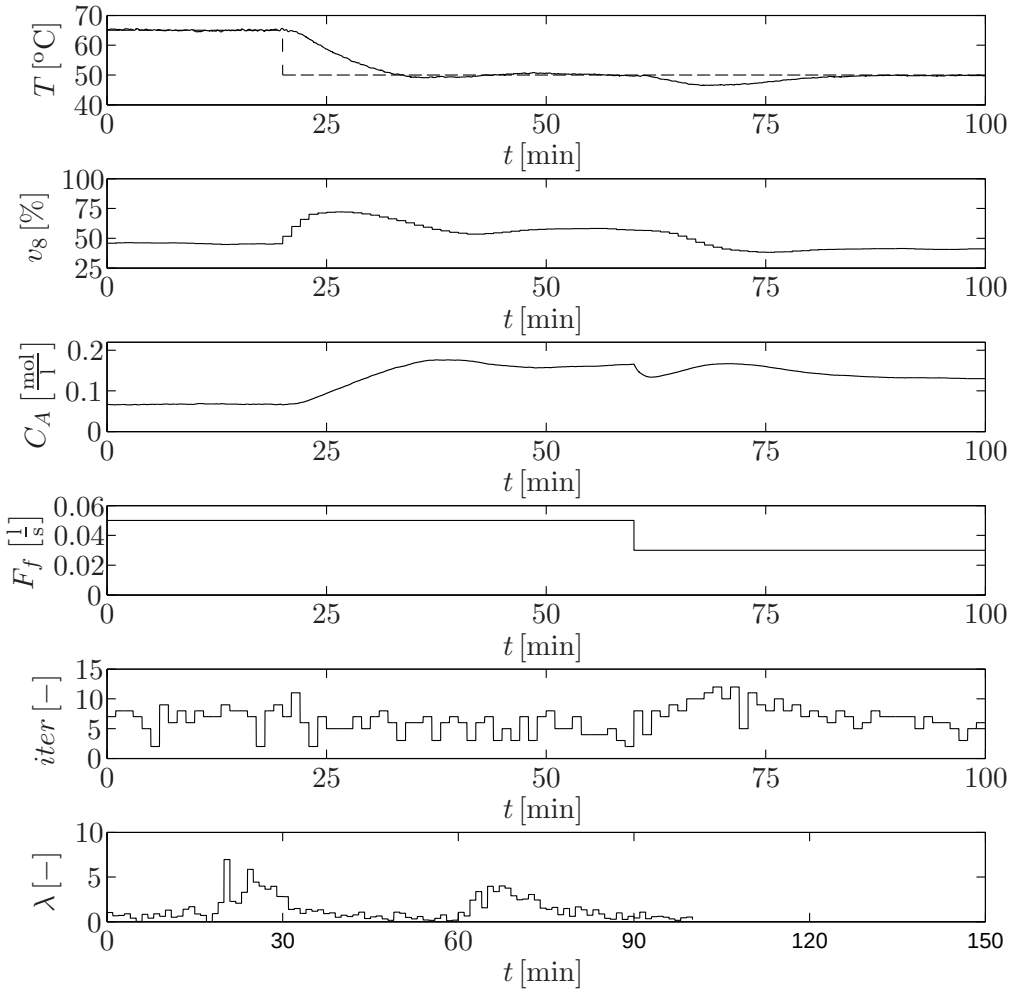


**Figure 5.6**: Disturbance rejection experiment (disturbance in the feed $F_f$) controlled by the proposed NMPC strategy based on the convexification approach. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, feed $F_f$, necessary iterations *iter* and weighting factor $\lambda$.

to the desired setpoint.

In the last experiment, shown in Fig. 5.6, the capabilities of the proposed NMPC strategy to reject an additive disturbance in the feed $F_f$ were tested. The chosen disturbance has a value of $\Delta F_f = -0.02\,\mathrm{l/s}$, which corresponds to an error of $-40\,\%$ with respect to the initial feed, and was applied in $t = 60\,\mathrm{min}$. The application of the disturbance results in a reduced concentration $C_A$ and, as a consequence, in a decreasing temperature with a maximum error of $-3.5\,^{\circ}\mathrm{C}$. The controller efficiently rejects the applied disturbance by reducing the opening of the valve $v_8$ and the system reaches steady state approximately in $t = 90\,\mathrm{min}$, 30 minutes after the application of the disturbance. Neither the temperature nor the valve opening show oscillations

during the application of the disturbance due to the smooth behavior of the proposed controller. With respect to the weighting factor, the optimization algorithm resulted always in a value $\lambda < 10$, except in the setpoint changes, requiring a higher value to assure convexity.

With the proposed NMPC strategy based on a convexification of the performance index, the calculation was carried out within the used sampling time of $t_s = 60\,\text{s}$. The average computation time to calculate a new input sequence in the shown experiments was $t_c^{avg} = 0.736\,\text{s}$, with a maximum of $t_c^{max} = 4.462\,\text{s}$ and a minimum of $t_c^{min} = 0.065\,\text{s}$. The average number of iterations to meet the final accuracy was 6.35, with a maximum of 12 and a minimum of 2 iterations.

### 5.4.2 Convexified optimization with guaranteed stability

The NMPC strategy using the optimization based on the convexification approach with guaranteed stability (see Section 5.3) was implemented in the Matlab/Simulink environment and tested on the pilot plant emulating an exothermic chemical reaction. With the mentioned control strategy, several experiments were carried out, including setpoint tracking and disturbance rejection. The control strategy was implemented with a prediction horizon of $N = 80$ and a control horizon of $N_u = 15$, satisfying the necessary condition $N \geq N_t + N_u$ to guarantee stability. Furthermore, for the final accuracy, a value of $\varepsilon = 10^{-3}$ was chosen and for the initial weighting factor a value of $\lambda^{(0)}$ was used. The following input constraints were considered:

$$\begin{aligned} 5 &\leq u(k+i|k) \leq 100, \quad i = 0, \ldots, 14 \\ -20 &\leq \Delta u(k+i|k) \leq 20, \quad i = 0, \ldots, 14 \\ u(k+14|k) &= u_r(k) \end{aligned} \tag{5.90}$$

where the equality constraint has been included for stability purposes. The experiments carried out with the proposed NMPC strategy correspond to the ones presented in Section 5.4.1.

In a first step, a setpoint tracking experiment (see Fig. 5.7) was carried with the pilot plant emulating an exothermic chemical reaction and controlled by the NMPC strategy based on the convexification approach with guaranteed stability. During the experiment, the setpoint was changed twice, first from 55 °C to 65 °C and later to 45 °C. Only after the first setpoint change, a small overshoot can be observed (approximately 1.2 °C). The control strategy reacts quite fast to the setpoint changes and compensates efficiently the divergence between the system output and the given reference. In comparison to the results shown in Fig. 5.3, the necessary modification in the cost function

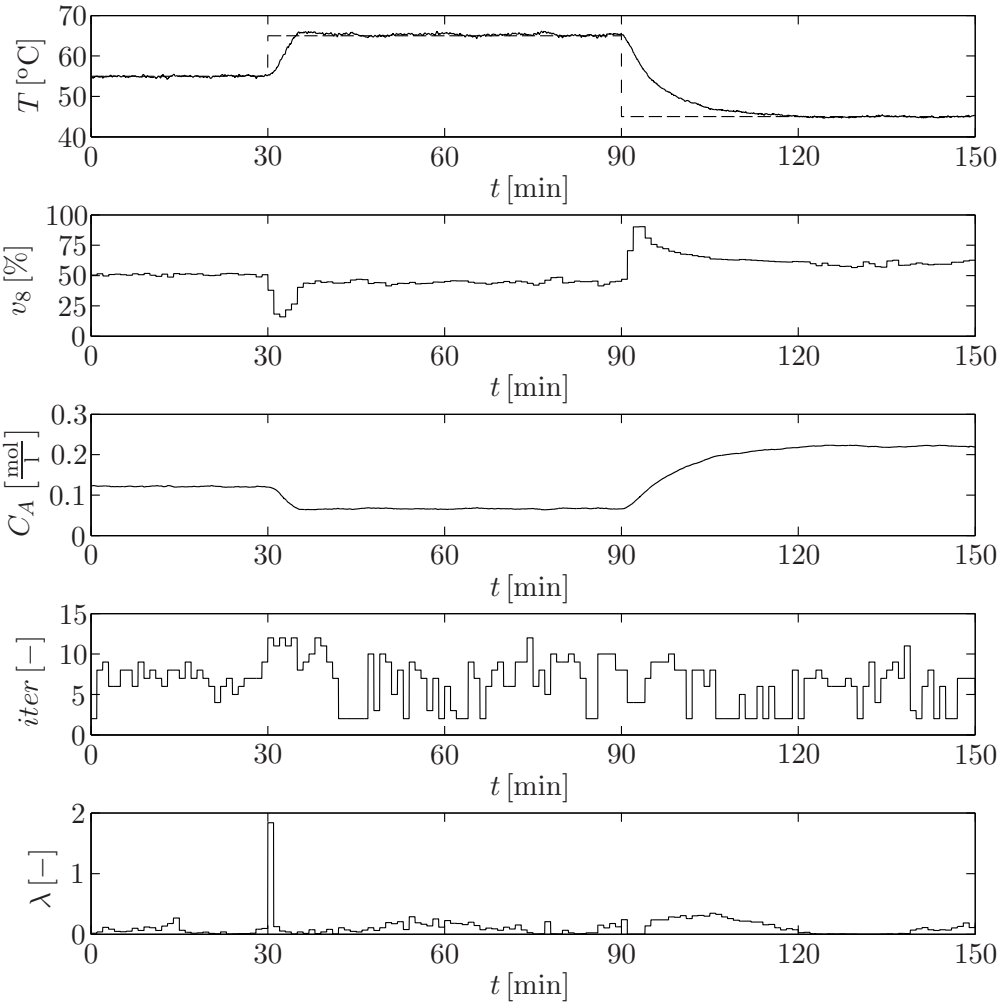


**Figure 5.7**: Setpoint tracking experiment controlled by the proposed NMPC strategy based on the convexification approach with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, necessary iterations *iter* and weighting factor $\lambda$.

to guarantee stability, i.e. the weighting of the difference between the input sequence and the steady-state input instead of the weighting of the control effort, results in a faster tracking after the first setpoint change and to a moderately slower reaction after the second modification of the reference.

For the second experiment, an error in one of the parameters of the underlying exothermic chemical reaction was introduced. Therefore, the value of the activation energy $E$ was increased by 3 % and held constant during the entire experiment. Due to the model mismatch, the results (see Fig. 5.8) show some overshoot after the setpoint changes (approximately 1.5 °C and −1 °C after the first and second change,

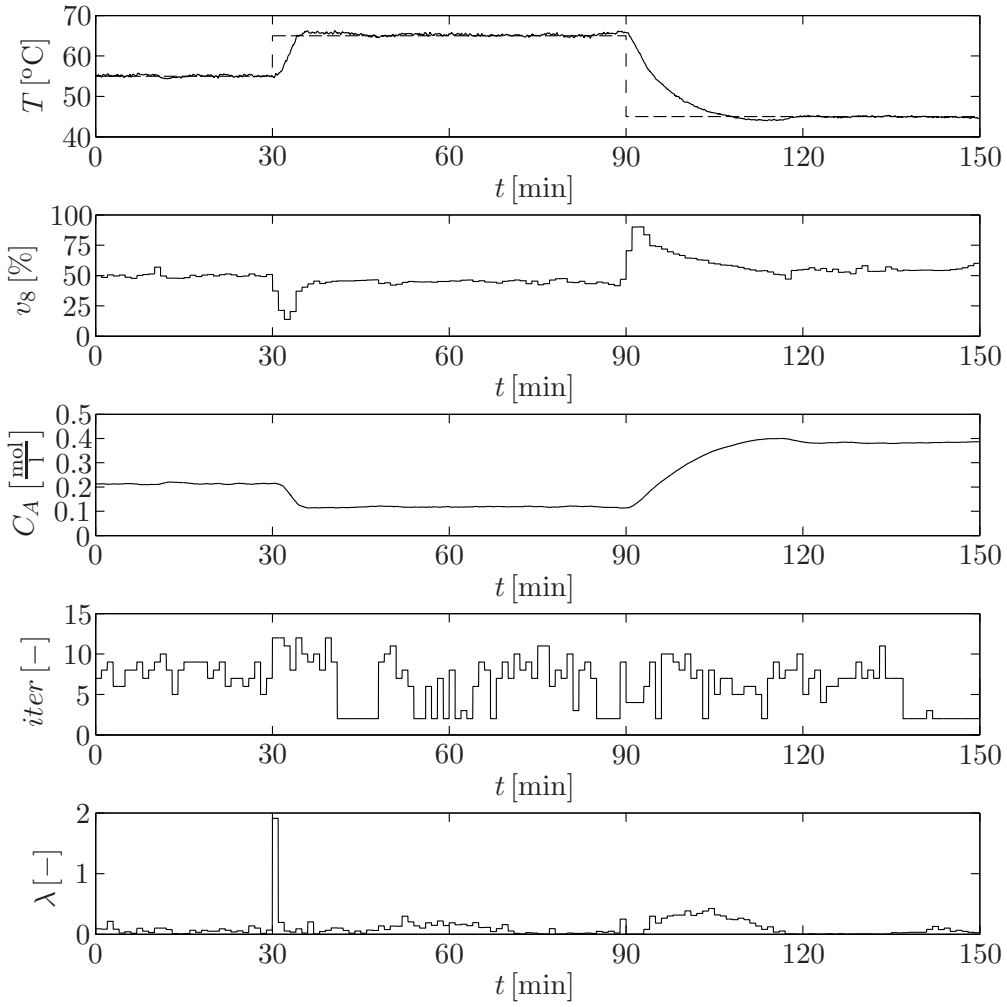


**Figure 5.8**: Disturbance rejection experiment (persistent disturbance in the emulated chemical reaction) controlled by the proposed NMPC strategy based on the convexification approach with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, necessary iterations $iter$, and weighting factor $\lambda$.

respectively). Nevertheless, the control strategy with guaranteed stability stabilizes the system on the reference and shows only marginal oscillations in the temperature and very small control moves. With respect to the results presented in Fig. 5.4, the proposed control strategy reduced considerably the overshoot after the setpoint changes and shows a more aggressive behavior after the first setpoint change. In spite of the uncertainty in the chemical reaction, the NMPC strategy showed a good behavior both in setpoint tracking and stabilization of the system output in a given reference.

In the third experiment, the disturbance rejection capabilities of the proposed control strategy were tested by means of an additive disturbance in the system input (see

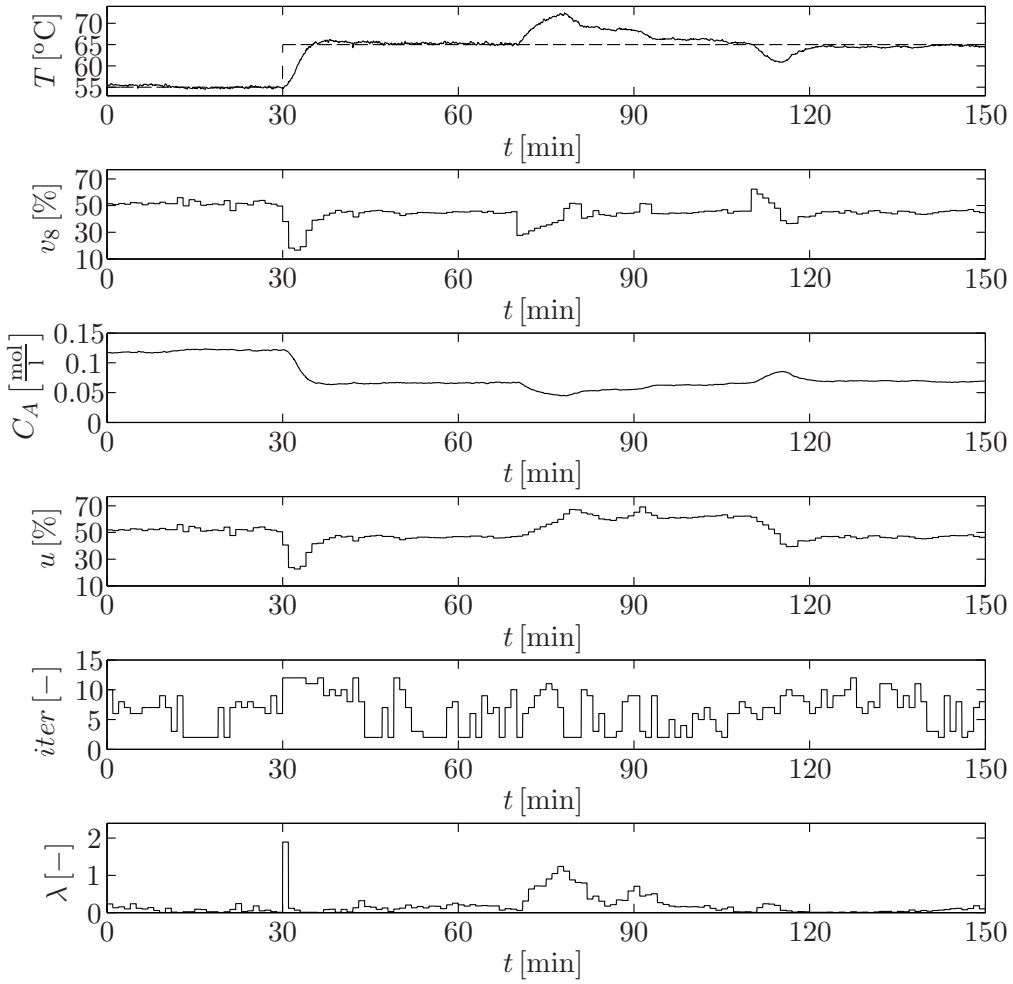


**Figure 5.9**: Disturbance rejection experiment (disturbance in the valve opening $v_8$) controlled by the proposed NMPC strategy based on the convexification approach with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, input value $u$ calculated by the controller, necessary iterations *iter* and weighting factor $\lambda$.

Fig. 5.9). Therefore, an error of $\Delta v_8 = -15\,\%$ was applied to the valve opening $v_8$ in $t = 70\,\text{min}$ and removed in $t = 110\,\text{min}$. After the application of the disturbance, the temperature increases rapidly due to the reduced effective opening of the valve $v_8$. Due to this error, the proposed controller increases the valve opening and reduces the divergence between the measured temperature and the setpoint and, as a consequence, rejects the applied disturbance successfully. After the disappearance of the disturbance at $t = 110\,\text{min}$, the controller rapidly stabilizes the temperature in the setpoint. Generally, the results underline the disturbance rejection capabilities of the proposed controller.

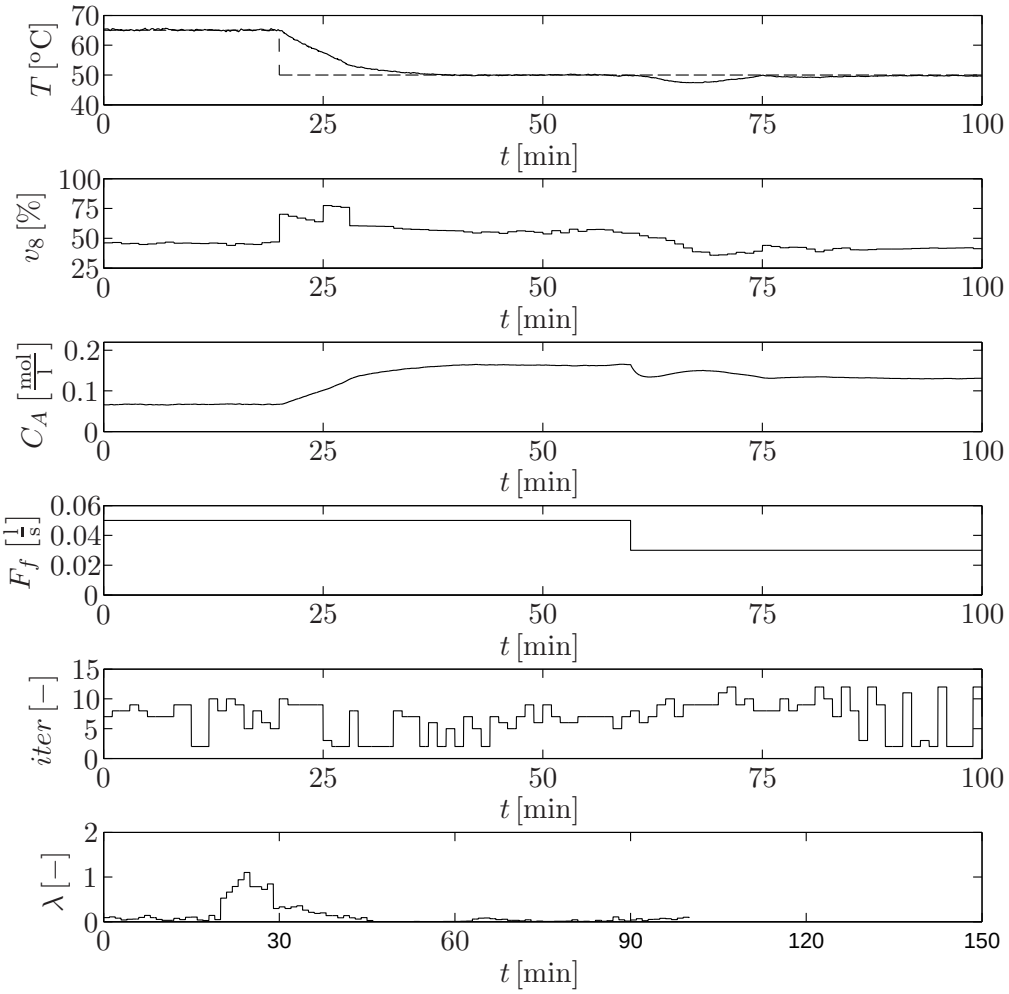


**Figure 5.10**: Disturbance rejection experiment (disturbance in the feed $F_f$) controlled by the proposed NMPC strategy based on the convexification approach with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, feed $F_f$, necessary iterations *iter* and weighting factor $\lambda$.

The last experiment, see results in Fig. 5.10, was carried out with a disturbance in the feed $F_f$. At $t = 60\,\text{min}$, the disturbance $\Delta F_f = -0.02\,\text{l/s}$ was applied to the system and held constant until the end of the experiment. The applied disturbance corresponds to an error of $-40\,\%$ in the initial feed and results in a reduced concentration $C_A$ in the plant reactor. As a result of the low concentration $C_A$ the temperature decreases and reaches a maximum error of $-2.6\,^{\circ}\text{C}$. The experimental results show that the proposed control strategy reduces the opening of the valve $v_8$ and successfully rejects the applied disturbance. In spite of the magnitude of the disturbance, the controller needs only 15 minutes to stabilize the system output in the given reference. In comparison with the experiment shown in Fig. 5.6, the use of the proposed control strategy results in a lower maximum output error and a considerably faster rejection of the disturbance.

Finally, it has to be mentioned that the proposed control strategy with guaranteed stability always solved the optimization problem within the used sampling time of $t_s = 60\,\text{s}$. During the experiments the average computation time to compute a new input sequence was $t_c^{avg} = 1.115\,\text{s}$, with a minimum of $t_c^{min} = 0.2\,\text{s}$ and a maximum of $t_c^{max} = 7.945\,\text{s}$. The average number of necessary iterations to find the solution to the optimization problem was 6.35, with a maximum of 12 and a minimum of 2 iterations. Note that the proposed NMPC strategy based on convexification with guaranteed stability (see Section 5.3) computes the input sequence with a considerably lower weighting factor $\lambda$ than the original NMPC strategy based on convexification (see Section 5.2.1). The resulting weighting factors can be compared directly in the experimental results shown in Section 5.4.1 and in Section 5.4.2.

## 5.5 Conclusions of the chapter

The current chapter has presented two NMPC strategies based on the convexification of the performance index to be minimized. Both control strategies have been developed for quadratic cost functions considering second order Volterra series prediction models.

The proposed control strategies are based on the approximation of the original cost function by convex quadratic functions. With a control effort weighting [5], convexity of the approximated functions can be assured by a suitable choice of the parameter $\lambda$. It can be shown that the original cost function with a weighting factor $\lambda$ can be approximated by means of a convex hull based on the pointwise maximum of several convex quadratic functions using the same parameter $\lambda$. Then, globally minimizing the convex hull, a new input sequence can be calculated. The convexification was included in an iterative optimization algorithm to be used as an NMPC strategy. It was shown that the difference between the cost of the original cost function and the convex hull decreases monotonically and, as a consequence, converges to the global minimum. For both control strategies input constraints have been considered in the optimization algorithm. Furthermore, for the control strategy proposed in Section 5.3, input-to-state stability can be guaranteed for a prediction horizon satisfying $N \geq N_t + N_u$.

Finally, both control strategies were applied to the pilot plant emulating an exothermic chemical reaction. With the control strategies, several experiments including set-point tracking and disturbance rejection were carried out. Both control strategies showed a good control behavior stabilizing the system output in the given reference

[5] In the control strategy proposed in Section 5.3, a weighting of the deviation between the input sequence and the necessary steady-state input, i.e. $\lambda(\boldsymbol{u} - \boldsymbol{u}_r)^T(\boldsymbol{u} - \boldsymbol{u}_r)$, has been considered.

and rejecting the applied disturbances in an efficient manner. The behavior of the control strategies was illustrated by means of experimental results.

# Chapter 6

# Linear Min-Max MPC

In Min-Max MPC (MMMPC), the control signal is computed for the worst case of a cost function that considers the effect of process model uncertainties and disturbances in the controller performance [25]. With an usually quadratic cost function $J$, the state vector $x$ and the input sequence $\boldsymbol{u}$, the optimal input sequence $\boldsymbol{u}^*$ is computed by minimization of the worst case cost depending on the disturbance and uncertainty vector $\boldsymbol{\theta}$:

$$\boldsymbol{u}^* = \arg \min_{\boldsymbol{u}} \max_{\theta} J(\boldsymbol{u}, x, \boldsymbol{\theta}) \tag{6.1}$$

possibly subject to constraints in the input sequence and the disturbance and uncertainty vector. The main drawback of this approach is the computational burden that takes to compute the control signal. This usually involves the solution of a NP-hard min-max problem [64, 122]. As a result, the number of applications of these control strategies is very small, even when there is evidence that they work better than standard predictive controllers in processes with uncertain dynamics [23].

Multi-parametric programming has been applied to show that the MMMPC control law is piecewise affine when a quadratic [106] or 1-norm based criterion [14, 55] is used as the cost function. Thus, explicit forms of the control law can be built. Such explicit forms can be evaluated very fast provided that the complexity of the state space partition is moderate, which is the case for many applications. However, if the process model or the controller tuning parameters change, the computation of the controller has to be redone.

A common solution to the computational burden issue is to use an upper bound of the worst case cost instead of computing it explicitly. This upper bound can be computed by using linear matrix inequalities (LMI) techniques such as in [59] and [74].

However, the LMI problems have a computational burden that cannot be neglected in certain applications. In this chapter, two MMMPC strategies for linear models based on computationally cheap upper-bounds of the worst case cost will be discussed.

The following section gives a detailed description of the general idea of MMMPC and defines the optimization problem based on a semi-feedback approach. Afterwards a min-max control strategy using a nonlinear upper bound of the worst case cost [105] will presented. For this control strategy, originally published in [105] without stability proof, guaranteed stability will be shown. In the second control strategy, published in [3], the min-max problem is replaced by a quadratic programming (QP) problem that provides a close approximation to the solution of the original min-max problem. Both control strategies have a much lower computational complexity than the original min-max optimization problem and can be used with prediction horizons typical in linear MPC. Finally, the control strategies will be applied to a benchmark system and their performance will be illustrated by experimental results.

It has to be mentioned that in this chapter for the sake of readability, a slightly different mathematical notation will be used. In contrast to the other chapters, vectors are not necessarily written in bold letters. The bold notation is used only for sequences along the prediction horizon, e.g. the future input sequence $\boldsymbol{u}$ or the disturbance sequence $\boldsymbol{\theta}$. In contrast, the vectors denoting the state, the disturbance or the input in a certain sampling period are denoted with non-bold letters, e.g. the current state vector $x(k)$ or the input vector $u(k+1|k)$ for $k+1$ computed in $k$.

## 6.1 General idea of min-max MPC

This section defines the general optimization problem of MMMPC. Furthermore, an augmented representation of the optimization problem will be given and a simple upper bound of the minimization problem will be given.

### 6.1.1 Problem description

Consider the following discrete-time state space model with bounded additive uncertainties:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + D\theta(k) \\ y(k) &= Cx(k) \end{aligned} \tag{6.2}$$

with $x(k) \in \mathbb{R}^{n_x}$ the state vector, $u(k) \in \mathbb{R}^{n_u}$ the input vector and $\theta(k) \in \{\theta \in \mathbb{R}^{n_\theta} : \|\theta\|_\infty \leq \varepsilon\}$ the uncertainty, that is supposed to be bounded. The system is subject to $n_c$ state and input time-invariant constraints $F_u u(k) + F_x x(k) \leq b_c$ where $F_u \in \mathbb{R}^{n_c \times n_u}$ and $F_x \in \mathbb{R}^{n_c \times n_x}$. It is assumed a semi-feedback approach in which the control input is given by:

$$u(k) = -Kx(k) + v(k) \tag{6.3}$$

where the feedback matrix $K$ is chosen to achieve some desired property such as nominal stability or linear quadratic regulator (LQR) optimality without constraints. The MMMPC controller will compute the optimal sequence of correction control inputs $v(k)$. With the semi-feedback approach, the state equation of system (6.2) can be rewritten as

$$x(k+1) = A_{cl}x(k) + Bv(k) + D\theta(k) \tag{6.4}$$

where $A_{cl} = (A - BK)$. The proposed strategies in the following sections also work without the semi-feedback approach, i.e. $u(k) = v(k)$. All the computational advantages of the strategies remain the same and the procedures described here can be used without any modification. Furthermore if the process is open-loop stable the stabilizing conditions, which will be discussed later, can be used without problems.

The cost function is a quadratic performance index:

$$\begin{aligned} J(x, \boldsymbol{v}, \boldsymbol{\theta}) \quad = \quad & \sum_{j=0}^{N-1} x(k+j|k)^T Q x(k+j|k) + \sum_{j=0}^{N-1} u(k+j|k)^T R u(k+j|k) \\ & + x(k+N|k)^T P x(k+N|k) \end{aligned} \tag{6.5}$$

where $x(k|k) = x$ is the current state and $x(k+j|k)$ is the prediction of the state for $k+j$ made at $k$. The current input signal is $u(k|k) = -Kx(k|k) + v(k|k)$ and the future input for $k+j$ made at $k$ is denoted $u(k+j|k) = -Kx(k+j|k) + v(k+j|k)$. The input correction sequence along the prediction horizon $N$ is defined generally as:

$$\boldsymbol{v} = \begin{bmatrix} v(k|k)^T \\ v(k+1|k)^T \\ \vdots \\ v(k+N-1|t)^T \end{bmatrix} \tag{6.6}$$

Note that the values of the input sequence and the states depend on the future values of the uncertainty. The sequence of future values of the uncertainty over the prediction horizon $N$ is denoted by:

$$\boldsymbol{\theta} = \begin{bmatrix} \theta(k)^T \\ \theta(k+1)^T \\ \vdots \\ \theta(k+N-1)^T \end{bmatrix} \tag{6.7}$$

and the set of possible uncertainty trajectories is defined by:

$$\boldsymbol{\Theta} = \{\boldsymbol{\theta} \in \mathbb{R}^{Nn_\theta} : \|\boldsymbol{\theta}\|_\infty \leq \varepsilon\} \tag{6.8}$$

The matrices $Q, P \in \mathbb{R}^{n_x \times n_x}$ and $R \in \mathbb{R}^{n_u \times n_u}$ are symmetric positive definite matrices used as weighting parameters.

A MMMPC strategy [25] minimizes the cost function (6.5) for the worst possible case of the predicted future evolution of the process state or output signal. This is accomplished through the solution of the min-max problem:

$$\begin{aligned} \boldsymbol{v}^* \;=\; & \arg \min_{\boldsymbol{v}} \max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} J(x, \boldsymbol{v}, \boldsymbol{\theta}) \\ & \text{s.t. } F_u u(k+j|k) + F_x x(k+j|k) \leq b_c \\ & \quad j = 0, \ldots, N, \ \forall \boldsymbol{\theta} \in \boldsymbol{\Theta} \\ & \quad x(k+N|k) \in \Omega, \ \ \forall \boldsymbol{\theta} \in \boldsymbol{\Theta} \end{aligned} \tag{6.9}$$

A terminal region constraint $x(k+N|k) \in \Omega$, where $\Omega$ is a polyhedron, is included to assure stability of the control law [85]. Furthermore, the terminal region $\Omega$ and the matrix $P$ are assumed to satisfy the following conditions:

- **C1**: If $x \in \Omega$ then $A_{CL}x + D\theta \in \Omega$, for every $\theta \in \{\theta \in \mathbb{R}^{n_\theta} : \|\theta\|_\infty \leq \varepsilon\}$.
- **C2**: If $x \in \Omega$ then $u(x) = -Kx + v \in U$, where $U \triangleq \{u : F_u u + F_x x \leq g\}$.
- **C3**: $P - A_{CL}^T P A_{CL} > Q + K^T R K$.

The stability of $A_{CL}$ guarantees the existence of a positive definite matrix $P$ satisfying C3.

The predictions $x(k+j|k)$ and control actions $u(k+j|k)$ depend linearly on $x$, $\boldsymbol{v}$ and $\boldsymbol{\theta}$. This means that it is possible to find a vector $\boldsymbol{b}_\theta \in \mathbb{R}^{n_c}$ and matrices $M_x$, $M_v$ and $M_\theta$ [23], such that all the robust linear constraints of problem (6.9) can be rewritten as:

$$M_x x + M_v \boldsymbol{v} + M_\theta \boldsymbol{\theta} \leq \boldsymbol{b}_\theta, \quad \forall \boldsymbol{\theta} \in \boldsymbol{\Theta} \tag{6.10}$$

Defining $M_x^{[i]}$, $M_v^{[i]}$, $M_\theta^{[i]}$ as the $i$-th rows of $M_x$, $M_v$ and $M_\theta$, respectively, and $b_\theta^{[i]}$ as the $i$-th component of $\boldsymbol{b}_\theta \in \mathbb{R}^{n_c}$ the robust linear constraints (6.10) can be expressed as:

$$M_x^{[i]} x + M_v^{[i]} \boldsymbol{v} + M_\theta^{[i]} \boldsymbol{\theta} \leq b_\theta^{[i]}, \quad i = 1 \ldots, n_c, \ \ \forall \boldsymbol{\theta} \in \boldsymbol{\Theta} \tag{6.11}$$

Denoting now $\|M_\theta^{[i]}\|_1$ the sum of the absolute values of row $M_\theta^{[i]}$ and taking into account that:

$$\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} M_\theta^{[i]} \boldsymbol{\theta} = \max_{\|\boldsymbol{\theta}\|_\infty \leq \varepsilon} M_\theta^{[i]} \boldsymbol{\theta} = \varepsilon \|M_\theta^{[i]}\|_1 \tag{6.12}$$

the robust fulfillment of the constraints is satisfied if and only if $M_x^{[i]}x + M_v^{[i]}\boldsymbol{v} + \varepsilon\|M_\theta^{[i]}\|_1 \leq b_\theta^{[i]}$ for $i = 1, \ldots, n_c$. Therefore, to guarantee robust constraint satisfaction, the set of linear constraints:

$$M_x x + M_v \boldsymbol{v} \leq \boldsymbol{b}_\varepsilon \tag{6.13}$$

must be satisfied, where the $i$-th component of $\boldsymbol{b}_\varepsilon$ is $b_\varepsilon^{[i]} = b_\theta^{[i]} - \varepsilon\|M_\theta^{[i]}\|_1$. Note that this is a necessary and sufficient condition.

Taking into account (6.3), (6.4) and (6.5), the performance index function can be evaluated as a quadratic function:

$$\begin{aligned} J(x, \boldsymbol{v}, \boldsymbol{\theta}) \;=\; & \boldsymbol{v}^T M_{vv} \boldsymbol{v} + \boldsymbol{\theta}^T M_{\theta\theta} \boldsymbol{\theta} + 2\,\boldsymbol{\theta}^T M_{\theta v} \boldsymbol{v} + \\ & 2\,x^T M_{vf}^T \boldsymbol{v} + 2\,\boldsymbol{\theta}^T M_{\theta f} x + x^T M_{ff} x \end{aligned} \tag{6.14}$$

where the matrices can be obtained from the system and the control parameters [23]. With (6.14) being a convex cost function in $\boldsymbol{\theta}$, the solution to the maximization problem in (6.9) can be found at least in one of the vertices of $\boldsymbol{\Theta}$. With the property of convexity in $\boldsymbol{\theta}$ and the reformulated robust linear constraints (6.13), the optimization problem (6.9) is equivalent to

$$\begin{aligned} \boldsymbol{v}^* = \;\arg\; & \min_{\boldsymbol{v}} \max_{\boldsymbol{\theta} \in \mathrm{vert}(\boldsymbol{\Theta})} J(x, \boldsymbol{v}, \boldsymbol{\theta}) \\ & \text{s.t. } M_x x + M_v \boldsymbol{v} \leq \boldsymbol{b}_\varepsilon \end{aligned} \tag{6.15}$$

where $\mathrm{vert}(\boldsymbol{\Theta})$ is the set of vertices of $\boldsymbol{\theta}$ [23]. Then, based on (6.14), the maximum cost for a given $x$ and $\boldsymbol{v}$ is denoted as

$$\begin{aligned} J^*(x, \boldsymbol{v}) \;&=\; \max_{\boldsymbol{\theta} \in \mathrm{vert}(\boldsymbol{\Theta})} J(x, \boldsymbol{v}, \boldsymbol{\theta}) \\ &=\; \max_{\boldsymbol{\theta} \in \mathrm{vert}(\boldsymbol{\Theta})} \boldsymbol{\theta}^T H \boldsymbol{\theta} + 2\,\boldsymbol{\theta}^T \boldsymbol{q}(x, \boldsymbol{v}) + J(x, \boldsymbol{v}, 0) \end{aligned} \tag{6.16}$$

where

$$H \;=\; M_{\theta\theta} \tag{6.17}$$

$$\boldsymbol{q}(x, \boldsymbol{v}) \;=\; M_{\theta v} \boldsymbol{v} + M_{\theta f} x \tag{6.18}$$

$$J(x, \boldsymbol{v}, 0) \;=\; \boldsymbol{v}^T M_{vv} \boldsymbol{v} + 2\,x^T M_{vf}^T \boldsymbol{v} + x^T M_{ff} x \tag{6.19}$$

with $J(x, \boldsymbol{v}, 0)$ denoting the part of the cost that does not depend on the uncertainty, i.e. the nominal cost. With this definition, problem (6.15) can be rewritten as

$$\begin{aligned} \boldsymbol{v}^* = \;\arg\; & \min_{\boldsymbol{v}} J^*(x, \boldsymbol{v}) \\ & \text{s.t. } M_x x + M_v \boldsymbol{v} \leq \boldsymbol{b}_\varepsilon \end{aligned} \tag{6.20}$$

and the system is controlled by $u(k|k) = -Kx(k) + v^*(k|k)$. The maximization of (6.16) is a well known NP-hard optimization problem which requires the evaluation of the $2^{Nn_\theta}$ vertices. Due to the exponential complexity of the maximization problem (6.16) and, as a consequence, of the min-max optimization problem (6.20), only small prediction horizons can be used.

### 6.1.2 Augmented optimization problem and simple upper bound

The maximum cost $J^*(x, \boldsymbol{v})$ can be represented as an augmented optimization problem in matrix form. Therefore, (6.16) is rewritten as:

$$J^*(x, \boldsymbol{v}) = \max_{\boldsymbol{\theta} \in \mathrm{vert}\{\boldsymbol{\Theta}\}} \begin{bmatrix} \boldsymbol{\theta} \\ 1 \end{bmatrix}^T \begin{bmatrix} H & \boldsymbol{q}(x, \boldsymbol{v}) \\ \boldsymbol{q}(x, \boldsymbol{v})^T & J(x, \boldsymbol{v}, 0) \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta} \\ 1 \end{bmatrix} \tag{6.21}$$

The augmented maximization problem (6.21) can be expressed as

$$J^*(x, \boldsymbol{v}) = \max_{\|\boldsymbol{z}\|_\infty \leq 1} \boldsymbol{z}^T M(\boldsymbol{v}) \boldsymbol{z} \tag{6.22}$$

with the vector $\boldsymbol{z} \in \mathbb{R}^{n_z}$ and the matrix $M(\boldsymbol{v}) \in \mathbb{R}^{n_z \times n_z}$

$$\boldsymbol{z} = \begin{bmatrix} \frac{\boldsymbol{\theta}}{\varepsilon} \\ 1 \end{bmatrix}, \quad M(\boldsymbol{v}) = \begin{bmatrix} \varepsilon^2 H & \varepsilon \boldsymbol{q}(x, \boldsymbol{v}) \\ \varepsilon \boldsymbol{q}(x, \boldsymbol{v})^T & J(x, \boldsymbol{v}, 0) \end{bmatrix} \tag{6.23}$$

and $n_z = N n_\theta + 1$. The augmented optimization problem (6.22) will be used in the later presented control strategies to compute an upper bound of the worst case cost.

The stability proofs of the later presented control strategies and the calculation of a candidate input sequence to obtain a quadratic upper bound in Section (6.3) are based on a simple upper bound of the worst case cost. This simple upper bound approximates (6.16) by defining a maximum value for the terms $\boldsymbol{\theta}^T H \boldsymbol{\theta}$ and $\boldsymbol{\theta}^T \boldsymbol{q}(x, \boldsymbol{v})$ depending on the uncertainty $\boldsymbol{\theta}$. For a given matrix $H$ (6.17) a diagonal matrix $T$ with the components:

$$T^{[ii]} = \sum_{j=1}^{N n_\theta} |H^{[ij]}| \quad \text{for} \quad i = 1, \ldots, N n_\theta \tag{6.24}$$

can be defined. The off-diagonal elements are given by $T^{[ij]} = 0 \; \forall \, i \neq j$, $i = 1, \ldots, N n_\theta$ and $j = 1, \ldots, N n_\theta$. With the new matrix $T$ an approximated cost function

$$\tilde{J}(x, \boldsymbol{v}, \boldsymbol{\theta}) = \boldsymbol{\theta}^T T \boldsymbol{\theta} + 2 \, \boldsymbol{\theta}^T \boldsymbol{q}(x, \boldsymbol{v}) + J(x, \boldsymbol{v}, 0) \tag{6.25}$$

can be defined. With $T \geq H$ the statement

$$\tilde{J}(x, \boldsymbol{v}, \boldsymbol{\theta}) \geq J(x, \boldsymbol{v}, \boldsymbol{\theta}) \tag{6.26}$$

holds, i.e. $\tilde{J}(x, \boldsymbol{v}, \boldsymbol{\theta})$ represents an upper bound for the cost function $J(x, \boldsymbol{v}, \boldsymbol{\theta})$ given in (6.14). Then, the maximum of $\tilde{J}(x, \boldsymbol{v}, \boldsymbol{\theta})$, corresponding to the worst case cost of the approximated cost function, can be calculated easily by:

$$\tilde{J}^*(x, \boldsymbol{v}) = J(x, \boldsymbol{v}, 0) + \|H\|_s \, \varepsilon^2 + 2 \, \varepsilon \left\| \boldsymbol{q}(x, \boldsymbol{v})^T \right\|_1 \tag{6.27}$$

where $\|H\|_s$ denotes the sum of the absolute values of the elements of $H$ or, identically $\|H\|_s = \text{trace}(T)$. With the worst case cost (6.27) being a convex function in $\boldsymbol{v}$, the minimization problem can be solved by quadratic programming (QP). Hence, the optimal solution based on the simple upper bound of the worst case cost (6.27) is defined as:

$$\begin{aligned} \tilde{\boldsymbol{v}}^* = \quad \arg \quad & \min_{\boldsymbol{v}} \ \tilde{J}^*(x, \boldsymbol{v}) \\ & \text{s.t. } M_x x + M_v \boldsymbol{v} \leq \boldsymbol{b}_\varepsilon \end{aligned} \tag{6.28}$$

## 6.2 MMMPC with a nonlinear upper bound of the worst case cost

The MMMPC strategy presented in this Section, published originally in [105], uses a nonlinear upper bound of the worst case cost. In the presented control strategy, the upper bound is represented by a diagonal matrix. The mentioned diagonal matrix can be obtained by simple matrix operations and minimizes the error introduced by the upper bound. The computationally efficient algorithm to compute the upper bound can then be used in a MPC strategy to calculate the input sequence for the system.

Consider the augmented maximization problem (6.22) and diagonal matrix $S \in \mathbb{R}^{n_z \times n_z}$ with the diagonal elements given by $S^{[ii]}$. For[1] $S \geq M$ the statement:

$$\boldsymbol{z}^T M(\boldsymbol{v}) \boldsymbol{z} \leq \boldsymbol{z}^T S \boldsymbol{z} = \sum_{i=1}^{n} S^{[ii]} (z^{[i]})^2 \leq \text{trace}(S) \|\boldsymbol{z}\|_\infty^2 \leq \text{trace}(S) \tag{6.29}$$

is true and therefore:

$$J^*(x, \boldsymbol{v}) \leq \text{trace}(S) \tag{6.30}$$

As a consequence the trace of $S$ represents an upper bound of $J^*(x, \boldsymbol{v})$. The least conservative bound is the one obtained with matrix $S$ being a solution of the following linear matrix inequality (LMI) problem:

$$\begin{aligned} \breve{J}^*(x, \boldsymbol{v}) \quad = \quad & \min \ \text{trace}(S) \\ & \text{s.t. } S \geq M(\boldsymbol{v}) \\ & \quad S \text{ diagonal} \end{aligned} \tag{6.31}$$

In order to solve the previous LMI problem special optimization algorithms, e.g. interior point methods, are required. One of the drawbacks of these methods is the

[1] In this work a matrix inequality of the type $S \geq M$ is fulfilled if and only if $S - M$ is positive semi-definite.

possible excessive computational cost. The following section presents a diagonalization algorithm based only on simple matrix operations, originally published in [105], to determine a conservative upper bound of the worst case cost.

### 6.2.1 Minimization of the upper bound

The strategy is to obtain a diagonal matrix $\hat{\Gamma} \geq M(\boldsymbol{v})$ with a computationally efficient procedure that keeps the upper bound, i.e. trace($\hat{\Gamma}$), close to $\breve{J}^*(x, \boldsymbol{v})$. The employed strategy computes the matrix $\hat{\Gamma}$ by adding $n_z - 1$ positive semi-definite matrices $\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T$ to $M(\boldsymbol{v})$:

$$M(\boldsymbol{v}) + \boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T + \boldsymbol{\varphi}_2 \boldsymbol{\varphi}_2^T + \boldsymbol{\varphi}_3 \boldsymbol{\varphi}_3^T + \cdots + \boldsymbol{\varphi}_{n_z-1} \boldsymbol{\varphi}_{n_z-1}^T = \hat{\Gamma} \tag{6.32}$$

being $\hat{\Gamma}$ a diagonal matrix and the vectors $\boldsymbol{\varphi}_i \in \mathbb{R}^{n_z}$. The idea is to find the vectors $\boldsymbol{\varphi}_i$ for $i = 1, \ldots, n_z - 1$ such that the resulting matrix $\hat{\Gamma}$ is diagonal and the conservatism of the bound is held as low as possible. The approach presented in this section diagonalizes one row and one column of the original matrix $M(\boldsymbol{v})$ with every term $\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T$ added. The matrix $M(\boldsymbol{v})$ can be defined in general form as:

$$M(\boldsymbol{v}) = \begin{bmatrix} a & \boldsymbol{b}^T \\ \boldsymbol{b} & M_r \end{bmatrix} \tag{6.33}$$

with $a \in \mathbb{R}$, $\boldsymbol{b} \in \mathbb{R}^{n_z-1}$ and $M_r \in \mathbb{R}^{(n_z-1)\times(n_z-1)}$. Then, adding the first term $\boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T$ to (6.33), the desired result can be written as:

$$\begin{bmatrix} a & \boldsymbol{b}^T \\ \boldsymbol{b} & M_r \end{bmatrix} + \boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T = \begin{bmatrix} d & 0 \\ 0 & \hat{M}_r \end{bmatrix} \tag{6.34}$$

being $d \in \mathbb{R}$. After the first diagonalization step (6.34), the original matrix has been partially diagonalized so that the first column and the first row correspond to a diagonal matrix. Thereafter, $\boldsymbol{\varphi}_2$ is computed in order to partially diagonalize the submatrix $\hat{M}_r$. This process is repeated until the matrix is completely diagonalized.

For the diagonalization the vectors $\boldsymbol{\varphi}_i$ for $i = 1, \ldots, n_z - 1$ have to be found such that $\hat{\Gamma}$ is the smallest diagonal matrix with $\hat{\Gamma} \geq M(\boldsymbol{v})$. In a first step, the vector $\boldsymbol{\varphi}_1$ is defined as $\boldsymbol{\varphi}_1 = [\alpha \;\; \boldsymbol{\beta}^T]^T$ with $\alpha$ being a scalar and $\boldsymbol{\beta} \in \mathbb{R}^{n_z-1}$. Then, the term $\boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T$ results to be:

$$\boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T = \begin{bmatrix} \alpha \\ \boldsymbol{\beta} \end{bmatrix} \begin{bmatrix} \alpha & \boldsymbol{\beta}^T \end{bmatrix} = \begin{bmatrix} \alpha^2 & \alpha \boldsymbol{\beta}^T \\ \alpha \boldsymbol{\beta} & \boldsymbol{\beta} \boldsymbol{\beta}^T \end{bmatrix} \tag{6.35}$$

with $\alpha > 0$. From (6.34) and (6.35) results that $\alpha \boldsymbol{\beta} = -\boldsymbol{b}$ and, as a consequence, $\boldsymbol{\beta} = \frac{-\boldsymbol{b}}{\alpha}$, $d = a + \alpha^2$ and $\hat{M}_r = M_r + \frac{\boldsymbol{b}\boldsymbol{b}^T}{\alpha^2}$. The vectors $\boldsymbol{\varphi}_i$ for $i = 2, \ldots, n_z - 1$ can be computed in an analogous way, diagonalizing the submatrix $\hat{M}_r$ (6.34). The parameter

$\alpha$ has to be calculated such that the error introduced by the diagonalization procedure is minimized. This way, the difference between the upper bound and the worst case cost is kept as small as possible. This error is defined by:

$$\boldsymbol{z}^T \boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T \boldsymbol{z} = \boldsymbol{z}^T \begin{bmatrix} \alpha \\ -\frac{\boldsymbol{b}}{\alpha} \end{bmatrix} \begin{bmatrix} \alpha & \frac{-\boldsymbol{b}^T}{\alpha} \end{bmatrix} \boldsymbol{z} \tag{6.36}$$

The error (6.36) is maximized for[2]:

$$\boldsymbol{z}^* = \operatorname{sign} \begin{bmatrix} \alpha \\ -\frac{\boldsymbol{b}}{\alpha} \end{bmatrix} \tag{6.37}$$

Taking into account that:

$$\begin{bmatrix} \alpha & \frac{-\boldsymbol{b}^T}{\alpha} \end{bmatrix} \boldsymbol{z}^* = \left\| \begin{matrix} \alpha \\ -\frac{\boldsymbol{b}}{\alpha} \end{matrix} \right\|_1 \tag{6.38}$$

the maximum error introduced by the diagonalization procedure is

$$\left\| \begin{matrix} \alpha \\ -\frac{\boldsymbol{b}}{\alpha} \end{matrix} \right\|_1^2 \tag{6.39}$$

The minimization of the introduced error (6.39) can be obtained by setting the derivative

$$\left\| \begin{matrix} \alpha \\ -\frac{\boldsymbol{b}}{\alpha} \end{matrix} \right\|_1 = \alpha + \frac{1}{\alpha} \|\boldsymbol{b}\|_1 \tag{6.40}$$

equal to 0. Then, the value of $\alpha$ is:

$$\alpha = \sqrt{\|\boldsymbol{b}\|_1} \tag{6.41}$$

Finally, with the diagonal matrix $\hat{\Gamma}$ satisfying $\hat{\Gamma} \geq M(\boldsymbol{v})$, the upper bound of the worst case cost is defined as:

$$\hat{J}^*(x, \boldsymbol{v}) = \operatorname{trace}(\hat{\Gamma}) \tag{6.42}$$

The upper bound (6.42) obtained by the mentioned diagonalization algorithm shows a low suboptimality with respect to the exact solution $J^*(x, \boldsymbol{v})$. For further details on the suboptimality of the diagonalization algorithm see [105].

