Trabajo de Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Study on the cybersecurity in distributed control systems based on MPC

Autor: Paula Chanfreut Palacio

Tutor: José María Maestre Torreblanca

Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2017

Trabajo de Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Study on the cybersecurity in distributed control systems based on MPC

Autor:

Paula Chanfreut Palacio

Tutor:

José María Maestre Torreblanca

Profesor Contratado Doctor

Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2017

Trabajo de Fin de Grado: Study on the cybersecurity in distributed control systems based on MPC

Autor: Paula Chanfreut Palacio
Tutor: José María Maestre Torreblanca

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

A mi familia, por todo el cariño mostrado a lo largo de estos años, y en especial a mi madre, por haberme enseñado tanto y haber sido para mí un ejemplo de trabajo y esfuerzo. A mis amigos, los de dentro y fuera de la Universidad, porque sin ellos esto no sería posible.

Finalmente me gustaría agradecer la confianza, el apoyo y la ayuda durante este año a mi tutor Pepe Maestre, que ha permitido la realización de este proyecto.

Paula Chanfreut Palacio
Sevilla, 2017

Resumen

Este trabajo se enmarca dentro del campo del control predictivo basado en modelo (MPC, por sus siglas en inglés), con especial enfásis en problemas de control distribuido (DMPC).

El proyecto está orientado a sistemas compuestos por múltiples subsistemas con dinámicas acopladas interactuando entre sí, y que a su vez están controlados en base al MPC. El objetivo será la evolución óptima del sistema global operando de manera descentralizada mediante la incorporación del algoritmo de control propuesto en [1]. De esta manera, el comienzo de este trabajo es el estudio e implementación de dicho algoritmo, el cual es analizado en primer lugar en condiciones estándares de funcionamiento.

Con ello, se procede a considerar la posible presencia de agentes maliciosos en el sistema dispuestos a inyectar información que pueda comprometer su evolución, lo que genera una importante brecha de seguridad. Es en esta última situación en la que el trabajo hace especial hincapié, de ahí el notable enfoque hacia problemas relacionados con ciberseguridad.

El estudio de los ataques que puede sufrir el algoritmo comienza con la presentación de distintas posibilidades con las que cuenta un agente malicioso durante el desarrollo del mismo para introducir información falsa, así como el mecanismo por el cual ésta es extendida por el sistema. Igualmente, se expone cómo un agente dispuesto a atacar puede optimizar algunas de dichas posibilidades para lograr un mayor grado de aprovechamiento.

Finalmente, se introduce brevemente la técnica denominada min-max con el fin de desarrollar un mecanismo para la reacción ante dichos ataques, de manera que se mitiguen los problemas derivados de éstos.

La exposición teórica de lo aquí comentado está sucedida por una serie de simulaciones que permitirán probar los resultados presentados analíticamente.

Abstract

This project falls within the field of Model Predictive Control (MPC), with particular emphasis in distributed control problems (DMPC).

This work is geared towards systems composed of several subsystems with coupled dynamics which interact with each other, and whose control is based on MPC. The goal in the first instance will be the control of the overall system acting on a decentralized basis through the application of the algorithm presented in [1]. Thus, this project begins with the study and implementation of the mentioned algorithm, which is first analyzed in standard operating conditions.

Therewith, we proceed to consider the possible presence of malicious agents in the system that are willing to inject information which might prejudice its evolution. The latter results in an important security breach which is highlighted in the development of this project, hence its approach to problems related to cybersecurity.

The study of the attacks that the algorithm may experience starts with a presentation of different possibilities available to a malicious agent for introducing false information during the development of it. Furthermore, the optimization of some of these possibilities in order to reach a higher level of effectiveness from the point of view of the attacker is also discussed.

Finally, the min-max approach is briefly introduced with the purpose of coping with the attacks, in such a way that it mitigates the problems derived from them.

The theoretical presentation is followed by the corresponding simulations with a view to prove the analytical results.

Contents

1 Introduction

1.1 Motivation of the project

This project focuses on the control of networked systems based on MPC. The objective will be addressed in a distributed manner such that the systemwide control goals are intended to be attained by the actions of several MPCs subsystems. For this reason, a division of the global system (or plant) into a set of components that will be in charge of different controllers (or agents) will be considered. The increasing pressence of large-scale networked systems in which control plays a very important role gives the distributed approach special relevance nowadays.

Amongst the first aims of this work is the presentation and study of an algorithm that enable us to deal with the problem as described. The algorithm used here is presented in [1] and proposes a cooperative distributed framework which considers the couplings between the subsystems in which the plant has been divided. Hence, the dependence and interaction between subsystems take particular importance along the development of this project.

Once the algorithm has been introduced, the objective will be the enhancement of the mentioned plant's performance applying the latter. This leads to the definition of a standard case with which the normal perfomance can be assessed. However, one of the main motivations of this work is not controlling under reliable circumstances but analysing different possibilities for malicious controllers to introduce false information in order to alter the standard performance in favour of themselves. It will be shown that there are various alternatives for acting locally and steering the global evolution towards a new situation, hence the good use of them supposes clear opportunities to be taken by the mentioned malicious agents. This leads to problems associated with cibersecurity when the algorithm is implemented and whose study is objective of this work. To this end, the theoretical presentation will be followed by the simulation of the alternatives on given examples.

Before continuing, we are going to present briefly the principle elements of MPC which will be used in this project.

1.2 A brief overview of MPC

Model predictive control (MPC) is a control strategy which is framed within the field of optimal controllers and whose inherent features have made it increasingly important. An MPC controller can deal with multiple inputs and outputs, nonlinear dynamics, multiple objectives and also copes with the corresponding constraints on the system state and input. The advantages it offers have provided it with a great success in both industry applications and research. Today's situation and knowledge in the field is consequence of having undergone an evolution in which improvements have been added and in which the possibilities that it offers have been extended.

Although it has been referred to as a single control strategy, MPC encloses a set of techniques which share certain commonalities. The basic elements used when applied MPC strategy are introduced below.

1.2.1 Prediction model

The prediction models are the mathematical expressions used in the application of the strategy to describe the future behaviour of the systems. For that reason, a good design should lead to those models that consider,

as far as possible, the dynamic characteristics of the process, as well as ensuing an useful tool for the analysis of the control problem.

The application field of MPC includes, as it has been mentioned, linear and non-linear systems that are modelled with the purpose of obtaining predicted information over a finite time horizon. This time horizon is denoted as prediction horizon and will be from now on represented by N. Despite the fact that in this project a state space representation will be used, other model structures, such as input-output representation, fall also into the mentioned technique. A representation in the state space can be seen as

$$\begin{aligned} x(k+1) &= f(x(k),u(k)) \\ y(k) &= g(x(k),u(k)) \end{aligned} \tag{1.1}$$

where the letters x, u and y indicate respectively the state, input and output of the system. In addition, the course of time will be reflected in the time index k.

1.2.2 Cost function

The cost function defines the optimization problem that is held each sample time and, which in turn determines the decision variables for a subsequent receding horizon implementation.

The application of MPC technique implies the resolution of a certain number of optimization problems with respect to the manipulating variables of the system at issue.

The objective functions that will appear in this project are defined as quadratic expressions that weigh the state error with respect to a predefined reference, as well as the input vector, both over a determined N.

The letter ϕ has been chosen to represent cost functions along this work.

$$\phi(\mathbf{x}(k),\mathbf{u}(k),k) = \sum_{n=0}^{N-1} \left[\left(x(k+n) - x_{ref}\right)^T Q(k+n) \left(x(k+n) - x_{ref}\right) + u(k+n)^T R(k+n) u(k+n) \right] \tag{1.2}$$

As a consequence of using the regulation of the system to the origin as control objective in many of the suceeding parts, the explicit dependence of the reference will not be shown, without loss of generality.

Hereon, when applying cost functions with this structure, the corresponding matrices Q and R are assumed to be such as $Q(k+n) > 0, R(k+n) > 0$, for all $n = 0,1,...,N-1$. The latter conditions imply that we will deal with convex positive definite functions.

1.2.3 Control Law and receding horizon implementation

The MPC control law denotes the solution $\mathbf{u}^*(\mathbf{k})$ that is arrived at when solving the optimization problem at time index k.

$$\mathbf{u}^*(k) = \arg\min_u \phi(\mathbf{x}(k),\mathbf{u}(k)) \tag{1.3}$$

Therefore, it represents a sequence of inputs calculated for the next sample times. In this project this number of sample times is again N, so that, per each k the control law can be repesented as (1.4).

$$u^*(k) = \begin{bmatrix} u^*(k|k) \\ u^*(k+1|k) \\ \vdots \\ u^*(k+N-1|k) \end{bmatrix} \tag{1.4}$$

A commonn characteristic of particular importance of MPC strategies is the application of a receding horizon implementation. This means that per instant time k only the first input of the solution in (1.3) is implemented. The receding horizon implementation, whose graphical representation can be seen in Figure 1.1, will be applied each time step.

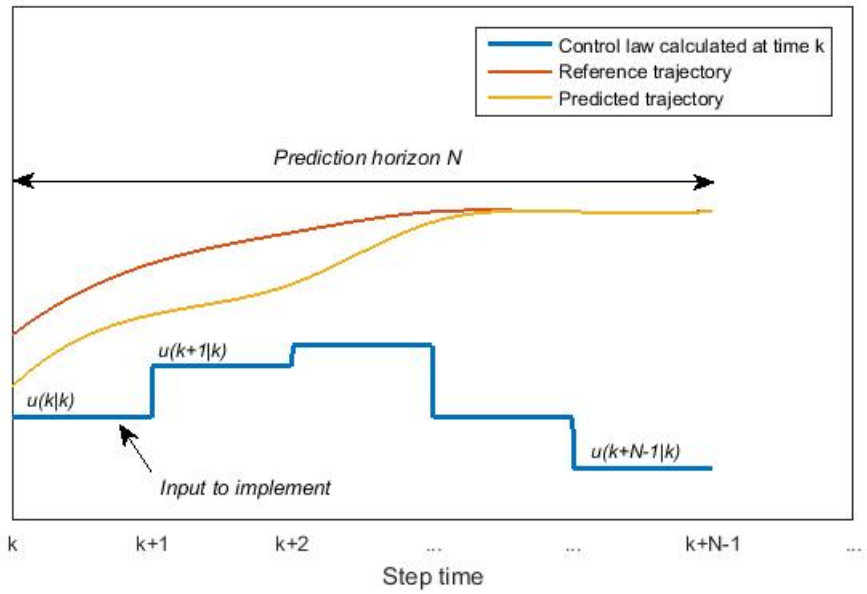

Figure 1.1 Scheme of receding horizon implementation.

1.2.4 Constraints

The optimization problem to reach $\mathbf{u}^*(k)$ will be done taking into account the current and future constraints for the N next sample times. In this work, the constraints will be imposed on the state and inputs trajectories calculated, so that the implementation of $\mathbf{u}^*(k)$ must lead to a sequence of states that complies with certain conditions. Let $\mathbf{x}^*(k)$ denote the resulting state trajectory, then the constraints can be expressed as

$$\begin{aligned} \mathbf{u}^*(k) &\in \mathscr{U} \\ \mathbf{x}^*(k) &\in \mathscr{X} \end{aligned} \tag{1.5}$$

where $\mathscr{U}$ and $\mathscr{X}$ are the sets that define them.

2 Model of the coupled system

In this project the plant/system is considered to comprise M subsystems, each of them controlled by its corresponding agent and denoted by the index i, where $i = 1,2,...M$.

One of the options to address the problem of controlling networked system is to use decentralized-models which assume insignificant the effects of the interactions (2.1).

$$\begin{aligned} x_{ii}(k+1) &= A_{ii}x_{ii}(k) + B_{ii}u_i(k) \\ y_i(k) &= C_{ii}x_{ii}(k) \end{aligned} \tag{2.1}$$

That is, a certain subsystem i is supposed not to be affected by the rest of subsystems j ($j \neq i$). Despite its computational simplicity, the reliability of this assumption, together with the lack of information that is shared, is questionable and may end up in a not suitable control performance. On the contrary, it is possible to define a discrete model to calculate the effects caused by interactions of a certain j on i ($j \neq i$). (2.2).

$$\begin{aligned} x_{ij}(k+1) &= A_{ij}x_j(k) + B_{ij}u_j(k) \\ y_i(k) &= \sum_{j=1}^{M} C_{ij}x_{ij}(k) \end{aligned} \tag{2.2}$$

A proposal to consider the information provided by both of them, is the mathematical model (2.3), which will be used hereafter to address the problem of distributed MPC. The purpose of the resulting combination is to deal with the couplings and possible exchange of variables between subsystems and taking account of them when calculating the optimal decisions. In short, (2.3) contains the effects of the interaction between the subsystem i and any other j in the plant, as well as the state information given by the decentralized model, expressing them as discrete LTI equations.

$$\begin{aligned} x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_{j=1, j\neq i}^{M} \left[A_{ij}x_j(k) + B_{ij}u_j(k)\right] \\ y_i(k) = C_i x_i(k) \\ i = 1,2,...,M. \end{aligned} \tag{2.3}$$

From now on, the number of components of the state x_i and input u_i for a certain i and all k will be denoted respectively as n_i and m_i. Therefore,

$$x_i \in \mathbb{R}^{n_i}, \qquad u_i \in \mathbb{R}^{m_i}$$

$$A_{il} \in \mathbb{R}^{n_i \times n_l}, \qquad B_{il} \in \mathbb{R}^{n_i \times m_l}, \qquad \forall l = 1,...,M$$

2.1 Extension of the model over a control horizon N

During the development of this project, and due to the application of MPC strategy, the use of vectors which represent predicted trajectories will be recurrent. For this reason, and basing the following reasoning in (2.3), the corresponding matrix representation when extending the model over a control horizon N is presented

(N has been supposed in the reasoning to be greater than 3). That is, the aim here is to arrive at a single expression which provides the predicted trajectory of x_i for the following N time steps.

The notation $(k+n|k)$ as time indicator has been used, in which the parameter on the right side, k, indicates the current discrete time and, therefore, the one from which current information is taken. The parameter on the left, $k+n$, where n is a positive number, specify the time index of the prediction, either for state or input.

Let's define $w_{ij}(k)$ as

$$w_i(k) = \sum_{j=1, j\neq i}^{M} \left[A_{ij}x_j(k) + B_{ij}u_j(k)\right] \tag{2.4}$$

for a simpler formulation of (2.3).

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k) + B_{ii}u_i(k) + w_i(k) \\ y_i(k) &= C_i x_i(k) \\ i &= 1,2,...,M. \end{aligned} \tag{2.5}$$

Making use of (2.5) with the aforementioned purpose, the way of proceeding for reaching the desired result is presented.

$$\begin{aligned}
x_i(k+1|k) =& A_{ii}x_i(k|k) + B_{ii}u_i(k|k) + w_i(k|k) \\
x_i(k+2|k) =& A_{ii}^2x_i(k|k) + A_{ii}B_iu_i(k|k) + A_{ii}w_i(k|k) + B_{ii}u_i(k+1|k) + w_i(k+1|k) \\
x_i(k+3|k) =& A_{ii}^3x_i(k|k) + A_{ii}^2B_{ii}u_i(k|k) + A_{ii}^2w_i(k|k) + A_{ii}B_{ii}u_i(k+1|k) + A_{ii}w_i(k+1|k) \\
&+ B_{ii}u_i(k+2|k) + w_i(k+2|k) \\
&\vdots \\
x_i(k+N|k) =& A_{ii}^Nx_i(k|k) + A_{ii}^{N-1}B_{ii}u_i(k|k) + A_{ii}^{N-1}w_i(k|k) + A_{ii}^{N-2}B_{ii}u_i(k+1|k) + A_{ii}^{N-2}w_i(k+1|k) \\
&+ A_{ii}^{N-3}B_{ii}u_i(k+2|k) + A_{ii}^{N-3}w_i(k+2|k) + ... + A_{ii}^{N-N}B_{ii}u_i(k+N-1|k) + A_{ii}^{N-N}w_i(k+N-1|k)
\end{aligned}$$

$$\begin{aligned}
\begin{bmatrix} x_i(k+1|k) \\ x_i(k+2|k) \\ \vdots \\ x_i(k+N|k) \end{bmatrix} =& \begin{bmatrix} A_{ii} \\ A_{ii}^2 \\ \vdots \\ A_{ii}^N \end{bmatrix} x_i(k|k) + \begin{bmatrix} B_{ii} & & & \\ A_{ii}B_{ii} & B_{ii} & & \\ \vdots & & \ddots & \\ A_{ii}^{N-1}B_{ii} & \cdots & \cdots & B_{ii} \end{bmatrix} \begin{bmatrix} u_i(k|k) \\ u_i(k+1|k) \\ \vdots \\ u_i(k+N-1|k) \end{bmatrix} + \\
& \begin{bmatrix} I & & & \\ A_{ii} & I & & \\ \vdots & & \ddots & \\ A_{ii}^{N-1} & \cdots & \cdots & I \end{bmatrix} \begin{bmatrix} w_i(k|k) \\ w_i(k+1|k) \\ \vdots \\ w_i(k+N-1|k) \end{bmatrix}
\end{aligned} \tag{2.6}$$

Similarly, the trajectory $\mathbf{w}_i(k)$ can be expressed, as consequence of its defintion, in terms of the states' and inputs' variables related to every subsystem $j \neq i$. Concretely,

$$\begin{bmatrix} w_i(k|k) \\ w_i(k+1|k) \\ \vdots \\ w_i(k+N-1|k) \end{bmatrix} = \sum_{j=1, j\neq 1}^{M} \left(\begin{bmatrix} A_{ij} \\ A_{ij}^2 \\ \vdots \\ A_{ij}^N \end{bmatrix} x_j(k|k) + \begin{bmatrix} B_{ij} & & & \\ A_{ij}B_{ij} & B_{ij} & & \\ \vdots & & \ddots & \\ A_{ij}^{N-1}B_{ij} & \cdots & \cdots & B_{ij} \end{bmatrix} \begin{bmatrix} u_j(k|k) \\ u_j(k+1|k) \\ \vdots \\ u_j(k+N-1|k) \end{bmatrix} \right) \tag{2.7}$$

Therefore, result (2.6) equates to

$$\begin{bmatrix} x_i(k+1|k) \\ x_i(k+2|k) \\ \vdots \\ x_i(k+N|k) \end{bmatrix} = \begin{bmatrix} A_{ii} \\ A_{ii}^2 \\ \vdots \\ A_{ii}^N \end{bmatrix} x_i(k|k) + \begin{bmatrix} B_{ii} & & & \\ A_{ii}B_{ii} & B_{ii} & & \\ \vdots & & \ddots & \\ A_{ii}^{N-1}B_{ii} & \cdots & \cdots & B_{ii} \end{bmatrix} \begin{bmatrix} u_i(k|k) \\ u_i(k+1|k) \\ \vdots \\ u_i(k+N-1|k) \end{bmatrix}$$
$$+ \sum_{j=1, j\neq i}^{M} \begin{bmatrix} I & & & \\ A_{ii} & I & & \\ \vdots & & \ddots & \\ A_{ii}^{N-1} & \cdots & \cdots & I \end{bmatrix} \begin{bmatrix} A_{ij} \\ A_{ij}^2 \\ \vdots \\ A_{ij}^N \end{bmatrix} x_j(k|k)$$
$$+ \sum_{j=1, j\neq i}^{M} \begin{bmatrix} I & & & \\ A_{ii} & I & & \\ \vdots & & \ddots & \\ A_{ii}^{N-1} & \cdots & \cdots & I \end{bmatrix} \begin{bmatrix} B_{ij} & & & \\ A_{ij}B_{ij} & B_{ij} & & \\ \vdots & & \ddots & \\ A_{ij}^{N-1}B_{ij} & \cdots & \cdots & B_{ij} \end{bmatrix} \begin{bmatrix} u_j(k|k) \\ u_j(k+1|k) \\ \vdots \\ u_j(k+N-1|k) \end{bmatrix}$$

Definitions:

$$G_{xi} = \begin{bmatrix} A_{ii} \\ A_{ii}^2 \\ \vdots \\ A_{ii}^N \end{bmatrix}, \qquad G_{ui} = \begin{bmatrix} B_{ii} & & & \\ A_{ii}B_{ii} & B_{ii} & & \\ \vdots & & \ddots & \\ A_{ii}^{N-1}B_{ii} & \cdots & \cdots & B_{ii} \end{bmatrix}, \qquad G_{wi} = \begin{bmatrix} I & & & \\ A_{ii} & I & & \\ \vdots & & \ddots & \\ A_{ii}^{N-1} & \cdots & \cdots & I \end{bmatrix}$$

$$G_{wi}^{x_j} = \begin{bmatrix} I & & & \\ A_{ii} & I & & \\ \vdots & & \ddots & \\ A_{ii}^{N-1} & \cdots & \cdots & I \end{bmatrix} \begin{bmatrix} A_{ij} \\ A_{ij}^2 \\ \vdots \\ A_{ij}^N \end{bmatrix}, \qquad G_{wi}^{u_j} = \begin{bmatrix} I & & & \\ A_{ii} & I & & \\ \vdots & & \ddots & \\ A_{ii}^{N-1} & \cdots & \cdots & I \end{bmatrix} \begin{bmatrix} B_{ij} & & & \\ A_{ij}B_{ij} & B_{ij} & & \\ \vdots & & \ddots & \\ A_{ij}^{N-1}B_{ij} & \cdots & \cdots & B_{ij} \end{bmatrix}$$

$$\mathbf{x}_i(k) = \begin{bmatrix} x_i(k|k) \\ x_i(k+1|k) \\ \vdots \\ x_i(k+N-1|k) \end{bmatrix}, \qquad \mathbf{u}_i(k) = \begin{bmatrix} u_i(k|k) \\ u_i(k+1|k) \\ \vdots \\ u_i(k+N-1|k) \end{bmatrix}, \qquad \mathbf{w}_i(k) = \begin{bmatrix} w_i(k|k) \\ w_i(k+1|k) \\ \vdots \\ w_i(k+N-1|k) \end{bmatrix}$$

Result:

The expressions in (2.8) represent the matricial form which arise when extending the subsystem's model in (2.3), for some i, through a determined time horizon N.

$$\begin{aligned} \mathbf{x}_i(k+1) &= G_{xi}x_i(k|k) + G_{ui}\mathbf{u}_i(k) + G_{wi}\mathbf{w}_i(k) \\ \mathbf{x}_i(k+1) &= G_{xi}x_i(k|k) + G_{ui}\mathbf{u}_i(k) + \sum_{j=1, j\neq i}^{M} \left[G_{wi}^{x_j}x_j(k|k) + G_{wi}^{u_j}\mathbf{u}_j(k) \right] \end{aligned} \tag{2.8}$$

2.2 Centralized model based on (2.3).

As well as using the model to reach a prediction for the following N steps, it may be also be used with the goal of designing a model in which all the subsystems equations are involved. To this end, the components of the state and input vector are defined to be the sequence of every x_i and u_i for $i = 1, 2, ...M$. After application of (2.3) to every i using the previous definitions, a single matrix framework is reached for the entire plant (2.9).The result we get to is a centralized representation which implicitly assumes all the subsystems' models.

The explicit dependence of the index i will, as consequence, be lost. The corresponding matrices A_{cen}, B_{cen} and C_{cen} are the ones shown below.

$$\underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}}_{x(k+1)}(k+1) = \underbrace{\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1M} \\ A_{21} & A_{22} & \cdots & A_{2M} \\ \vdots & & \ddots & \\ A_{M1} & A_{M2} & \cdots & A_{MM} \end{bmatrix}}_{A_{cen}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}}_{x(k)}(k) + \underbrace{\begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & & \ddots & \\ B_{M1} & B_{M2} & \cdots & B_{MM} \end{bmatrix}}_{B_{cen}} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_M \end{bmatrix}}_{u(k)}$$

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}}_{y(k)}(k) = \underbrace{\begin{bmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_M \end{bmatrix}}_{C_{cen}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}}_{x(k)}(k)$$

$$\rightarrow x(k+1) = A_{cen}x(k) + B_{cen}u(k), \qquad y(k) = C_{cen}x(k) \tag{2.9}$$

where

$$x \in \mathbb{R}^{\Sigma_i n_i}, \qquad u \in \mathbb{R}^{\Sigma_i m_i}, \qquad A_{cen} \in \mathbb{R}^{\Sigma_i n_i \times \Sigma_i ni}, \qquad B_{cen} \in \mathbb{R}^{\Sigma_i n_i \times \Sigma_i mi}$$

3 From communication-based MPC to cooperation-based MPC

The integration of the appointed M subsystems in the entire plant implies that the overall behavior will be consequence of the decisions taken by each of the agents ($i = 1,2,...,M$). Different formulations have been developed in order to address this issue, with a certain idea behind and a goal. Moreover, the definition of the controls is directly related to the assessment of the interactions in the interconected system we are dealing with.

3.1 Communication-based MPC

First of all, it may be a possibility to solve the problem under the assumption that each subsystem's MPC only knows its own cost function, which is presented in (3.1) and which supposes a weighted sum of the state and input's components. In addition, at any iteration, the information about the state and input trajectories of the rest of the subsystems at the previous iteration will be provided, and used for the corresponding update equations. In other words, if we denote the current iteration number as p, we have that every i at p will receive every $\mathbf{x}_{j\neq i}^{p-1}(k)$,$\mathbf{u}_{j\neq i}^{p-1}(k)$.

With that conditions applied parallelly for every i integrated, what is proposed in communication-based MPC is the individual resolution of the optimization problem (3.2) and the succeeding implementation of the optimal inputs calculated. It is clearly observable that the assumption taken by all i that every $j \neq i$ remain at iteration $p-1$ is maintained during the entire time length, so preceding information is always used in the optimizations. These assumptions, along with the fact that any of them knows how their decision will affect the others, and therefore the overall performance, induce the risk of ending up in disagreements between subsystems objectives. In the case that it leads up to an equilibrium, we will reach the named non-cooperative or Nash equilibrium. The latter is characterized by the fact that when it is reached, no single subsystem's MPC will improve its cost function if they decide to deviate from it. That is, the inputs calculated give each of them their best local situations.

Cost function communication-based MPC:

$$\begin{aligned}&\phi_i(\mathbf{x}_i(k),\mathbf{u}_i(k),\mathbf{x}_{j\neq i}^{p-1}(k),\mathbf{u}_{j\neq i}^{p-1}(k);x_i(k|k)) = \\ &\sum_{n=0}^{N-1} x_i^T(k+n|k)Q_i(k+n|k)x_i(k+n|k) + u_i^T(k+n|k)R_i(k+n|k)u_i(k+n|k)\end{aligned} \tag{3.1}$$

Optimization problem:

$$\mathbf{u}_{i,opt} = \arg \min_{\mathbf{u}_i} \ \phi_i(\mathbf{x}_i(k),\mathbf{u}_i(k),\mathbf{x}_{j\neq i}^{p-1}(k),\mathbf{u}_{j\neq i}^{p-1}(k);x_i(k|k)) \tag{3.2}$$

subject to the following constraints

$$\begin{aligned} x_i(k+n+1|k) = A_i x_i(k+n|k) + B_i u_i(k+n|k) + \sum_{j\neq i}\left[A_{ij}x_j^{p-1}(k+n|k) + B_{ij}u_j^{p-1}(k+n|k)\right] \\ u_i(k+n|k) \in \mathscr{U}_i, \;\; k \leq k+n \leq k+N-1 \\ u_i(k+n|k) = 0, \;\; n \geq N \\ i \in \{1,M\}. \end{aligned} \tag{3.3}$$

where $\mathscr{U}_i$ is the set of admissible control decisions for subsystem i. The omission of the superscript related to the iteration implies that it is a variable associated to p, either defined or to be defined. Due to the fact that the objective is the control of the plant, some disadvantages can be found in the use of the communication-based technique. This way of proceeding gives rise to the possibility of not taking the best decision for the plant, despite being the best by the point of view of each i. At this point, we should introduce the concept of cooperation.

3.2 Cooperation-based MPC

With the objective of finding the optimal achievable performance considering the aforementioned conflicts that may occur between subsystem's objectives, formulations which propose acting by means of cooperation have been developed. To characterize this concept, we have the so-called Pareto surface, which supposes an important term in simultaneous optimization of several objective functions. In a Pareto solution, we have that the cost of one i cannot be improved without affecting negatively the cost of one $j \neq i$. Mathematically, it could be possible to address the plantwide problem modifying the cost function to optimize by the agents. In that new formulation, it will be important to consider the influence of the decision taken by one subsystem to the rest of them, as well as their predicted evolution in the iteration before. The optimization problem to solve in the so-called feasible cooperation-based MPC can be written as the weighted sum in (3.4), in which preceding information will again be assumed but, in this case, the objective will assess the entire plant.

$$\phi_{i,c}(\mathbf{u}_i(k), \mathbf{u}_{j\neq i}^{p-1}(k); x(k|k)) = \sum_{l=1}^{M} \lambda_l \phi_l(\mathbf{u}_i(k), \mathbf{u}_{j\neq i}^{p-1}(k); x_l(k|k)) \tag{3.4}$$

where ϕ_l represents the function defined in (3.1) for agent l, and λ_l the weighting factor. The letter c is referred here to the term cooperation-based and it has been introduced to differentiate the entire summatory that define the function to optimize and $\phi_i(\cdot)$ presented above. Moreover, the dependance of the state's trajectory is avoided using the expression (2.8) as shown later.

Table 3.1 Summary (Communication-/Cooperation- based MPC).

