

# Robust to Early Termination Model Predictive Control

Mehdi Hosseinzadeh, *Member, IEEE*, Bruno Sinopoli, *Fellow, IEEE*,  
Ilya Kolmanovsky, *Fellow, IEEE*, and Sanjoy Baruah, *Fellow, IEEE*

**Abstract**—Model Predictive Control (MPC) is a popular control approach to ensure constraint satisfaction, while minimizing a cost function. Although MPC usually leads to very good results in terms of performance, its computational overhead is typically non-negligible, and its implementation for systems where the computing capacity is limited may be impossible. To address this issue, this technical note proposes a robust to early termination MPC. That is, the proposed scheme runs until available time for execution runs out, and the solution, while sub-optimal, is guaranteed to enforce the constraints and ensure recursive feasibility despite arbitrary early termination. Also, the closed-loop stability is maintained. Simulations are carried out on a F-16 aircraft to assess the effectiveness of the proposed scheme.

**Index Terms**—Model predictive control, limited computing capacity, early termination, barrier function, primal-dual flow.

## I. INTRODUCTION

THE systematic design of control laws for systems subject to constraints is one of the major challenges in controlling real-world systems. Currently, constrained control is dominated by optimization-based techniques; more specifically, Model Predictive Control (MPC) [1], [2] which is able to optimize the control performance while satisfying system's constraints. However, the use of online optimization required by MPC may require computational capabilities which may be unrealistic for certain problems, e.g., those characterized by fast dynamics and controlled by inexpensive microcontrollers.

One possible way to reduce online computation of MPC is to pre-compute optimal laws offline and store them in memory for future use. This approach is adopted in explicit MPC [3]–[5]; however, it requires high memory usage for larger state dimensional problems. Moreover, explicit MPC is not robust to early termination of the searching process.

Making use of triggering mechanisms is another way to address limited computing capacity. This idea has been exploited in self-triggered MPC [6]–[9] and event-triggered MPC [10]–[12]. However, in the presence of limited computational power,

there is no guarantee that online optimization converges when the triggering mechanism invokes MPC.

Another approach to address limited computational power in MPC implementation is to perform a fixed number of iterations [13]–[15]. The main issue is that there is no guarantee that required iterations can be carried out in the available time.

Anytime MPC has been presented in [16]–[18], where a relaxed logarithmic barrier function has been introduced to convert the MPC problem into an unconstrained optimization problem. Although anytime MPC ensures closed-loop stability even with only one iteration at every time instant, in general and unlike our approach, it does not guarantee safety.

The most recent approach to address limited computational power for MPC implementation is dynamically embedded MPC [19], in which the processor runs a virtual dynamical system whose trajectory converges to the optimal solution of the MPC problem. However, guaranteeing recursive feasibility with this approach is challenging. To address this issue, [20] augments the dynamically embedded MPC with an Explicit Reference Governor [21], [22]. However, this approach usually leads to slow response due to conservatism of ERG.

This technical note proposes REAP (Robust to EARly termination model Predictive control), which embeds the optimal solution of the MPC problem into the internal states of a virtual continuous-time dynamical system to be run in parallel with the process. The term robust indicates that the trajectory of the virtual system converges to the optimal solution, and provides a sub-optimal but feasible solution whenever its evolution is terminated. Also, the closed-loop stability is maintained. REAP allows us to implement MPC with a small sampling period (and consequently with a minimum performance degradation), while ensuring constraints satisfaction at all times.

The key contributions of this technical note are: i) developing REAP; ii) proving its properties; and iii) assessing its effectiveness for longitudinal control of an F-16 aircraft.

## II. PROBLEM STATEMENT

Consider the following discrete-time LTI system:

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t), \quad (1)$$

where  $x(t) = [x_1(t) \cdots x_n(t)]^\top \in \mathbb{R}^n$  is the state vector at time instant  $t$ ,  $u(t) = [u_1(t) \cdots u_p(t)]^\top \in \mathbb{R}^p$  is the control input at time instant  $t$ ,  $y(t) = [y_1(t) \cdots y_m(t)]^\top \in \mathbb{R}^m$  is the output at time instant  $t$ , and  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times p}$ ,  $C \in \mathbb{R}^{m \times n}$ , and  $D \in \mathbb{R}^{m \times p}$  are system matrices.

**Assumption 1:** The pair  $(A, B)$  is controllable.

This research has been supported by National Science Foundation under award numbers ECCS-1931738, ECCS-1932530, and CMMI-1904394.

M. Hosseinzadeh is with the School of Mechanical and Materials Engineering, Washington State University, Pullman, WA 99164, USA (email: mehdi.hosseinzadeh@wsu.edu).

B. Sinopoli is with the Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA (email: bsinopoli@wustl.edu).

I. Kolmanovsky is with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA (email: ilya@umich.edu).

S. Baruah is with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA (email: baruah@wustl.edu).

System (1) is subject to the following constraints:

$$x(t) \in \mathcal{X}, u(t) \in \mathcal{U}, \forall t \in \mathbb{Z}_{\geq 0}, \quad (2)$$

where  $\mathcal{X}$  and  $\mathcal{U}$  are convex polytopes:

$$\mathcal{X} = \{x \in \mathbb{R}^n : a_i^\top x + b_i \leq 0, i = 1, \dots, c_x\}, \quad (3a)$$

$$\mathcal{U} = \{u \in \mathbb{R}^p : c_i^\top u + d_i \leq 0, i = 1, \dots, c_u\}, \quad (3b)$$

with  $a_i \in \mathbb{R}^n$ ,  $b_i \in \mathbb{R}$ ,  $c_i \in \mathbb{R}^p$ ,  $d_i \in \mathbb{R}$ ,  $c_x$  as the number of state constraints, and  $c_u$  as the number of input constraints. We use  $\bar{c} := c_x + c_u$  to indicate the total number of constraints.

Let  $r \in \mathbb{R}^m$  be the desired reference. Suppose that its corresponding steady-state configuration  $(\bar{x}_r, \bar{u}_r)$  satisfying

$$\bar{x}_r = A\bar{x}_r + B\bar{u}_r, \quad r = C\bar{x}_r + D\bar{u}_r, \quad (4)$$

is such that  $\bar{x}_r \in \text{Int}(\mathcal{X})$  and  $\bar{u}_r \in \text{Int}(\mathcal{U})$ . Such a reference signal is called steady-state admissible reference; we denote the set of all steady-state admissible references by  $\mathcal{R}$ .

This technical note addresses the following problem.

**Problem 1:** Consider system (1), and suppose that the time available for control input computations is limited and unknown. Given  $r \in \mathcal{R}$ , develop a control scheme that drives the system's output to  $r$  without violating constraints (2).