### 6.2.2 Algorithm for the computation of the upper bound

The computation of the upper bound $\hat{J}^*(x, \boldsymbol{v})$ (6.42) can be realized with the simple procedure, diagonalizing successively the matrix $M(\boldsymbol{v})$. With $M(\boldsymbol{v}) \in \mathbb{R}^{n_z \times n_z}$ the proposed algorithm calculates the vectors $\boldsymbol{\varphi}_i$ for $i = 1, \ldots, n_z - 1$ such that the introduced

[2]The maximum error is also obtained for $-\boldsymbol{z}^*$.

error is kept as small as possible. The algorithm to compute the upper bound $\hat{J}^*(x, \boldsymbol{v})$ is the following:

**Procedure 6.1** *Algorithm to compute the diagonal matrix* $\hat{\Gamma}$ *and the upper bound of the worst case cost* $\hat{J}^*(x, \boldsymbol{v}) = \text{trace}(\hat{\Gamma}) \geq \max_{\|\boldsymbol{z}\|_\infty \leq 1} \boldsymbol{z}^T M(\boldsymbol{v}) \boldsymbol{z}$.

1. *Set* $\hat{\Gamma}^{(0)} = M(\boldsymbol{v}) \in \mathbb{R}^{n_z \times n_z}$.
2. *For* $i = 1$ *to* $n_z - 1$
3. *Obtain* $M_{sub}^{(i)} \in \mathbb{R}^{(n_z+1-i)\times(n_z+1-i)}$ *and* $\Upsilon^{(i)} \in \mathbb{R}^{(i-1)\times(i-1)}$ *with*[3]
$$\hat{\Gamma}^{(i-1)} = \begin{bmatrix} \Upsilon^{(i)} & \mathbf{0}^T_{(n_z+1-i)\times(i-1)} \\ \mathbf{0}_{(n_z+1-i)\times(i-1)} & M_{sub}^{(i)} \end{bmatrix}.$$
4. *Obtain* $a \in \mathbb{R}$, $\boldsymbol{b} \in \mathbb{R}^{n_z-i}$ *and* $M_r \in \mathbb{R}^{(n_z-i)\times(n_z-i)}$ *with* $M_{sub}^{(i)} = \begin{bmatrix} a & \boldsymbol{b}^T \\ \boldsymbol{b} & M_r \end{bmatrix}$.
5. *Compute* $\alpha = \sqrt{\|\boldsymbol{b}\|_1}$.
6. *If* $\alpha = 0$ *then* $\boldsymbol{\varphi}_i = \mathbf{0}_{n_z \times 1}$, *else* $\boldsymbol{\varphi}_i = \begin{bmatrix} \mathbf{0}^T_{(i-1)\times 1} & \alpha & \frac{-\boldsymbol{b}^T}{\alpha} \end{bmatrix}^T$.
7. *Partially diagonalize* $\hat{\Gamma}^{(i)}$ *with* $\hat{\Gamma}^{(i)} = \hat{\Gamma}^{(i-1)} + \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T$.
8. *Endfor*
9. *Set* $\hat{\Gamma} = \hat{\Gamma}^{(n_z-1)}$ *and compute the upper bound* $\hat{\Gamma}_u(M) = \text{trace}(\hat{\Gamma})$.

It is evident from the Procedure 6.1 that $\hat{\Gamma}$ is a diagonal matrix with $\hat{\Gamma} \geq M(\boldsymbol{v})$ and, as a consequence, the expression

$$\max_{\|\boldsymbol{z}\|_\infty \leq 1} \boldsymbol{z}^T M(\boldsymbol{v}) \boldsymbol{z} \leq \max_{\|\boldsymbol{z}\|_\infty \leq 1} \boldsymbol{z}^T \hat{\Gamma} \boldsymbol{z} = \text{trace}(\hat{\Gamma}) = \hat{J}^*(x, \boldsymbol{v}). \tag{6.43}$$

is satisfied. Using (6.43) in (6.16), the following statement for the worst case cost can be made:

$$\begin{aligned} J^*(x, \boldsymbol{v}) &= \max_{\|\boldsymbol{z}\|_\infty \leq 1} \boldsymbol{z}^T \begin{bmatrix} \varepsilon^2 H & \varepsilon \boldsymbol{q}(x, \boldsymbol{v}) \\ \varepsilon \boldsymbol{q}(x, \boldsymbol{v})^T & J(x, \boldsymbol{v}, 0) \end{bmatrix} \boldsymbol{z} \\ &\leq \hat{\Gamma}_u \left( \begin{bmatrix} \varepsilon^2 H & \varepsilon \boldsymbol{q}(x, \boldsymbol{v}) \\ \varepsilon \boldsymbol{q}(x, \boldsymbol{v})^T & J(x, \boldsymbol{v}, 0) \end{bmatrix} \right) \end{aligned} \tag{6.44}$$

[3]Note that for $i = 1$ the matrices $\Upsilon^{(i)} \in \mathbb{R}^{(i-1)\times(i-1)}$ and $\mathbf{0}_{(n_z+1-i)\times(i-1)}$ are empty, i.e. $M_{sub}^{(1)} = \hat{\Gamma}^{(0)}$.

As a consequence $\hat{J}^*(x, \boldsymbol{v})$ calculated in the Procedure 6.1 can be considered as an upper bound of the worst case cost.

The idea of Procedure 6.1 is to find a diagonal matrix $\hat{\Gamma}$ such that $\hat{\Gamma} \geq M(\boldsymbol{v}) \geq 0$. The maximization problem (6.22) is solved using $\hat{J}^*(x, \boldsymbol{v}) = \text{trace}(\hat{\Gamma})$. Assume that in a certain iteration $i$ of the procedure the partially diagonalized matrix $\hat{\Gamma}^{(i)} \geq M(\boldsymbol{v})$ has all elements of the submatrix $M_{sub}^{(i)}$ of step 3 nonnegative. Then

$$\max_{\|\boldsymbol{z}\|_\infty \leq 1} \boldsymbol{z}^T \hat{\Gamma}^{(i)} \boldsymbol{z} = \|\hat{\Gamma}^{(i)}\|_s \geq J^*(x, \boldsymbol{v}) \tag{6.45}$$

with $\|\hat{\Gamma}^{(i)}\|_s$ denoting the sum of the absolute values of all entries of matrix $\hat{\Gamma}^{(i)}$. Therefore it is not necessary to continue the diagonalization process as the continuation of the diagonalization would not improve the bound. Hence, the matrix $\hat{\Gamma}$ can be defined $\hat{\Gamma} = \hat{\Gamma}^{(i)}$ and the upper bound becomes $\hat{J}^*(x, \boldsymbol{v}) = \|\hat{\Gamma}\|_s$.

Note that for the calculation of the upper bound only simple matrix operations are required. Thus, the algorithm can easily be implemented, even with programming languages not destined for mathematical calculations, commonly found in industrial embedded systems. In order to illustrate the algorithm, Fig. 6.1 shows the implementation of the procedure in Matlab. The case that all elements of the submatrix $M_{sub}^{(i)}$ are nonnegative has been considered in the programme.

### 6.2.3 Control strategy using the upper bound

The upper bound calculated with Procedure 6.1 is used in a MMMPC strategy where the input correction sequence is the solution of the problem:

$$\begin{aligned} \hat{\boldsymbol{v}}^* = \arg \min_{\boldsymbol{v}} \hat{J}^*(x, \boldsymbol{v}) \\ \text{s.t. } M_x x + M_v \boldsymbol{v} \leq \boldsymbol{b}_\varepsilon \end{aligned} \tag{6.46}$$

Finally, the input correction sequence, calculated with the worst case cost $\hat{J}^*(x, \boldsymbol{v})$ (see Procedure 6.1) based on $H$ (6.17), $\boldsymbol{q}(\boldsymbol{v}, x)$ (6.18) and $J(x, \boldsymbol{v}, 0)$ (6.19), can be used in a receding horizon control strategy with the input defined by $\hat{K}_{MPC}(x(k)) = -Kx(k) + \hat{v}^*(k|k)$.

Note that each of the $n_z - 1$ iterations carried out by the Procedure 6.1 has a computational complexity of $O(n_z^2)$, leading to a computational complexity of $O(n^3)$ of the procedure. Hence, the number of necessary operations is approximately $(Nn_\theta + 1)^3$ for the matrix $M(\boldsymbol{v})$ with size $n_z = Nn_\theta + 1$. In contrast, the minimization of

```
function J_hat = upper_bound(M)

n_z = size(M,1);
S = M;
i=0;
positive = 0;

while (positive == 0) & (i<n_z)
    i = i+1;

    M_sub = S(i:end,i:end);
    a = M_sub(1,1);
    b = M_sub(2:end,1);
    alfa = sqrt(sum(abs(b)));

    if alfa == 0
        phi = zeros(n_z,1);
    else
        phi = [zeros(i-1,1); alfa; -b/alfa];
    end

    S = S + phi*phi';

    if min(min(S))>=0
        positive = 1;
    end
end

J_hat = sum(sum(S));
```

**Figure 6.1**: Matlab code to calculate the upper bound of the worst case cost as proposed in the Procedure 6.1.

the original min-max problem (6.20) requires approximately $2^{Nn_\theta+1}$ operations. That means that the use of the proposed upper bound reduces considerably the number of necessary operations and allows the evaluation of the cost function in polynomial time.

The minimization of the cost function, which corresponds to the computation of the input correction sequence, can be carried out with any nonlinear programming method, e.g. sequential quadratic programming (SQP). It has to be mentioned that the NP-hard complexity of (6.15) is a consequence of the maximization problem to be solved for every candidate input correction sequence. Hence, the substitution of the

maximization by an upper bound of the worst case cost results in a lower computational load.

### 6.2.4 Stability of the min-max MPC

In the first place, some necessary properties to assure stability of the proposed control strategy will be presented. Consider the solutions $\boldsymbol{v}^*$, $\tilde{\boldsymbol{v}}^*$ and $\hat{\boldsymbol{v}}^*$ of the minimization problems (6.20), (6.28) and (6.46) respectively. Denote also $J^*(x) = J^*(x, \boldsymbol{v}^*)$, $\tilde{J}^*(x) = \tilde{J}^*(x, \tilde{\boldsymbol{v}}^*)$ and $\hat{J}^*(x) = \hat{J}^*(x, \hat{\boldsymbol{v}}^*)$. Note that the optimization problems (6.20), (6.28) and (6.46) have the same feasibility region as the used constraints are the same.

**Property 6.2** *The minimum costs $\hat{J}^*(x)$ and $\tilde{J}^*(x)$ of the optimization problems (6.46) and (6.28) satisfy:*

$$\hat{J}^*(x) \leq \tilde{J}^*(x) \tag{6.47}$$

*when $\tilde{\boldsymbol{v}}^*$ is used as initial solution to the optimization problem (6.46).*

**Proof:** Taking into account the definition of $\hat{J}^*(x)$ and that $\hat{\boldsymbol{v}}^*$ is the minimizer of $\hat{J}^*(x, \boldsymbol{v})$ it is evident that

$$\hat{J}^*(x, \tilde{\boldsymbol{v}}^*) \geq \hat{J}^*(x) \tag{6.48}$$

Thus, in order to prove that $\hat{J}^*(x) \leq \tilde{J}^*(x)$ it suffices to show that $\tilde{J}^*(x, \tilde{\boldsymbol{v}}^*) \geq \hat{J}^*(x, \tilde{\boldsymbol{v}}^*)$. First, note that taking into account that $J(x, \boldsymbol{v}, 0) \geq 0$:

$$\begin{aligned} \tilde{J}^*(x, \tilde{\boldsymbol{v}}^*) &= \|M(\tilde{\boldsymbol{v}}^*)\|_1 = \left\| \begin{bmatrix} \varepsilon^2 H & \varepsilon \boldsymbol{q}(x, \tilde{\boldsymbol{v}}^*) \\ \varepsilon \boldsymbol{q}(x, \tilde{\boldsymbol{v}}^*)^T & J(x, \tilde{\boldsymbol{v}}^*, 0) \end{bmatrix} \right\|_1 \\ &= \left\| \begin{bmatrix} a & \boldsymbol{b}^T \\ \boldsymbol{b} & M_r \end{bmatrix} \right\|_1 \\ &= |a| + 2\|\boldsymbol{b}\|_1 + \|M_r\|_1 \end{aligned} \tag{6.49}$$

On the other hand $\hat{J}^*(x, \tilde{\boldsymbol{v}}^*)$ is equal to trace$(\hat{\Gamma})$, that is the sum of the elements of the diagonal matrix computed in Procedure 6.1 which also is equal to $\|\hat{\Gamma}\|_1$ as $\hat{\Gamma} \geq 0$. The initial value of $\hat{\Gamma}$ is $\hat{\Gamma}^{(0)} = M(\tilde{\boldsymbol{v}}^*)$, thus its 1-norm is equal to $\tilde{J}^*(x, \tilde{\boldsymbol{v}}^*)$. Taking into account (6.41) the 1-norm of $\hat{\Gamma}^{(1)}$ after the first diagonalization step is

$$\left\| \begin{matrix} a + \|\boldsymbol{b}\|_1 & 0 \\ 0 & M_r + \frac{\boldsymbol{b}\boldsymbol{b}^T}{\|\boldsymbol{b}\|_1} \end{matrix} \right\|_1 \leq |a| + \|\boldsymbol{b}\|_1 + \|M_r\|_1 + \left\| \frac{\boldsymbol{b}\boldsymbol{b}^T}{\|\boldsymbol{b}\|_1} \right\|_1 \tag{6.50}$$

Taking into account that $\left\| \frac{\boldsymbol{b}\boldsymbol{b}^T}{\|\boldsymbol{b}\|_1} \right\|_1 = \|\boldsymbol{b}\|_1$ it follows that

$$\left\| \begin{array}{cc} a + \|\boldsymbol{b}\|_1 & 0 \\ 0 & M_r + \frac{\boldsymbol{b}\boldsymbol{b}^T}{\|\boldsymbol{b}\|_1} \end{array} \right\|_1 \leq \|M(\tilde{\boldsymbol{v}}^*)\|_1 \tag{6.51}$$

and thus every diagonalization step decreases $\|\hat{\Gamma}^{(1)}\|_1$. This proves that:

$$\hat{J}^*(x, \tilde{\boldsymbol{v}}^*) \leq \tilde{J}^*(x, \tilde{\boldsymbol{v}}^*) \tag{6.52}$$

and this completes the proof. □

It is clear that the optimal solution $\hat{\boldsymbol{v}}^*$ of problem (6.46) is a suboptimal feasible solution for problem (6.20). As it is claimed in the following property, the difference between the optimal value of the original objective function and the value obtained with $\hat{\boldsymbol{v}}^*$ is bounded by $\text{trace}(T)\varepsilon^2$.

**Property 6.3** *It holds that:*

$$J^*(x, \hat{\boldsymbol{v}}^*) - \text{trace}(T)\varepsilon^2 \leq J^*(x) \tag{6.53}$$

**Proof:** Note that $J^*(x) = J^*(x, \boldsymbol{v}^*)$. On the other hand:

$$\tilde{J}(x, \boldsymbol{v}, \boldsymbol{\theta}) = J(x, \boldsymbol{v}, \boldsymbol{\theta}) + \boldsymbol{\theta}^T(T - H)\boldsymbol{\theta}. \tag{6.54}$$

Taking into account that $T \geq H \geq 0$, $\|\boldsymbol{\theta}\|_\infty \leq \varepsilon$ and that $T$ is diagonal matrix, the statement:

$$\tilde{J}(x, \boldsymbol{v}, \boldsymbol{\theta}) \leq J(x, \boldsymbol{v}, \boldsymbol{\theta}) + \boldsymbol{\theta}^T T\boldsymbol{\theta} \leq J(x, \boldsymbol{v}, \boldsymbol{\theta}) + \text{trace}(T)\varepsilon^2 \tag{6.55}$$

is true. From (6.55) it can be inferred that:

$$J^*(x, \boldsymbol{v}^*) \geq \tilde{J}^*(x, \boldsymbol{v}^*) - \text{trace}(T)\varepsilon^2 \tag{6.56}$$

Then, with $\tilde{\boldsymbol{v}}^*$ being the minimizer of $\tilde{J}^*(x, \boldsymbol{v})$ and $J^*(x) = J^*(x, \boldsymbol{v}^*)$, (6.56) can be rewritten as:

$$J^*(x) \geq \tilde{J}^*(x) - \text{trace}(T)\varepsilon^2 \tag{6.57}$$

Now, from the relation $\hat{J}^*(x) \leq \tilde{J}^*(x)$ (see Property 6.2) and (6.57) follows that:

$$J^*(x) \geq \hat{J}^*(x) - \text{trace}(T)\varepsilon^2 \tag{6.58}$$

Finally, with the upper bound $\hat{J}^*(x, \boldsymbol{v}) \geq J^*(x, \boldsymbol{v})$ and (6.58), the statement for the optimal cost:

$$J^*(x) \geq J^*(x, \hat{\boldsymbol{v}}^*) - \text{trace}(T)\varepsilon^2 \tag{6.59}$$

is satisfied. This completes the proof. □

The following property, which is proven in [1] will be used in the proof of the stability of the proposed approach (see Theorem 6.5 below).

**Property 6.4** *Consider that assumptions C1, C2 and C3 are satisfied. Let* $\boldsymbol{v} = [v(k|k), v(k+1|k), \ldots, v(k+N-1|k)]^T$ *and* $\boldsymbol{v}_s$ *a shifted version of* $\boldsymbol{v}$ *computed as* $\boldsymbol{v}_s = [v(k+1|k), v(k+2|k), \ldots, v(k+N-1|k), 0]^T$. *If* $\boldsymbol{v}$ *is feasible for problem (6.20) at* $x(k)$ *then* $\boldsymbol{v}_s$ *is also feasible at* $x(k+1)$ *and there is a* $\gamma > 0$ *such that for every feasible sequence* $\boldsymbol{v}$*:*

$$J^*(x(k+1), \boldsymbol{v}_s) \leq J^*(x(k), \boldsymbol{v}) - x(k)^T Q x(k) + \gamma\varepsilon^2 \tag{6.60}$$

**Proof:** See [1] for a proof. □

**Theorem 6.5** *Under the assumption that the conditions C1, C2 and C3 are satisfied, the control law given by* $\hat{K}_{MPC}(x(k)) = -Kx(k) + \hat{v}^*(k|k)$ *stabilizes system (6.2).*

**Proof:** Consider the input sequence $\hat{\boldsymbol{v}}_s^*$ being a shifted version (as in Property 6.4) of $\hat{\boldsymbol{v}}^*$. Due to non-optimality of $\hat{\boldsymbol{v}}_s^*$ for problem (6.20) it holds that

$$J^*(x(k+1)) \leq J^*(x(k+1), \hat{\boldsymbol{v}}_s^*) \tag{6.61}$$

Note that $\hat{\boldsymbol{v}}_s^*$ is feasible for both (6.46) and (6.20), thus by property 6.4:

$$J^*(x(k+1), \hat{\boldsymbol{v}}_s^*) \leq J^*(x(k), \hat{\boldsymbol{v}}^*) - x(k)^T Q x(k) + \gamma\varepsilon^2 \tag{6.62}$$

Furthermore, by Property 6.3 it holds that $J^*(x(k), \hat{\boldsymbol{v}}^*) \leq J^*(x(k)) + \text{trace}(T)\varepsilon^2$, thus taking into account this in (6.62) and using (6.61) $J^*(x(k+1)) \leq J^*(x(k)) - x(k)^T Q x(k) + (\gamma + \text{trace}(T))\varepsilon^2$ which can be rewritten:

$$J^*(x(k+1)) - J^*(x(k)) \leq -x(k)^T Q x(k) + (\gamma + \text{trace}(T))\varepsilon^2 \tag{6.63}$$

and leads to the set:

$$\Phi_\varepsilon = \left\{ x \in \mathbb{R}^n : (6.20) \text{ is feasible and } x(k)^T Q x(k) \leq (\gamma + \text{trace}(T))\varepsilon^2 \right\} \tag{6.64}$$

containing the origin. As a consequence, the system state is steered into $\Phi_\varepsilon$ from any arbitrary $x(k)$. After entering the set $\Phi_\varepsilon$ the state can remain inside or leave the set as it is not guaranteed that the optimal worst case cost decreases. Under consideration of $-x(k)^T Q x(k) \leq 0$ and (6.63), the cost in $k+1$ satisfies:

$$J^*(x(k+1)) \leq J^*(x(k)) + (\gamma + \text{trace}(T))\varepsilon^2 \tag{6.65}$$

Now, for every $x(k) \in \Phi_\varepsilon$ holds:

$$J(x(k)) + (\gamma + \text{trace}(T))\varepsilon^2 \leq \max_{x \in \Phi_\varepsilon} J(x) + (\gamma + \text{trace}(T)) = \beta \tag{6.66}$$

It is clear from (6.65) and (6.66) that for every $x(k) \in \Phi_\varepsilon$ the cost in $k+1$ satisfies:

$$J(x^*(k+1)) \leq \beta \tag{6.67}$$

As a consequence, whenever the state enters the set $\Phi_\varepsilon$ the system will evolve into the set:

$$\Omega_\beta = \{x \in \mathbb{R}^n : J^*(x) \leq \beta\} \tag{6.68}$$

Although the state can leave the set $\Phi_\varepsilon$, it will remain inside the set $\Omega_\beta$. With the state being in $\Omega_\beta$, the system is steered again and again into $\Phi_\varepsilon$. Hence, the state is ultimately bounded and the system is stabilized by the control law $\hat{K}_{MPC}(x(k)) = -Kx(k) + \hat{v}^*(k|k)$. □

## 6.3 MMMPC with a quadratic upper bound of the worst case cost

This section presents a MMMPC strategy, based on [3, 48], that replaces the optimization problem (6.20) by a tractable quadratic programming (QP) problem, providing a close approximation of the solution of the original problem. The use of an upper bound for the worst case cost leads, as in section (6.2), to a reduced computational complexity and enables the use of larger prediction horizons. The substitution of the NP-hard problem by a QP problem can be accomplished by the following steps:

1. Compute a candidate input correction sequence $\tilde{\boldsymbol{v}}^*$ by solving the QP problem (6.28) based on a simple upper bound of the worst case cost.

2. A quadratic function of $\boldsymbol{v}$ that bounds the worst case cost can be obtained using $\tilde{\boldsymbol{v}}^*$.

3. Calculate the input correction $\hat{\boldsymbol{v}}^*$ sequence by solving a QP problem.

The computation of the candidate input sequence $\tilde{\boldsymbol{v}}^*$ can be carried out as shown in Section 6.1.2. The computation of an upper bound of the worst case cost being a quadratic function of $\boldsymbol{v}$ and its use in a control strategy are explained in the following sections. The contribution of this thesis is the practical implementation of presented control strategy based on the theoretical developments in [3]. In Section 6.4.2 the MMMPC strategy will be used to control a benchmark system and the control performance will be illustrated by experimental results.

### 6.3.1 Quadratic upper bound of the worst case cost

The augmented maximization problem defined in (6.22) can be used to calculate an upper bound of the worst case cost. The procedure to compute the bound, similar to Procedure 6.1 used in Section 6.2.2, diagonalizes the matrix $M(\boldsymbol{v})$ for a given input correction sequence. During the procedure a set of parameters $\boldsymbol{\alpha}(\boldsymbol{v}) = [\alpha_1(\boldsymbol{v}), \ldots, \alpha_{n_z-1}(\boldsymbol{v}]^T$ is calculated. These parameters can later be used to substitute the original optimization problem by a QP problem, using an upper bound of the worst case cost close to the optimal worst case cost. The procedure to calculate the parameters is the following:

**Procedure 6.6** *Computation of the parameters* $\boldsymbol{\alpha}(\boldsymbol{v}) = [\alpha_1(\boldsymbol{v}), \ldots, \alpha_{n_z-1}(\boldsymbol{v})]^T$ *and the upper bound* $\Gamma(\boldsymbol{v})$ *of the worst case cost.*

1. *Set* $\Gamma^{(0)} = M(\boldsymbol{v}) \in \mathbb{R}^{n_z \times n_z}$.
2. *For* $i = 1$ *to* $n_z - 1$
3. *Obtain* $M_{sub}^{(i)} \in \mathbb{R}^{(n_z+1-i)\times(n_z+1-i)}$ *and* $\Upsilon^{(i)} \in \mathbb{R}^{(i-1)\times(i-1)}$ *with*[4]

$$\Gamma^{(i-1)} = \begin{bmatrix} \Upsilon^{(i)} & \mathbf{0}^T_{(n_z+1-i)\times(i-1)} \\ \mathbf{0}_{(n_z+1-i)\times(i-1)} & M_{sub}^{(i)} \end{bmatrix}.$$

4. *Obtain* $a \in \mathbb{R}$, $\boldsymbol{b} \in \mathbb{R}^{n_z-i}$ *and* $M_r \in \mathbb{R}^{(n_z-i)\times(n_z-i)}$ *with* $M_{sub}^{(i)} = \begin{bmatrix} a & \boldsymbol{b}^T \\ \boldsymbol{b} & M_r \end{bmatrix}$.
5. *Compute* $\alpha_i(\boldsymbol{v}) = \sqrt{\|\boldsymbol{b}\|_1}$.

[4]Note that for $i = 1$ the matrices $\Upsilon^{(i)} \in \mathbb{R}^{(i-1)\times(i-1)}$ and $\mathbf{0}_{(n_z+1-i)\times(i-1)}$ are empty, i.e. $M_{sub}^{(1)} = \Gamma^{(0)}$.

6. *If* $\alpha_i(\boldsymbol{v}) = 0$ *then* $\boldsymbol{\varphi}_i = \mathbf{0}_{n_z \times 1}$, *else* $\boldsymbol{\varphi}_i = \left[ \begin{array}{ccc} \mathbf{0}^T_{(i-1)\times 1} & \alpha_i(\boldsymbol{v}) & \frac{-\boldsymbol{b}^T}{\alpha_i(\boldsymbol{v})} \end{array} \right]^T$.

7. *Partially diagonalize* $\Gamma^{(i)}$ *with* $\Gamma^{(i)} = \Gamma^{(i-1)} + \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T$.

8. *Endfor*

9. *Set* $\Gamma(\boldsymbol{v}) = \Gamma^{(n_z-1)}$ *and compute the upper bound* $\Gamma_u(M) = \text{trace}(\Gamma)$.

Analogously to Procedure 6.1, the matrix $\Gamma(\boldsymbol{v})$ computed in Procedure 6.6 satisfies $\Gamma(\boldsymbol{v}) \geq M(\boldsymbol{v})$. Therefore, the statement:

$$J^*(x, \boldsymbol{v}) = \max_{\|\boldsymbol{z}\|_\infty \leq 1} \boldsymbol{z}^T M(\boldsymbol{v}) \boldsymbol{z} \leq \max_{\|\boldsymbol{z}\|_\infty \leq 1} \boldsymbol{z}^T \Gamma(\boldsymbol{v}) \boldsymbol{z} = \text{trace}(\Gamma(\boldsymbol{v})) \tag{6.69}$$

holds. Furthermore, based on the Property 6.2 it can be shown that:

$$\text{trace}(\Gamma(\boldsymbol{v})) \leq \tilde{J}^*(x, \boldsymbol{v}) \tag{6.70}$$

As a consequence $\text{trace}(\Gamma(\boldsymbol{v}))$ calculated in the Procedure 6.6 can be considered as an improved upper bound of $J^*(x, \boldsymbol{v})$ with respect to the upper bound $\tilde{J}^*(x, \boldsymbol{v})$ obtained in section 6.1.2.

Procedure 6.7 is the foundation to obtain an upper bound of the worst case cost being a quadratic function of $\boldsymbol{v}$ close to the optimal worst case cost but with the advantage of the lower computational burden of a QP problem. Therefore, the candidate input correction sequence $\tilde{\boldsymbol{v}}^*$ from Section 6.1.2 is used in Procedure 6.6 to compute the parameter vector $\boldsymbol{\alpha}(\tilde{\boldsymbol{v}}^*)$. This parameter vector can then be used to obtain a quadratic function $\hat{\Gamma}(\boldsymbol{v})$ representing an upper bound of the worst case cost. This is achieved by the following procedure:

**Procedure 6.7** *Computation of a quadratic function* $\hat{\Gamma}(\boldsymbol{v})$ *representing an upper bound of the worst case cost.*

1. *Obtain the candidate input correction sequence* $\tilde{\boldsymbol{v}}^*$ *from the QP problem defined in (6.1.2).*

2. *Compute* $\boldsymbol{\alpha}(\tilde{\boldsymbol{v}}^*)$ *by Procedure 6.6.*

3. *Set* $\hat{\Gamma}^{(0)}(\boldsymbol{v}) = M(\boldsymbol{v}) \in \mathbb{R}^{n_z \times n_z}$.

4. *For* $i = 1$ *to* $n_z - 1$

5. *Obtain* $M_{sub}^{(i)} \in \mathbb{R}^{(n_z+1-i)\times(n_z+1-i)}$ *and* $\Upsilon^{(i)} \in \mathbb{R}^{(i-1)\times(i-1)}$ *with*[5]

$$\hat{\Gamma}^{(i-1)}(\boldsymbol{v}) = \begin{bmatrix} \Upsilon^{(i)} & \mathbf{0}^T_{(n_z+1-i)\times(i-1)} \\ \mathbf{0}_{(n_z+1-i)\times(i-1)} & M^{(i)}_{sub}(\boldsymbol{v}) \end{bmatrix}.$$

6. *Obtain* $a(\boldsymbol{v}) \in \mathbb{R}$, $\boldsymbol{b}(\boldsymbol{v}) \in \mathbb{R}^{n_z-i}$ *and* $M_r(\boldsymbol{v}) \in \mathbb{R}^{(n_z-i)\times(n_z-i)}$ *with*

$$M^{(i)}_{sub}(\boldsymbol{v}) = \begin{bmatrix} a(\boldsymbol{v}) & \boldsymbol{b}(\boldsymbol{v})^T \\ \boldsymbol{b}(\boldsymbol{v}) & M_r(\boldsymbol{v}) \end{bmatrix}.$$

7. *If* $\alpha_i(\tilde{\boldsymbol{v}}^*) = 0$ *then* $\boldsymbol{\varphi}_i = \mathbf{0}_{n_z\times 1}$, *else* $\boldsymbol{\varphi}_i = [\mathbf{0}^T_{(i-1)\times 1} \;\; \alpha_i(\tilde{\boldsymbol{v}}^*) \;\; \frac{-\boldsymbol{b}^T}{\alpha_i(\tilde{\boldsymbol{v}}^*)}]^T$.

8. *Compute* $\hat{\Gamma}^{(i)}(\boldsymbol{v})$ *with* $\hat{\Gamma}^{(i)}(\boldsymbol{v}) = \hat{\Gamma}^{(i-1)}(\boldsymbol{v}) + \boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^T$.

9. *Endfor*

10. *Set* $\hat{\Gamma}(\boldsymbol{v}) = \hat{\Gamma}^{(n_z-1)}(\boldsymbol{v})$ *and compute the upper bound* $\hat{\Gamma}_u(M(\boldsymbol{v})) = \mathrm{trace}(\hat{\Gamma}(\boldsymbol{v}))$.

Analogously to Procedure 6.1, the diagonal matrix $\hat{\Gamma}(\boldsymbol{v})$ can be considered as an upper bound of the worst case cost $J^*(x, \boldsymbol{v})$ as the Procedure guarantees $\hat{\Gamma}(\boldsymbol{v}) \geq M(\boldsymbol{v})$. Hence, the following relation:

$$\max_{\|\boldsymbol{z}\|_\infty\leq 1} \boldsymbol{z}^T M(\boldsymbol{v})\boldsymbol{z} \leq \max_{\|\boldsymbol{z}\|_\infty\leq 1} \boldsymbol{z}^T\hat{\Gamma}(\boldsymbol{v})\boldsymbol{z} = \mathrm{trace}(\hat{\Gamma}(\boldsymbol{v})) = \hat{J}^*(x, \boldsymbol{v}) \tag{6.71}$$

is satisfied. Furthermore, based on Property 6.2, $\hat{J}^*(x, \boldsymbol{v})$ can be considered as an improved upper bound in comparison to (6.28), that means:

$$\hat{J}^*(x, \boldsymbol{v}) \leq \tilde{J}^*(x, \boldsymbol{v}) \tag{6.72}$$

It is important to mention that the upper bound of the worst case cost $\hat{J}^*(x, \boldsymbol{v})$ is a quadratic function of the input correction sequence $\boldsymbol{v}$ [3]. Therefore, consider the used matrix $M(\boldsymbol{v})$ (6.23) to have the following form:

$$M(\boldsymbol{v}) = \begin{bmatrix} \varepsilon^2 H_{11} & \varepsilon^2 H_{1r}^T & \varepsilon q_1(x, \boldsymbol{v}) \\ \varepsilon^2 H_{1r} & \varepsilon^2 H_{rr} & \varepsilon q_r(x, \boldsymbol{v}) \\ \varepsilon q_1(x, \boldsymbol{v}) & \varepsilon q_r(x, \boldsymbol{v})^T & J(x, \boldsymbol{v}, 0) \end{bmatrix} \tag{6.73}$$

with $H_{11} \in \mathbb{R}$, $q_1(x, \boldsymbol{v}) \in \mathbb{R}$, $J(x, \boldsymbol{v}, 0) \in \mathbb{R}$, $H_{1r} \in \mathbb{R}^{n_z-2}$, $q_r(x, \boldsymbol{v}) \in \mathbb{R}^{n_z-2}$ and $H_{rr} \in \mathbb{R}^{(n_z-2)\times(n_z-2)}$. Note that $q_1(x, \boldsymbol{v})$ and $q_r(x, \boldsymbol{v})$ are affine functions in $\boldsymbol{v}$ and that

[5] Note that for $i = 1$ the matrices $\Upsilon^{(i)} \in \mathbb{R}^{(i-1)\times(i-1)}$ and $\mathbf{0}_{(n_z+1-i)\times(i-1)}$ are empty, i.e. $M^{(1)}_{sub}(\boldsymbol{v}) = \hat{\Gamma}^{(0)}(\boldsymbol{v})$.

the nominal cost $J(x, \boldsymbol{v}, 0)$ is a quadratic function of $\boldsymbol{v}$. Furthermore, $M(\boldsymbol{v})$ is used used as initial matrix in Procedure 6.7, that is $\hat{\Gamma}^{(0)}(\boldsymbol{v}) = M(\boldsymbol{v})$. Now, using the parameters $\alpha_i(\tilde{\boldsymbol{v}}^*)$ calculated in Procedure 6.6, the matrix $\hat{\Gamma}^{(0)}(\boldsymbol{v})$ can be partially diagonalized with Procedure 6.7. With $\alpha_1(\tilde{\boldsymbol{v}}^*)$, (6.73) and Step 7 of Procedure 6.7 the vector[6]:

$$\boldsymbol{\varphi}_1^T = \left[ \begin{array}{ccc} \alpha_1 & -\dfrac{\varepsilon^2 H_{1r}^T}{\alpha_1} & -\dfrac{\varepsilon q_1(x, \boldsymbol{v})}{\alpha_1} \end{array} \right] \tag{6.74}$$

is obtained. Then, based on $\boldsymbol{\varphi}_1$ and Step 8 of Procedure 6.7, the new matrix $\hat{\Gamma}^{(1)}(\boldsymbol{v})$ can be written as:

$$\hat{\Gamma}^{(1)}(\boldsymbol{v}) = \begin{bmatrix} \varepsilon^2 H_{11} + \alpha_1^2 & 0 & 0 \\ 0 & \varepsilon^2 H_{rr} + \frac{\varepsilon^4 H_{1r}^T H_{1r}}{\alpha_1^2} & \varepsilon q_r(x, \boldsymbol{v}) + \frac{\varepsilon^3 H_{1r}^T q_1(x, \boldsymbol{v})}{\alpha_1^2} \\ 0 & \varepsilon q_r(x, \boldsymbol{v})^T + \frac{\varepsilon^3 H_{1r} q_1(x, \boldsymbol{v})}{\alpha_1^2} & J(x, \boldsymbol{v}, 0) + \frac{\varepsilon^2 q_{1]2}(x, \boldsymbol{v})^2}{\alpha_1^2} \end{bmatrix} \tag{6.75}$$

Note that $\hat{\Gamma}^{(1)}(\boldsymbol{v})$ has been partially diagonalized and that the element $\varepsilon^2 H_{11} + \alpha_1(\tilde{\boldsymbol{v}}^*)^2$ is not a function of $\boldsymbol{v}$ but a constant. Then, using the parameters $\alpha_i(\tilde{\boldsymbol{v}}^*)$ with $i = 2, \ldots, n_z - 1$ in the Procedure 6.7 results in the diagonal matrix $\hat{\Gamma}(\boldsymbol{v}) = \hat{\Gamma}^{(n_z-1)}(\boldsymbol{v})$ given by:

$$\hat{\Gamma}^{(n_z-1)}(\boldsymbol{v}) = \begin{bmatrix} \hat{\Gamma}^{[11]} & 0 & \ldots & 0 & 0 \\ 0 & \hat{\Gamma}^{[22]} & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \hat{\Gamma}^{[n_z-1, n_z-1]} & 0 \\ 0 & 0 & 0 & 0 & \hat{\Gamma}^{[n_z n_z]}(\boldsymbol{v}) \end{bmatrix} \tag{6.76}$$

where the elements $\hat{\Gamma}^{[ii]}$ with $i = 1, \ldots, n_z - 1$ are constants and $\hat{\Gamma}^{[n_z n_z]}(\boldsymbol{v})$ is a quadratic function of $\boldsymbol{v}$ defined as:

$$\begin{aligned} \hat{\Gamma}^{[n_z n_z]}(\boldsymbol{v}) \;=\; & J(x, \boldsymbol{v}, 0) + \frac{\varepsilon^2 q_1(x, \boldsymbol{v})^2}{\alpha_1^2} + \frac{\varepsilon^2 q_2(x, \boldsymbol{v})^2}{\alpha_2^2} + \\ & \frac{\varepsilon^4 H_{12}^2 q_1(x, \boldsymbol{v})^2}{\alpha_1^4 \alpha_2^2} + \frac{\varepsilon^4 H_{12} q_1(x, \boldsymbol{v}) q_2(x, \boldsymbol{v})}{\alpha_1^2 \alpha_2^2} + \ldots \end{aligned} \tag{6.77}$$

Hence, after obtaining the diagonal matrix $\hat{\Gamma}(\boldsymbol{v})$ by means of the Procedure 6.7 the upper bound of the maximum can be computed as:

$$\hat{J}^*(x, \boldsymbol{v}) = \hat{\Gamma}^{[n_z n_z]}(\boldsymbol{v}) + \sum_{i=1}^{n_z-1} \hat{\Gamma}^{[ii]} \tag{6.78}$$

Note that the Procedure 6.7 to compute a quadratic upper bound of the worst case cost can be repeated several times, improving the approximation of the worst case cost,

[6]For the sake of simplicity, the notation $\alpha_i = \alpha_i(\tilde{\boldsymbol{v}}^*)$ for $i = 1, \ldots, n_z - 1$ will be used in the following equations.

i.e. the difference between upper bound and exact worst case cost is reduced. This improvement can be obtained by using the minimizer $\hat{\boldsymbol{v}}^*$ of (6.78) as a new candidate input correction sequence instead of (6.28). With this new candidate input correction sequence a new and improved upper bound of the worst case cost can be computed. Minimizing (6.78) a new input correction sequence is obtained which can be applied to the system or used as a new initial guess to compute an upper bound.

### 6.3.2 Control strategy using the upper bound

For the computation of the input correction sequence $\hat{\boldsymbol{v}}^*$ the upper bound $\hat{J}^*(x, \boldsymbol{v})$ of the worst case cost has to be minimized. As the upper bound $\hat{J}^*(x, \boldsymbol{v})$ can be expressed as a quadratic function of $\boldsymbol{v}$ (for details see Section 6.3.1), the value of the input correction sequence is obtained by solving the following QP optimization problem:

$$\begin{aligned} \hat{\boldsymbol{v}}^* = \quad \arg \quad & \min_{\boldsymbol{v}} \hat{J}^*(x, \boldsymbol{v}) \\ & \text{s.t. } M_x x + M_v \boldsymbol{v} \leq \boldsymbol{b}_\varepsilon \end{aligned} \tag{6.79}$$

and the system is controlled by $\hat{K}_{MPC}(x(k)) = -Kx(k) + \hat{v}^*(k|k)$, where $\hat{v}^*(k|k)$ is the first element of $\hat{\boldsymbol{v}}^*$.

The computational burden of the proposed strategy is much lower than that of the exact MMMPC [3]. This computational burden is mostly due to the QP problems that must be solved to obtain the initial guess and the proposed solution itself. Note that in both cases, the complexity of each problem is the same as that of a standard constrained MPC using a quadratic cost function. In contrast to evaluate the maximum cost $J^*(x, \boldsymbol{v})$ it is necessary to evaluate the function for all the $2^{Nn_\theta}$ vertices of $\boldsymbol{\Theta}$ (because this is a well known NP-hard problem).

### 6.3.3 Stability of the min-max MPC

For the presented MMMPC strategy (see Section 6.3.2) with a quadratic upper bound of the worst case cost stability has been proven in [2, 3]. Generally, stability of the control strategy can be shown with the stability proof presented in Section 6.2.4. Therefore, consider the solutions $\boldsymbol{v}^*$, $\tilde{\boldsymbol{v}}^*$ and $\hat{\boldsymbol{v}}^*$ to the original optimization problem (6.20), the optimization problem based upon a simple upper bound (6.28) and the optimization problem using a quadratic upper bound of the worst case cost (6.79), respectively.

The minimized costs of the mentioned optimization problems are denoted generally

as $J^*(x) = J^*(x, \boldsymbol{v}^*)$, $\tilde{J}^*(x) = \tilde{J}^*(x, \tilde{\boldsymbol{v}}^*)$ and $\hat{J}^*(x) = \hat{J}^*(x, \hat{\boldsymbol{v}}^*)$. With the same linear constraints ($M_x x + M_v \boldsymbol{v} \leq \boldsymbol{b}_\varepsilon$) the three optimization problems possess the same feasibility region. With the proofs of the Property 6.2 and Property 6.3 it can be shown that $\hat{J}^*(x) \leq \tilde{J}^*(x)$ and $J^*(x, \hat{\boldsymbol{v}}^*) - \text{trace}(T)\varepsilon^2 \leq J^*(x)$, respectively. Then, with the feasible solution $\boldsymbol{v}_s$ satisfying the Property 6.4 it can be shown with the Theorem 6.5 that the MMMPC strategy based on the semi-feedback approach $\hat{K}_{MPC}(x(k)) = -Kx(k) + \hat{v}^*(k|k)$ is stable as a consequence of the ultimately bounded system state

## 6.4 Experimental results

The MMMPC strategies, based on a nonlinear upper bound (see Section 6.2) and a quadratic upper bound (see Section 6.3) of the worst case cost were validated in experiments with the pilot plant emulating an exothermic chemical reaction (see Section 3.1). For the implementation of the control strategies a suitable linear model had to be identified.

Based on the experimental data that has been used to identify a second order *Volterra* series model (see Fig. 3.7 in Section 3.1.3) and taking into account the response time of the system, the sampling period has been chosen to $t_s = 60\,\text{s}$. With the experimental data a least squares identification has been carried out and the following model has been identified:

$$y(k) = 0.941\, y(k-1) - 0.061\, u(k-1) \tag{6.80}$$

Thereby, the following input-output prediction model with integrated bounded additive uncertainty was obtained[7]:

$$y(k+1|k) = 0.941\, y(k) - 0.061\, u(k) + \frac{\theta(k|k)}{\Delta}. \tag{6.81}$$

Furthermore, based upon the one step ahead prediction error (see Fig. 6.2), the parameter $\varepsilon$ has been chosen to $\varepsilon = 0.4$. As a result, in 94% of the samples the one step ahead prediction error is bounded by the chosen value. In order to verify the goodness of fit of the identified model a second set of experimental data (see Fig. 3.9) has been used to calculate the one step ahead prediction error. In Fig. 6.3 it can be seen that the one step ahead prediction error for the second data set is bounded by $\varepsilon = 0.4$ nearly throughout the whole experiment, only in a few samples the prediction

[7]Note that the use of (6.81) instead of the state-space model (6.2) requires some minor changes in the implementation of the presented control strategies. These differences are explained in the Appendix A.7.

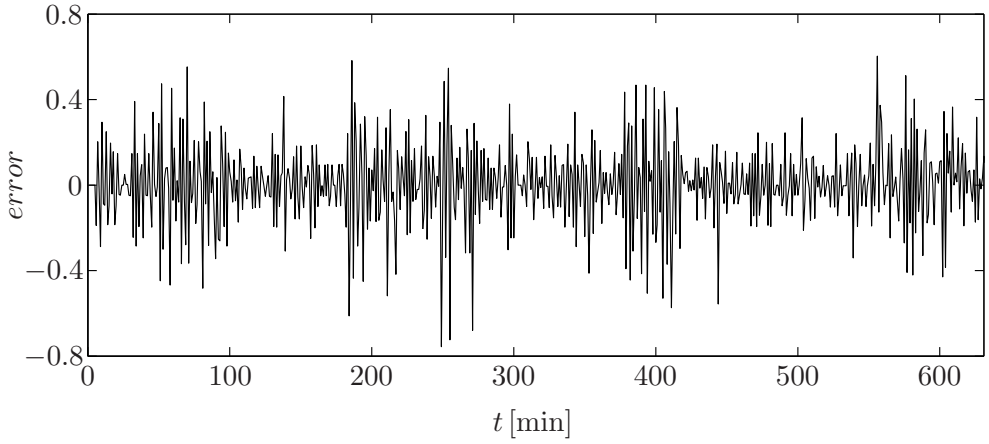


**Figure 6.2**: One step ahead prediction error during the experiment for the model identification.

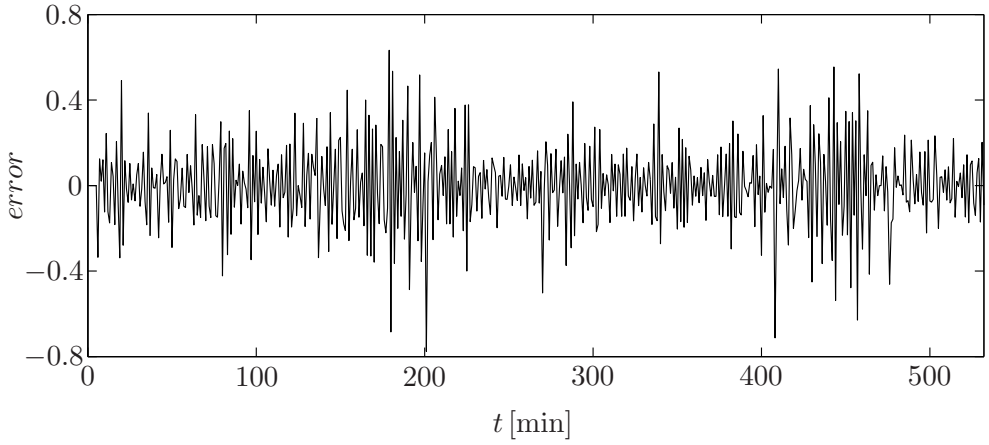


**Figure 6.3**: One step ahead prediction error during the experiment for the model validation.

error exceeds the bound. Therefore the verification confirms the election of $\varepsilon = 0.4$ as a valid choice.

For the implementation of the control strategies a prediction horizon of $N = 25$ and a control horizon of $N_u = 15$ were used. Note that the use of different prediction and control horizons ($N \neq N_u$) requires minor changes in the matrices $M_{vv}$, $M_{\theta v}$ and $M_{vf}$ in the quadratic cost function (6.14) as well as in the matrix $M_v$ in the considered min-max problem (6.15)[8]. In the implementation of the proposed control strategy the terminal constraint and the terminal cost have not been considered. With a prediction horizon of $N = 25$, including more than one time constant of the process, the terminal constraint

[8]Defining the variable $\sigma = N - N_u$ (the difference of the two horizons), the necessary adjustment of the matrices $M_{vv}$, $M_{\theta v}$, $M_{vf}$ and $M_v$ due to different prediction and control horizons leads to the elimination of the last $\sigma$ rows of $M_{vf}$, the last $\sigma$ columns of $M_v$ and $M_{\theta v}$ and the last $\sigma$ rows and columns of $M_{vv}$.

is not active for the region of interest. Also, the prediction horizon is sufficiently large and therefore, the effect of not including a terminal cost can be neglected. For a formal study about the possibility to disregard the terminal constraint and terminal cost see [85, 53, 68]. Due to the varying delay of the considered process a correction in the prediction of $y(k+1|k)$ has been used. With a Smith like predictor, the output prediction for $k+1$ made at $k$ using the nominal prediction $\hat{y}_n(k+1|k)$ is corrected as:

$$\hat{y}(k+1|k) = \hat{y}_n(k+1|k) + (\hat{y}_n(k|k) - y(k)) \tag{6.82}$$

being $y(k)$ the real process output at $k$. This simple correction improves the performance of MPC strategies in the case of systems with time delay [93].