	Agent i problem
Communication-based MPC	$\min_{\mathbf{u}_i} \;\; \phi_i(\mathbf{u}_i, \mathbf{u}_{j\neq i}^{p-1}; x_i(k))$
Cooperation-based MPC	$\min_{\mathbf{u}_i} \;\; \sum_{l=1}^{M} \lambda_l \phi_l(\mathbf{u}_i, \mathbf{u}_{j\neq i}^{p-1}; x_l(k))$

After determined transformations, the entire summation indicated in $\phi_{i,c}(\cdot)$ can be simplified without affecting the optimization's solution, so that the expression in (3.5) is reached as proved below. Given that the optimization problem in (3.4) equals the one in (3.5) it can therefore be used in the corresponding implementation.

$$\min_{\mathbf{u}_i} \qquad \frac{1}{2}\mathbf{u}_i(k)^T \mathscr{H}_i \mathbf{u}_i(k) + \left(r_i(\mathbf{u}_{j\neq i}^{p-1}(k)) + q_i(x(k|k))\right)^T \mathbf{u}_i(k) \tag{3.5}$$

Proof of (3.5):
1) Removal of the dependance of $\mathbf{x}_i$ in $\phi_i(\cdot)(i = 1,..,M)$.

$$\phi_i(\mathbf{u}_i, \mathbf{u}_{j\neq i}^{p-1}; x_i(k)) = \left(G_{xi}x_i(k|k) + G_{ui}\mathbf{u}_i(k) + G_{wi}\mathbf{w}_i^{p-1}(k)\right)^T \widehat{Q}_i \left(G_{xi}x_i(k|k) + G_{ui}\mathbf{u}_i(k) + G_{wi}\mathbf{w}_i^{p-1}(k)\right)$$

$$+\mathbf{u}_i^T(k)\widehat{R}_i\mathbf{u}_i(k)+cte$$

2) Introduction of the result in 1) for each i in the sumatory (3.4)

$$\lambda_i\left(G_{xi}x_i(k|k)+G_{ui}\mathbf{u}_i(k)+G_{wi}\mathbf{w}_i(k)\right)^T\widehat{Q}_i\left(G_{xi}x_i(k|k)+G_{ui}\mathbf{u}_i(k)+G_{wi}\mathbf{w}_i(k)\right)+\lambda_i\mathbf{u}_i^T(k)\widehat{R}_i\mathbf{u}_i(k)$$
$$+\sum_{l=1,l\neq i}^{M}\lambda_l\left(G_{xl}x_l(k|k)+G_{ul}\mathbf{u}_l^{p-1}(k)+\sum_{j\neq l,j\neq i}^{M}\left[G_{wl}^{x_j}x_j(k|k)+G_{wl}^{u_j}\mathbf{u}_j^{p-1}(k)\right]+G_{wl}^{x_i}x_i(k|k)+G_{wl}^{u_i}\mathbf{u}_i(k)\right)^T$$
$$\widehat{Q}_l\left(G_{xl}x_l(k|k)+G_{ul}\mathbf{u}_l^{p-1}(k)+\sum_{j\neq l,j\neq i}^{M}\left[G_{wl}^{x_j}x_j(k|k)+G_{wl}^{u_j}\mathbf{u}_j^{p-1}(k)\right]+G_{wl}^{x_i}x_i(k|k)+G_{wl}^{u_i}\mathbf{u}_i(k)\right)$$
$$+\sum_{l=1,l\neq i}^{M}\lambda_l\mathbf{u}_l^{p-1}(k)^T\widehat{R}_l\mathbf{u}_l^{p-1}(k)+cte$$

3) Avoidance of the terms that are independent of $\mathbf{u}_i(k)$ and definition of the problem.

$$\begin{aligned}\min_{\mathbf{u}_i}\ &\lambda_i\mathbf{u}_i^T\left(G_{ui}^T\widehat{Q}_iG_{ui}+\widehat{R}_i\right)\mathbf{u}_i+\lambda_i\left(2x_i(k|k)^TG_{xi}^T\widehat{Q}_iG_{ui}+2\mathbf{w}_i(k)^TG_{wi}^T\widehat{Q}_iG_{ui}\right)\mathbf{u}_i\\
&+\sum_{l=1,l\neq i}^{M}\lambda_l\mathbf{u}_i(k)\left(G_{wl}^{u_i\,T}\widehat{Q}_lG_{wl}^{u_i}+\widehat{R}_l\right)\mathbf{u}_i(k)\\
&+\sum_{l=1,l\neq i}^{M}\lambda_l2\left(x_l(k|k)^TG_{xl}^T\widehat{Q}_lG_{wl}^{u_i}+\mathbf{u}_l^{p-1}(k)^TG_{ul}^T\widehat{Q}_lG_{wl}^{u_i}+\sum_{j\neq l,j\neq i}^{M}\left[G_{wl}^{x_j}x_j(k|k)+G_{wl}^{u_j}\mathbf{u}_j^{p-1}(k)\right]^T\widehat{Q}_lG_{wl}^{u_i}\right)\mathbf{u}_i(k)\\
&+\sum_{l=1,l\neq i}^{M}\lambda_l2\left(x_i(k|k)^TG_{wl}^{x_i\,T}\widehat{Q}_lG_{wl}^{u_i}\right)\mathbf{u}_i(k)\end{aligned}$$

4) Regrouping of the terms and division by two. Final expression for the desired optimization reached.

$$\begin{aligned}\min_{\mathbf{u}_i}\quad&\frac{1}{2}\mathbf{u}_i^T\left(\lambda_i\left[G_{ui}^T\widehat{Q}_iG_{ui}+\widehat{R}_i\right]+\sum_{l=1,l\neq i}^{M}\lambda_l\left[G_{wl}^{u_i\,T}\widehat{Q}_lG_{wl}^{u_i}+\widehat{R}_l\right]\right)\mathbf{u}_i\\
&+\lambda_i\left(x_i(k|k)^TG_{xi}^T\widehat{Q}_iG_{ui}+\sum_{j\neq i}^{M}\left[G_{wi}^{x_j}x_j(k|k)+G_{wi}^{u_j}\mathbf{u}_j^{p-1}(k)\right]^T\widehat{Q}_iG_{ui}\right)\mathbf{u}_i\\
&+\sum_{l=1,l\neq i}^{M}\lambda_l\left(\mathbf{u}_l^{p-1}(k)^TG_{ul}^T\widehat{Q}_lG_{wl}^{u_i}+\sum_{j\neq l,j\neq i}^{M}\mathbf{u}_j^{p-1}(k)^TG_{wl}^{u_j\,T}\widehat{Q}_lG_{wl}^{u_i}\right)\mathbf{u}_i(k)\\
&+\sum_{l=1,l\neq i}^{M}\lambda_l\left(x_l(k|k)^TG_{xl}^T\widehat{Q}_lG_{wl}^{u_i}+\sum_{j\neq l}^{M}x_j(k|k)^TG_{wl}^{x_j\,T}\widehat{Q}_lG_{wl}^{u_i}\right)\mathbf{u}_i(k)\end{aligned}$$

$$\begin{aligned}\min_{\mathbf{u}_i}\quad&\frac{1}{2}\mathbf{u}_i^T\left(\lambda_i\left[G_{ui}^T\widehat{Q}_iG_{ui}+\widehat{R}_i\right]+\sum_{l=1,l\neq i}^{M}\lambda_l\left[G_{wl}^{u_i\,T}\widehat{Q}_lG_{wl}^{u_i}+\widehat{R}_l\right]\right)\mathbf{u}_i\\
&+\left(\lambda_i\left[\sum_{j\neq i}^{M}G_{wi}^{u_j}\mathbf{u}_j^{p-1}(k)\right]^T\widehat{Q}_iG_{ui}+\sum_{l=1,l\neq i}^{M}\lambda_l\left[G_{ul}\mathbf{u}_l^{p-1}(k)+\sum_{j\neq l,j\neq i}^{M}G_{wl}^{u_j}\mathbf{u}_j^{p-1}(k)\right]^T\widehat{Q}_lG_{wl}^{u_i}\right)\mathbf{u}_i\\
&+\left(\lambda_i\left[G_{xi}x_i(k|k)+\sum_{j\neq i}^{M}G_{wi}^{x_j}x_j(k|k)\right]^T\widehat{Q}_iG_{ui}+\sum_{l=1,l\neq i}^{M}\lambda_l\left[G_{xl}x_l(k|k)+\sum_{j\neq l}^{M}G_{wl}^{x_j}x_j(k|k)\right]^T\widehat{Q}_lG_{wl}^{u_i}\right)\mathbf{u}_i(k)\end{aligned}$$

Definitions:

$$r_i(\mathbf{u}_{j\neq i}^{p-1}(k)) = \lambda_i \left[\sum_{j\neq i}^{M} G_{wi}^{u_j}\mathbf{u}_j^{p-1}(k)\right]^T \widehat{Q}_i G_{ui} + \sum_{l=1,l\neq i}^{M} \lambda_l \left[G_{ul}\mathbf{u}_l^{p-1}(k) + \sum_{j\neq l, j\neq i}^{M} G_{wl}^{u_j}\mathbf{u}_j^{p-1}(k)\right]^T \widehat{Q}_l G_{wl}^{u_i}$$

$$q_i(x(k|k)) = \lambda_i \left[G_{xi}x_i(k|k) + \sum_{j\neq i}^{M} G_{wi}^{x_j}x_j(k|k)\right]^T \widehat{Q}_i G_{ui} + \sum_{l=1,l\neq i}^{M} \lambda_l \left[G_{xl}x_l(k|k) + \sum_{j\neq l}^{M} G_{wl}^{x_j}x_j(k|k)\right]^T \widehat{Q}_l G_{wl}^{u_i}$$

$$\mathscr{H}_i = \lambda_i \left[G_{ui}^T \widehat{Q}_i G_{ui} + \widehat{R}_i\right] + \sum_{l=1,l\neq i}^{M} \lambda_l \left[{G_{wl}^{u_i}}^T \widehat{Q}_l G_{wl}^{u_i} + \widehat{R}_l\right]$$

We have to underline here that the previous reasoning has introduced an additional penalty by a ponderation of the state's prediction at time $k+N$. This term is not included in (3.1), but to come up with the result above each matrix $\widehat{Q}_i$ has been defined as follows. This introduction should not disrupt the convexity of the optimization problem, so that the choice of $Q_i(k+N)$ must be done in accordance with a criterion that ensures it.

$$\widehat{Q}_i = \begin{bmatrix} Q_i(k+1) & & & \\ & Q_i(k+2) & & \\ & & \ddots & \\ & & & Q_i(k+N) \end{bmatrix} \qquad \widehat{R}_i = \begin{bmatrix} R_i(k) & & & \\ & R_i(k+1) & & \\ & & \ddots & \\ & & & R_i(k+N-1) \end{bmatrix}$$

4 Controller design procedure

To solve the feasible cooperation-based MPC problem, we can use the result in (3.5) to calculate the optimal input for each agent at iteration p. In the corresponding algorithm, a certain maximum of iterations and a stability condition will be applied to determine the decision variables to implement per step. In other words, each sample time, the problem (3.5) will be solved iteratively, together with the update of the state and input trajectories at every p, until convergence is reached or until a fixed p_{max}. The last control law arrived will define the controls for each subsystem. Here, a slightly different way of resolution have been chosen, but with the same purpose and fundamentals, in a manner that the same approach of distributed control will be implemented.

4.1 Centralized model

Firstly, we draw from the centralized model of the plant, whose equations can be seen in (2.9). Proceeding as in the reasoning (2.1), but using, in this case, the equations of the centralized model, we get to a matrix trajectory framework for N steps. Then, introducing this result into the centralized cost function, expressed equally in a matrix way, the optimization problem for the centralized case is defined.

$$\min_{\mathbf{u}(k)} \phi(\mathbf{x},\mathbf{u};x(k|k)) = \frac{1}{2}\mathbf{u}(k)^T H\mathbf{u}(k) + F^T\mathbf{u}(k) \tag{4.1}$$

$$H = G_u^T \widehat{Q} G_u + \widehat{R}$$
$$F = G_u^T \widehat{Q} G_x x(k|k)$$

Proof of (4.1)

1) Extension of (2.9) over a control time horizon N.

$$\begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \vdots \\ x(k+N|k) \end{bmatrix} = \begin{bmatrix} A_{cen} \\ A_{cen}^2 \\ \vdots \\ A_{cen}^N \end{bmatrix} x(k|k) + \begin{bmatrix} B_{cen} & & & \\ A_{cen}B_{cen} & B_{cen} & & \\ \vdots & & \ddots & \\ A_{cen}^{N-1}B_{cen} & \cdots & \cdots & B_{cen} \end{bmatrix} \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix} \tag{4.2}$$

Definitions:

$$\mathbf{x}(k+1) = \begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \vdots \\ x(k+N|k) \end{bmatrix}, G_x = \begin{bmatrix} A_{cen} \\ A_{cen}^2 \\ \vdots \\ A_{cen}^N \end{bmatrix}, G_u = \begin{bmatrix} B_{cen} & & & \\ A_{cen}B_{cen} & B_{cen} & & \\ \vdots & & \ddots & \\ A_{cen}^{N-1}B_{cen} & \cdots & \cdots & B_{cen} \end{bmatrix}, \mathbf{u}(k) = \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix}$$

2) Cost function (expressed matricially)

$$\phi(\mathbf{x},\mathbf{u};x(k|k)) = \mathbf{x}^T(k+1)\widehat{Q}\mathbf{x}(k+1) + \mathbf{u}^T(k)\widehat{R}\mathbf{u}(k)$$

Definitions:

$$\widehat{Q} = \begin{bmatrix} Q(k+1) & & & \\ & Q(k+2) & & \\ & & \ddots & \\ & & & Q(k+N) \end{bmatrix}, \qquad \widehat{R} = \begin{bmatrix} R(k) & & & \\ & R(k+1) & & \\ & & \ddots & \\ & & & R(k+N-1) \end{bmatrix}$$

$$Q(k) = \begin{bmatrix} Q_1(k) & & & \\ & Q_2(k) & & \\ & & \ddots & \\ & & & Q_M(k) \end{bmatrix}, \qquad R(k) = \begin{bmatrix} R_1(k) & & & \\ & R_2(k) & & \\ & & \ddots & \\ & & & R_M(k) \end{bmatrix}$$

3) Introduction of the step 1) into 2).

$$(G_x x(k|k) + G_u \mathbf{u}(k))^T \widehat{Q}(G_x x(k|k) + G_u \mathbf{u}(k)) + \mathbf{u}^T(k)\widehat{R}\mathbf{u}(k)$$

4) Desired optimization problem definition (after grouping terms)

$$\min_{\mathbf{u}(k)} \phi(\mathbf{u}(k); x(k|k)) = \frac{1}{2}\mathbf{u}(k)^T \left(G_u^T \widehat{Q} G_u + \widehat{R}\right)\mathbf{u}(k) + \left(x(k|k)^T G_x^T \widehat{Q} G_u\right)\mathbf{u}(k)$$

$$\rightarrow \min_{\mathbf{u}(k)} \phi(\mathbf{u}(k); x(k|k)) = \frac{1}{2}\mathbf{u}(k)^T H \mathbf{u}(k) + F^T \mathbf{u}(k)$$

4.2 Change of variable

At this point, the mentioned way to address the problem proposes a change of variable in order to comply what has been put forward by cooperation-based MPC. That is, the agents should be able to calculate their own optimal controls at iteration p, while the others are supposed to stay at $p-1$. This change is defined in (4.3) and, as it can be seen, supposes a division of the centralized input vector into a summation of each subsystems' inputs weighted by matrices denoted $\mathbf{M}_i$. The aim is the decoupling of each $\mathbf{u}_i(k)$ that comprise $\mathbf{u}(k)$, and the subsequent use of it to implement a control algorithm that acts in a distributed manner.

$$\mathbf{u}(k) = \sum_{i=1}^{M} \mathbf{M}_i \mathbf{u}_i(k) \tag{4.3}$$

Each of the matrices $\mathbf{M}_i$ will consist on the integration of N identity submatrices $\mathbf{1}$ into a set of submatrices of zeros, which will finally define an operator $\mathbf{M}_i \in \mathbb{R}^{(\sum_{i=1}^{M} m_i)N \times m_i N}$. Considering implicitely the dimensions of $\mathbf{1}$ and $\mathbf{0}$, and seeing $\mathbf{M}_i$ as a set of $NM \times N$ submatrices (M: number of subsystems), we have that each $\mathbf{1}$ will take up (from the column i to the last one) the row $(r-1)M+i,\ \forall r = 1,...,N$. The matrix forms of $\mathbf{1}$ and $\mathbf{0}$ are caused due to the fact that each $u_i(k+n|k)$ may have a determined number of components m_i. In other words, each of them will have as dimension $m_i \times m_i$.

This change of variable, together with construction of the matrices $\mathbf{M}_i$ will be shown for simplicity by an example. In this case, we have chosen $M = 2$, so the resultant transformation will be the presented below.

$$\begin{bmatrix} u_1(k|k) \\ u_2(k|k) \\ u_1(k+1|k) \\ u_2(k+1|k) \\ \vdots \\ u_1(k-N+1|k) \\ u_2(k-N+1|k) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}}_{\mathbf{M}_1} \begin{bmatrix} u_1(k|k) \\ u_1(k+1|k) \\ \vdots \\ u_1(k-N+1|k) \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \end{bmatrix}}_{\mathbf{M}_2} \begin{bmatrix} u_2(k|k) \\ u_1(k+1|k) \\ \vdots \\ u_2(k-N+1|k) \end{bmatrix}$$

Therewith we proceed to introduce the change of variable as it has been defined in (4.3) into the objective function in (4.1).

$$\frac{1}{2}\mathbf{u}(k)^T H\mathbf{u}(k)+F^T\mathbf{u}(k)$$

$$\frac{1}{2}\left(\sum_{i=1}^{M}\mathbf{M}_i\mathbf{u}_i(k)\right)^T H\left(\sum_{i=1}^{M}\mathbf{M}_i\mathbf{u}_i(k)\right)+F^T\left(\sum_{i=1}^{M}\mathbf{M}_i\mathbf{u}_i(k)\right)$$

$$\frac{1}{2}\left(\sum_{i=1}^{M}\mathbf{u}_i(k)^T\mathbf{M}_i^T H\right)\left(\sum_{i=1}^{M}\mathbf{M}_i\mathbf{u}_i(k)\right)+\left(\sum_{i=1}^{M}F^T\mathbf{M}_i\mathbf{u}_i(k)\right)$$

$$\frac{1}{2}\left(\mathbf{u}_1(k)^T\mathbf{M}_1^T H+...+\mathbf{u}_i(k)^T\mathbf{M}_i^T H+...+\mathbf{u}_M(k)^T\mathbf{M}_M^T H\right)\left(\mathbf{M}_1\mathbf{u}_1(k)+...+\mathbf{M}_i\mathbf{u}_i(k)+...+\mathbf{M}_M\mathbf{u}_M(k)\right)$$

$$+\left(F^T\mathbf{M}_1\mathbf{u}_1(k)+...+F^T\mathbf{M}_i\mathbf{u}_i(k)+...+F^T\mathbf{M}_M\mathbf{u}_M(k)\right) \tag{4.4}$$

An expression which depends on all the $\mathbf{u}_i(k)$ for $i=1,..,M$ is arrived (4.4). Taking into account that the objective is the plantwide control in a distributed way, this result supposes a useful approach. If we think about it from the point of view of each agent i, we have that each of them have the chance of partially minimizing a function that measures the entire system acting on their own decision variables $\mathbf{u}_i(k)$ at every k.

As it has been held in the proposed cooperation-based problem, per sampling interval the algorithm will enter an iterative procedure to determine the optimal control variables, until reaching either convergence or a certain maximum of iterations p_{max}. The cooperation among the controllers is implemented by means of an objective function that consider the whole plant, together with the fact of sharing information. All the agents at each p receive the trajectories $\mathbf{u}_j(k)$ for $j\neq i$ at $p-1$. So that, applying this data to the function (4.4) and using it as the cost function to optimize by each i, it will be transformed into an expression in terms of $\mathbf{u}_i(k)$, which is the one depicted in the result (4.5).

Result:

$$\frac{1}{2}\mathbf{u}_i(k)^T\left(\mathbf{M}_i^T H\mathbf{M}_i\right)\mathbf{u}_i(k)+\left(\sum_{j\neq i}^{M}\mathbf{u}_j^{p-1}(k)^T\mathbf{M}_j^T H\mathbf{M}_i+F^T\mathbf{M}_i\right)^T\mathbf{u}_i(k) \tag{4.5}$$

Definitions:

$$H_i=\mathbf{M}_i H\mathbf{M}_i$$

$$F_i^p(k)=\mathbf{M}_i^T H\sum_{j\neq i}^{M}\mathbf{M}_j\mathbf{u}_j^{p-1}(k)+\mathbf{M}_i^T F$$

where the dependance of $F_i^p(k)$ on the time index is not only caused by $\mathbf{u}_j^{p-1}(k)$, but also by the matrix F which contains $x(k|k)$ .With that, we can define the optimization problem that will have to be solved individually by all of the agents per iteration (4.6). As it can be observed, the results after implementation will equal the ones using (3.5), as this manner is just a possibility to address the same control problem respecting the same principles. The weighted factors represented explicitily in (3.5), λ_i, can be introduced using the weighted matrices in the cost function. The optimal input to apply to subsystem i per sampling interval will be corresponding $u_i(k|k)$, that is, only the first step in the resulting optimal control sequence is implemented (receding horizon implementation).

$$\mathbf{u}_{i,opt}^p(k)=\ \arg\ \min_{\mathbf{u}_i(k)}\ \frac{1}{2}\mathbf{u}_i(k)^T\left(\mathbf{M}_i^T H\mathbf{M}_i\right)\mathbf{u}_i(k)^T+\left(\sum_{j\neq i}^{M}\mathbf{u}_j^{p-1}(k)^T\mathbf{M}_j^T H\mathbf{M}_i+F^T\mathbf{M}_i\right)\mathbf{u}_i(k) \tag{4.6}$$

4.3 Constraints

We now focus on the issue of introducing constraints on the trajectories decided by the optimization problem, that can be a requirement of the process to control. Using the centralized model, we have that certain limits on the state trajectory vector $\mathbf{x}(k+1)$, can be expressed mathematically as the following matrix inequation,

where $A_x = [1,\ -1]^T$. In others words, all the components of the trajectory vector will have to fulfill certain specified constraints.

$$\begin{bmatrix} A_x & & & & \\ & A_x & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & A_x \end{bmatrix} \begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \vdots \\ x(k+N|k) \end{bmatrix} \leq \begin{bmatrix} b_x \\ b_x \\ \vdots \\ \vdots \\ b_x \end{bmatrix}$$

$$\widehat{A}_x \mathbf{x}(k+1) \leq \widehat{b}_x, \qquad \widehat{A}_x \in \mathbb{R}^{2N\Sigma_i^M n_i \times N\Sigma_i^M n_i}, \qquad \widehat{b}_x \in \mathbb{R}^{2N\Sigma_i^M n_i \times 1} \tag{4.7}$$

Proceeding in the same way (now $A_u = [1,\ -1]^T$), but this time considering the possible costraints in the input trajectory vector, it means, in the decision variables, we get to

$$\begin{bmatrix} A_u & & & & \\ & A_u & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & A_u \end{bmatrix} \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix} \leq \begin{bmatrix} b_u \\ b_u \\ \vdots \\ \vdots \\ b_u \end{bmatrix} \tag{4.8}$$

$$\widehat{A}_u \mathbf{u}(k) \leq \widehat{b}_u, \qquad \widehat{A}_u \in \mathbb{R}^{2N\Sigma_i^M m_i \times N\Sigma_i^M m_i}, \qquad \widehat{b}_u \in \mathbb{R}^{2N\Sigma_i^M m_i \times 1}$$

Both kinds of contrains can be grouped into one single inequation using the definition of $\mathbf{x}_i(k+1)$.

$$\widehat{A}_u \mathbf{u}(k) \leq \widehat{b}_u$$

$$\widehat{A}_x(G_x x(k|k) + G_u \mathbf{u}(k)) \leq \widehat{b}_x \rightarrow \widehat{A}_x G_u \mathbf{u}(k) \leq \widehat{b}_x - \widehat{A}_x G_x x(k|k)$$

$$\underbrace{\begin{bmatrix} \widehat{A}_x G_u \\ \widehat{A}_u \end{bmatrix}}_{AU} \mathbf{u}(k) \leq \underbrace{\begin{bmatrix} \widehat{b}_x - \widehat{A}_x G_x x(k|k) \\ \widehat{b}_u \end{bmatrix}}_{bU} \tag{4.9}$$

Thinking now about the distributed approach, it is possible to relate from this latter expression the constraints used in the optimization problems to solve individually by each of the agents. The change of variable explained above will be also applied here, to the end of defining the constraints on $\mathbf{u}_i(k)$ (4.10).

$$\begin{bmatrix} \widehat{A}_x G_u \\ \widehat{A}_u \end{bmatrix} (\sum_{i=1}^{M} \mathbf{M}_i \mathbf{u}_i(k)) \leq \begin{bmatrix} \widehat{b}_x - \widehat{A}_x G_x x(k|k) \\ \widehat{b}_u \end{bmatrix}$$

$$\begin{bmatrix} \widehat{A}_x G_u \\ \widehat{A}_u \end{bmatrix} \mathbf{M}_i \mathbf{u}_i(k) \leq \begin{bmatrix} \widehat{b}_x - \widehat{A}_x G_x x(k|k) \\ \widehat{b}_u \end{bmatrix} - \begin{bmatrix} \widehat{A}_x G_u \\ \widehat{A}_u \end{bmatrix} (\sum_{j \neq i}^{M} \mathbf{M}_j \mathbf{u}_j(k))$$

$$\underbrace{\begin{bmatrix} \widehat{A}_x G_u \mathbf{M}_i \\ \widehat{A}_u \mathbf{M}_i \end{bmatrix}}_{AU_{dec,i}} \mathbf{u}_i(k) \leq \underbrace{\begin{bmatrix} \widehat{b}_x - \widehat{A}_x G_x x(k|k) - \widehat{A}_x G_u (\sum_{j \neq i}^{M} \mathbf{M}_j \mathbf{u}_j(k)) \\ \widehat{b}_u - \widehat{A}_u (\sum_{j \neq i}^{M} \mathbf{M}_j \mathbf{u}_j(k)) \end{bmatrix}}_{bU_{dec,i}} \tag{4.10}$$

4.4 Algorithm

In the corresponding implementation of the proposed DMPC control problem, we have that per sampling time, the algorithm will involve an iterative procedure to determine what are the optimal control variables $\mathbf{u}_{i,opt}^{p}(k)$ for all i.

This algorithm can be summarized in the following steps for each k.

1. Determine the matrices H,F, AU and bU that were defined where we described the centralized model.

$$H = G_u \widehat{Q} G_u + \widehat{R}, \quad F^T = x(k|k)^T G_x^T \widehat{Q} G_u$$

$$AU = [\widehat{A}_x G u, \ \widehat{A}_u]^T, \quad bU = [\widehat{b}_x - \widehat{A}_x G_x x(k|k), \ \widehat{b}_u]^T$$

2. Enter a while loop conditioned by a maximum number of iterations p_{max} and a convergence condition. This latter one can be seen as $dist_i < \varepsilon$, for every i; where the parameter $dist$ represents the norm of diference between the state and input trajectories calculated at iteration p and the ones at $p-1$.

3. Define the matrices of (4.10) to impose the corresponding constraints for every i, using the change of variable and the result in 1.

$$AU_{dec,i} = AU\mathbf{M}_i, \ \ bU_{dec,i} = bU - AU(\sum_{j \neq i}^{M} \mathbf{M}_j \mathbf{u}_j^{p-1}(k))$$

4. Calculate the optimal $\mathbf{u}_{i,opt}^p(k)$, $\forall i$. Here we make use of (4.6), in which the parameters calculated in 1 are introduced, as well as taking into account the constraints above.

5. Define the control variables $\mathbf{u}_i^p(k)$. It has not been assigned directly the optimal value of $\mathbf{u}_{i,opt}^p(k)$ to $\mathbf{u}_i^p(k)$. In this case, what has been done is defining $\mathbf{u}_i^p(k) = \lambda_i \mathbf{u}_{i,opt}^p(k) + (1-\lambda_i)\mathbf{u}_i^{p-1}(k)$, which introduces a certain inertia in the algorithm .

6. Calculate the predicted trajectories $\mathbf{x}_i^p(k)$ with the values arrived, for which the idea of the inertia has been equally applied.

7. Measure the norms of the diference $[\mathbf{x}_i^p(k), \ \mathbf{u}_i^p(k)]^T - [\mathbf{x}_i^{p-1}(k), \ \mathbf{u}_i^{p-1}(k)]^T$ to give the corresponding values to $dist_i$.

8. Save the vectors $\mathbf{x}_i^p(k)$, $\mathbf{u}_i^p(k)$ for the following iteration and increment the value of p.

9. If $p > p_{max}$ or $dist_i < \varepsilon$ for all i, then

- Implement the first inputs of the finally decided input trajectories on every subsystems.

If $p < p_{max}$ and $dist_i > \varepsilon$ for some i, then

- Go back to step 2.

5 Stability of the algorithm

Achieving stability is a key issue in distributed control, which can be essential in many applications. For this reason, we present in this section a short analysis of the strategy and algorithm proposed concerning this aspect.

The study of the plant's stability should consider the models' equations that describe each subsystem which make it up. On the basis of the above, for cooperation-based MPC, as well as for communication-based strategy, we have made use of the models whose definition has been presented in (2.3). For their part, they suppose a matrix combination of parameters related to each subsystem itself and others related to the influence of the rest, which provides useful features to tackle the distributed problem. Figure 5.1 is a schematic representation of the situation for the particular case in which the number of subsystem is 2. The couplings between them can be seen by means of the block diagram, in which the funtions in the z-domain for $i,j \in \{1,2\}$ are defined as: $I_{ij}(z) = B_{ij}U_j(z)/(\mathbf{I}z - A_{ii})$ and $S_{ij}(z) = A_{ij}X_j(z)/(\mathbf{I}z - A_{ii})$ (the latter for $i \neq j$). It can be noted here that extending the plant to the case in which $M > 2$ would also lead to functions with $z\mathbf{I} - A_{ii}$ as characteristic polynomial when the system is represented in this way.

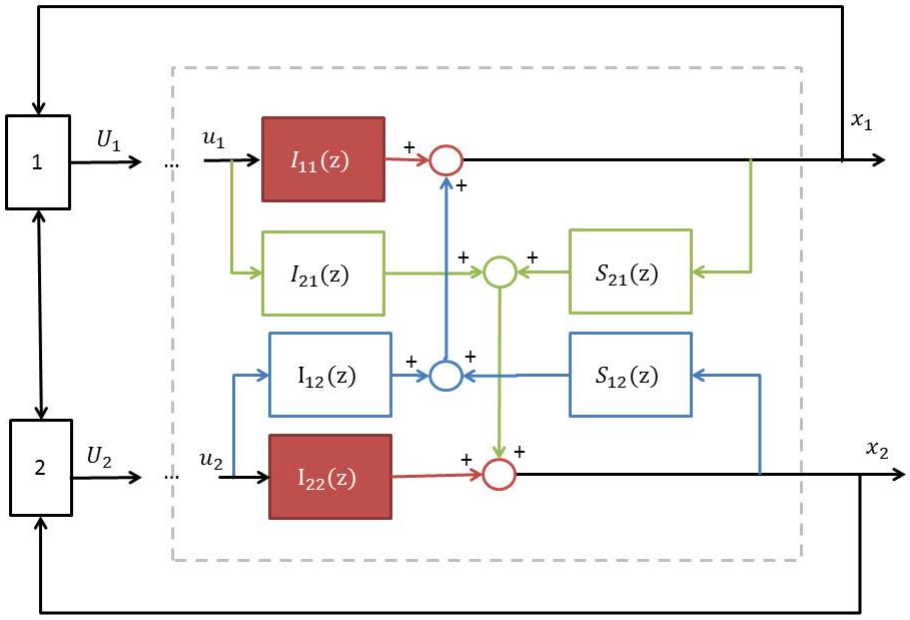

Figure 5.1 Scheme of the problem for $M = 2$.

The subject matter of this project revolves around what happens outside the box in dashed lines, that is in the actions we should take on the system to achieve our target and how to find them. The blocks 1 and 2 represent the agents in charge of coming to an agreement, which for practical purposes can be seen as optimal

input trajectories. With the application of the algorithm it is expected that the overall system with evolve to the desired point over time, which is equivalent to reducing when possible the corresponding cost. Given that, we are going to assume the stability of the plant's model and focus if we can trust the algorithm to meet this goal.

5.1 Quadratic forms

Since the development of MPC makes an extensive use of quadratic forms, we are going to present a short analysis of this specific kind of functions in order to come up with some results that will be useful now.

