Self-triggered MPC with Performance Guarantee for Tracking Piecewise Constant Reference Signals ⋆

Liang Lu [a], Daniel Limon [b], Ilya Kolmanovsky [c]

[a] *Ningbo Research Institute, Zhejiang University, Ningbo, China*

[b] *Departamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, Spain*

[c] *Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, United States*

Abstract

This paper considers a self-triggered MPC controller design strategy for tracking piecewise constant reference signals. The proposed triggering scheme is based on the relaxed dynamic programming inequality and the idea of reference governor; such a scheme computes both the updated control action and the next triggering time. The resulting self-triggered tracking MPC control law preserves stability and constraint satisfaction and also satisfies certain a priori chosen performance requirements without the need to impose stabilizing terminal conditions. An illustrative example shows the effectiveness of this self-triggered tracking MPC implementation.

Key words: Self-triggered Control, Tracking Model Predictive Control, Reference Governor, Relaxed Dynamic Programming.

1 Introduction

In conventional Model Predictive Control (MPC) [20,21] an open-loop optimal control problem is solved at each sampling time resulting in a sequence of control inputs of which only the first one is implemented. This leads to an algorithmically defined feedback control law.

The classical MPC can be viewed as "time-driven" because the control input profile is updated repeatedly after each fixed time interval. Specifically, at each sampling time a sequence of control values is computed, but only the first element is applied to the system while the rest is discarded. One may ask whether we can continue using the sequence we have computed for as long as possible while still guaranteeing stability and required performance. The answer leads to the so-called "event-driven" MPC in which the control computations only happen when some prescribed "event" occurs. This reduces the frequency of MPC updates and average computing power required. For instance, in [3, 4], the MPC computation is activated by comparing the measured state and its past prediction. However, this requires continuously taking measurements and monitoring the system.

In self-triggered MPC, the necessary measurement and computation only take place at a triggered time at which both the updated MPC control actions and the next triggering time are determined. In particular, in [8] a self-triggered linear quadratic control (LQR) strategy is developed for linear systems without constraints. The paper [13] considers a self-triggered receding horizon controller for multiple-loop unconstrained linear time-invariant (LTI) systems and proposes a co-design of control and sensor sampling strategy. For self-triggered MPC of constrained systems, in [9, 14, 18, 26], the control law and triggering conditions are co-designed to satisfy a specified closed-loop performance requirement. In particular, in [9], a self-triggered scheme is derived to get "group sparse" control signals by holding the control value at the triggering time to be constant for as long as possible. A similar idea is also pursued in [14], where control signals are kept the same between the triggering time instants. However, keeping the control signal con-

⋆ This paper was not presented at any IFAC meeting. This work was supported in part by MINERCO-Spain and FEDER funds under the grants PID2019-106212RB-C41.
Corresponding author Liang Lu. Tel. +86-0574-88229195.

Email addresses: liangup@gmail.com (Liang Lu), dlm@us.es (Daniel Limon), ilya@umich.edu (Ilya Kolmanovsky).

stant may cause the control and state mismatch as the state is evolving with time. Thus, consecutive updates may result which is undesirable. This kind of consecutive triggerings also happen when uncertainty is present [26] and it is necessary to guarantee a specified performance.

In order to guarantee recursive feasibility in MPC setting most of the existing results including [3,4,9,14,26] exploit terminal cost and terminal set constraints. However, making the MPC controller satisfy terminal constraints can degrade performance, and even cause infeasibility in engineering practice.

In comparison, the recent paper by Lu and Maciejowski [18] provides an alternative approach for self-triggered MPC, which makes full use of non-constant control sequences MPC computes at triggering times and can maintain stability and satisfy certain performance requirements without terminal constraints and penalties. Unlike [3,4,9,14,26], where terminal cost and terminal constraints are used, the asymptotic stability is ensured in [18] by exploiting the relaxed dynamic programming (RDP) inequality without terminal constraints [11,12,17]. Furthermore, the occurrence of consecutive updates is significantly reduced by introducing an extra slack variable in the RDP condition.

While all the exisiting self-triggered MPC schemes have been extensively studied in the regulation case, i.e. when the goal is to control the state of the system to the origin (a fixed setpoint), a control law and triggering condition co-design for reference tracking has not been addressed when setpoint is changing with time, e.g., it is a piecewise constant function in time. The synthesis approach of stabilizing MPC may not be viable at the new setpoint and the constraints could be violated [5,15,16]. For these reasons, under piecewise constant reference signals, a regulation self-triggered MPC, even with the extra slack variable technique introduced [18], may exhibit consecutive updates (cf. Fig. 3), which should be avoided to guarantee certain inter-event time that is required by wireless networked control systems (WNCS) [19,23]. Otherwise, if consecutive updates exist, the MPC updates must be computed intensively during some periods. The merits of event-driven MPC are lost.

In this paper, we extend the recent result on self-triggering MPC [18] to the case of changing references. Our approach integrates reference governors [6,7] into a novel control technique which governs the tail of shifted MPC sequences and guarantees constraint satisfaction, stability at the equilibrium and performance in a reference command tracking setting.

The organization of the paper is as follows. In Section 2, we give definitions and preliminary results that will be used in the rest of the paper to formulate the self-triggered tracking MPC problem. In Section 3, the relaxed dynamic programming approach is proposed with an integration of reference governor design which for piecewise constant in time reference commands is able to reduce the number of consecutive triggering times. Systematic implementation algorithms are developed for the design of self-triggered tracking policy. An illustrative example is presented in Section 4, and we conclude the paper in Section 5.

Notation: Let $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$ denote the set of real numbers, non-negative real numbers, integers, and non-negative integers, respectively, and let $\mathbb{Z}_{[a,b)}$ denote the set $\{\phi \in \mathbb{Z} \mid a \leq \phi < b\}$. Throughout this paper, t denotes sampling time, and k denotes the count of time-steps within the prediction horizon. Given two sets $\mathscr{X}$, $\mathscr{Y} \subseteq \mathbb{R}^n$, the Minkowski set addition is defined by $\mathscr{X} \oplus \mathscr{Y} := \{x + y | x \in \mathscr{X}, y \in \mathscr{Y}\}$. The Pontryagin set difference is defined by $\mathscr{X} \ominus \mathscr{Y} := \{z | z \oplus \mathscr{Y} \subseteq \mathscr{X}\}$. The ball of radius ϵ is denoted by $\mathscr{B}(\epsilon) = \{x \in \mathbb{R}^n : \|x\| \leq \epsilon\}$. For a given set $\mathscr{P}$ containing the origin, we let $\mathrm{int}_\epsilon(\mathscr{P})$ denote the ϵ-interior of $\mathscr{P}$, i.e., $\mathrm{int}_\epsilon(\mathscr{P}) \triangleq \mathscr{P} \ominus \mathscr{B}(\epsilon)$. Finally $\|x\|_Q^2 := \frac{1}{2}x^{\mathrm{T}}Qx$.