Finally, the control strategies were implemented in the Matlab/Simulink environment and the weighting factor for the control effort has been chosen equal to $R = 5$. In order to restrict the system input and output in the experiments, the following constraints have been used:

$$\begin{array}{rcccll} 5 & \leq & u(k+i|k) & \leq & 100, & i = 0, \ldots, 14 \\ -20 & \leq & \Delta u(k+i|k) & \leq & 20, & i = 0, \ldots, 14 \\ 30 & \leq & \hat{y}(t+j|t) & \leq 70, & j = 2, \ldots, 26, \ \forall\, \boldsymbol{\theta} \in \mathrm{vert}(\boldsymbol{\Theta}) & \end{array} \tag{6.83}$$

Note that in the output restrictions the effect of the uncertainty has to be considered.

### 6.4.1 Min-max MPC with nonlinear upper bound

With the MMMPC strategy based on a nonlinear upper bound of the worst case cost (see Section 6.2) several setpoint tracking and disturbance rejection experiments were carried out. As a benchmark system the pilot plant emulating an exothermic chemical reaction was used and the control strategy was implemented with the identified model (6.81) and the constraints (6.83).

In the first place, a setpoint tracking experiment with the presented min-max control strategy was carried out (see Fig. 6.4). Starting in steady state with a reference of 55 °C, the setpoint is changed in $t = 30$ min to 65 °C and in $t = 90$ min to 45 °C. It can be observed in the experimental results that the min-max control strategy applies the new setpoint changes rapidly to the system and stabilizes the temperature in the setpoint in less than 20 minutes after the application of the changes in the reference. After the first setpoint change no overshoot can be observed whereas a marginal overshoot (approximately $-0.7$ °C) appears after the second setpoint change. After reaching steady state the control signal shows only small variations without periodic oscillations and the concentration $C_A$ remains in a constant level.

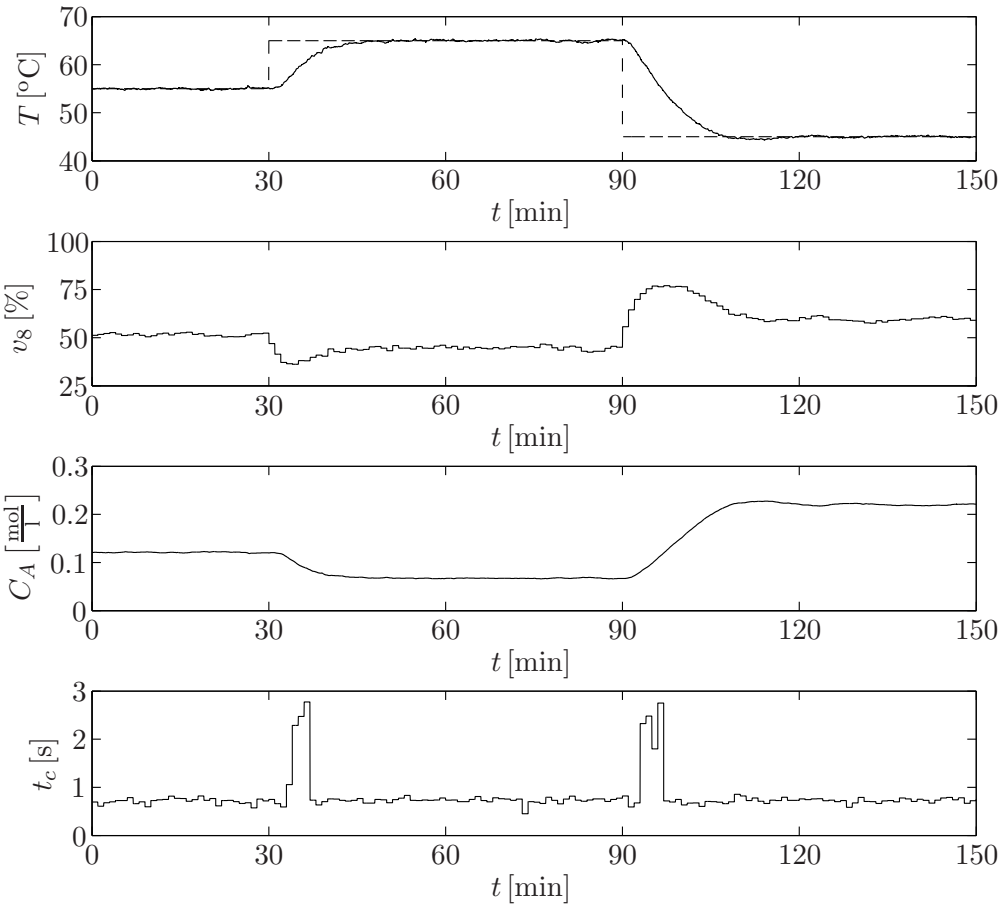


**Figure 6.4**: Setpoint tracking experiment controlled by the presented MMMPC strategy based on a nonlinear upper bound of the worst case cost. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and computation time $t_c$.

In the second experiment the presented min-max control strategy was used to control the pilot plant in presence of a constant error in the parameter $E$ of the underlying model of the chemical reaction. Due to the strong influence of the parameter $E$ on the dynamics of the system, the mentioned parameter was increased only by 3 % of the original value. The experimental results in Fig. 6.5 show that the control strategy stabilizes the system output in the desired reference without difficulties. Nevertheless, as a result of model mismatch, some overshoot can be observed after the setpoint changes (approximately 0.9 °C and $-2$ °C after the first and the second step, respectively). Comparing the result of Fig. 6.5 and Fig. 6.4 it is evident that the observed overshoot is mainly a result of the introduced model mismatch. However, the min-max control strategy shows its capability to stabilize the system in presence of a model uncertainty.

During the third experiment with the pilot plant controlled by the presented min-max control strategy an additive disturbance in the system input was considered (see

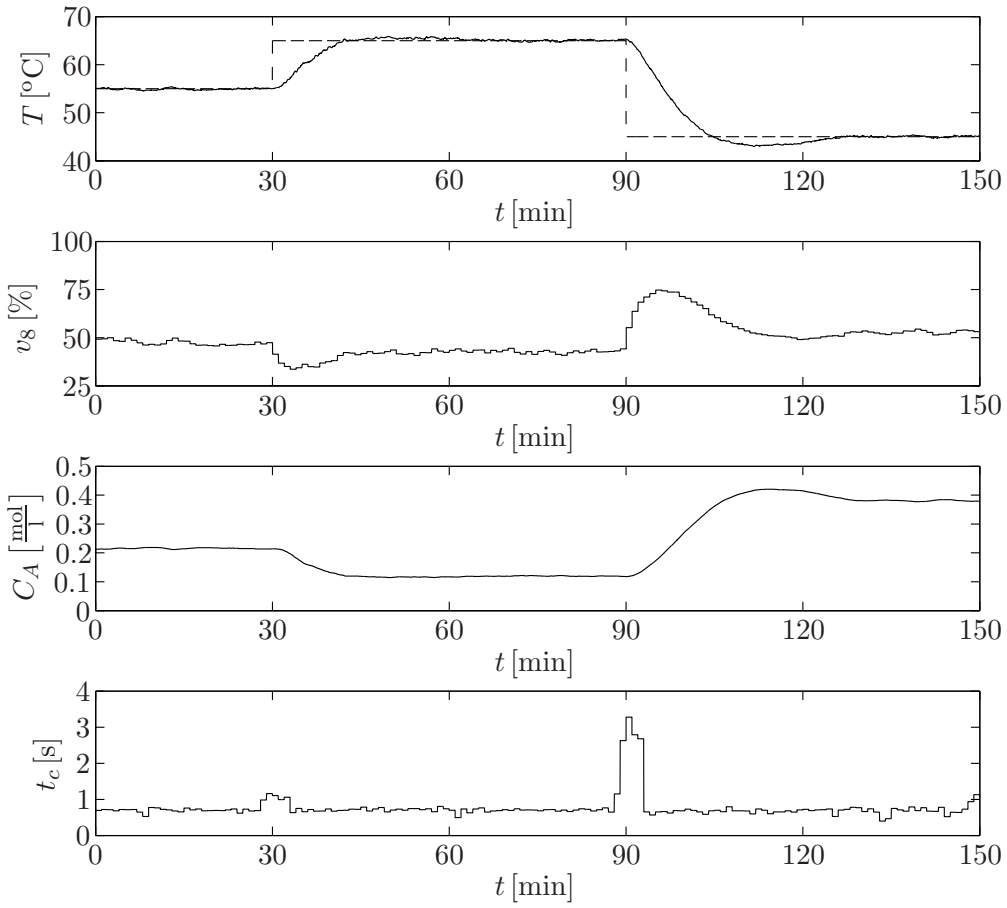


**Figure 6.5**: Disturbance rejection experiment (persistent disturbance in the emulated chemical reaction) controlled by the presented MMMPC strategy based on a nonlinear upper bound of the worst case cost. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and computation time $t_c$.

Fig. 6.6). The applied disturbance had a value of $\Delta v_8 = -15\,\%$ and was active from $t = 70\,\text{min}$ to $t = 110\,\text{min}$. During this interval the effectively applied system input is given by $v_8 = u + \Delta v_8$ whereas in absence of the disturbance the valve opening is defined by $v_8 = u$. After the application (or elimination) of the additive disturbance the min-max control strategy reacts rapidly to the increasing error in the temperature and compensates the divergence almost completely in only 10 minutes. During the application of the disturbance the calculated control $u$ is considerably higher and yields an effective valve opening $v_8$ which corresponds to the value of $v_8$ before the application of the disturbance. The presented min-max control strategy efficiently rejects the disturbance and stabilizes the system in the given setpoint in spite of the magnitude of the input error.

In the last experiment (see Fig. 6.7) with the pilot plant the disturbance rejection

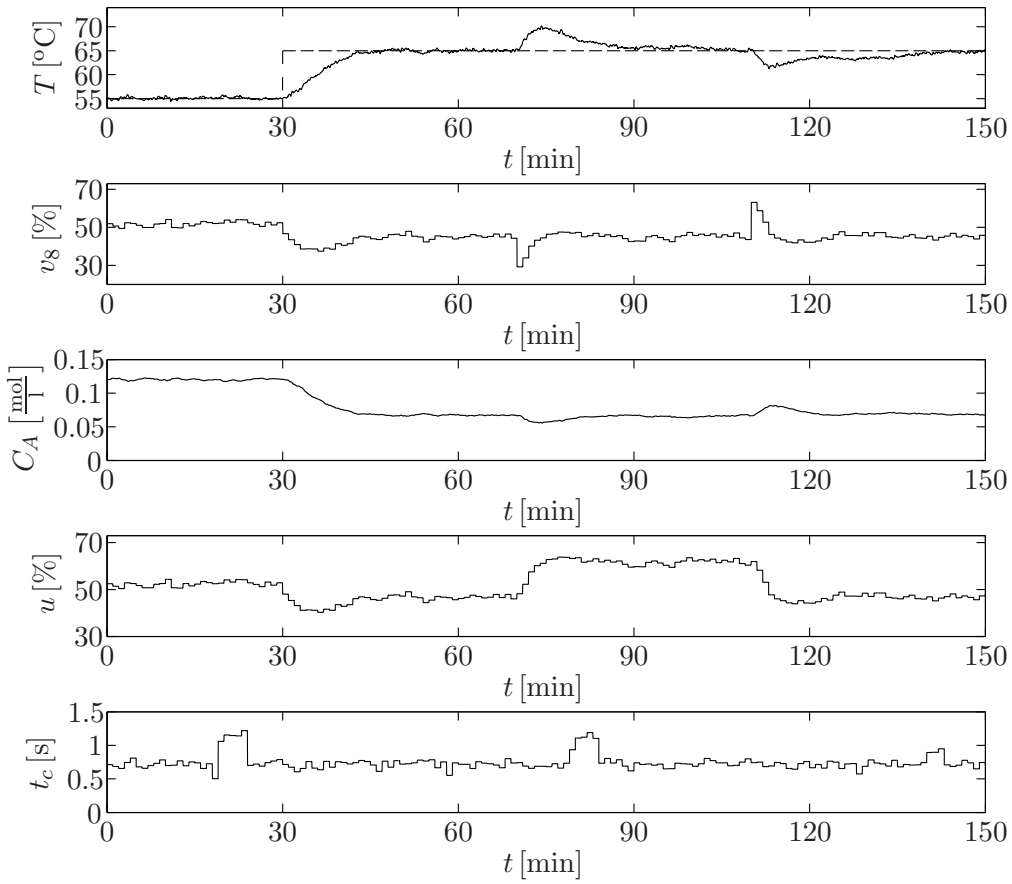


**Figure 6.6**: Disturbance rejection experiment (disturbance in the valve opening $v_8$) controlled by the presented MMMPC strategy based on a nonlinear upper bound of the worst case cost. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, input value $u$ calculated by the controller and computation time $t_c$.

capabilities of the min-max control strategy were tested by means of a disturbance in the feed $F_f$. Starting the experiment with the nominal feed $F_f = 0.05\,\mathrm{l/s}$, the disturbance $\Delta F_f = -0.02\,\mathrm{l/s}$ was applied to the system in $t = 60\,\mathrm{min}$. After the application of the disturbance it can be observed that the concentration $C_A$ decreases and, as a consequence, the temperature falls below the given setpoint and reaches a maximum divergence of $-2.5\,^{\circ}\mathrm{C}$. With an increasing error in the system output, the min-max control strategy partially closes the valve $v_8$ and compensates the effect of the reduced feed. The control strategy rejects the disturbance in approximately 20 minutes, but shows afterwards small oscillations in the temperature and the valve opening, totally justified by the magnitude of the disturbance ($\Delta F_f$ corresponds to an error of $-40\,\%$ with respect to the nominal feed).

The MMMPC strategy showed in the presented experiments a good behavior, both

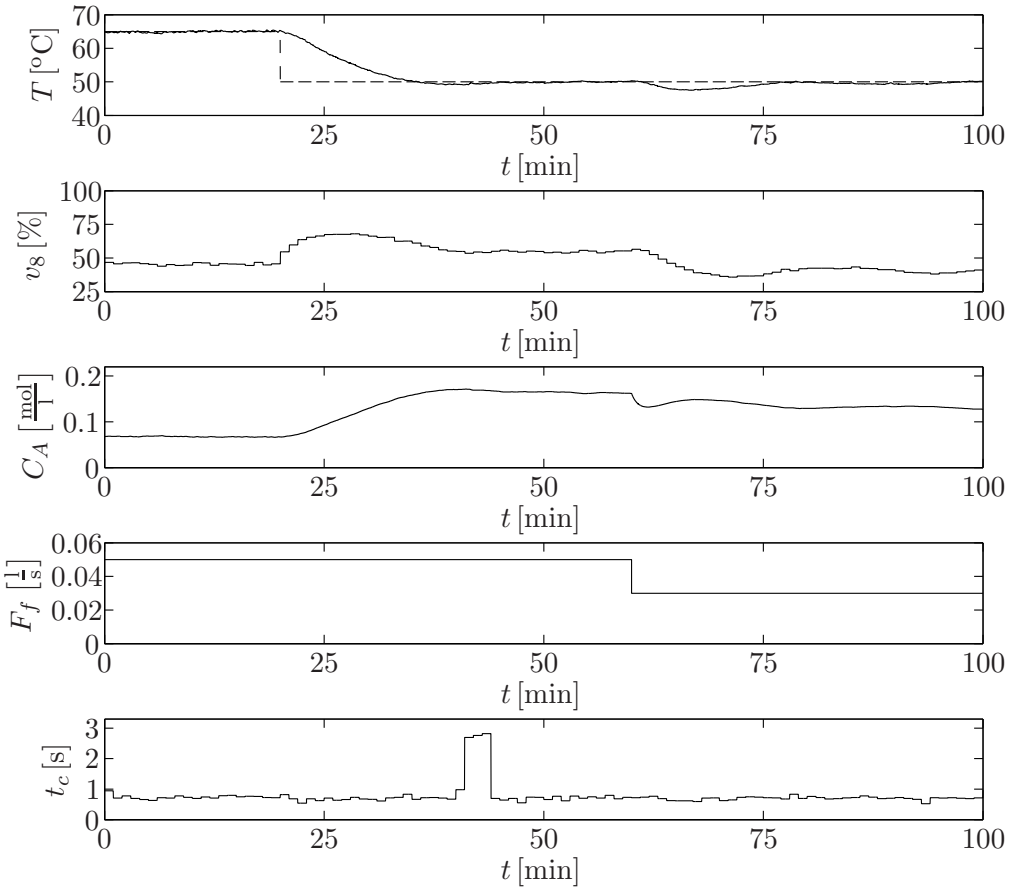


**Figure 6.7**: Disturbance rejection experiment (disturbance in the feed $F_f$) controlled by the presented MMMPC strategy based on a nonlinear upper bound of the worst case cost. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, feed $F_f$ and computation time $t_c$.

for setpoint tracking and disturbance rejection. Based on a nonlinear upper bound of the worst case cost, the computation of the input sequence required an average computation time of $t_c = 0.772\,\mathrm{s}$ with a maximum of $t_c = 3.277\,\mathrm{s}$ and a minimum of $t_c = 0.397\,\mathrm{s}$. The necessary time to compute a new input sequence is quite low with respect to the considered prediction horizon of $N = 25$ and lies clearly inside the used sampling time of $t_s = 60\,\mathrm{s}$. Note that the original optimization problem would require approximately $2^{Nn_\theta+1}$ operations and could not be solved within the sampling time.

### 6.4.2 Min-max MPC with quadratic upper bound

The MMMPC strategy using a quadratic upper bound of the worst case cost presented in Section 6.3 was applied to the pilot plant which emulates an exothermic chemical reaction. Several experiments, including setpoint tracking and disturbance rejection, were carried out in order to test the mentioned control strategy. The min-max control strategy for the pilot plant is based on the identified linear model (6.81) and has been implemented with the constraints given in (6.83).

The results of a setpoint tracking experiment with the pilot plant controlled by the MMMPC strategy are shown in Fig. 6.8. During the experiment the setpoint was changed in $t = 30\,\text{min}$ from $55\,^{\circ}\text{C}$ to $65\,^{\circ}\text{C}$ and in $t = 90\,\text{min}$ set to $45\,^{\circ}\text{C}$. After the first setpoint change the control strategy compensates the error in the system output rapidly and stabilizes the system output in the given reference without any overshoot. The second setpoint change results in a small overshoot of $-0.9\,^{\circ}\text{C}$, only slightly higher than the one observed in Fig. 6.4. After reaching steady state, the system output remains in the given setpoint and the concentration $C_A$ maintains a constant level.

In the second experiment, the results are given in Fig. 6.9, the disturbance rejection capabilities of the presented min-max control strategy were tested by means of an error in the used underlying model of the exothermic chemical reaction. The error was introduced in the parameter $E$ and held constant during the entire experiment. Due to the strong influence on the dynamic behavior of the system, the mentioned parameter was increased only by $3\,\%$ of the nominal value. Due to the model mismatch (a consequence of the introduced error in the parameter $E$, compare results in Fig. 6.8) an overshoot of $0.8\,^{\circ}\text{C}$ and $-2.2\,^{\circ}\text{C}$ can be observed after the setpoint changes. In spite of the error in the emulated model, the min-max control strategy stabilizes the system output in the reference.

In the third experiment carried out with the MMMPC strategy an additive disturbance was applied to the system input. The used disturbance had a value of $\Delta v_8 = -15\,\%$ and was active in the period from $t = 70\,\text{min}$ to $t = 110\,\text{min}$. It can be observed in Fig. 6.10 that the application of the disturbance in the system input leads to a low effective opening of the valve $v_8$ and, as a consequence, results in a rapidly increasing temperature. The MMMPC strategy reacts quickly to the error in the system output, but the stabilization of the system output in the reference requires nearly 30 minutes, both after the application and after the removal of the disturbance. The complete compensation of the output error underlines the disturbance rejection capabilities of the applied control strategy.

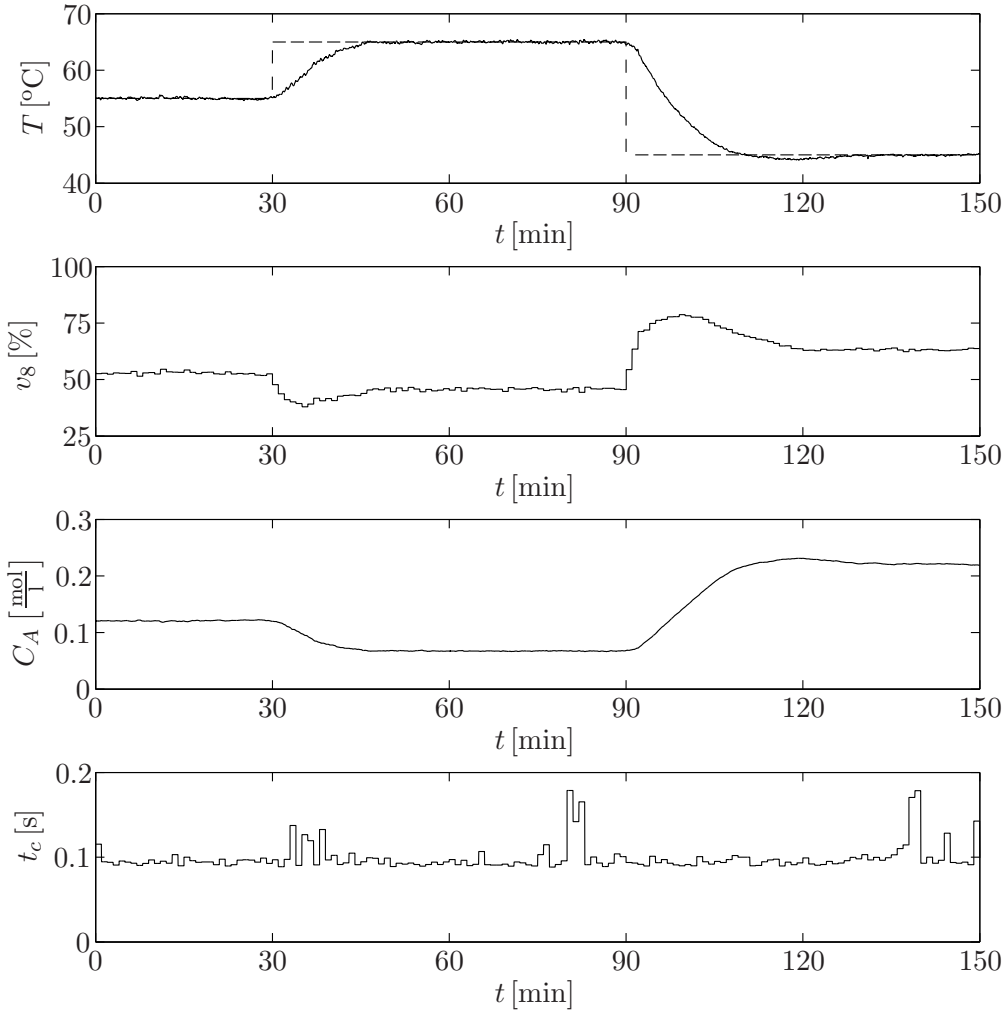


**Figure 6.8**: Setpoint tracking experiment controlled by the presented MMMPC strategy based on a quadratic upper bound of the worst case cost. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and computation time $t_c$.

In the fourth experiment (see Fig. 6.11) with the pilot plant and the presented min-max control strategy an error was introduced in the feed $F_f$. The disturbance $\Delta F_f = -0.02\,\mathrm{l/s}$, which corresponds to an error of $-40\,\%$ in the nominal feed, was applied in $t = 60\,\mathrm{min}$ and maintained constant until the end of the experiment. The application of the disturbance leads to a reduced feed $F_f$ and results in a lower concentration $C_A$. As a consequence, the temperature decreases and reaches a maximum output error of $-2.1\,^{\circ}\mathrm{C}$. The increasing difference between the system output and the given setpoint causes the control strategy to close gradually the valve $v_8$. The fast reaction of the controller compensates the output error almost completely in 20 minutes. Afterwards, the system shows a constant output error of approximately $-0.5\,^{\circ}\mathrm{C}$ during 11 minutes. Finally, the min-max control strategy stabilizes the temperature in the setpoint and

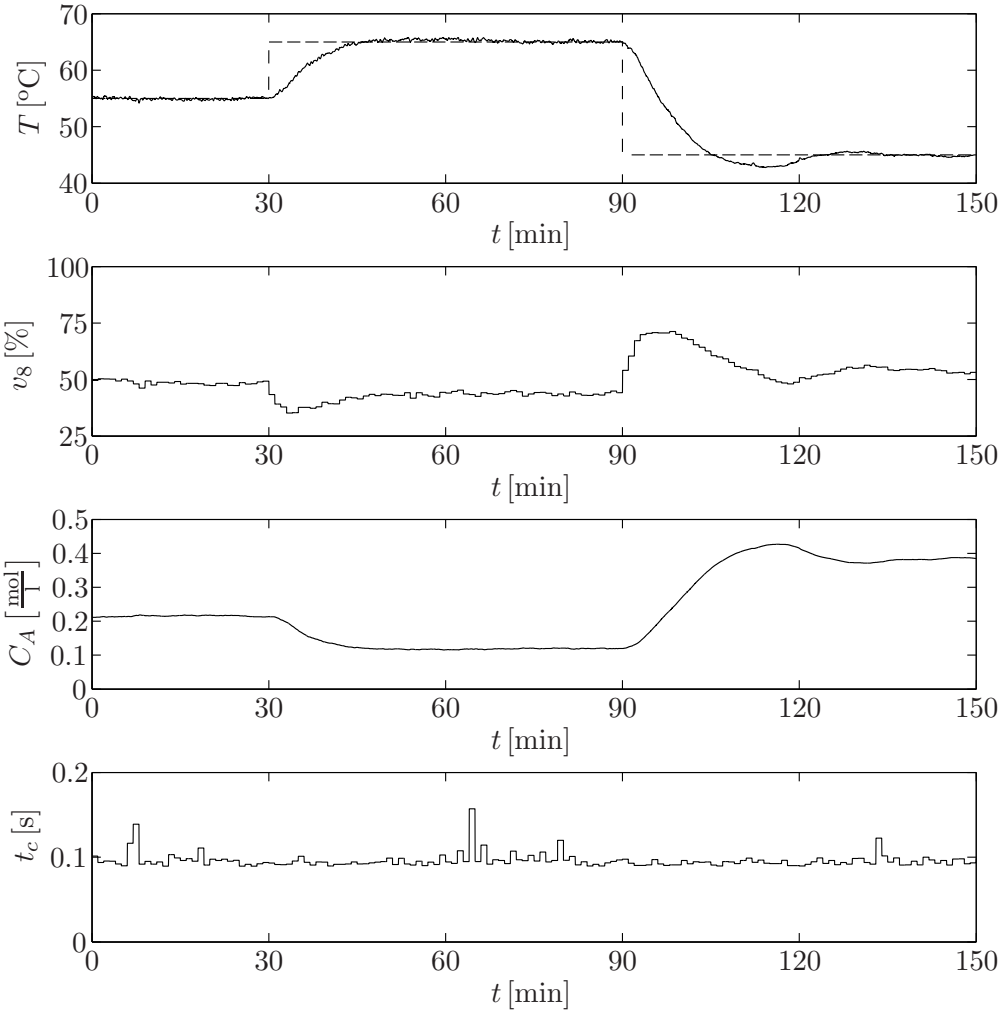


**Figure 6.9**: Disturbance rejection experiment (persistent disturbance in the emulated chemical reaction) controlled by the presented MMMPC strategy based on a quadratic upper bound of the worst case cost. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and computation time $t_c$.

the system reaches steady state. The magnitude of the applied disturbance justifies the required time to completely compensate the output error and shows the efficiency of the presented control strategy to reject disturbances.

The presented MMMPC strategy using a quadratic upper bound of the worst case cost showed in the experiments its ability for setpoint tracking and disturbance rejection. The controller stabilized the system output in the reference even in presence of strong disturbances. In the presented experimental results the control strategy needed an average computation time of $t_c^{avg} = 0.097\,\mathrm{s}$ to compute a new input sequence, with a maximum of $t_c^{max} = 0.179\,\mathrm{s}$ and a minimum of $t_c^{min} = 0.088\,\mathrm{s}$. For the conside-

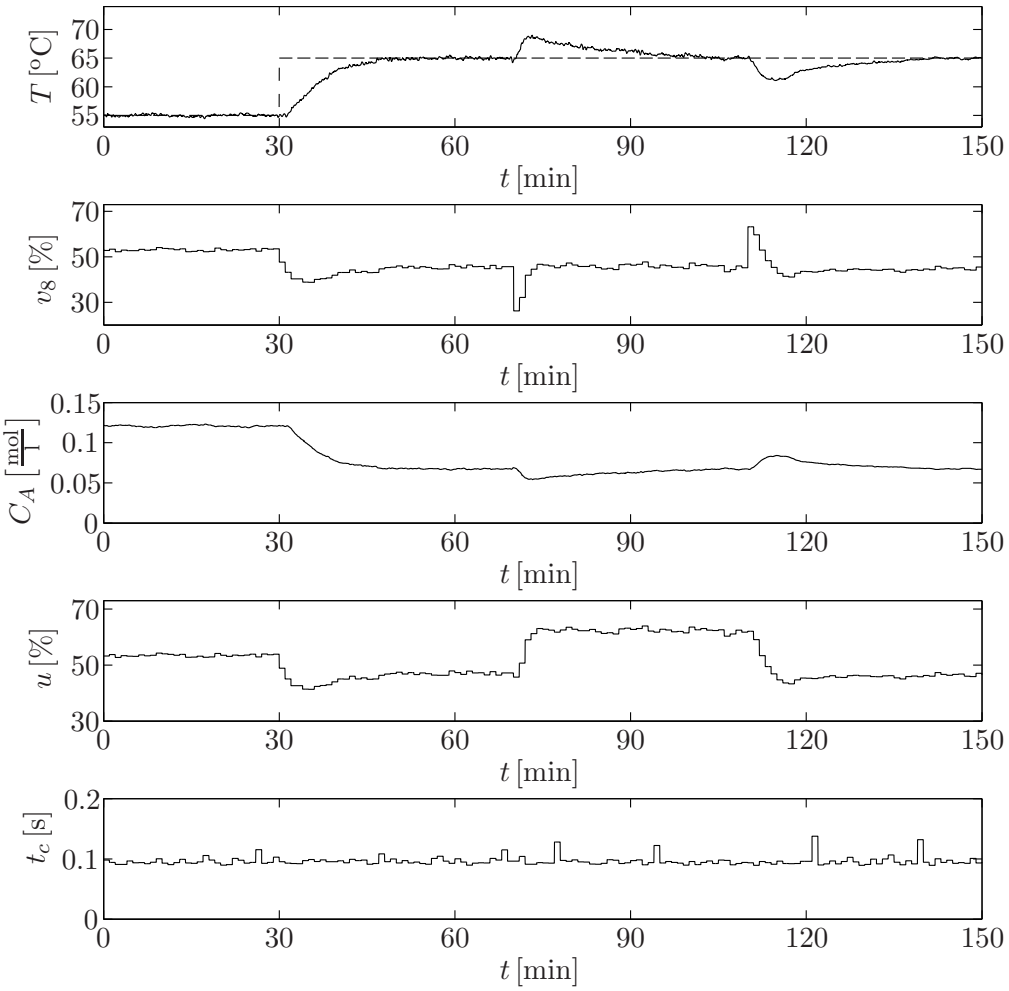


**Figure 6.10**: Disturbance rejection experiment (disturbance in the valve opening $v_8$) controlled by the presented MMMPC strategy based on a quadratic upper bound of the worst case cost. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, input value $u$ calculated by the controller and computation time $t_c$.

red prediction horizon $N = 25$ the mentioned computation times represent very low values which allow the implementation of the presented control strategy with typical prediction horizons used in MPC.

## 6.5 Comparison of simulation results

For comparison purposes both MMMPC strategies have been applied to a simulation model (3.1)-(3.5) of the pilot plant. The strategies have been implemented with the

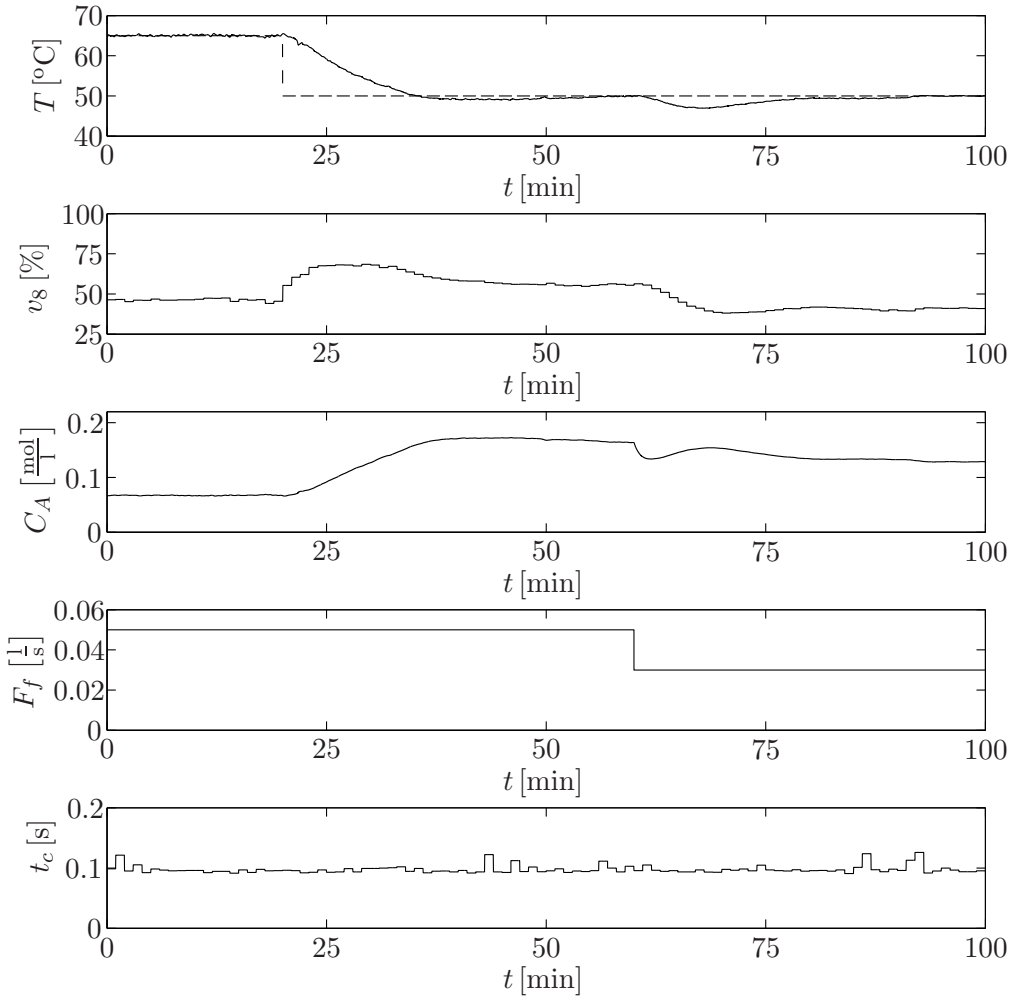


**Figure 6.11**: Disturbance rejection experiment (disturbance in the feed $F_f$) controlled by the presented MMMPC strategy based on a quadratic upper bound of the worst case cost. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, feed $F_f$ and computation time $t_c$.

parameters and constraints used in the experiments to allow an exact comparison of the simulation results. The simulations have been carried out on the same computer used for the experiments presented in the previous sections.

The results obtained in a setpoint tracking experiment with the nominal model are given in Fig. 6.12. It can be observed that both MMMPC strategies give very similar results with a slightly better control performance of the control strategy based on the nonlinear upper bound. The MMMPC based on the quadratic upper shows a more conservative behavior resulting in a slower reaction to setpoint changes. The second simulation, see Fig. 6.13, was carried out with a constant error of 3 % in the

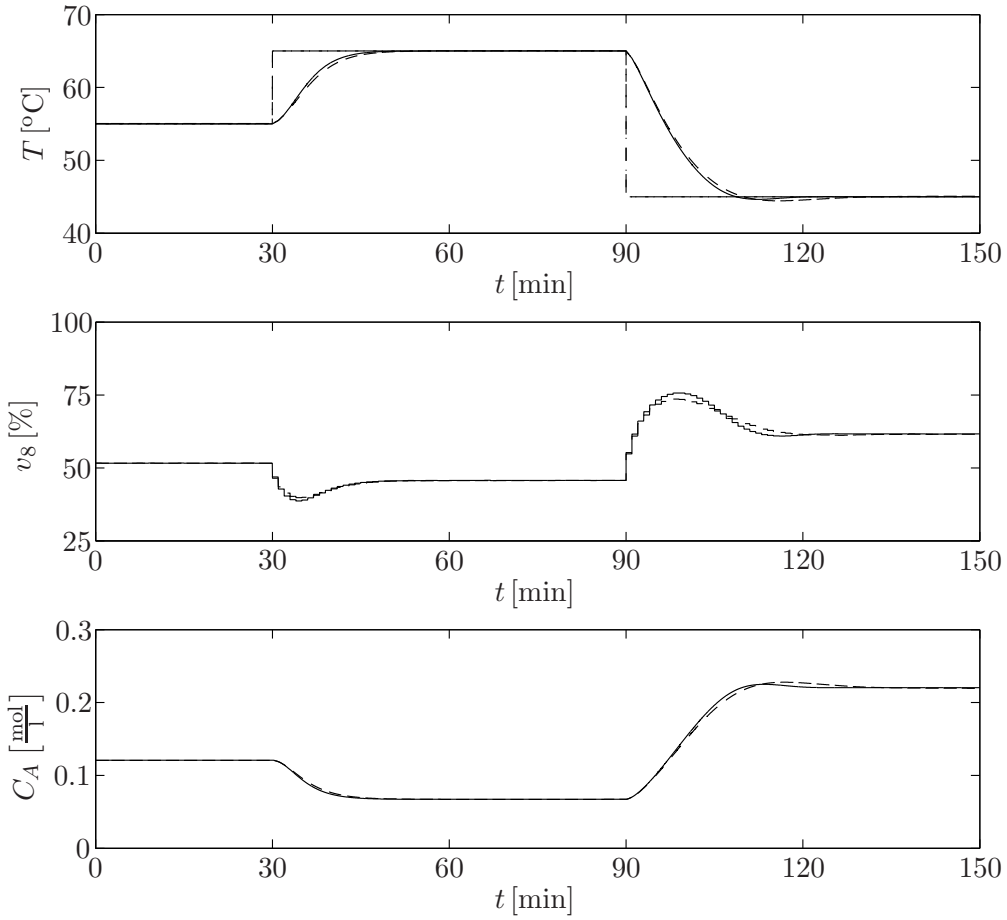


**Figure 6.12**: Comparison of the simulations results with the nominal model obtained with the MMMPC based on a nonlinear upper bound (solid line) and the MMMPC based on a quadratic upper bound (dashed line). From top to bottom: tank temperature $T$, aperture of the valve $v_8$ and concentration $C_A$.

activation energy $E$ of the simulation model. The results confirm the faster reaction of the MMMPC based on the nonlinear upper bound with a lower overshoot. The results show that both MMMPC strategies reach an offset-free steady state in spite of the model mismatch.

The quadratic upper bound used in the control strategy presented in Section 6.3.2 is a more conservative estimation of the worst case than the nonlinear upper bound of the strategy given in Section 6.2.3. As a consequence, the MMMPC based on the quadratic bound shows a more cautious reaction in comparison to the MMMPC based on the nonlinear bound. With respect to the computational burden, the use of a quadratic upper bound gives the possibility to solve the optimization problem, i.e. to calculate the input action, with QP techniques and leads to lower computation times. Tab. 6.1 presents the Sum of Square Errors for the different MMMPC strategies obtained in the

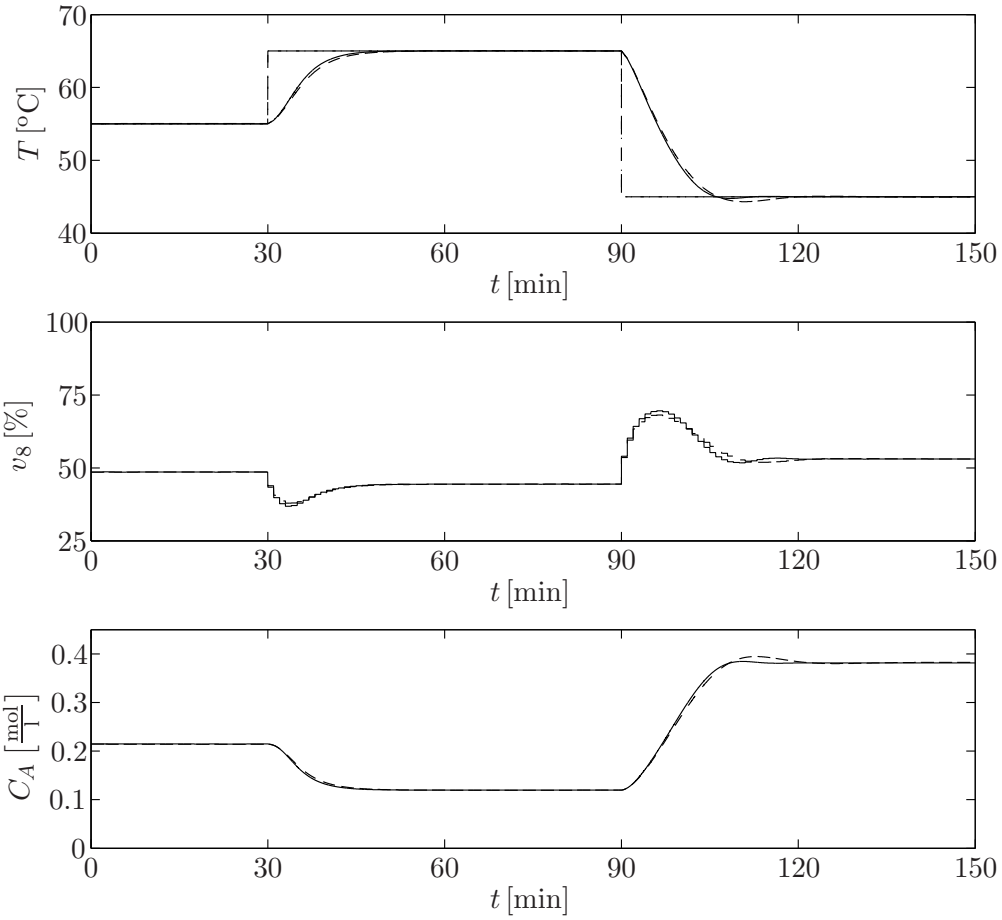


**Figure 6.13**: Comparison of the simulations results with a disturbance in the activation energy $E$ of the simulation model obtained with the MMMPC based on a nonlinear upper bound (solid line) and the MMMPC based on a quadratic upper bound (dashed line). From top to bottom: tank temperature $T$, aperture of the valve $v_8$ and concentration $C_A$.

| MMMPC | SSE (nominal) | SSE (model mismatch) | $t_c^{avg}$ |
|---|---|---|---|
| nonlinear b. | 1946.5 | 2021.4 | 0.094 s |
| quadratic b. | 2104.4 | 2188.7 | 0.778 s |

**Table 6.1**: Comparison between the two MMMPC strategies with respect to the Sum of Square Errors and the average computation times in the simulations.

simulations and the necessary computation times to calculate the input action.

## 6.6 Conclusions of the chapter

In this chapter two MMMPC strategies based on upper bounds of the worst case cost have been presented. Both control strategies have been developed to be used with linear models with an additive uncertainty.

The first MMMPC strategy, originally published in [105], uses a nonlinear upper bound of the worst case cost. This control strategy considerably reduces the computational complexity as the maximization problem, normally carried out by the evaluation of all $2^{n_z+1}$ possible vertices of the disturbance, is replaced by an algorithm which diagonalizes a quadratic matrix of size $n_z$. The algorithm includes only simple matrix operations and can be implemented even with programming languages usually used in embedded systems.

The second MMMPC strategy, proposed in [3], is based on a quadratic upper bound of the worst case cost. In this control strategy, the original optimization problem is replaced by a tractable quadratic programming problem which provides a close solution of the original problem. The modified optimization problem allows to obtain a new input sequence by solving only 2 QP problems, one to obtain an initial guess and one to solve the optimization problem. Each of the two QP problems have the same complexity as a constrained linear MPC problem.

The presented MMMPC strategies have a reduced complexity in comparison to the original optimization problem and can be used with prediction horizons typical in linear MPC. For validation purposes, both control strategies were applied to a pilot plant emulating an exothermic chemical reaction. The control strategies were tested in setpoint tracking and disturbance rejection experiments. The obtained results underlined the capability of the presented control strategies to stabilize the benchmark system even in presence of strong disturbances. Besides, a direct comparison of results obtained in simulations with the two MMMPC strategies was presented. The comparison showed that the MMMPC based on the nonlinear upper bound requires higher computation times to compute the control action, but leads to better results. The increased control performance is a result of the nonlinear upper bound which represents a less conservative estimation of the worst case. With the presented proof of stability, the MMMPC based on a nonlinear upper bound is a valid alternative to control uncertain processes.

# Chapter 7

# Volterra based min-max MPC

In this chapter, two novel nonlinear min-max MPC strategies using Volterra series models are presented. The min-max formulation is used to take into account the effect of model uncertainties and perturbations [25]. The computation of the input sequence is carried out considering the worst case with respect to the model uncertainties and the perturbations. Generally, the mentioned worst case is obtained by maximization of an usually quadratic cost function. Then, the input sequence is computed minimizing the previously mentioned worst case (compare Chapter 6) in the following way:

$$\boldsymbol{u}^* = \arg \min_{\boldsymbol{u}} \max_{\theta} J(\boldsymbol{u}, \boldsymbol{\theta}) \tag{7.1}$$

where the optimization of the cost function, depending on the input sequence $\boldsymbol{u}$ and the disturbance vector $\boldsymbol{\theta}$, is possibly subject to constraints in $\boldsymbol{u}$ and $\boldsymbol{\theta}$. Probably the main disadvantage of the min-max approach is the complexity of the optimization problem, leading to a high computational complexity. As already mentioned in Chapter 6, the computation of the control sequence considering the worst case cost with respect to a disturbance usually requires the solution of a NP-hard problem.

The min-max strategies for Volterra series models presented in this chapter allow to calculate the worst case in a computationally efficient way. The computation of the worst case exploits the non-autoregressive nature of the used Volterra series models. In contrast to the procedures presented in Chapter 6 using approximations, i.e. upper bounds, of the worst case cost, the approach presented in the following sections calculates the exact worst case cost.

In the following section, the computation of the exact worst case cost for a quadratic cost function in combination with a second order Volterra series model will be shown.

This worst case cost is then used in a receding horizon control strategy to calculate the optimal input sequence. Afterwards, the weighting of the control effort of the presented optimization will be modified in order to achieve stability. Then, input-to-state stability for the control strategy based on the modified optimization problem will be proven. Finally, both control strategies will be applied to a benchmark system and the control performance will be illustrated by experimental results.

## 7.1 Volterra based MMMPC with exact worst case cost

This section presents a min-max MPC based on a second order Volterra series model using the exact worst case cost to compute the input sequence. In a first step, the general optimization problem resulting from the use of a min-max approach in combination with a second order Volterra series prediction model will be shown. Afterwards, the maximization problem will be replaced by the exact worst case cost and the optimization problem will be included in an MPC strategy.

### 7.1.1 Problem description

Consider the general second order non-autoregressive Volterra model series to approximate fading memory systems [20] defined as in (2.10). This model can be extended to include a bounded additive disturbance in the following form:

$$y(k) = h_0 + \sum_{i=1}^{N_t} h_{1,i}u(k-i) + \sum_{i=1}^{N_t}\sum_{j=i}^{N_t} h_{2,ij}u(k-i)u(k-j) + w\theta(k) \tag{7.2}$$

where the disturbance $\theta(k) \in \mathbb{R}$ is bounded by $|\theta(k)| \leq \varepsilon$ and $w > 0$. For simplicity and without loss of generality $\varepsilon = 1$ is assumed[1]. The truncation order for the first order and second order term coefficients is denoted with $N_t$ (see section 2.3).

The future behavior of a second order Volterra series model is given generally by (2.43)-(2.44) [38]. Under consideration of a bounded additive uncertainty (7.2) and with the prediction horizon $N$, the control horizon $N_u$ and the truncation order $N_t$ the

[1]Otherwise, the parameter $w$ can be scaled to become $\varepsilon w$.

prediction model is defined as an extension of (2.43)-(2.44):

$$\hat{\boldsymbol{y}} = G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) + W\boldsymbol{\theta} \tag{7.3}$$

$$\boldsymbol{c} = H\boldsymbol{u}_p + \boldsymbol{g} + \boldsymbol{h}_0 + \boldsymbol{d} \tag{7.4}$$

where $\boldsymbol{\theta} \in \mathbb{R}^N$ denotes the future sequence of the disturbance over the prediction horizon and $W \in \mathbb{R}^{N\times N}$ is a matrix relating the disturbance sequence with the model output along the prediction horizon. The sequence $\boldsymbol{\theta}$ is given by:

$$\boldsymbol{\theta} = \begin{bmatrix} \theta(k+1|k) \\ \theta(k+2|k) \\ \vdots \\ \theta(k+N|k) \end{bmatrix} \tag{7.5}$$

and the set of possible disturbance trajectories is defined as:

$$\boldsymbol{\Theta} = \{\boldsymbol{\theta} \in \mathbb{R}^N : \|\boldsymbol{\theta}\|_\infty \leq 1\} \tag{7.6}$$

With the used prediction model (7.3)-(7.4) being non-autoregressive, the disturbance in a certain sampling period only influences the model output in the same sampling period. Hence, $W$ represents a diagonal matrix with all elements equal to $w$:

$$W = w\, I_N \tag{7.7}$$

For a detailed definition of the matrices and vectors used in the prediction model (7.3)-(7.4) the reader is referred to Section 2.3 and to the Appendix A.1.