For further use, the concept of symmetric and skew-symmetric matrices is reviewed. For a square matrix denoted as $\mathbf{P}$, we have:

- $\mathbf{P}$ is symmetric if $\mathbf{P} = \mathbf{P}^T$
- $\mathbf{P}$ is skew-symmetric if $\mathbf{P} = -\mathbf{P}^T$

With that, we can affirm that any square matrix can be decomposed into a summation of a symmetric and a skew-symmetric as shown below.

$$\mathbf{P} = \underbrace{\frac{\mathbf{P}+\mathbf{P}^T}{2}}_{Symm.} + \underbrace{\frac{\mathbf{P}-\mathbf{P}^T}{2}}_{Skew-Symm.}$$

Using now this definition into a generic quadratic function $\mathbf{v}^T\mathbf{P}\mathbf{v}$, we have that

$$\mathbf{v}^T\left(\frac{\mathbf{P}+\mathbf{P}^T}{2}+\frac{\mathbf{P}-\mathbf{P}^T}{2}\right)\mathbf{v} = \mathbf{v}^T\left(\frac{\mathbf{P}+\mathbf{P}^T}{2}\right)\mathbf{v} + \mathbf{v}^T\left(\frac{\mathbf{P}-\mathbf{P}^T}{2}\right)\mathbf{v}$$

$$\mathbf{v}^T\left(\frac{\mathbf{P}-\mathbf{P}^T}{2}\right)\mathbf{v} = -\mathbf{v}^T\left(\frac{\mathbf{P}-\mathbf{P}^T}{2}\right)\mathbf{v}, \ \forall\mathbf{v} \ \longleftrightarrow \ \mathbf{v}^T\left(\frac{\mathbf{P}-\mathbf{P}^T}{2}\right)\mathbf{v} = 0$$

$$\mathbf{v}^T\mathbf{P}\mathbf{v} = \mathbf{v}^T\left(\frac{\mathbf{P}+\mathbf{P}^T}{2}\right)\mathbf{v} \rightarrow \mathbf{v}^T\mathbf{P}\mathbf{v} = \mathbf{v}^T\widehat{\mathbf{P}}\mathbf{v} \ \text{ where } \ \widehat{\mathbf{P}} \text{ is a symmetric matrix}$$

.

*Result:*It can be assumed that the expression is always equal to a quadratic function with a symmetric matrix.

5.1.1 Convexity

Now we analyse the convexity of expressions which the form $\frac{1}{2}\mathbf{v}^T\widehat{\mathbf{P}}\mathbf{v}+\mathbf{c}^T\mathbf{v}$, where $\widehat{\mathbf{P}}$ is a symmetric positive definite matrix.

Definiton: Convex functions

A function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex function if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y), \qquad \forall x,y \in \mathbb{R}^n, \qquad \forall\lambda \in [0,1]$$

Applying this definition to $f(\mathbf{v}) = \frac{1}{2}\mathbf{v}^T\widehat{\mathbf{P}}\mathbf{v}+\mathbf{c}^T\mathbf{v}$, we have

$$f(\lambda\mathbf{v}_1+(1-\lambda)\mathbf{v}_2) = f(\lambda(\mathbf{v}_1-\mathbf{v}_2)+\mathbf{v}_2) = \frac{1}{2}(\lambda(\mathbf{v}_1-\mathbf{v}_2)+\mathbf{v}_2)^T\widehat{\mathbf{P}}(\lambda(\mathbf{v}_1-\mathbf{v}_2)+\mathbf{v}_2)+\mathbf{c}^T(\lambda(\mathbf{v}_1-\mathbf{v}_2)+\mathbf{v}_2)$$

$$\frac{1}{2}\lambda^2(\mathbf{v}_1-\mathbf{v}_2)^T\widehat{\mathbf{P}}(\mathbf{v}_1-\mathbf{v}_2)+\frac{1}{2}\lambda(\mathbf{v}_1-\mathbf{v}_2)^T\widehat{\mathbf{P}}\mathbf{v}_2+\frac{1}{2}\mathbf{y}^T\widehat{\mathbf{P}}\lambda(\mathbf{v}_1-\mathbf{v}_2)+\frac{1}{2}\mathbf{v}_2^T\widehat{\mathbf{P}}\mathbf{v}_2+\lambda\mathbf{c}^T(\mathbf{v}_1-\mathbf{v}_2)+\mathbf{c}^T\mathbf{v}_2$$

$$\leq \frac{1}{2}\lambda(\mathbf{v}_1-\mathbf{v}_2)^T\widehat{\mathbf{P}}(\mathbf{v}_1-\mathbf{v}_2)+\frac{1}{2}\lambda(\mathbf{v}_1-\mathbf{v}_2)^T\widehat{\mathbf{P}}\mathbf{y}+\frac{1}{2}\mathbf{v}_2^T\widehat{\mathbf{P}}\lambda(\mathbf{v}_1-\mathbf{v}_2)+\frac{1}{2}\mathbf{v}_2^T\widehat{\mathbf{P}}\mathbf{v}_2+\lambda\mathbf{c}^T(\mathbf{v}_1-\mathbf{v}_2)+\mathbf{c}^T\mathbf{v}_2$$

$$= \frac{1}{2}\lambda(\mathbf{v}_1-\mathbf{y})^T\widehat{\mathbf{P}}\mathbf{v}_1+\frac{1}{2}\mathbf{v}_2^T\widehat{\mathbf{P}}\lambda(\mathbf{v}_1-\mathbf{v}_2)+\frac{1}{2}\mathbf{v}_2^T\widehat{\mathbf{P}}\mathbf{v}_2+\lambda\mathbf{c}^T(\mathbf{v}_1-\mathbf{v}_2)+\mathbf{c}^T\mathbf{v}_2$$

$$= \frac{1}{2}\lambda \mathbf{v}_1^T \widehat{\mathbf{P}} \mathbf{v}_1 - \frac{1}{2}\lambda \mathbf{v}_2^T \widehat{\mathbf{P}} \mathbf{v}_2 + \frac{1}{2}\mathbf{v}_2^T \widehat{\mathbf{P}} \mathbf{v}_2 + \lambda \mathbf{c}^T \mathbf{v}_1 - \lambda \mathbf{c}^T \mathbf{v}_2 + \mathbf{c}^T \mathbf{v}_2$$

$$\leq \lambda f(\mathbf{v}_1) + (1-\lambda) f(\mathbf{v}_2)$$

Hence, $f(\mathbf{v})$ is a convex function.

The theoretical minimization of functions with this structure implies finding the $\mathbf{v}$ that makes the gradient equal to 0 in order to have an optimum at this point. We proof here that in the case of these convex funtions, the fact of reaching this $\mathbf{v}_{opt}$ will lead to a minimization of $f(\mathbf{v})$.

$$\nabla f(\mathbf{v}) = \widehat{\mathbf{P}}\, \mathbf{v}_{opt} + \mathbf{c} = 0$$

$$f(\mathbf{v}) = f(\mathbf{v}_{opt} + (\mathbf{v} - \mathbf{v}_{opt}))$$

$$f(\mathbf{v}) = \frac{1}{2}(\mathbf{v}_{opt} + (\mathbf{v} - \mathbf{v}_{opt}))^T \widehat{\mathbf{P}} (\mathbf{v}_{opt} + (\mathbf{v} - \mathbf{v}_{opt})) + \mathbf{c}^T (\mathbf{v}_{opt} + (\mathbf{v} - \mathbf{v}_{opt}))$$

$$f(\mathbf{v}) = \frac{1}{2}(\mathbf{v}_{opt}^T \widehat{\mathbf{P}} \mathbf{v}_{opt} + (\mathbf{v} - \mathbf{v}_{opt})^T \widehat{\mathbf{P}} \mathbf{v}_{opt}) + \frac{1}{2}(\mathbf{v}_{opt}^T \widehat{\mathbf{P}} (\mathbf{v} - \mathbf{v}_{opt}) + (\mathbf{v} - \mathbf{v}_{opt})^T \widehat{\mathbf{P}} (\mathbf{v} - \mathbf{v}_{opt})) + \mathbf{c}^T (\mathbf{v}_{opt} + (\mathbf{v} - \mathbf{v}_{opt}))$$

$$f(\mathbf{v}) = \frac{1}{2}(\mathbf{v}_{opt}^T \widehat{\mathbf{P}} \mathbf{v}_{opt}) + \frac{1}{2}((\mathbf{v} - \mathbf{v}_{opt})^T \widehat{\mathbf{P}} (\mathbf{v} - \mathbf{v}_{opt})) + \mathbf{c}^T \mathbf{v}_{opt}$$

$$f(\mathbf{v}) = f(\mathbf{v}_{opt}) + \frac{1}{2}((\mathbf{v} - \mathbf{v}_{opt})^T \widehat{\mathbf{P}} (\mathbf{v} - \mathbf{v}_{opt}))$$

$$f(\mathbf{v}) \geq f(\mathbf{v}_{opt})$$

5.2 An insight on the objective functions

The objective functions applied and their characteristics determine the optimization problems to be solved by the agents, hence the performace of the control strategy. This objective functions has been defined for each agent i ($i = 1,...,M$) as

$$\min_{\mathbf{u}_i} \sum_{l=1}^{M} \lambda_l \phi_l(\mathbf{u}_i(k), \mathbf{u}_{j\neq i}^{p-1}; x_l(k|k))$$

which supposes a weighted summation of the basic cost funciton presented in (3.1) and repeated below.

$$\phi_i(\mathbf{x}_i(k), \mathbf{u}_i(k), \mathbf{x}_{j\neq i}^{p-1}(k), \mathbf{u}_{j\neq i}^{p-1}(k); x_i(k|k)) =$$
$$\sum_{n=0}^{N-1} x_i^T(k+n|k) Q_i(k+n|k) x_i(k+n|k) + u_i^T(k+n|k) R_i(k+n|k) u_i(k+n|k)$$

It has been previously indicated with respect to them that at all times we have considered $Q_i(k+n) > 0$, $R_i(k+n) > 0, \forall n = 0,1,...,N-1$ and $\forall i$ with $i = 1,...,M$. What this fact implies is that the corresponding matrices are positive definite, or in other words, that all the its eigenvalues are strictly positive. Assuming this mathematical condition enables us to apply the entailing properties, which provide useful features in the minimization problems. The definition of positive definite matrix involves that

$$\forall x_i \neq \mathbf{0}, \qquad x_i^T Q_i x_i > 0$$

$$\forall u_i \neq \mathbf{0}, \qquad u_i^T R_i u_i > 0$$

so that, all individual funcitons $\phi_i(\cdot)$ will never reach negative values and, therefore, the weighted sumation with positive weights will not reach them either. At this point, it is important to remind that the objective functions have been redefined as a matrix equation terms of the controls actions and in which a terminal cost

has been implicetely included.

$$\phi_i(\mathbf{u}_i(k),\mathbf{u}_{j\neq i}^{p-1}(k);x_i(k|k)) = \frac{1}{2}\mathbf{u}_i^T(k)\left(G_{ui}^T\widehat{Q}_iG_{ui}+\widehat{R}_i\right)\mathbf{u}_i(k)+\left(x_i(k|k)^TG_{xi}^T\widehat{Q}_iG_{ui}+\mathbf{w}_i^{p-1}(k)^TG_{wi}^T\widehat{Q}_iG_{ui}\right)\mathbf{u}_i(k) \tag{5.1}$$

The expression above shows the features and form of a generic quadratic form $\frac{1}{2}\mathbf{v}^T\mathbf{P}\mathbf{v}+\mathbf{c}^T\mathbf{v}$, which involves the possibility of using what has been above proved for the latter. Therefore, from now on we can assumme that $\phi_i(\cdot)$ is a convex positive definite funcition. Consequently the weighted summation that define the objective functions for each of the agent i ($i=1,...,M$), where $\lambda_l \geq 0$ ($l=1,...,M.$), are also covex p.d. problems, as the sumation of convex functions will not alter the condition of convexity.

5.3 The algorithm in the negotiating process and progression over time

Henceforth, it will be study whether this cost function is a decreasing sequence lower-bounded by zero. This issue will be addresed from two different approaches. Firsly, we will focus on a certain time step k, for which the the question: *What will happen along the successive iterations at time k?* is desired to be answered. After it, the evolution will be analysed from the point of view of the simulation time lenght, that is: *What will happen as k increases?*

5.3.1 Decrease of the cost with the iterations

The cooperation strategy suggests minimizing (3.4) per iteration p ($p < p_{max}$) and per agent i , so that each of them calculates its own optimal control action $\mathbf{u}_{i,opt}^p(k)(i=1,...,M)$. This leads to the determination of a set of optimal inputs associated to each p that are denoted as $\{\mathbf{u}_{1,opt}^p(k),...,\mathbf{u}_{M,opt}^p(k)\}$, which after the iterative procedure will define the controls implemented to the subsystems.

So that, $\phi_{i,c}(\mathbf{u}_i(k),\mathbf{u}_{j\neq i}^{p-1}(k);x(k|k))$ denotes the objective function to minimize by agent i at iteration p and step time k. Let $\phi_{i,c}^p(\cdot)$ denote the value that the cost function take when $\mathbf{u}_i^p(k)$ is introduced as a parameter.

$$\phi_{i,c}^p(\mathbf{u}_i^p(k),\mathbf{u}_{j\neq i}^{p-1}(k);x(k|k)) = \sum_{l=1}^{M}\lambda_l\phi_l(\mathbf{u}_i^p(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k))$$

The convexity property of $\phi_{i,c}^p(\cdot)$ and the proof in the previous section enables us to affirm that, theoretically, if the minimization is done correctly, then

$$\phi_{i,c}(\mathbf{u}_{i,opt}^p(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k)) \leq \phi_{i,c}^{p-1}(\mathbf{u}_i^{p-1}(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k))$$

However, in the algorithm proposed the control action implemented is not directly the optimal calculated but it also considers what was the value of the input at the iteration before, in a manner of

$$\mathbf{u}_i^p(k) = \lambda_i\mathbf{u}_{i,opt}^p(k)+(1-\lambda_i)\mathbf{u}_i^{p-1}(k)$$

Using now the mathematical definition of convex funtions:

$$\phi_{i,c}^p(\lambda_i\mathbf{u}_{i,opt}^p(k)+(1-\lambda_i)\mathbf{u}_i^{p-1}(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k))$$

$$\leq \lambda_i\phi_{i,c}^p(\mathbf{u}_{i,opt}^p(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k))+(1-\lambda_i)\phi_{i,c}^p(\mathbf{u}_i^{p-1}(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k))$$

$$\phi_{i,c}^p(\lambda_i\mathbf{u}_{i,opt}^p(k)+(1-\lambda_i)\mathbf{u}_i^{p-1}(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k))$$

$$\leq \phi_{i,c}^p(\mathbf{u}_i^{p-1}(k),\mathbf{u}_{j\neq i}^{p-1}(k)x(k|k))+\lambda_i\left(\phi_{i,c}^p(\mathbf{u}_{i,opt}^p(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k))-\phi_{i,c}^p(\mathbf{u}_i^{p-1}(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k))\right)$$

where

$$\phi_{i,c}^p(\mathbf{u}_{i,opt}^p(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k))-\phi_{i,c}^p(\mathbf{u}_i^{p-1}(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k)) \leq 0$$

Hence, the values that cost function of each agent will take per iteration will be a decreasing sequence at each step time k.

Result:

$$\phi_{i,c}^{p}(\mathbf{u}_i^p(k),\mathbf{u}_{j\neq i}^p(k),x(k|k)) \leq \phi_{i,c}^{p-1}(\mathbf{u}_i^{p-1}(k),\mathbf{u}_{j\neq i}^{p-1}(k),x(k|k)) \tag{5.2}$$

5.3.2 Decrease of the cost with the time

Once the evolution of the cost has been studied for a certain time k, focusing on what will happen while the iterative procedure is held, what will be addressed now is the problem from the point of view of the time. That is, we wonder here what will happen when the time k increases.

Firstly, it will be assumed that at time k the iteration process is finished after f_k iterations, where thus f_k denotes an integer in $\{0,p_{max}\}$. Given that, at time $k-1$, the corresponding iteration is f_{k-1}.

We have that when the step time is changed, for example turning to k from $k-1$, the initial value of the cost function at k will be determined by the control actions defined at $k-1$. At this point, we are going to introduce a definition of the vectors $\mathbf{u}_i^0(k)$, $i = 1,...,M$, which are specified by the inputs trajectories agreed at the previous time step but in which morevoer we take account of the constraint which implies that $u_i(k+n) = 0$ for all $n \geq N$ and for all i. Therefore $\mathbf{u}_i^0(k)$ are vectors which are associated to time k and which provide information to ponderate the objective functions at the initial point.

$$\mathbf{u}_i^0(k) = \begin{bmatrix} u_i^{f_{k-1}}(k|k-1) \\ u_i^{f_{k-1}}(k+1|k-1) \\ \vdots \\ u_i^{f_{k-1}}(k+N-1|k-1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Let $\phi_c(\mathbf{u}^{f_{k-1}}(k),x(k-1|k-1))$ represent the value of the summatory $\sum_{l=1}^{M}\lambda_l\phi_l(\mathbf{u}^{f_{k-1}};x_l(k-1|k-1))$. In other words, the value that the cost reaches at $k-1$. The initial cost at k can be definded as $\phi_c(\mathbf{u}^0(k);x(k|k))$. With that, the following holds

$$\phi_c(\mathbf{u}^0(k);x(k|k)) = \phi_c(\mathbf{u}^{f_{k-1}}(k-1);x(k-1|k-1))$$
$$-\sum_{i=1}^{M}\left(\lambda_i x_i(k-1|k-1)^T Q_i x_i(k-1|k-1) + u_i^{f_{k-1}}(k-1|k-1)^T R_i u_i^{f_{k-1}}(k-1|k-1)\right)$$

Finally, and considering the result in the previous (i), we get to

$$\phi_c(\mathbf{u}^{p_k}(k);x(k|k)) \leq \phi_c(\mathbf{u}^0(k);x(k|k)) \leq \phi_c(\mathbf{u}^{f_{k-1}}(k-1);x(k-1|k-1)) \leq \phi_c(\mathbf{u}^0(0);x(0|0))$$

where p denotes any iteration number greater than 0 and smaller than p_{max}, and $\mathbf{u}^0(0)$ and $x(0|0)$ the initial input and states, respectively.

Following the reference article [1], we could affirm that: under the assumption of stability of the models, together with setting $Q_i(k+n) > 0$ (for $n = 0,1..,N$) and $R_i(k+n) > 0$ (for $n = 0,1..,N-1$), the origin is an exponentially stable equilibrium for the closed-loop (state-feedback) system.

6 Attacks to the DMPC scheme

Hitherto, it has been assumed the algorithm works in a reliable information exchange setting in which all the agents proceed such as the strategy indicates. In this part, the fact of having an agent which is not willing to perform so is studied. That is, we address the problem here of having a controller which introduces false information in the course of the algorithm's application, thus represents a clear risk that can make the control lose optimality and so affects negatively the overall performance. This kind of controllers use misleading information in different ways whith the purpose of profiting from the rest of the agents. Some of these possibilities that they can take to carry out the deception are presented here. Afterward, it will be studied the effects it would cause in the examples presented, focusing this analysis in how the algorithm's optimality for the global system is affected.

6.1 Attacker's objective

Henceforth, let $a \in \{1,..,M\}$ denotes the malicious controller that will alter the normal development of the DMPC algorithm proposed.

In first place, and for a further use in the attacks' analysis, it should be clearly stated what is the purpose which motivates the malicious performance of the attacker and how the mischief can be evaluated when implemented.

The scheme under consideration pursues a convenient global behaviour based on a cooperative iterative negotiation in which the agents realign their own controls with the others until an agreement which fulfills certain conditions is reached. It gives rise to the possibility of being locally better off when applied a different input with respect to the one agreed. The latter is the starting point to carry out an attack.

From a local point of view, the welfare of an agent i ($i = 1,...,M$) is assessed by expression (6.1), therefore a decrease of the cumulative cost pertaining to the attacker will be a reflection of the attack's effectiveness.

$$\begin{aligned} &J_i(\mathbf{x}(k+1),\mathbf{u}(k);x_i(k|k)) = \\ &\sum_{n=0}^{N-1}\left[\left(x_i(k+n|k)-x_{i,ref}\right)^T Q_i(k+n|k)\left(x_i(k+n|k)-x_{i,ref}\right)^T + u_i^T(k+n|k)R_i(k+n|k)u_i(k+n|k)\right] \\ &+\left(x_i(k+N|k)-x_{i,ref}\right)^T Q_i(k+N|k)\left(x_i(k+N|k)-x_{i,ref}\right) \end{aligned} \tag{6.1}$$

Likewise, the associated expression to (6.1) is declared, in which the dependance of the states is removed, due to its usefulness for this problem.

$$\begin{aligned} &J_i(\mathbf{u}(k);x_i(k|k)) = \\ &\left(G_{xi}\left(x_i(k|k)-x_{i,ref}\right)+G_{ui}\mathbf{u}_i(k)+G_{wi}\mathbf{w}_i(k)\right)^T \widehat{Q}_i\left(G_{xi}\left(x_i(k|k)-x_{i,ref}\right)+G_{ui}\mathbf{u}_i(k)+G_{wi}\mathbf{w}_i(k)\right)+ \\ &\mathbf{u}_i^T(k)\widehat{R}_i\mathbf{u}_i(k)+cte \end{aligned}$$

Given that this function will be treated with a view to its optimization and that the way of action on the plant is through the inputs $\mathbf{u}_i(k)$, it will also be simplified to (6.2), where the introduced modification eliminates

information that has no influence for the problem.

$$J_i(\mathbf{u}(k);x_i(k|k)) = \frac{1}{2}\mathbf{u}_i(k)^T\overline{H}_i\mathbf{u}_i(k) + \overline{F}_i^T(k)\mathbf{u}_i(k) \tag{6.2}$$

$$H_{i,loc} = G_{ui}\widehat{Q}_i G_{ui} + \widehat{R}_i$$
$$F_{i,loc}^T(k) = \left(x_i(k|k) - x_{i,ref}\right)^T G_{xi}^T\widehat{Q}_i G_{ui} + \mathbf{w}_i^T(k)G_{wi}^T\widehat{Q}_i G_{ui}$$
$$H_{i,loc} \in \mathbb{R}^{Nm_i\times Nm_i}, \qquad F_{i,loc} \in \mathbb{R}^{Nm_i\times 1}$$

$J_i(\cdot)$ represents approximately the same as $\phi_i(\cdot)$ with just the difference that there is no variance of iterations concerning the variables. Therefore, it represents theoretically the objective, but given the way of communication between the controllers, the one useful for the implementation is $\phi_i(\cdot)$.

6.2 False reference

The first possible attack to the DMPC scheme presented is the one that takes action by means of the introduction of a false reference, which will be denoted from now on as $x_{a,ref}^f$. The misleading information enters the DMPC scheme through the cost function optimized by the attacker, hence the subindex a. On the basis of the above, the objective of the malicious agent is a reduction of (6.1) for $i = a$, and consequently of the corresponding cumulative cost. The goal is to be attained by the change of $x_{a,ref}$ to $x_{a,ref}^f$ in the optimization problem solved by a, that is, the original objective function is amended for $i = a$ as follows (6.3).

$$\begin{aligned}\phi_{a,c}(\mathbf{u}_a(k),\mathbf{u}_{j\neq a}^{p-1}(k);x(k|k)) &= \sum_{l=1,l\neq a}^{M}\lambda_l\phi_l(\mathbf{u}_a,\mathbf{u}_{j\neq a}^{p-1};x_l(k))\\ &+\lambda_a\sum_{n=0}^{N-1}\left(x_a(k+n+1|k)-x_{a,ref}^f\right)^T Q_i(k+n+1|k)\left(x_a(k+n+1|k)-x_{a,ref}^f\right)\\ &+\lambda_a u_a^T(k+n|k)R_a(k+n|k)u_a(k+n|k)\end{aligned} \tag{6.3}$$

When one of the controllers decides to alter the regular development of the algorithm by introducing a false reference, the immediate mathematical consequence will not be just a change in its own objective function but also in the ones to optimize by the rest of the agents $i \neq a$. In other words, the inherent communicative aspect of the DMPC algorithm is the mean that the attacker uses to spread the false information to the whole plant. Moreover, the cooperation between agents will lead to misleading results and thus the attacker's aim will be propitiating that these are in favour of a. Mathematically, what has been described can be seen clearly hereunder.

- Problem solved by agent a:

$$\min_{\mathbf{u}_a(k)} \qquad \phi_{a,c}(\mathbf{u}_a(k),\mathbf{u}_{j\neq a}^{p-1}(k);x(k|k))$$

$$\rightarrow \mathbf{u}_{a,opt}^p(k) = \arg\min_{\mathbf{u}_a(k)} \qquad \phi_{a,c}(\mathbf{u}_a(k),\mathbf{u}_{j\neq a}^{p-1}(k);x(k|k))$$

$$\rightarrow \mathbf{u}_{a,opt}^p(k) \text{ contains false information as consequence of } x_{a,ref}^f$$

- Problem solved by agents $i \neq a$ ($i \in \{1,...,M\}\setminus\{a\}$):

$$\min_{\mathbf{u}_i(k)} \qquad \phi_{i,c}(\mathbf{u}_i(k),\mathbf{u}_{j\neq i}^{p-1}(k);x(k|k)) = \sum_{l=1}^{M}\lambda_l\phi_l(\mathbf{u}_i(k),\mathbf{u}_{j\neq i}^{p-1}(k);x(k|k))$$

$$\rightarrow \mathbf{u}_{i,opt}^p(k) = \arg\min_{\mathbf{u}_i(k)} \ \phi_{i,c}(\mathbf{u}_i(k),\mathbf{u}_{j\neq i}^{p-1}(k);x(k|k))$$

$\rightarrow \mathbf{u}_{i,opt}^p(k)$ affected by $\mathbf{u}_a^{p-1}(k)$, therefore, the negotation process is steered to a new situation

Consequently, the local improvement is based on the appropiated choice of the false reference, which will meet the target when causing the solutions $\mathbf{u}_{a,opt}^{p}(k)$, and its impact on the rest of subsystems, to be in support of agent a.

Clearly, for this kind of attack, the dependence of the objective functions for all i on x_{ref} is of particular importance , as it is the fact that the attacker seizes to go ahead with its deception.

At this point, it should be remarked that x_{ref} represents a vector in $\mathbb{R}^{\sum_{i=1}^{M} n_i \times 1}$ with the state references for all subsystems in the plant, which allows the application of the change of variable below. Every $\mathbf{1}$ of each $\mathbf{P}_i$ reprentes an identity matrix in $\mathbb{R}^{n_i \times n_i}$. On the other hand, each $\mathbf{0}$ in row j of $\mathbf{P}_i$ corresponds to null matrices of dimension $n_j \times n_i$.

$$x_{ref} = \begin{bmatrix} x_{1,ref} \\ \vdots \\ x_{a,ref} \\ \vdots \\ x_{M,ref} \end{bmatrix} ; \longrightarrow x_{ref} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \vdots \\ \vdots \end{bmatrix} x_{1,ref} + \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ \vdots \end{bmatrix} x_{2,ref} + \cdots + \begin{bmatrix} \mathbf{0} \\ \vdots \\ \vdots \\ \mathbf{1} \end{bmatrix} x_{M,ref}$$

$$x_{ref} = \sum_{i=1}^{M} \mathbf{P}_i x_{i,ref} \tag{6.4}$$

The practicality of the latter lies in the fact of reaching certain decoupling of each agent's reference, which will be used afterwards.

6.2.1 Pursuit of an optimal $x_{a,ref}^{f}$

In this part, the problem addressed is focused on reaching the greatest effectiveness when conducting a *false reference* attack, considering the mentioned effectiveness from the point of view of the attacker. In other words, it sought to find the optimal value of the misleading $x_{a,ref}^{f}$ that will lead to the best possible minimization of (6.1) for $i = a$ along the negotiation.

It is given, for the considered algorithm, that the problems which agents $i \neq a$ carry out are such as:

$$\min_{\mathbf{u}_i(k)} \ \frac{1}{2}\mathbf{u}_i(k)^T(\mathbf{M}_i H \mathbf{M}_i)\mathbf{u}_i(k) + \left(\sum_{j\neq i}^{M} \mathbf{u}_j^{p-1}(k)^T \mathbf{M}_j^T H \mathbf{M}_i + F^T(k)\mathbf{M}_i\right)\mathbf{u}_i(k)$$

where

$$F^T(k) = (x(k|k) - x_{ref})^T G_x^T \widehat{Q} G_u$$

Analytically, the attack that a conducts leads to a redefinition of the matrix $F(k)$ that appears in the linear term, which is an immediate consequence of the change from $x_{a,ref}$ to the false $x_{a,ref}^{f}$. Let denote the misleading matrix arrived as $F(k, x_{ref}^{f})$ and assume its use in the calculation of $\mathbf{u}_a^p(k)$. It must be concluded that the problem of agent a is the one that follows.

Problem agent a:

$$\min_{\mathbf{u}_a(k)} \ \frac{1}{2}\mathbf{u}_a(k)^T(\mathbf{M}_a H \mathbf{M}_a)\mathbf{u}_a(k) + \left(\sum_{j\neq a}^{M} \mathbf{u}_j^{p-1}(k)^T \mathbf{M}_j^T H \mathbf{M}_a + F^T(k, x_{a,ref}^{f})\mathbf{M}_a\right)\mathbf{u}_a(k)$$

where

$$F^T(k, x_{a,ref}^{f}) = (x(k|k) - x_{ref}^{f})^T G_x^T \widehat{Q} G_u = \left(x(k|k) - \sum_{i=1, i\neq a}^{M} \mathbf{P}_i x_{i,ref} - \mathbf{P}_a x_{a,ref}^{f}\right)^T G_x^T \widehat{Q} G_u \tag{6.5}$$

Let consider hereon that $x^f_{a,ref}$ is not a parameter any more and that it comes into play as a variable, affecting directly the computing of every $\mathbf{u}^p_{a,opt}(k)$. Let also denote this dependance as $\mathbf{u}^p_{a,opt}(k, x^f_{a,ref})$, therefore

$$\mathbf{u}^p(k) = \sum_{i=1}^{M} \mathbf{M}_i \mathbf{u}^p_i(k)$$

$$\mathbf{u}^p_a(k)\text{: Function of } x^f_{a,ref} \longrightarrow \mathbf{u}^p = \mathbf{u}^p(k, x^f_{a,ref})$$

As it has been described, the objective to keep in mind is the improvement of the attacker's local cost, which is approximated by

$$\phi_a(\mathbf{u}_a(k), \mathbf{u}^{p-1}_{j\neq a}(k); x_a(k|k)) = \frac{1}{2}\mathbf{u}^T_a(k)\left(G^T_{ua}\widehat{Q}_a G_{ua} + \widehat{R}_a\right)\mathbf{u}_a(k) +$$

$$\left((x_a(k|k) - x^f_{a,ref})^T G^T_{xa}\widehat{Q}_a G_{ua} + \mathbf{w}^{p-1}_a(k)^T G^T_{wa}\widehat{Q}_a G_{ua}\right)\mathbf{u}_a(k)$$

This last expression allows the computing of the optimal value of $x^f_{a,ref}$ that the malicious agent should take. In the corresponding implementation, the values agreed along the negotations will be used to update the information with the purpose of steering the negotation in an appropiated way. In short, the optimization that agent a will solve per iteration (together with the one which determine its input) to get to the optimal $x^f_{a,ref}$ that should be aplied is

$$\min_{x^f_{a,ref}} \quad \frac{1}{2}\mathbf{u}^T_a(k)\left(G^T_{ua}\widehat{Q}_a G_{ua} + \widehat{R}_a\right)\mathbf{u}_a(k) + \left((x_a(k|k) - x^f_{a,ref})^T G^T_{xa}\widehat{Q}_a G_{ua} + \mathbf{w}^{p-1}_a(k)^T G^T_{wa}\widehat{Q}_a G_{ua}\right)\mathbf{u}_a(k) \tag{6.6}$$

Therefore,

$$x^f_{a,ref}{}^* = \arg\min_{x^f_{a,ref}} \frac{1}{2}\mathbf{u}^T_a(k)\left(G^T_{ua}\widehat{Q}_a G_{ua} + \widehat{R}_a\right)\mathbf{u}_a(k) + \left((x_a(k|k) - x^f_{a,ref})^T G^T_{xa}\widehat{Q}_a G_{ua} + \mathbf{w}^{p-1}_a(k)^T G^T_{wa}\widehat{Q}_a G_{ua}\right)^T \mathbf{u}_a(k)$$

6.2.2 Particular case. Non-constrained problem

Considering the problem under assumption that no constraints are applicable would allow us to determine the optimization's problem solutions analytically. In addition to what precedes, this special case is studied here with the purpose of defining the mentioned expression for the optimal false reference.