### III. CONVENTIONAL MPC FOR TRACKING PROBLEMS

Given  $r \in \mathcal{R}$  as the desired reference and  $N \in \mathbb{Z}_{>0}$  as the prediction horizon size, the typical MPC approach for a tracking problem is to solve the following optimization problem at any  $t$  to compute the optimal control sequence  $\mathbf{u}^*(t) := [(u^*(0|t))^\top \dots (u^*(N-1|t))^\top]^\top \in \mathbb{R}^{Np}$ :

$$\begin{aligned} \min_{\mathbf{u}} \quad & \sum_{k=0}^{N-1} \|\hat{x}(k|t) - \bar{x}_r\|_{Q_x}^2 + \sum_{k=0}^{N-1} \|u(k|t) - \bar{u}_r\|_{Q_u}^2 \\ & + \|\hat{x}(N|t) - \bar{x}_r\|_{Q_N}^2, \end{aligned} \quad (5a)$$

subject to

$$\hat{x}(k+1|t) = A\hat{x}(k|t) + Bu(k), \quad \hat{x}(0|t) = x(t), \quad (5b)$$

$$\hat{x}(k|t) \in \mathcal{X}, \quad k \in \{0, \dots, N-1\}, \quad (5c)$$

$$u(k|t) \in \mathcal{U}, \quad k \in \{0, \dots, N-1\}, \quad (5d)$$

$$(\hat{x}(N|t), r) \in \Omega, \quad (5e)$$

where  $k$  indicates the time instant along the prediction horizon,  $Q_x = Q_x^\top \succeq 0$  ( $Q_x \in \mathbb{R}^{n \times n}$ ),  $Q_u = Q_u^\top \succ 0$  ( $Q_u \in \mathbb{R}^{p \times p}$ ), and  $Q_N \in \mathbb{R}^{n \times n}$ . Given the terminal control law  $u(k) = \kappa(\hat{x}(k|t), r)$  with  $\kappa : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$ , the terminal constraint set  $\Omega \subset \mathbb{R}^n \times \mathbb{R}^m$  is such that if  $(\hat{x}(N|t), r) \in \Omega$ , we have  $(\hat{x}(k|t), \kappa(\hat{x}(k|t), r)) \in \mathcal{X} \times \mathcal{U}$ ,  $\forall k \geq N$ . In general, determining  $\Omega$  is challenging [23]. However, as mentioned in [20], [24], having the terminal control law  $\kappa(\cdot)$  fixed, it is possible to obtain a subset of  $\Omega$ , denoted by  $\bar{\Omega}$ , by resorting to the level sets of the associated Lyapunov function, which will be discussed next. Note that although approximating the set  $\Omega$  may lead to a conservative solution, such approximation is adequate and effective in many cases [24]. Furthermore, Remark 2 will provide an approach to enlarge such a subset.

#### A. Obtaining a Subset of the Terminal Constraint Set $\Omega$

In what follows, we define the terminal set in (5e) so that it is invariant under the terminal control law  $\kappa(x(t), r) = \bar{u}_r + K(x(t) - \bar{x}_r)$ , where  $K$  is such  $A + BK$  is Schur, and is constraint admissible.

Defining the tracking error as  $e(t) := x(t) - \bar{x}_r$ , it is easy to show that if constraints are inactive, then  $e(t+1) = (A + BK)e(t)$ , which implies  $e(t) \rightarrow 0$  as  $t \rightarrow \infty$ . Thus, there exist a Lyapunov function for the closed-loop system with the terminal controller in the following form

$$V(x(t), r) = (x(t) - \bar{x}_r)^\top \Psi (x(t) - \bar{x}_r), \quad (6)$$

with  $\Psi = \Psi^\top \succ 0$  ( $\Psi \in \mathbb{R}^{n \times n}$ ) satisfying  $(A + BK)^\top \Psi (A + BK) - \Psi \prec 0$ , that proves the stability of  $\bar{x}_r$ .

**Remark 1:** Assumption 1 guarantees the existence of a stabilizing feedback gain matrix  $K$ . It is convenient to select  $Q_N$  in (5a) as the solution of the algebraic Riccati equation  $Q_N = A^\top Q_N A - (A^\top Q_N B)(Q_u + B^\top Q_N B)^{-1}(B^\top Q_N A) + Q_x$ , and the feedback gain matrix  $K$  in the terminal control law as  $K = -(Q_u + B^\top Q_N B)^{-1}(B^\top Q_N A)$ ; such a selection is optimal for the unconstrained problem.

By taking into account the terminal control law, the set of input constraints given in (3b) can be rewritten as:

$$\mathcal{U} = \{x \in \mathbb{R}^n : c_i^\top Kx + \bar{d}_i \leq 0, i = 1, \dots, c_u\}, \quad (7)$$

where  $\bar{d}_i = d_i + c_i^\top K\bar{u}_r - c_i^\top K\bar{x}_r$ . Thus, by considering the terminal control law, constraints (3) can be expressed as a single constraint set on the states of the system, as follows:

$$\Lambda = \{x \in \mathbb{R}^n : \eta_i^\top x + \gamma_i \leq 0, i = 1, \dots, \bar{c}\} \quad (8)$$

where  $\eta_i \in \mathbb{R}^n$  and  $\gamma_i \in \mathbb{R}$ .

Constraints (8) can be mapped into a convex constraint on the Lyapunov function [25], [26], so that  $\bar{\Omega}$  can be defined as:

$$\bar{\Omega} = \{(x, r) : V(x, r) - \Gamma_i(r) \leq 0, i = 1, \dots, \bar{c}\} \subseteq \Omega, \quad (9)$$

where  $\Gamma_i$  is the largest Lyapunov level set that does not violate  $i$ -th state constraint given in (8) computed as follows:

$$\Gamma_i(r) = (\eta_i^\top \bar{x}_r + \gamma_i)^2 / (\eta_i^\top \Psi^{-1} \eta_i). \quad (10)$$

**Remark 2:** As shown in [20], [27], it is convenient to select  $\Psi$  in (6) individually for each constraint such that the resulting Lyapunov level sets are as large as possible.

### IV. PROPOSED SOLUTION—REAP

Common approaches to solve optimization problem (5) is to use interior-point methods. Though these methods are fast and efficient, in general, iterates are not necessarily feasible [28], [29]. Thus, there is no guarantee that early termination (due to limited computing capacity) yields a feasible point. This section addresses this practical challenge.

### A. Continuous-Time Dynamical System

We proceed by first tightening constraint sets as follows:

$$\mathcal{X}_\beta = \{x : a_i^\top x + b_i + 1/\beta \leq 0, \ i = 1, \dots, c_x\}, \quad (11a)$$

$$\mathcal{U}_\beta = \{u : c_i^\top u + d_i + 1/\beta \leq 0, \ i = 1, \dots, c_u\}, \quad (11b)$$

$$\bar{\Omega}_\beta = \{(x, r) : V(x, r) - \Gamma_i(r) + 1/\beta \leq 0, \ i = 1, \dots, \bar{c}\}, \quad (11c)$$

where  $\beta > 0$  is sufficiently large to make sure that the feasible set of optimization problem (5) with the tightened sets remains nonempty. We denote the solution of (5) with the tightened sets by  $\mathbf{u}^\dagger(t)$ . It is evident that  $\lim_{\beta \rightarrow \infty} \mathbf{u}^\dagger(t) = \mathbf{u}^*(t)$ .

*Remark 3:* Constraint tightening has been widely used in distributed MPC (e.g., [30]–[34]) to ensure feasibility and sub-optimality even after only a finite number of iterations. The main issue with these schemes is that there is no guarantee that required iterations can be carried out in the available time.

The modified<sup>1</sup> barrier function  $\mathcal{B}(x(t), r, \mathbf{u}, \lambda)$  associated with optimization problem (5) with the tightened constraint sets (11) is as in (12), where  $\lambda = [\lambda_{1,0}^x \ \dots \ \lambda_{c_x,(N-1)}^x \ \lambda_{1,0}^u \ \dots \ \lambda_{c_u,(N-1)}^u \ \lambda_1^{\bar{\Omega}} \ \dots \ \lambda_{\bar{c}}^{\bar{\Omega}}]^\top \in \mathbb{R}_{\geq 0}^{(N+1)\bar{c}}$  is the vector of dual parameters. At any time instant  $t$ , we denote the vector of optimal dual parameters by  $\lambda^\dagger(t)$ .