Consider the second order Volterra series prediction model with additive uncertainty to be used in the quadratic performance index (2.47) with penalization of the control effort. With the transformation of the penalization term as shown in (2.51), the performance index based on (7.3)-(7.4) can be written as:

$$\begin{aligned} J(\boldsymbol{u},\boldsymbol{\theta}) = {} & \boldsymbol{\theta}^T M_{\theta\theta}\boldsymbol{\theta} + 2\,\boldsymbol{\theta}^T M_{\theta u}\boldsymbol{u} + 2\,\boldsymbol{\theta}^T M_{\theta f}\mathbf{f}(\boldsymbol{u}) + 2\,\boldsymbol{\theta}^T \boldsymbol{m}_{\theta c} + \\ & \boldsymbol{u}^T M_{uu}\boldsymbol{u} + 2\,\boldsymbol{u}^T M_{uf}\mathbf{f}(\boldsymbol{u}) + 2\,\boldsymbol{u}^T \boldsymbol{m}_{uc} + \\ & \mathbf{f}(\boldsymbol{u})^T M_{ff}\mathbf{f}(\boldsymbol{u}) + 2\,\mathbf{f}(\boldsymbol{u})^T \boldsymbol{m}_{fc} + m_{cc} \end{aligned} \tag{7.8}$$

with the matrices defined by:

$$\begin{aligned} M_{\theta\theta} &= W^T W, & M_{uf} &= G^T \\ M_{\theta u} &= W^T G, & \boldsymbol{m}_{uc} &= G^T(\boldsymbol{c}-\boldsymbol{r}) - \lambda L_u^T \boldsymbol{u}_l \\ M_{\theta f} &= W^T, & M_{ff} &= I_N \\ \boldsymbol{m}_{\theta c} &= W^T(\boldsymbol{c}-\boldsymbol{r}), & \boldsymbol{m}_{fc} &= \boldsymbol{c}-\boldsymbol{r} \\ M_{uu} &= G^T G + \lambda L_u^T L_u, & m_{cc} &= \boldsymbol{c}^T\boldsymbol{c} + \boldsymbol{r}^T\boldsymbol{r} - 2\,\boldsymbol{c}^T\boldsymbol{r} + \lambda \boldsymbol{u}_l^T \boldsymbol{u}_l \end{aligned} \tag{7.9}$$

where $\boldsymbol{r}$ denotes the reference trajectory along the prediction horizon and the matrix $L_u$ is defined as in (2.50). Note that the used prediction model (7.3)-(7.4) is affine in $\boldsymbol{\theta}$ and, as a consequence, the quadratic cost function (7.8) is convex in $\boldsymbol{\theta}$. Furthermore, it has to be mentioned that the cost function is of fourth order with respect to the input signal $\boldsymbol{u}$ due to the quadratic function $\mathbf{f}(\boldsymbol{u})$.

The considered initial scheme of predictive control is the MMMPC [23] in which the optimal sequence $\boldsymbol{u}^*$ is calculated by solving a min-max problem. Considering only input constraints[2], the optimization problem can be written as:

$$\begin{aligned} \boldsymbol{u}^* &= \arg\min_{\boldsymbol{u}} J^*(\boldsymbol{u}) \\ &\quad \text{s.t. } L_c\boldsymbol{u} \leq \boldsymbol{b}_c \end{aligned} \tag{7.10}$$

with the worst case cost given by:

$$J^*(\boldsymbol{u}) = \max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} J(\boldsymbol{u},\boldsymbol{\theta}) \tag{7.11}$$

and $L_c \in \mathbb{R}^{n_c\times N_u}$ and $\boldsymbol{b}_c \in \mathbb{R}^{n_c}$ (being $n_c$ the number of restrictions). The solution of this problem is applied using a receding horizon strategy, as in all predictive control schemes.

Then, with the cost function (7.8) being convex in $\boldsymbol{\theta}$, the solution to the maximization problem (7.11) can be found (at least) in one of the vertices of $\boldsymbol{\Theta}$ (compare Section 6.1.1 for the case of a linear model). Hence, the maximization problem (7.11) is equivalent to:

$$J^*(\boldsymbol{u}) = \max_{\boldsymbol{\theta}\in\text{vert}\{\boldsymbol{\Theta}\}} J(\boldsymbol{u},\boldsymbol{\theta}) \tag{7.12}$$

The evaluation of each one of the $2^N$ vertices of $\boldsymbol{\Theta}$ leads to an exponential complexity and is a well known NP-hard problem. Therefore, the maximization problem (7.12) and, as a consequence, the computation of the input sequence $\boldsymbol{u}^*$ (7.10) can only be solved for small horizons when the cost has to be evaluated for all possible vertices.

The adopted strategy in the control scheme used in this chapter is directed to reduce the computational cost of the maximization problem (7.12). The special character of the second order Volterra series model, with (7.2) being non-autoregressive, is exploited to reduce the computational burden of the maximization. With the new algorithm, the exact worst case cost can be calculated with a complexity of $O(N^2)$ instead of $O(2^N)$.

[2]Output constraints can be easily considered but they result in nonlinear constraints in the minimization problem due to the nonlinearity of the Volterra series model. These nonlinear constraints can be handled with advanced optimization algorithms such as sequential quadratic programming (SQP) techniques.

### 7.1.2 Calculation of the worst case cost

This section presents the computation of the exact worst case cost for the maximization problem given in (7.12). The exact worst case cost can be determined with simple mathematical operations and allows an easy implementation.

Using the cost function (7.8) based on a second order Volterra series prediction model in (7.12), the maximization problem for a given input sequence $\boldsymbol{u}$ can be expressed as (compare (6.16) in Section 6.1.1):

$$J^*(\boldsymbol{u}) = \max_{\boldsymbol{\theta} \in \mathrm{vert}\{\boldsymbol{\Theta}\}} \boldsymbol{\theta}^T T \boldsymbol{\theta} + 2\,\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u}) + J(\boldsymbol{u}, 0) \tag{7.13}$$

with

$$\begin{aligned} T &= M_{\theta\theta} \\ \boldsymbol{q}(\boldsymbol{u}) &= M_{\theta u}\boldsymbol{u} + M_{\theta f}\mathbf{f}(\boldsymbol{u}) + \boldsymbol{m}_{\theta c} \\ J(\boldsymbol{u}, 0) &= \boldsymbol{u}^T M_{uu}\boldsymbol{u} + 2\,\boldsymbol{u}^T M_{uf}\mathbf{f}(\boldsymbol{u}) + 2\,\boldsymbol{u}^T \boldsymbol{m}_{uc} + \\ &\quad \mathbf{f}(\boldsymbol{u})^T M_{ff}\mathbf{f}(\boldsymbol{u}) + 2\,\mathbf{f}(\boldsymbol{u})^T \boldsymbol{m}_{fc} + m_{cc} \end{aligned} \tag{7.14}$$

where $J(\boldsymbol{u}, 0)$ represents the nominal cost. The matrix $T \in \mathbb{R}^{N \times N}$ is given with (7.7) and (7.9) as:

$$T = w^2\, I_N \tag{7.15}$$

and is defined as a positive definite diagonal matrix. Then, the calculation of $J^*(\boldsymbol{u})$ requires the maximization of the sum of the terms $\boldsymbol{\theta}^T T \boldsymbol{\theta}$ and $\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u})$ with respect to the disturbance vector $\boldsymbol{\theta}$. Obviously, with $J(\boldsymbol{u}, 0)$ not being a function of $\boldsymbol{\theta}$, the nominal cost cannot be maximized (nor minimized) with respect to the uncertainty.

With $T$ a positive definite diagonal matrix, the term $\boldsymbol{\theta}^T T \boldsymbol{\theta}$ can be written as:

$$\boldsymbol{\theta}^T T \boldsymbol{\theta} = \mathrm{trace}(T) \quad \forall\, \boldsymbol{\theta} \in \mathrm{vert}\{\boldsymbol{\Theta}\} \tag{7.16}$$

and represents a constant term for all possible vertices, i.e. $\boldsymbol{\theta}^T T \boldsymbol{\theta}$ adopts its maximum value independently of the chosen vertex $\boldsymbol{\theta}$. From (7.15), it is clear that $\mathrm{trace}(T) = Nw^2$ and the maximization problem (7.13) can be expressed as:

$$J^*(\boldsymbol{u}) = Nw^2 + \max_{\boldsymbol{\theta} \in \mathrm{vert}\{\boldsymbol{\Theta}\}} 2\,\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u}) + J(\boldsymbol{u}, 0) \tag{7.17}$$

Now, with the first term in (7.17) being a constant, the optimization problem is solved by maximizing $\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u})$. This can be easily achieved when all elements of $\boldsymbol{\theta}$ have the same leading signs as the corresponding elements of $\boldsymbol{q}(\boldsymbol{u})$, more specifically the multiplication of an element of $\boldsymbol{\theta}$ with the corresponding element of $\boldsymbol{q}(\boldsymbol{u})$ leads to a positive result. Hence, the disturbance vector $\boldsymbol{\theta}^T$ maximizes the term $\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u})$ for

$\boldsymbol{\theta} = \text{sgn}(\boldsymbol{q}(\boldsymbol{u}))$. The resulting maximum corresponds to the sum of absolute values of $\boldsymbol{q}(\boldsymbol{u})$:

$$\max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} \boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u}) = \text{sgn}(\boldsymbol{q}(\boldsymbol{u}))^T \boldsymbol{q}(\boldsymbol{u}) = \|\boldsymbol{q}(\boldsymbol{u})\|_1 \tag{7.18}$$

Then, being $J(\boldsymbol{u}, 0)$ a constant function with respect to the disturbance vector $\boldsymbol{\theta}$ and using (7.18) in (7.17), the initial maximization problem (7.13) becomes:

$$J^*(\boldsymbol{u}) = Nw^2 + 2\,\|\boldsymbol{q}(\boldsymbol{u})\|_1 + J(\boldsymbol{u}, 0) \tag{7.19}$$

With $J(\boldsymbol{u}, 0)$ being the nominal cost and $\boldsymbol{\theta}^T T \boldsymbol{\theta}$ maximized for every $\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}$ (7.16), only the value of $\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u})$ is influenced directly by the election of $\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}$. As a consequence, the vertex maximizing the term $\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u})$ also maximizes the cost function $J(\boldsymbol{u}, \boldsymbol{\theta})$. Hence, the vertex maximizing the cost function $J(\boldsymbol{u}, \boldsymbol{\theta})$ is defined as:

$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} J(\boldsymbol{\theta}, \boldsymbol{u}) = \arg \max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} \boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u}) = \text{sgn}(\boldsymbol{q}(\boldsymbol{u})) \tag{7.20}$$

Note that (7.19) is the exact solution to the maximization problem and can be calculated easily by simple mathematical operations, i.e. the computation of the worst case cost does not require an evaluation of the $2^N$ vertices in $\boldsymbol{\Theta}$. The reduction of the complexity of the maximization to $O(N^2)$ is especially relevant for the implementation when using longer prediction horizons. It has to be mentioned that the exact and easy computation of the worst case cost $J^*(\boldsymbol{u})$ is a result of the non-autoregressive character of the used second order Volterra series model with additive uncertainty (7.2). The non-autoregressive character of the used model leads to the diagonal positive definite matrix $T$ which is the most relevant issue to compute the exact worst case cost. In the case of an autoregressive second order Volterra series model with additive uncertainty, i.e. the matrix $T$ is non-diagonal and not necessarily positive definite, the Procedure 6.1 to compute an upper bound of the worst case cost shown in Section 6.2.2 can be used. It has to be mentioned that the Procedure 6.1 used with a non-autoregressive second order Volterra series model with additive uncertainty (7.2) results in a diagonal matrix $T$ with only positive entries and, as a consequence, the mentioned procedure computes the exact solution of the maximization problem.

**Remark 7.1** *Consider the optimization problem (7.13) with the terms* $\boldsymbol{\theta}^T T \boldsymbol{\theta}$ *and* $\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u})$ *to be minimized with respect to* $\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}$. *The maximization of these terms satisfy the statement:*

$$\max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} \boldsymbol{\theta}^T T \boldsymbol{\theta} + 2\,\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u}) \leq \max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} \boldsymbol{\theta}^T T \boldsymbol{\theta} + \max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} 2\,\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u}) \tag{7.21}$$

*for all possible trajectories* $\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}$. *In (7.21) the equality holds if and only if both terms reach their maximum for the same vector* $\boldsymbol{\theta}$.

*The term* $\boldsymbol{\theta}^T T \boldsymbol{\theta} = \text{trace}(T)$ *represents a constant, i.e.* $\boldsymbol{\theta}^T T \boldsymbol{\theta}$ *adopts its maximum value for every* $\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}$*. Hence, the disturbance vector* $\boldsymbol{\theta}^*$ *which maximizes the term* $\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u})$ *also maximizes the term* $\boldsymbol{\theta}^T T \boldsymbol{\theta}$*. As a consequence, the equality in (7.21) holds for the optimization problem (7.13).*

An alternative demonstration to obtain an explicit expression of the exact worst case cost is given in the Appendix (A.8). This method is based on the diagonal and semi-definite positive character of the matrix $W$ and is not limited to second order Volterra series models but can be used for other non-autoregressive models, e.g. finite impulse response models or step response models.

### 7.1.3 Control strategy using the exact worst case cost

The presented MMMPC problem (7.1) can be solved easily with the explicit expression of the worst case cost $J^*(\boldsymbol{u})$ (7.19). In fact, using the explicit solution, the original optimization problem is reduced to a minimization problem similar to the ones presented in Chapter 4 and Chapter 5 and the solution is defined as shown in (7.10).

The variable $\boldsymbol{q}(\boldsymbol{u})$ in the exact solution (7.19) can be rewritten with the definitions (7.14) and (7.9) as $W^T \left(G\boldsymbol{u} + \mathbf{f}(\boldsymbol{u}) + \boldsymbol{c} - \boldsymbol{r}\right)$. With the exact solution of the worst case cost the optimization problem (7.10) to compute an input sequence can be rewritten as:

$$\begin{aligned} \boldsymbol{u}^* = \quad \arg \quad & \min_{\boldsymbol{u}} \; J(\boldsymbol{u}, \mathbf{0}) + Nw^2 + 2\|W^T \left(G\boldsymbol{u} + \mathbf{f}(\boldsymbol{u}) + \boldsymbol{c} - \boldsymbol{r}\right)\|_1 \\ & \text{s.t. } L_c \boldsymbol{u} \leq \boldsymbol{b}_c \end{aligned} \tag{7.22}$$

subject to linear constraints. The calculated input sequence $\boldsymbol{u}^*$ can then be used in an MPC strategy where in every sampling period only the first element of $\boldsymbol{u}^*$ is applied to the controlled system.

Note that the use of an explicit solution of the maximization problem reduces considerably the computational complexity of the control strategy with respect to strategies using a vertex search approach or an upper bound of the worst case cost. Furthermore, the term $Nw^2$ in (7.22) does not depend on $\boldsymbol{u}$ and can be removed from the minimization problem. The last term in (7.22) consists only of a summation of absolute values and on simple matrix and vector operations. Therefore, the optimization problem considering an uncertainty or disturbance has a computational complexity similar to the optimization of the nominal model. Finally, the calculation of the input sequence can be carried out solving the optimization problem with nonlinear programming methods as sequential quadratic programming (SQP).

## 7.2 Volterra based MMMPC with exact worst case cost and guaranteed stability

In this section, the nonlinear MMMPC strategy based on a second order Volterra series model will be reformulated with some minor changes in the cost function and an additional condition for the prediction horizon in order to achieve input-to-state stability. In the first place, the control strategy with guaranteed stability will be given. Then, the optimization problem will be transformed to its state-space representation and a simple feasible solution to the optimization problem will be defined. Finally, based on the feasible solution, input-to-state stability of the proposed control strategy can be proven.

In a first step, the following quadratic performance index:

$$J(\boldsymbol{u},\boldsymbol{\theta}) = (\hat{\boldsymbol{y}}-\boldsymbol{r})^T(\hat{\boldsymbol{y}}-\boldsymbol{r}) + \lambda(\boldsymbol{u}-\boldsymbol{u}_r)^T(\boldsymbol{u}-\boldsymbol{u}_r) \tag{7.23}$$

will be used in the optimization where the weighting term $\lambda\boldsymbol{\Delta u}^T\boldsymbol{\Delta u}$ used in the cost function (2.47) has been substituted by $\lambda(\boldsymbol{u}-\boldsymbol{u}_r)^T(\boldsymbol{u}-\boldsymbol{u}_r)$. The vector $\boldsymbol{u}_r \in \mathbb{R}^N$ contains the necessary steady-state input for the given reference $\boldsymbol{r}$ and has been defined in (4.30). The argument $\boldsymbol{\theta}$ in the definition of the cost function indicates that the predicted output $\hat{\boldsymbol{y}}$ depends on the uncertainty $\boldsymbol{\theta}$. Then, using the second order Volterra series prediction model with uncertainty (7.3)-(7.4) in (7.23), the performance index can be expressed in the form given in (7.8) with the partly modified matrices defined by (compare (7.9) in Section 7.1.1):

$$\begin{array}{rclcrcl}
M_{\theta\theta} &=& W^TW, & & M_{uf} &=& G^T \\
M_{\theta u} &=& W^TG, & & \boldsymbol{m}_{uc} &=& G^T(\boldsymbol{c}-\boldsymbol{r}) - \lambda\boldsymbol{u}_r \\
M_{\theta f} &=& W^T, & & M_{ff} &=& I_N \\
\boldsymbol{m}_{\theta c} &=& W^T(\boldsymbol{c}-\boldsymbol{r}), & & \boldsymbol{m}_{fc} &=& \boldsymbol{c}-\boldsymbol{r} \\
M_{uu} &=& G^TG + \lambda I_N, & & m_{cc} &=& \boldsymbol{c}^T\boldsymbol{c} + \boldsymbol{r}^T\boldsymbol{r} - 2\,\boldsymbol{c}^T\boldsymbol{r} + \lambda\boldsymbol{u}_r^T\boldsymbol{u}_r
\end{array} \tag{7.24}$$

With $M_{\theta\theta} = W^TW = w^2I_N$ the cost function is convex in $\boldsymbol{\theta}$ and the maximization of (7.23) can be found at least in one of the vertices of $\boldsymbol{\Theta}$. Hence, the maximization problem is defined by (7.12) where $J^*(\boldsymbol{u})$ denotes the resulting worst case cost. Finally, based on $J^*(\boldsymbol{u})$, the input sequence can be calculated with:

$$\begin{array}{rcl}
\boldsymbol{u}^* &=& \arg\min\limits_{\boldsymbol{u}} J^*(\boldsymbol{u}) \\
& & \text{s.t. } L_c\boldsymbol{u} \le \boldsymbol{b}_c \\
& & \quad u(k+N_u-1|k) = u_r
\end{array} \tag{7.25}$$

where the equality constraint is not necessary for the upcoming stability proof, but it has been included to allow the use of the vectors and matrices defined in the Appendix

A.1 (for further details see the Remark 7.2). Using the explicit formulation of the worst case cost (7.19) and $\boldsymbol{q}(\boldsymbol{u}) = W^T\left(G\boldsymbol{u} + \mathbf{f}(\boldsymbol{u}) + \boldsymbol{c} - \boldsymbol{r}\right)$ (7.14), the min-max optimization problem can be expressed as:

$$\begin{aligned} \boldsymbol{u}^* = \quad \arg \quad & \min_{\boldsymbol{u}} \; J(\boldsymbol{u}, \mathbf{0}) + Nw^2 + 2\|W^T\left(G\boldsymbol{u} + \mathbf{f}(\boldsymbol{u}) + \boldsymbol{c} - \boldsymbol{r}\right)\|_1 \\ & \text{s.t. } L_c\boldsymbol{u} \leq \boldsymbol{b}_c \\ & \quad u(k + N_u - 1|k) = u_r \end{aligned} \tag{7.26}$$

and the resulting input sequence can be used in a receding horizon control strategy where the first element of $\boldsymbol{u}^*$ is applied to the system. Note that the effect of the modified cost function influences only the nominal cost $J(\boldsymbol{u}, 0)$ and has no effect on the computation of the worst case cost.

For the presented MMMPC (7.26) based on the modified cost function (7.23) input-to-state stability can be proven for a prediction horizon satisfying the condition $N \geq N_t + N_u$. In the following sections the stability of the control strategy (7.26) will be proven and the necessary conditions will be given.

### 7.2.1 Stability proof

In this section the stability of the proposed MMMPC strategy (7.26) based on a second order Volterra series model will be proven. In a first step, the considered optimization problem will be reformulated in state-space representation. For the state-space optimization problem a feasible solution will be determined. Finally, input-to-state stability for the proposed MMMPC strategy can be proven showing the convergence of the resulting cost. The stability proof is formulated in a general manner and holds for other non-autoregressive models with additive uncertainty.

#### 7.2.1.1 Optimization problem in state-space representation

The prediction model based on a second order Volterra series model considering an estimation error and a future uncertainty can be expressed, analogously to (2.61), as[3]:

$$\begin{aligned} \boldsymbol{x}(k + i + 1|k) &= A\boldsymbol{x}(k + i|k) + Bu(k + i|k) \\ y(k + i|k) &= \mathrm{f}(\boldsymbol{x}(k + i|k)) + d(k) + \theta(k + i|k) \end{aligned} \tag{7.27}$$

[3]Note that in following stability proof the uncertainty $\theta(k+i|k)$ in (7.27) is not scaled by the factor $w$ in order to keep the mathematical notation simple. The use of the scaling in the previous sections simplified the computation of the exact worst case cost, but does not influence the proof of stability.

where the uncertainty is bounded by $\theta(\cdot) \leq \varepsilon$. With respect to the nominal model (2.60), the output prediction considering the current estimation error and future disturbance is defined by:

$$y(k+i|k) = \tilde{y}(k+i|k) + d(k) + \theta(k+i|k) \tag{7.28}$$

with $\tilde{y}(k+i|k)$ being the nominal model output. Then, with the model considering an estimation error and an uncertainty, the optimization problem of the MMMPC strategy (7.26) proposed in Section 7.2 can be rewritten in a general manner in state-space as[4]:

$$\begin{aligned}
\min_{\boldsymbol{u}(k)} \max_{\boldsymbol{\theta}(k)} \quad & J(\boldsymbol{x}(k), \boldsymbol{u}(k), d(k), \boldsymbol{\theta}(k)) \\
\text{s.t. } & u(k+i|k) \in U, && i = 0, \ldots, N_u - 1 \\
& h(\boldsymbol{x}(k+i|k)) \in U, && i = N_u, \ldots, N-1 \\
& \boldsymbol{x}(k+i|k) \in X, && i = 1, \ldots, N-1 \\
& \boldsymbol{x}(k+N|k) \in \Omega(d(k), \theta(k+N|k)) \\
& \theta(k|k) = 0 \\
& \theta(k+i|k) \in \Theta, && i = 1, \ldots, N-1
\end{aligned} \tag{7.29}$$

where the set $\Theta$ is defined by the bound $\theta(\cdot) \leq \varepsilon$. The input sequence $\boldsymbol{u}(k)$ is calculated considering the worst case cost with respect to the sequence of future uncertainties $\boldsymbol{\theta}(k)$. The optimization of the cost is carried out over the finite prediction horizon $N$ and the finite control horizon $N_u$ with $N_u \leq N$. The initial state is denoted $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k+i|k)$ for $i = 1, \ldots, N$ represent the predictions of the future states made at $k$. These predictions are calculated with the input signal $u(k+i|k)$ for $i = 0, \ldots, N_u - 1$ and with the local control law $h(\boldsymbol{x}(k+i|k))$ for $i = N_u, \ldots, N-1$. The sequence of future uncertainties is defined generally by:

$$\boldsymbol{\theta}(k) = \; [\theta(k|k), \;\; \theta(k+1|k), \;\; \ldots, \;\; \theta(k+N-1|k)]^T \tag{7.30}$$

and the sequence $\boldsymbol{u}(k)$ calculated at $k$ minimizing the cost function (7.29) is defined in a general manner as:

$$\boldsymbol{u}(k) = \; [u(k|k), \;\; u(k+1|k), \;\; \ldots, \;\; u(k+N_u-1|k)]^T \tag{7.31}$$

The terminal set is given by:

$$\Omega(d(k), \theta(k+N-1|k)) = \{\boldsymbol{x} : \mathrm{f}(\boldsymbol{x}(k+N|k)) + d(k) + \theta(k+N-1|k) = r(k)\} \tag{7.32}$$

and has been included in (7.29) to assure that the output prediction for $k+N+1$ made at $k$ corresponds to the given reference $r(k)$. Depending on the input signal $u(k+i|k)$

[4]The equality constraint $\theta(k|k) = 0$ considered in the optimization problem (7.29) is necessary to proof stability, but can be disregarded in the implementation of the control strategy. For further details see the Remark 7.3.

for $i = 0, \ldots, N_u - 1$ and the local control law $h(\boldsymbol{x}(k+i|k))$ for $i = N_u, \ldots, N$ the prediction of the successor state is calculated with

$$\begin{aligned} \boldsymbol{x}(k+i+1|k) &= \boldsymbol{\phi}(\boldsymbol{x}(k+i|k), u(k+i|k)), \qquad i = 0, \ldots, N_u - 1 \\ \boldsymbol{x}(k+i+1|k) &= \boldsymbol{\phi}(\boldsymbol{x}(k+i|k), h(\boldsymbol{x}(k+i|k))), \quad i = N_u, \ldots, N \end{aligned} \tag{7.33}$$

With the initial state $\boldsymbol{x}(k)$, the future input sequence $\boldsymbol{u}(k)$, the local control law $h(\boldsymbol{x}(k+i|k))$ and the sequence of the uncertainty $\boldsymbol{\theta}(k)$ as well as the current estimation error $d(k)$, the cost function $J(\cdot,\cdot,\cdot,\cdot)$ used in (7.29) is defined as:

$$\begin{aligned} J(\boldsymbol{x}(k), \boldsymbol{u}(k), d(k), \boldsymbol{\theta}(k)) = & \sum_{i=0}^{N_u-1} L(\boldsymbol{x}(k+i|k), u(k+i|k), d(k), \theta(k+i|k)) + \\ & \sum_{i=N_u}^{N} L_h(\boldsymbol{x}(k+i|k)) \end{aligned} \tag{7.34}$$

with the quadratic stage costs $L(\cdot,\cdot,\cdot,\cdot)$ and $L_h(\cdot)$ given by:

$$\begin{aligned} & L(\boldsymbol{x}(k+i|k), u(k+i|k), d(k), \theta(k+i|k)) = \\ & \quad \|\mathrm{f}(\boldsymbol{x}(k+i|k)) + d(k) + \theta(k+i|k) - r(k)\|_Q^2 + \|u(k+i|k) - u_r(k)\|_R^2 \\ & L_h(\boldsymbol{x}(k+i|k)) = \\ & \quad \|\mathrm{f}(\boldsymbol{x}(k+i|k)) + d(k) + \theta(k+i|k) - r(k)\|_Q^2 + \|h(\boldsymbol{x}(k+i|k)) - u_r(k)\|_R^2 \end{aligned} \tag{7.35}$$

where f is a nonlinear function defined in (2.57), $r(k)$ denotes the desired reference for the system output and $u_r(k)$ is the necessary steady-state input signal for a given reference.

For the later proof of convergence the last element of the output prediction should satisfy $y(k+N-1|k) = r(k)$. Under this assumption the prediction $y(k+N-1|k)$ (7.28) can be substituted by the desired reference $r(k)$ and the predicted output of the nominal model for $k+N-1$ made at $k$ can be written as:

$$\tilde{y}(k+N-1|k) = r(k) - d(k) - \theta(k+N-1|k) \tag{7.36}$$

That means that the steady-state input has to be chosen in a way that the predicted nominal output $\tilde{y}(k+N-1|k)$ corresponds to (7.36). Hence, the steady-state input is based on the predicted nominal model output for $k+N-1$ made at $k$ and can be written as[5]:

$$\begin{aligned} u_r(k) &= \chi^{-1}(\tilde{y}(k+N-1|k)) \\ &= \chi^{-1}(r(k) - d(k) - \theta(k+N-1|k)) \end{aligned} \tag{7.37}$$

[5]Note that the steady-state input depends on the given reference $r(k)$, the known estimation error $d(k)$ and the uncertainty $\theta(k+N-1|k)$. For the sake of simplicity, the notation $u_r(k) = u_r(r(k), d(k), \theta(k+N-1|k))$ has been chosen. Furthermore it has to be mentioned that a constant reference has been assumed, i.e. $r(k) = r(k+1)$.

The given steady-state input (7.37) is used to define the local control law used in the initial optimization problem (7.29):

$$h(\boldsymbol{x}(k+i|k)) = u_r(k), \quad i = N_u, \ldots, N-1 \tag{7.38}$$

#### 7.2.1.2 Feasibility of the solution

Consider the sequence:

$$\boldsymbol{u}^*(k) = \; [u^*(k|k), \;\; u^*(k+1|k), \;\; \ldots, \;\; u^*(k+N_u-1|k)]^T \tag{7.39}$$

being at $k$ the optimal solution for the MMMPC problem (7.29) with the associated cost $J^*(\boldsymbol{x}(k))$ and the predicted states $\boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots, N$. Furthermore, consider the shifted solution $\boldsymbol{u}^f(k+1)$ for $k+1$:

$$\boldsymbol{u}^f(k+1) = \; [u^f(k+1|k+1), \;\; u^f(k+2|k+1), \;\; \ldots, \;\; u^f(k+N_u|k+1)]^T \tag{7.40}$$

where the elements can be defined by means of the optimal solution in $k$ and the local control law:

$$u^f(k+i|k+1) = \begin{cases} u^*(k+i|k) & \text{for } i = 1, \ldots N_u - 1 \\ h(\boldsymbol{x}^f(k+N_u|k+1)) & \text{for } i = N_u \end{cases} \tag{7.41}$$

The states predicted with the shifted solution $\boldsymbol{u}^f(k+1)$ are denoted $\boldsymbol{x}^f(k+N_u|k+1)$ for $i = 2, \ldots, N+1$ and $J^f(\boldsymbol{x}(k+1))$ represents the resulting cost.

With the optimal input sequence $\boldsymbol{u}^*(k)$ and the local control law, corresponding to the necessary steady-state input $u_r(k)$, the states $\boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots N$ can be predicted. As the optimal solution and the local control law are computed under consideration of the conditions given in (7.29), the statement:

$$\boldsymbol{x}^*(k+N|k) \in \Omega(d(k), \theta(k+N|k)) \tag{7.42}$$

is true.

Taking into account that the estimation error $d(k)$ and the uncertainty $\theta(k+i|k)$ for $k = 1, \ldots, N-1$ have no effect on the predictions, the optimal predicted state satisfies $\boldsymbol{x}^*(k+1|k) = \boldsymbol{x}(k+1)$. With $\boldsymbol{u}^f(k+1)$ being the shifted sequence $\boldsymbol{u}^*(k)$, i.e. $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \ldots N_u - 1$, the predictions satisfy:

$$\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k), \quad i = 2, \ldots N_u \tag{7.43}$$

During the remaining prediction horizon the local control law will be used to predict the states $\boldsymbol{x}^f(k+i|k+1)$ for $i = N_u+1, \ldots N+1$. Now, with the local control law

computed according to the conditions defined in the optimization problem (7.29), the state prediction for $k+N+1$ made at $k+1$ satisfies:

$$\boldsymbol{x}^f(k+1+2|k+1) \in \Omega(d(k), \theta(k+N+1|k+1)) \tag{7.44}$$

Hence, the shifted sequence $\boldsymbol{u}^f(k+1)$ can be considered a feasible solution to the optimization problem in $k+1$.

#### 7.2.1.3 Convergence

Consider the cost $J^*(\boldsymbol{x}(k))$ at $k$ based on the optimal solution $\boldsymbol{u}^*(k)$ minimizing the problem (7.29). Furthermore consider the cost $J^f(\boldsymbol{x}(k+1))$ at $k+1$ depending on the feasible solution $\boldsymbol{u}^f(k+1)$ (see section 7.2.1.2). Convergence can be guaranteed if the calculated cost for $k+1$ is monotonically decreasing with respect to the cost for $k$.

With the general definition of the cost function (7.34), the optimal cost $J^*(\boldsymbol{x}(k))$ at $k$ is given by:

$$\begin{aligned} J^*(\boldsymbol{x}(k)) \quad = \quad & \sum_{i=0}^{N_u-1} L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k), d(k), \theta(k+i|k)) + \\ & \sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}^*(k+i|k)) \end{aligned} \tag{7.45}$$

Analogously, the cost $J^f(\boldsymbol{x}(k+1))$ at $k+1$ based on the feasible solution (7.40) can be expressed as:

$$\begin{aligned} & J^f(\boldsymbol{x}(k+1)) = \\ & \sum_{i=1}^{N_u-1} L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1), d(k+1), \theta(k+i|k+1)) + \\ & \sum_{i=N_u}^{N} L_h(\boldsymbol{x}^f(k+i|k+1)) \end{aligned} \tag{7.46}$$

With (7.45) and (7.46) the difference of the cost functions, defined as $\Delta J(k+1) =$

$J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k))$, becomes:

$$\begin{aligned}\Delta J(k+1) = \\ & L_h(\boldsymbol{x}^f(k+N|k+1)) - L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), \theta(k|k)) + \\ & \sum_{i=1}^{N_u-1} \Big( L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1), d(k+1), \theta(k+i|k+1)) - \\ & \qquad L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k), d(k), \theta(k+i|k)) \Big) + \\ & \sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}^f(k+i|k+1)) - L_h(\boldsymbol{x}^*(k+i|k))\end{aligned} \tag{7.47}$$

Now consider the term $L_h(\boldsymbol{x}^f(k+N|k+1))$ and the nilpotent character ($A^{N_t} = 0$) of the used prediction model (7.27). For a prediction horizon of $N \geq N_u + N_t$, the local control law, defined by the steady-state input (7.38), is used (at least) in the last $N_t$ sampling periods. As a consequence of the property $A^{N_t} = 0$, the state $\boldsymbol{x}^f(k+N|k+1)$ for $k+N$ at $k+1$ reaches steady state. Taking into account the definition of the steady-state input (7.37) it is clear that the nominal output in $k+N$ is given by $\tilde{y}(k+N|k+1) = r(k+1) - d(k+1) - \theta(k+N|k+1)$. From the definition of the nominal output (4.42), it follows that:

$$\mathrm{f}(\boldsymbol{x}^f(k+N|k+1)) = r(k+1) - d(k+1) - \theta(k+N|k+1) \tag{7.48}$$

Finally, with (7.48) and the local control law $h(\boldsymbol{x}^f(k+N|k+1)) = u_r(k+1)$ it can be shown that $L_h(\boldsymbol{x}^f(k+N|k+1)) = 0$. Hence, the difference of the two cost functions (7.47) can be written in the following form:

$$\Delta J(k+1) = -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), \theta(k|k)) + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \tag{7.49}$$

with the terms given by:

$$\alpha_1 = \sum_{i=1}^{N_u-1} \Big( \|\mathrm{f}(\boldsymbol{x}^f(k+i|k+1)) + \theta(k+i|k+1) + d(k+1) - r(k+1)\|_Q^2 - \|\mathrm{f}(\boldsymbol{x}^*(k+i|k)) + \theta(k+i|k) + d(k) - r(k)\|_Q^2 \Big) \tag{7.50}$$

$$\alpha_2 = \sum_{i=1}^{N_u-1} \Big( \|u^f(k+i|k+1) - u_r(k+1)\|_R^2 - \|u^*(k+i|k) - u_r(k)\|_R^2 \Big) \tag{7.51}$$

$$\alpha_3 = \sum_{i=N_u}^{N-1} \Big( \|\mathrm{f}(\boldsymbol{x}^f(k+i|k+1)) + \theta(k+i|k+1) + d(k+1) - r(k+1)\|_Q^2 - \|\mathrm{f}(\boldsymbol{x}^*(k+i|k)) + \theta(k+i|k) + d(k) - r(k)\|_Q^2 \Big) \tag{7.52}$$

$$\alpha_4 = \sum_{i=N_u}^{N-1} \Big( \|h(\boldsymbol{x}^f(k+i|k+1)) - u_r(k+1)\|_R^2 - \|h(\boldsymbol{x}^*(k+i|k)) - u_r(k)\|_R^2 \Big) \tag{7.53}$$

The Lemmas 4.1-4.4 presented in Chapter 4 will be used to define upper bounds for the terms $\alpha_1$, $\alpha_2$, $\alpha_3$ and $\alpha_4$ used in the difference of the cost functions given in (7.49).

**Term $\boldsymbol{\alpha_1}$:** With $\boldsymbol{x}^*(k+1|k) = \boldsymbol{x}(k+1)$ and $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \ldots, N_u-1$, the predicted states computed with the optimal and the feasible solutions satisfy $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots, N_u-1$. For the reference applies $r(k) = r(k+1)$ and the increments in the estimation error and the uncertainty are given generally as $\Delta d = d(k+1) - d(k)$ and $\Delta\theta(k+i) = \theta(k+i|k+1) - \theta(k+i|k)$, respectively. Defining the auxiliary variable $z(k+i|k) = \mathrm{f}(\boldsymbol{x}^*(k+i|k)) + d(k) + \theta(k+i|k) - r(k)$, the term $\alpha_1$ can be expressed as:

$$\alpha_1 = \sum_{i=1}^{N_u-1} \|z(k+i|k) + \Delta d + \Delta\theta(k+i)\|_Q^2 - \|z(k+i|k)\|_Q^2 \tag{7.54}$$

and bounded under consideration of Lemma 4.1 by:

$$\alpha_1 \leq L_q \sum_{i=1}^{N_u-1} \|\Delta d + \Delta\theta(k+i)\| \tag{7.55}$$

With the uncertainty limited by $\|\theta(\cdot)\| \leq \varepsilon$ the increment in the uncertainty is bounded by $\|\Delta\theta(\cdot)\| \leq 2\,\varepsilon$. With this boundary the term $\alpha_1$ is limited to:

$$\alpha_1 \leq c_1(Q, N_u) \cdot \|\Delta d + 2\,\varepsilon\| \tag{7.56}$$

being $c_1(\cdot,\cdot)$ a constant depending on the weighting factor $Q$ and the control horizon $N_u$.

□

**Term $\alpha_2$:** Consider the optimal solution and the feasible solution satisfying $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \ldots, N_u - 1$ (7.41). Defining the increment in the steady-state input as $\Delta u_r = u_r(k+1) - u_r(k)$, the term $\alpha_2$ can be expressed with the auxiliary variable $z_1(k+i|k) = u^*(k+i|k) - u_r(k)$ in the form:

$$\alpha_2 = \sum_{i=1}^{N_u-1} \|z_1(k+i|k) - \Delta u_r\|_R^2 - \|z_1(k+i|k)\|_R^2 \tag{7.57}$$

Then, applying Lemma 4.1 to (7.57), the term $\alpha_2$ can be bounded by:

$$\alpha_2 \leq c_2(R, N_u) \cdot \|\Delta u_r\| \tag{7.58}$$

Taking in account the definition (7.37), the increment in the steady-state input can be written in the following form:

$$\begin{array}{rcl} \Delta u_r & = & \chi^{-1}(r(k+1) - d(k+1) - \theta(k+N|k+1)) - \\ & & \chi^{-1}(r(k) - d(k) - \theta(k+N-1|k)) \end{array} \tag{7.59}$$

The increment in the estimation error is defined as $\Delta d = d(k+1) - d(k)$ and the difference between the two shifted uncertainty terms can be written as $\Delta\theta^+ = \theta(k+N|k+1) - \theta(k+N-1|k)$. Then, with $r(k) = r(k+1)$ and the auxiliary variable $z_2 = r(k) - d(k) - \theta(k+N-1|k)$ the increment in the steady-state input can be expressed as:

$$\Delta u_r = \chi^{-1}(z_2 - \Delta d - \Delta\theta^+) - \chi^{-1}(z_2) \tag{7.60}$$

Under consideration of Lemma 4.2 the norm of $\Delta u_r$ can be bounded by $\|\Delta u_r\| \leq L_\chi\|\Delta d + \Delta\theta^+\|$. With the uncertainty limited by $\|\theta(\cdot)\| \leq \varepsilon$ the difference between the two shifted uncertainty terms is bounded by $\|\Delta\theta^+\| \leq 2\,\varepsilon$. Hence, the increment in the steady-state input is bounded by:

$$\|\Delta u_r\| \leq L_\chi\|\Delta d + 2\,\varepsilon\| \tag{7.61}$$

Using (7.61) in (7.58), the term $\alpha_2$ can be finally bounded by:

$$\alpha_2 \leq c_2(R, L_\chi, N_u) \cdot \|\Delta d + 2\,\varepsilon\| \tag{7.62}$$

where $c_2(\cdot,\cdot,\cdot)$ represents a constant parameter.

□

**Term $\alpha_3$:** Consider the predictions $\boldsymbol{x}^*(k+i|k)$ and $\boldsymbol{x}^f(k+i|k+1)$ for $i = N_u, \dots N-1$ made at $k$ and $k+1$ with the optimal and the feasible solution, respectively. The difference between these predictions is defined as $\Delta\boldsymbol{x}(k+i) = \boldsymbol{x}^f(k+i|k+1) - \boldsymbol{x}^*(k+i|k)$ for $i = N_u, \dots N-1$ and the initial condition is $\boldsymbol{x}^f(k+N_u-1|k+1) = \boldsymbol{x}^*(k+N_u-1|k)$. Furthermore, consider the increment in the estimation error $\Delta d = d(k+1) - d(k)$ and the difference in the uncertainty $\Delta\theta(k+i) = \theta(k+i|k+1) - \theta(k+i|k)$. With the auxiliary variables:

$$\begin{aligned} z_1(k+i) &= \mathrm{f}(\boldsymbol{x}^*(k+i|k) + \Delta\boldsymbol{x}(k+i)) - \mathrm{f}(\boldsymbol{x}^*(k+i|k)) + \Delta d + \Delta\theta(k+i) \\ z_2(k+i) &= \mathrm{f}(\boldsymbol{x}^*(k+i|k)) + d(k) + \theta(k+i|k) - r(k) \end{aligned}$$

and a constant reference, i.e. $r(k+1) = r(k)$, the term $a_3$ can be expressed as:

$$\alpha_3 = \sum_{i=N_u}^{N-1} \|z_1(k+i) + z_2(k+i)\|_Q^2 - \|z_2(k+i)\|_Q^2 \tag{7.63}$$

Applying Lemma 4.1 to (7.63) the term $a_3$ can be bounded in the following form:

$$\alpha_3 \leq c_3(Q, N, N_u) \cdot \|z_1(k+i)\| \tag{7.64}$$

Furthermore, with the function f being Lipschitz continuous, the term $z_1(k+i|k)$ can be bounded by Lemma 4.3:

$$\begin{aligned} \|z_1(k+i|k)\| &\leq \|\mathrm{f}(\boldsymbol{x}^*(k+i|k) + \Delta\boldsymbol{x}(k+i)) - \mathrm{f}(\boldsymbol{x}^*(k+i|k))\| + \\ &\quad \|\Delta d + \Delta\theta(k+i)\| \\ &\leq L_f\|\Delta\boldsymbol{x}(k+i)\| + \|\Delta d + \Delta\theta(k+i)\| \end{aligned} \tag{7.65}$$

Hence, using (7.65) in (7.64), the upper bound of $\alpha_3$ can be expressed as:

$$\alpha_3 \leq c_3(Q, N, N_u) \cdot (L_f\|\Delta\boldsymbol{x}(k+i)\| + \|\Delta d + \Delta\theta(k+i)\|) \tag{7.66}$$

With the help of Lemma 4.4 the difference of the predicted states based on the optimal and the feasible solution can be bounded with $\|\Delta\boldsymbol{x}(k+i)\| \leq \|c_{\boldsymbol{x}}\,\Delta u_r\|$. Using this upper bound, (7.66) can be rewritten as:

$$\alpha_3 \leq c_3(Q, N, N_u) \cdot (L_f\|c_{\boldsymbol{x}}\,\Delta u_r\| + \|\Delta d + \Delta\theta(k+i)\|) \tag{7.67}$$

Finally, with Lemma 4.2 the increment in the steady-state input can be bounded by $\|\Delta u_r\| \leq L_\chi\|\Delta d + 2\,\varepsilon\|$ (see explanation for $\alpha_2$). Furthermore, with the uncertainty limited by $\|\theta(\cdot)\| \leq \varepsilon$ the difference in the uncertainty is bounded by $\|\Delta\theta(\cdot)\| \leq 2\,\varepsilon$. Hence, the upper bound of $\alpha_3$ (7.67) is defined by:

$$\alpha_3 \leq c_3(Q, L_\chi, L_f, c_{\boldsymbol{x}}, N, N_u) \cdot \|\Delta d + 2\,\varepsilon\| \tag{7.68}$$

and depends on the constant $c_3(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$ as well as on the increments in the estimation error and the bound of the uncertainty.

□

**Term $\alpha_4$:** Under consideration of the local control law (7.38), the statement:

$$\alpha_4 = 0 \tag{7.69}$$

holds.

□

Finally, upper bounds for the terms $\alpha_1$, $\alpha_2$ and $\alpha_3$ have been defined and it has been shown that $\alpha_4 = 0$. These upper bounds depend only on the bound of the uncertainty and the variation in the estimation error. Using these upper bounds in (7.49) the cost difference $\Delta J(k+1) = J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k))$ is bounded by:

$$J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k)) \leq -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), \theta(k|k)) + c_V \cdot \|\Delta d + 2\,\varepsilon\| \tag{7.70}$$

with $L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), \theta(k|k))$ being positive (except in the origin where this term converges to zero) and the parameter $c_V$ given by:

$$c_V = c_1(Q, N_u) + c_2(R, L_\chi, N_u) + c_3(Q, L_\chi, L_f, c_{\boldsymbol{x}}, N, N_u)$$

The terms $c_V \cdot \|\Delta d + 2\,\varepsilon\| > 0$ and $-L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), \theta(k|k)) \leq 0$ ensure that the cost based on the feasible solution cost will decrease as long as the stage cost satisfies $-L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), \theta(k|k)) > c_V \cdot \|\Delta d + 2\,\varepsilon\|$. Hence, the system is steered into the set::

$$\Psi_d = \{\boldsymbol{x}\colon\ L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), \theta(k|k)) \leq c_V \cdot \|\Delta d + 2\,\varepsilon\|\} \tag{7.71}$$

from any arbitrary $\boldsymbol{x}$. Nevertheless, when the state is inside the set $\Psi_d$, the system may remain in it or evolve out of it as it is not guaranteed that the cost decreases. For any $\boldsymbol{x}$, the stage cost always satisfies $-L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), \theta(k|k)) \leq 0$, hence (7.70) can be written in the form:

$$J^f(\boldsymbol{x}(k+1)) \leq J^*(\boldsymbol{x}(k)) + c_V \cdot \|\Delta d + 2\,\varepsilon\| \tag{7.72}$$

Besides, for any $\boldsymbol{x}(k) \in \Psi_d$, the inequality:

$$J^*(\boldsymbol{x}(k)) + c_V \cdot \|\Delta d + 2\,\varepsilon\| \leq \max_{\boldsymbol{x} \in \Psi_d} J^*(\boldsymbol{x}) + c_V \cdot \|\Delta d + 2\,\varepsilon\| = \beta_d \tag{7.73}$$

holds. Now, from (7.72) and (7.73) follows that:

$$J^f(\boldsymbol{x}(k+1)) \leq \beta_d, \quad \forall\ \boldsymbol{x}(k) \in \Psi_d \tag{7.74}$$

Whenever the state enters into $\Psi_d$, it evolves into the set:

$$\Psi_\beta = \{\boldsymbol{x}\colon J^f(\boldsymbol{x}) \leq \beta_d\} \tag{7.75}$$

Finally, the system may evolve out of $\Psi_d$, but will remain in the set $\Psi_\beta$. From the set $\Psi_\beta$ the system will be steered again into $\Psi_d$ and so on. As a consequence, the state is ultimately bounded and the system is stabilized using the feasible solution.

Furthermore, for the cost $J^*(\boldsymbol{x}(k+1))$ based on the optimal solution $\boldsymbol{u}^*(k+1)$ at $k+1$ the statement:

$$J^*(\boldsymbol{x}(k+1)) \leq J^f(\boldsymbol{x}(k+1)) \tag{7.76}$$

holds. Then, with (7.74), (7.75) and (7.76) the proposed MMMPC strategy is also ultimately bounded by the set $\Psi_\beta$. Hence, the MMMPC strategy based on a second order Volterra series model is input-to-state stable and maintains the system inside the set $\Psi_\beta$.

**Remark 7.2** *The equality constraint for the last element of the input sequence in (7.26), also considered in the state-space optimization problem (7.29), is not necessary to assure stability, i.e. the presented stability proof also holds without this constraint. The consideration of this constraint allows to use the matrices and vectors defined in the Appendix A.1. Without this constraint, the matrices and vectors used in the prediction model (7.3)-(7.4) have to be modified in order to consider the local control law* $h(\boldsymbol{x}(k+i|k)) = u_r$ *for* $i = N_u, \ldots, N-1$.

**Remark 7.3** *Without loss of generality the uncertainty in k can be assumed to satisfy* $\theta(k|k) = 0$ *as any error in the output has been already considered by the estimation error* $d(k)$. *The equality constraint is necessary for the stability proof as the stage cost* $L(\cdot,\cdot,\cdot,\cdot)$ *for k has to be considered in the cost function (7.34). In contrast, the effective implementation of the proposed min-max control strategy considers only the future output values and, as a consequence, does not require this constraint as the uncertainty* $\theta(k|k)$ *is not included in the optimization problem.*

## 7.3 Experimental results

The two proposed MMMPC strategies based on second order Volterra series model considering an additive uncertainty were implemented in the Matlab/Simulink environment and applied to the pilot plant emulating an exothermic chemical reaction. The maximal uncertainty has been determined with the help of the two data sets used in Section 3.1.3 to identify and validate a second order Volterra series model. The maximum error between the measured output and the model output represents the maximum uncertainty which has to be considered in the min-max control strategy. Finally, exploiting the mentioned data sets, a value of $w = 6$ has been identified. In the following sections, the experimental results obtained with the proposed min-max control strategies will be presented.

### 7.3.1 Volterra based MMMPC with exact worst case cost

The MMMPC strategy presented in Section 7.1.3 was tested on the pilot plant by means of setpoint tracking and disturbance rejection experiments. The minimization of the worst case cost was carried out by Sequential Quadratic Programming (SQP) and for the computation of the worst case the explicit solution (7.19) was used. The control strategy was implemented with a prediction horizon of $N = 25$, a control horizon of $N_u = 15$, a weighting of the control effort of $\lambda = 5$ and a bound of $w = 6$ for the uncertainty. Furthermore, the following input constraints have been considered in the computation of the input sequence:

$$\begin{array}{rcccl} 5 & \leq & u(k+i|k) & \leq & 100, \quad i = 0, \ldots, 14 \\ -20 & \leq & \Delta u(k+i|k) & \leq & 20, \quad i = 0, \ldots, 14 \end{array} \tag{7.77}$$

Note that the constraints $\theta(k+i|k) \leq 1$ for $i = 1, \ldots, 25$ are already included in the computation of the worst case cost and do not require a consideration during the minimization of the resulting cost.

In a first step, the proposed MMMPC strategy was validated in a setpoint tracking experiment (see Fig. 7.1). Starting in steady state and with a reference of 55 °C, the setpoint was changed in $t = 30$ min to a value of 65 °C and in $t = 90$ min to 45 °C. After the setpoint changes a small overshoot of approximately 1 °C and −0.5 °C can be observed. After the stabilization of the temperature, the input signal shows only small changes, necessary to maintain the system output in the given reference.