Let's consider the objective functions (4.5) and remark the dependance of $F^p_i(k)$ on every given p and time instant k. As a result, the optimal inputs $\mathbf{u}^p_{i,opt}(k)$ calculated under this special conditions will be

$$H_i \mathbf{u}^p_{i,opt}(k) = -F^p_i(k) \longrightarrow \mathbf{u}^p_{i,opt}(k) = -H^{-1}_i F^p_i(k)$$

in which all squared matrices H_i are supposed to be non singular. Equivallently, we would have

$$\mathbf{u}^p_{a,opt}(k, x^f_{a,ref}) = -H^{-1}_a F^p_a(k, x^f_{a,ref})$$

$$H_a = \mathbf{M}_a H \mathbf{M}_a, \qquad F^{pT}_a(k, x^f_{a,ref}) = \sum_{j\neq a}^{M} \mathbf{u}^{p-1}_j(k)^T \mathbf{M}^T_j H \mathbf{M}_a + F^T(k, x^f_{a,ref})\mathbf{M}_a$$

With that, we have to go back to (6.6) and try to find what is the optimal $x^{f*}_{a,ref}$. Let's use:

$$\overline{H}_a = G^T_{ua}\widehat{Q}_a G_{ua} + \widehat{R}_a$$

$$\overline{F}^T_a = (x_a(k|k) - x^f_{a,ref})^T G^T_{xa}\widehat{Q}_a G_{ua} + \mathbf{w}^{p-1}_a(k)^T G^T_{wa}\widehat{Q}_a G_{ua}$$

Then, the problem to solve at any iteration p is:

$$\min_{x^f_{a,ref}} \quad \frac{1}{2}\left(H_a^{-1}F_a^p(k,x^f_{a,ref})\right)^T \overline{H}_a \left(H_a^{-1}F_a^p(k,x^f_{a,ref})\right) - \overline{F}_a^T(k)H_a^{-1}F_a^p(k,x^f_{a,ref})$$

$$\min_{x^f_{a,ref}} \quad \frac{1}{2}\left(\sum_{j\neq a}^{M}\mathbf{u}_j^{p-1}(k)^T\mathbf{M}_j^T H\mathbf{M}_a + F^T(k,x^f_{a,ref})\mathbf{M}_a\right) {H_a^{-1}}^T\overline{H}_a H_a^{-1}\left(\sum_{j\neq a}^{M}\mathbf{u}_j^{p-1}(k)^T\mathbf{M}_j^T H\mathbf{M}_a + F^T(k,x^f_{a,ref})\mathbf{M}_a\right)^T$$

$$-\overline{F}_a^T(k)H_a^{-1}\left(\sum_{j\neq a}^{M}\mathbf{u}_j^{p-1}(k)^T\mathbf{M}_j^T H\mathbf{M}_a + F^T(k,x^f_{a,ref})\mathbf{M}_a\right)^T$$

Given that we address an optimization in terms of the false reference, we remove in the latter the terms that are not funcition of $x^f_{a,ref}$ and thus do not contribute to the solution.

$$\min_{x^f_{a,ref}} \quad \frac{1}{2}F^T(k,x^f_{a,ref})\mathbf{M}_a {H_a^{-1}}^T\overline{H}_a H_a^{-1}\mathbf{M}_a^T F(k,x^f_{a,ref}) +$$

$$\left(-{H_a^{-1}}^T\overline{F}_a(k) + {H_a^{-1}}^T\overline{H}_a H_a^{-1}\left(\sum_{j\neq a}^{M}\mathbf{u}_j^{p-1}(k)^T\mathbf{M}_j^T H\mathbf{M}_a\right)^T\right)^T \mathbf{M}_a^T F(k,x^f_{a,ref})$$

For a clearer notation, C_1 and C_2 are defined.

$$C_1 = \mathbf{M}_a {H_a^{-1}}^T\overline{H}_a H_a^{-1}\mathbf{M}_a^T, \qquad C_2 = \mathbf{M}_a\left(-{H_a^{-1}}^T\overline{F}_a(k) + {H_a^{-1}}^T\overline{H}_a H_a^{-1}\left(\sum_{j\neq a}^{M}\mathbf{u}_j^{p-1}(k)^T\mathbf{M}_j^T H\mathbf{M}_a\right)^T\right)$$

$$C_1 \in \mathbb{R}^{N\sum_i m_i \times N\sum_i m_i}, \qquad C_2 \in \mathbb{R}^{N\sum_i m_i \times 1}$$

$$x^{f*}_{a,ref} = \arg\min_{x^f_{a,ref}} \quad \frac{1}{2}F^T(k,x^f_{a,ref})C_1F(k,x^f_{a,ref}) + C_2^T F(k,x^f_{a,ref})$$

At this point, the partial derivative of $F(k,x^f_{a,ref})$ with respect to $x^f_{a,ref}$ is used for the application of the chain rule to find the desired one.

$$\frac{\partial\left(F(k,x^f_{a,ref})\right)}{\partial x^f_{a,ref}} = -\mathbf{P}_a^T G_x^T \widehat{Q} G_u$$

$$D = -\mathbf{P}_a^T G_x^T \widehat{Q} G_u, \qquad D \in \mathbb{R}^{n_a \times N\sum_i m_i}$$

With that,

$$\frac{\partial\left(\frac{1}{2}F^T(k,x^f_{a,ref})C_1F(k,x^f_{a,ref}) + C_2^T F(k,x^f_{a,ref})\right)}{\partial x^f_{a,ref}} =$$

$$\frac{\partial F(k,x^f_{a,ref})}{\partial x^f_{a,ref}}\frac{\partial\left(\frac{1}{2}F^T(k,x^f_{a,ref})C_1F(k,x^f_{a,ref}) + C_2^T F(k,x^f_{a,ref})\right)}{\partial F(k,x^f_{a,ref})} =$$

$$D\left(C_1F(k,x^f_{a,ref}) + C_2\right)$$

$$D\left(C_1F(k,x^f_{a,ref}) + C_2\right) \in \mathbb{R}^{n_a\times 1}$$

Finally, to arrive to the optimal desired value the latter should be equal to zero when $x^f_{a,ref} = x^{f*}_{a,ref}$.Therefore,

$$C_1 F(k, x^{f*}_{a,ref}) = -C_2$$

Assuming now the non singularity of the squared matrix C_1, it would be possible to state what follows.

$$F(k, x^{f*}_{a,ref}) = G_u^T \widehat{Q} G_x \left(x(k|k) - \sum_{i=1, i\neq a}^{M} \mathbf{P}_i x_{i,ref} - \mathbf{P}_a x^{f*}_{a,ref} \right), \qquad F(k, x^{f*}_{a,ref}) \in \mathbb{R}^{N\sum_i m_i \times 1}$$

$$F(k, x^{f*}_{a,ref}) = -{C_1}^{-1} C_2$$

The latter equation also shows consistency in the dimensions of the matrices, which can be seen as a good indication about the above procedure.

$$G_u^T \widehat{Q} G_x \mathbf{P}_a x^{f*}_{a,ref} = G_u^T \widehat{Q} G_x \left(x(k|k) - \sum_{i=1, i\neq a}^{M} \mathbf{P}_i x_{i,ref} \right) + {C_1}^{-1} C_2$$

$$b = G_u^T \widehat{Q} G_x \left(x(k|k) - \sum_{i=1, i\neq a}^{M} \mathbf{P}_i x_{i,ref} \right) + {C_1}^{-1} C_2, \qquad b \in \mathbb{R}^{N\sum_i m_i \times 1}$$

$$A = G_u^T \widehat{Q} G_x \mathbf{P}_a, \qquad A \in \mathbb{R}^{N\sum_i m_i \times n_a}$$

$$A x^{f*}_{a,ref} = b \tag{6.7}$$

The analytical resolution ends up in $N\sum_i m_i$ equations to determine n_a variables. This fact supposes a clear likehood that the system under consideration will be over- or under- determined. In case of overdetermination, when no exact solution exists, a possible approximation to the desired solution could be reached using the normal equations as it is shown below to conclude this section.

$$A x^{f*}_{a,ref} = b \longrightarrow x^{f*}_{a,ref} = (A^T A)^{-1} A^T b$$

6.3 Fake weights

In a *fake weights* attackt agent a conducts the introduction of false information through a change in the values that the weights factor λ_j ($j = 1,...,M$) take in its own optimization problem. This kind of attack is directly related with the definition of the objective functions ($\phi_{i,c}(\cdot)$) that determine the minimization problems for the agents the plant comprises ($i = 1,...,M$).

$$\phi_{i,c}(\mathbf{u}_i(k), \mathbf{u}^{p-1}_{j\neq i}(k); x(k|k)) = \sum_{l=1}^{M} \lambda_l \phi_l(\mathbf{u}_i(k), \mathbf{u}^{p-1}_{j\neq i}(k); x(k|k))$$

Then, a malicious agent can play with the possibility of altering the negotation process by means of a change of λ_j. An increase of the corresponding λ_a which weights $\phi_a(\cdot)$ in ($\phi_{a,c}(\cdot)$) entail that the input $\mathbf{u}^p_a(k)$ will tend to decline the cumulative cost of a rather than reaching the certain global performance defined by the original optimization.

Let's focus on the optimization problem to be solved by the attacker and assume that all λ_j for $j \neq a$ remain with their original values. Therefore, agent a will solve an optimization like the one depicted below.

$$\min_{\mathbf{u}_a} \ \phi_{a,c}(\mathbf{u}_a(k), \mathbf{u}^{p-1}_{j\neq a}(k); x(k|k)) = \sum_{l=1, l\neq a}^{M} \lambda_l \phi_l(\mathbf{u}_a(k), \mathbf{u}^{p-1}_{j\neq a}(k); x_l(k|k)) + \lambda^f_a \phi_a(\mathbf{u}_a(k), \mathbf{u}^{p-1}_{j\neq a}(k); x_a(k|k)) \tag{6.8}$$

where λ^f_a denotes the fake applied weight. Consequently, the interests of a will receive more importance in comparison to the real framework, as its individual cost funciton will be weighted by $\lambda^f_a > \lambda_a$. The rest of the agents will make an effort to provide a control action in favour of controller a so, as before, the attacker not only introduce false information, but expects the plant to cooperate to reach a solution inclined towards its preferences.

In the case in which λ_i are set to

$$\lambda_i = \frac{1}{M}, \qquad \forall i = 1,...,M$$

the redefinition which supposes the root cause for the steering of the negotiation can be clearer represented. Let use $\lambda = \frac{1}{M}$, then the optimization problem to solve by a will become

$$\min_{\mathbf{u}_a} \ \phi_{a,c}(\mathbf{u}_a(k), \mathbf{u}_{j\neq a}^{p-1}(k); x(k|k)) = \sum_{l=1,l\neq a}^{M} \phi_l(\mathbf{u}_a, \mathbf{u}_{j\neq a}^{p-1}; x_l(k)) + \frac{\lambda_a^f}{\lambda}\phi_a(\mathbf{u}_a, \mathbf{u}_{j\neq a}^{p-1}; x_a(k))$$

$$\frac{\lambda_a^f}{\lambda} > 1$$

The spreading mechanism over the whole system is the equivalent to the one described for a *false reference* attack, just with the difference that in this case $\mathbf{u}_{a,opt}^p(k)$ contains malicious information as a result of λ_a^f for all $p < p_{max}$ and time instant k. Therefrom the impact on $\mathbf{u}_{j,opt}^p(k)$ ($j \neq a$) and the possibility of steering the process towards a point in which the the solutions obatained are biased in favour of a.

6.3.1 Particular case: *Selfish agent*

Within this attack is the case in which a decides to optimize only its own subsystem, which then supposes an important particular scenario that should be analysed.

The subsystem a will be defined by A_a and the set of matrices B_{aj}. This case involves a selfish resolution of the optimization problem that solves the malicious controller through setting

$$\lambda_j = 0, \qquad \forall j \neq a$$

$$\lambda_a = 1$$

In other words, $\mathbf{u}_a^p(k)$ will be calculated irrespective of the cooperative feature of the DMPC scheme, solving in its case an optimization defined mathematically by the following objective objective function

$$\phi_{a,c}(\mathbf{x}_a(k), \mathbf{u}_a(k), \mathbf{x}_j^{p-1}(k), \mathbf{u}_j^{p-1}(k); x_a(k|k)) = \sum_{n=0}^{N} x_a^T(k+n|k)Q_a(k+n|k)x_a(k+n|k) + u_a^T(k+n|k)R_a(k+n|k)u_a(k+n|k)$$

Hence, the input arrived will be

$$\mathbf{u}_{a,opt}^p(k) = \arg\min_{\mathbf{u}_a} \ \phi_a(\mathbf{x}_a(k), \mathbf{u}_a(k), \mathbf{x}_j^{p-1}(k), \mathbf{u}_j^{p-1}(k); x_a(k|k))$$

The selfish problem is directly rewritten as

$$\begin{aligned} \mathbf{u}_{a,opt}^p(k) = \arg\min_{\mathbf{u}_a} \quad & \frac{1}{2}\mathbf{u}_a(k)^T \left(G_{ua}\widehat{Q}_a G_{ua} + \widehat{R}_a\right)\mathbf{u}_a(k) + \\ & \left(\left(x_a(k|k) - x_{a,ref}\right)^T G_{xa}^T \widehat{Q}_a G_{ua} + \mathbf{w}_a^{p-1}(k)^T G_{wa}^T \widehat{Q}_a G_{ua}\right)^T \mathbf{u}_a(k) \end{aligned} \tag{6.9}$$

This particular case supposes an application of a *fake weights* attack taken to the extreme, which might induce a fail of the global DMPC scheme performance. This aspect is one of the manifestations of the interactions and influence between agents in the plant, which entail that an improvement from a local perspective does not involve an improvement for the overall system.

6.4 Fake constraints

In this case, the original DMPC framework is threated by means of modifying the constraints imposed on the state's and input's evolution related to subsystem a. It raises the possibility of leading the system to a more advantageous situation for agent a when changing the intervals in which its controls and states can take

values. Mathematically, if we denote the sets that define the constraints for each subsystem i as $\mathscr{X}_i$ and $\mathscr{U}_i$, we could represent this attack for all $n = 0,1,...,N$ and for all k as

$$x_a(k+n|k) \in \mathscr{X}_a^f$$

$$u_a(k+n|k) \in \mathscr{U}_a^f$$

where $\mathscr{X}_a^f$ and U_a^f are the sets after the introduction of the misleading information.

The solution of every optimization problem performed by a is done subject to certain constraints previously defined, consequently, an ease or tightening of them cause changes in the results provided. Therewith, we are in a similar situation as in the attacks above in the sense that the rest of agents j ($j \neq a$) will optimize a function affected by the corresponding malicious information that a shares. It is important to underline here that in the previously presented attacks all minimization problems for each i are solved under the conditions imposed by $\mathscr{X}_i$ and $\mathscr{U}_i$, although it has not been explicitely specified for simplicity.

A *fake constraints* attack differs with regard to the others in where the root of the threat is. In other words, either in a *fake constraints* or a *false reference* attack agent a carry out a modification of its objective function to optimize, and that is the starting point to make a profit for itself. However, in this case $\phi_{a,c}(\cdot)$ remain the same as firstly proposed and the way of action is through the conditions imposed in its resolution.

Given that, it is consistent that an ease of $\mathscr{X}_a$ and $\mathscr{U}_a$ at iteration p and time instant k might benefit indirectly the attacker but the real effect on the algorithm would be that agent a would dispose of greater sets to find the solution $\mathbf{u}_{a,opt}^p(k)$ that is globally better. So that, it could be said that a acts out of malice. If the condition that the attacker knows everything about the evolution of the entire plant is assumed, then a clear opportunity to take advantage from the others would arise. This fact implies that a would have the knowledge to determine what is the region where the inputs that would cause the greatest decline of its cost are located. Therefore, this information could be used in the determination of $\mathbf{u}_{a,opt}^p(k)$ by stablishing the fake constraints that would guide itself to the more beneficial solutions. By doing so the attacker's malice would become a indisputable fact. Once again, the procedure for transfering the false information is the same as for the previous attacks, therefore the false information is extended throughout the global system.

In order to show the results after the implementation of the DMPC algorithm that has been arrived theoretically in the preceding parts, later some examples will be presented. The purpose will be going into the performance of the algorithm from a practical point of view, as well as considering the risks and consequences that suppose for them acting under the effects of the attacks described above.

6.5 A bidimensional depiction

To conclude the attack's presentation, we show here in a visual way the purpose and effects that have been described.

Setting the prediction length to $N = 1$ and the number of subsystem to $M = 2$ allows us to represent clearly in the plane $u_1^p(k)/u_2^p(k)$ all that happens along iterations and thus how the attacker manages to steer the agreement for its own benefit. With that parameters, the optimal trajectories calculated each p by the agents 1 or 2 becomes a single number, so that a point in the plane $u_1^p(k)/u_2^p(k)$. Moreover, the the cost functions are then defined in three dimensions with the possibility of representing their level curves in the mentioned plane. By doing so, we are going to present a short analysis of the different kinds of attacks disscussed.

Figures 6.1 to 6.3 show the results for a false reference, fake weights and fake constraints attacks respectively. The effectiveness of the attack is reflected in the tendence of the points $(u_1^p(k),u_2^p(k))$ to those curves which represent lower local cost por the attacker, which is in this graphics is 1. This aspect is shown in all of them as the malice information has been chosen in order to meet this purpose. That is, in the case of a fake constraints attack, the set $\mathscr{U}_1^f$ has been defined as a modification of the original one to another that includes those values of $u_1^p(k)$ associated with lower costs for 1 and the initial $u_1^0(k)$. Equally, when acting by means of $x_{ref}^f = constant$, this parameter has been chosen to misdirect appropiately the negotation.

The other important feature to note is the difference arrived when the malicious controller also optimizes the misleading parameter which is the root of the problem or either acts in a complete selfish way. These cases lead to a much more notable deviation of the inputs agreed with respect to the normal development of the algorithm.

Moreover, as it can be seen, performing under reliable conditions will end up with a solution which does not privilege any subsystem over the others but is chosen in favour of the global behaviour.

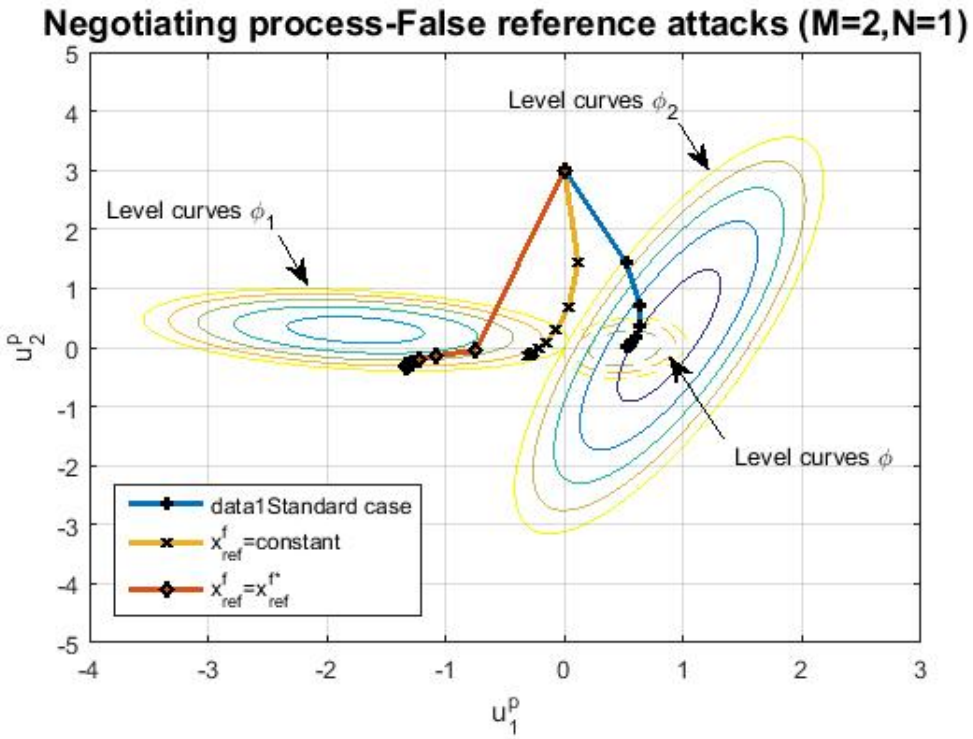

Figure 6.1 Trajectories in $u_1^p(k)/u_2^p(k)$ for false reference attacks.

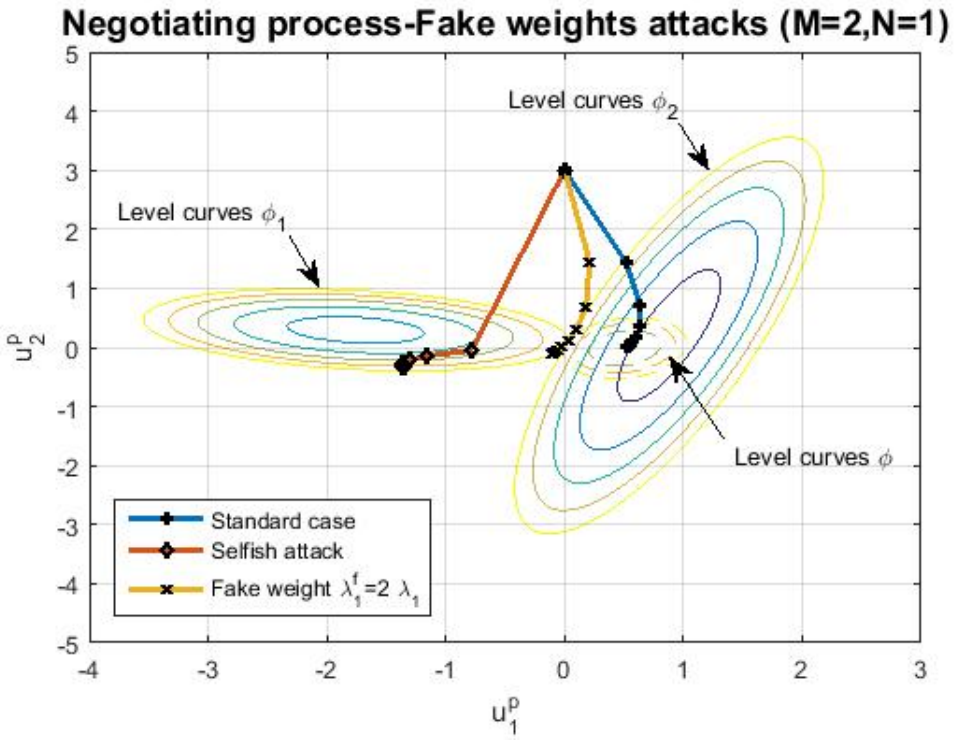

Figure 6.2 Trajectories in $u_1^p(k)/u_2^p(k)$ for fake weights attacks.

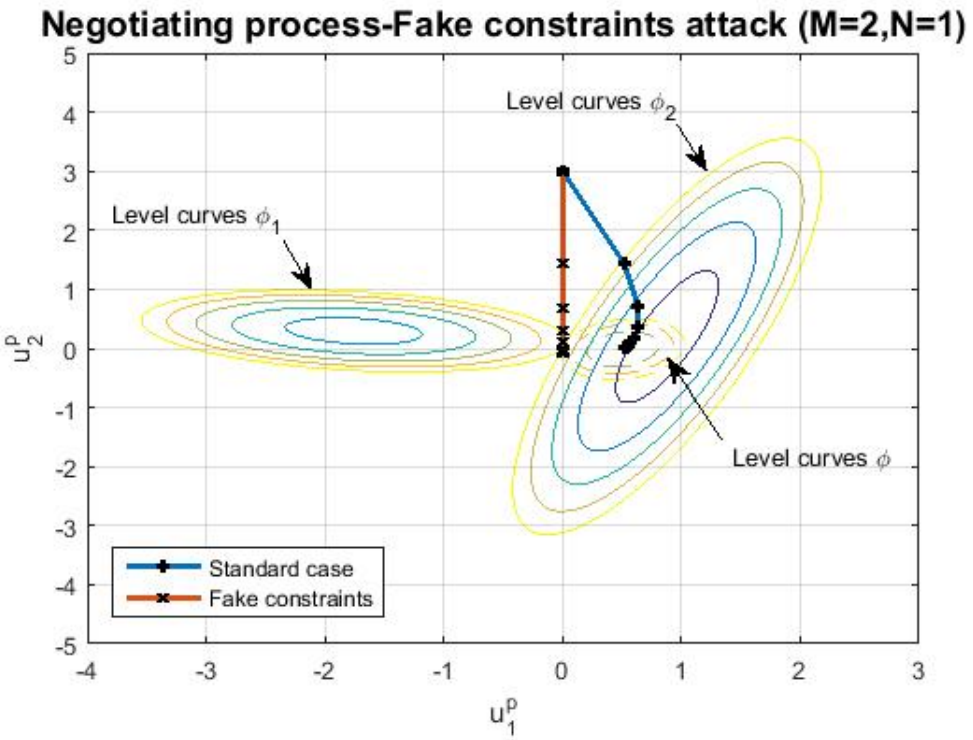

Figure 6.3 Trajectories in $u_1^p(k)/u_2^p(k)$ for fake constraints attacks.

6.6 Key Performace Indicator

This project focuses on problems characterized by the presence of multiple agents which interact with each other and which are controlled in a distributed way. The algorithm proposed supports a strategic behaviour based on collaborative settings, however it has been outlined the possibility of dealing with self-interested agents which, in different manner, will affect negatively the reliability of the information that is transferred during the development of the algorithm. Own independent objectives are provided to each of the agents and their way of working will determine the global behaviour of the system as a whole. The objective of this part is the assessment of the consequences of introducing false information, in a manner that a certain number, which will be a reflection of the severity of the attack, will be associated. Given that this number will provide information with which the behaviour can be assessed, it will be used as *Key Performance Indicator* (KPI).

Hereafter, the standard case will be pointed out repeatedly. For that reason it is important to indicate before that it is referred to that case in which the algorithm presented in 4.4 is developed being guaranteed that there is no false information circulating among the agents.

6.6.1 Price of Anarchy

Performance loss caused by a lack of coordination can be evaluated using the so-called Price of Anarchy (PoA), the ratio between the objective function value of an equilibrium and that of an optimal outcome. For its application here, we will assume that the mentioned optimal outcome will be the results obtained with centralized MPC. The PoA will be a reflection of the change introduce with the cooperation, hence values above 1 will be expected.

$$Price\ of\ Anarchy = \frac{\text{Cumulative cost when all act individually}}{\text{Cumulative cost Centralized MPC}} \tag{6.10}$$

6.6.2 Price of Corruption

To assess the impact of each of the attacks presented we have defined a ratio with a similar background as the PoA. This number will depict the consequences of the cyber-security faults in the algorithm in hand, and will be denoted as *Price of Corruption* (PoC).

Now, the values are referred to those arrived in the standard case as the objective is the study of the algorithm in different situations.

$$Price\ of\ Corruption = \frac{\text{Cumulative cost with presence of attack}}{\text{Cumulative cost in the standard case}} \tag{6.11}$$

A value of the $PoC > 1$ will imply a fall in the optimality concerning the global system. Therefore, it is expected that in case of attack this ratio will reach values above 1, being greater for those which cause more damage to the plant.

6.6.3 Effectivity of the attacks

Despite the latter, the real purpose of an attacker a is not an increase of the PoC but a decrease of its cumulative cost concerning its corresponding in the standard case. The probably resulting increment of the PoC is consequence of that this objective is achieved at expense of the rest of the agents. Given that, we could also define a ratio to assess the reduction when acting in a misleading way by the point of view of the attacker.

$$Attack's\ effectivity = \frac{\text{Cumulative cost agent } a \text{ with attack}}{\text{Cumulative cost in the standard case of } a} \tag{6.12}$$

6.7 Min-max approach

Hereafter, let h denote a determined honest agent of the considered system ($h \in \{1, M\}$). As a result of the couplings among subsystems, every agent h is influenced by decisions adopted by outer agents $j \neq h$. Hence, if h wants to determine its state, it will need external information to itself, which leads to the fact that there is always uncertainties about the variables that are not directly controlled by h but have a considerable effect on it. The presence of any malicious agent a will cause a deviation of the expected results in a manner that a

loss of performance is observed. The main purpose now is the study of the possibility of using min-max strategy in order to cope with these attacks.

6.7.1 Detecting the attack

We see the problem now from the point of view of an honest agent and see how we could protect them and avoid that at least certain harmful situations are reached. The first step is therefore to detect when a certain h is being attacked. To this end, the couplings will be the key principle to define a condition to be used by the honest agent in order to realize that someone is taking advantage of itself.

One of the first possibilities for dealing with uncertainties is considering the worst possible case. So that, when working with cost functions, the optimized control will be accordingly to the maximum value that the cost fuction could take due to the uncertainties. In other words, the problem results in a maximization followed by a minimization. The strategies in MPC that are based in the resolution of a min-max optimization are know as Min-Max MPC. Despite the availability of a wide range of variants concerning the latter, its application in this project does not go beyond the consideration of the information received by an outer subsystem within the plant as disturbance and then solving a problem for the worst possible case that could come about.

In the proposed algorithm, we have that the optimization problem to be solved each iteration p by a certain agent i is defined by the objective function (6.13).

$$\phi_{i,c}(\mathbf{u}_i(k),\mathbf{u}_j^{p-1}(k);x(k|k)) = \frac{1}{2}\mathbf{u}_i^T(k)\left(\mathbf{M}_i^T H\mathbf{M}_i\right)\mathbf{u}_i(k) + \left(\sum_{j\neq i}^{M}\mathbf{u}_j^{p-1}(k)^T\mathbf{M}_j^T H\mathbf{M}_i + F(k)^T\mathbf{M}_i\right)\mathbf{u}_i(k) \quad (6.13)$$

Let's assume that we are at time index k, so that the values are $x_l(k|k)$ for $l = 1,...,M$ are parameters known. The negotiating process seen by the point a view of agent $i = h$ is such that the rest of the plant set their proposed inputs $\mathbf{u}_j(k)$ and transfer these proposals to h, which make use of this information for the calculation of its own input.

The fact that determining the inputs for h in such a way that acting for the plant leads to a situation that is worse than the worst one which would be arrived when acting independently is a clear incentive to think that at least there is something wrong. In this case, h would reach an undesired situation caused by acting for the plant without receiving nothing in return.