Now, we propose the following primal-dual gradient flow:

$$\frac{d}{ds} \hat{\mathbf{u}}(s) = -\sigma \nabla_{\hat{\mathbf{u}}} \mathcal{B}(x(t), r, \hat{\mathbf{u}}(s), \hat{\lambda}(s)), \quad (13a)$$

$$\frac{d}{ds} \hat{\lambda}(s) = +\sigma \left( \nabla_{\hat{\lambda}} \mathcal{B}(x(t), r, \hat{\mathbf{u}}(s), \hat{\lambda}(s)) + \Phi(s) \right), \quad (13b)$$

where  $\sigma \in \mathbb{R}_{>0}$  is a tuning parameter that determines the rate of the evolution of the system (13),  $s$  is the auxiliary time variable that REAP spends on solving the MPC problem, and  $\Phi(s) \in \mathbb{R}^{(N+1)\bar{c}}$  is the projection operator onto the normal cone of  $\lambda$  [36] whose  $i$ -th entry (denoted by  $[\Phi(s)]_i$ ) is<sup>2</sup>

$$[\Phi(s)]_i = \begin{cases} 0, & \text{if } ([\hat{\lambda}(s)]_i > 0) \text{ or } ([\hat{\lambda}(s)]_i = 0 \\ & \text{and } [\nabla_{\hat{\lambda}} \mathcal{B}(\cdot)]_i \geq 0) \\ -[\nabla_{\hat{\lambda}} \mathcal{B}(\cdot)]_i, & \text{otherwise} \end{cases} \quad (14)$$

with  $[\hat{\lambda}(s)]_i$  and  $[\nabla_{\hat{\lambda}} \mathcal{B}(\cdot)]_i$  as the  $i$ -th entry of  $\hat{\lambda}(s)$  and  $\mathcal{B}(x(t), r, \hat{\mathbf{u}}(s), \hat{\lambda}(s))$ , respectively. The differential equations (13) build a virtual continuous-time dynamical system (see Subsection IV-C for the definition of its initial conditions) which should be run at any time instant  $t$ . In the following subsection, we will discuss properties of system (13).

<sup>1</sup>Note that modified barrier functions do not grow to infinity when the current approximation approaches the solution [35], which allows converging to the optimal solution  $\mathbf{u}^\dagger(t)$ .

<sup>2</sup>For the sake of brevity, we denote  $\mathcal{B}(x(t), r, \hat{\mathbf{u}}(s), \hat{\lambda}(s))$  by  $\mathcal{B}(\cdot)$ .

*Remark 4:* Given  $\Upsilon(s) = \begin{bmatrix} (\hat{\mathbf{u}}(s))^\top & (\hat{\lambda}(s))^\top \end{bmatrix}^\top$ , since the Hessian matrix  $\nabla_{\Upsilon\Upsilon} \mathcal{B}(\cdot)$  is not full rank, we cannot use methods requiring the inverse of Hessian matrix, e.g., [37], [38]. Also, methods requiring an approximation of  $(\nabla_{\Upsilon\Upsilon} \mathcal{B}(\cdot))^{-1}$ , like quasi-Newton method, have a slower convergence [39].

### B. Properties

In this subsection, we prove convergence and constraint-handling properties of system (13). Theorem 1 shows that the trajectory of system (13) converges to the optimal solution  $(\mathbf{u}^\dagger(t), \lambda^\dagger(t))$ , and Theorem 2 proves that they satisfies constraints (5c)–(5e) at all times. See Appendix for proofs.

*Theorem 1:* Let  $(\hat{\mathbf{u}}(s), \hat{\lambda}(s))$  be the trajectory of (13). Given a feasible initial condition  $(\hat{\mathbf{u}}(0), \hat{\lambda}(0))$ ,  $(\hat{\mathbf{u}}(s), \hat{\lambda}(s))$  exponentially converges to  $(\mathbf{u}^\dagger(t), \lambda^\dagger(t))$  as  $s \rightarrow \infty$ .

*Theorem 2:* Let  $(\hat{\mathbf{u}}(s), \hat{\lambda}(s))$  be the solution of (13). Given a feasible initial condition  $(\hat{\mathbf{u}}(0), \hat{\lambda}(0))$ ,  $\hat{\mathbf{u}}(s)$  satisfies constraints (5c)–(5e) for all  $s$ .

*Remark 5:* By virtue of imposing the terminal constraint set as in (5e), Theorem (2) ensures that recursive feasibility is preserved irrespective of early termination.

*Remark 6:* In this paper, REAP is introduced as a continuous-time scheme. This facilitates the analysis and the derivation of its theoretical properties. This is not dissimilar to how control schemes are derived and analyzed. For instance, a similar strategy reliant on optimization algorithms in continuous time is employed in the recent book [40]. For discrete-time implementation of system (13), one can use the difference quotient with a sufficiently small sampling period. Our numerical experiments suggest that a discrete-time implementation maintains the desired properties of our algorithm. Furthermore, if the current step of the approximating discrete-time system jumps over the constraints due to discretization, REAP will stop and will use the solution from the previous step which is feasible. According to the employed acceptance/rejection mechanism (see Subsection IV-D), this approach will not hamper the convergence or feasibility properties. We leave the study of theoretical guarantees for the discrete-time implementation to future work.

### C. Warm-starting

The warm-starting strategy is used to define  $\hat{\mathbf{u}}(0)$  at time instant  $t$  based on the previously computed control input sequence augmented with the terminal control law for the new prediction time, i.e.

$$\hat{\mathbf{u}}(0) = \begin{bmatrix} \mathbf{u}(t-1)_{p+1:Np} & (\bar{\mathbf{u}}_r + K(\hat{x}(N|t-1) - \bar{x}_r))^\top \end{bmatrix}^\top, \quad (15)$$

$$\begin{aligned} \mathcal{B}(\cdot) = & \sum_{k=0}^{N-1} \|\hat{x}(k|t) - \bar{x}_r\|_{Q_x}^2 + \sum_{k=0}^{N-1} \|\mathbf{u}(k|t) - \bar{\mathbf{u}}_r\|_{Q_u}^2 + \|\hat{x}(N|t) - \bar{x}_r\|_{Q_N}^2 - \sum_{i=1}^{c_x} \sum_{k=0}^{N-1} \lambda_{i,k}^x \log(-\beta(a_i^\top \hat{x}(k|t) + b_i + 1/\beta) + 1) \\ & - \sum_{i=1}^{c_u} \sum_{k=0}^{N-1} \lambda_{i,k}^u \log(-\beta(c_i^\top \mathbf{u}(k|t) + d_i + 1/\beta) + 1) - \sum_{i=1}^{\bar{c}} \lambda_i^{\bar{\Omega}} \log(-\beta(V(\hat{x}(N|t), r) - \Gamma_i(r) + 1/\beta) + 1). \end{aligned} \quad (12)$$

where  $\mathbf{u}(t-1)_{p+1:Np} \in \mathbb{R}^{(N-1)p}$  is<sup>3</sup> the entries  $p+1$  to  $Np$  of  $\mathbf{u}(t-1)$  (which is the control input computed at time instant  $t-1$ ) and  $\bar{u}_r + K(\hat{x}(N|t-1) - \bar{x}_r)$  is the terminal control law for the new prediction time. This selection is reasonable, as in most applications, from one time instant to the next, states of the system do not change substantially.