The second experiment with the proposed min-max control strategy was carried out with an error in the emulated exothermic chemical reaction (see Fig. 7.2). The error was introduced in the activation energy $E$ and held constant during the entire experiment. With the parameter $E$ having a strong influence on the dynamics of the underlying chemical model, the value of the mentioned parameter was increased only by 3 %. Furthermore, the setpoint changes applied to the system were the same as the ones shown in Fig. 7.1. In the experimental results shown in Fig. 7.2, the setpoint changes result in an overshoot of 1.2 °C and −1.3 °C, only slightly higher than in the experiments with the nominal model. Generally, the control strategy stabilizes the system in the given setpoint and controls the system without difficulties in spite of the introduced error.

In the third experiment, the disturbance rejection capabilities of the proposed control strategy were validated by means of an additive disturbance in the system input (see Fig. 7.3). The additive disturbance had a value of $\Delta v_8 = -15$ % and was active from $t = 70$ min until $t = 110$ min. In absence of the disturbance, the valve opening is

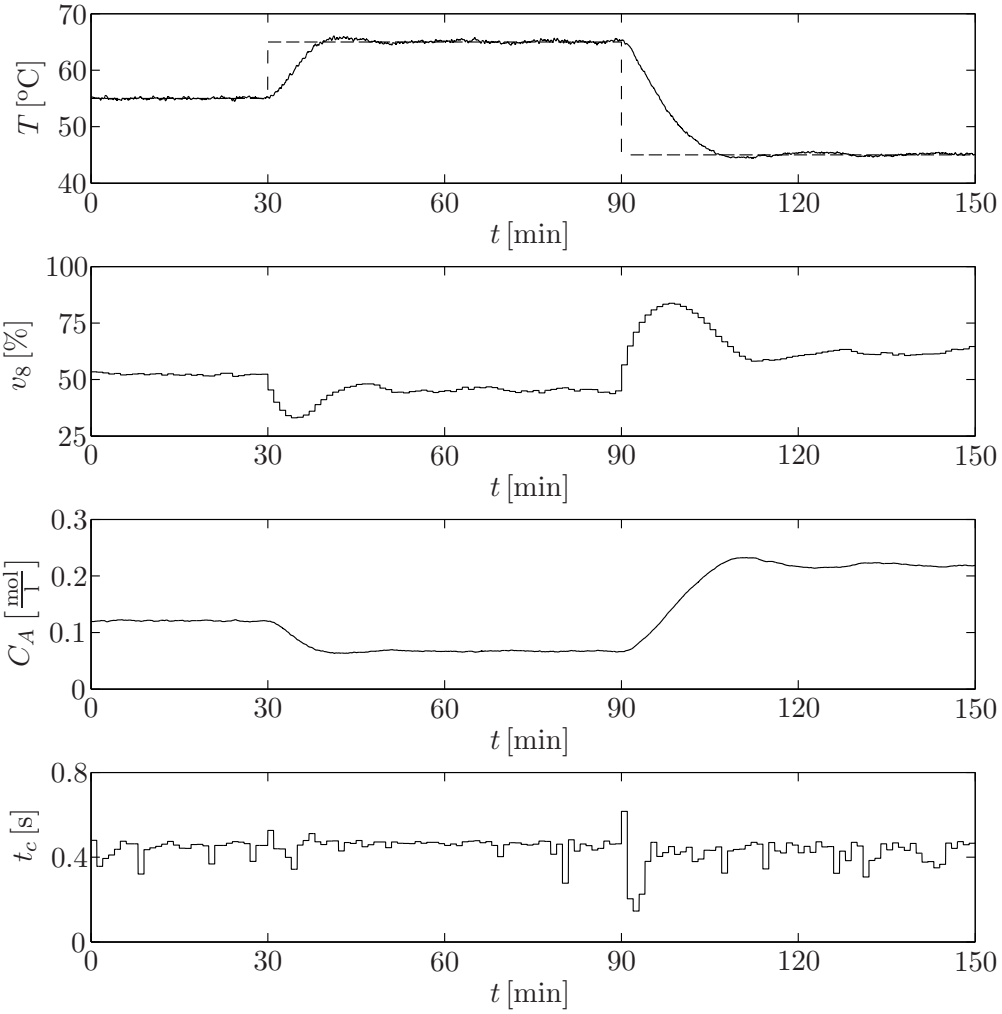


**Figure 7.1**: Setpoint tracking experiment controlled by the proposed MMMPC strategy. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and computation time $t_c$.

given by $v_8 = u$, i.e. the valve opening corresponds to the value computed by the control strategy. During the application of the disturbance, the effective valve opening is defined by $v_8 = u + \Delta v_8$. After the application of the disturbance, the control strategy efficiently compensates the error in the temperature by increasing the control signal $u$. After the sudden disappearance of the disturbance, the effective opening of the valve is to high and results in a negative error in the temperature. The reduction of the valve opening compensates the output error and stabilizes the system in the desired reference. The results show that the control strategy efficiently rejects the disturbance in the system input.

Finally, in the last experiment (see Fig. 7.4) the proposed MMMPC strategy was

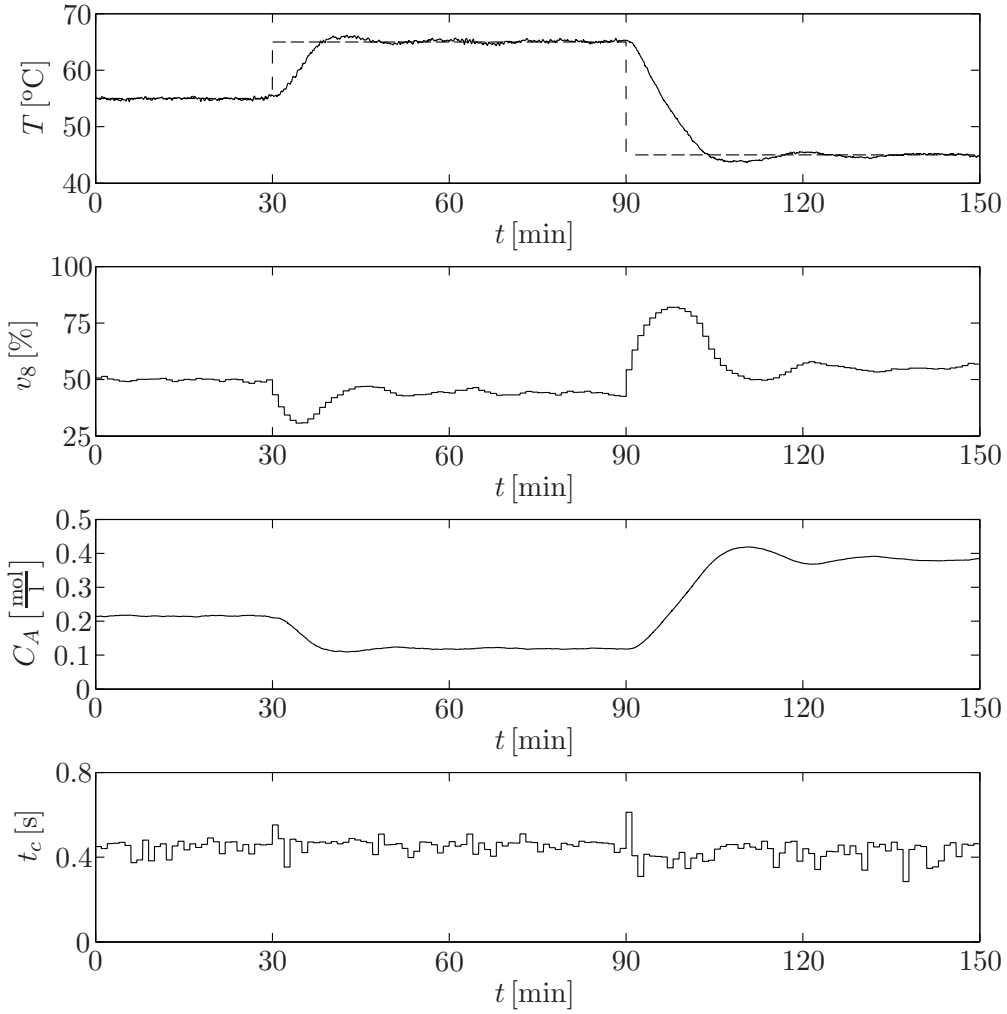


**Figure 7.2**: Disturbance rejection experiment (persistent disturbance in the emulated chemical reaction) controlled by the proposed MMMPC strategy. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and computation time $t_c$.

tested applying an additive disturbance in the feed $F_f$. Starting the experiment with the nominal value of the feed, a disturbance of $\Delta F_f = -0.02\,\mathrm{l/s}$, which corresponds to an error of $-40\,\%$, was introduced in $t = 60\,\mathrm{min}$ and held constant until the end of the experiment. After the appearance of the disturbance, the concentration $C_A$ decreases and, as a consequence, the temperature falls below the given setpoint and results in a maximum error of $-2.5\,^{\circ}\mathrm{C}$. The control strategy reacts rapidly and efficiently compensates the error in the system output in approximately 15 minutes. In spite of the magnitude of the introduced disturbance, the control strategy shows a good behavior without significant oscillations in the input signal and the temperature.

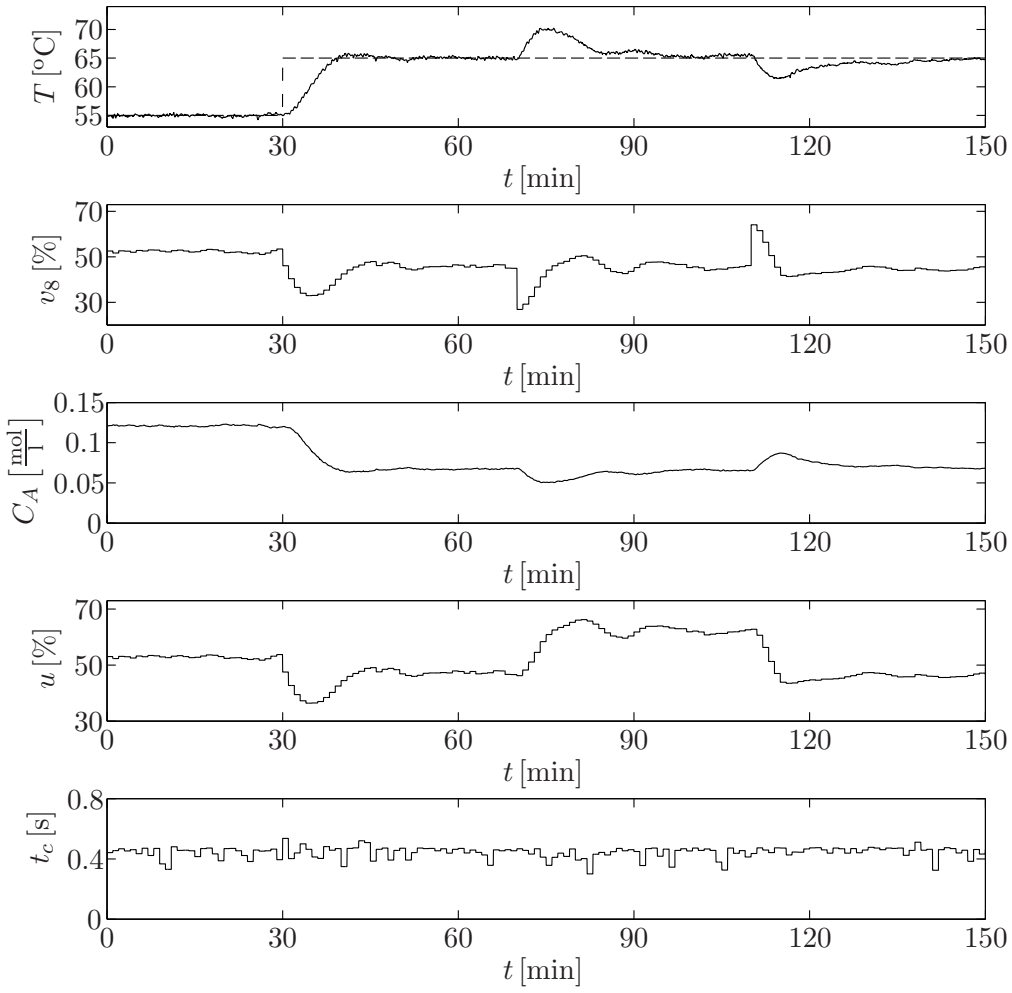


**Figure 7.3**: Disturbance rejection experiment (disturbance in the valve opening $v_8$) controlled by the proposed MMMPC strategy. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, input value $u$ calculated by the controller and computation time $t_c$.

The obtained results show that the proposed MMMPC strategy stabilizes the benchmark system in the given setpoint, even in presence of strong disturbances. The errors in the system output as a result of disturbances have been efficiently compensated without difficulties and underline the disturbance rejection capabilities of the proposed control strategy.

In the presented experiments, the control strategy needed an average time of $t_c^{avg} = 0.442\,\text{s}$ to compute the input signal, with a maximum of $t_c^{max} = 0.616\,\text{s}$ and a minimum of $t_c^{min} = 0.146\,\text{s}$. Hence, the optimization problem was solved within the used sampling time of $t_s = 60\,\text{s}$.

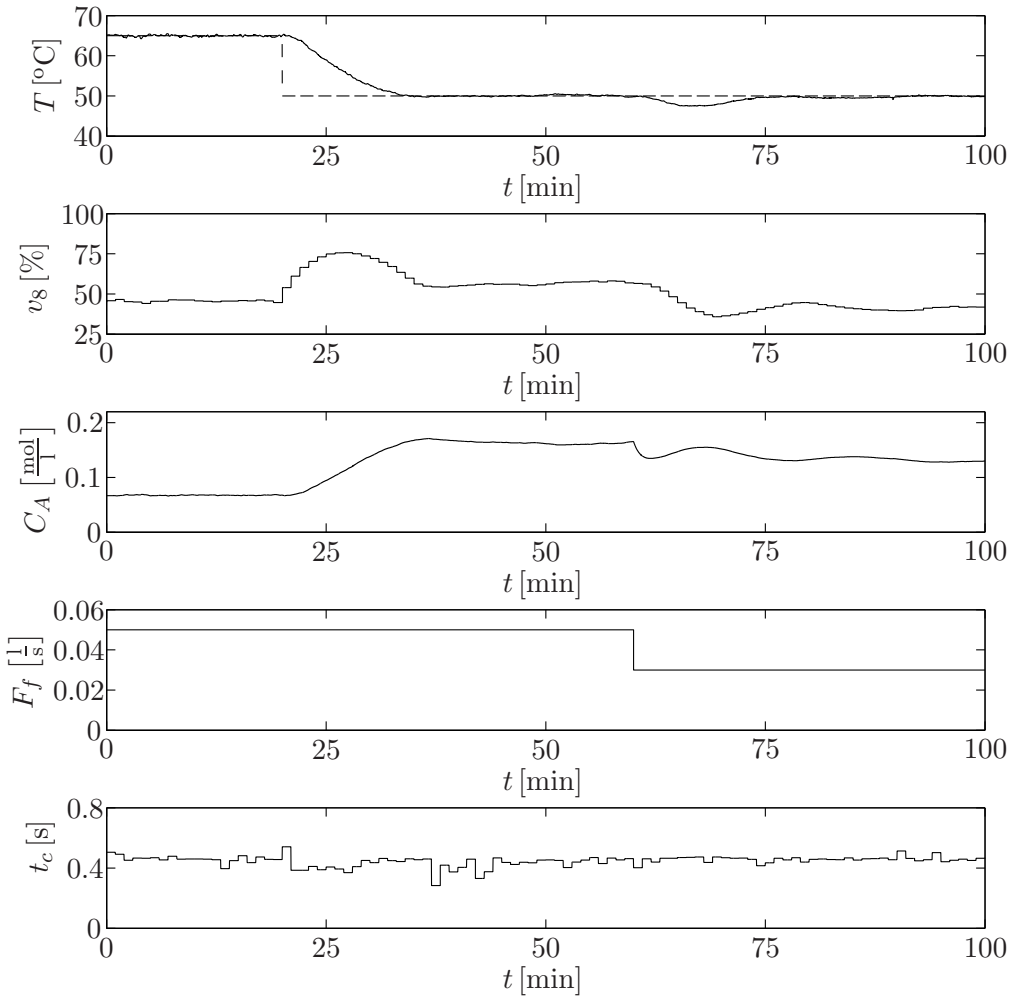


**Figure 7.4**: Disturbance rejection experiment (disturbance in the feed $F_f$) controlled by the proposed MMMPC strategy. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, feed $F_f$ and computation time $t_c$.

### 7.3.2 Volterra based MMMPC with exact worst case cost and guaranteed stability

The MMMPC strategy with guaranteed stability, proposed in Section 7.2, was validated in experiments with the pilot plant emulating an exothermic chemical reaction. Several experiments were carried out to test the setpoint tracking and disturbance rejection capabilities of the developed control strategy. The computation of the input sequence was implemented with sequential quadratic programming minimizing the worst case cost given by (7.19). In order to satisfy the necessary condition $N \geq N_t + N_u$ for guaranteed stability, a prediction horizon of $N = 80$ and a control horizon of $N_u = 15$ were used. Analogously to Section 7.3.1, the control effort was weighted with $\lambda = 5$

and a bound of $w = 5$ for the uncertainty was employed. Furthermore, the constraints:

$$\begin{array}{rcccl} 5 & \leq & u(k+i|k) & \leq & 100, \quad i = 0, \ldots, 14 \\ -20 & \leq & \Delta u(k+i|k) & \leq & 20, \quad i = 0, \ldots, 14 \\ u(k+14|k) & = & u_r(k) & & \end{array} \tag{7.78}$$

were considered in the computation of the input sequence. The constraints $\theta(k+i|k) \leq 1$ for $i = 1, \ldots, 80$ have been considered implicitly in the computation of the worst case cost.

In the first experiment carried out with the pilot plant, the setpoint tracking quality of the proposed MMMPC strategy with guaranteed stability was validated. During the experiment, see the results in Fig. 7.5, the setpoint was changed in $t = 30\,\text{min}$ from $55\,^{\circ}\text{C}$ to $65\,^{\circ}\text{C}$ and set in $t = 90\,\text{min}$ to $45\,^{\circ}\text{C}$. The setpoint changes lead to a very fast reaction of the control strategy which stabilizes the system in the given reference without any overshoot. After the stabilization of the system, neither the temperature nor the input signal show oscillations. In steady state, only small modifications in the input signal can be observed, necessary to maintain the system output in the setpoint. With respect to the results presented in Fig. 7.1, the higher prediction horizon leads to a slower setpoint tracking, but also eliminates the overshoot resulting from the step-wise change of the reference.

In the second experiment the disturbance rejection capability of the proposed control strategy was validated by means of an error in the parameter $E$ of the underlying exothermic chemical reaction. The introduced error, increasing the parameter $E$ by $3\,\%$ of the nominal value, was held constant during the entire experiment while two setpoint changes were applied to the system. After the application of the setpoint changes, the control strategy reacts rapidly and stabilizes the system in the given reference. In spite of the model mismatch (due to the error in the underlying chemical reaction model) only a small overshoot can be observed, approximately $0.6\,^{\circ}\text{C}$ and $-0.7\,^{\circ}\text{C}$ after the first and the second change, respectively. In comparison to the results presented in Fig. 7.2, the MMMPC strategy with guaranteed stability shows a slightly slower adoption of the new setpoint, but also reduces considerably the resulting overshoot.

In the third experiment (see Fig. 7.7) with the proposed control strategy an additive disturbance in the system input was applied to the system. The disturbance had a value of $\Delta v_8 = -15\,\%$ and was active in the interval from $t = 70\,\text{min}$ to $t = 110\,\text{min}$. Without the disturbance, the effective opening of the valve corresponds to the input signal computed by the control strategy, i.e. $v_8 = u$, whereas during the application of the disturbance the valve opening is given by $v_8 = u + \Delta v_8$. The application of the disturbance leads to a lower valve opening $v_8$ and results in an increasing temperature. With an increasing error in the system output, the control strategy gradually opens

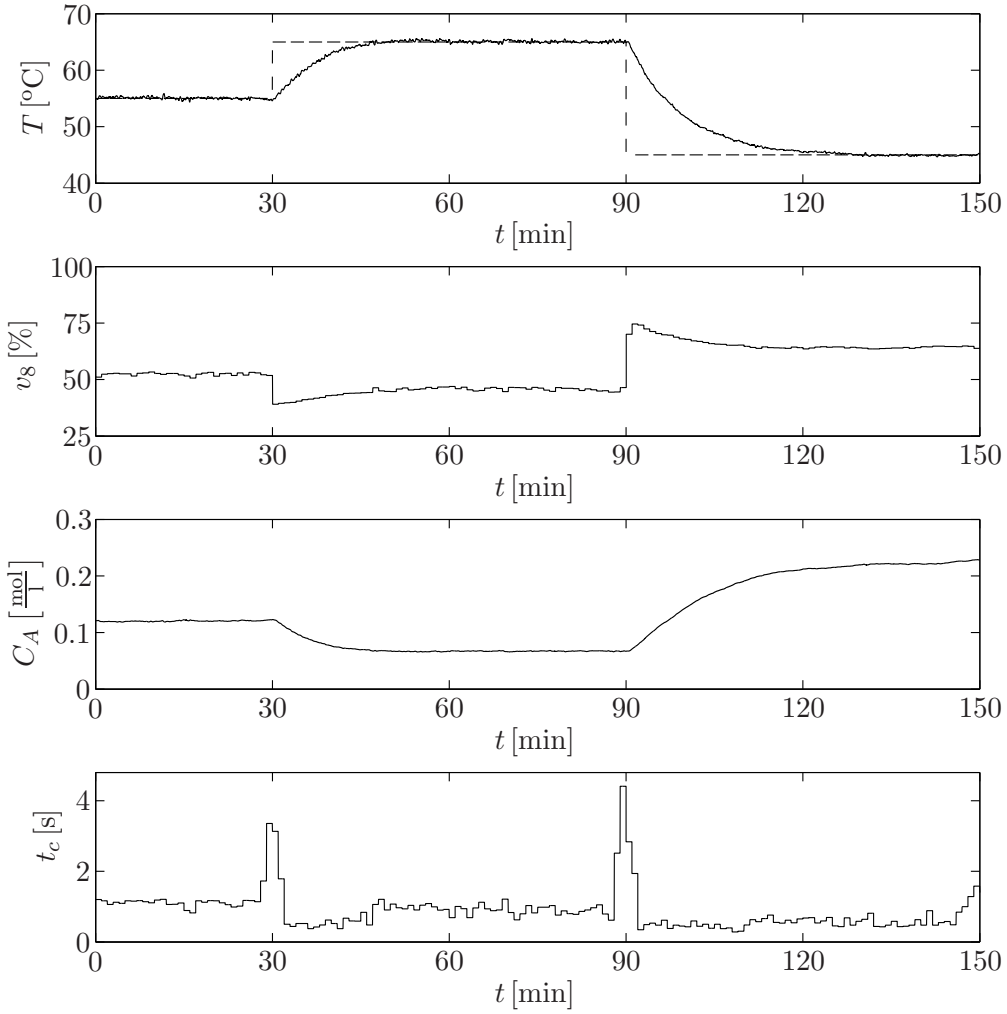


**Figure 7.5**: Setpoint tracking experiment controlled by the proposed MMMPC strategy with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and computation time $t_c$.

the valve and compensates the divergence. When the system reaches steady state in $t = 100\,\text{min}$, the effective valve opening corresponds to the one before the application of the disturbance. After the disappearance of the disturbance, the proposed control strategy reduces the input signal and stabilizes the system output in the given reference. The complete disturbance rejection underlines the robustness of the proposed control strategy.

In the last experiment (see Fig. 7.8) with the pilot plant emulating an exothermic chemical reaction, the proposed MMMPC strategy with guaranteed stability was validated by means of a disturbance in the feed $F_f$. The disturbance $\Delta F_f = -0.02\,\text{l/s}$, which corresponds to an error of $-40\,\%$ with respect to the nominal feed, was applied

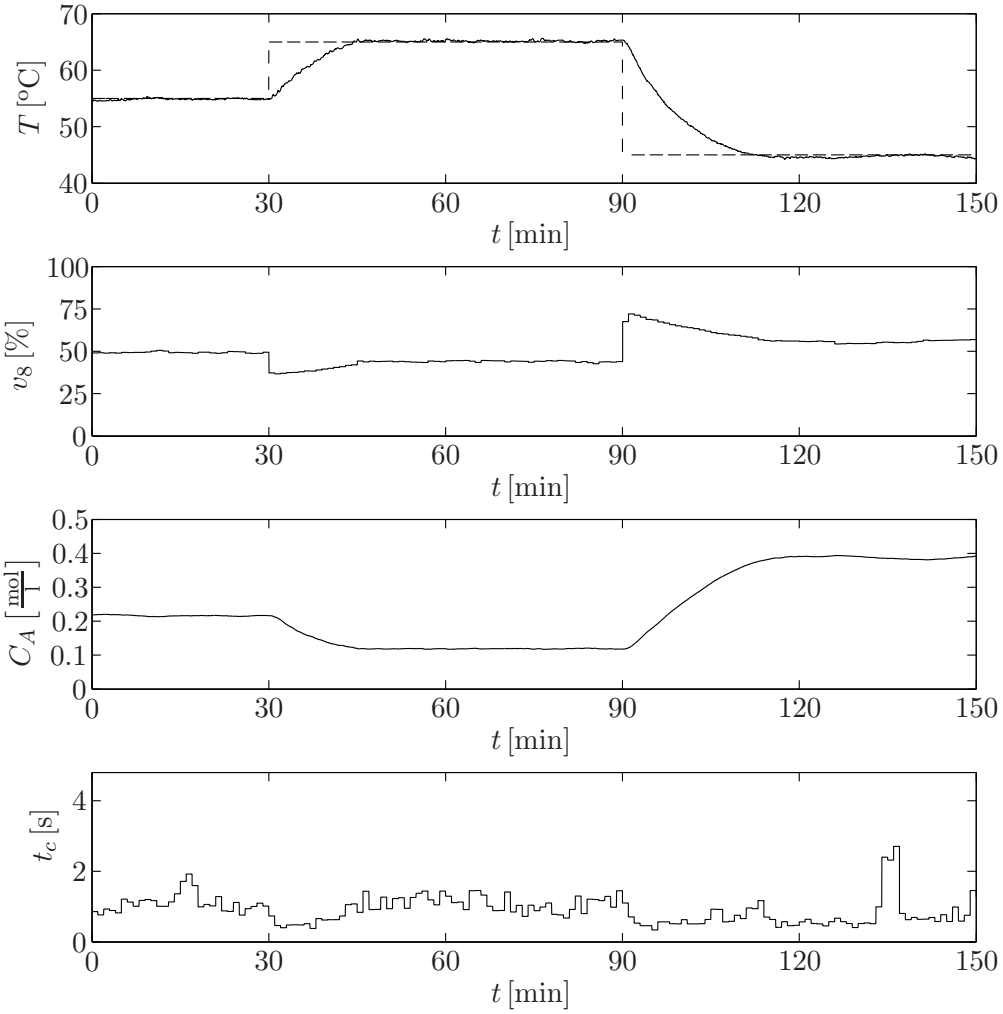


**Figure 7.6**: Disturbance rejection experiment (persistent disturbance in the emulated chemical reaction) controlled by the proposed MMMPC strategy with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$ and computation time $t_c$.

to the system in $t = 60\,\text{min}$ and held constant until the end of the experiment. The disturbance in the feed reduces the supply of the reactive and leads to a decreasing concentration $C_A$. As a consequence, the chemical reaction slows down and the temperature falls below the given setpoint. With an increasing error in the system output (with a maximum error of $-2.6\,^{\circ}\text{C}$), the control strategy reduces the valve opening and rejects the disturbance in approximately 15 minutes. After stabilizing the system output, the input signal shows small oscillations, justified by the magnitude of the applied disturbance.

The proposed MMMPC strategy with guaranteed stability was successfully applied

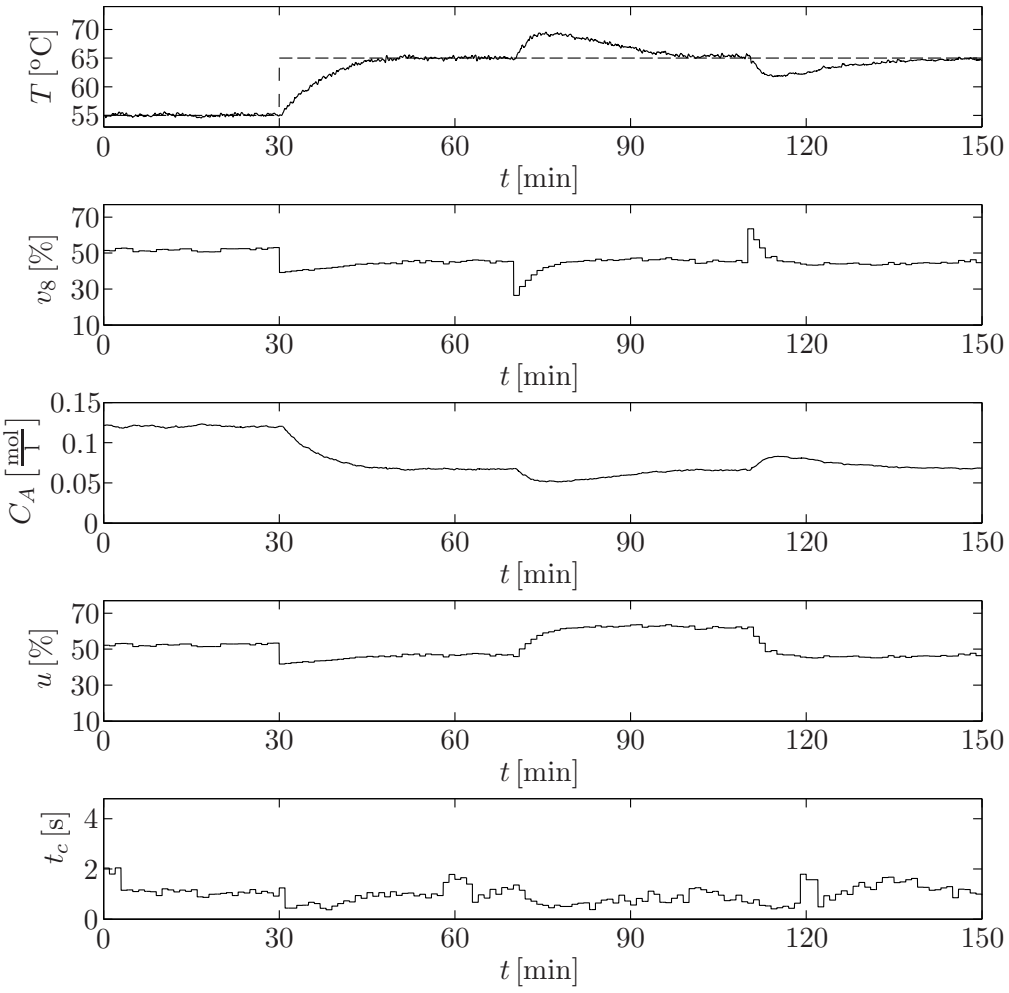


**Figure 7.7**: Disturbance rejection experiment (disturbance in the valve opening $v_8$) controlled by the proposed MMMPC strategy with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, input value $u$ calculated by the controller and computation time $t_c$.

to the pilot plant emulating an exothermic chemical reaction. The obtained experimental results showed that the control strategy stabilizes the system output in the setpoint and efficiently rejects uncertainties and disturbances. The compensation of output errors as a result of disturbances underlines the robustness of the min-max control strategy. Furthermore, it is worth mentioning that the computation of the input signal was always carried out within the used sampling time of $t_s = 60\,\mathrm{s}$. In the presented results, the optimization problem was solved in an average time of $t_c^{avg} = 0.925\,\mathrm{s}$, with a minimum of $t_c^{min} = 0.285\,\mathrm{s}$ and a maximum of $t_c^{max} = 4.408\,\mathrm{s}$. Note that the triplication of the prediction horizon in comparison to Section 7.3.1 results only in a duplication of the necessary average computation time.

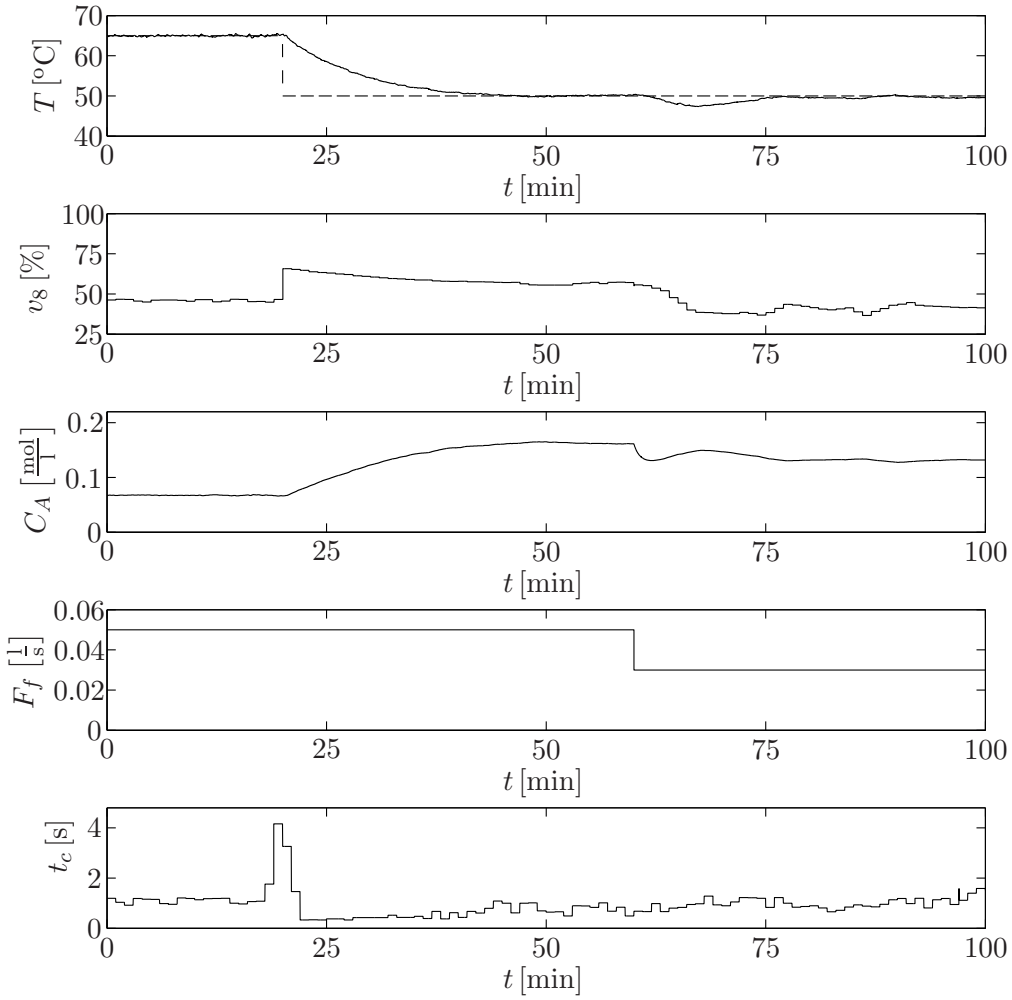


**Figure 7.8**: Disturbance rejection experiment (disturbance in the feed $F_f$) controlled by the proposed MMMPC strategy with guaranteed stability. From top to bottom: tank temperature $T$, aperture of the valve $v_8$, emulated concentration $C_A$, feed $F_f$ and computation time $t_c$.

## 7.4 Conclusions of the chapter

In this chapter two MMMPC strategies based on second order Volterra series models have been presented. The min-max approach has been used to take into account model uncertainties and to increase the robustness of the control strategies. The uncertainty has been considered as an additive term and the factor $w$, which corresponds to the upper bound of the uncertainty, can be easily obtained from experimental data.

The proposed control strategies calculate the input sequence by minimizing the worst case with respect to the model uncertainty. Usually, the computation of the worst case cost requires the evaluation of $2^N$ possible vertices of the uncertainty. For

the used second order Volterra series model it was shown that the exact worst case cost can be calculated easily under consideration of the non-autoregressive character of the model. As a consequence, the resulting optimization problem has practically the same computational complexity as a constrained MPC problem based on a second order Volterra series model. Note that the easy computation of the exact worst case cost is not limited to the used second order Volterra series models, but can be used for other non-autoregressive models. The reduced computational complexity of the optimization problem allows the implementation of the proposed min-max control strategies with larger prediction horizons than the ones used when evaluating all possible vertices of the uncertainty. Furthermore, for the control strategy presented in Section 7.2, input-to-state stability was proven for a prediction horizon satisfying $N \geq N_u + N_t$.

After the theoretical development, both control strategies were implemented and validated in experiments. With the used benchmark system, the pilot plant emulating an exothermic chemical reaction, several setpoint and disturbance rejection experiments were carried out. The proposed control strategies confirmed their robustness stabilizing the system even in presence of strong disturbances. With the explicit formulation of the worst case, the computation of the input sequence was carried out without difficulties within the used sampling time. Finally, the performance of the proposed control strategies was illustrated by means of experimental results.

# Chapter 8

# Conclusions

This thesis dealt with the development of novel MPC strategies both for nonlinear and uncertain systems. Particular attention has been paid to the practical applicability of the new control strategies in consideration of stabilizing issues. Discrete-time Volterra series models played an important role in the development of MPC strategies, but also discrete-time uncertain linear systems have been considered.

The unarguable success of linear MPC in industrial applications has several reasons, amongst the most important ones the intuitive formulation of the optimization problem, the consideration of constraints, the easy implementation of the resulting control law as well as the possibility to control processes in a wide range of different industrial areas. In contrast, NMPC and MMMPC are used seldom in practical applications and receive much less attention in industrial fields. The minor interest in these advanced MPC techniques has its origin in several factors, important ones being the difficulty to obtain suitable prediction models and the computational complexity to solve the optimization problem. Besides, the NLP algorithms used to solve the possibly non-convex optimization problem of NMPC are less reliable than the optimization algorithms used in linear MPC. Other more theoretical problems when dealing with advanced control techniques as NMPC and MMMPC include stability, robustness and optimality issues.

With the mentioned problems in mind, several robust and nonlinear MPC strategies, presented in the previous chapters, have been developed. The considered models to approximate the dynamics of a given system are easily obtainable from experimental input-output data and prevent the problems of a difficult and expensive identification. Two different model types have been considered in the development of control strategies: Volterra series models and linear models in state-space representation. Both mod-

| Algorithm | Stab. | Prog. | $t_c^{min}$ | $t_c^{max}$ | $t_c^{avg}$ |
|---|---|---|---|---|---|
| Iterative optim. | | QP | 0.021 s | 0.481 s | 0.059 s |
| Iterative optim. | yes | QP | 0.031 s | 0.284 s | 0.076 s |
| Convexification | | NLP | 0.066 s | 4.462 s | 0.736 s |
| Convexification | yes | NLP | 0.200 s | 7.945 s | 1.115 s |
| MMMPC (nonl. bound) | yes | NLP | 0.398 s | 3.276 s | 0.783 s |
| MMMPC (quad. bound) | yes | QP | 0.088 s | 0.179 s | 0.097 s |
| nonlin. MMMPC | | NLP | 0.146 s | 0.616 s | 0.442 s |
| nonlin. MMMPC | yes | NLP | 0.285 s | 4.408 s | 0.939 s |
| MPC | | QP | 0.021 s | 0.112 s | 0.031 s |
| MPC | yes | QP | 0.031 s | 0.160 s | 0.060 s |

**Table 8.1**: Comparison between the different control strategies with respect to the necessary solvers and the required computation times in the experiments with the pilot plant.

els can be identified with least squares estimation methods, a standard identification technique in control engineering and widely used in industry. For the development of robust MPC strategies, both models have been extended to include persistent bounded additive disturbances. The corresponding bound can be easily determined by means of a comparison of the measured system output and the predicted model output.

With respect to the numerical complexity to solve the optimization problem of MMMPC and NMPC strategies, different methods to reduce the computational burden have been presented. The proposed control strategies have been implemented and applied to at least one benchmark system. The control performance has been illustrated by experimental results, both from setpoint tracking and disturbance rejection experiments, and the practical applicability has been shown. Furthermore, it was shown that a modification in the cost function and a sufficiently large prediction horizon can ensure robust stability of the MPC strategies based on Volterra series models. Besides, it was proven that the linear MMMPC based on a nonlinear upper bound ultimately bounds the system state and, as a consequence, guarantees robust stability.

To give a small overview of the presented MPC strategies, the Tables 8.1 and 8.2 summarize the required times to compute a new input sequence (with average, minimum and maximum computation times) and the Sum of Square Errors (SSE) obtained in the different experiments with the pilot plant. Furthermore, the mentioned tables provide information about the type of programming problem and if the proposed strat-

egy guarantees closed-loop stability. For comparison issues, the numerical results of a linear MPC have been included in the overview (for detailed information on the applied MPC see Appendix A.9). The presented values show that the iterative optimization algorithm for NMPC based on Volterra series models (Chapter 4), the linear MMMPC based on a quadratic upper bound (Section 6.3) and the linear MPC (Appendix A.9), i.e. the optimization problems which represent a quadratic programming (QP) problem, require lower computation times to compute an input sequence and can be used to control processes with fast dynamics. The convexification approach for NMPC based on Volterra series models (Chapter 5), the linear MMMPC based on a nonlinear upper bound (Section 6.2) and the nonlinear MMMPC based on Volterra series models (Chapter 7) are Nonlinear Programming (NLP) problems with a computational complexity considerably superior to the one of the QP problems. Nevertheless, with modern NLP methods, e.g. SQP solvers, the optimization problems of the proposed MPC strategies are solvable in a reasonable time and are valid for processes with slow and intermediate fast dynamics. Obviously, the comparison of the control performance of the proposed control strategies is a difficult task[1] as some strategies worked better in one experiment and other strategies worked better in another experiment, e.g. the best SSE values in the setpoint tracking experiment with the nominal model are associated to the NMPC based on the convexification approach whereas in the experiment with a perturbation in the valve the nonlinear MMMPC strategy gave the best results.

Nevertheless, it can be seen clearly that the application of the MMMPC and NMPC strategies to the pilot plant results in all cases in lower SSE values than the linear MPC.

Summarizing the previous chapters, the main contributions of this thesis are the following key issues:

- As a basic necessity for the practical application of NMPC, several different processes with nonlinear dynamics have been approximated by second order Volterra series models. The complex dynamic behavior of processes is difficult to approximate by mathematical models, even with nonlinear models. Nevertheless, the nonlinear character of Volterra series models offers the possibility to improve the approximation quality with respect to the employment of linear models. Besides, one of the benchmark system has been modeled by an uncertain linear model with an additive persistent disturbance. The used model structures allow to determine the parameters with linear identification techniques and have been chosen with

[1] An analytical comparison of the different MPC strategies with respect to their control performance should be based on the results from computer simulations in order to avoid the influence of non-reproducible exogenous disturbances. Furthermore, differentiated performance indices should be used for a detailed analysis of the control performance with respect to setpoint tracking, disturbance rejection and stabilization issues.

| Algorithm | Stab. | nominal | error param. | pert. feed | pert. valve |
|---|---|---|---|---|---|
| Iterative optim. | | 2056.51 | 2098.15 | 1120.80 | 1088.31 |
| Iterative optim. | yes | 2140.14 | 2136.82 | 1447.52 | 1495.71 |
| Convexification | | 1934.44 | 1930.41 | 1119.80 | 1123.22 |
| Convexification | yes | 1660.83 | 1690.92 | 803.72 | 764.42 |
| MMMPC (nonl. bound) | yes | 2420.20 | 2301.37 | 1071.37 | 684.03 |
| MMMPC (quad. bound) | yes | 2463.15 | 2294.94 | 1129.73 | 677.65 |
| nonlin. MMMPC | | 2224.91 | 2208.78 | 1042.3 | 633.40 |
| nonlin. MMMPC | yes | 2316.90 | 2318.31 | 1020.65 | 695.73 |
| MPC | | 2675.26 | 2635.40 | 1549.22 | 1535.22 |
| MPC | yes | 2621.67 | 2517.47 | 1529.92 | 1767.12 |

**Table 8.2**: Comparison between the different control strategies with respect to the Sum of Square Errors (SSE) obtained in the presented experiments with the pilot plant.

the idea of a simple identification process.

- A computationally efficient unconstrained iterative optimization algorithm for NMPC based on second order Volterra series models without weighting of the control effort has been extended to include both linear constraints and a penalization of the control effort. Furthermore, it was shown that a minor change in the used cost function and a suitable prediction horizon ensures input-to-state stability of the constrained iterative optimization algorithm.

- In view of the possibly non-convex optimization problem resulting from a quadratic cost function in combination with second order Volterra series models, an NMPC based on a convexification approach has been developed. This novel approach approximates the original cost function by a series of convex quadratic functions. These approximations represent a convex hull of the original cost function and can be minimized globally. After a small modification in the used quadratic cost function it has been proven that the NMPC using the proposed convexification approach guarantees stability for sufficiently long prediction horizons.

- Robust stability of a linear MMMPC strategy has been proven. This control strategy uses a nonlinear upper bound of the worst case cost and reduces considerably the numerical complexity to solve the optimization problem. The reduction in the complexity is achieved by a diagonalization algorithm which allows a fast computation of an upper bound of the worst case cost. Then, instead of the evaluation of all possible vertices of the sequence of future uncertainties only a

matrix has to be diagonalized. The reduction of the computational complexity allows a faster computation of the input sequence and the use of longer prediction horizons.

- A novel MMMPC strategy based on uncertain second order Volterra series has been developed. The persistent bounded uncertainty has been considered in the prediction model as an additive term where the bound can be easily determined from a comparison of the system output and the predicted model output. It was shown that an explicit expression of the worst case cost can be obtained, i.e. the min-max optimization problem is reduced to a mere minimization problem. Finally, a small modification in the cost function allowed to prove robust stability of the proposed control strategy for a suitable prediction horizon.

- With respect to the practical applicability, the presented control strategies (with exception of the unconstrained iterative optimization algorithm which was used as a starting point for the development of new control strategies) have been implemented and validated in experiments with at least one of the benchmark systems. In the experiments it was observed that the different NMPC and MMMPC strategies had an improved control performance in comparison to linear MPC. The successful application of the different control strategies shown in this work joins the small number of NMPC and MMMPC applications reported in specialized literature.

Finally, some possible directions for future research include:

- The presented iterative optimization algorithms for NMPC based on second order Volterra series models are computationally very efficient, but suffer from convergence problems in the case of a non-convex optimization problem. Nevertheless, convergence of the iterative algorithm can be assured by a suitable choice of the weighting of the control effort or the use of a filter for the reference. A study of the necessary conditions to guarantee convergence would improve the iterative optimization algorithm.

- The presented convexification approach approximates the possibly non-convex cost by convex quadratic functions and allows the global minimization of the initial optimization problem. The simultaneous minimization of several convex quadratic functions represents a quadratic programming problem which can be solved by modern convex optimization algorithms as SQP methods. The use of Linear Matrix Inequalities (LMIs) could reduce the computational burden to solve the optimization problem allowing a faster computation of the input sequence.

- The MMMPC strategy based on uncertain second order Volterra series is based on an explicit formulation of the worst case, i.e. the min-max optimization problem degenerates to a minimization problem solvable by NLP algorithms. Here, the possibility to solve the minimization problem by the iterative optimization algorithm or the convexification approach could be studied.

- Non-autoregressive Volterra series models can be used to approximate a wide range of open-loop stable systems, but are not valid for unstable systems. Here, feedback controllers can be used to pre-stabilize the system enabling the possibility to approximate the extended system (process + feedback controller) by a non-autoregressive Volterra series model. The hierarchical control with an exterior control loop containing a control strategy based on a Volterra series model and an interior control loop with the stabilizing feedback controller could amplify the range of possible applications of Volterra series models in NMPC and MMMPC.

- The finite control horizon and the constraints considered in the optimization problem of an MPC strategy can result in unreachable states/output values for a given reference, especially after setpoint changes. In this case the feasibility of the optimization problem is not always ensured. In order to avoid the loss of feasibility the tracking problem can be considered in the design of MPC strategies which guarantee constraint satisfaction for any admissible setpoint.

- Different MMMPC strategies have been presented, both for linear models and second order Volterra series models, and successfully applied to a benchmark system. On a more theoretical basis, the increase in robustness of the proposed MPC strategies could be analyzed.

- Another possibility is the use of parametric uncertainties and state-dependent uncertainties in MMMPC based on Volterra series models. A possible future research line could consider these uncertainties in the development of new MMMPC strategies and analyze the closed-loop robustness, an important issue in industrial applications.

# Appendix A

# Mathematical definitions

## A.1 Prediction model

The future output of the second order Volterra series model (2.10) is calculated with the definitions of vectors and matrices taken from [77, 38] using the control horizon $N_u$, the prediction horizon $N$ and the truncation order $N_t$. It is assumed that the horizons and the truncation order satisfy the condition: $N_u \leq N \leq N_t$. With the following definitions, the prediction model (2.43)-(2.44) in matricial form can be used to calculate the predicted output of a second order Volterra series model.

The vector $\hat{\boldsymbol{y}} \in \mathbb{R}^N$ of the estimated future output at $k$ has the form:

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+N|k) \end{bmatrix} \tag{A.1}$$

The input sequence $\boldsymbol{u} \in \mathbb{R}^{N_u}$ computed at $k$ is defined as:

$$\boldsymbol{u} = \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N_u-1|k) \end{bmatrix} \tag{A.2}$$

The vector $\boldsymbol{u}_p \in \mathbb{R}^{N_t}$ contains the input signals applied to the system in the previous

sampling periods and can be written as:

$$\boldsymbol{u}_p = \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-N_t) \end{bmatrix} \tag{A.3}$$

The estimation error, defined as the difference between the measured output $y(k)$ in the current instant and the output estimation $\hat{y}(k|k-1)$ made in the previous sampling period, can be written as[1]:

$$d(k) = y(k) - \hat{y}(k|k-1) \tag{A.4}$$

As the estimation error can be calculated only for the current sampling period $k$ and not for every sampling period over the entire prediction horizon, the vector $\boldsymbol{d} \in \mathbb{R}^N$ is given by:

$$\boldsymbol{d} = \begin{bmatrix} d(k) \\ d(k) \\ \vdots \\ d(k) \end{bmatrix} \tag{A.5}$$

The matrix $G \in \mathbb{R}^{N \times N_u}$ represents the linear influence of the future control actions on the system output. Assuming $u(k+j|k) = u(k+N_u-1|k)\ \forall\ N_u \leq j \leq N-1$, i.e. a constant control signal after reaching the end of the control horizon[2], the matrix $G$ containing the linear term parameters of the model is defined as:

$$G = \begin{bmatrix} h_1(1) & 0 & \dots & 0 \\ h_1(2) & h_1(1) & \ddots & 0 \\ \vdots & \vdots & \ddots & h_1(1) \\ \vdots & \vdots & \ddots & h_1(1)+h_1(2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(N) & h_1(N-1) & \dots & \sum_{i=1}^{N-N_u+1} h_1(i) \end{bmatrix} \tag{A.6}$$

Analogously, the matrix $H \in \mathbb{R}^{N \times N_t}$, considering the influence of the past input values over the future system output, contains the linear term parameters. If the prediction

[1] The definition of the estimation error $d(k)$ corresponds to the one presented in (2.39). In order to define completely the prediction model (2.43)-(2.44), the equation is repeated in this section.