To apply this idea and be able to compare both costs, we have to go back to the local cost function for any agent i, which has been is (6.14).

$$\phi_i(\mathbf{u}(k);x(k|k)) = \frac{1}{2}\mathbf{u}_i^T(k)\left(G_{ui}^T\widehat{Q}_i G_{ui}\right)\mathbf{u}_i(k) + \left(x_i(k|k)^T G_{xi}^T\widehat{Q}_i G_{ui} + \mathbf{w}_i(k)^T G_{wi}^T\widehat{Q}_i G_{ui}\right)\mathbf{u}_i(k) \quad (6.14)$$

where

$$G_{wi}\mathbf{w}_i(k) = \sum_{j=1,j\neq i}^{M}\left[G_{wi}^{x_j}x_j(k|k) + G_{wi}^{u_j}\mathbf{u}_j(k)\right]$$

In case h acts independently, in the worst possible situation its cost would be the one defined by

$$\min_{\mathbf{u}_h(k)}\ \max_{\mathbf{w}_h(k)}\ \frac{1}{2}\mathbf{u}_h^T(k)\left(G_{uh}^T\widehat{Q}_h G_{uh}\right)\mathbf{u}_h(k) + \left(x_h(k|k)^T G_{xh}^T\widehat{Q}_h G_{uh} + \mathbf{w}_h(k)^T G_{wh}^T\widehat{Q}_h G_{uh}\right)\mathbf{u}_h(k) \quad (6.15)$$

The values that $\mathbf{w}_h(k)$ can take at every iteration is limited by the constraints imposed, so what is done here is assuming that this vector is inside a certain set in $\mathbb{R}^{Nn_h\times 1}$.

$$w_h(k) \in \bigoplus_{j\in\{1,M\}\setminus\{h\}} A_{hj}\mathscr{X}_j \oplus B_{hj}\mathscr{U}_j$$

The presented above can be summarized in the following condition. It supposes a very conservative condition but which leads to a point in which clearly something should be done. In short, if the following

inequation is true, it will be an indicator to act in a different way as expected.

$$\min_{\mathbf{u}_h(k)} \max_{\mathbf{w}_h(k)} \frac{1}{2}\mathbf{u}_h^T(k)\left(G_{uh}^T\widehat{Q}_hG_{uh}\right)\mathbf{u}_h(k)+\left(x_h(k|k)^TG_{xh}^T\widehat{Q}_hG_{uh}+\mathbf{w}_h(k)^TG_{wh}^T\widehat{Q}_hG_{uh}\right)\mathbf{u}_h(k) \tag{6.16}$$

$$< \min_{\mathbf{u}_h(k)} \frac{1}{2}\mathbf{u}_h^T(k)\left(\mathbf{M}_h^TH\mathbf{M}_h\right)\mathbf{u}_h(k)+\left(\sum_{j\neq h}^{M}\mathbf{u}_j^{p-1}(k)^T\mathbf{M}_j^TH\mathbf{M}_h+F(k)^T\mathbf{M}_h\right)\mathbf{u}_h(k)$$

6.7.2 Response to the attack

As it has been seen in the preceding parts, the agents can be steered towards a different and much worse situation than the one they should arrive when working under standard conditions. The evolution with presence of attacks is not only determined by the attacker but by all the decisions agreed in the plant, so that we can see the attacker a just as the root cause of the loss of performance. Therefore, we have that every h has the chance of parcipating in the negotiating process in such a way that its optimized inputs will not be in favour of agent a or diminish the strength of the attack.

If (6.16) holds at iteration p and time index k, it would be better to forget about he cooperative aspect and solve the problem just from its local point of view to avoid higher costs. In other words, in this case agent h would optimize its input as:

$$\mathbf{u}_{h,opt}^p(k)=\arg\min_{\mathbf{u}_h(k)} \frac{1}{2}\mathbf{u}_h^T(k)\left(G_{uh}^T\widehat{Q}_hG_{uh}\right)\mathbf{u}_h(k)+\left(x_h(k|k)^TG_{xh}^T\widehat{Q}_hG_{uh}+\mathbf{w}_h^{p-1}(k)^TG_{wh}^T\widehat{Q}_hG_{uh}\right)\mathbf{u}_h(k) \tag{6.17}$$

It is important to reiterate the conservatism of this approach in order to be well protected against the effects of false information. When it is implemented, it only causes changes when the situation represented by (6.16) is reached, which for its part implies expecting the worst from an outer controller. Given that this limit situation could not be the case even when an attacker is performing effectively, the fact that an agent is protected with this method does not imply that it would not see a poorer performance.

7 Example 1: Two double integrators with coupled inputs

In this first example, a plant composed of 2 subsystems ($M = 2$) and a prediction time length of 5 ($N = 5$) have been considered. Firstly, we present the matrices of the model (2.3) that will describe the performace of each subsystem. Along with this, we will have the associated centralized model, whose matrices are also shown below.

As specification of the problem, some constraints on the states and inputs will be imposed in the resolution and a certain reference will be provided. Finally, an initial state has to be fixed, due to the fact that the values of certain variables at the previous iteration are used in the algorithm, so it will be necessary for the start of the simulation.

$$A_{11} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \qquad B_{11} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad B_{12} = \begin{bmatrix} 0 \\ 0.4 \end{bmatrix} \qquad n_1 = 2, \qquad m_1 = 2$$

$$A_{22} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \qquad B_{21} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad B_{22} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad n_2 = 2, \qquad m_2 = 2$$

$$A_{cen} = \begin{bmatrix} 1 & 0 & & \\ 1 & 1 & & \\ & & 1 & 0 \\ & & 1 & 1 \end{bmatrix}, \qquad B_{cen} = \begin{bmatrix} 0 & 0 \\ 1 & 0.4 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \qquad n_x = 4, \qquad m_u = 2$$

Constraints:

$$\begin{bmatrix} -2 \\ -2 \end{bmatrix} \leq x_1 \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \qquad \begin{bmatrix} -2 \\ -2 \end{bmatrix} \leq x_2 \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \qquad |u_1| \leq 5, \qquad |u_2| \leq 5$$

Initial state:

$$x_1(0) = \begin{bmatrix} 0.1 \\ -0.2 \end{bmatrix}, \qquad x_2(0) = \begin{bmatrix} -0.1 \\ 1 \end{bmatrix}$$

In addition, the weighting matrices that will define the cost functions to be optimized are

$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad Q = \begin{bmatrix} 1 & 0 & & \\ 0 & 1 & & \\ & & 1 & 0 \\ & & 0 & 1 \end{bmatrix}$$

$$R_1 = 0.1, \qquad R_2 = 0.1 \qquad R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

With all this data, and the previous coding of $\widehat{A}_x$, $\widehat{A}_u$, $\widehat{b}_x$ and $\widehat{b}_u$, we proceed to the resolution of the centralized problem, and the subsequent application of the change of variable to solve it in a distributed way.

Given that this double integrator is used as introductory example, we are going to detail some of the particularized DMPC algorithm's mathematical expressions. For the specific case of $M = 2$, the change

of variable will be $\mathbf{u} = \mathbf{M}_1\mathbf{u}_1 + \mathbf{M}_2\mathbf{u}_2$, which leads us to the following optimizations problems, that are respectively the one that will be addressed by agent 1, and the one for 2.

$$\min_{\mathbf{u}_1} \frac{1}{2}\mathbf{u}_1(k)^T \left(\mathbf{M}_1^T H \mathbf{M}_1\right)\mathbf{u}_1(k) + \left(\mathbf{u}_2(k)^T \mathbf{M}_2^T H \mathbf{M}_1 + F^T \mathbf{M}_1\right)\mathbf{u}_1(k)$$

$$\min_{\mathbf{u}_2} \frac{1}{2}\mathbf{u}_2(k)^T \left(\mathbf{M}_2^T H \mathbf{M}_2\right)\mathbf{u}_2(k) + \left(\mathbf{u}_1(k)^T \mathbf{M}_1^T H \mathbf{M}_2 + F^T \mathbf{M}_2\right)\mathbf{u}_2(k)$$

They will be solved iteratively as it has been described, considering the constraints

$$\begin{bmatrix} \widehat{A}_x G_u \mathbf{M}_1 \\ \widehat{A}_u \mathbf{M}_1 \end{bmatrix} \mathbf{u}_1(k) \leq \begin{bmatrix} \widehat{b}_x - \widehat{A}_x G_x x(k|k) - \widehat{A}_x G_u \mathbf{M}_2 \mathbf{u}_2(k) \\ \widehat{b}_u - \widehat{A}_u \mathbf{M}_2 \mathbf{u}_2(k) \end{bmatrix}$$

$$\begin{bmatrix} \widehat{A}_x G_u \mathbf{M}_2 \\ \widehat{A}_u \mathbf{M}_2 \end{bmatrix} \mathbf{u}_2(k) \leq \begin{bmatrix} \widehat{b}_x - \widehat{A}_x G_x x(k|k) - \widehat{A}_x G_u \mathbf{M}_1 \mathbf{u}_1(k) \\ \widehat{b}_u - \widehat{A}_u \mathbf{M}_1 \mathbf{u}_1(k) \end{bmatrix}$$

The maximum number of iterations has been fixed to $p_{max} = 100$ and the value of the parameter ε has been 0.01.

Regarding this example, it is important to notice that we are dealing with a non controllable system which can be easily observe in the fact that the first components of x_i for $i = 1,2.$ will remain unaltered during the entire simulation length at the value of $x_i(0)$. More properly, the contrability property has been tested via the corresponding rank of the subsystem's controllability matrices, or equally, with the centralized model, for which this rank is 2. Therefore, this system will not allow us to move all componets of the state x from any initial state to another final one in a finite time by acting on the inputs. Given that the purpose of this example is to assess the performance of the algorithm when it is implemented, we will not go beyond to what is necessary for this aim. In other words, some parameters are going to be fixed with view to a clearly presentation of this behaviour rather than going into an specific control problem of the system.

The mentioned first states' components cause that a constant term K_i might be added each k to the corresponding cumulative cost and moreover it will affect the objective functions to optimize.

$$K_i = \sum_{n=0}^{N-1} (x_i^1(0) - x_{i,ref}^1)^2$$

To avoid a bias of the negotiation process just due to a choice of the initial state and the reference, it has been set

$$x_{1,ref} = \begin{bmatrix} 0.1 \\ 1.5 \end{bmatrix}, \qquad x_{2,ref} = \begin{bmatrix} -0.1 \\ -1 \end{bmatrix}$$

However, despite not affecting the cost functions, the agents should consider the influence in the controllable component of the states of this pecularity.

7.1 Standard case with Example 1

The results obtained solving the parallel cooperation-based optimizations (without being influenced by any possible attacks) are presented in Figures 7.1 to 7.5.

In the first figure, inputs u_1 and u_2 over a simulation length of 25 time units can be seen. The coupling between the two subsystems of this example is caused by the influence of the input of one on the other, hence both evolutions are decisive in any state change, either for 1 or 2. The algorithm leads to a sequence of control actions implemented that barely differ from the one applied when controlling the system in a centralized way, which supposes a meaningful aspect with regard to the good performance of the negotation process. We have to remark here that the control is achieved through a receding horizon implementation of the decision variables at the last iteration per sampling time. It implies that just the first input of the trajectory calculated is the one implemented and represented in the figures. That is, $u_1(k|k)$ and $u_2(k|k)$ at the iteration p which fulfils either $p > p_{max}$ or $dist_1$ and $dist_2$ smaller than ε.

Figures 7.2 and 7.2 shows the state representation. The notation x_i^c means component c of the state vector x_i. As the number of states of each subsystem equals 2, it shows four different curves, in which the application of the control law, together with the intrinsic properties of the model equations above, are reflected. Both

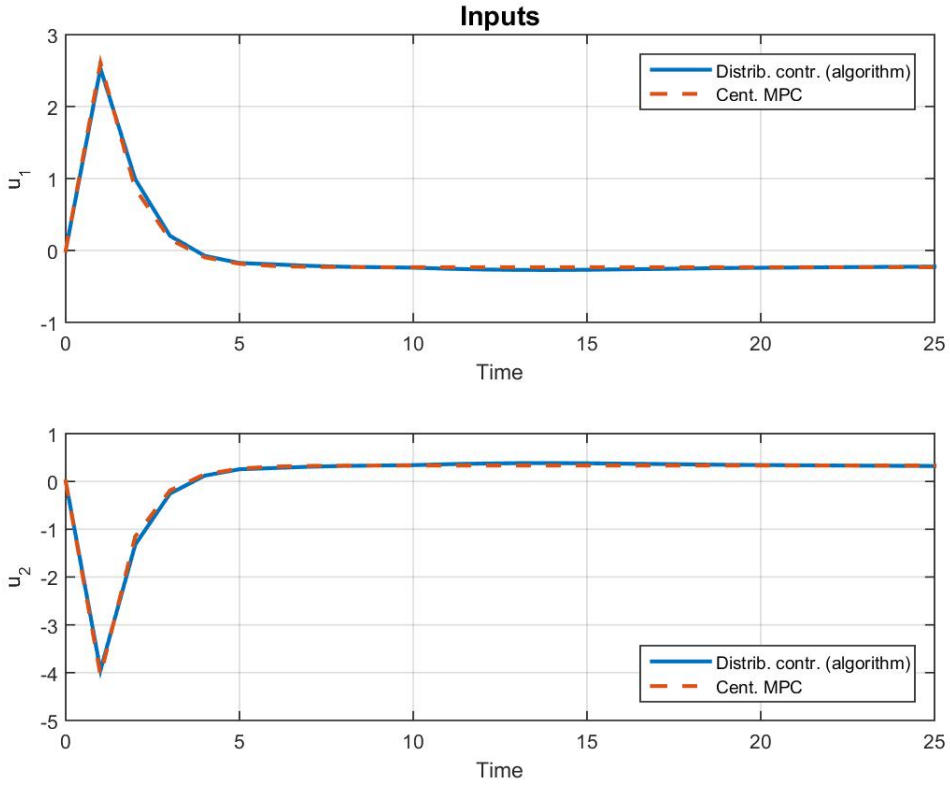

Figure 7.1 Inputs evolution (standard case).

subsystem's first components stay constant at the corresponding initial value while the other present an evolution towards values relatively closed to the reference. This example allow us to present the differences that can be achieved by changes in the stop conditions, that is, the possibility of making the algorithm more or less accurate. To illustrate it, it is shown in Figure 7.4 the change in in the curves of x_1^2 and x_2^2 when just ε is redefined as 0.1.

Finally, in Figure 7.5 shows the corresponding cumulative costs for subsystems 1 and 2 for a further comparison when the attacks are introduced. These curves will be used as references to see the effect that the mentioned attacks will have on them. In principle, the cumulative cost for 1 derived from the application of the algorithm without any threat of malicious information has resulted to be around 2 when $k = 25$.

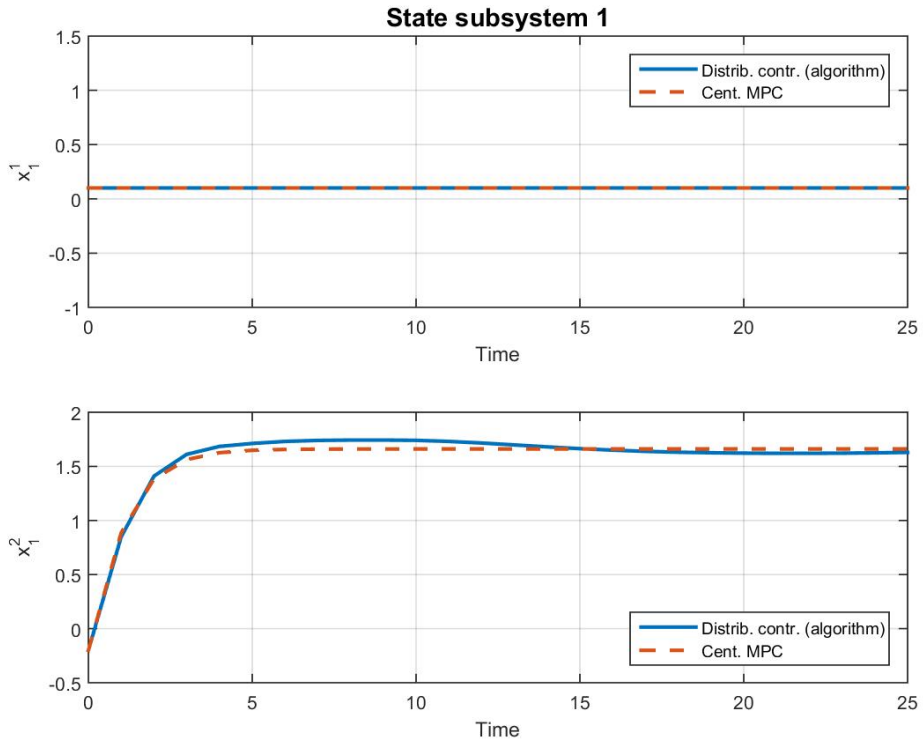

Figure 7.2 State evolution of subsystem 1 (standard case).

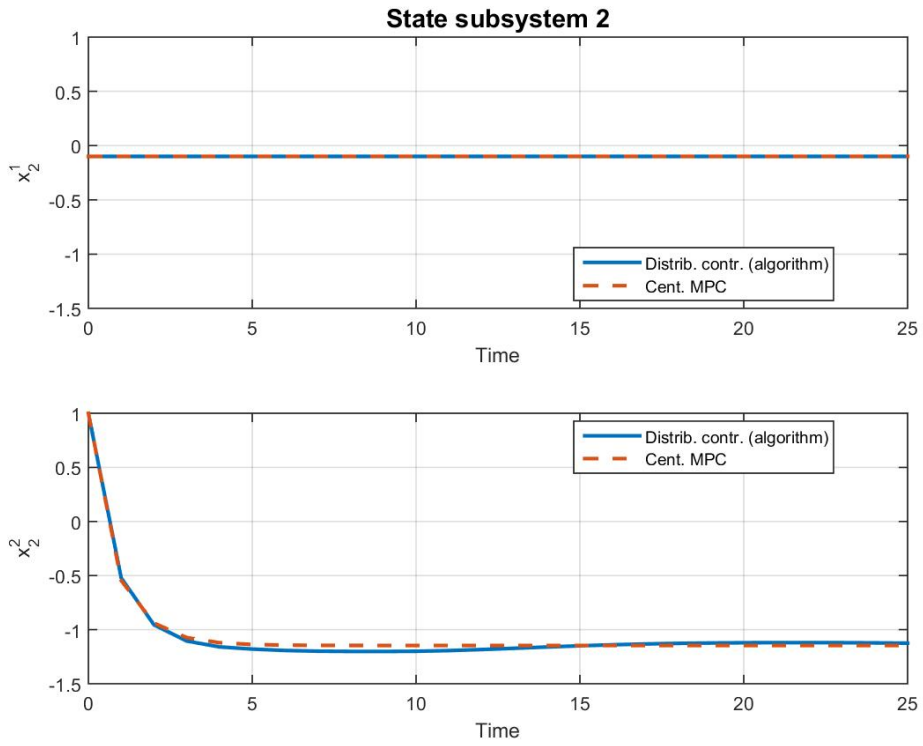

Figure 7.3 State evolution of subsystem 2 (standard case).

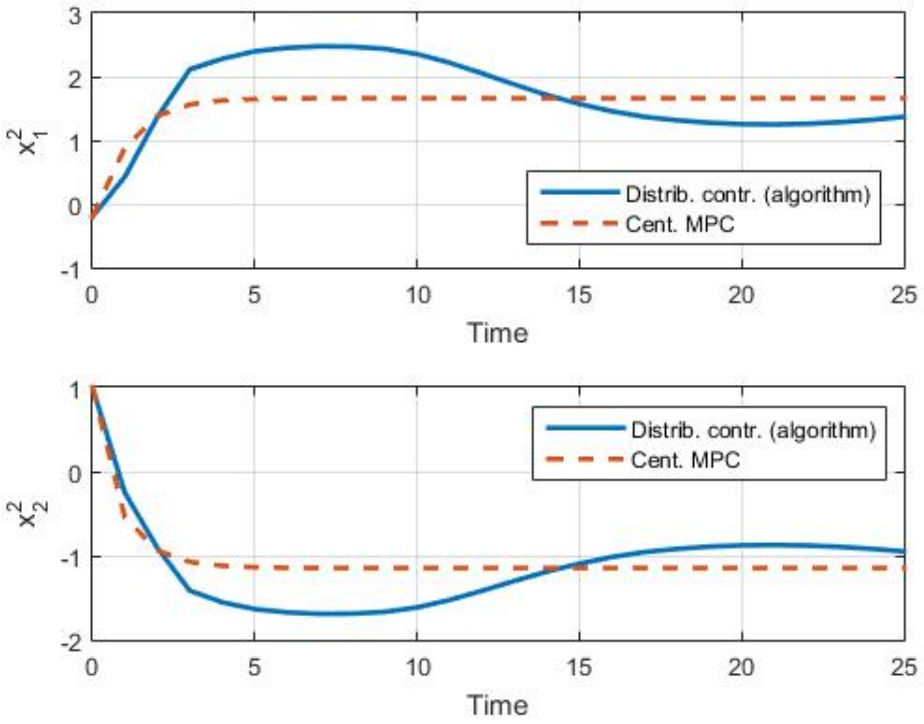

Figure 7.4 x_1^2 and x_2^2 evolutions when $\varepsilon = 0.1$ (standard case).

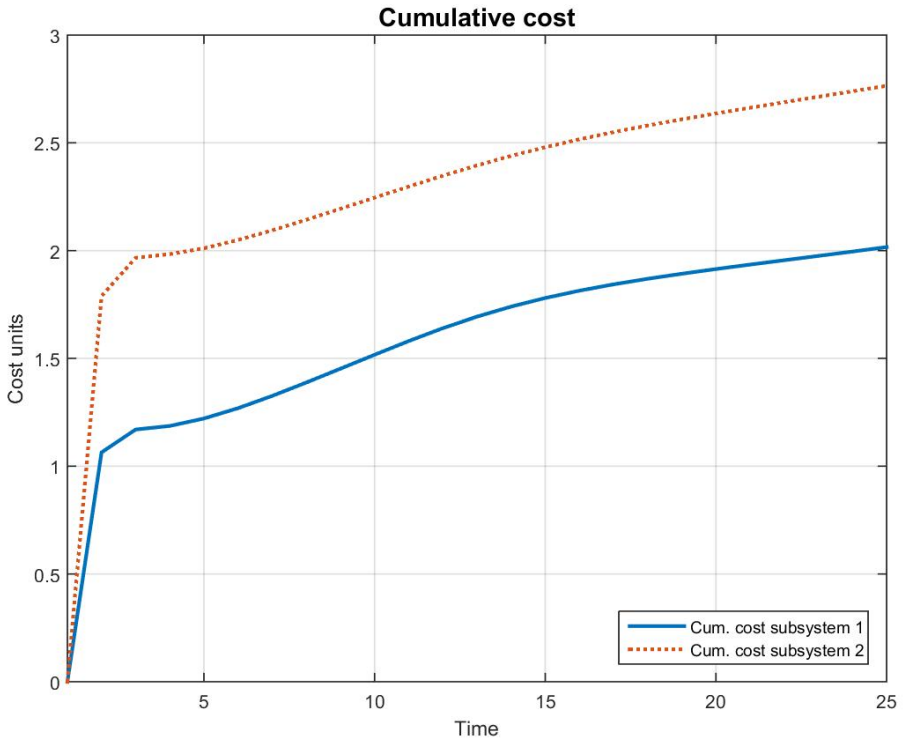

Figure 7.5 Cumulative cost for subsystems 1 and 2 (standard case).

7.2 *False reference* attack to Example 1

In this firs attack's presentation, we address the issue of introducing a false reference in the normal course of the algorithm. It has been considered that agent 1 is a malicious controller which introduces a misleading reference denoted as $x^f_{1,ref}$, so $a = 1$. Given the intended purpose of every attack, that is, the aim of taking advantages from the others to get benefits for the attacker, the implementation of this false reference attack has been done in conjunction with the determination of a certain $x^f_{1,ref}$ which leads to the desired goal. The latter has been done by means of the described method in the precending parts to determine the optimal $x^{f*}_{1,ref}$. So that it can also be analysed whether the performance obtained when this method is implemented here is satisfactory or not.

We have arrived the results presented in Figures 7.6 to 7.9, in which Figure 7.9 takes special importance as shows the difference in the evolution of the cumulative cost. It reflects how the misleading information has been introduced in favour of agent 1 and how this fact has made itself be better off from its local point of view. For now, it can be seen the pursuit of an appropiate $x^{f*}_{1,ref}$ has been done effectively as it fulfills the attacker's goal.

An important difference regarding the standard case can be observed in the state evolution of 1 towards the reference. To conclude the presentation of results related to this attack, it has been added the graphic with the values that the components of $x^f_{1,ref}$ take along time k, that is, the ones that affect directly the computing of the inputs impleted each time instant (Figure 7.10).

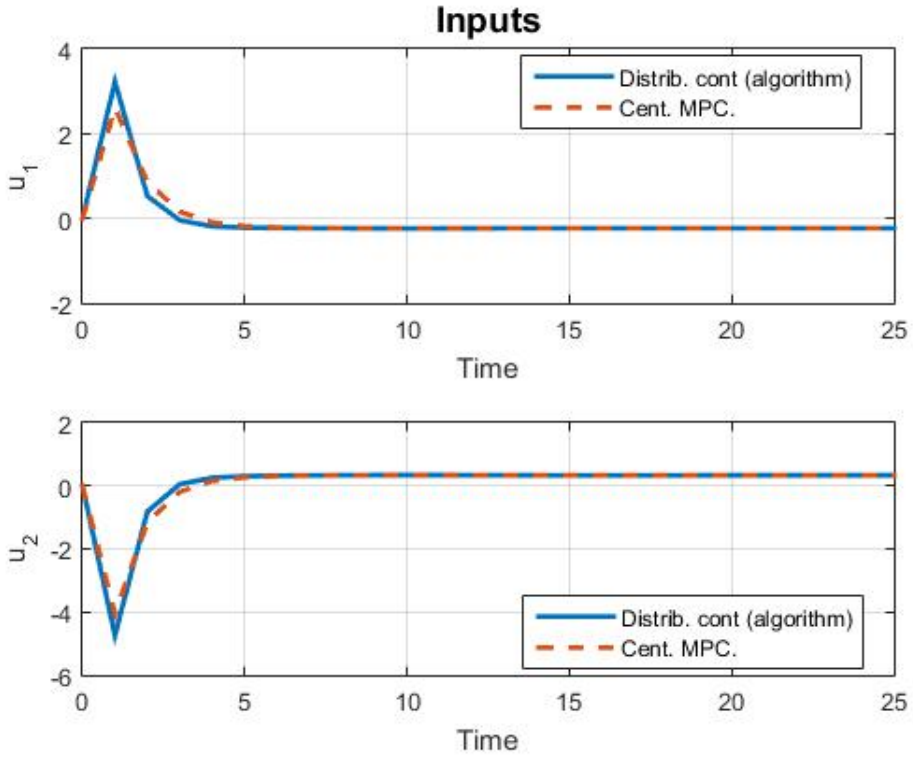

Figure 7.6 Inputs evolution ($x^f_{1,ref} = x^{f*}_{1,ref}$).

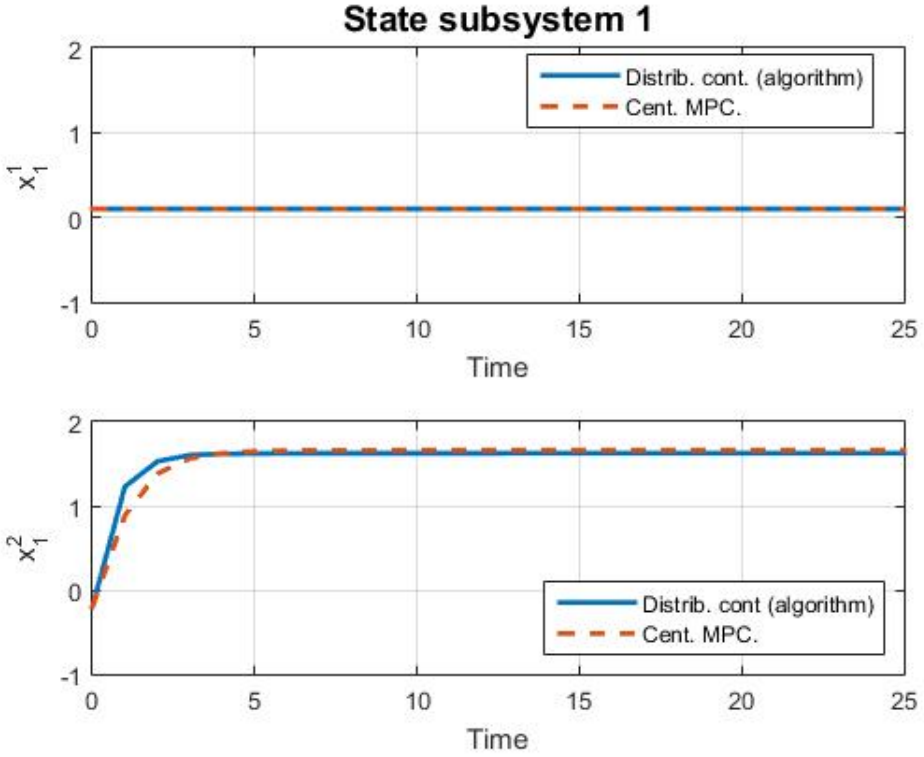

Figure 7.7 States evolution of subsystem 1 ($x^f_{1,ref} = x^{f*}_{1,ref}$).

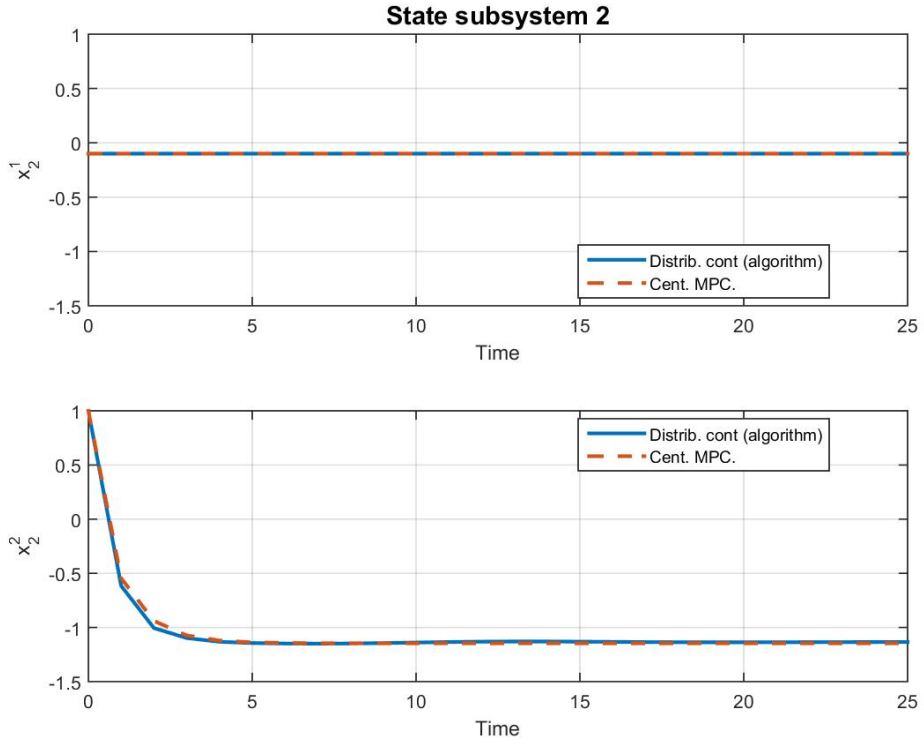

Figure 7.8 States evolution of subsystem 2 ($x^f_{1,ref} = x^{f*}_{1,ref}$).