2 Problem Setup

Consider a linear system,

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = \bar{x}, \tag{1}$$
$$y(t) = Cx(t) + Du(t) - g(t), \tag{2}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$ and $g(t) \in \mathbb{R}^p$ are the state, input, (generalized) output and external signal at time instant t, respectively. The convex sets $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathbb{U} \subseteq \mathbb{R}^m$ are closed sets which represent state and input constraints and contain the origin in their interiors. We assume that:

(i) (A, B) is stabilizable.
(ii) Matrix D is of full column rank.

For the tracking problem, we can view $g(t)$ as an *artificial setpoint*, g_{sp}, or an *artificial reference function*, for instance, $g_{\mathrm{sp}} = Cx_{\mathrm{s}} + Du_{\mathrm{s}}$ where $(x_{\mathrm{s}}, u_{\mathrm{s}})$ is a steady state and input pair which guarantees the artificial setpoint g_{sp} is equal (or closest) to the desired setpoint r_{sp}.

For unconstrained systems, g_{sp} would essentially coincide with and be set to the desired reference setpoint r_{sp}. Then, from (1)-(2), the steady state satisfies

$$\left[\begin{array}{c:c} I - A & -B \\ \hdashline C & D \end{array}\right] \begin{bmatrix} x_{\mathrm{s}} \\ u_{\mathrm{s}} \end{bmatrix} = \begin{bmatrix} 0 \\ g_{\mathrm{sp}} \end{bmatrix}. \tag{3}$$

The matrix in the left-hand side of equation (3) is an $(n + p) \times (n + m)$ matrix. From linear algebra, for the linear equation (3) to have a solution for all g_{sp}, it is sufficient that the rows of the matrix on the left-hand side are linearly independent, which requires $p \leq m$. However, normally we have more (generalized) outputs than manipulated references. So we choose a matrix H and set $g_c = Hg$ to select particular linear combinations of the generalized outputs. The variable $g_c \in \mathbb{R}^{n_c}$ is referred to as the *controlled variable*. In particular, if the columns of the matrix on the left-hand side are linearly independent, the linear equation (3) has a unique solution. If the solution is non-unique, the steady state pair $(x_{\mathrm{s}}, u_{\mathrm{s}})$ can be determined by solving an optimization problem

$$\begin{aligned} \min_{x_{\mathrm{s}}, u_{\mathrm{s}}, g_{\mathrm{sp}}} \;& (u_{\mathrm{s}} - u_{\mathrm{sp}})^{\mathrm{T}} R_{\mathrm{s}} (u_{\mathrm{s}} - u_{\mathrm{sp}}) + \phi(H g_{\mathrm{sp}} - r_{\mathrm{sp}}), \quad (4) \\ \text{s.t.} \;& \begin{bmatrix} I - A & -B \\ HC & HD \end{bmatrix} \begin{bmatrix} x_{\mathrm{s}} \\ u_{\mathrm{s}} \end{bmatrix} = \begin{bmatrix} 0 \\ H g_{\mathrm{sp}} \end{bmatrix}, \\ & x_{\mathrm{s}} \in \mathrm{int}_{\epsilon}(\mathbb{X}), \\ & u_{\mathrm{s}} \in \mathrm{int}_{\epsilon}(\mathbb{U}), \end{aligned}$$

where the first term penalizes the control effort w.r.t the desired steady input u_{sp} and $\phi(H g_{\mathrm{sp}} - r_{\mathrm{sp}}) = \|H g_{\mathrm{sp}} - r_{\mathrm{sp}}\|_{\infty}$ penalizes the deviation between the desired setpoint r_{sp} and the artificial setpoint g_{sp}.

The set of admissible setpoints such that the constraints are not active is defined as follows:

$$\mathscr{G}_{\mathrm{sp}} = \{g_{\mathrm{sp}} = C x_{\mathrm{s}} + D u_{\mathrm{s}} : x_{\mathrm{s}} \in \mathrm{int}_{\epsilon}(\mathbb{X}),\ u_{\mathrm{s}} \in \mathrm{int}_{\epsilon}(\mathbb{U})\}.$$

2.1 Finite-horizon Tracking MPC

The objective of the reference tracking problem is to steer the output $y(t)$ to zero, while keeping $H g_{\mathrm{sp}}$ as close as possible to r_{sp}. Here g_{sp} can be computed based on (4).

In tracking MPC, we take a finite horizon $N \in \mathbb{Z}_+$ and solve the following optimization problem at each sampling time, t:

$$\begin{aligned} \min_{\mathbf{u} \triangleq [u_0^{\mathrm{T}}, \cdots, u_{N-1}^{\mathrm{T}}]^{\mathrm{T}}} \;& J^{(N)}(\bar{x}, \bar{g}, \mathbf{u}) \triangleq \sum_{k=0}^{N-1} \|y_k\|_2^2, \quad (5) \\ \text{s.t.} \;& x_k \in \mathbb{X}, \quad k = 1, \ldots, N, \\ & u_k \in \mathbb{U}, \quad k = 0, 1, \ldots, N-1, \\ & x_0 = \bar{x}, \\ & g_t = \bar{g}, \\ & x_{k+1} = A x_k + B u_k, \quad k = 0, 1, \ldots, N-1, \\ & y_k = C x_k + D u_k - g_t, \quad k = 0, 1, \ldots, N-1. \end{aligned}$$

Solving the above optimization problem at each sampling time for a particular $\bar{x}$ and $\bar{g}$ leads to a unique sequence of optimal control laws from time t to time $t + N - 1$, given by $\mathbf{u}^*(\bar{x}, \bar{g}) = [u_0^{*\mathrm{T}}(\bar{x}, \bar{g}), u_1^{*\mathrm{T}}(\bar{x}, \bar{g}), \ldots, u_{N-1}^{*\mathrm{T}}(\bar{x}, \bar{g})]^{\mathrm{T}}$.

The (finite-horizon) value function is defined as

$$V^{(N)}(x(t), g(t)) \triangleq J^{(N)}(x(t), g(t), \mathbf{u}^*). \quad (6)$$

The tracking MPC control law is given by applying the first control move of the open-loop optimal control sequence $\mathbf{u}^*(x(t), g(t))$ to the system, i.e.

$$u(t) = \mu(x(t), g(t)) := u_0^*(x(t), g(t)). \quad (7)$$

Then the closed-loop system is given by

$$x(t+1) = A x(t) + B \mu(x(t), g(t)), \quad (8)$$

$$y_{\mu}(t) = C x(t) + D \mu(x(t), g(t)) - g(t). \quad (9)$$

2.2 Recursive Feasibility

In this subsection we review the background definitions and results from [18] needed for the subsequent developments.