Regarding the initial condition for dual parameters, note that  $\lambda_{i,k}^x(t-1) \neq 0$  ( $\lambda_{i,k}^u(t-1) \neq 0$ ;  $\lambda_i^\Omega \neq 0$ ) implies that the  $i$ -th constraint on the states at prediction time  $k$  ( $i$ -th constraint on the control input at prediction time  $k$ ;  $i$ -th terminal constraint) was active at time instant  $t-1$ . This condition moves one step backward at time instant  $t$ . The same condition holds for inactive constraints. Thus, there is a one-step time shift in the active and inactive constraints. Hence, we propose the following initial condition for  $\hat{\lambda}(s)$  at time instant  $t$ :

$$\hat{\lambda}(0) = \begin{bmatrix} \lambda_{1,1}^x(t-1) & \cdots & \lambda_{c_x,(N-1)}^x(t-1) & \lambda_{c_x,(N-1)}^x(t-1) \\ \lambda_{1,1}^u(t-1) & \cdots & \lambda_{c_u,(N-1)}^u(t-1) & \lambda_{c_u,(N-1)}^u(t-1) \\ \lambda_1^\Omega(t-1) & \cdots & \lambda_{\bar{c}}^\Omega(t-1) \end{bmatrix}^\top, \quad (16)$$

where  $\lambda_{c_x,(N-1)}^x(t-1) \in \mathbb{R}_{\geq 0}$  and  $\lambda_{c_u,(N-1)}^u(t-1) \in \mathbb{R}_{\geq 0}$  are used as initial guesses for the value of dual parameters at the new prediction time, and  $\lambda_i^\Omega(t-1) \in \mathbb{R}_{\geq 0}$ ,  $i \in \{1, \dots, \bar{c}\}$  are used as initial guesses for the values of dual parameters at the prediction time  $N$  (terminal constraints). Note that any non-negative values for dual parameters are feasible; however, our numerical experiments show that such an ad hoc strategy for picking the last elements of (16) is very effective.

*Remark 7:* At the time instant  $t = 0$ , the dual parameter  $\hat{\lambda}(0)$  can be set to any vector whose elements are all positive, and  $\hat{\mathbf{u}}(0)$  can be set to any feasible sequence. Note that since the MPC given in (5) typically responds to desired reference changes, a feasible control sequence at  $t = 0$  can always be determined by using the feasibility governor described in [20].

#### D. Acceptance/Rejection Mechanism

Theorem 1 showed that at any time instant  $t$ ,  $(\hat{\mathbf{u}}(s), \hat{\lambda}(s)) \rightarrow (\mathbf{u}^\dagger(t), \lambda^\dagger(t))$  exponentially fast as  $s \rightarrow \infty$ . However, since the evolution of system (13) might be terminated before convergence due to limited computing capacity, and since, in general, the behavior of  $\hat{\mathbf{u}}(s) - \mathbf{u}^\dagger(t)$  is not monotonic, there is a need for a logic-based method to accept or reject  $\hat{\mathbf{u}}(s)$  once the evolution of system (13) is terminated.

Drawing inspiration from [41], we develop a simple yet efficient acceptance/rejection mechanism. As discussed in Subsection IV-C,  $\hat{\mathbf{u}}(0)$  as in (15) is a feasible and a sub-optimal solution for the optimization problem (5) with the tightened constraint sets. Thus, given the termination time  $s_t$ , we accept  $\hat{\mathbf{u}}(s_t)$  (i.e.,  $\mathbf{u}(t) = \hat{\mathbf{u}}(s_t)$  and  $\lambda(t) = \hat{\lambda}(s_t)$ ) if the value of the cost function given in (5a) with  $\hat{\mathbf{u}}(s_t)$  is smaller than that with  $\hat{\mathbf{u}}(0)$ , and reject (i.e.,  $\mathbf{u}(t) = \hat{\mathbf{u}}(0)$  and  $\lambda(t) = \hat{\lambda}(0)$ ) otherwise. It is obvious that the above-mentioned mechanism does not discard the optimal solution if system (13) converges.

<sup>3</sup>Due to limited computing capacity, the computed control input at time instant  $t$  is not necessarily the optimum (i.e.,  $\mathbf{u}^\dagger(t)$ ). For this reason, we drop the  $\dagger$  when referring to the computed control input at time instant  $t$ . We do the same when referring to dual parameter.

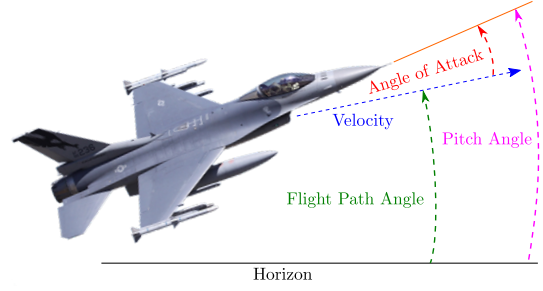


Fig. 1. Longitudinal control of a F-16 aircraft.

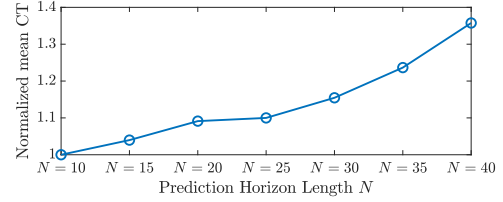


Fig. 2. Impact of the prediction horizon length on the mean computing time of the conventional MPC.

## V. SIMULATION STUDY

The objective of this section is to validate REAP and assess its effectiveness. The simulations are run on a computer with Intel(R) Core(TM) i7-7500U CPU 2.70 GHz processor and 16.00 GB of RAM. We use YALMIP toolbox [42] to implement the computations of the conventional MPC scheme. For comparison purposes, we define the Performance Index as  $PI \triangleq \int (\|x(\tau) - \bar{x}_r\|_{Q_x}^2 + \|u(\tau) - \bar{u}_r\|_{Q_u}^2) d\tau$ , where the integration is performed over the duration of the simulations.

As shown in [43], the longitudinal dynamical model of an F-16 aircraft (see Fig. 1) around the origin can be expressed by the continuous-time LTI model  $\dot{x} = A_c x + B_c u$ , where  $x = [\theta \ q \ \alpha \ \delta_e \ \delta_f]^\top$  with  $\theta$  as the pitch angle,  $q$  as the pitch rate,  $\alpha$  as the angle of attack,  $\delta_e$  as the elevator deflection, and  $\delta_f$  as the flaperon deflection, and  $u = [u_e \ u_f]^\top$  with  $u_e$  as the elevator deflection command and  $u_f$  as the flaperon deflection command. We use the matrices  $A_c$  and  $B_c$  given in [20]. The desired pitch angle and angle of attack are 9 and 6 [deg], respectively. The system is required to satisfy the state constraints  $|q| \leq 4$  [deg/s] and  $\alpha \leq 6.5$  [deg]. The weight matrices in the cost function (5a) are  $Q_x = \text{diag}(1, 0.1, 1, 0.1, 0.1)$  and  $Q_u = \text{diag}(0.1, 0.1)$ . For the model discretized with sampling period of 100 msec, Fig. 2 reports normalized mean Computing Time (CT) of the conventional MPC from 2000 runs, where the mean CT for  $N = 10$  is used as the basis for normalization. As expected, as we look further into the future, the dimension of the problem becomes larger, which results in a larger CT.