[2] As already mentioned in Section 2.3, it is not necessary define the future input signal beyond the prediction horizon, therefore the constant input signal is defined only for $N_u \leq j \leq N-1$.

horizon is shorter than the truncation order, i.e. $N < N_t$, the matrix $H$ is defined as:

$$H = \begin{bmatrix} h_1(2) & h_1(3) & \dots & \dots & \dots & h_1(N_t) & 0 \\ h_1(3) & h_1(4) & \dots & \dots & h_1(N_t) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_1(N) & \dots & \dots & h_1(N_t) & 0 & \dots & 0 \\ h_1(N+1) & \dots & h_1(N_t) & 0 & 0 & \dots & 0 \end{bmatrix}, \text{ for } N < N_t \quad \text{(A.7)}$$

For $N = N_t$ the matrix $H$ can be written in the following form:

$$H = \begin{bmatrix} h_1(2) & h_1(3) & \dots & \dots & h_1(N_t) & 0 \\ h_1(3) & h_1(4) & \dots & h_1(N_t) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_1(N_t-1) & h_1(N_t) & 0 & \dots & \dots & 0 \\ h_1(N_t) & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & \dots & 0 \end{bmatrix}, \text{ for } N = N_t \quad \text{(A.8)}$$

The function $\mathbf{f}(\boldsymbol{u}) : \mathbb{R}^{N_u} \mapsto \mathbb{R}^N$ maps the input sequence $\boldsymbol{u}$ to the term containing the future-future and future-past cross terms. As $\mathbf{f}(\boldsymbol{u})$ is used in (2.43) it will be used like a vector in $\mathbb{R}^N$ and is defined generally as:

$$\mathbf{f}(\boldsymbol{u}) = \begin{bmatrix} f(k+1) \\ f(k+2 \\ \vdots \\ f(k+N) \end{bmatrix} \quad \text{(A.9)}$$

with the elements

$$f(k+i) = \boldsymbol{f}_1(k+i)^{\mathrm{T}} \cdot B \cdot \boldsymbol{f}_2(k+i) \quad \text{(A.10)}$$

where the matrix $B$ is given in (A.18). The vectors $\boldsymbol{f}_1(k+i) \in \mathbb{R}^{N_t}$ and $\boldsymbol{f}_2(k+i) \in \mathbb{R}^{N_t}$ are defined for the case $i \leq N_u$ as:

$$\boldsymbol{f}_1(k+i) = \begin{bmatrix} u(k+i-1|k) \\ u(k+i-2|k) \\ \vdots \\ u(k|k) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad i = 1, \dots, N_u \quad \text{(A.11)}$$

$$\boldsymbol{f}_2(k+i) = \begin{bmatrix} u(k+i-1|k) \\ u(k+i-2|k) \\ \vdots \\ u(k|k) \\ u(k-1) \\ u(k-2) \\ \vdots \\ u(k-N_t+i) \end{bmatrix}, \quad i = 1, \ldots, N_u \tag{A.12}$$

and for the case $i > N_u$, under consideration of a a constant control signal after reaching the end of the control horizon, as:

$$\boldsymbol{f}_1(k+i) = \begin{bmatrix} u(k+N_u-1|k) \\ \vdots \\ u(k+N_u-1|k) \\ u(k+N_u-2|k) \\ \vdots \\ u(k|k) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad i = N_u+1, \ldots, N \tag{A.13}$$

$$\boldsymbol{f}_2(k+i) = \begin{bmatrix} u(k+N_u-1|k) \\ \vdots \\ u(k+N_u-1|k) \\ u(k+N_u-2|k) \\ \vdots \\ u(k|k) \\ u(k-1) \\ u(k-2) \\ \vdots \\ u(k-N_t+i) \end{bmatrix}, \quad i = N_u+1, \ldots, N \tag{A.14}$$

Analogously, the vector $\boldsymbol{g} \in \mathbb{R}^N$, containing the nonlinear terms depending exclusively

on the past input signals, can be written in a general way as[3]:

$$\boldsymbol{g} = \begin{bmatrix} g(k+1) \\ g(k+2 \\ \vdots \\ g(k+N) \end{bmatrix} \tag{A.15}$$

with the elements

$$g(k+i) = \boldsymbol{g}_1(k+i)^{\mathrm{T}} \cdot B \cdot \boldsymbol{g}_1(k+i) \tag{A.16}$$

The vectors $\boldsymbol{g}_1(k+i) \in \mathbb{R}^{N_t}$ are defined for $1 \leq i \leq N$ as:

$$\boldsymbol{g}_1(k+i) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u(k-1) \\ u(k-2) \\ \vdots \\ u(k-N_t+i) \end{bmatrix} \tag{A.17}$$

The matrix $B \in \mathbb{R}^{N_t \times N_t}$ used to calculate the vectors $\mathbf{f}(\boldsymbol{u})$ (A.9) and $\boldsymbol{g}$ (A.15) contains the second order term parameters of the Volterra series model. It can be written as:

$$B = \begin{bmatrix} h_2(1,1) & h_2(1,2) & h_2(1,3) & \dots & h_2(1,N_t) \\ 0 & h_2(2,2) & h_2(2,3) & \dots & h_2(2,N_t) \\ 0 & 0 & h_2(3,3) & \dots & h_2(3,N_t) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & h_2(N_t,N_t) \end{bmatrix} \tag{A.18}$$

The vector $\boldsymbol{h}_0 \in \mathbb{R}^N$ considering the offset $h_0$ of the second order Volterra series model in the prediction model (2.43)-(2.44) is defined as:

$$\boldsymbol{h}_0 = \begin{bmatrix} h_0 \\ h_0 \\ \vdots \\ h_0 \end{bmatrix} \tag{A.19}$$

Finally, with the definition of the vectors and matrices $\boldsymbol{u}$ (A.2), $\boldsymbol{u}_p$ (A.3), $\boldsymbol{d}$ (A.5), $G$ (A.6), $H$ (A.7)-(A.8), $\mathbf{f}(\boldsymbol{u})$ (A.9), $\boldsymbol{g}$ (A.15) and $\boldsymbol{h}_0$ (A.19) the model (2.43)-(2.44):

[3]Note that $\boldsymbol{g}$ is a function $\boldsymbol{g}(\boldsymbol{u}_p) : \mathbb{R}^{N_t} \mapsto \mathbb{R}^N$ mapping the vector $\boldsymbol{u}_p$ of past input values to the term containing the past-past cross terms. As $\boldsymbol{g}(\boldsymbol{u}_p)$ depends on the input values already applied, i.e. possesses a constant argument, the function $\boldsymbol{g}(\boldsymbol{u}_p)$ is denoted as a constant vector $\boldsymbol{g}$ oppressing the argument $\boldsymbol{u}_p$. Compare the definition of $\mathbf{f}(\boldsymbol{u})$ (A.9) where the argument has not been oppressed.

$$\hat{\boldsymbol{y}} = G\boldsymbol{u} + \boldsymbol{c} + \mathbf{f}(\boldsymbol{u}) \tag{A.20}$$
$$\boldsymbol{c} = H\boldsymbol{u}_p + \boldsymbol{g} + \boldsymbol{h}_0 + \boldsymbol{d} \tag{A.21}$$

can be used as an prediction model for systems represented by second order Volterra series models.

A common technique to reduce the number of parameters is the use of diagonal Volterra series models. For these models the second order term parameters satisfy the condition $h_2(i,j) = 0$ for $i \neq j$. In this case the calculation of $\mathbf{f}(\mathbf{u})$ and $\mathbf{g}$ can be simplified as no cross terms depending on $\boldsymbol{u}$ and $\boldsymbol{u}_p$ have to be considered. Therefore, the term $\mathbf{f}(\mathbf{u})$ considering only quadratic terms is defined as:

$$\mathbf{f}(\boldsymbol{u}) = G_{diag} \cdot \mathrm{diag}(\boldsymbol{u}) \cdot \boldsymbol{u} \tag{A.22}$$

and analogously the term $\boldsymbol{g}$ can be written as:

$$\boldsymbol{g} = H_{diag} \cdot \mathrm{diag}(\boldsymbol{u}_p) \cdot \boldsymbol{u}_p \tag{A.23}$$

The matrix $G_{diag} \in \mathbb{R}^{N \times N_u}$, containing the second order term parameters, is defined in a similar way as the matrix $G$ (A.6):

$$G_{diag} = \begin{bmatrix} h_2(1,1) & 0 & \dots & 0 \\ h_2(2,2) & h_2(1,1) & \ddots & 0 \\ \vdots & \vdots & \ddots & h_2(1,1) \\ \vdots & \vdots & \ddots & h_2(1,1) + h_2(2,2) \\ \vdots & \vdots & \ddots & \vdots \\ h_2(N,N) & h_2(N-1,N-1) & \dots & \sum_{i=1}^{N-N_u+1} h_2(i,i) \end{bmatrix} \tag{A.24}$$

The matrix $H_{diag} \in \mathbb{R}^{N \times N_t}$ is defined analogously to $H$ (A.7)-(A.8) for $N < N_t$ as:

$$H_{diag} = \begin{bmatrix} h_2(2,2) & h_2(3,3) & \dots & \dots & \dots & h_2(N_t,N_t) & 0 \\ h_2(3,3) & h_2(4,4) & \dots & \dots & h_2(N_t,N_t) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_2(N,N) & \dots & \dots & h_2(N_t,N_t) & 0 & \dots & 0 \\ h_2(N+1,N+1) & \dots & h_2(N_t,N_t) & 0 & 0 & \dots & 0 \end{bmatrix}$$
$$\text{for} \quad N < N_t \tag{A.25}$$

and in the case $N = N_t$ as:

$$H_2 = \begin{bmatrix} h_2(2,2) & h_2(3,3) & \dots & \dots & h_2(N_t,N_t) & 0 \\ h_2(3,3) & h_2(4,4) & \dots & h_2(N_t,N_t) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_2(N_t-1,N_t-1) & h_2(N_t,N_t) & 0 & \dots & \dots & 0 \\ h_2(N_t,N_t) & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & \dots & 0 \end{bmatrix}$$
$$\text{for} \quad N = N_t \tag{A.26}$$

## A.2 Objective function for MPC

This section defines the input increment sequence $\boldsymbol{\Delta u}$ and the reference trajectory $\boldsymbol{r}$ used in the objective function in Section 2.4. The matrices and vectors of the prediction model based on the second order Volterra series model have been defined in Appendix A.1.

The input increment sequence $\boldsymbol{\Delta u} \in \mathbb{R}^{N_u}$ represents the calculated control increments along the control horizon $N_u$ and is defined as:

$$\boldsymbol{\Delta u} = \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix} = \begin{bmatrix} u(k|k) - u(k-1) \\ u(k+1|k) - u(k|k) \\ \vdots \\ u(k+N_u-1) - u(k+N_u-2) \end{bmatrix} \tag{A.27}$$

The vector $\boldsymbol{r} \in \mathbb{R}^N$ denotes the reference trajectory along the prediction horizon $N$ and can be defined as:

$$\boldsymbol{r} = \begin{bmatrix} r(k+1) \\ r(k+2) \\ \vdots \\ r(k+N) \end{bmatrix} \tag{A.28}$$

## A.3 Matrix scaling

In the model (3.20) used to approximate the dynamics of the fuel cell a nonlinear scaling by the last measured value of the disturbance has been included. As a consequence of

this scaling the matrices $G$ and $H$ of the prediction model (2.43)-(2.44) and the matrix $B$ used to compute the vectors $\boldsymbol{f}$ (A.9) and $\boldsymbol{g}$ (A.15) have to be modified in order to contain the mentioned scaling.

The matrix $G \in \mathbb{R}^{N \times N_u}$, previously defined in (A.6), containing the scaling by the disturbance is defined as:

$$G = \frac{1}{w(k)} \begin{bmatrix} h_1(1) & 0 & \dots & 0 \\ h_1(2) & h_1(1) & \ddots & 0 \\ \vdots & \vdots & \ddots & h_1(1) \\ \vdots & \vdots & \ddots & h_1(1) + h_1(2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(N) & h_1(N-1) & \dots & \sum_{i=1}^{N-N_u+1} h_1(i) \end{bmatrix} \tag{A.29}$$

Analogously, the matrix $H \in \mathbb{R}^{N \times N_t}$, initially defined in (A.7) and (A.8), can be written for $N < N_t$ as:

$$H = \frac{1}{w(k)} \begin{bmatrix} h_1(2) & h_1(3) & \dots & \dots & \dots & h_1(N_t) & 0 \\ h_1(3) & h_1(4) & \dots & \dots & h_1(N_t) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_1(N) & \dots & \dots & h_1(N_t) & 0 & \dots & 0 \\ h_1(N+1) & \dots & h_1(N_t) & 0 & 0 & \dots & 0 \end{bmatrix} \tag{A.30}$$

for $N < N_t$

and for $N = N_t$ as:

$$H = \frac{1}{w(k)} \begin{bmatrix} h_1(2) & h_1(3) & \dots & \dots & h_1(N_t) & 0 \\ h_1(3) & h_1(4) & \dots & h_1(N_t) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_1(N_t-1) & h_1(N_t) & 0 & \dots & \dots & 0 \\ h_1(N_t) & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & \dots & 0 \end{bmatrix} \tag{A.31}$$

for $N = N_t$

In the same way, the matrix $B \in \mathbb{R}^{N_t \times N_t}$, defined in (A.18), is modified to include the scaling by the disturbance and is defined by:

$$B = \frac{1}{w(k)} \begin{bmatrix} h_2(1,1) & h_2(1,2) & h_2(1,3) & \dots & h_2(1,N_t) \\ 0 & h_2(2,2) & h_2(2,3) & \dots & h_2(2,N_t) \\ 0 & 0 & h_2(3,3) & \dots & h_2(3,N_t) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & h_2(N_t,N_t) \end{bmatrix} \tag{A.32}$$

## A.4 Stability of nominal MPC based on the iterative optimization

This section will proof the asymptotic stability for the constrained optimization presented in Section 4.4 based on the nominal prediction model, i.e. (2.43)-(2.44) without considering an estimation error. The initial optimization problem is already defined by (4.28)-(4.29). In a first step, the optimization problem will be rewritten in state-space representation and a feasible solution to the problem will be defined. Based on the feasible solution, asymptotic stability for the initial optimization problem can be shown.

### A.4.1 Optimization problem in state-space representation

For the stability proof of the proposed nonlinear predictive control strategy (4.28)-(4.29) based on the second order Volterra series model without considering an estimation error (2.60) the following general model predictive control optimization problem is considered:

$$\begin{aligned} \min_{\boldsymbol{u}(k)} \quad & J(\boldsymbol{x}(k), \boldsymbol{u}(k)) \\ \text{s.t. } & u(k+i|k) \in U, && i = 0, \ldots, N_u - 1 \\ & h(\boldsymbol{x}(k+i|k)) \in U, && i = N_u, \ldots, N-1 \\ & \boldsymbol{x}(k+i|k) \in X, && i = 0, \ldots, N-1 \\ & \boldsymbol{x}(k+N|k) \in \Omega \end{aligned} \tag{A.33}$$

being $J(\boldsymbol{x}(k), \boldsymbol{u}(k))$ the cost for a finite prediction horizon $N$ and finite control horizon $N_u$ with $N_u \leq N$. The initial state is denoted $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$ is the input sequence with the $N_u$ future input values. After reaching the end of the control horizon, the local control law $h(\boldsymbol{x}(k+i|k))$ for $i = N_u, \ldots, N-1$ is used to calculate the predictions $\boldsymbol{x}(k+i|k)$ for $i = N_u + 1, \ldots, N$. The successor state is generally defined by:

$$\begin{aligned} \boldsymbol{x}(k+i+1|k) &= \boldsymbol{\phi}(\boldsymbol{x}(k+i|k), u(k+i|k)), && i = 0, \ldots, N_u - 1 \\ \boldsymbol{x}(k+i+1|k) &= \boldsymbol{\phi}(\boldsymbol{x}(k+i|k), h(\boldsymbol{x}(k+i|k))), && i = N_u, \ldots, N \end{aligned} \tag{A.34}$$

and the sequence $\boldsymbol{u}(k)$ is calculated at $k$ minimizing (A.33) is given by:

$$\boldsymbol{u}(k) = \; [u(k|k), \quad u(k+1|k), \quad \ldots, \quad u(k+N_u-1|k)]^T \tag{A.35}$$

The cost function $J(\boldsymbol{x}(k), \boldsymbol{u}(k))$ for a prediction horizon $N$ and a control horizon $N_u$ can be written as:

$$J(\boldsymbol{x}(k), \boldsymbol{u}(k)) = \sum_{i=0}^{N_u-1} L(\boldsymbol{x}(k+i|k), u(k+i|k)) + \sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}(k+i|k)) \tag{A.36}$$

where $L(\cdot,\cdot)$ represents the stage cost along the control horizon and $L_h(\cdot)$ denotes the stage cost based on the local control law[4]:

$$\begin{aligned} L(\boldsymbol{x}(k+i|k), u(k+i|k)) &= \|\mathrm{f}(\boldsymbol{x}(k+i|k)) - r(k)\|_Q^2 + \\ &\quad \|u(k+i|k) - u_r(k)\|_R^2 \\ L_h(\boldsymbol{x}(k+i|k)) &= \|\mathrm{f}(\boldsymbol{x}(k+i|k)) - r(k)\|_Q^2 + \\ &\quad \|h(\boldsymbol{x}(k+i|k)) - u_r(k)\|_R^2 \end{aligned} \tag{A.37}$$

where $\mathrm{f} : \mathbb{R}^{N_t} \mapsto \mathbb{R}$ (2.57) maps the state vector $\boldsymbol{x}(k)$ to the output $y(k) = \mathrm{f}(\boldsymbol{x}(k))$ (2.60). Furthermore, $r(k)$ represents the reference of the system output and the variable $u_r(k)$ denotes the necessary steady state input for a given reference $r(k)$.

For the later proof of stability it is necessary that the output prediction for $k+N-1$ made at $k$ satisfies $y(k+N-1|k) = r(k)$. That means that the steady state input $u_r(k)$ has to be chosen so that the given condition will be satisfied. Hence, the steady state input[5] $u_r(k)$ is based on the predicted output $y(k+N-1|k)$:

$$\begin{aligned} u_r(k) &= \chi^{-1}(y(k+N-1|k)) \\ &= \chi^{-1}(r(k)) \end{aligned} \tag{A.38}$$

The steady state input (A.38) is used directly to define the local control law in the initial optimization problem (A.33):

$$h(\boldsymbol{x}(k+i|k)) = u_r(k), \quad i = N_u, \ldots, N-1 \tag{A.39}$$

### A.4.2 Feasibility of the solution

Consider the optimal input sequence

$$\boldsymbol{u}^*(k) = \; [u^*(k|k), \quad \ldots, \quad u^*(k+N_u-1|k)] \tag{A.40}$$

for the initial optimization problem (A.33). The optimal control sequence $\boldsymbol{u}^*(k)$ as well as the local control law, defined in (A.39) with the steady state input $u_r(k)$, lead to the optimal predicted states $\boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots, N$ and the associated optimal cost $J^*(\boldsymbol{x}(k))$. Furthermore, consider the solution $\boldsymbol{u}^f(k+1)$ for $k+1$ defined by:

$$\boldsymbol{u}^f(k+1) = \; [u^f(k+1|k+1), \quad u^f(k+2|k+1), \quad \ldots, \quad u^f(k+N_u|k+1)]^T \tag{A.41}$$

[4]For the sake of simplicity the notation $L_h(\boldsymbol{x}(k+i|k)) = L(\boldsymbol{x}(k+i|k), h(\boldsymbol{x}(k+i|k)))$ has been chosen.

[5]Note that the steady state input depends on the given reference. For the sake of simplicity, the notation $u_r(k) = u_r(r(k))$ has been chosen. Furthermore it has to be mentioned that the reference is constant, i.e. $r(k) = r(k+1)$.

with the elements $u^f(k+i|k+1)$ for $i = 1, \ldots, N_u$ given by:

$$u^f(k+i|k+1) = \begin{cases} u^*(k+i|k) & \text{for} \quad i = 1, \ldots, N_u - 1 \\ h(\boldsymbol{x}^f(k+N_u|k+1)) & \text{for} \quad i = N_u \end{cases} \tag{A.42}$$

Note that the input sequence $\boldsymbol{u}^f(k+1)$ corresponds to the shifted sequence $\boldsymbol{u}^*(k)$ plus the additional term of the local control law. With the shifted solution $\boldsymbol{u}^f(k+1)$ and the local control law $h(\boldsymbol{x}(k+i|k+1)) = u_r(k+1)$ for $i = N_u+1, \ldots, N$ the predicted states $\boldsymbol{x}^f(k+i|k+1)$ and the cost $J^f(\boldsymbol{x}(k+1))$ are obtained.

With the real system and the model being identical, the state prediction for $k+1$ made at $k$ satisfies $\boldsymbol{x}^*(k+1|k) = \boldsymbol{x}(k+1)$. Hence, with $u^f(k+i|k) = u^*(k+i|k)$ for $i = 1, \ldots N_u - 1$ the predictions based on the two input sequences $\boldsymbol{u}^*(k)$ and $\boldsymbol{u}^f(k+1)$ satisfy:

$$\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k), \quad i = 2, \ldots N_u \tag{A.43}$$

After reaching the end of the control horizon $N_u$, the local control law $h(\boldsymbol{x}^*(k+i|k)) = u_r(k)$ for $i = N_u, \ldots, N-1$ is used to predict the states $\boldsymbol{x}^*(k+i|k)$ for $i = N_u+1, \ldots, N$. Analogously, the states $\boldsymbol{x}^f(k+i|k+1)$ for $i = N_u+2, \ldots, N+1$ are predicted with the $h(\boldsymbol{x}^f(k+i|k+1)) = u_r(k+1)$ for $i = N_u+1, \ldots, N$. Furthermore, with $r(k) = r(k+1)$ the local control laws are identical, i.e. in both cases the steady state input $u_r(k) = u_r(k+1)$ is used to predict in $k$ and $k+1$ the future states. Hence, with $\boldsymbol{x}^f(N_u|k+1) = \boldsymbol{x}^*(N_u|k)$ and the same local control law the predicted states fulfill:

$$\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k), \quad i = N_u+1, \ldots N \tag{A.44}$$

Since the optimal solution $\boldsymbol{u}^*(k)$ and the local control law $h(\boldsymbol{x}^*(k+i|k))$ for $i = N_u, \ldots, N-1$ have been calculated considering the conditions given in (A.33), the predicted state satisfies $\boldsymbol{x}^*(k+N|k) \in \Omega$ being $\Omega$ an invariant set. With $\boldsymbol{x}^f(k+N|k+1) = \boldsymbol{x}^*(k+N|k)$ (A.44) the statement $\boldsymbol{x}^f(k+N|k+1) \in \Omega$ is true. As a consequence, the prediction for $k+N+1$ made at $k+1$ satisfies $\boldsymbol{x}^f(k+N+1|k+1) \in \Omega$ and, as a consequence, the solution $\boldsymbol{u}^f(k+1)$ to the optimization problem is feasible.

### A.4.3 Convergence

Consider the cost $J^*(\boldsymbol{x}(k), \boldsymbol{u}^*(k))$ at $k$ based on the optimal solution $\boldsymbol{u}^*(k)$ and the local control law $h(\boldsymbol{x}^*(k+i|k))$ for $i = N_u, \ldots, N-1$ as well as the cost $J^f(\boldsymbol{x}(k+1), \boldsymbol{u}^f(k+1))$ at $k+1$ calculated with the feasible solution $\boldsymbol{u}^f(k+1)$ and the local control law

$h(\boldsymbol{x}^f(k+i|k+1))$ for $i = N_u+1, \ldots, N$. The convergence of the proposed control strategy (A.33) can be guaranteed if the costs are monotonically decreasing.

The optimal cost $J^*(\boldsymbol{x}(k))$ at $k$ is defined with (A.36) as:

$$J^*(\boldsymbol{x}(k)) = \sum_{i=0}^{N_u-1} L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k)) + \sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}^*(k+i|k)) \tag{A.45}$$

and the cost $J^f(\boldsymbol{x}(k+1))$ at $k+1$ has the form:

$$\begin{aligned} J^f(\boldsymbol{x}(k+1)) \;=\; & \sum_{i=1}^{N_u} L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1)) + \\ & \sum_{i=N_u+1}^{N} L_h(\boldsymbol{x}^f(k+i|k+1)) \end{aligned} \tag{A.46}$$

With $u^f(k+N_u|k+1) = h(\boldsymbol{x}^f(k+N_u|k+1))$ the stage cost for $k+N_u$ based on the feasible solution can be expressed as $L(\boldsymbol{x}^f(k+N_u|k+1), u^f(k+N_u|k+1)) = L_h(\boldsymbol{x}^f(k+N_u|k+1))$. As a consequence, the difference $\Delta J(k+1) = J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k))$ of the costs can be written generally as:

$$\begin{aligned} \Delta J(k+1) \;=\; & -L(\boldsymbol{x}^*(k|k|k), u^*(k|k)) + L_h(\boldsymbol{x}^f(k+N|k+1)) \\ & + \sum_{i=1}^{N_u-1} \Big( L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1)) \\ & \qquad - L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k)) \Big) \\ & + \sum_{i=N_u}^{N-1} \Big( L_h(\boldsymbol{x}^f(k+i|k+1)) - L_h(\boldsymbol{x}^*(k+i|k+1)) \Big) \end{aligned} \tag{A.47}$$

With the definition (A.37) as well as $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots N$, $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \ldots N_u - 1$, $h(\boldsymbol{x}^f(k+i|k+1)) = h(\boldsymbol{x}^*(k+i|k))$ for $i = N_u, \ldots N-1$, $r(k+1) = r(k)$ and $u_r(k+1) = u_r(k)$ it can be shown that the stage costs $L(\cdot,\cdot)$ for $i = 1, \ldots N_u - 1$ and $L_h(\cdot)$ for $i = N_u, \ldots N-1$ satisfy:

$$\begin{aligned} L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1)) &= L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k)) \\ L_h(\boldsymbol{x}^f(k+i|k+1)) &= L_h(\boldsymbol{x}^*(k+i|k)) \end{aligned} \tag{A.48}$$

As a consequence, the difference of costs $\Delta J(k+1)$ (A.47) can be rewritten in the form:

$$\Delta J(k+1) = -L(\boldsymbol{x}^*(k|k), u^*(k|k)) + L_h(\boldsymbol{x}^f(k+N|k+1)) \tag{A.49}$$

where the first term is always non-positive as $L(\cdot,\cdot)$ has been defined as a quadratic function (A.37). With the nilpotent character of the state matrix $A$ (2.59) it can be

shown that[6] $L_h(\boldsymbol{x}^f(k+N|k+1)) = 0$ under certain circumstances. With $A^{N_t} = 0$ every input signal influences the system state and, as a consequence, the system output at most during $N_t$ sampling periods. Consider a prediction horizon of $N \geq N_u + N_t$ to be used in the initial optimization problem (A.33), i.e. with the local control law $h(\boldsymbol{x}^*(k+i|k+1)) = u_r(k+1)$ for $i = N_u + 1, \ldots, N$ the last $N_t$ control actions correspond to the steady state input $u_r(k+1)$. Hence, the prediction $\boldsymbol{x}^f(k+N|k+1)$ is calculated exclusively with the steady state input. Due to the nilpotent character of the state matrix ($A^{N_t} = 0$) the predicted state $\boldsymbol{x}^f(k+N|k+1)$ reaches steady. As a consequence the output prediction $y(k+N|k+1)$, defined in (2.60), also reaches steady state. Following the definition of the steady state input (A.38) the output prediction satisfies $y(k+N|k+1) = r(k+1)$. Therefore, with $y(k+N|k+1) = r(k+1)$ and $h(\boldsymbol{x}^f(k+N|k+1)) = u_r(k+1)$ the stage cost for $k+N$ at $k+1$ satisfies $L_h(\boldsymbol{x}^f(k+N|k+1)) = 0$.

Finally, with $N \geq N_u + N_t$, the difference of the costs (A.49) can be simplified to:

$$J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k)) = -L(\boldsymbol{x}(k), u^*(k|k)) \tag{A.50}$$

It is clear that for the optimal cost $J^*(\boldsymbol{x}(k+1))$ at $k+1$ and the suboptimal cost $J^f(\boldsymbol{x}(k+1))$ based on a feasible solution the relation $J^*(\boldsymbol{x}(k+1)) \leq J^f(\boldsymbol{x}(k+1))$ is true. With this relation the statement:

$$\begin{aligned} J^*(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k)) &\leq J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k)) \\ &= -L(\boldsymbol{x}(k), u^*(k|k)) \end{aligned} \tag{A.51}$$

holds. With $L(\boldsymbol{x}(k), u^*(k|k))$ (A.37) strictly positive definite (except in the origin) it can be shown that

$$J^*(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k)) < 0 \tag{A.52}$$

The strictly decreasing cost guarantees nominal asymptotic stability of the model predictive control strategy based on a second order Volterra series model.

Note that with a horizon of $N = N_u + N_t$ not only the output prediction $y(k+N|k+1) = r(k+1)$ reaches steady, but also $y(k+N+1|k+1) = r(k+1)$. The same applies to the output prediction for $k+N$ made at $k$, i.e. $y(k+N|k) = r(k)$. Therefore, without an output error in the output predictions for $k+N$ made at $k$ and for $k+N+1$ made at $k+1$ a terminal cost in the initial optimization problem is not required.

[6]The second term in (A.49) has to satisfy $L_h(\boldsymbol{x}^f(k+N|k+1)) \leq 0$. Being a quadratic function the stage cost $L_h(\cdot,\cdot)$ cannot adopt negative values. Therefore the condition $L_h(\boldsymbol{x}^f(k+N|k+1)) \leq 0$ has been substituted by $L_h(\boldsymbol{x}^f(k+N|k+1)) = 0$.

## A.5 Matrices used for convexification

The nonlinear term $\mathbf{f}(\boldsymbol{u})$ used in the prediction model (2.43) can be written as:

$$\mathbf{f}(\boldsymbol{u}) = \mathbf{f}_{nl}(\boldsymbol{u}) + \mathbf{f}_{l}(\boldsymbol{u}) \tag{A.53}$$

with $\mathbf{f}_{nl}(\boldsymbol{u}) : \mathbb{R}^{N_u} \mapsto \mathbb{R}^N$ being a nonlinear function depending on $\boldsymbol{u}$ and $\mathbf{f}_{l}(\boldsymbol{u}) : \mathbb{R}^{N_u} \mapsto \mathbb{R}^N$ representing a linear function in $\boldsymbol{u}$. These functions, used in (5.26) and (5.28), are based on the matrices $B_l$ and $B_{nl}$ which will be defined in this section. With the truncation order $N_t$, the prediction horizon $N$ and the control horizon $N_u$ the matrix $B_l \in \mathbb{R}^{N_t \times N_u \times N}$ or the submatrices $B_{l,i} \in \mathbb{R}^{N_t \times N_u}$ are defined for $i = 1, \ldots, N$ if $N < N_t$ and for $i = 1, \ldots, N-1$ if $N = N_t$ as:

$$B_{l,i} = \begin{bmatrix} h_2(i,i{+}1) & h_2(i{-}1,i{+}1) & \ldots & h_2(i{-}N_u{+}2,i{+}1) & \sum_{j=1}^{i-N_u+1} h_2(j,i{+}1) \\ h_2(i,i{+}2) & h_2(i{-}1,i{+}2) & \ldots & h_2(i{-}N_u{+}2,i{+}2) & \sum_{j=1}^{i-N_u+1} h_2(j,i{+}2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_2(i,N_t) & h_2(i{-}1,N_t) & \ldots & h_2(i{-}N_u{+}2,N_t) & \sum_{j=1}^{i-N_u+1} h_2(j,N_t) \\ 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.54}$$

For the case $i = N$ and $N = N_t$ the submatrix $B_{l,N}$ has the form:

$$B_{l,N} = \mathbf{0} \tag{A.55}$$

Analogously the matrix $B_{nl} \in \mathbb{R}^{N_u \times N_u \times N}$ or the submatrices $B_{nl,i} \in \mathbb{R}^{N_u \times N_u}$ with $i = 1, \ldots, N$ are defined as:

$$B_{nl,i} = \begin{bmatrix} h_2(i,i) & h_2(i{-}1,i) & \ldots & h_2(i{-}N_u{+}2,i) & \sum_{j=1}^{i-N_u+1} h_2(j,i) \\ 0 & h_2(i{-}1,i{-}1) & \ldots & h_2(i{-}N_u{+}2,i{-}1) & \sum_{j=1}^{i-N_u+1} h_2(j,i{-}1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & h_2(i{-}N_u{+}2,i{-}N_u{+}2) & \sum_{j=1}^{i-N_u+1} h_2(j,i{-}N_u{+}2) \\ 0 & 0 & \ldots & 0 & \sum_{k=1}^{i-N_u+1} \sum_{j=1}^{k} h_2(j,k) \end{bmatrix} \tag{A.56}$$

It has to be mentioned that in (A.54) and (A.56) the parameters $h_2(i,j)$ for $i \leq 0$ or $j \leq 0$ are defined as $h_2(i,j) = 0$.

## A.6 Stability of the nominal MPC based on the convexification approach

In this section the asymptotic stability of the control strategy based on a nominal second order Volterra series prediction model, i.e. (2.43)-(2.44) without considering an estimation error, and the convexification of the cost function will be shown. The considered optimization problem, already defined in Section 5.3, has to be transformed to its state-space representation. After the definition of a feasible solution to the optimization problem, asymptotic stability for the proposed control strategy can be proven.

### A.6.1 Optimization problem in state-space representation

The proposed control strategy based on the convexification approach (see Section 5.3) for a finite prediction horizon $N$ and a finite control horizon $N_u$ with $N \geq N_u$ considers the general cost function:

$$\begin{aligned} J(\boldsymbol{x}(k), \boldsymbol{u}(k), d(k), R^{(j)}) \;=\; & \sum_{i=0}^{N_u-1} L(\boldsymbol{x}(k+i|k), u(k+i|k), d(k), R^{(j)}) + \\ & \sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}(k+i|k), R^{(j)}) \end{aligned} \tag{A.57}$$

to be minimized. The vector $\boldsymbol{x}(k)$ represents the initial state vector and the sequence $\boldsymbol{u}(k)$ of $N_u$ future input values is defined by:

$$\boldsymbol{u}(k) = \; [u(k|k), \;\; u(k+1|k), \;\; \ldots, \;\; u(k+N_u-1|k)]^T \tag{A.58}$$

Furthermore, $d(k)$ denotes the estimation error in $k$ and the matrix $R^{(j)} \in \mathbb{R}^{N_u \times N_u}$ is the considered variable weighting of the control effort. The weighting matrix is generally defined as $R^{(j)} = \lambda^{(j)} I$ with $\lambda^{(j)}$ to be calculated by the optimization algorithm presented in Section (5.3). The terms $L(\cdot,\cdot,\cdot,\cdot)$ and $L_h(\cdot,\cdot)$ represent the stage costs based on the input sequence and a local control law, respectively. The optimization, based on the convexification algorithm, is subject to

$$\begin{aligned} & u(k+i|k) \in U, && i = 0, \ldots, N_u - 1 \\ & h(\boldsymbol{x}(k+i|k)) \in U, && i = N_u, \ldots, N-1 \\ & \boldsymbol{x}(k+i|k) \in X, && i = 0, \ldots, N-1 \\ & \boldsymbol{x}(k+N|k) \in \Omega \end{aligned} \tag{A.59}$$

where $h(\boldsymbol{x}(k+i|k))$ for $i = N_u, \ldots, N-1$ denotes a local control law used to predict the future states $\boldsymbol{x}(k+i|k)$ for $i = N_u+1, \ldots, N$. With $\boldsymbol{x}(k|k) = \boldsymbol{x}(k)$ being the initial state, the predictions based on the input sequence $\boldsymbol{u}(k)$ or the local control $h(\boldsymbol{x}(k+i|k))$ for $i = N_u, \ldots, N-1$ are given by:

$$\begin{array}{ll} \boldsymbol{x}(k+i+1|k) = \boldsymbol{\phi}(\boldsymbol{x}(k+i|k), u(k+i|k)), & i = 0, \ldots, N_u - 1 \\ \boldsymbol{x}(k+i+1|k) = \boldsymbol{\phi}(\boldsymbol{x}(k+i|k), h(\boldsymbol{x}(k+i|k))), & i = N_u, \ldots, N-1 \end{array} \tag{A.60}$$

The performance index (A.57) is based on the quadratic stage costs $L(\cdot,\cdot,\cdot,\cdot)$ and $L_h(\cdot,\cdot)$ given by:

$$\begin{array}{rcl} L(\boldsymbol{x}(k+i|k), u(k+i|k), R^{(j)}) & = & \|\mathrm{f}(\boldsymbol{x}(k+i|k)) - r(k)\|_Q^2 + \\ & & \|u(k+i|k) - u_r(k)\|_{R^{(j)}}^2 \\ L_h(\boldsymbol{x}(k+i|k), R^{(j)}) & = & \|\mathrm{f}(\boldsymbol{x}(k+i|k)) - r(k)\|_Q^2 + \\ & & \|h(\boldsymbol{x}(k+i|k)) - u_r(k)\|_{R^{(j)}}^2 \end{array} \tag{A.61}$$

where the nonlinear function $\mathrm{f} : \mathbb{R}^{N_t} \mapsto \mathbb{R}$, defined in (2.57), maps the state vector to the output $y(k) = \mathrm{f}(\boldsymbol{x}(k))$ (2.60) and $u_r(k)$ represents the steady-state input for the given reference $r(k)$.

In order to proof stability, the output prediction for $k+N-1$ made at $k$ has to satisfy $y(k+N-1|k) = r(k)$. Under the assumption that the prediction $y(k+N-1|k)$ reaches steady-state and satisfies de mentioned condition, the steady-state input is defined by:

$$\begin{array}{rcl} u_r(k) & = & \chi^{-1}(y(k+N-1|k)) \\ & = & \chi^{-1}(r(k)) \end{array} \tag{A.62}$$

where $\chi : \mathbb{R} \mapsto \mathbb{R}$ is a static nonlinearity which maps the steady-state input to the steady-state output. Finally, the steady state input (A.62) is used directly to define the local control law used in the model predictive control strategy subject to the conditions given in (A.59):

$$h(\boldsymbol{x}(k+i|k)) = u_r(k), \quad i = N_u, \ldots, N-1 \tag{A.63}$$

## A.6.2 Feasibility of the solution

Consider the input sequence:

$$\boldsymbol{u}^s(k) = \; [u^s(k|k), \;\; \ldots, \;\; u^s(k+N_u-1|k)] \tag{A.64}$$

being the optimal or suboptimal solution to the optimization problem calculated at $k$ with the iterative optimization algorithm (see Section 5.3). Furthermore, consider the sequence:

$$\boldsymbol{u}^f(k+1) = \; [u^f(k+1|k+1), \;\; u^f(k+2|k+1), \;\; \ldots, \;\; u^f(k+N_u|k+1)]^T \tag{A.65}$$

with the elements $u^f(k+i|k+1)$ for $i = 1, \ldots, N_u$ given by:

$$u^f(k+i|k+1) = \begin{cases} u^s(k+i|k) & \text{for} \quad i = 1, \ldots, N_u - 1 \\ h(\boldsymbol{x}^f(k+N_u|k+1)) & \text{for} \quad i = N_u \end{cases} \tag{A.66}$$

obtained by shifting the optimal or suboptimal solution $\boldsymbol{u}^s(k)$ by one element. Based on the solution $\boldsymbol{u}^s(k)$ and the local control law (A.63) the predictions $\boldsymbol{x}^s(k+i|k)$ for $i = 1, \ldots, N$ can be obtained. Analogously, the states $\boldsymbol{x}^f(k+i|k+1)$ for $i = 2, \ldots, N+1$ can be predicted with the shifted solution $\boldsymbol{u}^f(k+1)$ and the local control law $h(\boldsymbol{x}^f(k+i|k+1)) = u_r(k+1)$ for $i = N_u+1, \ldots, N$.

Consider the predictions $\boldsymbol{x}^s(k+i|k)$ for $i = 1, \ldots N_u - 1$ as well as the predictions $\boldsymbol{x}^f(k+i|k+1)$ for $i = 2, \ldots N_u$. Assuming a perfect model, i.e. the model and the system are identical, the prediction for $k+1$ made at $k$ satisfies $\boldsymbol{x}^s(k+1|k) = \boldsymbol{x}(k+1)$. Hence, with $\boldsymbol{x}^s(k+1|k) = \boldsymbol{x}^f(k+1|k+1)$ and identical inputs $u^f(k+i|k+1) = u^s(k+i|k)$ for $i = 1, \ldots N_u - 1$ the predictions based on the input sequences $\boldsymbol{u}^s(k)$ and $\boldsymbol{u}^f(k+1)$ satisfy:

$$\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^s(k+i|k), \quad i = 2, \ldots, N_u \tag{A.67}$$

Furthermore, with a constant reference $r(k) = r(k+1)$ the used steady state input values are identical, i.e. $u_r(k) = u_r(k+1)$. Hence, with $h(\boldsymbol{x}^s(k+i|k)) = u_r(k)$ and $h(\boldsymbol{x}^f(k+i|k+1)) = u_r(k+1)$ the predictions $\boldsymbol{x}^s(k+i|k)$ and $\boldsymbol{x}^f(k+i|k+1)$ with $i = N_u+1, \ldots, N$ are based on the same input values. As a consequence, the predictions satisfy:

$$\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^s(k+i|k), \quad i = N_u+1, \ldots, N \tag{A.68}$$

From the considered conditions (A.59) it is clear that the state prediction for $k+N$ made at $k$ satisfies $\boldsymbol{x}^s(k+N|k) \in \Omega$ being $\Omega$ an invariant set. From $\boldsymbol{x}^f(k+N|k+1) = \boldsymbol{x}^s(k+N|k)$ follows directly that $\boldsymbol{x}^f(k+N|k) \in \Omega$. With $h(\boldsymbol{x}^f(k+N|k+1))$ computed considering the conditions (A.59), the prediction for $k+N+1$ made at $k+1$ also satisfies $\boldsymbol{x}^f(k+N+1|k+1) \in \Omega$ and, as a consequence, the solution $\boldsymbol{u}^f(k+1)$ to the optimization problem is feasible.

### A.6.3 Convergence

Consider the cost $J_0^s(\boldsymbol{x}(k)) = J^s(\boldsymbol{x}(k), \boldsymbol{u}^s(k), R_0)$ computed with the initial state vector $\boldsymbol{x}(k)$, the optimal or suboptimal solution $\boldsymbol{u}^s(k)$ and the initial weighting factor $R_0$. Furthermore, consider the cost $J_0^f(\boldsymbol{x}(k+1)) = J^f(\boldsymbol{x}(k+1), \boldsymbol{u}^f(k+1), R_0)$ based on the initial state vector $\boldsymbol{x}(k+1)$, the feasible solution $\boldsymbol{u}^f(k)$ and the weighting parameter $R_0$. The predictive control strategy (5.3) based on the convexification approach and a

nominal second order Volterra series prediction model is stable if the costs are strict monotonically decreasing (except in the origin).

The cost $J_0^s(\boldsymbol{x}(k))$ at $k$ based on the optimal or suboptimal solution $\boldsymbol{u}^s(k)$ is given by (A.57) as:

$$J_0^s(\boldsymbol{x}(k)) = \sum_{i=0}^{N_u-1} L(\boldsymbol{x}^s(k+i|k), u^s(k+i|k), R_0) + \sum_{i=N_u}^{N-1} L_h(\boldsymbol{x}^s(k+i|k), R_0) \tag{A.69}$$

and the cost $J_0^f(\boldsymbol{x}(k+1))$ at $k+1$ as a result of the feasible input sequence $\boldsymbol{u}^f(k+1)$ has the form[7]:

$$\begin{aligned} J_0^f(\boldsymbol{x}(k+1)) \;=\; & \sum_{i=1}^{N_u-1} L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1), R_0) + \\ & \sum_{i=N_u}^{N} L_h(\boldsymbol{x}^f(k+i|k+1), R_0) \end{aligned} \tag{A.70}$$

With (A.69) and (A.70) the difference of the costs given by $\Delta J_0(k+1) = J_0^f(\boldsymbol{x}(k+1)) - J_0^s(\boldsymbol{x}(k))$ can be written generally as:

$$\begin{aligned} \Delta J_0(k+1) \;=\; & -L(\boldsymbol{x}^s(k|k), u^s(k|k), R_0) + L_h(\boldsymbol{x}^f(k+N|k+1), R_0) \\ & + \sum_{i=1}^{N_u-1} \Big( L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1), R_0) \\ & \qquad - L(\boldsymbol{x}^s(k+i|k), u^s(k+i|k), R_0) \Big) \\ & + \sum_{i=N_u}^{N-1} \Big( L_h(\boldsymbol{x}^f(k+i|k+1), R_0) - L_h(\boldsymbol{x}^s(k+i|k+1), R_0) \Big) \end{aligned} \tag{A.71}$$

For the nominal model the predictions satisfy $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^s(k+i|k)$ for $i = 1, \ldots N$ (see Section A.6.2) due to $u^f(k+i|k+1) = u^s(k+i|k)$ for $i = 1, \ldots N_u-1$ (A.66) and identical local control laws $h(\boldsymbol{x}^f(k+i|k+1)) = h(\boldsymbol{x}^s(k+i|k))$ for $i = N_u, \ldots N-1$ (as a consequence of a constant reference $r(k+1) = r(k)$). Hence, the stage costs in (A.71) satisfy:

$$\begin{aligned} L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1), R_0) &= L(\boldsymbol{x}^s(k+i|k), u^s(k+i|k), R_0) \\ L_h(\boldsymbol{x}^f(k+i|k+1), R_0) &= L_h(\boldsymbol{x}^s(k+i|k), R_0) \end{aligned} \tag{A.72}$$

for $i = 1, \ldots N_u - 1$ and $i = N_u, \ldots N-1$, respectively. With the identical stage costs, the difference of costs (A.71) can be simplified to:

$$\Delta J_0(k+1) = -L(\boldsymbol{x}^s(k|k), u^s(k|k), R_0) + L_h(\boldsymbol{x}^f(k+N|k+1), R_0) \tag{A.73}$$

[7]Note that $L(\boldsymbol{x}^f(k+N_u|k+1), u^f(k+N_u|k+1), R_0) = L_h(\boldsymbol{x}^f(k+N_u|k+1), R_0)$ as $u^f(k+N_u|k+1) = h(\boldsymbol{x}^f(k+N_u|k+1))$. Therefore the term $L(\boldsymbol{x}^f(k+N_u|k+1), u^f(k+N_u|k+1), R_0)$ has been considered in the second sum in (A.70).

In order to proof stability of the proposed predictive control strategy it has to be shown that (A.73) monotonically decreases.

Now consider a prediction horizon $N \geq N_t + N_u$ and the nilpotent character of the state matrix $A$ (2.59) of the used prediction model (2.60). From $A^{N_t} = 0$ follows that an input signal influences at most during $N_t$ sampling periods the states and the output. With the considered prediction horizon the last $N_t$ input values correspond in $k+1$ to the local control law, i.e. $h(\boldsymbol{x}^f(k+i|k+1)) = u_r(k+1)$ for $i = N_u, \ldots, N$. As a consequence the prediction $\boldsymbol{x}^f(k+N|k+1)$ is based only on the steady-state input $u_r(k+1)$. With $A^{N_t} = 0$ it is clear that both the predicted state $\boldsymbol{x}^f(k+N|k+1)$ and the output prediction $y(k+N|k+1)$ reach steady-state. Following the definition of the steady-state input (A.62) the output prediction satisfies $y(k+N|k+1) = r(k+1)$ and, as a consequence, the statement $\mathrm{f}(\boldsymbol{x}^f(k+N|k+1)) = r(k+1)$ holds. Hence, with $\mathrm{f}(\boldsymbol{x}^f(k+N|k+1)) = r(k+1)$ and $h(\boldsymbol{x}^f(k+N|k+1)) = u_r(k+1)$ the stage cost in (A.73) becomes $L_h(\boldsymbol{x}^f(k+N|k+1), R_0) = 0$. Then, the difference of costs (A.73) can be rewritten as:

$$J_0^f(\boldsymbol{x}(k+1)) - J_0^s(\boldsymbol{x}(k)) = -L(\boldsymbol{x}^s(k|k), u^s(k|k), R_0) \tag{A.74}$$

where the term $-L(\boldsymbol{x}^s(k|k), u^s(k|k), R_0)$ is always non-positive due to its quadratic character (A.61).

Consider the proposed iterative optimization (see Section 5.3) initialized in $k+1$ with the feasible solution $\boldsymbol{u}^{(0)}(k+1) = \boldsymbol{u}^f(k+1)$ and the initial weighting factor $R_0 = R^{(0)} = \lambda^{(0)}\boldsymbol{I}$ where $\lambda^{(0)} \geq 0$. The optimization algorithm computes in every iteration a new candidate input sequence $\boldsymbol{u}^{(i)}(k+1)$ and a new weighting matrix $R^{(j)} = \lambda^{(j)}\boldsymbol{I}$ with $\lambda^{(j)} \geq \lambda^{(j-1)}$. Based on the proof of the condition C2 (see Section 5.2.2) it can be shown that the cost $J^{(j)}(\boldsymbol{x}(k+1)) = J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), R^{(j)})$ satisfies:

$$J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), R^{(j)}) \leq J_0^f(\boldsymbol{x}(k+1)) \tag{A.75}$$

Then, with $\lambda^{(j)} \geq \lambda^{(j-1)}$ and, as a consequence, $R^{(j)} \geq R^{(i-1)}$ follows directly:

$$J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), R^{(0)}) \leq J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), R^{(j)}) \tag{A.76}$$

With (A.75) and (A.76) the statement:

$$J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), R^{(0)}) \leq J_0^f(\boldsymbol{x}(k+1)) \tag{A.77}$$

holds for all $\boldsymbol{u}^{(j)}(k+1)$. Finally, (A.77) can be used to rewrite (A.74) in the following form:

$$J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), R^{(0)}) - J_0^s(\boldsymbol{x}(k)) \leq -L(\boldsymbol{x}^s(k|k), u^s(k|k), R_0) \tag{A.78}$$

The strictly decreasing cost (except in the origin) guarantees nominal asymptotic stability of the model predictive control strategy based on the iterative control strategy using a convexification algorithm for second order Volterra series models. It has to be mentioned that the proposed control strategy is also asymptotic stable using the feasible solution based on the result given in (A.74). Hence, the optimization algorithm can be stopped in an arbitrary iteration without loss of the asymptotic stability. Nevertheless, the cost $J(\boldsymbol{x}(k+1), \boldsymbol{u}^{(j)}(k+1), R^{(0)})$ decreases monotonically in every iteration (see the proof of the condition C2 in Section 5.2.2) and leads to a higher optimality of the computed solution and a better convergence of the proposed control strategy.