Definition 1 *A control sequence* $\mathbf{u} = \{u(0), u(1), \ldots, u(N-1)\}$ *is said to be* admissible *for* $x(0) \in \mathbb{X}$, *if* $(x(t), u(t)) \in \mathbb{X} \times \mathbb{U}$ *for all* $t \in \{0, 1, \ldots, N-1\}$. *The set of all admissible control sequences of length* N *is denoted by* $\mathscr{U}^N(x(0))$.

The N step feasible region is defined as

$$\mathbb{I}_N := \{x \in \mathbb{X} : \mathscr{U}^N(x) \neq \emptyset\}.$$

The region $\mathbb{I}_{\infty}$ is called *viability kernel* [2], which characterizes the set of the infinite horizon feasible initial conditions of system (1) subject to input and state constraints.

Remark 1 *Any admissible equilibrium point* x_{s} *is in the viability kernel. If the initial state is in an equilibrium point, then the proposed tracking MPC will be feasible.*

The sequence of feasible sets $\mathbb{I}_N$'s becomes *stationary*, if there exists $N_0 \in \mathbb{Z}_+$, such that $\mathbb{I}_N = \mathbb{I}_{N_0}$ holds for all $N \geq N_0$.

Definition 2 *A set* $\mathscr{P} \subseteq \mathbb{R}^n$ *is called a* (controlled) positively invariant (PI) set *or a* viable set *for the closed-loop system (1), if* $\mathscr{P} \subseteq \mathbb{X}$ *and for all* $x \in \mathscr{P}$, *there is a* $u \in \mathbb{U}$, *such that* $Ax + Bu \in \mathscr{P}$.

$\mathbb{I}_{\infty}$ is also called the *maximal positively invariant (MPI) set*, which includes all the possible PI set $\mathscr{P}$, i.e. $\mathscr{P} \subseteq \mathbb{I}_{\infty}$.

Definition 3 *A set $\mathscr{P}$ is called* RH N-invariant *or* recursively feasible *with respect to a horizon $N \in \mathbb{Z}_+$ if $\mathscr{P} \subseteq \mathbb{I}_N$ is a PI set for the closed-loop system (8) under the MPC controller (7) with a receding horizon (RH) N, i.e.*

$$x(0) \in \mathscr{P} \Rightarrow x(t) \in \mathscr{P}, \quad \forall t \in \mathbb{Z}_+.$$

The following proposition shows that for a sufficiently large horizon N, MPC controller will generate recursive feasibility on the whole feasibility kernel $\mathbb{I}_\infty$. This property is inferred from *stationarity* of the feasible sets $\mathbb{I}_N$'s [2].

Proposition 1 *Suppose that $g_{\mathrm{sp}} \in \mathscr{G}_{\mathrm{sp}}$ in (5). If $V^{(\infty)}(x(t), g_{\mathrm{sp}}) < c$ holds for some $c \in \mathbb{R}_+$ and all $x(t) \in \mathbb{I}_\infty$ and for all $g_{\mathrm{sp}} \in \mathscr{G}_{\mathrm{sp}}$, the feasible sets $\mathbb{I}_N$'s become stationary for some $N_0 \in \mathbb{Z}_+$, i.e., $\mathbb{I}_{N_0} = \mathbb{I}_{N_0+1} = \mathbb{I}_{N_0+2} = \cdots = \mathbb{I}_\infty$.*

Assumption 1 *In this paper it is assumed that*

$$\sup_{x \in \mathbb{I}_\infty, g \in \mathscr{G}_{\mathrm{sp}}} V^{(\infty)}(x, g) = c < \infty.$$

For a given horizon N and a positive scalar ν, in order to determine a RH N-invariant set, we define the sub-level $S_\nu^N \subset \mathbb{R}^n \times \mathbb{R}^{n_c}$ of finite horizon value function $V^N(x, g)$

$$S_\nu^N = \{(x, g) \in \mathbb{I}_N \times \mathbb{R}^{n_c} : V^{(N)}(x, g) \leq \nu\}.$$

2.3 Relaxed Dynamic Programming

In this paper, we use the relaxed dynamic programming result in [17] to develop a triggering condition for self-triggered MPC to ensure stability and to obtain a performance guarantee in terms of the infinite horizon quadratic cost.

The next proposition is a variant of the main proposition stated in [12, 17] for approximating the Bellman's equation based on the finite-horizon value function $V^{(N)}(x(t), g(t))$ defined in Section 2.1 and its corresponding optimal control policy $\mu(x(t), g(t))$.

Proposition 2 *Consider the system (1)-(2) with the feedback control law $\mu(x, g)$, and suppose that the following inequality is satisfied:*

$$V^{(N)}(x(t), g(t)) \geq V^{(N)}(x(t+1), g(t+1)) + \alpha \|y_\mu(t)\|_2^2, \tag{10}$$

for a given scalar $\alpha \in (0, 1]$ and all $(x(t), g(t)) \in S_\nu^N$. Then,

$$\alpha \sum_{t=0}^{\infty} \|y_\mu(t)\|_2^2 \leq \sup_{g \in \mathscr{G}_{\mathrm{sp}}} V^{(\infty)}(x(0), g), \tag{11}$$

where $x(t+1)$ and $y_\mu(t)$ are obtained by applying $\mu(x(t), g(t))$ to the closed-loop system, i.e., $x(t+1) = Ax(t) + B\mu(x(t), g(t))$ and $y_\mu(t) = Cx(t) + D\mu(x(t), g(t)) - g(t)$.

Throughout this paper, we make the following assumption.

Assumption 2 *The control horizon $N \geq N_0$ is assumed so that the RDP inequality (10) is satisfied for a specified $\alpha \in (0, 1]$ and all $(x(t), g(t)) \in S_\nu^N$.*

Results to determine the smallest horizon N_0 and computation of α in the RDP inequality for stabilizing horizons N for various systems can be found in the references [1, 10, 25].

While under reasonable assumptions on the system (1), we can ensure N_0 exists, it may be difficult to compute it a priori. Our triggering mechanism will detect if N was chosen too small, such that it can be adapted [18].

2.4 Reference Governor

Consider a control law for tracking a constant reference $r(t) = r_{\mathrm{sp}}$, of the form,

$$u(t) = Kx + \Gamma r, \tag{12}$$

where K is a feedback gain matrix such that $A + BK$ is Schur and Γ is a feedforward gain. The goal is to make the selected output $Hy(t)$ to be zero. Note that the steady state for the closed-loop system is given by

$$x_{\mathrm{s}} = (I - A - BK)^{-1} B\Gamma r \tag{13}$$

Hence, Γ should be chosen as the right inverse of $H((C + DK)(I - (A + BK))^{-1}B + D)$ to guarantee $r = Hg_{\mathrm{sp}} = HCx_{\mathrm{s}}$.