#### A. System Performance—Comparing With Conventional MPC

Assuming that the prediction horizon size is  $N = 20$ , the worst-case execution time of the conventional MPC from 2000 runs is  $\sim 380$  ms. We consider the following three cases: i) there is no computational limitation and we can implement

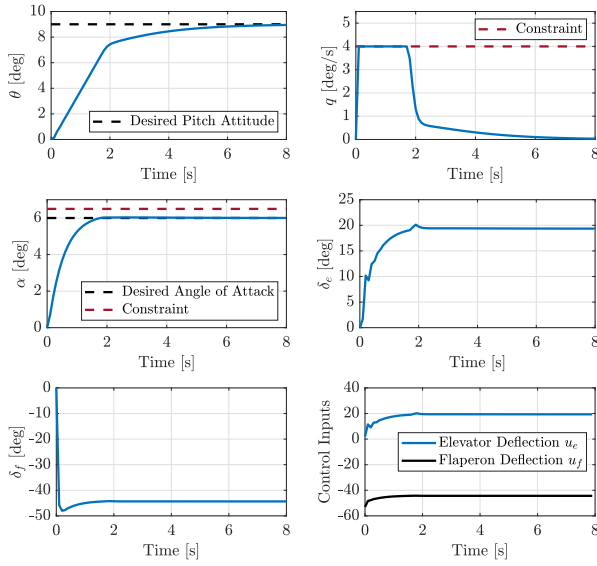


Fig. 3. Simulation results for longitudinal control with REAP.

TABLE I  
SYSTEM PERFORMANCE ANALYSIS

|               | MPC (100 ms) | MPC (400 ms) | REAP |
|---------------|--------------|--------------|------|
| Normalized PI | 1            | 1.52         | 1.08 |

the conventional MPC every 100 ms; ii) to address limited computing capacity, we implement the conventional MPC every 400 ms; and iii) we implement REAP every 100 ms with  $\sigma = 1$  and  $\beta = 10^5$ . Note that to implement REAP, we use the optimal control input computed offline at time zero.

For space limitations, we only show simulation results for REAP in Fig. 3. Normalized achieved PIs for all cases are reported in TABLE I, where the achieved PI for the conventional MPC with the sampling period 100 ms is used as the basis for normalization. As seen in this table, implementing MPC with a large sampling period (i.e., the conventional MPC with the sampling period 400 ms) degrades the performance. However, REAP yields better performance by computing a sub-optimal solution every 100 ms.

#### B. Sensitivity Analysis—Impact of $N$

We conducted sensitivity analysis of the performance of REAP with respect to the prediction horizon length  $N$ . Fig. 4 shows how the prediction horizon length  $N$  impacts the performance of REAP in comparison with the conventional MPC. In this figure, for each  $N$ , the conventional MPC is implemented with the given sampling period  $\Delta$  which is determined based on the worst-case execution time from 2000 runs (note that the worst-case execution time increases with the prediction horizon). In Fig. 4, the normalized PI means the ratio of the REAP's PI implemented every 100 ms with the given  $N$  to the PI of the conventional MPC implemented every given  $\Delta$  ms with the given  $N$ . As seen in Fig. 4, as  $N$  increases, the performance of REAP in comparison with the conventional MPC improves. This is consistent with our expectations; while the performance of REAP remains

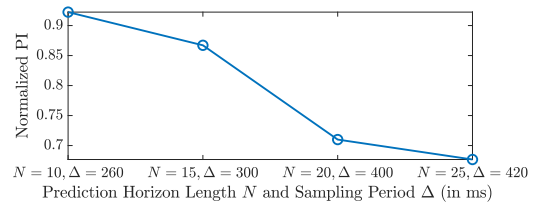


Fig. 4. Impact of the prediction horizon length on the performance of REAP against the conventional MPC.

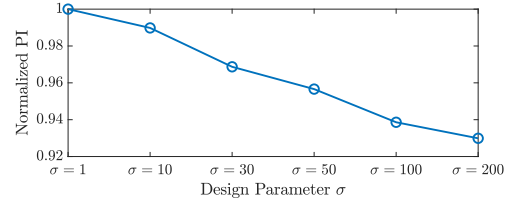


Fig. 5. Impact of the design parameter  $\sigma$  on the performance of REAP.

almost constant for different values of  $N$  (as it is implemented every 100 ms), the performance of the conventional MPC deteriorates with the longer sampling period  $\Delta$ .

#### C. Sensitivity Analysis—Impact of $\sigma$

In this subsection, we study the sensitivity of the REAP's performance to the design parameter  $\sigma$ . Fig. 5 reports the normalized PI to show how the design parameter  $\sigma$  impacts the system performance, where the system performance for  $\sigma = 1$  is used as the basis for normalization. From Fig. 5, we observe that as  $\sigma$  increases, the REAP's performance improves. This is consistent with our expectations from Theorem 2 (to be more precise, from inequality A.4). It should be noted that to maintain REAP's properties, the larger the design parameter  $\sigma$  is, the smaller the sampling period should be selected for discrete-time implementation of REAP (see Remark 6).

#### D. System Performance—Comparing With [44]

In [44, Subsection IV-A], the authors propose a novel MPC scheme where a control law is constructed to control a constrained system to a desired set. Similar to REAP, the method in [44], ensures constraint satisfaction at all times. However, method [44] appears to give worse performance in comparison with REAP, as [44] employs a fixed control law, even though its parameters are determined by solving an optimization problem. This point is shown in Fig. 6 which presents simulations results for REAP (implemented with  $\sigma = 100$  and  $\beta = 10^5$ ) and method [44]. From this figure, REAP performs  $\sim 38\%$  better in comparison with method [44] in terms of tracking performance.

#### E. System Performance—Comparing With [16]

In this section, we compare our method with the anytime MPC presented in [16]. Note that even though both methods maintain closed-loop stability if computations are terminated early, method [16] may lead to constraint violation, while REAP guarantees constraint satisfaction at all times.



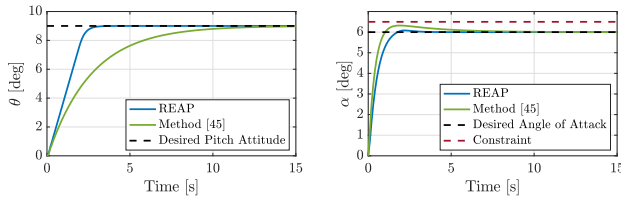


Fig. 6. Simulation results with REAP and the method in [44].

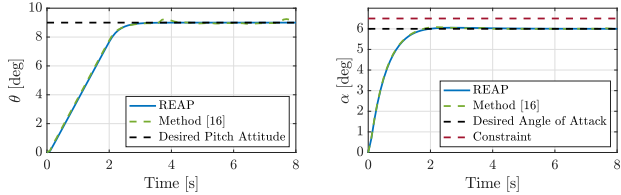


Fig. 7. Simulation results with REAP and the method in [16].

To implement the method [16], we used binary variables to convert the logical conditions of the barrier functions, and used bnb solver to solve the corresponding optimization problems. Also, we set the relaxation parameter and the barrier function weighting parameter to  $10^{-3}$ , and we implemented REAP with  $\sigma = 100$  and  $\beta = 10^5$ . Simulation results when there is a sufficient computing resource and no early termination occurs for the method [16] are shown in Fig. 7. As seen in this figure, when the computing capacity is not limited, REAP and method [16] yield comparable tracking performance.