With a horizon $N = N_u + N_t$ the output predictions reach steady-state with $y(k+N-1|k) = r(k)$ and $y(k+N|k+1) = r(k+1)$. As a direct consequence, the predictions also satisfy $y(k+N|k) = r(k)$ and $y(k+N+1|k+1) = r(k+1)$. Without an output error in the predictions $y(k+N|k)$ and $y(k+N+1|k+1)$ a terminal cost is not required in the initial cost function.

## A.7 Linear min-max MPC using input-output models

The use of a model in input-output representation as (6.81) instead of the state-space model (6.2) requires some minor changes in the implementation of the min-max control strategies presented in Chapter 6. In the case of a CARIMA (controlled autoregressive integrated moving average) model, the output is defined as:

$$A(q^{-1})y(k) = B(q^{-1})u(k) + C(q^{-1})\frac{\theta(k)}{\Delta} \tag{A.79}$$

being $q^{-1}$ the backward shift operator. Based on the CARIMA model (A.79) the evolution of the output can be written in short form as:

$$\boldsymbol{y} = G_u\boldsymbol{u} + G_\theta\boldsymbol{\theta} + G_x\boldsymbol{x} \tag{A.80}$$

where $G_u$, $G_\theta$ and $G_x$ are matrices which can be obtained from the system model and the parameters of the control strategy by means of different methods as solving the Diophantine equation [23]. Another more intuitive form is computing $G_u$ with the coefficients of the step impulse response of the system and to calculate the free response vector $G_x\boldsymbol{x}$ by means of the evolution of the system from the current state under the assumption of a constant input signal. On the other side, as the disturbance can be considered as an additional system input, the matrix $G_\theta$ can be computed with the same method as $G_u$ but assuming that $B(q^{-1}) = 1$. Then, the matrix $G_\theta$ can be calculated by solving the Diophantine equation or by using the free response of the

system to a step in $\theta(k)$. Note that in the case of the CARIMA model the vector $\boldsymbol{u}$ contains the future control increments and the state vector is composed of the current and past values of the output as well as the past values of the control increments:

$$\boldsymbol{x} = \left[ \; y(k), \;\; \ldots, \;\; y(k-n_a), \;\; \Delta u(k-1), \;\; \ldots, \;\; \Delta u(k-n_b) \; \right]^T \tag{A.81}$$

where $n_a$ and $n_b$ denotes the polynomial size of $A(\cdot)$ and $B(\cdot)$, respectively. With the prediction (A.80) and assuming that $Q$ is the identify matrix the quadratic cost function (6.5) can be expressed as[8]:

$$J(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) = (G_u \boldsymbol{u} + G_\theta \boldsymbol{\theta} + G_x \boldsymbol{x})^T (G_u \boldsymbol{u} + G_\theta \boldsymbol{\theta} + G_x \boldsymbol{x}) + \boldsymbol{u}^T R \boldsymbol{u} \tag{A.82}$$

Then, the cost function (A.82) can be expressed similar to (6.14) as:

$$\begin{aligned} J(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) \;\; = \;\; & \boldsymbol{u}^T M_{uu} \boldsymbol{u} + \boldsymbol{\theta}^T M_{\theta\theta} \boldsymbol{\theta} + 2\, \boldsymbol{\theta}^T M_{\theta u} \boldsymbol{u} + \\ & 2\, \boldsymbol{x}^T M_{uf}^T \boldsymbol{u} + 2\, \boldsymbol{x}^T M_{\theta f}^T \boldsymbol{\theta} + \boldsymbol{x}^T M_{ff} \boldsymbol{x} \end{aligned} \tag{A.83}$$

with $M_{uu} = G_u^T G_u + R$, $M_{\theta\theta} = G_\theta^T G_\theta$, $M_{\theta u} = G_\theta^T G_u$, $M_{uf} = G_u^T G_x$, $M_{\theta f} = G_\theta^T G_x$, $M_{ff} = G_x^T G_x$. Finally, the cost function (A.83) based on the general CARIMA model (A.79) can be used in the min-max model predictive control strategies presented in Chapter 6.

## A.8 Alternative approach for an explicit expression of the worst case cost

This section gives an alternative approach to the one shown in Section 6.2.2 to express the worst case cost explicitly. The approach presented in the following paragraphs is not limited to second order Volterra series models but can be used for a wide range of non-autoregressive models in input-output representation.

Consider the output prediction $\hat{\boldsymbol{y}} \in \mathbb{R}^N$ for a non-autoregressive model with uncertainty given by:

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}_n + W \boldsymbol{\theta} \tag{A.84}$$

where $\hat{\boldsymbol{y}}_n \in \mathbb{R}^N$ denotes the nominal prediction and $W \in \mathbb{R}^{N \times N}$ is a diagonal positive definite matrix given by $W = w\, I_N$ (7.7). The condition of $W$ being diagonal, positive

[8] In the given cost function (A.82) the terminal cost has not been included as this term has been disregarded in the implementation of the control strategies in Section 6.4. Furthermore, the experimental results presented in Section 6.4 have been obtained without the semi-feedback approach (6.3), i.e. the control law $u(k) = -Kx(k)$ has been used.

definite and with identical elements on the main diagonal is a direct consequence of the non-autoregressive character of the used model. The vector $\boldsymbol{\theta} \in \mathbb{R}^N$ represents the bounded future uncertainty sequence and the set of possible uncertainty trajectories is given by $\boldsymbol{\Theta} = \{\boldsymbol{\theta} \in \mathbb{R}^N : \|\boldsymbol{\theta}\|_\infty \leq 1\}$. Note that the possibility to separate (A.84) in a term denoting the nominal output and a term depending only on the disturbance vector is a result of the non-autoregressive character of the model used to generate the prediction model.

Consider the prediction model (A.84) in the general quadratic cost function without penalization of the control effort (2.46):

$$J(\boldsymbol{u}, \boldsymbol{\theta}) = (\hat{\boldsymbol{y}}_n + W\boldsymbol{\theta} - \boldsymbol{r})^T (\hat{\boldsymbol{y}}_n + W\boldsymbol{\theta} - \boldsymbol{r}) \tag{A.85}$$

where $\boldsymbol{r} \in \mathbb{R}^N$ denotes the reference trajectory. This cost function can be rewritten as:

$$J(\boldsymbol{u}, \boldsymbol{\theta}) = \hat{\boldsymbol{y}}_n^T \hat{\boldsymbol{y}}_n - 2\,\hat{\boldsymbol{y}}_n^T \boldsymbol{r} + \boldsymbol{r}\boldsymbol{r} + 2\,(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T W\boldsymbol{\theta} + \boldsymbol{\theta}^T W^T W\boldsymbol{\theta} \tag{A.86}$$

With (A.86) being a convex function in $\boldsymbol{\theta}$ ($W^T W$ is a positive definite diagonal matrix), the worst case cost with respect to the uncertainty can be found at least in one of the vertices of $\boldsymbol{\Theta}$. Hence, the maximum cost $J^*(\boldsymbol{u})$ is defined as shown in (7.12).

In order to obtain the worst case cost, the terms $2\,(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T W\boldsymbol{\theta}$ and $\boldsymbol{\theta}^T W^T W\boldsymbol{\theta}$ of (A.86) have to be maximized with respect to $\boldsymbol{\theta}$. For the term $\boldsymbol{\theta}^T W^T W\boldsymbol{\theta}$ it can easily be shown that:

$$\boldsymbol{\theta}^T W^T W\boldsymbol{\theta} = \text{trace}(W^T W) = w^2 N \tag{A.87}$$

is true and, as a consequence, the value of this term does not depend on the uncertainty $\boldsymbol{\theta}$ (or reaches its maximum for every $\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}$). Consider now the function $\sigma : \mathbb{R}^N \mapsto \mathbb{R}^N$ defined by:

$$\sigma(x) = \begin{cases} 1 & \text{if} \quad x \geq 0, \\ -1 & \text{if} \quad x < 0 \end{cases} \tag{A.88}$$

Then, using the signum-like function (A.88) in (A.87), the term $\boldsymbol{\theta}^T W^T W\boldsymbol{\theta}$ can be expressed as:

$$\boldsymbol{\theta}^T W^T W\boldsymbol{\theta} = w^2 N = w^2 \sigma(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T \sigma(\hat{\boldsymbol{y}}_n - \boldsymbol{r}) \tag{A.89}$$

With the term $\boldsymbol{\theta}^T W^T W\boldsymbol{\theta}$ reaching its maximum value independently from $\boldsymbol{\theta}$, the unitary hypercube $\boldsymbol{\theta}^*$ which maximizes the term $2\,(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T W\boldsymbol{\theta}$ also maximizes the cost (A.86). With the new function $\sigma$, the maximum value of the term $2\,(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T W\boldsymbol{\theta}$ can be written as[9]

$$\max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} 2\,(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T W\boldsymbol{\theta} = 2\,w(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T \sigma(\hat{\boldsymbol{y}}_n - \boldsymbol{r}) \tag{A.90}$$

[9]Note that a separate maximization of the terms $2\,(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T W\boldsymbol{\theta}$ and $\boldsymbol{\theta}^T W^T W\boldsymbol{\theta}$ is possible if and only if both terms reach their maximum for the same hypercube $\boldsymbol{\theta}$. For further details see Remark 7.1.

Finally, the worst case cost, i.e. the cost function (A.86) maximized with respect to the uncertainty $\boldsymbol{\theta}$, can be expressed with (A.89) and (A.90) as:

$$\begin{aligned} J^*(\boldsymbol{u}) &= \hat{\boldsymbol{y}}_n^T\hat{\boldsymbol{y}}_n - 2\hat{\boldsymbol{y}}_n^T\boldsymbol{r} + \boldsymbol{r}\boldsymbol{r} + 2\,w(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T\sigma(\hat{\boldsymbol{y}}_n - \boldsymbol{r}) + \\ &\quad w^2\sigma(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T\sigma(\hat{\boldsymbol{y}}_n - \boldsymbol{r}) \\ &= \left(\hat{\boldsymbol{y}}_n - \boldsymbol{r} + w\sigma(\hat{\boldsymbol{y}}_n - \boldsymbol{r})\right)^T\left(\hat{\boldsymbol{y}}_n - \boldsymbol{r} + w\sigma(\hat{\boldsymbol{y}}_n - \boldsymbol{r})\right) \end{aligned} \tag{A.91}$$

It is obvious that the worst case cost given in (A.91) can be evaluated easily.

The above defined function $\sigma$ (A.88) can easily be built by means of the sign function, e.g. $\sigma(x) = \operatorname{sgn}(2\operatorname{sgn}(x) + 1)$. Furthermore it has to be mentioned that the definition of the function $\sigma$ (A.88) with a return value $\sigma(0) = -1$ does not change the result of the maximization. Alternatively, (A.91) can be expressed directly with the sign function, i.e without building the function $\sigma$, as:

$$\begin{aligned} J^*(\boldsymbol{u}) &= \left(\hat{\boldsymbol{y}}_n - \boldsymbol{r} + w\operatorname{sgn}(\hat{\boldsymbol{y}}_n - \boldsymbol{r})\right)^T\left(\hat{\boldsymbol{y}}_n - \boldsymbol{r} + w\operatorname{sgn}(\hat{\boldsymbol{y}}_n - \boldsymbol{r})\right) \\ &\quad + w^2\left(N - \operatorname{sgn}(\hat{\boldsymbol{y}}_n - \boldsymbol{r})^T\operatorname{sgn}(\hat{\boldsymbol{y}}_n - \boldsymbol{r})\right) \end{aligned} \tag{A.92}$$

## A.9 Experimental results of a linear MPC

For comparison purposes, a linear MPC has been used to control the pilot plant emulating an exothermic chemical reaction. The used MPC is based on a finite impulse response model identified and validated with the experimental data presented in Fig. 3.7 and Fig. 3.9, respectively. The finite impulse response model was, analogously to the second order Volterra series model, identified with a sampling time of $t_m = 60\,\mathrm{s}$ and a truncation order of $N_t = 60$.

In a first step the MPC was implemented with a prediction horizon of $N = 25$, a control horizon of $N_u = 15$, a weighting factor of $\lambda = 5$ and the constraints given in (4.77). Here, the weighting factor was used to penalize the increments $\boldsymbol{\Delta u}$ in the input sequence. In a second step, the MPC was implemented with $N = 80$, $N_u = 15$ and $\lambda = 5$ for the prediction horizon, the control horizon and the weighting factor. In order to guarantee stability, the cost function was slightly modified to consider a weighting of the difference between the input sequence and the necessary steady-state input, i.e. $\boldsymbol{u} - \boldsymbol{u}_r$. Besides, the constraints given in (4.80) have been considered.

The MPC control strategies were applied to the pilot plant and several setpoint tracking and disturbance rejection experiments were carried out. The results in Fig. A.1 have been obtained in setpoint tracking experiments with the linear MPC implementations. The experiment was repeated (see Fig. A.2), but with an error of 3 %

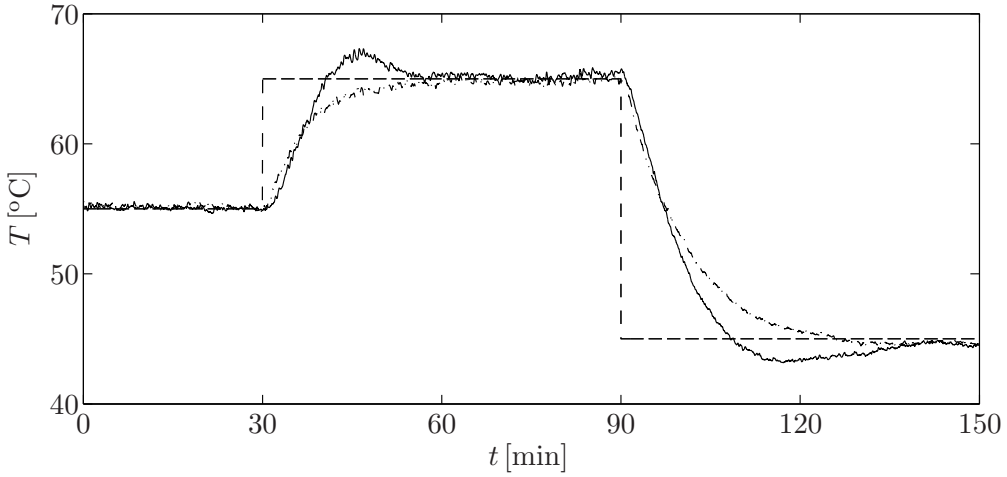


**Figure A.1**: Setpoint tracking experiment controlled by linear MPC strategies with a prediction horizon of $N = 25$ (solid line) and $N = 80$ (dash-dotted line).

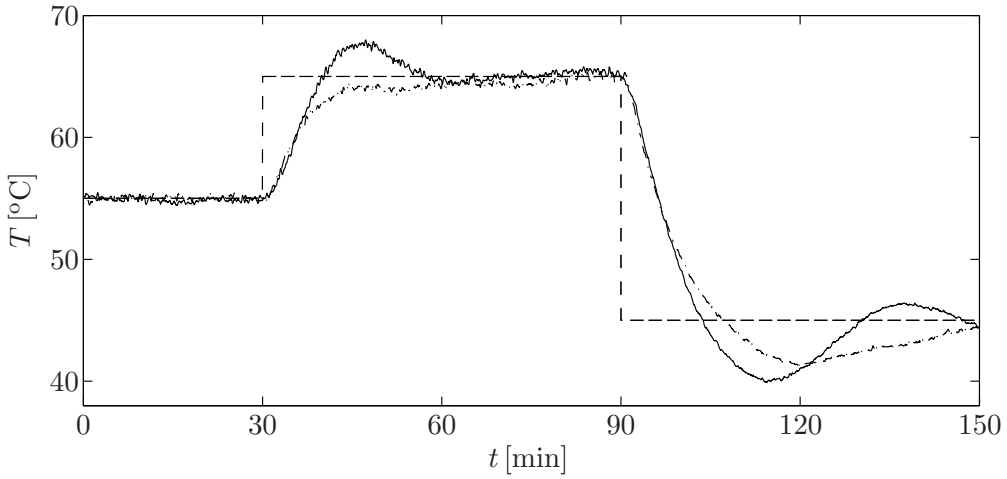


**Figure A.2**: Disturbance rejection experiment (persistent disturbance in the emulated chemical reaction) controlled by linear MPC strategies with a prediction horizon of $N = 25$ (solid line) and $N = 80$ (dash-dotted line).

in the parameter $E$ of the underlying model of the chemical reaction. The results of the third experiment, given in Fig. A.3, show the effect of an additive disturbance of $\Delta v_8 = -15\,\%$ in the system input, active in the period from $t = 70\,\text{min}$ to $t = 110\,\text{min}$. Finally, Fig. A.4 gives the results of a disturbance rejection experiment with a disturbance of $\Delta F_f = -0.02\,\text{l/s}$ in the feed $F_f$ applied in $t = 60\,\text{min}$.

As expected, the application of the linear MPC strategies to a process with nonlinear dynamics results in considerable errors in the system output with oscillations in the case of the MPC with a prediction horizon of $N = 25$ or a very slow error compensation

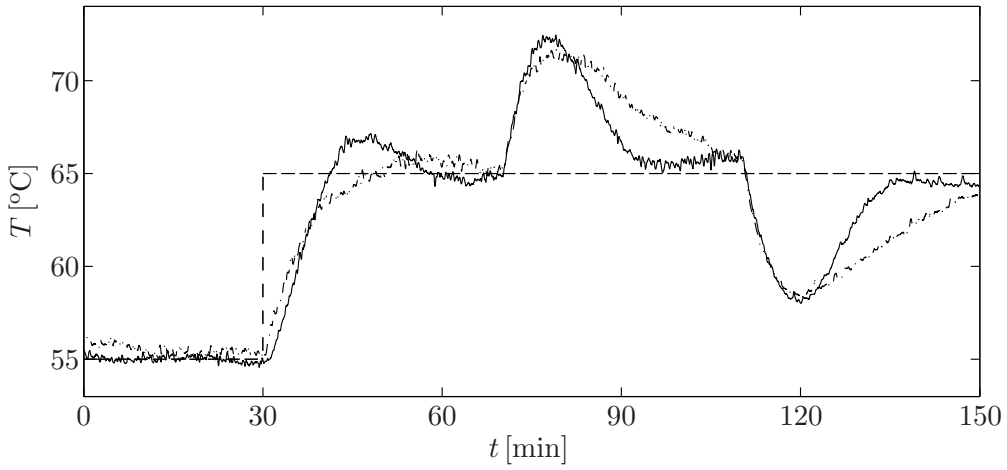


**Figure A.3**: Disturbance rejection experiment (disturbance in the valve opening $v_8$) controlled by linear MPC strategies with a prediction horizon of $N = 25$ (solid line) and $N = 80$ (dash-dotted line).

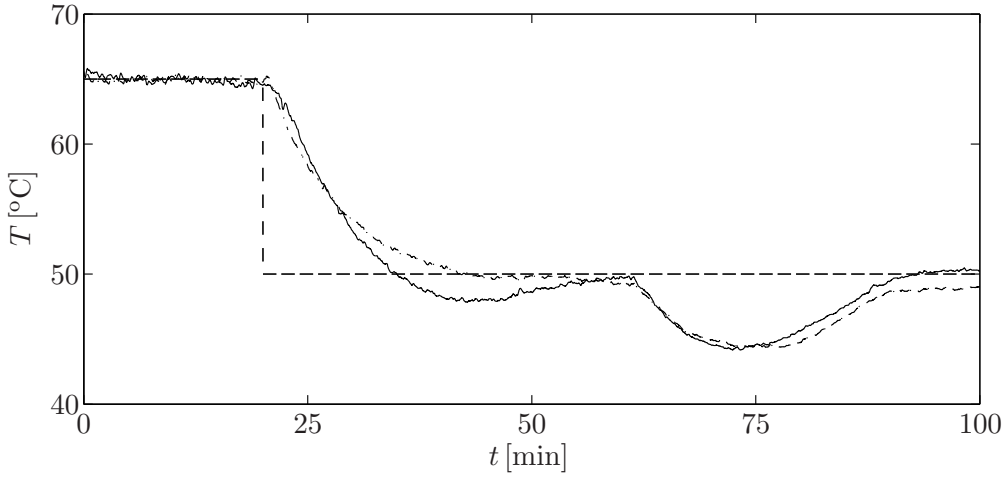


**Figure A.4**: Disturbance rejection experiment (disturbance in the feed $F_f$) controlled by linear MPC strategies with a prediction horizon of $N = 25$ (solid line) and $N = 80$ (dash-dotted line).

for the MPC with a prediction horizon of $N = 80$. Comparing the results shown in the previous chapters it is clear that the NMPC and MMMPC strategies proposed in this thesis have a better control performance for the used process. For the sake of completeness, the necessary computation times will be given. The optimization problem for a prediction horizon of $N = 25$ was solved in an average time of f $t_c^{avg} = 0.031\,\text{s}$, with a minimum of $t_c^{min} = 0.021\,\text{s}$ and a maximum of $t_c^{max} = 0.112\,\text{s}$. For a prediction horizon of $N = 80$, the computation of the input sequence required an average of $t_c^{avg} = 0.060\,\text{s}$, a minimum of $t_c^{min} = 0.031\,\text{s}$ and a maximum of $t_c^{max} = 0.160\,\text{s}$.

# Appendix B

# Spanish translations

This appendix presents the Spanish translations of the introduction (Chapter 1) and conclusions (Chapter 8) of this thesis.

## B.1 Capítulo 1: Introducción

### B.1.1 Motivación

Muchas técnicas de control moderno están basadas en modelos matemáticos que aproximan la evolución dinámica del sistema que se desea controlar. El controlador usa el modelo matemático para calcular la acción de control según cierto criterio. Por lo tanto, la calidad del modelo matemático usado para aproximar la dinámica del sistema tiene una influencia decisiva en el funcionamiento del controlador. En el control predictivo basado en modelo (MPC, del inglés *Model Predictive Control*) se realiza el cálculo de la acción de control a partir de la predicción que hace el modelo matemático de la evolución futura del sistema. Hoy en día, el MPC representa una de las técnicas de control avanzado más aplicadas a procesos industriales [23, 75]. Entre los factores que han contribuido al empleo frecuente del MPC destacan, entre otros, la formulación intuitiva del problema del control, la posibilidad de controlar una gran variedad de procesos, la consideración de restricciones en el cálculo de la señal de entrada y, en el caso del MPC basado en modelos lineales, la fácil implementación de la ley de control. Otra ventaja es la posibilidad de utilizar modelos matemáticos que se pueden obtener fácilmente del proceso considerado, p.ej. modelos de respuesta ante escalón en *Dynamic*

*Matrix Control* (DMC) [31] o modelos impulsionales en *Model Predictive Heuristic Control*[1] (MPHC) [111, 112]. A nivel industrial, diferentes estudios [104, 130, 13] indican que el MPC es una de las técnicas de control que se utiliza con mayor frecuencia.

Prácticamente todos los procesos dinámicos de importancia industrial exhiben cierto comportamiento no lineal [97]. Sin embargo, la gran mayoría de técnicas de control ha sido desarrollada para sistemas con dinámicas lineales. En el caso de procesos con no linealidades leves, tales técnicas de control habitualmente proporcionan buenos resultados. Además, muchas técnicas de control lineal muestran un buen rendimiento en torno a un punto operación dado. El predominio industrial de las técnicas de control lineal se entiende dada la facilidad de implementación de éstas y los resultados frecuentemente que ofrecen. Además, los modelos lineales se pueden obtener con métodos muy comunes en ingeniería de control, p.ej. respuestas ante escalón o impulso (que se pueden utilizar en DMC o MAC, respectivamente) de un sistema en cierto punto de operación o el uso de técnicas de identificación ampliamente conocidas para la obtención de funciones de transferencia con la ayuda de datos experimentales. Adicionalmente, existe una gran variedad de resultados teóricos para técnicas de control lineal incluyendo temas como estabilidad, optimalidad y robustez [23, 75, 85, 88].

Sin embargo, en muchos casos el rendimiento de un controlador lineal aplicado a un proceso dinámico no lineal es ineficiente. Esto se debe en muchas ocasiones a que el proceso real consta de no linealidades fuertes al variar el punto de operación del mismo. Con un modelo lineal determinado para cierto punto de operación, el modelo usado no es capaz de aproximar el proceso con suficiente calidad en un intervalo amplio de operación. Además, algunos procesos exhiben fuertes no linealidades incluso en la vecindad de un punto de operación y tienen un efecto negativo sobre el comportamiento del sistema en bucle cerrado. La omisión de la dinámica no lineal resulta con frecuencia en un rendimiento de control inaceptable y, en el peor de los casos, puede provocar una desestabilización del sistema controlado. También existen procesos que se encuentran continuamente en un modo transitorio como los procesos por lotes (*batch mode*), que nunca están en régimen estacionario o que experimentan modos de operación lejos de régimen estacionario durante algunos periodos, tales como el arranque o la parada. En los casos mencionados se deben tener en cuenta técnicas de control basadas en modelos no lineales con el fin de mejorar el rendimiento del sistema. Sin embargo, la consideración de utilizar modelos no lineales en el control de procesos plantea nuevas cuestiones Uno de los puntos más esenciales cuando se trata con modelos no lineales es la elección de una estructura de modelo apropiada. Esta elección no sólo define implícitamente la capacidad para aproximar las no linealidades del proceso, es decir, no

[1] El *Model Predictive Heuristic Control* es también conocido como *Matrix Algorithmic Control* (MAC).

todas las estructuras de modelo son convenientes para aproximar ciertas no linealidades, sino que también influye en un alto grado la técnica de control usada para calcular la acción de control.

Obviamente, una de las ventajas principales de los modelos no lineales es la posibilidad de considerar la dinámica no lineal de un proceso. Existe una gran variedad de modelos para aproximar hasta cierto grado los procesos con dinámica no lineal. Hay que subrayar que los modelos no lineales no representan un grupo homogéneo y no son tan fáciles de clasificar como modelos lineales. De hecho, la diferencia entre dos modelos no lineales puede ser mayor que la diferencia entre un modelo lineal y un modelo no lineal. Además, muchos de los modelos no lineales para control son más difíciles de obtener que los modelos obtenido a partir de la respuesta impulsional o ante escalón. Habitualmente la obtención de modelos no lineales se realiza mediante identificación con datos de entrada-salida o con primeros principios, considerando las leyes de conservación de energía y masa [23]. Otro problema de los modelos no lineales es la posible identificación de dinámicas no existentes y de ruido, como consecuencia de una sobre-parametrización o del elevado grado de libertad de la estructura de modelo no lineal elegida. A todo ello se suma la dificultad de comprensión e interpretación del significado físico de los modelos no lineales, que en muchos casos requiere un conocimiento profundo del proceso considerado.

El concepto básico de MPC ofrece la posibilidad de usar modelos no lineales para predecir la evolución futura del sistema. Por tanto, el MPC en combinación con modelos no lineales, conocido como control predictivo no lineal (NMPC, del inglés *Nonlinear Model Predictive Control*), puede ser empleado para calcular la acción de control. Aunque la idea principal del MPC no excluye la consideración de modelos no lineales, el uso de este tipo de modelos ocasiona diferentes problemas. Desde un punto de vista práctico, la consideración de modelos no lineales en una función de coste cuadrática resulta en un problema de optimización posiblemente no convexo con varios mínimos. Mientras que el MPC lineal requiere en cada muestreo la solución de un problema convexo, habitualmente realizado con Programación Cuadrática (QP, del inglés *Quadratic Programming*), NMPC necesita resolver (al menos parcialmente) un problema de optimización mediante Programación No Lineal (NLP, del inglés *Nonlinear Programming*) [17]. La dificultad de resolver el problema de optimización resulta en un incremento importante del tiempo de cálculo, limitando en muchos casos el uso de NMPC a procesos lentos o a horizontes de control cortos. Desde un punto de vista más teórico, la posible no convexidad del problema de optimización complica considerablemente el análisis de estabilidad o robustez. Además, los modelos no lineales pueden ser muy diversos y requieren frecuentemente un procedimiento específico para resolver el problema de optimización y para el análisis de aspectos teóricos. Debido a estos problemas, el uso de NMPC en la industria es muy escaso y se reduce a una pocas aplicaciones en áreas como

refino, química, polímeros, así como procesamiento de aire y gas [104]. El aumento de la carga computacional derivado de la complejidad de los problemas de optimización no convexos y de la posibilidad de mínimos múltiples, convierte la aplicación del NMPC en una tarea difícil con muchos aspectos no resueltos como el análisis de estabilidad, robustez y otros.

El cambio de modelos lineales a modelos no lineales permite la aproximación de procesos en un intervalo de operación más amplio y la consideración de un comportamiento dinámico más complejo. Incluso con modelos no lineales es difícil de reproducir las dinámicas complejas de procesos. Además, modelos lineales y no lineales ignoran la posible influencia de perturbaciones sobre la futura evolución del sistema. Aunque se puede comprobar la estabilidad de MPC o NMPC, la existencia de incertidumbres o perturbaciones puede desestabilizar el sistema en bucle abierto bajo ciertas condiciones o resultar en un rendimiento de control insuficiente [23, 118]. Para evitar este efecto no deseado, se pueden considerar incertidumbres y perturbaciones explícitamente en la formulación del modelo usado para predecir la futura evolución del sistema. Las incertidumbres paramétricas y las perturbaciones exógenas acotadas representan las formulaciones más comunes para tener en cuenta cierta incertidumbre en el modelado. Sin embargo, el uso de modelos con incertidumbre para la predicción no resulta en una única trayectoria, sino que genera un conjunto de posibles trayectorias. Un punto muy importante para considerar la incertidumbre de modelado en modelos lineales y no lineales es la elección de una realización de incertidumbre adecuada. La elección de la estructura para considerar una incertidumbre o perturbación tiene que asegurar que el modelo puede aproximar suficientemente bien la dinámica del proceso considerado. De no ser así, ninguna trayectoria del conjunto de predicciones corresponderá a la futura evolución del sistema.

Como en el caso de los modelos no lineales, el concepto de MPC permite usar una gran variedad de diferentes modelos inciertos para predecir la futura evolución del sistema considerado. No obstante, el cálculo de la acción de control mediante una estrategia de MPC basada en un modelo incierto no es trivial y trae consigo diversas dificultades computacionales y teóricas. En el caso de considerar incertidumbres acotadas, el conjunto de posibles trayectorias también es acotado. Esa cota representa el peor caso con respecto a la incertidumbre y se puede obtener un control más robusto calculando la acción de control mediante la minimización del coste asociado al peor caso. La minimización del coste del peor caso para calcular la secuencia de control es conocida como control predictivo mín-máx (MMMPC, del inglés *Min-Max Model Predictive Control*) [25]. El inconveniente principal de este enfoque es la carga computacional para calcular la secuencia de control con una complejidad creciendo exponencialmente en función del horizonte de predicción considerado. Como consecuencia directa, el número de aplicaciones de MMMPC es muy bajo a pesar de que hay evi-

dencia de que el enfoque mín-máx proporciona habitualmente mejores resultados que el MPC estándar en procesos con incertidumbre en la dinámica. La consideración de incertidumbres complica el estudio de temas teóricos como el de la estabilidad, siendo un campo de investigación abierto.

Esta tesis aborda el estudio y el desarrollo de nuevas técnicas de MPC basado en modelos de Volterra y en modelos inciertos, cubriendo así el marco general de la tesis NMPC y MMMPC. Se presta atención especial a la aplicabilidad de las técnicas de MPC con la idea de evitar o, por lo menos, reducir los problemas anteriormente mencionados. Aparte de este enfoque práctico con respecto a la aplicación de estrategias de MPC, en esta tesis el estudio de la estabilidad desempeña un papel muy importante en el diseño de nuevas técnicas de MPC. La consideración simultánea de aplicabilidad y estabilidad constituye un importante desafío, ya que en muchos casos las condiciones necesarias para asegurar estabilidad contrarrestan la aplicabilidad del controlador. En el caso de MMMPC, el enfoque principal es la reducción de la cota computacional asociada al problema de maximización, es decir, la determinación del peor caso para permitir el uso de horizontes de predicción más largos. En el diseño de técnicas de NMPC, la mayor dificultad estriba en encontrar la solución de un problema posiblemente no convexo.

Es evidente que la elección de una determinada estructura de modelo tiene un efecto decisivo en el posterior desarrollo de nuevas estrategias de MPC. En esta tesis, se utiliza principalmente como estructura de modelo un modelo no lineal de Volterra en tiempo discreto [132], con énfasis especial en los modelos de Volterra de segundo orden. La decisión de usar modelos de Volterra para la aproximación de sistemas dinámicos complejos se encuentra en la estructura particular de estos modelos, que representan la extensión natural de los modelos de convolución con la no linealidad considerada en un término aditivo. Esta estructura facilita la separación de los términos lineales y no lineales y puede ser explotada en el desarrollo de técnicas de MPC basado en modelos de Volterra. El segundo modelo, un modelo lineal en espacio de estados, es uno de los modelos más comunes utilizados en ingeniería de control. Aunque este modelo no tiene la capacidad de aproximar dinámicas no lineales, se pueden incluir explícitamente incertidumbres en la formulación del modelo para obtener un buen rendimiento del controlador para procesos no lineales y en presencia de perturbaciones. Se puede considerar el uso de dicho modelo como un paso necesario para ampliar el concepto de MMMPC a MPC basado en modelos de Volterra.

### B.1.2 Estado del arte

Para la aproximación de procesos no lineales se usan modelos de Volterra en muchos campos, entre otros en aplicaciones biomédicas [58, 78, 87, 10], acústica [128, 60], electrónica [110, 136] y control de procesos [38, 4, 35], pero especialmente en el procesamiento de señales para el diseño de filtros no lineales [94, 56]. Una lista bibliográfica extensa sobre la identificación de sistemas no lineales en el procesamiento de señales, incluidos los modelos de Volterra, se puede encontrar en [42]. El interés de usar modelos de Volterra para la aproximación de procesos no lineales en las diferentes áreas tiene varias razones, probablemente las más importantes son:

- Los modelos de Volterra, a pesar de representar modelos no lineales, son lineales en sus parámetros. Debido a ello se pueden estimar los parámetros de los modelos de Volterra mediante técnicas de identificación habitualmente usadas para modelos lineales, p.ej. métodos de mínimos cuadrados.
- Existe la posibilidad de estimar los parámetros de los modelos de Volterra a partir de datos experimentales de tipo entrada-salida. Por tanto, no hace falta un conocimiento profundo del proceso que se aproxima.
- Los modelos de Volterra pueden ser usados para modelar una gran variedad de dinámicas no lineales diferentes, como sistemas de fase no mínima.

Además de las ventajas mencionadas anteriormente, los modelos de Volterra tienen unas características de especial interés para el control de procesos:

- Representan la extensión lógica de los modelos de respuesta finita al impulso (FIR, del inglés *Finite Impulse Response*) que se han usado ampliamente en MPC lineal. Debido a ello, los modelos de Volterra muestran un comportamiento cualitativo muy similar a los modelos FIR, de los cuales son una generalización.
- Poseen un término adicional para considerar la no linealidad aproximada. Esta estructura especial permite la separación de los términos lineales y no lineales y puede ser explotada en el diseño de nuevas técnicas de control.

Uno de los principales inconvenientes de los modelos de Volterra es el elevado número de parámetros para describir la dinámica no lineal de un proceso. La identificación de un alto número de parámetros exige conjuntos de datos grandes que frecuentemente son costosos de obtener. Otro punto importante de los modelos de Volterra es la imposibilidad de aproximar procesos inestables, estando por tanto limitados a la aproximación de sistemas estables.

Las razones mencionadas convierten los modelos de Volterra en buenos candidatos para el desarrollo de nuevas técnicas de NMPC. La flexibilidad de este tipo de modelo permite el modelado de procesos dinámicos no lineales en una amplia gama de áreas sin la necesidad de una compresión profunda del sistema considerado [120, 38]. La estructura de dichos modelos es muy variable e incluye, bajo ciertas condiciones, la representación de modelos bilineales [70]. En [20] se ha demostrado que los modelos mencionados pueden aproximar arbitrariamente bien cualquier sistema estable con memoria evanescente. No obstante, una mejor aproximación requiere modelos de altos órdenes o de secuencias de memoria más largas. En la práctica es necesario encontrar un compromiso entre el número de parámetros y la calidad de aproximación de la dinámica del sistema.

Frecuentemente se identifican los parámetros de los modelos de Volterra mediante datos experimentales de tipo entrada-salida obtenidos del proceso que se desea aproximar. Para obtener una identificación apropiada se propone en [96] el uso de secuencias pseudoaleatorias multinivel (PRMS, del inglés *Pseudo-Random Multilevel Sequence*). Tales secuencias alteran el proceso considerado al objeto de obtener datos adecuados para la posterior estimación de los parámetros del modelo. En [101] se presentan secuencias de entrada que permiten la identificación de los parámetros de los términos lineales y no lineales por separado. Las secuencias propuestas están consideradas suaves ("plant-friendly"), ya que poseen un número reducido de transiciones. Otra manera de obtener datos apropiados para la estimación de parámetros de un modelo es el uso de señales multi-seno restringidas. Estas señales permiten la obtención de datos experimentales de tipo entrada-salida en un tiempo razonable mientras se satisfacen las restricciones impuestas por el usuario.

La identificación de modelos autorregresivos con un número reducido de parámetros ha sido propuesto en [52, 71]. A partir del modelo autorregresivo identificado se pueden calcular fácilmente los parámetros correspondientes de un modelo no autorregresivo de Volterra. Otro enfoque es la reducción de la complejidad paramétrica usando funciones con base ortonormal, como las funciones de Laguerre [24] o de Kautz [32] o funciones de base ortonormal generalizadas [124, 92]. En [70] se relaciona el rendimiento de control en bucle cerrado con el error de modelado en bucle abierto usando control de modelo no lineal interno (NIMC, del inglés *Nonlinear Internal Model Control*). En base a esa relación se propone un método de optimización que permite una reducción relevante de los modelos de Volterra para su uso en estrategias de control.

### B.1.2.1 NMPC basado en modelos de Volterra

Aunque se pueden usar los modelos de Volterra para aproximar una gran variedad de sistemas diferentes, se han publicado sólo unos pocos trabajos científicos sobre estrategias de NMPC basado en modelos de Volterra. Debido a que la investigación de NMPC basado en modelos de Volterra es muy escasa, sólo se han reportado un un pequeño número de aplicaciones prácticas.

Con respecto a la utilización de modelos de Volterra en NMPC, en [37, 77] propusieron un método computacionalmente eficiente para calcular la secuencia de futuras acciones de control. La optimización se realiza sin consideración de restricciones mediante un algoritmo iterativo basado en la separación de los términos lineales y no lineales de un modelo de Volterra de segundo orden. La estrategia de control consiste en un controlador lineal convencional extendido con un bucle auxiliar para incluir la dinámica no lineal en el proceso de optimización. La estrategia de control ha sido validada en simulaciones [37] con una reacción de polimerización en un reactor continuo de tanque agitado (CSTR, del inglés *Continuous Stirred Tank Reactor*) y una reacción isotérmica basada en la cinética de van de Vusse. La capacidad de la estrategia de NMPC propuesta para controlar sistemas de múltiple entrada y múltiple salida (MIMO, del inglés *Multiple-Input Multiple-Output*) se demostró en [77] con la simulación de un CSTR multivariable. Se puede considerar la estrategia de NMPC mencionada como el punto de partida de esta tesis para el desarrollo de técnicas de NMPC novedosas basadas en modelos de Volterra. El enfoque de optimización iterativo ha sido modificado en [51] para considerar modelos autorregresivos de Volterra en la predicción de la evolución futura del sistema. El NMPC modificado ha sido aplicado en simulaciones a un proceso de dos tanques con una entrada y una salida (SISO, del inglés *Single-Input Single-Output*) y se ha demostrado la viabilidad de la estrategia de control modificada.

En [50] se han presentado dos estrategias de NMPC subóptimas basadas en modelos de Volterra de segundo orden. La primera estrategia está basada en la suposición de incrementos constantes en acción de control sobre el horizonte de control entero, es decir la secuencia de entrada resultante tiene una pendiente constante. La segunda estrategia genera una secuencia de control donde los elementos de la secuencia tienen el mismo valor a lo largo del horizonte de control. En ambas estrategias de NMPC, dependiendo sólo de una única variable de decisión (dependiendo de la pendiente constante o de la señal de entrada constante), el problema de optimización posiblemente no convexo se convierte en un polinomio de orden cuatro. Se puede minimizar fácilmente este problema de optimización unidimensional, que no necesariamente es convexo, con respecto a la única variable de decisión. A continuación se evalúa la función del coste

para todos los valores de la variable de decisión asociados a mínimos no complejos y se elige el valor de la variable de decisión que corresponde al mínimo global de la función de coste. El problema de optimización reducido permite calcular analíticamente el valor de la variable de decisión que minimiza el coste y, en consecuencia, resulta en un cálculo de la secuencia de entrada muy rápido. Independientemente del método de optimización elegido, analítico o numérico, se pueden incluir restricciones de la entrada en el la minimización del coste.

En [36, 35] se extiende el NMPC propuesto en [37, 77] con la ponderación del esfuerzo de control en la función de coste. La inclusión de la ponderación del esfuerzo de control no cambia la complejidad computacional del problema de optimización y, en consecuencia, no cambia la eficiencia del enfoque iterativo. Además se ha estudiado en [35] cuestiones de convexidad del problema de optimización que resulta de la inclusión de la ponderación del esfuerzo de control. Se ha demostrado que se puede asegurar convexidad del problema de optimización mediante una elección apropiada del factor de ponderación y, por lo tanto, garantizar convergencia del enfoque iterativo. Además, [35] analiza la estrategia de NMPC para procesos con tiempo muerto cuando éste se considera implícitamente en el modelo de Volterra o se utiliza un compensador de tiempos muertos.

El NMPC para modelos de Volterra de segundo orden [37, 77] ha sido generalizado en [66] para su uso en combinación con modelos de Volterra de ordenes más altos. Análogamente al enfoque de optimización original, el problema de es resuelto por un algoritmo de optimización iterativo basado en la separación de los términos lineales y no lineales. A diferencia de la estrategia de NMPC original [37, 77] donde se considera sólo el termino de segundo orden del modelo de Volterra en el bucle auxiliar, [66] propone la inclusión de los términos de orden tres o más altos en el lazo auxiliar. La estrategia de control generalizada fue validada en simulación con un proceso de polimerización no lineal identificado mediante un modelo de Volterra de orden tres. La comparación del NMPC basado en un modelo de Volterra de orden tres con un MPC lineal demostró que el uso de modelos de Volterra de alto orden mejora el rendimiento de control en caso de procesos altamente no lineales.

Recientemente, [4] ha presentado la aplicación de una estrategia de NMPC basado en un modelo de Volterra de segundo orden a un modelo detallado que permite la simulación de un proceso de refino de petróleo crudo. El proceso considerado es un sistema subactuado (de tipo MIMO con 2 entradas y 4 salidas) y muestra un fuerte acoplamiento entre los diferentes estados. El NMPC aplicado no logra estabilizar los 4 estados en la referencia dada debido a la complejidad del proceso y la interconexión entre los estados. En un segundo paso se ha considerado un subsistema con 2 entradas

y 2 salidas para la validación del NMPC. El NMPC basado en un modelo de Volterra de segundo orden podía estabilizar los estados del subsistema en diferentes simulaciones de seguimiento de referencia y rechazo de perturbaciones. Un PI tradicional aplicado al mismo sistema obtuvo resultados inaceptables con oscilaciones importantes en las entradas y salidas por el acoplamiento entre los estados.

Con respecto a la estabilidad de las estrategias de NMPC es importante tener en cuenta que los modelos de Volterra representan una generalización de los modelos FIR y representan sistemas con olvido. Varios autores han demostrado la estabilidad en bucle cerrado para estrategias de MPC basado en modelos lineales de tipo FIR, entre otros [41, 135, 91]. En [85] se propone el uso de una restricción terminal de igualdad para garantizar la estabilidad de sistemas con memoria finita como es el caso en los modelos de Volterra. Otras publicaciones como [82, 86, 33] enfatizan que se puede demostrar estabilidad de estrategias de NMPC bajo ciertas condiciones mediante funciones de Lyapunov. A pesar de las proposiciones mencionadas el tema de estabilidad de NMPC basado en modelos de Volterra es un campo de investigación abierto con pocos resultados.

#### B.1.2.2 Control predictivo mín-máx

Los modelos no lineales, en comparación con los modelos lineales, permiten aproximar sistemas con dinámicas más complejas. Sin embargo, incluso con modelos matemáticos no lineales complejos es difícil de capturar por completo la dinámica de un proceso físico. La posible discrepancia entre el sistema y el modelo así como perturbaciones externas pueden resultar en una predicción deficiente de la evolución futura del sistema considerado. Con el fin de obtener un control más robusto se pueden usar modelos con incertidumbre en el marco de MMMPC, inicialmente propuesto por [134]. En general se dividen las estrategias de MMMPC en dos tipos, los controladores basados en la predicción en bucle abierto y los controladores basados en la predicción en bucle cerrado. El MMMPC en bucle abierto minimiza el peor caso sin realimentación de las predicciones calculadas [25, 5]. Además, todas las posibles trayectorias de la evolución del sistema deben cumplir las restricciones impuestas, resultando a menudo en un rendimiento de control considerablemente conservador. La complejidad del problema de optimización depende de forma exponencial del horizonte de predicción utilizado y representa un problema de tipo NP-duro [73]. El enfoque de bucle abierto ha sido una de las primeras estrategias de MPC basada en la formulación mín-máx del problema de optimización.

Con el fin de reducir el conservadurismo de la ley de control, [123] propuso el uso de

MMMPC basado en las predicciones en bucle cerrado. En el MMMPC en bucle cerrado se minimiza el problema de optimización considerando explícitamente la realimentación de la evolución predicha del sistema [85]. El enfoque en bucle cerrado resulta en un problema de optimización de dimensión infinita y es mucho más compleja que el problema de optimización correspondiente al MMMPC en bucle abierto. En general es muy difícil de calcular la secuencia de entrada y se pueden encontrar sólo unos pocos algoritmos para sistemas lineales [89, 43]. La carga computacional elevada limita el uso de MMMPC en bucle cerrado a procesos simples u horizontes de predicción pequeños. En esta tesis se considerará exclusivamente el concepto de MMMPC en bucle abierto.

Recientemente se ha demostrado en [14, 55] que la ley de control de MMMPC en bucle abierto es afín a trozos cuando se usa una función de coste basada en la norma 1. Análogamente, [106] probó afinidad a trozos para funciones de coste cuadráticas. Esta propiedad del problema de optimización permite la construcción de formas explícitas de la ley de control con una importante reducción de la complejidad [90]. Otro enfoque comúnmente utilizado está basado en una cota superior para el peor coste en lugar de calcular explícitamente el coste exacto. El problema de optimización resultante se puede resolver mediante desigualdades lineales matriciales (LMIs, del inglés *Linear Matrix Inequalities*). En el supuesto caso de usar una cota superior del peor caso la diferencia entre la solución exacta y la cota calculada aumenta el conservadurismo de la ley de control.

En [34] se demostró que el problema de optimización de MMMPC en bucle cerrado basado en costes y restricciones convexos puede ser representado por un problema de optimización convexo de dimensión finita. En el caso especial de un coste cuadrático se puede resolver el problema de optimización mediante programación cuadrática con restricciones cuadráticas (QCQP, del inglés *Quadratically Constrained Quadratic Program*). Además de las distintas estrategias de MMMPC, hay que mencionar las técnicas de MPC basado en tubos que permiten el cálculo de la señal de control mediante la solución de un QP estándar [83, 84, 69]. El concepto de los tubos se introdujo en [16] y está basado en la idea de una secuencia de conjuntos donde un conjunto es alcanzable desde el conjunto anterior. Las estrategias de MPC basado en tubos permiten la satisfacción robusta de restricciones para sistemas lineales [27] y no lineales [79, 22].

En [81] se presentaron condiciones suficientes para diseñar estrategias de MMMPC asintóticamente estables para sistemas cuyas incertidumbres dependen del estado, esto es, las incertidumbres desaparecen en el punto de equilibrio del sistema. Para sistemas con incertidumbres persistentes, la estabilidad entrada a estado (ISS, del inglés *Input-to-State Stability*) representa un marco apropiado para el análisis de la capacidad estabilizadora de MPC [126, 54]. En [68] se presentaron condiciones suficientes para

garantizar la estabilidad de sistemas con ciertos tipos de incertidumbres acotadas, controlados con estrategias de MMMPC. Considerando a priori las condiciones suficientes de estabilidad robusta, [62] presentó un enfoque nuevo para el diseño de MMMPC en bucle cerrado que garantiza ISS para sistemas no lineales. Actualmente hay muchos grupos de investigación que invierten un considerable esfuerzo en el análisis de la estabilidad de sistemas discretos con incertidumbres. Para una visón general de temas relacionados con la estabilidad robusta se remite al lector a [67].

### B.1.3 Objetivos

El objetivo general de esta tesis es contribuir al desarrollo de nuevas estrategias de MPC no lineal y robusto. El enfoque principal se hará a partir de los modelos de Volterra de tiempo discreto y su posible uso en estrategias de control de horizonte deslizante. Además, se considerarán los modelos lineales con incertidumbres en el marco de MPC robusto como paso previo para el desarrollo de MMMPC basado en modelos de Volterra. En el diseño de nuevas estrategias de control se prestará especial atención a la aplicabilidad práctica, relacionada con la carga computacional necesaria para resolver el problema de optimización, y también al estudio de estabilidad.

A pesar de la importancia de los procesos no lineales en la industria, rara vez se usan estrategias de MPC no lineal y robusto en aplicaciones industriales. Además, en el área de técnicas de MPC avanzado se puede observar una separación importante entre la investigación académica y la práctica industrial. Uno de los objetivos de esta tesis es el desarrollo de nuevas técnicas de MPC que cumplan los requisitos de las aplicaciones industriales. Una razón importante para el escaso éxito de MPC no lineal y robusto en la práctica industrial se encuentra en la dificultad para obtener modelos de predicción apropiados. Por este motivo, los modelos considerados en esta tesis son de fácil obtención a partir de datos experimentales de tipo entrada-salida, una práctica común en la industria. En el caso de los modelos de Volterra, representando la extensión lógica de los modelos de convolución lineal, se pueden identificar los parámetros con técnicas de identificación lineal. En los modelos lineales se ha considerado la incertidumbre persistente como término aditivo y acotado, con la posibilidad de obtener la cota directamente de la comparación de la salida del sistema y la salida predicha.