The constraints $(x(t), u(t)) \in \mathbb{X} \times \mathbb{U}$ can be expressed as inequality constraints

$$Ex(t) + Fu(t) \leq h, \quad t \geq 0 \tag{14}$$

Applying the control law (12) and replacing the desired reference $r(t)$ with $v(t)$ in the system (1), the closed-loop system has the form,

$$x(t+1) = \Phi x(t) + Gv(t), \quad x(0) = \bar{x}, \tag{15}$$

where $\Phi = A + BK$ is Schur and $G = B\Gamma$. The inequality constraints (14) can be restated as

$$\begin{bmatrix} \Psi & \Theta \end{bmatrix} \begin{bmatrix} v(t) \\ x(t) \end{bmatrix} \leq h, \quad t \geq 0 \tag{16}$$

where $\Theta = E + FK$ and $\Psi = F\Gamma$.

For system (15) with constraints (16), we define the *maximal constraint admissible set* O_∞ for constant input v as

$$O_\infty = \left\{ (v, x(0)) : \begin{bmatrix} \Psi & \Theta \end{bmatrix} \begin{bmatrix} v \\ x(t) \end{bmatrix} \leq h, \ \forall t \in \mathbb{Z}_+ \right\}. \tag{17}$$

A finitely determined approximation of O_∞, $\tilde{O}_\infty$, can be obtained by

$$\tilde{O}_\infty = \left\{ \begin{bmatrix} \Psi & \Theta \\ \Theta G + \Psi & \Theta\Phi \\ \vdots & \vdots \\ \Lambda & \Theta\Phi^{t^\star} \\ \Xi & 0 \end{bmatrix} \begin{bmatrix} v \\ x(0) \end{bmatrix} \leq \begin{bmatrix} h \\ h \\ \vdots \\ h \\ (1-\epsilon)h \end{bmatrix} \right\} \tag{18}$$

where $\Lambda = \Theta(I - \Phi)^{-1}(I - \Phi^{t^\star})G + \Psi$, $\Xi = \Theta(I - \Phi)^{-1}G + \Psi$ and t^* is sufficiently large.

$\tilde{O}_\infty$ can be simplified by eliminating almost redundant inequalities and tightening the remaining constraints to obtain a closed set P satisfying

$$P \subseteq \tilde{O}_\infty, \tag{19}$$

which can be expressed as set of linear inequalities of the form

$$P = \{(v, x) : M_x x + M_v v \leq b\}. \tag{20}$$

The reference governor [6] behaves as a pre-filter which, based on the current state $x(t)$ and the desired reference $r(t)$, generates a modified reference $v(t)$ which fulfills the constraints $(x(t), u(t)) \in \mathbb{X} \times \mathbb{U}$. The updates for $v(t)$ take the form

$$v(t) = v(t-1) + \kappa(t)(r(t) - v(t-1)), \tag{21}$$

where the scalar $\kappa(t) \in [0, 1]$ is chosen by solving the optimization problem,

$$\begin{aligned} \kappa(t) \triangleq & \max_{\kappa \in [0,1]} \kappa \\ \text{s.t.} \quad & v = v(t-1) + \kappa(r(t) - v(t-1)), \\ & (v, x(t)) \in P \subseteq \tilde{O}_\infty. \end{aligned} \tag{22}$$

The computation of the scalar κ reduces to finding the maximum κ such that

$$M_x x(t) + M_v[v(t-1) + \kappa(t)(r(t) - v(t-1))] \leq b.$$

The solution can be obtained in closed form as

$$\kappa(t) = \min\left\{ \min_{j \in \mathbb{J}^+} \left\{ \frac{b_j - M_{x,j}x(t) - M_{v,j}v(t-1)}{M_{v,j}(r(t) - v(t-1))} \right\}, 1 \right\} \tag{23}$$

where $M_{x,j}$ and $M_{v,j}$ are the jth entry of M_x and M_v, b_j is the jth row of b, and $\mathbb{J}^+$ is the set of indices such that $M_{v,j}(r(t) - v(t-1)) > 0$.

3 RDP-Based Approach for Tracking

In this section, we will adapt the relaxed dynamic programming inequality in Proposition 2 to the self-triggered tracking MPC setting.

Define the triggering times $\{t_l \mid l \in \mathbb{Z}_+\}$, which satisfy $t_{l+1} > t_l$ for all $l \in \mathbb{Z}_+$ and $t_{l+1} - t_l < N$. Within the time interval $[t_l, t_{l+1})$, we set

$$u(t) = \tilde{\mu}(t, x(t_l), g(t_l)) := u^*_{(t-t_l)}(x(t_l), g(t_l)), t \in \mathbb{Z}_{[t_l, t_{l+1})}. \tag{24}$$

When MPC update (5) is triggered at time t_l, we have to decide on both the control and the next triggering time t_{l+1} which should be as large as possible while reference tracking is achieved and a certain required performance is guaranteed. The computation of t_{l+1} will be based on checking the RDP inequality and for the setpoint changes.

In the self-triggered tracking MPC setting, multiple open-loop control moves of MPC sequence at time t_l may be applied before the next MPC update at time t_{l+1} is executed. We keep $g(t)$ constant in-between the triggering times t_l and t_{l+1}, and we amend the RDP condition as follows:

$$\begin{aligned} V^{(N)}(x(t_l), g(t_l)) \geq & V^{(N)}(x(t_{l+1}), g(t_{l+1})) \\ & + \alpha \sum_{t=t_l}^{t_{l+1}-1} \|y_{\tilde{\mu}}(t)\|_2^2, \end{aligned} \tag{25}$$

where $\sum_{t=t_l}^{t_{l+1}-1} \|y_{\tilde{\mu}}(t)\|_2^2$ denotes the sum of the running costs at the triggering times $t_l, t_l + 1, \ldots, t_{l+1} - 1$ with the control policy $u(t)$ defined as in (24). As the optimal value $V^{(N)}(x(t_{l+1}), g(t_{l+1}))$ at the next triggering time is not available at t_l, we will construct an upper bound for it. Besides, we will also exploit an extra slack variable which reflects the decay of the Lyapunov function $V^{(N)}(x(t), g(t))$ at the previous triggering times. The main theorem of this paper is presented below. It demonstrates that after all of the above mentioned modifications to the RDP inequality, a certain bound on performance and reference tracking are still guaranteed.

Theorem 1 *Suppose $(x(0), g(0)) \in S_\nu^N$ and an upper bound $\bar{V}^{(N)}(x(t), g(t))$ can be found for $t \in \{t_l \mid l \in \mathbb{Z}_+\}$*

such that

$$\bar{V}^{(N)}(x(t), g(t)) \geq V^{(N)}(x(t), g(t)). \tag{26}$$

Suppose, furthermore, the inequality

$$V^{(N)}(x(t_l), g(t_l)) - \bar{V}^{(N)}(x(t_{l+1}), g(t_l)) \geq e(t_l) + \alpha \sum_{t=t_l}^{t_{l+1}-1} \|y_{\tilde{\mu}}(t)\|_2^2, \tag{27}$$

is enforced for a given scalar $\alpha \in (0,1)$, where the sequence $\{e(t_l)\}$ is defined in (28) for all $l \geq 2$ and if $g(t_l) \neq g(t_{l-1})$ or $t = t_1$, we set $\bar{V}^{(N)}(x(t_l), g(t_{l-1})) = V^{(N)}(x(t_l), g(t_l))$. Then:

$$\alpha \sum_{t=t_l}^{\infty} \|y_{\tilde{\mu}}(t)\|_2^2 \leq \sup_{g \in \mathscr{G}_{\text{sp}}} V^{(\infty)}(x(t_l), g) = c, \tag{29}$$

and $y_{\tilde{\mu}}(t)$ asymptotically converges to 0.