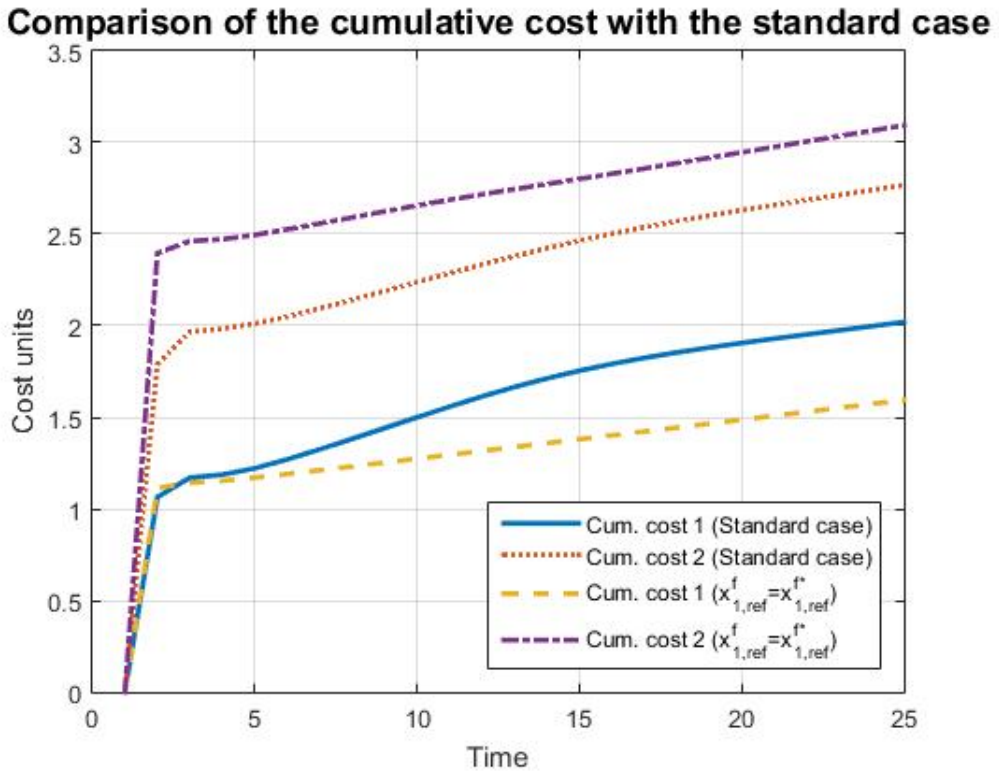

Figure 7.9 Comparison with the standard case ($x^{f}_{1,ref} = x^{f*}_{1,ref}$).

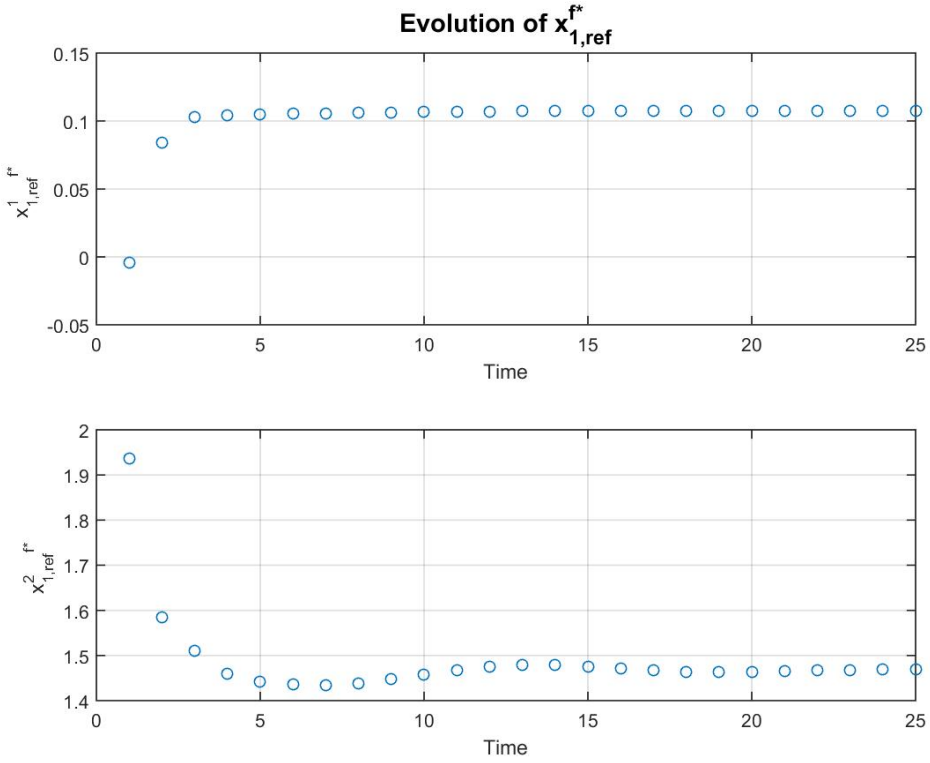

Figure 7.10 Evolution of $x^{f*}_{1,ref}$.

7.3 *Fake weights* attack to Example 1

Hereunder, we address the issue of having a fake weights attack in the plant, which will be again carried out by agent 1. We are going to analyse the response of the double integrator to this attack from two different perspective. On the one hand, we will present the results that will be arrived for the case in which the value of the corresponding λ_1 in the attacker's optimization problem is changed to $\lambda_1^f > \lambda_1$ while λ_2 remains with its original value. On the other hand, they will be compared with the ones arrived for the particular case related to this attack, that is, when agent 1 acts in a complete selfish way by removing the consideration of subsystem 2 in its objective funtion ($\lambda_2^f = 0$).

7.3.1 $\lambda_1^f = 1.5\lambda_1$

Setting $\lambda_1^f = 1.5\lambda_1$ and applying the attack as it was described, leads to the results presented in Figures 7.11 to 7.14. The enhancement that controller 1 makes for itself can be reflected in the last of these figures, however, in view of the result of the previous attack, there is still scope to improve.

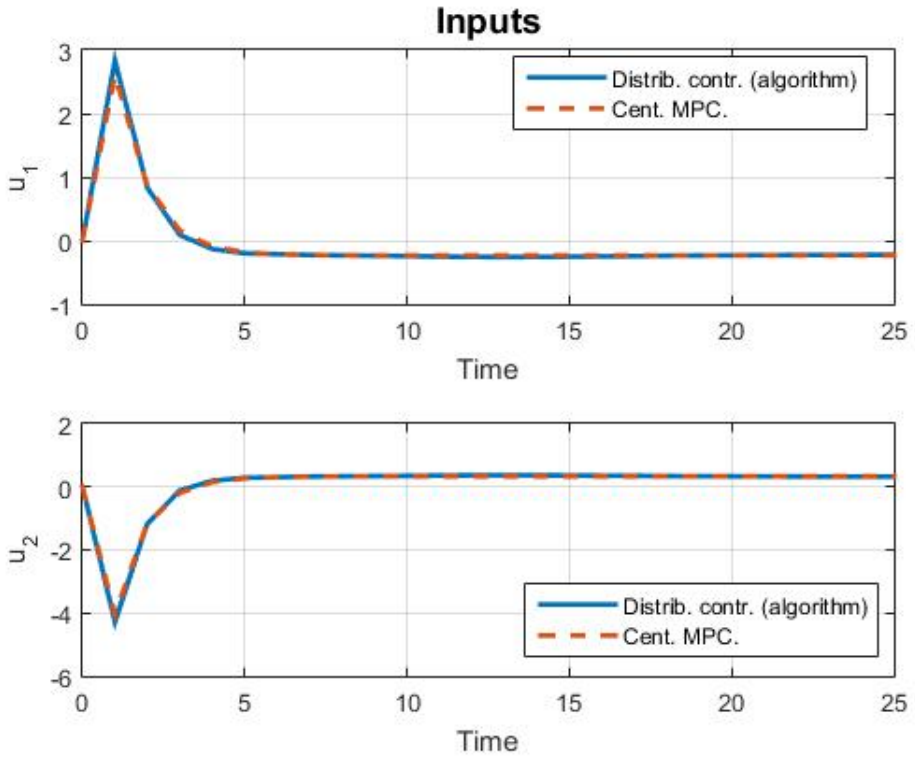

Figure 7.11 Inputs evolution ($\lambda_1^f = 1.5\lambda_1$).

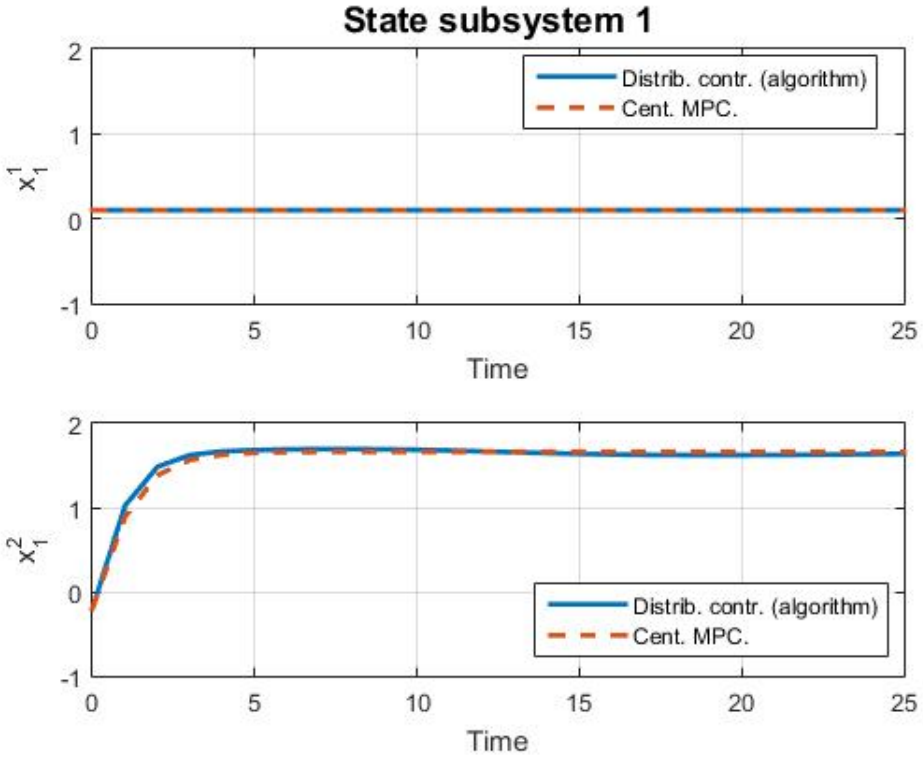

Figure 7.12 States evolution of subsystem 1 ($\lambda_1^f = 1.5\lambda_1$).

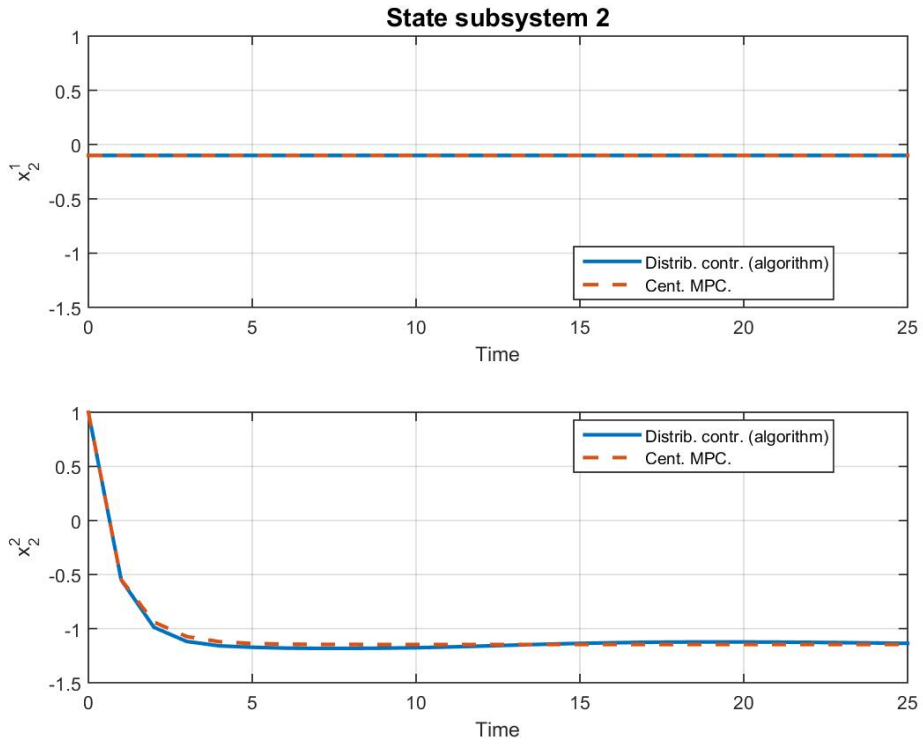

Figure 7.13 States evolution of subsystem 2 ($\lambda_1^f = 1.5\lambda_1$).

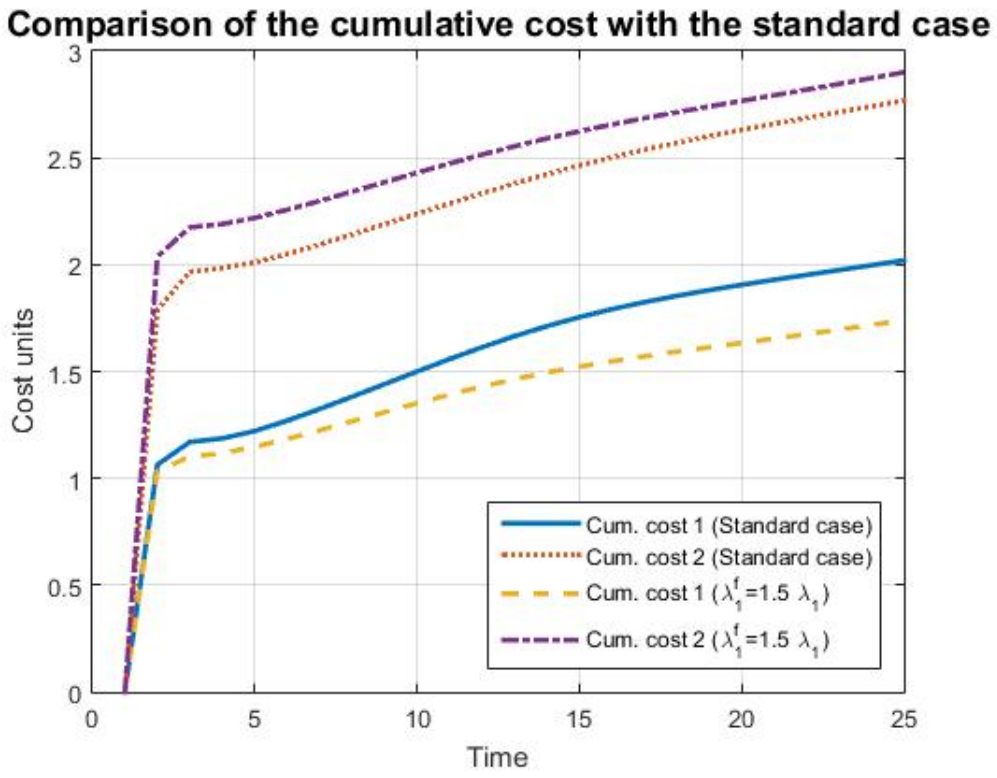

Figure 7.14 Comparison with the standard cas ($\lambda_1^f = 1.5\lambda_1$).

7.3.2 $\lambda_1^f = 1, \lambda_2^f = 0$

At this point, we are going to compare them with a *selfish agent* attack. Figure 7.18 is a proof that the latter causes more harmful effects on the overall performance as it was expected.

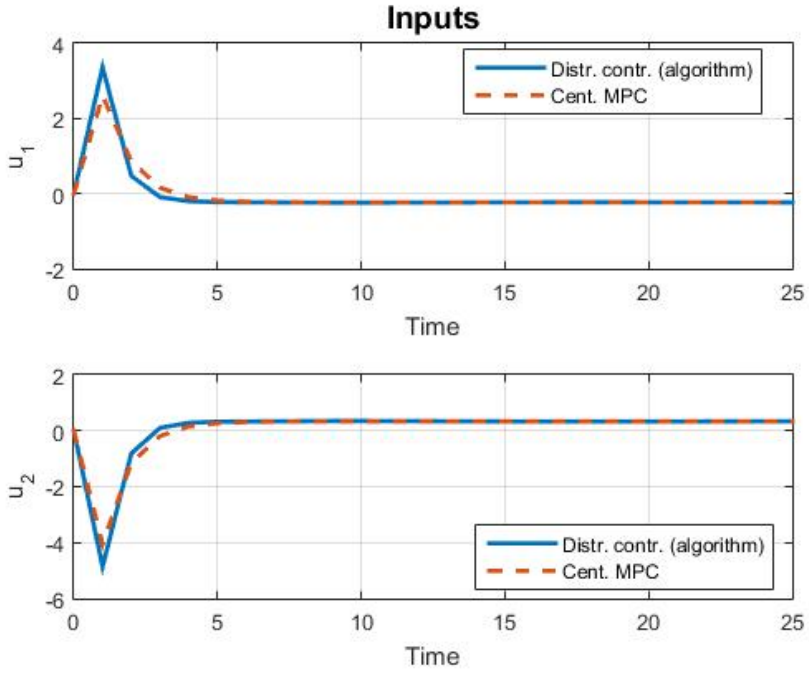

Figure 7.15 Inputs evolution ($\lambda_1^f = 1, \lambda_2^f = 0$).

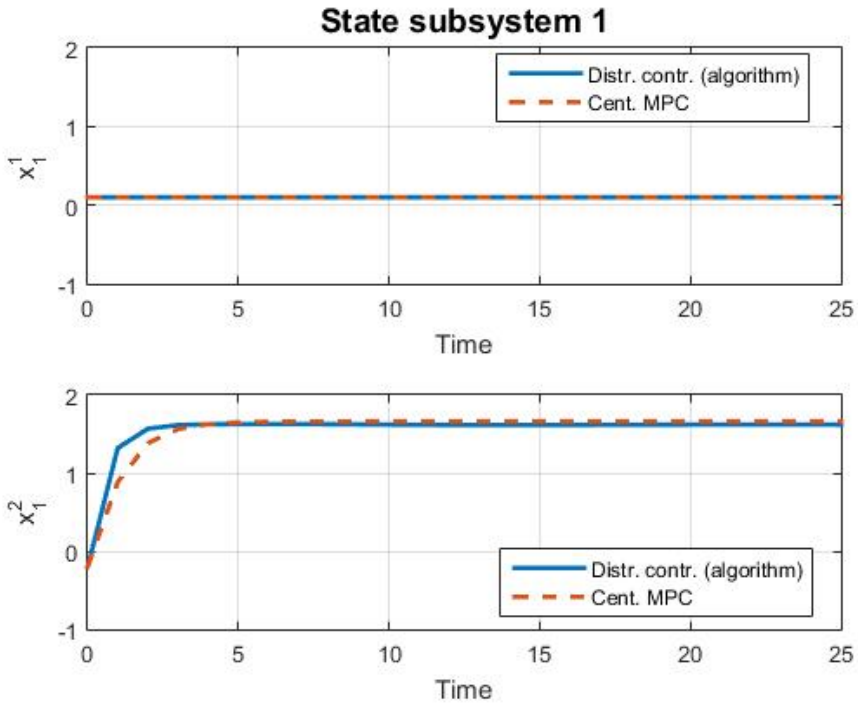

Figure 7.16 States evolution of subsystem 1 ($\lambda_1^f = 1, \lambda_2^f = 0$).

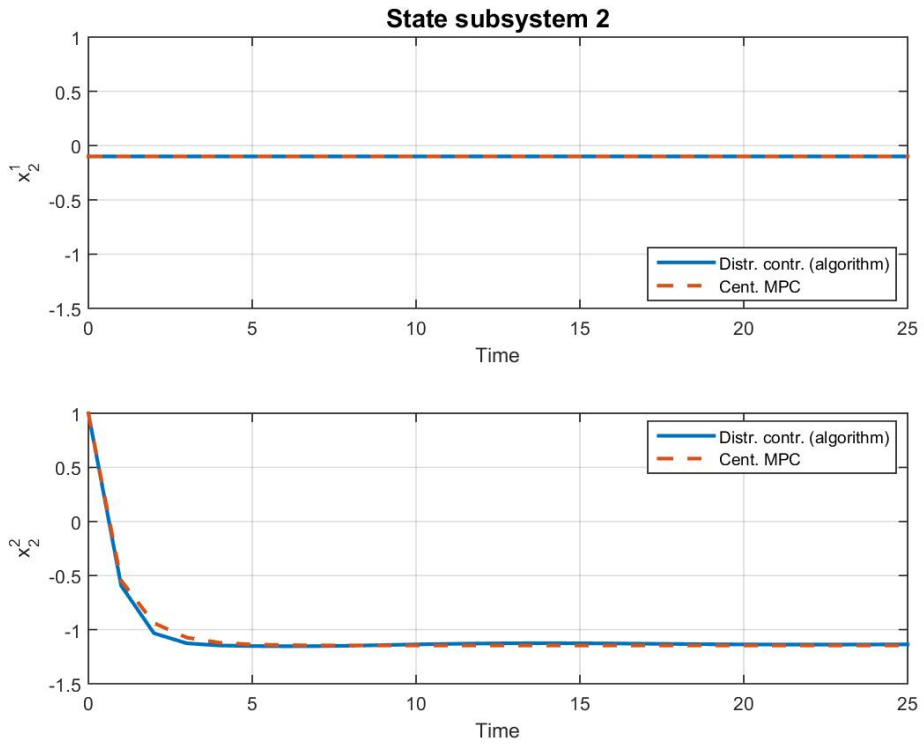

Figure 7.17 States evolution of subsystem 2 ($\lambda_1^f = 1, \lambda_2^f = 0$).

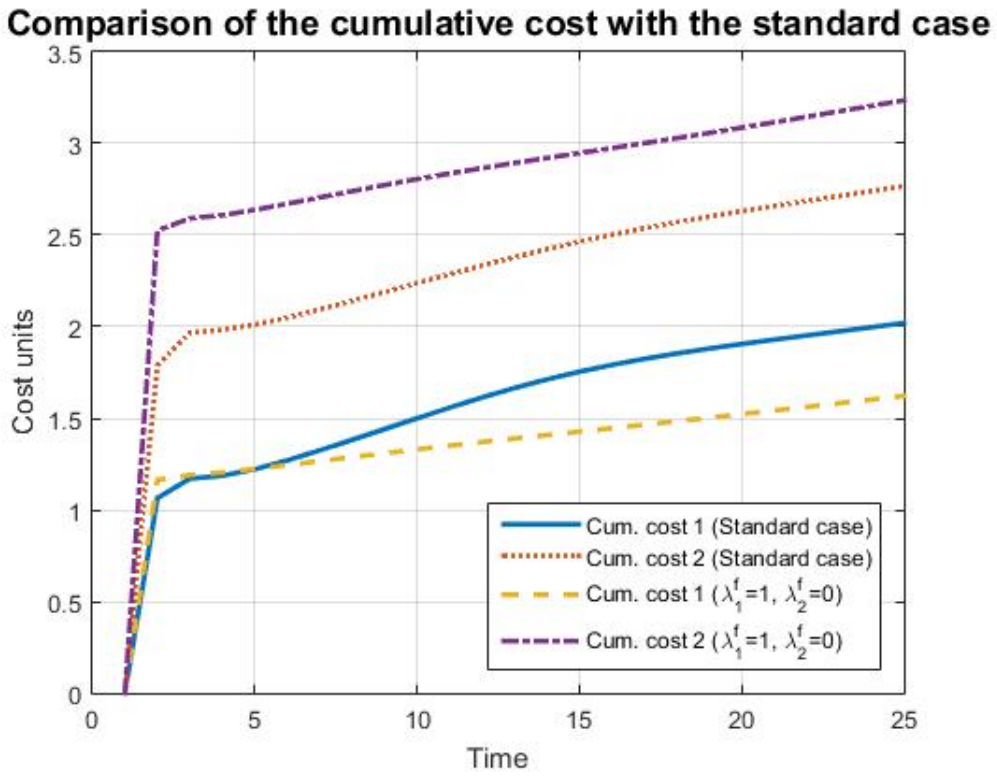

Figure 7.18 Comparison of the cumulative costs (selfish attack and standard case).

7.3.3 $\lambda_1^f = \lambda_1^{f*}$

All this results raises the question of if it could be possible to achieve the similar behaviour reached by a selfish agent maintaining $\lambda_2^f = \lambda_2$ and optimizing the values that λ_1^f takes. That is, we have implemented a similar algorithm to the one to calculate $x_{a,ref}^{f*}$, but in this case the variable will be λ_1^f. The results arrived are presented and compared in Figure 7.19, in which the achievements related to the different fake weights attacks are reflected. It can be obseved that the pursuit of an optimal λ_1^{f*} also leads us to a satisfactory result as the desired reduction is arrived. Consequently, in this example it is not necessary making $\lambda_2^f = 0$ to reach the best local situation seen until now. Finally, those applied values of λ_1^{f*} per instant time are shown in Figure 7.20. It is important to mention that in the corresponding implementation the maximum value which this weight can take has been limited to 20.

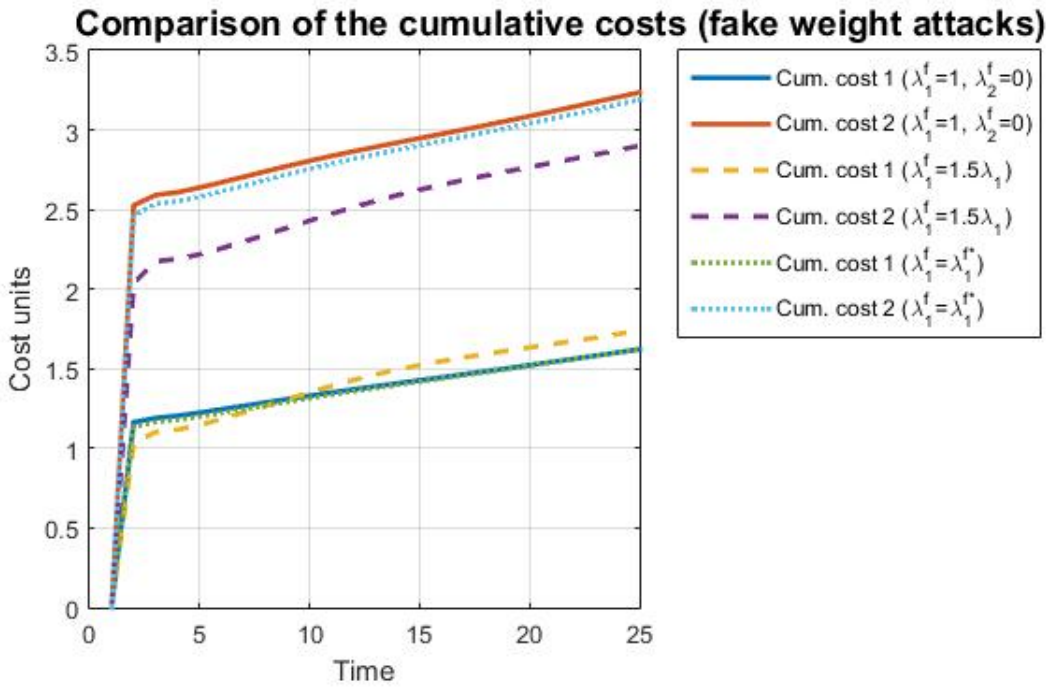

Figure 7.19 Comparison of the cumulative costs (fake weights attacks).

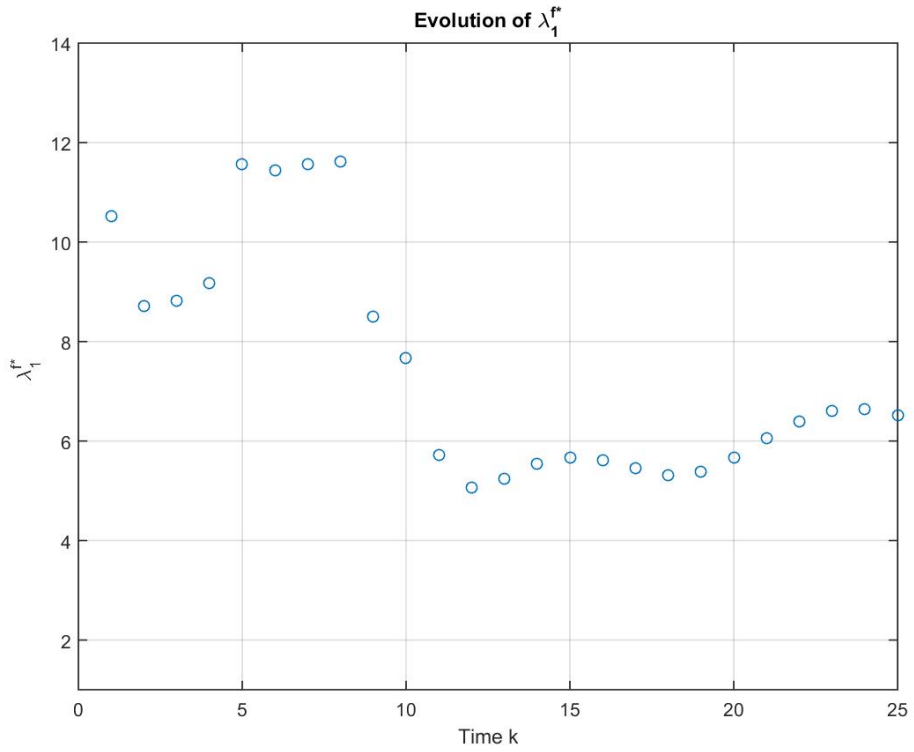

Figure 7.20 Evolution of λ_1^{f*} .

7.4 Evaluation of the *Key Performance Indicators*

In this section we present the values that the Key Performance Indicators takes for the different attacks applied, all of them for a time horizon of 25. The purpose is to associate to the results that the graphics describe different numbers with a particular meaning.

Firstly, the standard case of the algorithm will be compare with the centralized strategy. These first values would allow us to assess how the algorithm works in standard conditions, so that the next step is to evaluate what are the effects of the attacks.

Table 7.3 shows the costs arrived at for both subsystems and for the entire plant when the attacks are carried out. The reduction in the cumulative cost of 1 is notable, which is a reflection of the effectivity of the attack. It has to be underline here that we have arrived to a peculiar situation as we have that the overall cost of the plant is in some cases slightly lower when introducing the false information than when we work under reliable conditions. This is just consequence of the fact that the decline that 1 reaches for itself is greater than the increase of the cost of 2 as result of having taken advantage of the latter. As it can be seen we have that the overall cost is almost the same as in the standard case with another distribution of it in which agent 2 suffers more to help 1 be in a better situation.