To compare constraint satisfaction property of both methods, we assume that the available time (in ms) for executing the methods at each time instant is given by  $100 - \chi$ , where  $\chi$  is selected randomly from a Weibull distribution<sup>4</sup> with shape parameter 2, location parameter 20, and scale parameter 4. Our results from 2000 runs show that method [16] leads to constraint violation in  $\sim 13\%$  of the runs, while REAP always ensures constraint satisfaction.

#### F. System Performance—Comparing With [47]

In this section, we compare our method with the robust to early termination command governor presented in [47], which is used to augment a nominal state feedback controller. We assume that the available time (in ms) for executing the methods at each time instant is given by  $100 - \chi$ , where  $\chi$  is selected randomly from a Weibull distribution with shape parameter 2, location parameter 20, and scale parameter 4. Simulation results are shown in Fig. 8. From this figure, although both methods ensure feasibility at all times, REAP (implemented with  $\sigma = 100$  and  $\beta = 10^5$ ) improves performance by  $\sim 16\%$  in comparison with the method [47].

### VI. CONCLUSION

This technical note proposed a robust-to-early termination MPC (REAP) which is an algorithm capable of maintaining feasibility of MPC in the presence of limited and possibly

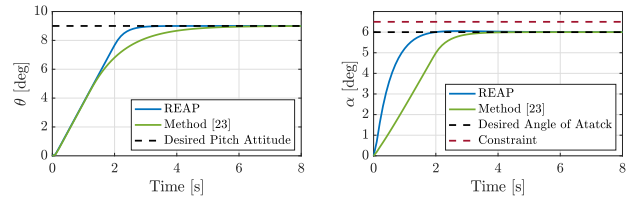


Fig. 8. Simulation results with REAP and the method in [47].

time-varying computing resources. REAP is developed based on the barrier function and continuous-time primal-dual gradient flow method. It was shown that REAP converges to the optimal solution of the MPC problem, and provides a sub-optimal but feasible and effective solution if early terminated. The effectiveness of REAP was validated through simulation studies. This technical note showed that REAP can effectively address the issue of limited computational resources, and can satisfy control objectives without violating constraints.

### REFERENCES

- [1] E. F. Camacho and C. B. Alba, *Model Predictive Control*, 2nd ed. Springer-Verlag London, 2007.
- [2] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Nob Hill Publishing, LLC, 2017.
- [3] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear model predictive control: Towards new challenging applications*, L. Magni, D. M. Raimondo, and F. Allgower, Eds. Springer Berlin Heidelberg, 2009, pp. 345–369.
- [4] M. Kvasnica, P. Bakarac, and M. Klauco, "Complexity reduction in explicit MPC: A reachability approach," *Systems & Control Letters*, vol. 124, pp. 19–26, Feb. 2019.
- [5] X. Xiu and J. Zhang, "Grid k-d tree approach for point location in polyhedral data sets—application to explicit MPC," *International Journal of Control*, vol. 93, 2020.
- [6] E. Henriksson, D. E. Quevedo, H. Sandberg, and K. H. Johansson, "Self-triggered model predictive control for network scheduling and control," in *Proc. 8th IFAC Symposium on Advanced Control of Chemical Processes*, Singapore, Jul. 10–13, 2012, pp. 432–438.
- [7] Z. Sun, L. Dai, K. Liu, D. V. Dimarogonas, and Y. Xia, "Robust self-triggered MPC with adaptive prediction horizon for perturbed nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4780–4787, Nov. 2019.
- [8] Q. Cao, Z. Sun, Yuanqing, Xia, Li, and Dai, "Self-triggered MPC for trajectory tracking of unicycle-type robots with external disturbance," *Journal of the Franklin Institute*, vol. 356, no. 11, pp. 5593–5610, Jul. 2019.
- [9] L. Lu and J. M. Maciejowski, "Self-triggered MPC with performance guarantee using relaxed dynamic programming," *Automatica*, vol. 114, Apr. 2020.
- [10] X. Mi, Y. Zou, and S. Li, "Event-triggered MPC design for distributed systems toward global performance," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 4, pp. 1474–1495, Mar. 2018.
- [11] J. Yoo and K. H. Johansson, "Event-triggered model predictive control with a statistical learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 4, pp. 2571–2581, Apr. 2021.
- [12] B. Wang, J. Huang, C. Wen, J. Rodriguez, C. Garcia, H. B. Gooi, and Z. Zeng, "Event-triggered model predictive control for power converters," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 1, pp. 715–720, Jan. 2021.
- [13] G. Cimini and A. Bemporad, "Exact complexity certification of active-set methods for quadratic programming," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6094–6109, Dec. 2017.
- [14] R. Ghaemi, J. Sun, and I. V. Kolmanovskiy, "An integrated perturbation analysis and sequential quadratic programming approach for model predictive control," *Automatica*, vol. 45, no. 10, pp. 2412–2418, Oct. 2009.

<sup>4</sup>The approach of using the Weibull distribution to characterize the available time for a computing task is well-accepted in the literature, e.g., [45], [46].