The proof of this theorem can be found in Appendix A.

We note that as the update is triggered every time the setpoint changes, the proposed approach is most effective in phases when the reference command is piecewise constant or can be approximated by piecewise constant.

At an MPC update time $t_l \in \mathbb{Z}_+$ with $l \in \mathbb{Z}_+$, we compute the MPC control update according to (5). In order to implement our RDP-based triggering scheme, after obtaining $\mathbf{u}^*(x(t_l), g(t_l)) = [u_0^{*\mathrm{T}}(x(t_l), g(t_l)), u_1^{*\mathrm{T}}(x(t_l), g(t_l)), \ldots, u_{N-1}^{*\mathrm{T}}(x(t_l), g(t_l))]^{\mathrm{T}}$ at time t_l, the first step is to find the last component $u^*_{\bar{N}-1}$ in $\mathbf{u}^*$ sequence such that $x(t_l + \bar{N}) \in Proj_x(P)$, where P is given by (19) and where $\bar{N} \in \mathbb{Z}_{[1,N-1]}$. Algorithm 1 is developed for this purpose.

Algorithm 1 Determine $\bar{N}$

Input: Triggered state $x(t_l)$, $\mathbf{u}^*(x(t_l), g(t_l))$
Output: $\bar{N}$, $x(t_l + \bar{N})$

```
for k = 1 to N do
    compute x(t_l + k) = Ax(t_l + k - 1) + Bu*_k
    store x(t_l + k)
end for
for k = N - 1 : -1 : 0 do
    if x(t_l + k + 1) ∈ Proj_x(P) then
        save N̄ = k
        break
    end if
    store x(t_l + N̄)
end for
```

The next MPC update time t_{l+1} can be calculated by

$$t_{l+1} = t_l + \mathscr{N}_{t_l}(x(t_l)), \tag{30}$$

where the inter-triggering interval $\mathscr{N}_{t_l}(x(t_l))$ is given by

$$\mathscr{N}_{t_l}(x(t_l)) \triangleq \max\{N_{t_l} \in \mathbb{Z}_{[1,\bar{N}-1]}\} \tag{31}$$

$$\text{s.t. } (i)\ V^{(N)}(x(t_l), g(t_l)) - \bar{V}^{(N)}(x(t_l + N_{t_l}), g(t_l)) \geq e(t_l) + \alpha \left(\sum_{t=t_l}^{t_l+N_{t_l}-1} \|y_{\tilde{\mu}}(t)\|_2^2 \right), \tag{32}$$

$$(ii)\ g(t_l + N_{t_l}) = g(t_l). \tag{33}$$

In order to calculate the upper bound $\bar{V}^{(N)}(x(t_l + N_{t_l}), g(t_l))$ and the forward predicted state $\bar{x}(t_l + \bar{N} + i)$ for $i \in \mathbb{Z}_{[1,N_{t_l}]}$ at time t_l, we apply a "shifted" input sequence $\bar{U}_N(x(t_l + N_{t_l})) = [u^{*}_{N_{t_l}}{}^{\mathrm{T}}(x(t_l), g(t_l)), \ldots, u^{*}_{\bar{N}-1}{}^{\mathrm{T}}(x(t_l), g(t_l)), \bar{u}^{\mathrm{T}}(t_l + \bar{N}), \ldots, \bar{u}^{\mathrm{T}}(t_l + N_{t_l} + \bar{N} - 1)]^{\mathrm{T}}$. The reason for adding the tail inputs $\bar{u}(t_l + \bar{N} + i - 1)$, $i \in \mathbb{Z}_{[1,N_{t_l}]}$ for extending the sequence at the end is to avoid violating the constraints.

For a given $x(t) \in Proj_x(P)$, we can determine $v(t)$ to guarantee $(v(t), x(t)) \in P \subseteq \tilde{O}_\infty$, for instance, by solving the following QP:

$$\begin{aligned} \min_{v} \quad & (v(t) - g(t))^{\mathrm{T}}(v(t) - g(t)), \\ \text{s.t.} \quad & M_x x(t) + M_v v(t) \leq b. \end{aligned} \tag{34}$$

To get $\bar{u}(t_l + \bar{N} + i - 1)$, $i \in \mathbb{Z}_{[1,N_{t_l}]}$, we set $v(t_l + \bar{N})$ from (34) for $t = t_l + \bar{N}$ and solve (23) for $v(t_l + \bar{N} + i)$ for $i \in \mathbb{Z}_{[1,N_{t_l}]}$. The sequence $\bar{x}(t_l + \bar{N} + i)$ and then $\bar{u}(t_l + \bar{N} + i - 1)$ can be obtained by forward simulation.

Then we have Algorithm 2:

Algorithm 2 Tail sequence computation for $\bar{U}_N(x(t_l + N_{t_l}))$

Input: $r(t_l + \bar{N}) := g(t_l)$, $v(t_l + \bar{N})$, $x(t_l + \bar{N})$
Output: $\bar{x}(t_l + \bar{N} + i)$, $\bar{u}(t_l + \bar{N} + i - 1)$, $i \in \mathbb{Z}_{[1,N_{t_l}]}$

```
x̄(t_l + N̄ + 1) = Φx(t_l + N̄) + Gv(t_l + N̄) and set r(t_l + N̄ + 1) = r(t_l + N̄)
for i = 1 to N_{t_l} do
    compute κ(t_l + N̄ + i) by (23)
    update v(t_l + N̄ + i) = v(t_l + N̄ + i − 1) + κ(t_l + N̄ + i)(r(t_l + N̄ + i) − v(t_l + N̄ + i − 1))
    compute x̄(t_l + N̄ + i) = Φx̄(t_l + N̄ + i − 1) + Gv(t_l + N̄ + i − 1)
    update ū(t_l + N̄ + i − 1) = Kx̄(t_l + N̄ + i − 1) + Γv(t_l + N̄ + i − 1)
end for
```