Anyway, it should be outlined that if we implement the algorithm without any added inertia the results obtained change and we arrive at those expected values greater than 1 in cases that here are below it. For example, the global cost in the standard case is 4.6039, the one for a selfish attack 4.8911. Therefore, we see the loss of perfomance expected and show with it that the inertia is present and has its effect in the algorithm.

Table 7.1 Performance comparison of the standard case of the algorithm with centralized MPC. Values associated to subsystems 1 and 2 (Example 1).

	Cumulative cost 1	**Cumulative cost 2**
Standard case of the algorithm ($p_{max} = 100, \varepsilon = 0.01$)	2.0381	2.7905
Centralized MPC	1.8165	2.9823

Table 7.2 Performance comparison of the standard case of the algorithm with centralized MPC. Values associated to the whole system (Example 1).

	Cumulative cost in the plant
Standard case of the algorithm ($p_{max} = 100, \varepsilon = 0.01$)	4.8286
Centralized MPC	4.4988

Table 7.3 Cumulative costs reached under the effects of the different attacks (Example 1).

	Cumulative cost 1	**Cumulative cost 2**	**Cumulative cost in the plant**
Optimal false reference	1.6158	3.1227	4.7385
Selfish agent	1.6418	3.2654	4.9072
Fixed fake weight	1.7643	2.9290	4.6933
Optimal fake weight	1.6459	3.2213	4.8672

Table 7.4 Evaluation of the Price of Corruption and improvement achieved by agent 1 (Example 1).

	PoC	$\frac{\text{Cum. Cost 1}}{\text{Cum. Cost 1 (standard)}}$
Optimal false reference	0.9813	0.7928
Selfish agent	1.0163	0.8056
Fixed fake weight	0.9720	0.8657
Optimal fake weight	1.0080	0.8076

8 Example 2: *A Four Tank Plant*

In this example, the case of a system comprised of four interconnected water tanks whose parameters have been taken from [2] is studied. Hereafter, the problem of controlling the plant will be addressed by applying the cooperative DMPC algorithm proposed. It will be done considering the aim of reaching a certain water level in each of the tanks, while taking into account the corresponding couplings and constraints.

An illustrative scheme of the plant can be seen in Figure 8.1. It consists of two top tanks to which the numbers 3 and 4 will be asigned, and two others at the bottom that will be denoted as 1 and 2. The systems works in a manner that the tanks at the top discharge into the ones at the bottom at the same time that all of them are filled with a flow that comes from a storage tank. The latter is done via the actuation of two pumbs denoted as q_A and q_B. Moreover, it is given that this flow can be managed by means of three-way valves, which can also be observed in the scheme representation.

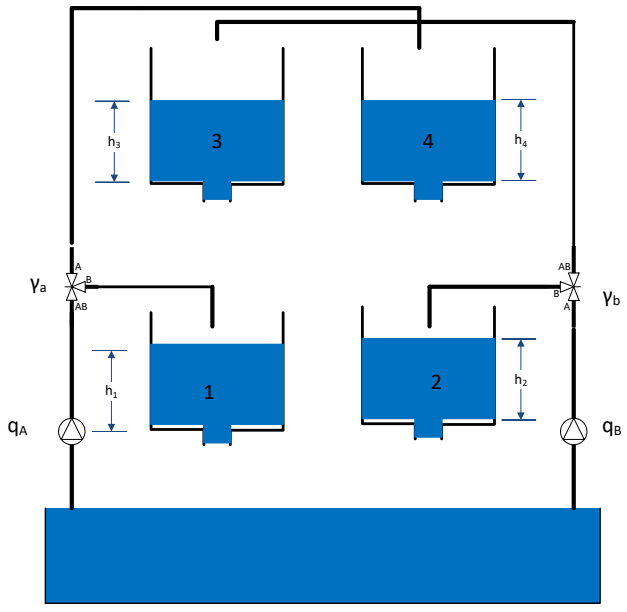

Figure 8.1 Diagram of the plant.

Firsly, the equations which model phisically the evolution of the water levels over continuous time t, are presented (8.1). These levels are related to a certain height that will be indicated from now on by h_i, with

$i = 1,2,3,4$.

$$\begin{aligned}
\frac{dh_1}{dt} &= \frac{-a_1}{S_1}\sqrt{2gh_1} + \frac{a_3}{S_3}\sqrt{2gh_1} + \frac{\gamma_a}{S_1}q_A \\
\frac{dh_2}{dt} &= \frac{-a_2}{S_2}\sqrt{2gh_2} + \frac{a_4}{S_4}\sqrt{2gh_4} + \frac{\gamma_b}{S_2}q_B \\
\frac{dh_3}{dt} &= \frac{-a_3}{S_3}\sqrt{2gh_3} + \frac{1-\gamma_b}{S_3}q_B \\
\frac{dh_4}{dt} &= \frac{-a_4}{S_4}\sqrt{2gh_4} + \frac{1-\gamma_a}{S_4}q_A
\end{aligned} \tag{8.1}$$

where S_i represents the cross section of the duct, a_i a constant which characterises the discharge, and γ_m (m indicates indistinctly A or B) the ratio of the three-ways valves.

Table 8.1 Discharge constants, cross sections of the ducts and ratios of the three-way valves.

a_1	$1.31 \times 10^{-4} m^2$	S_1	$0.06m^2$	γ_a	0.3
a_2	$1.507 \times 10^{-4} m^2$	S_2	$0.06m^2$	γ_b	0.4
a_3	$9.627 \times 10^{-5} m^2$	S_3	$0.06m^2$		
a_4	$8.31 \times 10^{-5} m^2$	S_4	$0.06m^2$		

Following the reference article [2], we will use since now the discrete-time LTI model associated (8.2) and essential information for the problem's resolution provided below.

LTI model:

$$x(k+1) = \begin{bmatrix} 0.9705 & 0 & 0.0205 & 0 \\ 0 & 0.9661 & 0 & 0.0195 \\ 0 & 0 & 0.9792 & 0 \\ 0 & 0 & 0 & 0.9802 \end{bmatrix} x(k) + \begin{bmatrix} 0.0068 & 0.0011 \\ 0.0002 & 0.0091 \\ 0 & 0.0137 \\ 0.0160 & 0 \end{bmatrix} u(k) \tag{8.2}$$

Operating point:

The operating point is determined by q_A^0, q_B^0 and h_i^0, which results in the following definitons of each component of the state and input vector, $x(k)$ and $u(k)$, respectively. In addiction, Table 8.2 specifies the corresponding values.

$$\text{Components of } x(k) : h_i(k) - h_i^0$$
$$\text{Componets of } u(k) : q_m(k) - q_m^0$$

Table 8.2 Values of the operating point.

h_1^0	$0.65m$	q_A^0	$1.63m^3/h$
h_2^0	$0.65m$	q_B^0	$2m^3/h$
h_3^0	$0.65m$		
h_4^0	$0.65m$		

Constraints:

As in Example 1, the problem is subject to certain constraints that will be imposed for an appropiated performance of the plant. In this case, the constraints are specified by different bounded intervals that restrain the possible values that the water levels h_i can reach, as well as the admitted controls of the pumps q_A and q_B. So that, for all time instant k and for every iteration p, the indications in Table 8.3 must be respected.

Table 8.3 Constraints.

$0.2\ m \leq h_1(k), h_3(k) \leq 1.36\ m$	$0\ m^3/h \leq q_A(k) \leq 3.26\ m^3/h$
$0.2\ m \leq h_2(k), h_4(k) \leq 1.36\ m$	$0\ m^3/h \leq q_B(k) \leq 4\ m^3/h$

References:

Finally, the point towards the system should be steered is indicated.

$$h_{1,ref} = 0.5\ m, \qquad h_{2,ref} = 0.6\ m, \qquad h_{3,ref} = 0.7\ m, \qquad h_{4,ref} = 0.8\ m$$

With all of this information, we address the control using the algorithm in the preceding parts with the objective of reaching an appropiate distributed action.

It has been suggested a division of the global system into two subsystems, one which considers tanks 1 and 3, and, therefore, another one composed of tanks 2 and 4. To proceed with the same nomenclature that has been used until now, we will translate the data above to the following matrices. It is important to underline before continuing that the letter *i*, which has differentiated the subsystems in the previous parts, has been used in this example as index for the tanks, which are not directly subsystems, so it should not be confused when the index 1 or 2 are referred to subsystems themselves or to a determined tank. Taking that into account, we have

$$A_1 = \begin{bmatrix} 0.9705 & 0.0205 \\ 0 & 0.09792 \end{bmatrix}, \qquad B_{1A} = \begin{bmatrix} 0.0068 \\ 0 \end{bmatrix}, \qquad B_{1B} = \begin{bmatrix} 0.0011 \\ 0.0137 \end{bmatrix} \qquad n_1 = 2, \qquad m_1 = 2$$

$$A_2 = \begin{bmatrix} 0.9961 & 0.0195 \\ 0 & 0.9802 \end{bmatrix}, \qquad B_{2A} = \begin{bmatrix} 0.0002 \\ 0.0160 \end{bmatrix}, \qquad B_{2B} = \begin{bmatrix} 0.0091 \\ 0 \end{bmatrix} \qquad n_2 = 2, \qquad m_2 = 2$$

$$A_{cen} = \begin{bmatrix} 0.9705 & 0.0205 & & \\ 0 & 0.09792 & & \\ & & 0.9961 & 0.0195 \\ & & 0 & 0.9802 \end{bmatrix}, \qquad B_{cen} = \begin{bmatrix} 0.0068 & 0.0011 \\ 0 & 0.0137 \\ 0.0002 & 0.0091 \\ 0.0160 & 0 \end{bmatrix}, \qquad n_x = 4, \qquad m_u = 2$$

Hence, at k the states at this current time are given by

$$x_1(k|k) = \begin{bmatrix} h_1(k|k) - h_1^0 \\ h_3(k|k) - h_3^0 \end{bmatrix}, \qquad x_2(k|k) = \begin{bmatrix} h_2(k|k) - h_2^0 \\ h_4(k|k) - h_4^0 \end{bmatrix}, \qquad x(k|k) = \begin{bmatrix} h_1(k|k) - h_1^0 \\ h_3(k|k) - h_3^0 \\ h_2(k|k) - h_2^0 \\ h_4(k|k) - h_4^0 \end{bmatrix}$$

Let $x_j^i(\cdot)$ denote the component of subsystem j ($j = 1,2$) referred to tank i ($i = 1,2,3,4$). The constraints for all $n = 0,1,...,N$ and for all k can be expressed as

$$\begin{bmatrix} -0.45 \\ -0.45 \end{bmatrix} \leq \begin{bmatrix} x_1^1(k+n|k) \\ x_1^3(k+n|k) \end{bmatrix} \leq \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix}, \qquad \begin{bmatrix} -0.45 \\ -0.45 \end{bmatrix} \leq \begin{bmatrix} x_2^2(k+n|k) \\ x_2^4(k+n|k) \end{bmatrix} \leq \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix}$$

and, moreover, the references are

$$\begin{bmatrix} x_{1_{ref}}^1 \\ x_{1_{ref}}^3 \end{bmatrix} = \begin{bmatrix} -0.15 \\ 0.05 \end{bmatrix}, \qquad \begin{bmatrix} x_{2_{ref}}^2 \\ x_{2_{ref}}^4 \end{bmatrix} = \begin{bmatrix} -0.05 \\ 0.15 \end{bmatrix}$$

To finish the parametrization of the problem, the weighting matrices that will define the cost functions are chosen as

$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad Q = \begin{bmatrix} 1 & 0 & & \\ 0 & 1 & & \\ & & 1 & 0 \\ & & 0 & 1 \end{bmatrix}$$

$$R_1 = 0.01, \qquad R_2 = 0.01 \qquad R = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

The length of the simulation has been set to 100 time units, so that $k \in \{0,100\}$, and the control horizon has been defined such as $N = 5$. Further parameters that must be specified are the maximum number of iterations per sample time $p_{max} = 50$ or the value assigned to ε, which has been 0.05.

Then, we will make use of the same change of variable, that is $\mathbf{u} = \mathbf{M}_1\mathbf{u}_1 + \mathbf{M}_2\mathbf{u}_2$, which leads us to the defined optimizations problems and constraints interval in Example 1 but with the application of the new plant's data.

8.1 Standard case with Example 2

The results obtained solving the cooperation-based optimizations of this distributed MPC framework are presented in Figures 8.2 to 8.4. The first one shows the actions performed on the pumps along time, thus, they are the most direct reflection of the agreements reached each k. It can be noticed that the curves associated to the distributed algorithm almost overlaps the ones that would be arrived with a centralized control. Furthermore, Figure 8.3 shows the corresponding heights reached by the stored water in each tank when working under reliable conditions. In Figure 8.4 the cumulative costs are represented, which in principle evidences a better local situation of subsystem 2. Henceforth, these results will be used as reference for assessing how the system responds to the described attacks.

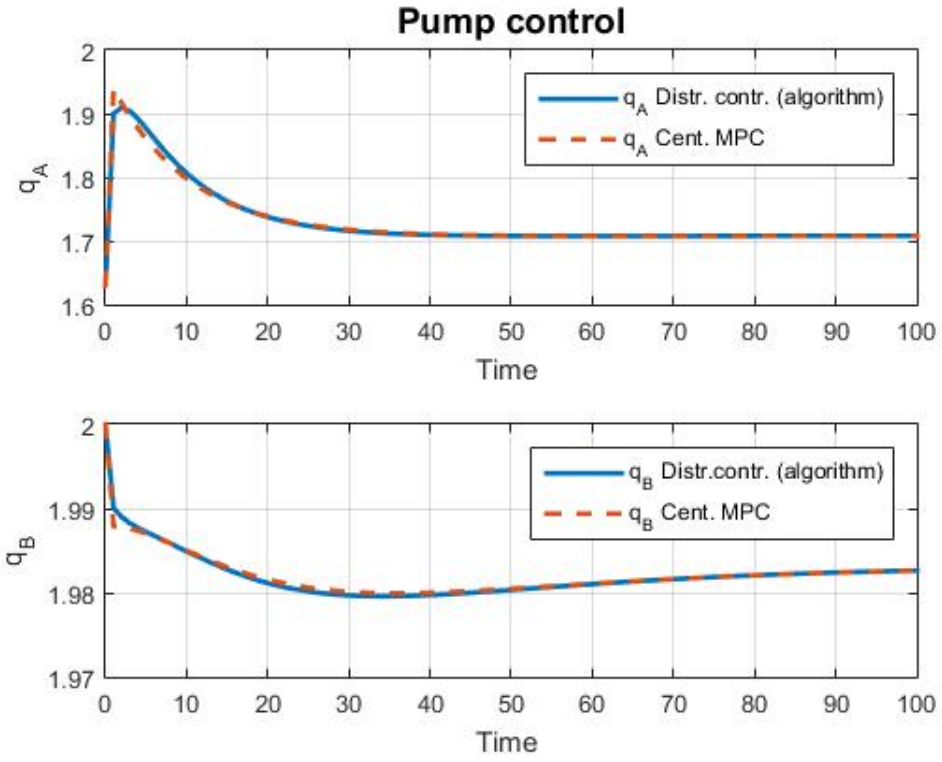

Figure 8.2 Pump control evolution (standard case).

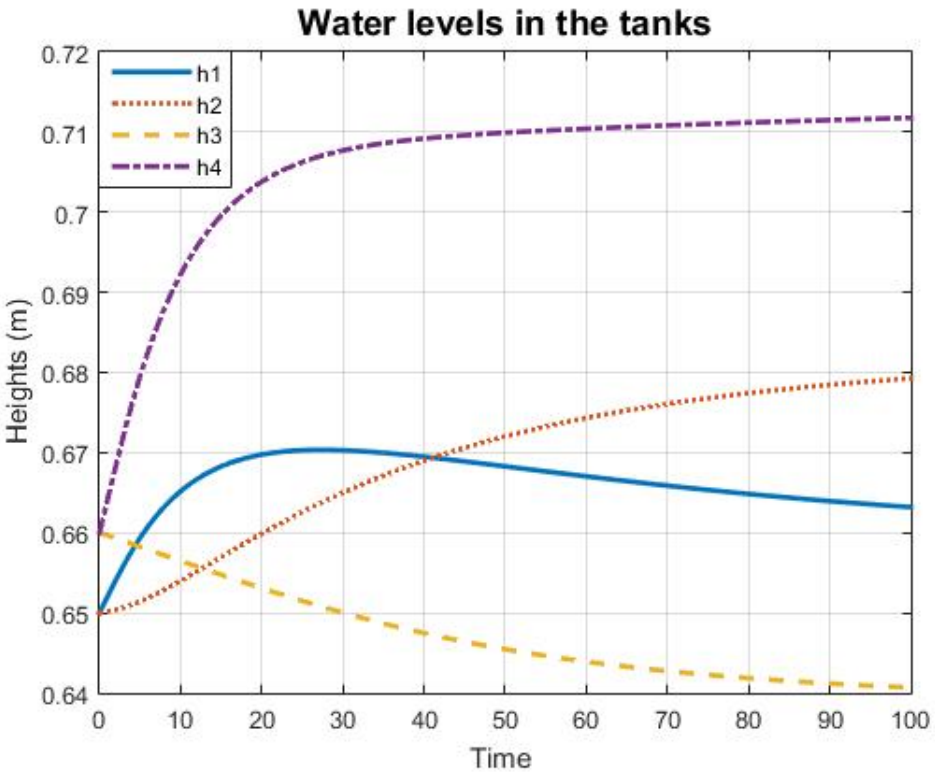

Figure 8.3 Evolution of the tanks' water levels (standard case).

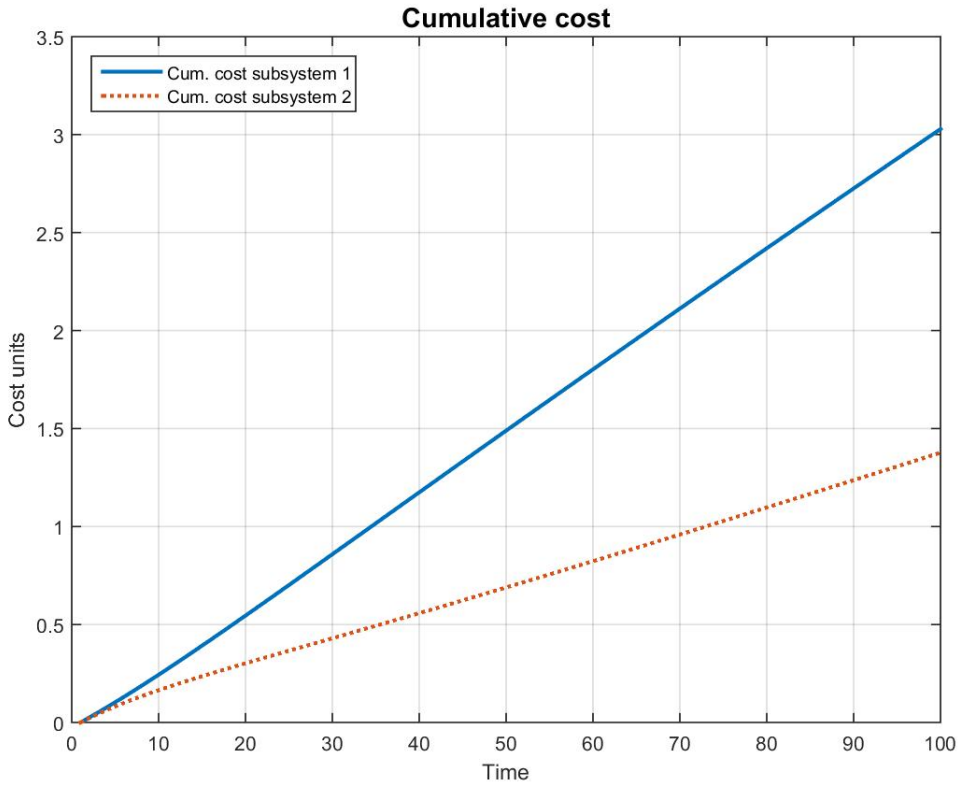

Figure 8.4 Cumulative cost for subsystems 1 and 2 (standard case).

8.2 False reference attacks to Example 2

The implementation of this first attack has been addressed in two different ways. Firstly, it has been cosidered the case in which a constant false reference is introduced from the beginning in the optimization problem solved by the attacker. As it has been described, a convenient choice of the value that this false reference takes can lead to a reduction of the attacker's local cost and thus get its objective. Secondly, it has been presented what happens when the proposed method of determining the optimal false reference is applied. Finally, both results are compared in order to see wheter the algorithm performs as intented or not.

8.2.1 Constant false reference

Figures 8.5 to 8.7 show the results when agent 1 ($a = 1$) considers $x^f_{1,ref} = [-0.21; 0.01]$ to bias the negotiation through the trajectories $\mathbf{u}^p_1(k)$ calculated per iteration and time instant. This attack causes a clear impact in the water level's arrived in the tanks and their evolution. It is also remarkable the worsening for 4 and the notable improvement for 1 regarding the real reference ($h_{1,ref} = 0.5$).

Figure 8.7 deserves special focus as it shows the difference in the cumulative costs reached and they are the motivation for the attack. It has been directly represented together with the standard case to prove that this choice of $x^f_{1,ref}$ leads to a reduction of the cost of 1 at the expense of 2. In principle, there is no guarantee that this decline is the optimal that can be induced and, besides that, another arbitrary choice of the false reference could have resulted in not meeting the target.

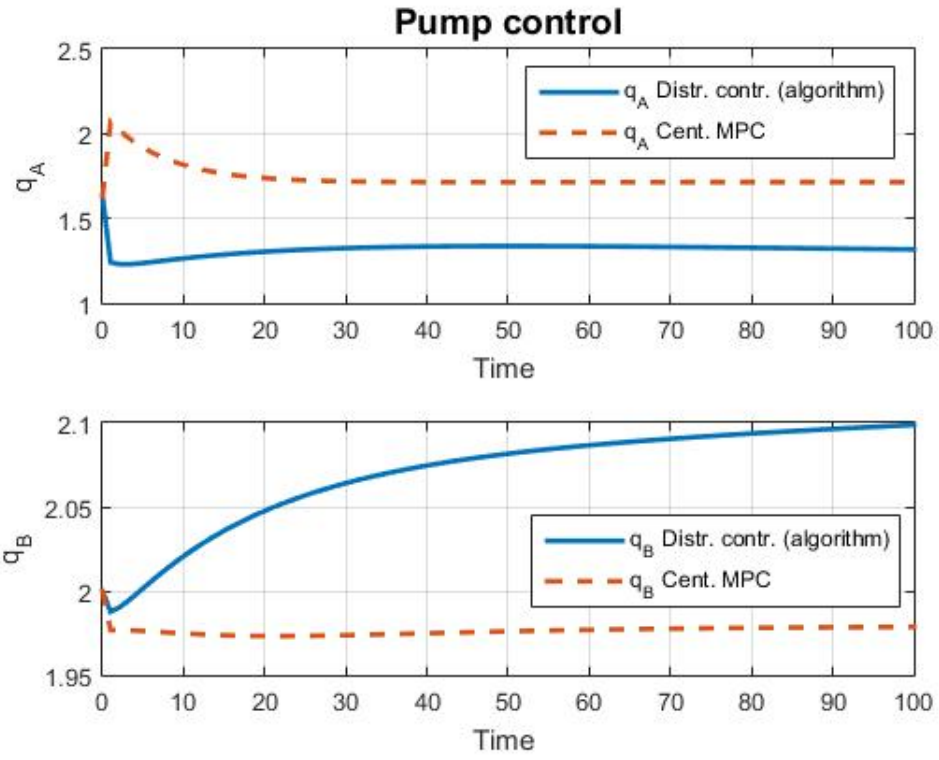

Figure 8.5 Pump control evolution ($x^f_{1,ref} = [-0.21; 0.01]$).

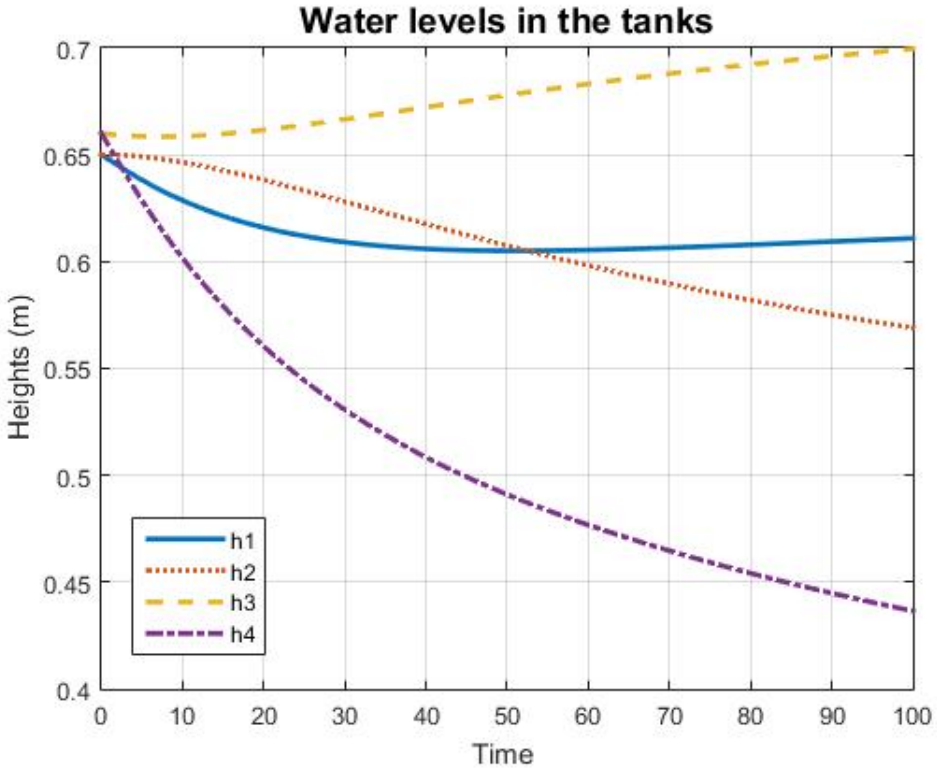

Figure 8.6 Evolution of the tanks' heights ($x^f_{1,ref} = [-0.21; 0.01]$).

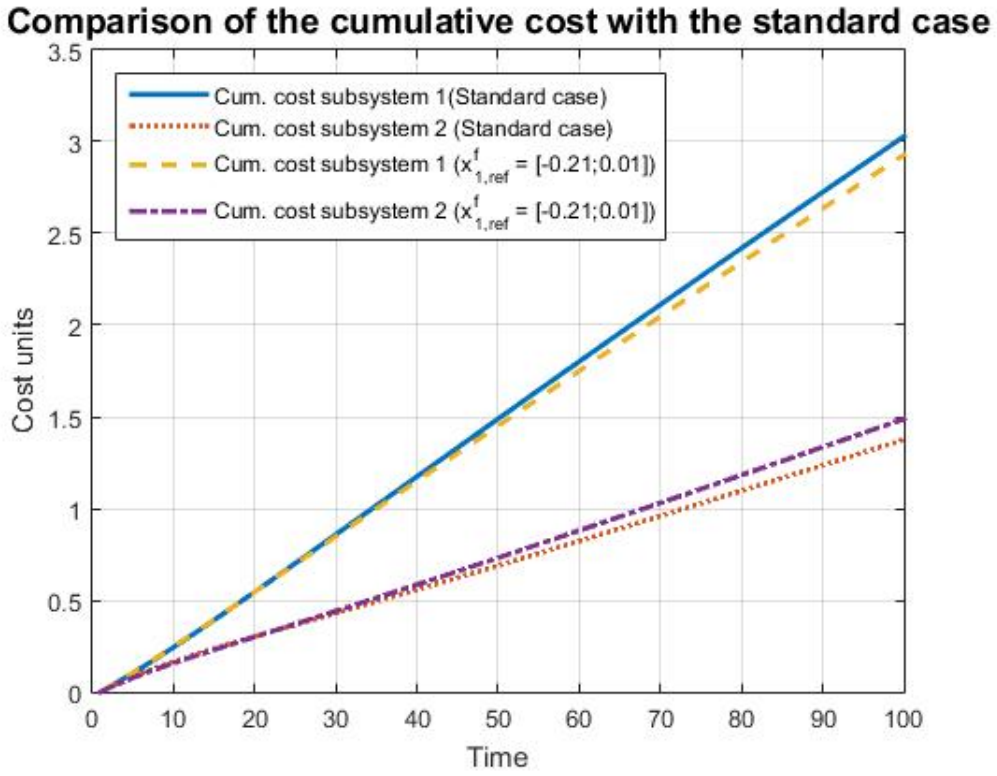

Figure 8.7 Comparison with the standard case ($x^f_{1,ref} = [-0.21; 0.01]$).

8.2.2 Optimal false reference

Here, the attacker 1 determines the optimal $x^f_{1,ref}$ to be used in its optimization problem. Figure 8.8 shows the pump control's evolution in which can be seen the wide range of deviation concerning the centralized solution especially in the case of q_A. The consequence of their implementation is represented in Figure 8.9, in which the curves associated to subsystem 2 are adversely affected while h_1 and h_3 present the best evolution towards the real reference achieved until now.

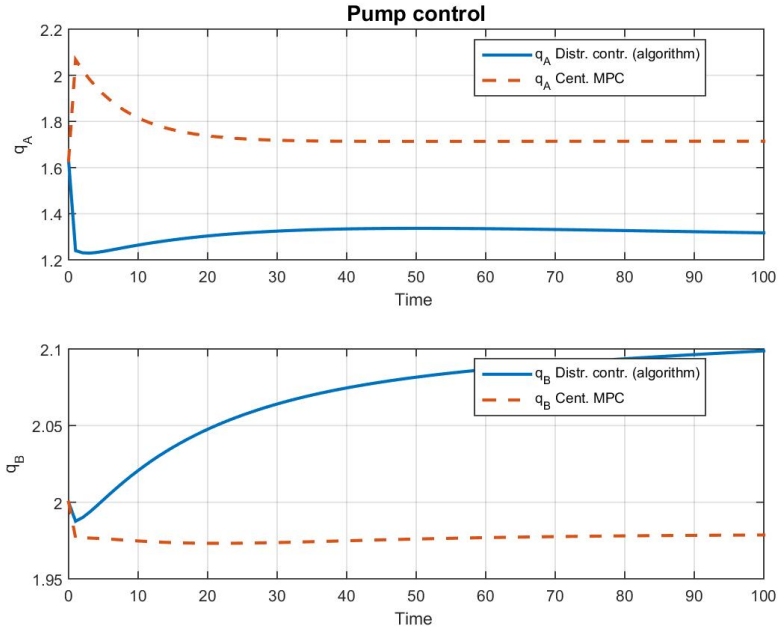

Figure 8.8 Pump control evolution ($x^f_{1,ref} = x^{f*}_{1,ref}$).