- [15] D. Liao-McPherson, M. M. Nicotra, and I. Kolmanovsky, "Time-distributed optimization for real-time model predictive control: Stability, robustness, and constraint satisfaction," *Automatica*, vol. 117, Jul. 2020.
- [16] C. Feller and C. Ebenbauer, "A stabilizing iteration scheme for model predictive control based on relaxed barrier functions," *Automatica*, pp. 328–339, Jun. 2017.
- [17] —, "Sparsity-exploiting anytime algorithms for model predictive control: A relaxed barrier approach," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 2, pp. 425–435, Mar. 2020.
- [18] M. Gharbi and C. Ebenbauer, "Anytime MHE-based output feedback MPC," in *Proc. 7th IFAC Conference on Nonlinear Model Predictive Control*, Bratislava, Slovakia, Jul. 11–14, 2021, pp. 264–271.
- [19] M. M. Nicotra, D. Liao-McPherson, and I. V. Kolmanovsky, "Dynamically embedded model predictive control," in *Proc. 2018 Annual American Control Conference*, Milwaukee, WI, USA, Jun. 27–29, 2018, pp. 4957–4962.
- [20] —, "Embedding constrained model predictive control in a continuous-time dynamic feedback," *IEEE Transactions on Automatic Control*, vol. 64, no. 5, pp. 1932–1946, May 2019.
- [21] M. Hosseinzadeh and E. Garone, "An explicit reference governor for the intersection of concave constraints," *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 1–11, Jan. 2020.
- [22] M. Hosseinzadeh, A. Cottruelo, D. Limon, and E. Garone, "Constrained control of linear systems subject to combinations of intersections and unions of concave constraints," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 571–576, Jul. 2019.
- [23] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, "Mpc for tracking piecewise constant references for constrained linear systems," *Automatica*, vol. 44, no. 9, pp. 2382–2387, Sep. 2008.
- [24] A. Cottruelo, M. Hosseinzadeh, D. R. Ramirez, D. Limon, and E. Garone, "Reference dependent invariant sets: Sum of squares based computation and applications in constrained control," *Automatica*, vol. 129, p. 109614, Jul. 2021.
- [25] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Birkhäuser Basel, 2008.
- [26] E. Garone, M. M. Nicotra, and L. Ntogramatzidis, "Explicit reference governor for linear systems," *International Journal of Control*, vol. 91, no. 6, pp. 1415–1430, 2018.
- [27] M. Hosseinzadeh, K. van Heusden, G. A. Dumont, and E. Garone, "An explicit reference governor scheme for closed-loop anesthesia," in *Proc. 18th European Control Conf.*, Naples, Italy, Jun. 25–28, 2019, pp. 1294–1299.
- [28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [29] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, Mar. 2010.
- [30] M. D. Doan, T. Keviczky, and B. D. Schutter, "A dual decomposition-based optimization method with guaranteed primal feasibility for hierarchical MPC problems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 392–397, Jan. 2011.
- [31] —, "A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraints," in *Proc. 50th IEEE Conf. Decision and Control*, Orlando, FL, USA, Dec. 12–15, 2011, pp. 5236–5241.
- [32] I. Necoara and V. Nedelcu, "On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems," *Automatica*, vol. 55, pp. 209–216, May 2015.
- [33] J. Köhler and M. A. Müller and F. Allgöwer, "Distributed model predictive control—recursive feasibility under inexact dual optimization," *Automatica*, vol. 102, pp. 1–9, Apr. 2019.
- [34] B. Jin, H. Li, W. Yan, and M. Cao, "Distributed model predictive control and optimization for linear systems with global constraints and time-varying communication," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 3393–3400, Jul. 2021.
- [35] R. Polyak, "Modified barrier functions (theory and methods)," *Mathematical Programming*, vol. 54, no. 1–3, pp. 177–222, Feb. 1992.
- [36] K. Ryu and S. Boyd, "Primer on monotone operator methods," *Appl. Comput. Math.*, vol. 15, no. 1, pp. 3–43, Jan. 2016.
- [37] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, "Interior point method for dynamic constrained optimization in continuous time," in *Proc. American Control Conference*, Boston, MA, USA, Jul. 6–8, 2016, pp. 5612–5618.
- [38] T. Ohtsuka, "A continuation/GMRES method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563–574, Apr. 2004.
- [39] M. Gilli, D. Maringer, and E. Schumann, *Numerical methods and optimization in finance*. Academic Press, 2019.
- [40] S. P. Meyn, *Control Systems and Reinforcement Learning*. Cambridge University Press, 2022.
- [41] E. Garone and I. Kolmanovsky, "Command governors with inexact optimization and without invariance," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 8, pp. 1–6, Aug. 2022.
- [42] J. Lofberg, "YALMIP: a toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Conf. Robotics and Automation*, Taipei, Taiwan, Sep. 2–4, 2004, pp. 284–289.
- [43] K. M. Sobel and E. Y. Shapiro, "A design methodology for pitch pointin flight control systems," *J. Guid., Control Dyn.*, vol. 8, no. 2, pp. 181–187, 1985.
- [44] A. Bemporad, D. Bernardini, and P. Patrino, "A convex feasibility approach to anytime model predictive control," *arXiv:1502.07974*, 2015. [Online]. Available: <https://arxiv.org/abs/1502.07974>
- [45] F. E. Ophelders, S. Chakraborty, and H. Corporaal, "Intra- and inter-processor hybrid performance modeling for MPSoC architectures," in *Proc. 6th IEEE/ACM/IFIP int. conf. Hardware/Software codesign and system synthesis*, Atlanta, GA, USA, Oct. 19–24, 2008, pp. 91–96.
- [46] Y. Lu, T. Nolte, I. Bate, and L. Cucu-Grosjean, "A statistical response-time analysis of real-time embedded systems," in *Proc. IEEE 33rd Real-Time Systems Symposium*, San Juan, PR, USA, Dec. 4–7, 2012, pp. 351–362.
- [47] M. Hosseinzadeh, B. Sinopoli, I. Kolmanovsky, and S. Baruah, "ROTEC: Robust to early termination command governor for systems with limited computing capacity," *Systems & Control Letters*, 2022.
- [48] A. D. Alexandrov, *Convex Polyhedra*. Springer-Verlag Berlin Heidelberg, 2005.

## APPENDIX

### A. Proof of Theorem 1

Similar to [47], it can be shown that there exists  $\mu > 0$  such that the operator  $\left[ (\nabla_{\hat{\mathbf{u}}}\mathcal{B}(\cdot))^{\top} - (\nabla_{\hat{\lambda}}\mathcal{B}(\cdot))^{\top} \right]^{\top}$  satisfies:

$$\begin{bmatrix} \nabla_{\hat{\mathbf{u}}}\mathcal{B}(\cdot) \\ -\nabla_{\hat{\lambda}}\mathcal{B}(\cdot) \end{bmatrix}^{\top} \begin{bmatrix} \hat{\mathbf{u}}(s) - \mathbf{u}^{\dagger}(t) \\ \hat{\lambda}(s) - \lambda^{\dagger}(t) \end{bmatrix} \geq \mu \left\| \begin{bmatrix} \hat{\mathbf{u}}(s) - \mathbf{u}^{\dagger}(t) \\ \hat{\lambda}(s) - \lambda^{\dagger}(t) \end{bmatrix} \right\|^2. \quad (\text{A.1})$$

Now, consider the following Lyapunov function:

$$W(\cdot) = \frac{1}{2\sigma} \|\hat{\mathbf{u}}(s) - \mathbf{u}^{\dagger}(t)\|^2 + \frac{1}{2\sigma} \|\hat{\lambda}(s) - \lambda^{\dagger}(t)\|^2, \quad (\text{A.2})$$

whose time derivative w.r.t. the auxiliary time variable  $s$ , and according to (13) and (14), satisfies the following inequality:

$$\begin{aligned} \frac{d}{ds} W(\cdot) &\leq -(\hat{\mathbf{u}}(s) - \mathbf{u}^{\dagger}(t))^{\top} \nabla_{\hat{\mathbf{u}}}\mathcal{B}(\cdot) \\ &\quad + (\hat{\lambda}(s) - \lambda^{\dagger}(t))^{\top} \nabla_{\hat{\lambda}}\mathcal{B}(\cdot). \end{aligned} \quad (\text{A.3})$$

According to (A.1), (A.3) can be expressed as  $\frac{d}{ds} W(\cdot) \leq -2\sigma\mu W(\cdot)$ , implying that

$$\left\| \begin{bmatrix} \hat{\mathbf{u}}(s) - \mathbf{u}^{\dagger}(t) \\ \hat{\lambda}(s) - \lambda^{\dagger}(t) \end{bmatrix} \right\|^2 \leq \left\| \begin{bmatrix} \hat{\mathbf{u}}(0) - \mathbf{u}^{\dagger}(t) \\ \hat{\lambda}(0) - \lambda^{\dagger}(t) \end{bmatrix} \right\|^2 e^{-2\sigma\mu s}, \quad (\text{A.4})$$

which completes the proof.  $\blacksquare$

### B. Proof of Theorem 2

To simplify the notation, we denote  $-\beta(a_j^{\top}\hat{\mathbf{x}}(k|t) + b_j + 1/\beta) + 1$ ,  $j \in \{1, \dots, c_x\}$ ,  $k \in \{0, \dots, N-1\}$  by  $\vartheta_i$ ,  $i \in \{1, \dots, c_x N\}$ ;  $-\beta(c_j^{\top}u(k|t) + d_j + 1/\beta) + 1$ ,  $j \in \{1, \dots, c_u\}$ ,  $k \in \{0, \dots, N-1\}$  by  $\vartheta_i$ ,  $i \in \{c_x N + 1, \dots, \bar{c}N\}$ ;  $-\beta(V(\hat{\mathbf{x}}(N|t), r) - \Gamma_j(r) + 1/\beta) + 1$ ,  $j \in \{1, \dots, \bar{c}\}$  by  $\vartheta_i$ ,  $i \in \{\bar{c}N + 1, \dots, \bar{c}(N+1)\}$ ;  $\lambda_{j,k}^x$ ,  $j \in$