Otro problema en la aplicación de técnicas avanzadas de MPC es la complejidad computacional del problema de optimización resultante. En el caso de NMPC, el cálculo de la acción de control requiere la solución de un problema posiblemente no convexo. En ese caso, la estructura particular de los modelos de Volterra, es decir, la separabilidad de los términos lineales y no lineales, puede ser explotada para encontrar algoritmos

computacionalmente eficientes para resolver el problema de optimización. La complejidad numérica de las estrategias de MMMPC se encuentra en el cálculo del peor caso y crece de forma exponencial en función del horizonte de predicción. Teniendo en cuenta la aplicabilidad práctica, se estudiarán nuevos enfoques para determinar el peor caso. Además, las nuevas estrategias de control deben permitir una implementación simple y, en consecuencia, facilitar su utilización en aplicaciones industriales.

El análisis de la estabilidad es fundamental en el diseño de nuevas estrategias de MPC. La estabilidad es especialmente relevante en la implementación práctica de MPC cuando se consideran explícitamente incertidumbres en el modelo de predicción. Un objetivo importante en esta tesis será el análisis y la prueba de estabilidad tanto para MMMPC como NMPC.

Resumiendo los objetivos anteriormente mencionados, esta tesis tratará los siguientes temas:

- Desarrollo de novedosas estrategias de NMPC y MMMPC basado en modelos de Volterra y modelos lineales con incertidumbre
- Análisis de las propiedades estabilizantes de las estrategias de control y formulación de las condiciones necesarias para lograr la estabilidad
- Implementación de las diferentes estrategias de MPC y aplicación a procesos industriales no lineales

### B.1.4 Estructura de la tesis

La tesis está organizada de la siguiente forma:

- En el Capítulo 2 se dará una introducción general a los modelos de Volterra, con énfasis especial en los modelos de Volterra no autorregresivos de segundo orden en tiempo discreto y su uso en el MPC. Se explicarán la identificación de los parámetros a partir de datos empíricos y la transformación de la forma autorregresiva en su representación no autorregresiva. Para los modelos de Volterra de segundo orden se especificará un modelo de predicción que puede ser usado en el MPC. A continuación se presentará la función de coste, habitualmente cuadrática, del MPC y se definirá el problema general de optimización de MPC basado en modelos de Volterra de segundo orden. La última sección presentará los modelos de Volterra en espacio de estados, especialmente importantes para las pruebas de estabilidad necesarias.

- El Capítulo 3 dará una descripción detallada de los diferentes sistemas usados en esta tesis como bancos de pruebas, para la aplicación de estrategias de MPC. Los sistemas considerados incluyen una planta piloto, una pila de combustible y un invernadero, los cuales representan procesos no lineales. Se aproximarán las dinámicas no lineales de los diferentes procesos mediante modelos de Volterra de segundo orden para su posterior uso en estrategias de MPC. Los parámetros de los modelos se identificarán a partir de datos entrada-salida obtenidos mediante experimentos con dichos procesos.

- En el Capítulo 4 se estudiarán algoritmos iterativos para resolver el problema de optimización de NMPC basado en modelos de Volterra de segundo orden. Empezando con una formulación básica del problema de optimización sin restricciones, se introducirán diversas modificaciones en la función de coste y en el algoritmo iterativo, con el fin de considerar restricciones y una ponderación del esfuerzo de control. El algoritmo iterativo considerando restricciones será modificado para garantizar la estabilidad en bucle cerrado. Finalmente, se aplicarán los dos algoritmos iterativos con restricciones a distintos procesos y se ilustrará el rendimiento de los controladores mediante los resultados experimentales obtenidos.

- El Capítulo 5 presentará un enfoque de convexificación para estrategias de NMPC basado en modelos de Volterra de segundo orden. El enfoque permite la minimización global del problema de optimización, teniendo en cuenta una ponderación adecuada del esfuerzo de control. Mediante algunas modificaciones en el enfoque de convexificación se podrá garantizar la estabilidad de la estrategia de NMPC resultante. Además se verifica la aplicabilidad práctica de las estrategias de NMPC desarrolladas mediante experimentos con uno de los procesos usados como banco de pruebas.

- El Capítulo 6 se centra en dos estrategias de MMMPC para sistemas lineales con incertidumbre aditiva y persistente. La primera estrategia de control está basada en una cota superior del peor caso y reduce considerablemente la complejidad computacional del problema de optimización. En el caso de la segunda estrategia, se calcula una aproximación del peor caso mediante la solución de un problema QP. Para ambas estrategias de control, cada una con una carga computacional considerablemente más baja que el problema de optimización original, se puede garantizar la estabilidad. Por último, se aplican las estrategias de MMMPC a uno de los procesos y se presentan los resultados experimentales obtenidos.

- En el Capítulo 7 se desarrollará una estrategia de MMMPC basado en un modelo de Volterra de segundo orden con incertidumbre aditiva y persistente. Se demostrará que el carácter no autorregresivo del modelo permite la obtención de

una formulación explícita del peor caso. La carga computacional para calcular la secuencia de entrada es mucho menor que la del problema mín-máx original, ya que el problema de optimización degenera en un problema de minimización. Después de algunos cambios en la función de coste se puede demostrar estabilidad robusta del MMMPC desarrollado. Finalmente, se implementan y aplican las estrategias de MMMPC basado en un modelo de Volterra de segundo orden a uno de los sistemas usados como banco de pruebas y se exponen los resultados obtenidos con los experimentos.

- El Capítulo 8 resumirá las contribuciones y resultados de esta tesis y dará posibles direcciones para futuras investigaciones.
- El Apéndice A da una definición detallada del modelo de predicción basado en modelos de Volterra de segundo orden, así como otras definiciones matemáticas. En el Apéndice B se encuentran las traducciones al castellano de los Capítulos 1 y 8.

### B.1.5 Lista de publicaciones

Los siguientes artículos han sido publicados o enviados para su publicación durante la elaboración de esta tesis:

**Capítulos de libros:**

1. J. K. Gruber, C. Bordons (2010). Regulation of the air supply in a fuel cell using Model Predictive Control. *Power Plant Applications of Advanced Control Techniques*. ProcessEng Engineering GmbH, Viena, Austria. Enviado para publicación.

**Artículos de revista:**

1. J. K. Gruber, J. L. Guzmán, F. Rodríguez, C. Bordons, M. Berenguel, J. A. Sánchez (2010). Nonlinear MPC based on a Volterra series model for greenhouse temperature control using natural ventilation. *Control Engineering Practice*. Enviado para publicación.
2. J. K. Gruber, A. Oliva, C. Bordons (2010). Nonlinear MPC for the airflow in a PEM fuel cell using a Volterra series model. *Control Engineering Practice*. Enviado para publicación.

3. J. K. Gruber, C. Bordons, R. Bars, R. Haber (2009). Nonlinear predictive control of smooth nonlinear systems based on Volterra models. Application to a pilot plant. *International Journal of Robust and Nonlinear Control*. DOI: 10.1002/rnc.1549. Aceptado para publicación.

4. J. K. Gruber, D. R. Ramírez, T. Alamo, C. Bordons, E. F. Camacho (2009). Control of a pilot plant using QP based min-max predictive control. *Control Engineering Practice* **17**(11): 1358-1366.

5. J. K. Gruber, M. Doll, C. Bordons (2009). Design and experimental validation of a constrained MPC for the air feed of a fuel cell. *Control Engineering Practice* **17**(8): 874-885.

6. D. R. Ramírez, J. K. Gruber, T. Álamo, C. Bordóns, E. F. Camacho (2008). Control Predictivo Mín-Máx de una Planta Piloto (en castellano). *Revista Iberoamericana de Automática e Informática Industrial* **5**(3): 37-47.

7. J. K. Gruber, C. Bordons (2007). Control predictivo no lineal basado en modelos de Volterra. Aplicación a una planta piloto (en castellano). *Revista Iberoamericana de Automática e Informática Industrial* **4**(3): 34-45.

**Artículos de congreso:**

1. A. Ferramosca, J. K. Gruber, D. Limon, E. F. Camacho (2010). MPC for tracking of constrained nonlinear systems. Application to a pilot plant. Enviado a *49th IEEE Conference on Decision and Control*, Atlanta, EE.UU.

2. J. K. Gruber, T. Alamo, D. R. Ramírez, C. Bordons (2010). A convex approach for NMPC based on second order Volterra series models. Enviado a *49th IEEE Conference on Decision and Control*, Atlanta, EE.UU.

3. J. K. Gruber, D. R. Ramírez, T. Alamo, C. Bordons (2009). Nonlinear Min-Max Model Predictive Control Based on Volterra Models. Application to a Pilot Plant. En *Proceedings of the 2009 European Control Conference*, Budapest, Hungría.

4. J. K. Gruber, J. L. Guzmán, F. Rodríguez, M. Berenguel, C. Bordóns (2009). Nonlinear Model Predictive Control of Greenhouse Temperature Using a Volterra Model. En *Proceedings of the 2009 European Control Conference*, Budapest, Hungría.

5. J. K. Gruber, C. Bordons, F. Dorado (2008). Nonlinear control of the air feed of a fuel cell. En *Proceedings of the 2008 American Control Conference*, Seattle, EE.UU.

6. J. K. Gruber, D. R. Ramírez, T. Alamo, C. Bordons, E. F. Camacho (2008). Min-Max Model Predictive Control of a pilot plant. En *Proceedings of the 2008 American Control Conference*, Seattle, EE.UU.

7. J. K. Gruber, D. R. Ramírez, T. Alamo, C. Bordons, E. F. Camacho (2008). Min-Max Predictive Control of a Pilot Plant Using a QP Approach. En *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancún, México.

8. J. K. Gruber, F. Rodríguez, C. Bordóns, J. L. Guzmán, M. Berenguel, E. F. Camacho (2008). A Volterra Model of the Greenhouse Temperature Using Natural Ventilation. En *Proceedings of the 17th IFAC World Congress*, Seúl, Corea.

9. J. K. Gruber, C. Bordons, M. Doll (2008). Control de la Tasa de Exceso de Oxígeno de una Pila de Combustible (en castellano). En *Actas de las XXIX Jornadas de Automática*, Tarragona, España.

10. J. K. Gruber, F. Dorado, C. Bordons (2008). Nonlinear Predictive Control Using Volterra Models. Application to a pilot Plant. En *Proceedings of the Controlo 2008 Conference - 8th Portuguese Conference on Automatic Control*, Vila Real, Portugal.

11. J. K. Gruber, F. Rodríguez, C. Bordóns, J. L. Guzmán, M. Berenguel (2007). Modeling greenhouse temperature using Volterra models. En *Proceedings of the 2007 IFAC Agricontrol Conference*, Osijek, Croacia.

12. J. K. Gruber, C. Bordons, R. Ortega, R. Talj (2007). Diseño de un Controlador PI no Lineal para el Suministro de Aire de una Pila de Combustible (en castellano). In *Actas de las XXVIII Jornadas de Automática*, Huelva, España.

13. F. Dorado, J. K. Gruber, C. Bordons, E. F. Camacho (2007). Volterra Model Based Predictive Control. Application to a PEM Fuel Cell. En *Proceedings of the 14th Nordic Process Control Workshop*, Helsinki, Finlandia.

14. J. K. Gruber, C. Bordons (2006). Aplicación de MPC no Lineal Basado en Modelos de Volterra a una Planta Piloto (en castellano). En *Actas de las XXVII Jornadas de Automática*, Almería, España.

## B.2 Capítulo 8: Conclusiones

Esta tesis se dedicó al desarrollo de nuevas estrategias de MPC tanto para sistemas no lineales, como para sistemas con incertidumbre. Se ha prestado especial atención a la

aplicabilidad de las nuevas estrategias, teniendo en cuenta cuestiones de estabilidad. Los modelos de Volterra en tiempo discreto desempeñaron un papel fundamental en el desarrollo de estrategias de MPC, pero también se consideraron modelos lineales inciertos en tiempo discreto.

El éxito indiscutible del MPC lineal en aplicaciones industriales tiene varias razones, entre las más importantes destacan la formulación intuitiva del problema de optimización, la consideración de restricciones, la fácil implementación de la ley de control, así como la posibilidad de controlar procesos en una gama muy amplia de diferentes áreas industriales. En contraste, el NMPC y el MMMPC se utilizan escasamente en aplicaciones prácticas y, en consecuencia, estas estrategias reciben menos atención en sectores industriales. El menor interés en estas técnicas de MPC avanzado tiene su origen en varios factores, los más importantes son la dificultad de obtener modelos de predicción adecuados y la complejidad computacional para resolver el problema de optimización. Además, los algoritmos de programación no lineal, necesarios para resolver el problema de optimización posiblemente no convexo del NMPC son menos fiables que los algoritmos de optimización utilizados en el MPC lineal. Otros problemas más teóricos del estudio de técnicas de control avanzado como el NMPC y el MMMPC incluyen temas de estabilidad y robustez así como cuestiones de optimización.

Teniendo en cuenta los problemas mencionados, se desarrollaron en los capítulos anteriores varias estrategias de MPC robusto y no lineal. Los modelos para aproximar la dinámica de los procesos considerados se obtienen fácilmente a partir de datos experimentales de tipo entrada-salida y previenen los problemas de una identificación difícil y costosa. Se han considerado dos tipos diferentes de modelos en el desarrollo de estrategias de control: modelos de Volterra y modelos lineales en espacio de estados. Ambos modelos pueden ser identificados con métodos de estimación basada en mínimos cuadrados. Estos métodos representan una técnica de identificación estándar en ingeniería de control que es ampliamente utilizada en la industria. En ambos modelos se incluyeron incertidumbres aditivas, acotadas y persistentes con el objetivo de desarrollar estrategias de MPC robustas. Mediante la comparación de la salida medida del sistema con la salida predicha del modelo, se puede determinar fácilmente la cota de la incertidumbre.

Con respecto a la complejidad numérica para resolver el problema de optimización de estrategias de MMMPC y NMPC, se presentaron diferentes métodos para reducir la carga computacional. Se implementaron y aplicaron las estrategias de control al menos a uno de los sistemas usados como banco de pruebas. Mediante los resultados obtenidos en experimentos de seguimiento de referencia y rechazo de perturbación se ilustró el rendimiento de los controladores propuestos y se verificó la aplicabilidad

| Algoritmo | Estab. | Prog. | $t_c^{min}$ | $t_c^{mx}$ | $t_c^{avg}$ |
|---|---|---|---|---|---|
| Optim. iterativa | | QP | 0.021 s | 0.481 s | 0.059 s |
| Optim. iterativa | sí | QP | 0.031 s | 0.284 s | 0.076 s |
| Convexificación | | NLP | 0.066 s | 4.462 s | 0.736 s |
| Convexificación | sí | NLP | 0.200 s | 7.945 s | 1.115 s |
| MMMPC (cota no lin.) | sí | NLP | 0.398 s | 3.276 s | 0.783 s |
| MMMPC (cota cuadr.) | sí | QP | 0.088 s | 0.179 s | 0.097 s |
| MMMPC no lineal | | NLP | 0.146 s | 0.616 s | 0.442 s |
| MMMPC no lineal | sí | NLP | 0.285 s | 4.408 s | 0.939 s |
| MPC | | QP | 0.021 s | 0.112 s | 0.031 s |
| MPC | sí | QP | 0.031 s | 0.160 s | 0.060 s |

**Cuadro B.1**: Comparación entre las diferentes estrategias de control con respecto al tipo de problema de programación y al tiempo necesario para calcular la acción de control en los experimentos con la planta piloto.

práctica. Se demostró que una modificación en la función de coste y el uso de un horizonte suficientemente largo garantizan la estabilidad robusta de las estrategias de MPC basado en modelos de Volterra. Además, se comprobó que la estrategia de MMMPC lineal basado en una cota superior no lineal del peor caso, acota el estado del sistema y, en consecuencia, garantiza la estabilidad robusta.

Con el fin de dar una visión general de las estrategias de MPC desarrolladas se presenta en los Cuadros B.1 y B.2 un resumen de los tiempos necesarios para calcular una nueva secuencia de entrada (con los valores medios, mínimos y máximos de los tiempos de cálculo) y las sumas de errores cuadráticos (SSE, del inglés *Sum of Square Errors*) obtenidas en los diferentes experimentos con la planta piloto. Asimismo, los cuadros especifican el tipo de problema de programación y si la estrategia propuesta garantiza estabilidad en bucle cerrado. Además, por razones de comparación, se incluyeron en los cuadros los resultados numéricos de un MPC lineal (para información detallada sobre la aplicación del MPC véase el Apéndice A.9). Los resultados muestran que los algoritmos iterativos para el NMPC basado en modelos de Volterra (Capítulo 4), el MMMPC lineal basado en una cota superior cuadrática (Sección 6.3) y el MPC lineal (Apéndice A.9), es decir, los problemas de optimización que representan un problema de programación cuadrática (QP, del inglés *Quadratic Programming*), requieren poco tiempo para calcular la secuencia de control y pueden ser usados para controlar procesos con dinámica rápida. El enfoque de convexificación para el NMPC basado en

modelos de Volterra (Capítulo 5), el MMMPC lineal basado en una cota superior no lineal (Sección 6.2) y el MMMPC no lineal basado en modelos de Volterra (Capítulo 7) constituyen problemas de programación no lineal (NLP, del inglés *Nonlinear Programming*) con una complejidad computacional considerablemente superior a la de los problemas QP. Sin embargo, con métodos modernos de NLP, p.ej. algoritmos de programación cuadrática secuencial (SQP, del inglés *Sequential Quadratic Programming*), se pueden resolver los problemas de optimización de las estrategias de MPC propuestas en un tiempo razonable y, por tanto, tales estrategias de control son válidas para procesos con dinámica lenta o intermedia. Obviamente, la comparación de los resultados de las estrategias de control propuestas es una tarea difícil[2], ya que algunas estrategias dieron mejores resultados en unos experimentos y otras rindieron mejor en otros, p.ej. los mejores valores de SSE en el experimento de seguimiento de referencia se obtenían con el NMPC basado en el enfoque de convexificación, mientras que el MMMPC no lineal dio los mejores resultados en los experimentos con una perturbación en la válvula.

Sin embargo, se puede ver claramente que la aplicación de las estrategias de MMMPC y NMPC a la planta piloto resulta en todos los casos en índices de errores cuadráticos más bajos que el MPC lineal.

Resumiendo los anteriores capítulos, las contribuciones principales de esta tesis son los siguientes temas claves:

- Se han aproximado diferentes tipos de procesos con dinámica no lineal mediante modelos de Volterra de segundo orden, para su uso posterior en aplicaciones prácticas de NMPC. El comportamiento dinámico complejo de los procesos es difícil de aproximar mediante modelos matemáticos, incluso con modelos no lineales. Sin embargo, el carácter no lineal de los modelos de Volterra ofrece la posibilidad de mejorar la calidad de la aproximación en comparación con los modelos lineales. Además, uno de los sistemas considerados ha sido aproximado por un modelo lineal con incertidumbre persistente y aditiva. Las diferentes estructuras de modelo utilizadas permiten determinar los parámetros con técnicas de identificación lineal y han sido elegidas con la idea de una fácil identificación.

- Se ha estudiado un algoritmo de optimización iterativo para el NMPC basado en modelos de Volterra. Este algoritmo es computacionalmente eficiente, pero no considera restricciones en la optimización, además la función de coste del

[2]Un análisis comparativo de las diferentes estrategias de MPC con respecto a su rendimiento debe basarse en resultados de simulaciones por ordenador, a fin de evitar la influencia de perturbaciones exógenas no reproducibles. Además, para un análisis detallado de las estrategias propuestas con respecto al seguimiento de referencia, el rechazo de perturbaciones y los problemas de estabilización se deben usar índices de rendimiento diferenciados.

| Algoritmo | Estab. | nominal | error parám. | dist. alim. | dist. valvula |
|---|---|---|---|---|---|
| Optim. iterativa | | 2056.51 | 2098.15 | 1120.80 | 1088.31 |
| Optim. iterativa | sí | 2140.14 | 2136.82 | 1447.52 | 1495.71 |
| Convexificación | | 1934.44 | 1930.41 | 1119.80 | 1123.22 |
| Convexificación | sí | 1660.83 | 1690.92 | 803.72 | 764.42 |
| MMMPC (cota no lin.) | sí | 2420.20 | 2301.37 | 1071.37 | 684.03 |
| MMMPC (cota cuadr.) | sí | 2463.15 | 2294.94 | 1129.73 | 677.65 |
| MMMPC no lineal | | 2224.91 | 2208.78 | 1042.3 | 633.40 |
| MMMPC no lineal | sí | 2316.90 | 2318.31 | 1020.65 | 695.73 |
| MPC | | 2675.26 | 2635.40 | 1549.22 | 1535.22 |
| MPC | sí | 2621.67 | 2517.47 | 1529.92 | 1767.12 |

**Cuadro B.2**: Comparación entre las diferentes estrategias de control con respecto a las sumas de errores cuadráticos obtenidas en los experimentos con la planta piloto.

NMPC no incluye una ponderación del esfuerzo de control. En primer lugar se ha extendido el algoritmo iterativo y el problema de optimización para incluir tanto restricciones como una penalización del esfuerzo de control. Con una modificación adicional en la función de coste y un horizonte de predicción apropiado se ha demostrado estabilidad entrada a estado (ISS, del inglés *Input-to-State Stability*) del algoritmo iterativo con restricciones.

- Con vista al problema de optimización posiblemente no convexo, resultado de la combinación de una función de coste cuadrática y un modelo no lineal, se ha desarrollado un enfoque de convexificación para estrategias de NMPC basado en modelos de Volterra de segundo orden. En este nuevo enfoque se aproxima la función de coste original mediante una serie de funciones cuadráticas convexas. Estas aproximaciones representan una envoltura convexa de la función de coste original y permiten la minimización global del problema original con algoritmos estándares de optimización convexa. Después de una modificación en la función de coste, se ha demostrado estabilidad robusta del enfoque de convexificación para horizontes de predicción que satisfacen cierta condición.

- Se ha demostrado estabilidad robusta para una estrategia de MMMPC lineal basado en una cota superior del peor caso. Esta estrategia reduce considerablemente la complejidad numérica del problema de optimización usando un algoritmo de diagonalización para determinar la cota superior. Por tanto, el cálculo de la cota requiere sólo la diagonalización de una matriz, en lugar de la evaluación

de todos los posibles vértices de la secuencia de perturbaciones futuras. En consecuencia, el MMMPC permite una computación rápida de la cota mencionada, así como el uso de horizontes de predicción más largos.

- Con el fin de considerar posibles perturbaciones, se ha desarrollado una novedosa estrategia de MMMPC basado en modelos de Volterra. En el modelo de predicción se ha considerado la incertidumbre como término aditivo y persistente, donde la estructura del modelo permite una fácil obtención de la cota de la incertidumbre, mediante la comparación entre la salida del sistema y la salida predicha del modelo. Por último, con una modificación en la función de coste, se ha demostrado que la estrategia de control propuesta garantiza estabilidad robusta para un horizonte de predicción suficientemente largo.

- Para comprobar la aplicabilidad práctica, se han implementado y validado las diferentes estrategias de control presentadas (con excepción del algoritmo iterativo de optimización sin restricciones, que se puede considerar como punto de partida para el desarrollo de nuevas estrategias de control) en experimentos con al menos uno de los sistemas usados como banco de pruebas. En los resultados de los experimentos se ha observado que las estrategias de NMPC y MMMPC tienen un mejor rendimiento de control en comparación con un MPC lineal. El empleo exitoso de las diferentes estrategias se une al bajo número de aplicaciones de NMPC y MMMPC existentes en la literatura especializada.

Por último, posibles futuras investigaciones podrían incluir las siguientes direcciones:

- Los algoritmos iterativos de optimización para el NMPC basado en modelos de Volterra de segundo orden son computacionalmente muy eficientes, pero muestran problemas de convergencia en el caso de un problema de optimización no convexa. Sin embargo, se puede garantizar la convergencia del algoritmo iterativo mediante la elección de una ponderación apropiada del esfuerzo de control o un prefiltro de la referencia. Por tanto, se sugiere un análisis detallado de las condiciones necesarias para asegurar la convergencia de los algoritmos iterativos.

- El enfoque de convexificación presentado aproxima una función de coste posiblemente no convexa mediante funciones cuadráticas convexas y permite la minimización global del problema inicial. La minimización simultánea de varias funciones cuadráticas convexas representa un problema de programación cuadrática que se puede resolver mediante algoritmos modernos de optimización convexa como métodos de SQP. El uso de desigualdades lineales matriciales (LMI, del inglés *Linear Matrix Inequalities*) podría reducir la carga computacional necesaria para resolver el problema de optimización y, en consecuencia, permitir calcular más rápido la secuencia de control.

- La estrategia de MMMPC para modelos de Volterra de segundo orden con incertidumbre persistente y aditiva, está basada en una formulación explícita del peor caso, es decir, el problema de optimización mín-máx degenera en un problema de minimización y se puede resolver usando métodos de NLP. Para esta estrategia se propone el estudio de posibles soluciones del problema de minimización mediante algoritmos iterativos o el enfoque de convexificación.

- Los modelos de Volterra no autorregresivos permiten la aproximación de una gran variedad de sistemas estables en bucle abierto, pero no son válidos para modelar sistemas inestables. No obstante, se puede pre-estabilizar el sistema con la ayuda de controladores por realimentación y aproximar el sistema extendido (proceso + controlador por realimentación) por un modelo de Volterra no autorregresivo. Un esquema de control jerárquico que consiste en un bucle exterior con el controlador basado en un modelo de Volterra y un bucle interior con el control por realimentación, podría ampliar el rango de posibles aplicaciones de MPC basado en modelos de Volterra.

- El horizonte finito y las restricciones consideradas en un problema de optimización de una estrategia de MPC puede resultar en estados inalcanzables para una referencia dada. Especialmente después de cambios en la consigna, no siempre se puede garantizar la factibilidad del problema de optimización. Con el fin de evitar la pérdida de factibilidad se puede considerar el problema de seguimiento en el diseño de estrategias de MPC, de manera que se asegure la satisfacción de las restricciones para cualquier referencia admisible.

- Se han presentado diferentes estrategias de MMMPC, tanto para modelos lineales, como para modelos de Volterra de segundo orden y sus aplicaciones a los sistemas usados como bancos de pruebas. En un estudio más teórico se podría analizar el aumento de robustez obtenido con las estrategias propuestas.

- Otra posibilidad en el MMMPC basado en modelos de Volterra es el uso de incertidumbres paramétricas e incertidumbres que dependen del estado del sistema. Una posible línea futura de investigación podría considerar tales incertidumbres en el desarrollo de nuevas estrategias de MMMPC y analizar la robustez en bucle cerrado, un punto importante en aplicaciones industriales.

# Bibliography


[1] T. Alamo, D. Muñoz de la Peña, D.M. Limon, and E.F. Camacho. Constrained min-max predictive control: modifications of the objective function leading to polynomial complexity. *IEEE Transactions on Automatic Control*, 50(5):710–714, 2005.

[2] T. Alamo, D.R. Ramírez, and D. Muñoz de la Peña. Min-Max MPC using a tractable QP Problem. In *Proceedings of the 44th IEEE Conference on Decision and Control and the 2005 European Control Conference*, pages 6210–6215, Sevilla, Spain, 2005.

[3] T. Alamo, D.R. Ramírez, D. Muñoz de la Peña, and E.F. Camacho. Min-max MPC using a tractable QP problem. *Automatica*, 43(4):693–700, 2007.

[4] T.L. Aliyev, A.T. Stamps, and E.P. Gatzke. Improved crude oil processing using second-order Volterra models and nonlinear Model Predictive Control. In *Proceedings of the 2008 American Control Conference*, pages 2215–2220, Seattle, WA, 2008.

[5] J.C. Allwright and G.C. Papavasiliou. On linear programming and robust model-predictive control using impulse-responses. *Systems & Control Letters*, 18(2):159–164, 1992.

[6] M.R. Arahal, M. Berenguel, and E.F. Camacho. Neural identification applied to predictive control of a solar plant. *Control Engineering Practice*, 6(3):333–344, 1998.

[7] M.R. Arahal, F. Rodríguez, A. Ramírez-Arias, and M. Berenguel. Discrete-time nonlinear FIR models with integrated variables for greenhouse indoor temperature simulation. In *Proceedings of the 44th IEEE Conference on Decision and Control and the 2005 European Control Conference*, pages 4158–4162, Sevilla, Spain, 2005.

[8] A. Arce, A.J. del Real, C. Bordons, and D.R. Ramírez. Real-Time Implementation of a Constrained MPC for Efficient Airflow Control in a PEM Fuel Cell. *IEEE*

*Transactions on Industrial Electronics*, 2009. DOI: 10.1109/TIE.2009.2029524. Accepted for publication.

[9] A. Arce, D.R. Ramirez, A.J. del Real, and C. Bordons. Constrained explicit predictive control strategies for PEM fuel cell systems. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 6088–6093, Louisiana, LA, 2007.

[10] M.H. Asyali and M. Juusola. Use of Meixner functions in estimation of Volterra kernels of nonlinear systems with delay. *IEEE Transactions on Biomedical Engineering*, 52(2):229–237, 2005.

[11] Ballard. *Nexa Power Module User's Manual*. Ballard Power Systems Inc., Burnaby, Canada, 2003.

[12] F. Barbir. *PEM Fuel Cells: Theory and Practice*. Academic Press, 2005.

[13] M. Bauer and I.K. Craig. Economic assessment of advanced process control Ű A survey and framework. *Journal of Process Control,*, 18(1):2–18, 2008.

[14] A. Bemporad, F. Borrelli, and M. Morari. Min-max Control of Constrained Uncertain Discrete-Time Linear Systems. *IEEE Transactions on Automatic Control*, 48(9):1600–1606, 2003.

[15] M. Berenguel, F. Rodríguez, J.L. Guzmán, D. Lacasa, and J. Pérez-Parra. Greenhouse diurnal temperature control with natural ventilation based on empirical models. *Acta Horticulturae*, 179(1):205–211, 2006.

[16] D.P. Bertsekas and I.B. Rhodes. On the minimax reachability of target sets and target tubes. *Automatica*, 7(2):233–247, 1971.

[17] L.T. Biegler. Efficient Solution of Dynamic Optimization and NMPC Problems. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, Progress in Systems and Control Theory, pages 219–243. Birkhäuser Verlag, 2000.

[18] C. Bordons, A. Arce, and A.J. del Real. Constrained predictive control strategies for PEM fuel cells. In *Proceedings of the 2006 American Control Conference*, pages 2486–2491, Minneapolis, MN, 2006.

[19] T. Boulard and A. Baille. Modelling of air exchange rate in a greenhouse equipped with continuous roof vents. *Journal of Agricultural Engineering Research*, 61(1):37–48, 1995.

[20] S. Boyd and L.O. Chua. Fading Memory and the Problem of Approximating Nonlinear Operators with Volterra Series. *IEEE Transactions on Circuits and Systems*, 32(11):1150–1161, 1985.

[21] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2006.

[22] J.M. Bravo, T. Alamo, and E.F. Camacho. Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica*, 42(10):1745–1751, 2006.

[23] E.F. Camacho and C. Bordons. *Model Predictive Control*. Springer, second edition, 2004.

[24] R.J.G.B. Campello, G. Favier, and W.C. do Amaral. Optimal expansions of discrete-time Volterra models using Laguerre functions. *Automatica*, 40(5):815–822, 2004.

[25] P.J. Campo and M. Morari. Robust Model Predictive Control. In *Proceedings of the 1987 American Control Conference*, pages 1021–1026, Minneapolis, MN, 1987.

[26] H.-W. Chen. Modeling and Identification of Parallel Nonlinear Systems: Structural Classification and Parameter Estimation Methods. *Proceedings of the IEEE*, 83(1):39–66, 1995.

[27] L. Chisci, J.A. Rossiter, and G. Zappa. Systems with persistent disturbances: predictive control with restricted constraints. *Automatica*, 37(7):1019–1028, 2001.

[28] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized Predictive Control - Part I. The Basic Algorithm. *Automatica*, 23(2):137–148, 1987.

[29] J.R. Cueli. *Control predictivo para procesos repetitivos (in Spanish)*. PhD thesis, Departamento de Ingeniería de Sistemas y Automática. Universidad de Sevilla, 2005.

[30] J.R. Cueli and C. Bordons. Iterative nonlinear model predictive control. Stability, robustness and applications. *Control Engineering Practice*, 16(9):1023–1034, 2008.

[31] C.R. Cutler and B.C. Ramaker. Dynamic Matrix Control – a computer control algorithm. In *Proceedings of the 1980 Joint Automatic Control Conference*, San Francisco, CA, 1980.

[32] A. da Rosa, R.J.G.B. Campello, and W.C. Amaral. Choice of free parameters in expansions of discrete-time Volterra models using Kautz functions. *Automatica*, 43(6):1084–1091, 2007.

[33] G. de Nicolao, L. Magni, and R. Scattolini. On the robustness of receding-horizon control with terminal constraints. *IEEE Transactions on Automatic Control*, 41(3):451–453, 1996.

[34] M. Diehl. Formulation of Closed-Loop Min-Max MPC as a Quadratically Constrained Quadratic Program. *IEEE Transactions on Automatic Control*, 52(2):339–343, 2007.

[35] F. Dorado. *Control predictivo no lineal basado en modelos de Volterra (in Spanish)*. PhD thesis, Departamento de Ingeniería de Sistemas y Automática. Universidad de Sevilla, 2006.

[36] F. Dorado and C. Bordons. Constrained nonlinear predictive control using Volterra models. In *Proceedings of the 10th IEEE Conference on Emerging Technologies and Factory Automation*, pages 1007–1013, Catania, Italy, 2005.

[37] F.J. Doyle, B.A. Ogunnaike, and R.K. Pearson. Nonlinear model-based control using second-order Volterra models. *Automatica*, 31(5):697–714, 1995.

[38] F.J. Doyle, R.K. Pearson, and B.A. Ogunnaike. *Identification and Control Using Volterra Models*. Springer, London, 2001.

[39] C.G. Economou. *An Operator Theory Approach to Nonlinear Controller Design*. PhD thesis, California Institute of Technology, 1986.

[40] W. Garcia-Gabin, F. Dorado, and C. Bordons. Real-time implementation of a sliding mode controller for air supply on a PEM fuel cell. *Journal of Process Control*, 20(3):325–336, 2010.

[41] H. Genceli and M. Nikolaou. Robust stability analysis of constrained $l_1$-norm model predictive control. *AIChE Journal*, 39(12):1954–1965, 1993.

[42] G.B. Giannakis and E. Serpedin. A bibliography on nonlinear system identification. *Signal Processing*, 81(3):533–580, 2001.

[43] P.J. Goulart, E.C. Kerrigan, and J.M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533, 2006.

[44] J.K. Gruber and C. Bordons. Control predictivo no lineal basado en modelos de Volterra. Aplicación a una planta piloto (in Spanish). *Revista Iberoamericana de Automática e Informática Industrial*, 4(3):34–45, 2007.

[45] J.K. Gruber, C. Bordons, R. Bars, and R. Haber. Nonlinear predictive control of smooth nonlinear systems based on Volterra models. Application to a

pilot plant. *International Journal of Robust and Nonlinear Control*, 2009. DOI: 10.1002/rnc.1549. Accepted for publication.

[46] J.K. Gruber, M. Doll, and C. Bordons. Design and experimental validation of a constrained MPC for the air feed of a fuel cell. *Control Engineering Practice*, 17(8):874–885, 2009.

[47] J.K. Gruber, F. Dorado, and C. Bordons. Nonlinear predictive control using Volterra models. Application to a pilot plant. In *Proceedings of the 8th Portuguese Conference on Automatic Control*, pages 124–129, Vila Real, Portugal, 2008.

[48] J.K. Gruber, D.R. Ramirez, T. Alamo, C. Bordons, and E.F. Camacho. Control of a pilot plant using QP based min-max predictive control. *Control Engineering Practice*, 17(11):1358–1366, 2009.

[49] J.K. Gruber, F. Rodríguez, C. Bordons, J.L. Guzmán, M. Berenguel, and E.F. Camacho. A Volterra model of the greenhouse temperature using natural ventilation. In *Proceedings of the 17th IFAC World Congress*, pages 2925–2930, Seoul, Korea, 2008.

[50] R. Haber. Predictive control of nonlinear dynamic processes. *Applied Mathematics and Computation*, 70(2-3):169–184, 1995.

[51] R. Haber, R. Bars, and F. Arousi. An Iterative GPC-like Nonlinear Predictive Control Algorithm for the Parametric Volterra Model. In *Proceedings of the 2009 European Control Conference*, pages 3383–3388, Budapest, Hungary, 2009.

[52] R. Haber and L. Keviczky. *Nonlinear System Identification – Input-Output Modeling Approach*. Kluwer Academic Publishers, 1999.

[53] B. Hu and A. Linnemann. Toward Infinite-Horizon Optimality in Nonlinear Model Predictive Control. *IEEE Transactions on Automatic Control*, 47(4):679–682, 2002.

[54] Z.-P. Jiang and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

[55] E.C. Kerrigan and J.M. Maciejowski. Feedback min-max model predictive control using a single linear program: robust stability and the explicit solution. *International Journal of Robust and Nonlinear Control*, 14(4):395–413, 2004.

[56] A.Y. Kibangou, G. Favier, and M.M. Hassani. Selection of generalized orthonormal bases for second-order Volterra filters. *Signal Processing*, 85(12):2371–2385, 2005.

[57] C. Kittas, T. Boulard, and G. Papadakis. Natural ventilation of a greenhouse with ridge and side openings: Sensitivity to temperature and wind effects. *Transactions of the ASAE*, 40(2):415–425, 1997.

[58] M.J. Korenberg and I.W. Hunter. The identification of nonlinear biological systems: Volterra kernel approaches. *Annals of Biomedical Engineering*, 24(2):250–268, 1996.

[59] M.V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear model inequalities. *Automatica*, 32(10):1361–1379, 1996.

[60] F. Kuech and W. Kellermann. Orthogonalized power filters for nonlinear acoustic echo cancellation. *Signal Processing*, 86(6):1168–1181, 2006.

[61] J. Larminie and A. Dicks. *Fuel cell systems explained*. J. Wiley & Sons, 2nd edition, 2003.

[62] M. Lazar, D. Muñoz de la Peña, W.P.M.H. Heemels, and T. Alamo. On input-to-state stability of min-max nonlinear model predictive control. *Systems & Control Letters*, 57(1):39–48, 2008.

[63] J.H. Lee, K.S. Lee, and W.C. Kim. Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*, 36(5):641–657, 2000.

[64] J.H. Lee and Z. Yu. Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, 33(5):763–781, 1997.

[65] C.E. Lemke. Bimatrix equilibrium points and mathematical programming. *Management Science*, 11(7):681–689, 1965.

[66] Y. Li and H. Kashiwagi. High-order Volterra Model Predictive Control and its application to a nonlinear polymerisation process. *International Journal of Automation and Computing*, 2(2):208–214, 2005.

[67] D. Limon, T. Alamo, D.M. Raimondo, D. Muñoz de la Peña, J.M. Bravo, A. Ferramosca, and E.F. Camacho. Input-to-State Stability: A Unifying Framework for Robust Model Predictive Control. In L. Magni, D.M. Raimondo, and F. Allgöwer, editors, *Nonlinear Model Predictive Control*, Lecture Notes in Control and Information Sciences, pages 1–26. Springer, 2009.

[68] D. Limon, T. Alamo, F. Salas, and E.F. Camacho. On the stability of constrained MPC without terminal constraint. *IEEE Transactions on Automatic Control*, 51(5):832–836, 2006.

[69] D. Limon, I. Alvarado, T. Alamo, and E.F. Camacho. Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, 20(3):248–260, 2010.

[70] W.-M. Ling and D.E. Rivera. Control relevant model reduction of Volterra series models. *Journal of Process Control*, 8(2):79–88, 1998.

[71] W.-M. Ling and D.E. Rivera. A methodology for control-relevant nonlinear system identification using restricted complexity models. *Journal of Process Control*, 11(2):209–222, 2001.

[72] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, second edition, 1999.

[73] J. Löfberg. *Minimax Approaches to Robust Model Predictive Control*. PhD thesis, Department of Electrical Engineering, Linköping University, 2003.

[74] Y. Lu and Y. Arkun. Quasi-Min-Max MPC Algorithms for LPV Systems. *Automatica*, 36(4):527–540, 2000.

[75] J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2000.

[76] M.M. Mäkelä. Survey of bundle methods for nonsmooth optimization. *Optimization Methods and Software*, 17(1):1–29, 2002.

[77] B.R. Maner, F.J Doyle, B.A. Ogunnaike, and R.K. Pearson. Nonlinear Model Predictive Control of a Simulated Multivariable Polymerization Reactor Using Second-order Volterra Models. *Automatica*, 32(9):1285–1301, 1996.

[78] V.Z. Marmarelis. Modeling methology for nonlinear physiological systems. *Annals of Biomedical Engineering*, 25(2):239–251, 1997.

[79] D.L. Marruedo, T. Alamo, and E.F. Camacho. Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 4619–4624, Las Vegas, NV, 2002.

[80] C.F. Martin and Y. Zhou. Carleman linearization of linear systems with polynomial output. Report No. 09 (2002/2003), ISSN 1103-467X, The Royal Swedish Academy of Sciences, 2002.

[81] D.Q. Mayne. Control of constrained dynamic systems. *European Journal of Control*, 7(2-3):87–99, 2001.

[82] D.Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990.

[83] D.Q. Mayne, S.V. Raković, R. Findeisen, and F. Allgöwer. Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42(7):1217–1222, 2006.

[84] D.Q. Mayne, S.V. Raković, R. Findeisen, and F. Allgöwer. Robust output feedback model predictive control of constrained linear systems: Time varying case. *Automatica*, 45(9):2082–2087, 2009.

[85] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[86] H. Michalska and D.Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

[87] G.D. Mitsis and V.Z. Marmarelis. Modeling of Nonlinear Physiological Systems with Fast and Slow Dynamics. I. Methodology. *Annals of Biomedical Engineering*, 30(2):272–281, 2004.

[88] M. Morari and J.H.Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682, 1999.

[89] D. Muñoz de la Peña, T. Alamo, A. Bemporad, and E.F. Camacho. A Decomposition Algorithm for Feedback Min-Max Model Predictive Control. *IEEE Transactions on Automatic Control*, 51(10):1688–1692, 2006.

[90] D. Muñoz de la Peña, D.R. Ramírez, E.F. Camacho, and T. Alamo. Explicit solution of min-max MPC with additive uncertainties and quadratic criterion. *Systems & Control Letters*, 55(4):266–274, 2006.

[91] G. De Nicolao, L. Magni, and R. Scattolini. Robustness of receding horizon control for nonlinear discrete-time systems. In A. Garulli, A. Tesi, and A. Vicino, editors, *Robustness in identification and control*, Lecture Notes in Control and Information Sciences, pages 408–421. Springer, 1999.

[92] B. Ninness and F. Gustafsson. A unifying construction of orthonormal bases for system identification. *IEEE Transactions on Automatic Control*, 42(2):515–521, 1997.

[93] J.E. Normey-Rico and E.F. Camacho. *Control of Dead-Time Processes*. Springer-Verlag, 2007.

[94] R.D. Nowak and B.D. Van Veen. Nonlinear system identification with pseudo-random multilevel excitation sequences. In *Proceedings of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 456–459, Minneapolis, MN, 1993.

[95] R.D. Nowak and B.D. Van Veen. Efficient methods for identification of Volterra filter models. *Signal Processing*, 38(3):417–428, 1994.

[96] R.D. Nowak and B.D. Van Veen. Random and Pseudorandom Inputs for Volterra Filter Identification. *IEEE Transactions on Signal Processing*, 42(8):2124–2135, 1994.

[97] B.A. Ogunnaike. Nonlinear control of industrial processes. In B. Kouvaritakis and M. Cannon, editors, *Nonlinear predictive control: theory and practice*, IEE Control Series, pages 205–221. The Institution of Engineering and Technology, 2001.

[98] G.H.C. Oliveira, W.C. Amaral, and K. Latawiec. CRHPC using Volterra Models and Orthonormal Basis Functions: An Application to CSTR Plants. In *Proceedings of the 2003 IEEE Conference on Control Applications*, pages 718–723, Istanbul, Turkey, 2003.

[99] M.G. Ortega and F.R. Rubio. Systematic design of weighting matrices for the $H_\infty$ mixed sensitivity problem. *Journal of Process Control*, 14(1):89–98, 2004.

[100] M.G. Ortega, M. Vargas, F. Castaño, and F.R. Rubio. Improved design of the weighting matrices for the S/KS/T mixed sensitivity problem – Application to a multivariable thermodynamic system. *IEEE Transactions on Control Systems Technology*, 14(1):82–90, 2006.

[101] R.S. Parker, D. Heemstra, F.J. Doyle, R.K. Pearson, and B.A. Ogunnaike. The identification of nonlinear models for process control using tailored "plant-friendly" input sequences. *Journal of Process Control*, 11(2):237–250, 2001.

[102] J.T. Pukrushpan, H. Peng, and A.G. Stefanopoulou. Control-oriented modeling and analysis for automotive fuel cell systems. *Journal of Dynamic Systems, Measurement and Control*, 126(1):14–25, 2004.

[103] J.T. Pukrushpan, A.G. Stefanopoulou, and H. Peng. *Control of Fuel Cell Power Systems: Principles, Modeling, Analysis and Feedback Design*. Springer, London, 2004.

[104] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

[105] D.R. Ramirez, T. Alamo, E.F. Camacho, and D. Muñoz de la Peña. Min-Max MPC based on a computationally efficient upper-bound of the worst case cost. *Journal of Process Control*, 16(5):511–519, 2006.

[106] D.R. Ramirez and E.F. Camacho. Piecewise Affinity of Min-Max MPC with bounded additive uncertainties and a quadratic criterion. *Automatica*, 42(2):295–302, 2006.

[107] D.R. Ramírez, E.F. Camacho, and M.R. Arahal. Implementation of min-max MPC using hinging hyperplanes. Application to a heat exchanger. *Control Engineering Practice*, 12(9):1197–1205, 2004.

[108] D.R. Ramírez, D. Limón, J.G. Ortega, and E.F. Camacho. Model Based Predictive Control Using Genetic Algorithms. Application to a Pilot Plant. In *Proceedings of the 1999 European Control Conference*, pages 81–85, Karlsruhe, Germany, 1999.

[109] C.A. Ramos-Paja, C. Bordons, A. Romero, R. Giral, and L. Martínez-Salamero. Minimum Fuel Consumption Strategy for PEM Fuel Cells. *IEEE Transactions on Industrial Electronics*, 56(3):685–696, 2009.

[110] M.J. Reed and M.O.J. Hawksford. Identification of discrete Volterra series using maximum length sequences. *IEE Proceedings on Circuits, Devices and Systems*, 143(5):241–248, 1996.

[111] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Algorithmic control of industrial processes. In *Proceedings of the 4th IFAC Symposium on Identification and System Parameter Estimation*, Tbilisi, USSR, 1976.

[112] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model Predictive Heuristic Control: Applications to Industrial Processes. *Automatica*, 14(5):413–428, 1978.

[113] D.E. Rivera, H. Lee, H.D. Mittelmann, and M.W. Braun. Constrained multisine input signals for plant-friendly identification of chemical process systems. *Journal of Process Control*, 19(4):623–635, 2009.

[114] P. Rodatz, G. Paganelli, A. Sciarretta, and L. Guzzella. Optimal power management of an experimental fuel cell/supercapacitor-powered hybrid vehicle. *Control Engineering Practice*, 13:41–53, 2005.

[115] F. Rodríguez, M.R. Arahal, and M. Berenguel. Application of Artificial NN for Greenhouse Climate Modeling. In *Proceedings of the 1999 European Control Conference*, pages 300–305, Karlsruhe, Germany, 1999.

[116] F. Rodríguez, L.J. Yebra, M. Berenguel, and S. Dormido. Modelling and simulation of greenhouse climate using Dymola. In *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, 2002.

[117] F. Rodríguez Díaz. *Modelado y Control Jerárquico de Crecimiento de Cultivos en Invernadero (in Spanish)*. PhD thesis, Departamento de Lenguajes y Computación, Universidad de Almería, 2002.

[118] J.A. Rossiter. *Model-Based Predictive Control: A Practical Approach*. CRC Press, 2003.

[119] F.R. Rubio and M.J. López. *Control Adpatativo y Robusto (in Spanish)*. Secretariado de Publicaciones de la Universidad de Sevilla, 1996.

[120] W.J. Rugh. *Nonlinear System Theory: The Volterra/Wiener Approach*. Johns Hopkins University Press, Baltimore, MD, 1981.

[121] L.O. Santos, P.A.F.N.A. Afonso, J.A.A.M. Castro, N.M.C. Oliveira, and L.T. Biegler. On-line implementation of nonlinear MPC: an experimental case study. *Control Engineering Practice*, 9(8):847–857, 2001.

[122] P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.

[123] P.O.M. Scokaert, D.Q. Mayne, and J.B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3):648–654, 1999.

[124] C. Seretis and E. Zafiriou. Nonlinear dynamical system identification using reduced Volterra models with generalised orthonormal basis functions. In *Proceedings of the 1997 American Control Conference*, pages 3042–3046, Albuquerque, NM, 1997.

[125] T. Söderström and P. Stoica. Comparison of some instrumental variable methods – Consistency and accuracy aspects. *Automatica*, 17(1):101–115, 1981.

[126] E.D. Sontag and Y. Wang. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24(5):351–359, 1995.

[127] A.G. Stefanopoulou and K.-W. Suh. Mechatronics in fuel cell systems. *Control Engineering Practice*, 15(3):277–289, 2007.

[128] A. Stenger, L. Trautmann, and R. Rabenstein. Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filters. In *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 887–882, Phoenix, AZ, 1999.

[129] F. Szeifert, T. Chovan, and L. Nagy. Process dynamics and temperature control of fed-batch reactors. *Computers & Chemical Engineering*, 19(1):447–452, 1995.

[130] H. Takatsu, T. Itoh, and M. Araki. Future needs for the control theory in industries – report and topics of the control technology survey in Japanese industry. *Journal of Process Control*, 8(5-6):369–374, 98.

[131] P. Thounthong and P. Sethakul. Analysis of a fuel starvation phenomenon of a PEM fuel cell. In *Proceedings of the 4th Power Conversion Conference*, pages 731–738, Nagoya, Japan, 2007.

[132] V. Volterra. *Theory of Functionals and of Integral and Integro-Differential Equations*. Dover Publications, 1959.

[133] Y.J. Wang and J.B. Rawlings. A new robust model predictive control method I: theory and computation. *Journal of Process Control*, 14(3):231–247, 2004.

[134] H.S. Witsenhausen. A min-max control problem for sampled linear systems. *IEEE Transations on Automatic Control*, 13(1):5–21, 1968.

[135] A. Zheng and M. Morari. Stability of model predictive control with mixed constraints. *IEEE Transactions on Automatic Control*, 40(10):1818–1823, 1995.

[136] D. Zhou and V. DeBrunner. Efficient adaptive nonlinear filters for nonlinear active noise control. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(3):669–681, 2007.