Remark 2 *The requirement for Algorithm 2 to work is to guarantee $(v(t_l + \bar{N}), x(t_l + \bar{N})) \in P \subseteq \tilde{O}_\infty$. In between the triggering times, the closed-loop state trajectory might go out of P.*

$$e(t_l) = \begin{cases} 0, & \text{if } g(t_l) \neq g(t_{l-1}) \text{ or } t = t_0, \\ e(t_{l-1}) + \alpha \sum_{t=t_{l-1}}^{t_l-1} \|y_{\tilde{\mu}}(t)\|_2^2 + \bar{V}^{(N)}(x(t_l), g(t_{l-1})) - \bar{V}^{(N)}(x(t_{l-1}), g(t_{l-2})), & \text{otherwise.} \end{cases} \tag{28}$$

Remark 3 *Typically in practice, we start the system in steady-state corresponding to some v_0 for which constraints strictly hold. Then the system starts responding to changing reference $g(t)$ for $t > 0$. So if we start with $x(0)$ in a steady state, Algorithm 1 and Algorithm 2 will guarantee that at every triggering point $x(t_l)$ is in $Proj_x(P)$.*

Hence, the upper bound can be defined as

$$\bar{V}^{(N)}(x(t_l + N_{t_l}), g(t_l)) \triangleq J^{(N)}(x(t_l + N_{t_l}), g(t_l), \bar{U}_N(x(t_l + N_{t_l}))). \tag{35}$$

Remark 4 *In the existing self-triggered MPC papers [8, 9, 13, 14, 18, 26], the authors add the tail of "shifted" input sequence by keeping the last input element constant, i.e. $\bar{U}_N(x(t_l + N_{t_l})) = [{u^*_{N_{t_l}}}^{\mathrm{T}}(x(t)), \ldots, {u^*_{N-1}}^{\mathrm{T}}(x(t)), \mathrm{repmat}(u_{\mathrm{s}}^{\mathrm{T}}, 1, N_{t_l})]^{\mathrm{T}}$, or by adding constant steady input, $\bar{U}_N(x(t_l + N_{t_l})) = [{u^*_{N_{t_l}}}^{\mathrm{T}}(x(t)), \ldots, {u^*_{N-1}}^{\mathrm{T}}(x(t)), \mathrm{repmat}({u^*_{N-1}}^{\mathrm{T}}(x(t)), 1, N_{t_l})]^{\mathrm{T}}$. These tails add the control and state mismatch for the constructed $\bar{V}^{(N)}(x(t_l + N_{t_l}), g(t_l))$, especially when the reference changes and it is moved close to the constraint boundary, this shifted sequence will make the open-loop trajectory have overshoot and thus violate the constraints very easily. Thus consecutive updates keep occurring quite often in the implementation (cf. Section 4). Otherwise, the closed-loop trajectory would diverge.*

Under Assumption 1, in virtue of Theorem 1, we can conclude that if the RDP checking condition (32) is satisfied, $(x(t_l+N_{t_l}), g(t_l+N_{t_l})) \in S_\nu^N$. Thus, state constraints are always satisfied by our RDP-based self-triggered MPC scheme and S_ν^N is a RH N-invariant w.r.t. N.

4 Illustrative Example

We consider a helicopter flight envelope protection example studied in [22,24]. The linearized continuous-time model for the helicopter dynamics is described by

$$\dot{x} = A_c x + B_c u,$$

where five states and one input are:

- γ: forward speed;
- q: pitch rate;
- θ: pitch angle;
- a: pitch angle of the virtual rotor disc;
- c: angle of the rotor stabilizer bar;
- δ_s: swash plate angle;

and

$$A_c = \begin{bmatrix} -0.0505 & 0 & -9.81 & -9.81 & 0 \\ -0.0561 & 0 & 0 & 82.6 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -21.7391 & 14 \\ 0 & -1 & 0 & 0 & -0.342 \end{bmatrix},$$

$$B_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2.174 \\ -0.7573 \end{bmatrix}, \quad x = \begin{bmatrix} v \\ q \\ \theta \\ a \\ c \end{bmatrix}, \quad u = \delta_s.$$

The discrete-time linear model of (1) is obtained assuming a sampling frequency of 60Hz. As the problem is a tracking problem, the output is chosen to be $y_k = Cx_k + Du_k - g_{k+t}$, where

$$C = \begin{bmatrix} 7.0711 & 0 & 0 & 0 & 0 \\ 0 & 0.3162 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3.1623 & 0 \\ 0 & 0 & 0 & 0 & 0.3162 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$$HC = \begin{bmatrix} 7.0711 & 0 & 0 & 0 & 0 \end{bmatrix}, HD = 0$$

and g_t is the set-point for γ provided by a human operator or a higher level planner in the overall helicopter control system.

The state and input constraints are enforced within the ranges

$$\begin{bmatrix} -5 \\ -5 \\ -3 \\ -1 \\ -2 \end{bmatrix} \leq x \leq \begin{bmatrix} 10 \\ 5 \\ 3 \\ 1 \\ 2 \end{bmatrix}, -5 \leq \delta_s \leq 5.$$

The control horizon is chosen as $N = 40$ and the performance degradation parameter as $\alpha = 0.7$. The simulation results are presented in Fig. 1 and Fig. 2.

If we do not add the self-triggering mechanism to the MPC algorithm, it takes 510 MPC updates. The results by Lu et al. [18] are shown in the first and third rows in Fig. 1, which has 78 MPC updates. The responses with the proposed triggering scheme are shown in the second and fourth rows in Fig. 1. In this case, our self-triggered tracking MPC only needs 19 updates. Hence, the proposed strategy can significantly reduce the number of MPC update times while achieving reference tracking. The triggering instants are recorded in Fig. 3.

In our triggering scheme, α plays a role of a "discount factor" and the values of α close to 1 result in improved performance but more frequent update triggering.

In order to further demonstrate the effectiveness of our controller in reducing computations while still maintaining performance guarantees, we next increase the value of α in our self-triggered tracking MPC and plot the triggering instants for $\alpha = 0.8$ and $\alpha = 0.9$ in Fig. 4. For $\alpha = 0.8$, it takes 23 updates and for $\alpha = 0.9$, it takes 29 updates. This shows our self-triggered tracking MPC is able to save computation and communication resources with no noticeable performance degradation.