$\{1, \dots, c_x\}$ ,  $k \in \{0, \dots, N-1\}$  by  $\alpha_i$ ,  $i \in \{1, \dots, c_x N\}$ ;  $\lambda_{j,k}^u$ ,  $j \in \{1, \dots, c_u\}$ ,  $k \in \{0, \dots, N-1\}$  by  $\alpha_i$ ,  $i \in \{c_x N + 1, \dots, \bar{c}N\}$ ; and  $\lambda_j^{\bar{c}}$ ,  $j \in \{1, \dots, \bar{c}\}$  by  $\alpha_i$ ,  $i \in \{\bar{c}N + 1, \dots, \bar{c}(N+1)\}$ . Thus, the modified barrier function (12) takes the following form:

$$\begin{aligned} \mathcal{B}(\cdot) = & \sum_{k=0}^{N-1} \|\hat{x}(k|t) - \bar{x}_r\|_{Q_x}^2 + \sum_{k=0}^{N-1} \|u(k|t) - \bar{u}_r\|_{Q_u}^2 \\ & + \|\hat{x}(N|t) - \bar{x}_r\|_{Q_N}^2 - \sum_{i=1}^{\bar{c}(N+1)} \alpha_i \log(\vartheta_i). \end{aligned} \quad (\text{B.1})$$

*Remark B.1:* From (13) and (B.1),  $\frac{d\hat{\alpha}_i(s)}{ds} > 0$  if  $0 < \vartheta_i < 1/\beta$ , implying that when  $\vartheta_i$  is in a close proximity of  $1/\beta$ : i) there exists  $\underline{\alpha} > 0$  such that  $\hat{\alpha}_i(s) \geq \underline{\alpha}$ ; and ii)  $\Phi_i(s) = 0$ .

According to the definition of  $\vartheta_i$  as mentioned above, the constraints (5c)-(5e) are satisfied if  $\vartheta_i > 0$ ,  $\forall i$ . Note that  $\mathcal{B}(\cdot) \rightarrow \infty$  only if  $\vartheta_j \rightarrow 0^+$ ,  $j \in \{1, \dots, \xi\}$ , with  $\xi \leq \bar{c}$ . Thus, the boundedness of  $\mathcal{B}(\cdot)$  from above is equivalent to the constraint satisfaction at all  $s$ . We prove the boundedness of  $\mathcal{B}(\cdot)$  by showing that

$$\lim_{\vartheta_j \rightarrow 0^+, \forall j} \frac{d}{ds} \mathcal{B}(\cdot) < 0, \quad (\text{B.2})$$

which asserts that  $\mathcal{B}(\cdot)$  must decrease along the system trajectories when these trajectories are near the boundary.

According to (B.1) and (13), the time derivative of  $\mathcal{B}(\cdot)$  w.r.t. the auxiliary time variable  $s$  is given by

$$\frac{d}{ds} \mathcal{B}(\cdot) = -\sigma \left( \|\nabla_{\hat{\mathbf{u}}} \mathcal{B}(\cdot)\|^2 - \|\nabla_{\hat{\alpha}} \mathcal{B}(\cdot)\|^2 - (\nabla_{\hat{\alpha}} \mathcal{B}(\cdot))^\top \Phi(\cdot) \right) \quad (\text{B.3})$$

According to Remark B.1, the limiting behavior of the terms  $(\nabla_{\hat{\alpha}} \mathcal{B}(\cdot))^\top \Phi(\cdot)$  and  $\|\nabla_{\hat{\alpha}} \mathcal{B}(\cdot)\|^2$  as  $\vartheta_j \rightarrow 0^+$ ,  $\forall j$  is<sup>5</sup>:

$$(\nabla_{\hat{\alpha}} \mathcal{B}(\cdot))^\top \Phi(\cdot) = \mathcal{O}(1), \quad (\text{B.4})$$

$$\|\nabla_{\hat{\alpha}} \mathcal{B}(\cdot)\|^2 = \sum_{j=1}^{\xi} \log(\vartheta_j)^2 + \mathcal{O}(1) \leq \xi (\log(\underline{\vartheta}))^2 + \mathcal{O}(1), \quad (\text{B.5})$$

where  $\underline{\vartheta} := \min_{j \in \{1, \dots, \xi\}} \vartheta_j$ , and  $\underline{\vartheta} \rightarrow 0^+$  as  $\vartheta_j \rightarrow 0^+$ ,  $\forall j$ . Also, the limiting behavior of the term  $\|\nabla_{\hat{\mathbf{u}}} \mathcal{B}(\cdot)\|^2$  as  $\vartheta_j \rightarrow 0^+$ ,  $\forall j$  is

$$\|\nabla_{\hat{\mathbf{u}}} \mathcal{B}(\cdot)\|^2 = \frac{\beta^2}{\underline{\vartheta}^2} \left\| \sum_{j=1}^{\xi} \frac{\vartheta_j}{\vartheta_j} \alpha_j \left( \frac{\partial \vartheta_j}{\partial \hat{\mathbf{u}}} \right)^\top \right\|^2 + \mathcal{O}(1), \quad (\text{B.6})$$

where  $\left( \frac{\partial \vartheta_j}{\partial \hat{\mathbf{u}}} \right)^\top$  are outward vectors normal to curves associated with active constraints. According to Remark B.1, and by applying the lemma in Appendix of [47] and the Alexandrov's theorem [48, pp. 333], it can be concluded that there exists  $\epsilon > 0$  such that the limiting behavior of the term  $\|\nabla_{\hat{\mathbf{u}}} \mathcal{B}(\cdot)\|^2$  as  $\vartheta_j \rightarrow 0^+$ ,  $\forall j$  satisfies

$$\|\nabla_{\hat{\mathbf{u}}} \mathcal{B}(\cdot)\|^2 \geq \frac{\epsilon \beta^2 \underline{\alpha}^2}{\underline{\vartheta}^2} + \mathcal{O}(1). \quad (\text{B.7})$$

<sup>5</sup>Given  $f : \mathbb{R}^m \rightarrow \mathbb{R}$ ,  $f(x) = \mathcal{O}(1)$  means that  $\exists M > 0$  such that  $|f(x)| < M$ .

Therefore, by taking the limit from both sides of (B.3) as  $\vartheta_j \rightarrow 0^+$ ,  $\forall j$ , and by applying the L'Hôpital's rule, we get

$$\lim_{\vartheta_j \rightarrow 0^+, \forall j} \frac{d}{ds} \mathcal{B}(\cdot) \leq \lim_{\underline{\vartheta} \rightarrow 0^+} -\sigma \left( \frac{\epsilon \beta^2 \underline{\alpha}^2}{\underline{\vartheta}^2} - \xi (\log(\underline{\vartheta}))^2 \right) < 0, \quad (\text{B.8})$$

which completes the proof.  $\blacksquare$