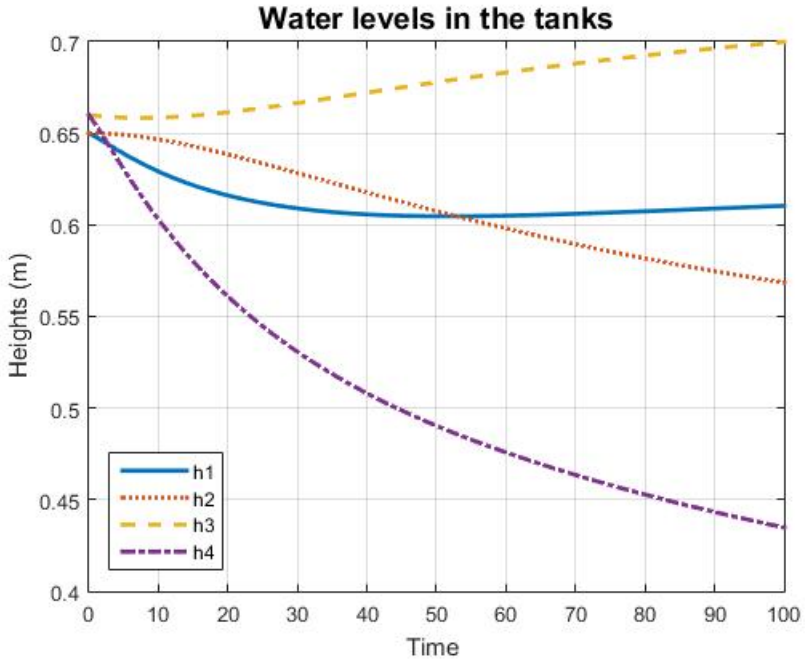

Figure 8.9 Evolution of the tanks' heights ($x^f_{1,ref} = x^{f*}_{1,ref}$).

Figure 8.10 compares the outcome with the standard case while Figure 8.11 contrasts the costs for both examples of false references attacks. This figure indends to illustrate the effectiveness and improvement related with the pursuit of the optimal false reference and, therefore, to show the good behaviour of the method proposed theoretically.

Lastly, Figure 7.10 has been added to show the values that the components of the false reference take along time. It is important to mention that they are the ones directly used in the computing of the inputs applied, that is, the false reference arrived each k after the iterative negotiation.

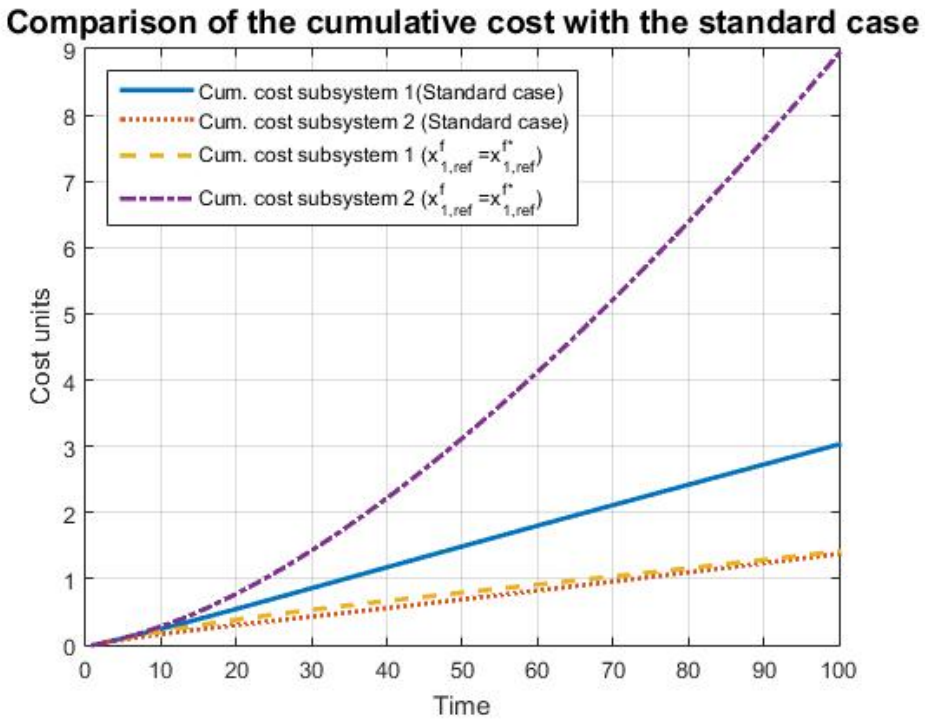

Figure 8.10 Comparison with the standard case ($x^f_{1,ref} = x^{f*}_{1,ref}$) .

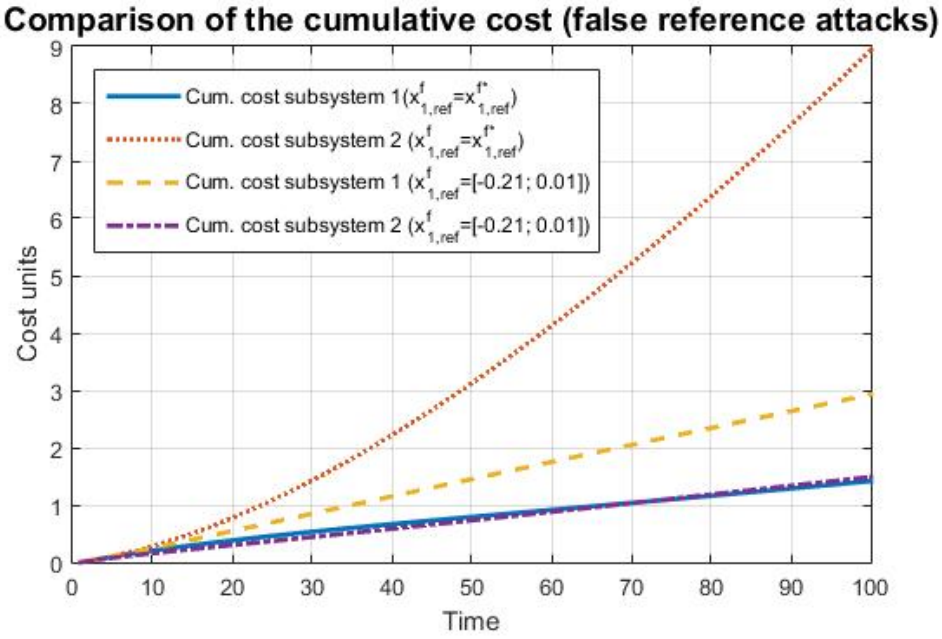

Figure 8.11 Comparison of both false references attacks' cumulative costs .

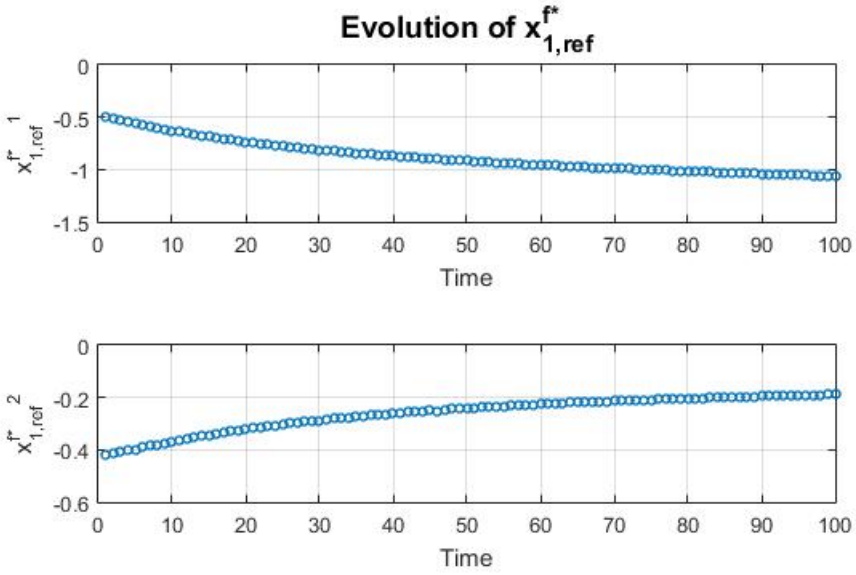

Figure 8.12 Evolution of $x^{f*}_{1,ref}$.

8.3 Selfish agent attack to Example 2

This attack does not only implies the introduction of an extreme attacker in which a controller just thinks about itself, but also confirms the mentioned effectiveness of the preceeding optimal false reference case. The reason for that is that the results arrived are almost the same, so that what has been done before is forgetting about the welfare of 2 and try to take advantage of it to improve 1. The implementation differs but the outcomes coincide. Figures 8.13 and 8.14 correspond to the inputs and water level's evolution respectively, where we can see again how the attack makes the negotiation's agreements to be in favour of 1. Finally, the cumulative costs have been compared to the standard case in Figure 8.15.

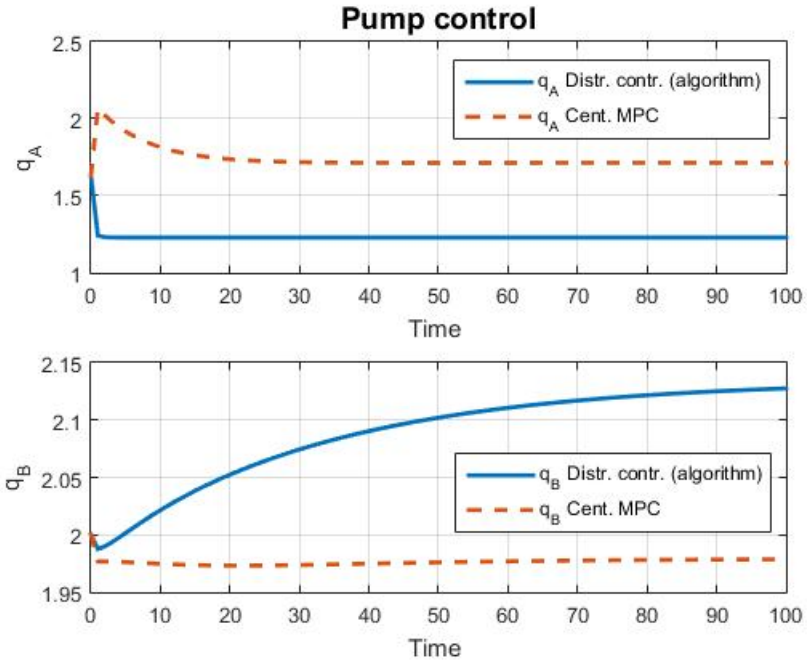

Figure 8.13 Pump control evolution (selfish agent).

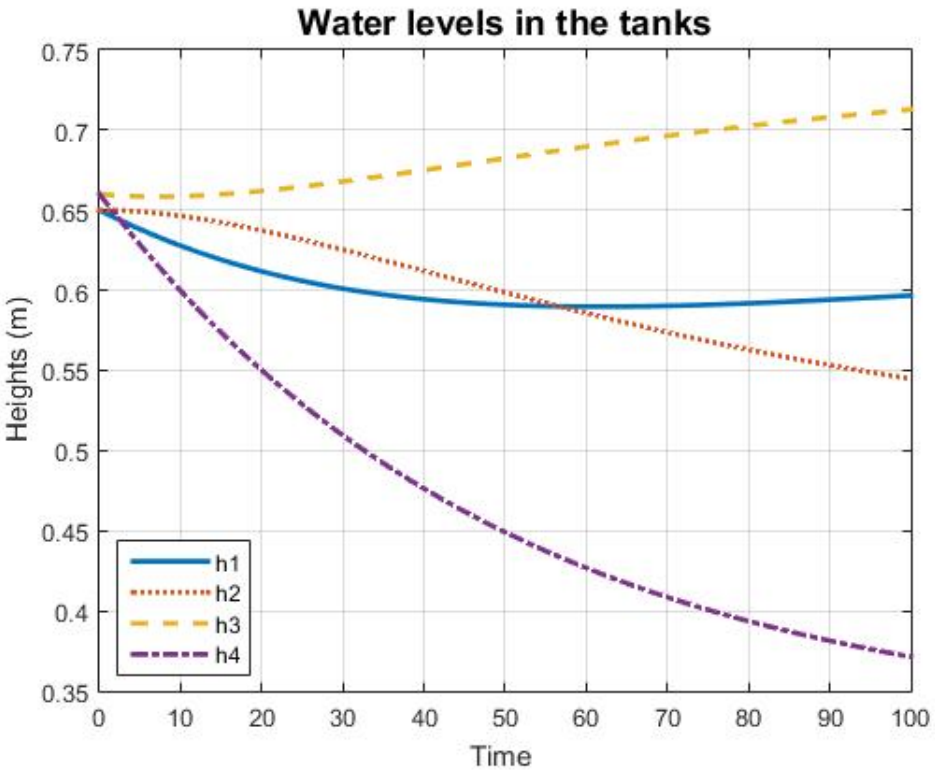

Figure 8.14 Evolution of the tanks' heights (selfish agent).

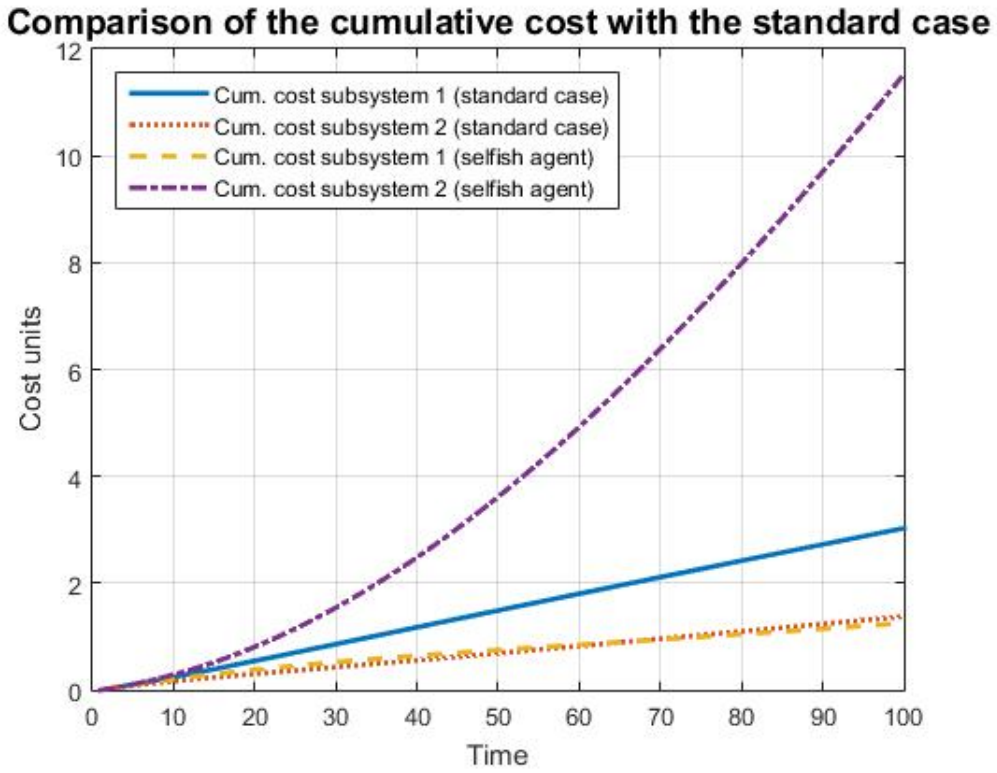

Figure 8.15 Comparison with the standard case (selfish agent).

8.4 Fake constraints attack to Example 2

This last attack is a proof of that benefits can also be gotten by changes in the constraints as described theoretically. In this case it has been used $\mathscr{U}_1^* = 0.05 \times \mathscr{U}_1$. As before, the inputs, states and cumulative cost are presented (Figures 8.16 to 8.18). First time steps of the corresponding graph to u_1 reflects the application of the new limitation . In this case the latter does not strongly affect the water levels reached at time $k = 100$ with respect to the standard case, however it has leads to a slight improvement of the attacker's local situation.

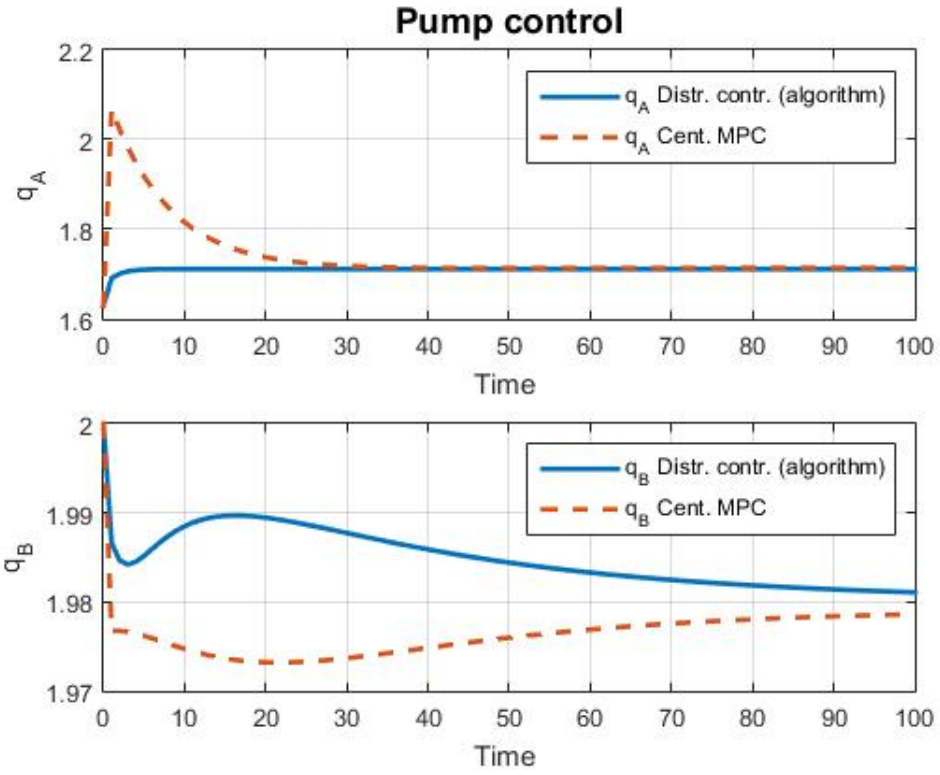

Figure 8.16 Pump control evolution (fake constraints).

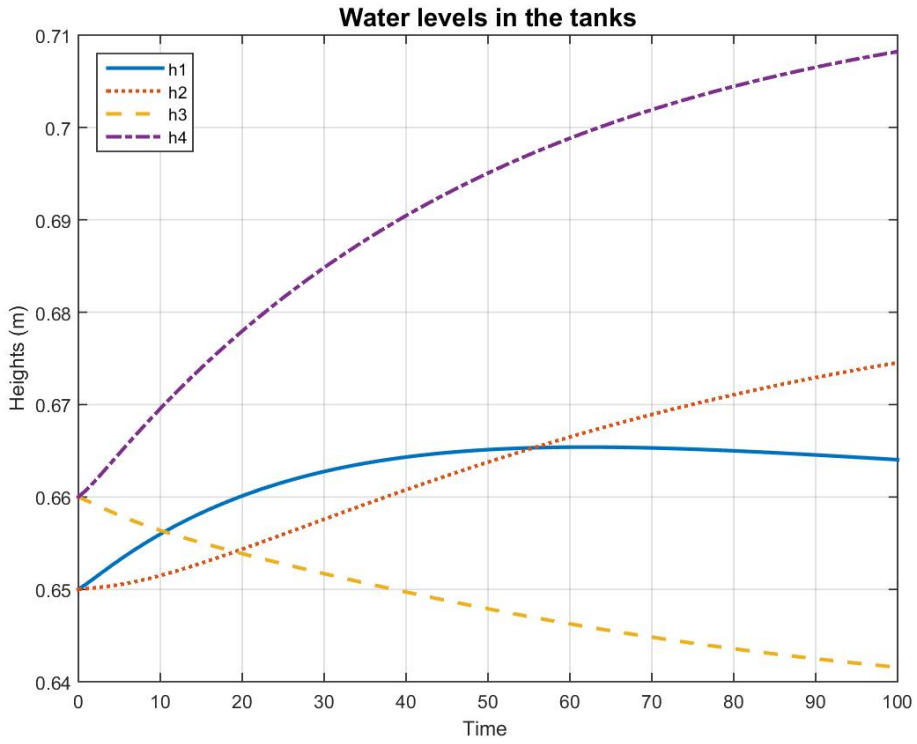

Figure 8.17 Evolution of the tanks' heights (fake constraints).

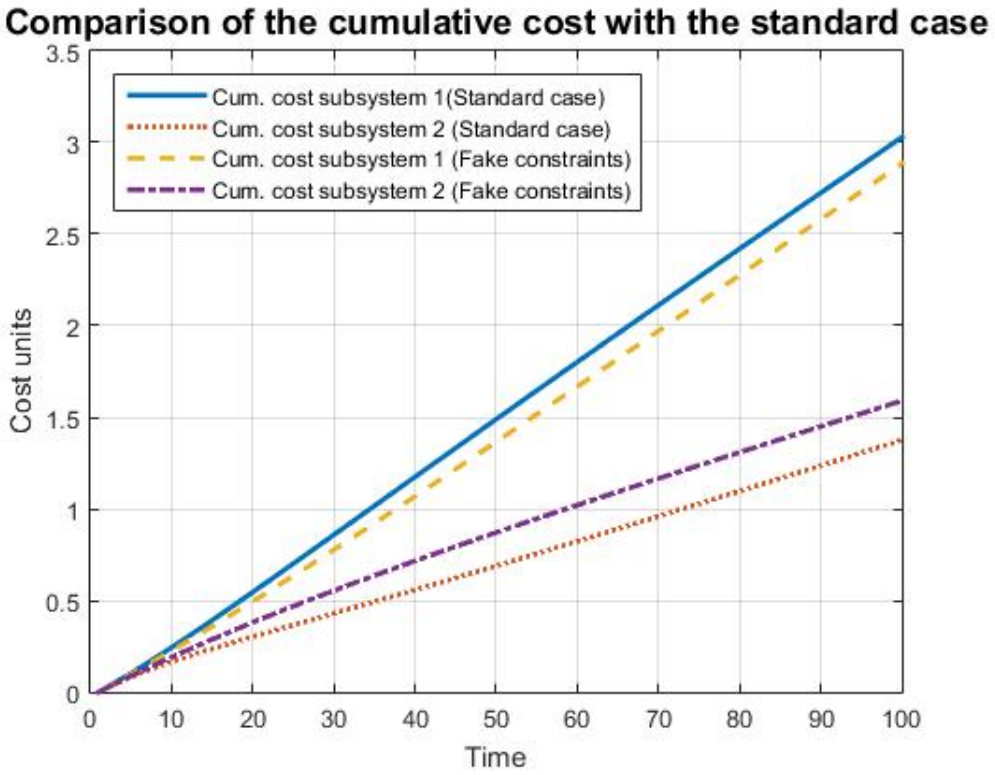

Figure 8.18 Comparison with the standard case (fake constraints).

8.5 Evaluation of the *Price of Corruption*

The following tables show the results presented above translated to a serie of numbers which assess the damage caused (all of them for a time horizon of 100). It is proved that the higher PoC, and therefore, the harder loss of performance, is associated with the selfish attack which almost leads to trebling the cost for the plant concerning the reliable development of the algorithm. Table 8.8 has been added to represent the decrease in the cost of the attacker reached in all cases, being the latter greater for those values higher of the PoC.

Table 8.4 Performance comparison of the standard case of the algorithm with Centralized MPC. Values associated to subsystems 1 and 2 (Example 2).

	Cumulative cost 1	**Cumulative cost 2**
Standard case of the algorithm ($p_{max} = 50, \varepsilon = 0.05$)	3.1115	1.3025
Centralized MPC	3.0317	1.3721

Table 8.5 Performance comparison of the standard case of the algorithm with Centralized MPC. Values associated to the whole system (Example 2).

	Cumulative cost in the plant
Standard case of the algorithm ($p_{max} = 50, \varepsilon = 0.05$)	4.4140
Centralized MPC	4.4038

Table 8.6 Cumulative costs reached under the effects of the different attacks (Example 2).

	Cumulative cost 1	**Cumulative cost 2**	**Cumulative cost in the plant**
Fixed false reference	2.9570	1.5037	4.4607
Fake constraints	2.9149	1.6042	4.5191
Optimal false reference	1.4323	9.0300	10.4623
Selfish agent	1.2659	11.7109	12.9768

Table 8.7 Evaluation of the Price of Corruption (Example 2).

	PoC
Fixed false reference	1.0106
Fake constraints	1.0238
Optimal false reference	2.3703
Selfish agent	2.9399

Table 8.8 Improvement achieved by agent 1 (Example 2).

	$\frac{\text{Cum. Cost 1}}{\text{Cum. Cost 1 (standard)}}$
Fixed false reference	0.9493
Fake constraints	0.9368
Optimal false reference	0.4603
Selfish agent	0.4068

8.6 Min-Max approach with Example 2

In this last section, we show the behaviour when the described min-max method is implemented. Again, it proves the conservadurism of the latter as just a small improvement is achieved with the parameters of the problem. However, the change introduced can be more notable for different parameters. Figure 8.20 shows the case in which just the second component of $h_{1,ref}$ has been modified such that $h_{1,ref} = [0.5 \;\; 0.8]^T$. For this case, the reduction of the cost associated to 2 with respect to the selfish attack without defense is more significant as well as the increase in the cost of the attacker.

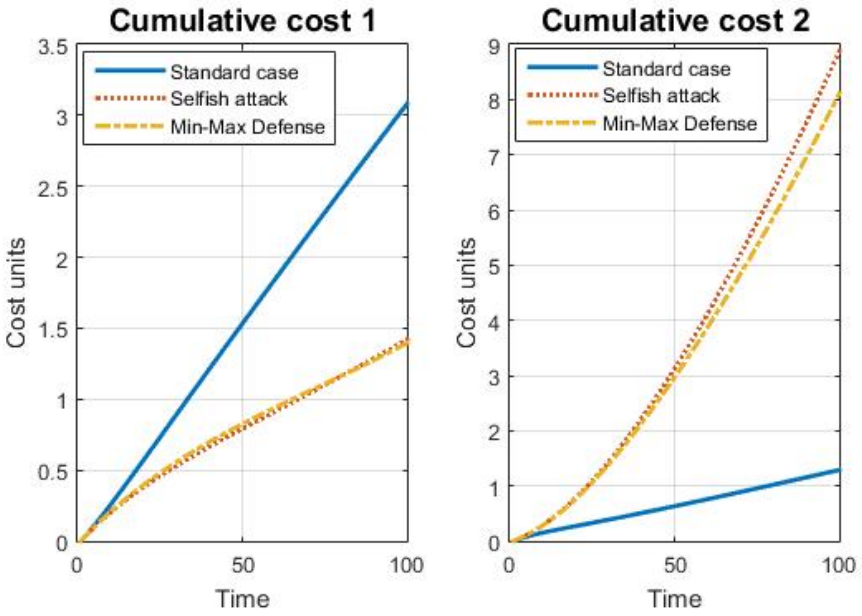

Figure 8.19 Results when introducing the min-max defense .

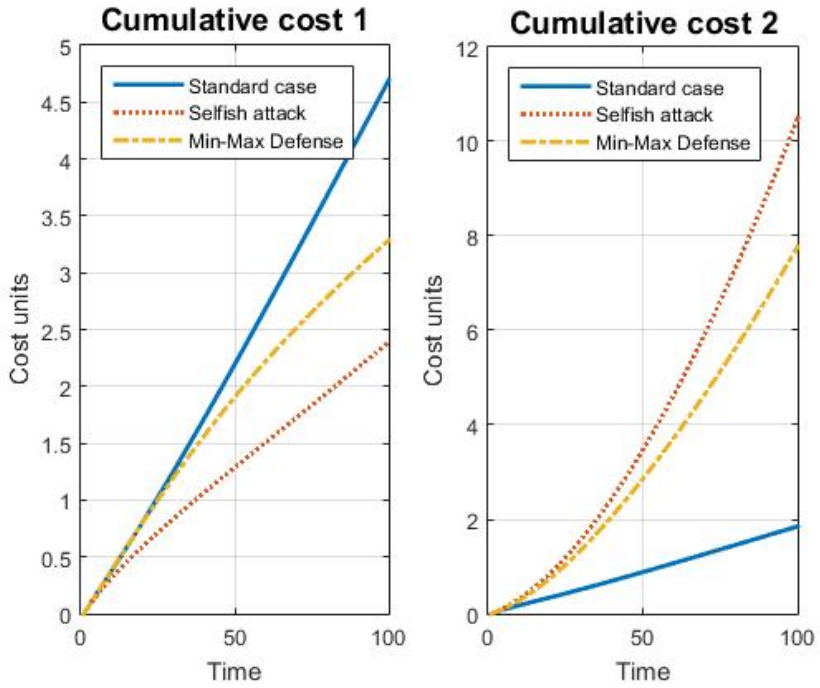

Figure 8.20 Results when introducing the min-max defense (change of $h_{1,ref}$ to $h_{1,ref} = [0.5 \;\; 0.8]^T$).

List of Figures

List of Tables

Bibliography

[1] A. Venkat, J. Rawlings, and S. Wright, "Stability and optimality of distributed model predictive control," in *Proceedings of the 44th Conference on Decision and Control, and the European Control Conference 2005*, Sevilla, December 2005.

[2] P. Velarde, J. Maestre, H.Ishii, and R. Negenborn, "Scenario based defense mechanism for distributed model predictive control," 2017.

[3] A. Venkat, "Distributed model predictive control: Theory and applications," Ph.D. dissertation, University of Wisconsin-Madison, 2006.

[4] P. Scokaert and D. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 8, pp. 1136–1142, 1998.

[5] J. Maestre, D. M. de la Peña, E. Camacho, and T. Álamo, "Distributed model predictive control based on agent negotiation," *Journal of Process Control*, vol. 21, pp. 685–697, 2011.

[6] W.Langson, I.Chryssochoos, S. Raković, and D. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, pp. 125–133, 2004.

[7] L. Fagiano and A. Teel, "Generalized terminal state constraint for model predictive control," *Automatica 49*, pp. 2622–2631, 2013.

[8] J. Maestre, D. M. de la Peña, and E. F. Camacho, "Distributed model predictive control based on a cooperative game," *Optimal Control Applications and Methods*, vol. 32, pp. 153–176, 2011.

[9] J. Rawlings, D. Bonné, J. B. Jorgensen, A. Venkat, and S. B. Jorgensen, "Unreachable setpoints in model predictive control," *IEEE Transaction on Automatic Control*, vol. 53, no. 9, pp. 2209–2215, 2008.

[10] J.B.Rawlings, A. Venkat, and S. Wright, "Distributed model predictive control of large-scaled systems," 2005.

[11] P. Trodden and J. Maestre, "Distributed predictive control with minimization of mutual disturbances," *Automatica*, vol. 77, pp. 31–43, 2017.

[12] B. Stewart, A. Venkat, J. B. Rawlings, S.J.Wright, and G.Pannocchia, *Cooperative distributed model predictive control*. Elsevier, 2010, vol. 59, no. 8.

[13] J. Maestre and R. Negenborn, *Distributed Model Predictive Control Made Easy*. Springer, 2014.

[14] D. Rodríguez and C. Bordóns, "Apuntes de ingeniería de control," ETSI. Universidad de Sevilla, 2007.

[15] D. R. Ramírez, "Perspectiva general del control predictivo min-max," ETSI. Universidad de Sevilla.

[16] R. M. Freund, "Quadratic functions, optimization, and quadratic forms," Massachussetts Institute of Technology, 2004.

[17] T. Roughgarden, "Selfish routing and the price of anarchy," Cornell University.

[18] S. Robinson, “The price of anarchy,” *SIAM News*, vol. 37, no. 5, 2004.

[19] M. Grasmair, “Basic properties of convex functions,” Norwegian University of Science and Technology.