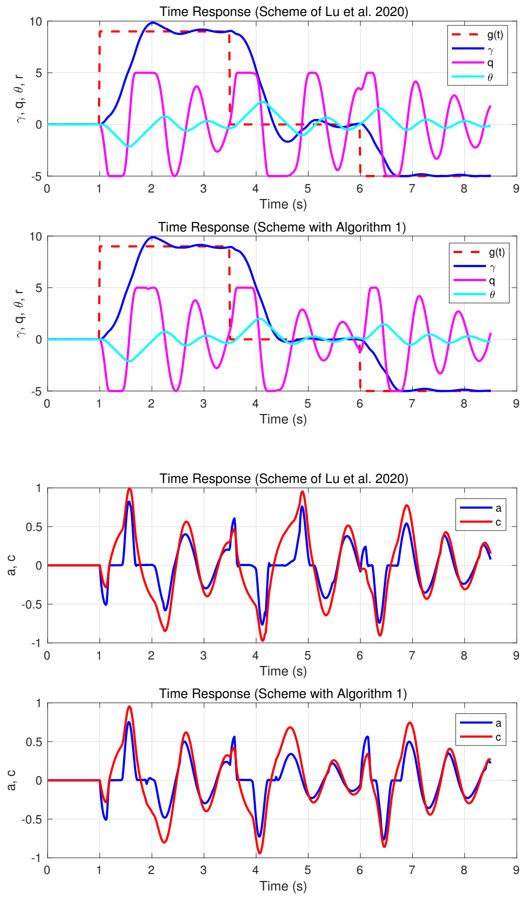

Figure 1. State response trajectories with $\alpha = 0.7$. The upper figure shows the fuselage states and the reference (the speed reference is in red dash line, γ is in blue, q is in magenta and θ is in cyan), and the bottom figure shows the rotor states (a is in blue and c is in red).

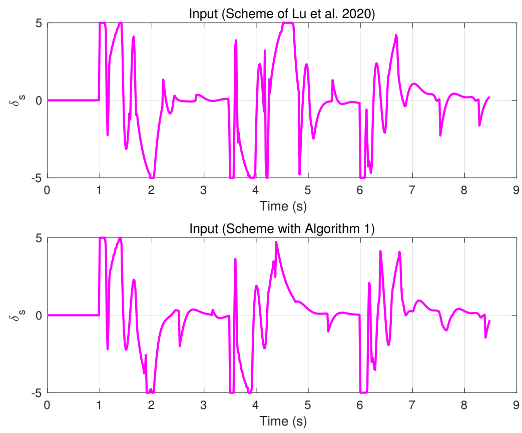

Figure 2. Control input trajecory with $\alpha = 0.7$.

5 Conclusions

This paper proposed a self-triggered tracking MPC co-design procedure for constrained linear systems based on the relaxed dynamic programming inequality and reference governor scheme. The inter-triggering time is maximized by governing the tails of shifted control sequences for constructing triggering conditions of tracking MPC such that the overall closed-loop system can not only maintain asymptotic stability, but also achieve a certain prescribed performance level. The illustrative example showed that the number of consecutive updates in the self-triggered tracking MPC is significantly reduced compared to the existing self-triggered MPC schemes for regulation problem. An extension of the idea to robust case is being explored currently.

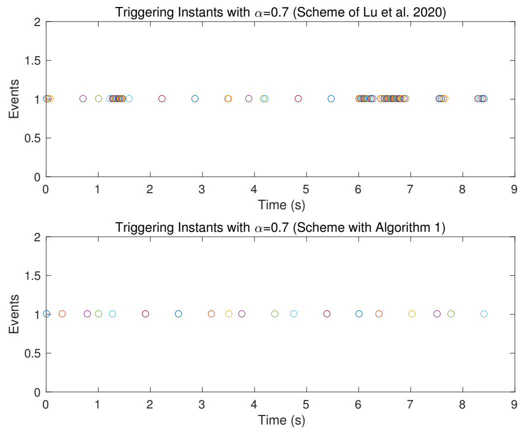

Figure 3. Event triggering instants with $\alpha = 0.7$. The triggering instants are marked with the circles with the value 1.

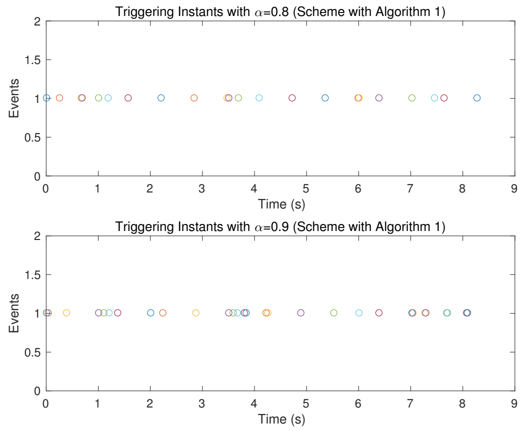

Figure 4. Event triggering instants with $\alpha = 0.8$ (upper) and $\alpha = 0.9$ (lower).

Appendix A Proof of Theorem 1

The proof of Theorem 1 can be divided into two parts. First, it is proved that for any change of the reference, the recursive feasibility holds, i.e. if the reference is abruptly changed, the controller is well defined. In the second part, we prove the asymptotic convergence, i.e. if the reference holds constant subsequently, the system converges to the reference.

Recursive feasibility: In our setting, the inequality $V^{(N)}(x, g) \leq V^{(\infty)}(x, g)$ holds. Hence, S_ν^N contains S_ν^∞.

The feasibility region of the controller is $\mathbb{I}_N$. From Proposition 1, taking $N \geq N_0$ we have that $\mathbb{I}_N = \mathbb{I}_\infty$ (stationarity of the feasible sets). From Assumption 1, $V^{(\infty)}(\mathbb{I}_\infty, g)$ is bounded for any $g \in \mathscr{G}_{\text{sp}}$, then S_ν^N contains $\mathbb{I}_\infty$ for $\nu \geq \sup_{g\in\mathscr{G}_{\text{sp}}} V^{(\infty)}(\mathbb{I}_\infty, , g)$, hence S_ν^N also contains $\mathbb{I}_\infty$, which implies that $\mathbb{I}_\infty$ is contained in the domain of attraction. Therefore, the domain of attraction is forward invariant for any g, and then, it is recursively feasible under any change on the reference.

Asymptotic convergence: Consider at triggering time t_l, $g(t_l) \neq g(t_{l-1})$, i.e. t_l is the first triggering time after reference switching from $g(t_{l-1})$ to $g(t_l)$.

Consider a time interval $[t_l, t_{l+T}]$ with $g(t_l) = g(t_{l+1}) = \cdots = g(t_{l+T}) = g \in \mathscr{G}_{\text{sp}}$ and $g(t_{l+T+1}) = \tilde{g} \in \mathscr{G}_{\text{sp}}$, where $g \neq \tilde{g}$ and $T \in \mathbb{Z}_+$.

From (28), we get $e(t_l) = 0$ and derive $e(t_{l+T})$ by induction as

$$\begin{aligned}
e(t_{l+T}) =& e(t_{l+T-1}) + \alpha \sum_{t=t_{l+T-1}}^{t_{l+T}-1} \|y_{\tilde{\mu}}(t)\|_2^2 + \bar{V}^{(N)}(x(t_{l+T}), g) \\
& - \bar{V}^{(N)}(x(t_{l+T-1}), g) = \cdots = e(t_{l+1}) \\
& + \alpha \sum_{t=t_{l+1}}^{t_{l+T}-1} \|y_{\tilde{\mu}}(t)\|_2^2 + \bar{V}^{(N)}(x(t_{l+T}, g) \\
& - \bar{V}^{(N)}(x(t_{l+1}), g) = \alpha \sum_{t=t_l}^{t_{l+T}-1} \|y_{\tilde{\mu}}(t)\|_2^2 \\
& + \bar{V}^{(N)}(x(t_{l+T}), g) - V^{(N)}(x(t_l), g).
\end{aligned}$$

It follows

$$\begin{aligned}
0 \leq \alpha \sum_{t=t_l}^{t_{l+T}-1} \|y_{\tilde{\mu}}(t)\|_2^2 =& e(t_{l+T}) - \bar{V}^{(N)}(x(t_{l+T}), g) \\
& + V^{(N)}(x(t_l), g). \qquad (36)
\end{aligned}$$

Furthermore, from (27), we have

$$e(t_{l+T}) \leq V^{(N)}(x(t_{l+T}), g) - \bar{V}^{(N)}(x(t_{l+T+1}), g).$$

Insert this into (36) gives

$$\begin{aligned}
\alpha \sum_{t=t_l}^{t_{l+T}-1} \|y_{\tilde{\mu}}(t)\|_2^2 \leq& V^{(N)}(x(t_l), g) - \bar{V}^{(N)}(x(t_{l+T}), g) \\
& + V^{(N)}(x(t_{l+T}), g) - \bar{V}^{(N)}(x(t_{l+T+1}), g).
\end{aligned}$$

Because

$$V^{(N)}(x(t_{l+T}), g) - \bar{V}^{(N)}(x(t_{l+T}), g) \leq 0,$$

we get

$$\begin{aligned}
\alpha \sum_{t=t_l}^{t_{l+T}-1} \|y_{\tilde{\mu}}(t)\|_2^2 \leq& V^{(N)}(x(t_l), g) - \bar{V}^{(N)}(x(t_{l+T+1}), g) \\
\leq& V^{(N)}(x(t_l), g) \leq V^{(\infty)}(x(t_l), g).
\end{aligned}$$

As $T \to \infty$, we get (29). Furthermore, given the boundedness of $V^{(\infty)}(x(t_0), g)$, and the positive definiteness of the term $\|y_{\tilde{\mu}}(t)\|_2^2$, we get $y_{\tilde{\mu}}(t)$ asymptotically converges to 0.

This completes the proof. $\square$

Acknowledgements

The first author would like to thank the kind comments from Professor Lars Grüne at the University of Bayreuth for this topic and the anonymous reviewers for their careful reading of the manuscript and comments.

References

[1] N. Altmüller. *Model Predictive Control for Partial Differential Equations.* PhD thesis, University of Bayreuth, Germany, 2014.

[2] A. Boccia, L. Grüne, and K. Worthmann. Stability and feasibility of state constrained MPC without stabilizing terminal constraints. *Systems & Control Letters*, 72:14–21, 2014.

[3] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos. Event-triggered strategies for decentralized model predictive controllers. *IFAC Proceedings Volumes*, 44(1):10068–10073, 2011.

[4] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos. Event-based model predictive control for the cooperation of distributed agents. In *2012 American Control Conference (ACC)*, pages 6473–6478. IEEE, 2012.

[5] A. Ferramosca. *Model Predictive Control of Systems with Changing Setpoints. Control predictivo de sistemas con punto de operación cambiante.* Ph.D. dissertation, Univ. de Sevilla, Sevilla, Spain, 2011.

[6] E. Garone, S. Di Cairano, and I. Kolmanovsky. Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75:306–328, 2017.

[7] E. G. Gilbert and I. Kolmanovsky. Fast reference governors for systems with state and control constraints and disturbance inputs. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 9(15):1117–1141, 1999.

[8] T. Gommans, D. Antunes, T. Donkers, P. Tabuada, and M. Heemels. Self-triggered linear quadratic control. *Automatica*, 50(4):1279–1287, 2014.

[9] T. Gommans and W. Heemels. Resource-aware mpc for constrained nonlinear systems: A self-triggered control approach. *Systems & Control Letters*, 79:59–67, 2015.

[10] L. Grüne. Analysis and design of unconstrained nonlinear mpc schemes for finite and infinite dimensional systems. *SIAM Journal on Control and Optimization*, 48(2):1206–1228, 2009.

[11] L. Grüne and J. Pannek. Stability and suboptimality without stabilizing terminal conditions. In *Nonlinear Model Predictive Control*, pages 121–176. Springer, 2017.

[12] L. Grüne and A. Rantzer. On the infinite horizon performance of receding horizon controllers. *IEEE Transactions on Automatic Control*, 53(9):2100–2111, 2008.

[13] E. Henriksson, D. E. Quevedo, E. G. Peters, H. Sandberg, and K. H. Johansson. Multiple-loop self-triggered model predictive control for network scheduling and control. *IEEE Transactions on Control Systems Technology*, 23(6):2167–2181, 2015.

[14] H. Li, W. Yan, and Y. Shi. Triggering and control codesign in self-triggered model predictive control of constrained systems: With guaranteed performance. *IEEE Transactions on Automatic Control*, 63(11):4008–4015, 2018.

[15] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho. Mpc for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9):2382–2387, 2008.

[16] D. Limon, A. Ferramosca, I. Alvarado, and T. Alamo. Nonlinear MPC for tracking piece-wise constant reference signals. *IEEE Transactions on Automatic Control*, 63(11):3735–3750, 2018.

[17] B. Lincoln and A. Rantzer. Relaxing dynamic programming. *IEEE Transactions on Automatic Control*, 51(8):1249–1260, 2006.

[18] L. Lu and J. M. Maciejowski. Self-triggered MPC with performance guarantee using relaxed dynamic programming. *Automatica*, 114:108803, 2020.

[19] S. Lucia, M. Kögel, P. Zometa, D. E. Quevedo, and R. Findeisen. Predictive control, embedded cyberphysical systems and systems of systems–a perspective. *Annual Reviews in Control*, 41:193–207, 2016.

[20] J. M. Maciejowski. *Predictive Control: with Constraints.* Pearson education, 2002.

[21] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[22] B. Mettler. *Identification modeling and characteristics of miniature rotorcraft.* Springer Science & Business Media, 2013.

[23] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson. Wireless network design for control systems: A survey. *IEEE Communications Surveys & Tutorials*, 20(2):978–1013, 2017.

[24] D. Simon. *Fighter Aircraft Maneuver Limiting Using MPC: Theory and Application*, volume 1881. Linköping University Electronic Press, 2017.

[25] K. Worthmann, M. W. Mehrez, M. Zanon, G. K. Mann, R. G. Gosine, and M. Diehl. Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs. *IEEE Transactions on Control Systems Technology*, 24(4):1394–1406, 2015.

[26] K. Zhang, C. Liu, and Y. Shi. Self-triggered adaptive model predictive control of constrained nonlinear systems: A Min-Max approach. *arXiv preprint arXiv:1912.06978*.