MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Reference Tracking with Guaranteed Error Bound for Constrained Linear Systems

Di Cairano, S.; Borrelli, F.

TR2015-124 October 2015

Abstract

We propose a control design for a constrained linear system to track reference signals within a given bounded error. The admissible reference signals are generated as output trajectories of a reference generator, which is a linear system driven by unknown bounded inputs. The controller has to track the reference signals and to never violate a given tracking error bound, while satisfying state and input constraints, for any admissible reference. The design is based on a model predictive controller (MPC) enforcing a polyhedral robust control invariant set defined by the system and reference generator models and constraints. We describe an algorithm to compute the robust control invariant set and how to design the tracking MPC law that guarantees satisfaction of the tracking error bound and of the system constraints, and achieves persistent feasibility. We demonstrate the proposed method in two case studies.

2015 IEEE Transactions on Automatic Control

Reference Tracking with Guaranteed Error Bound for Constrained Linear Systems

Stefano Di Cairano, Francesco Borrelli

***Abstract*—We propose a control design for a constrained linear system to track reference signals within a given bounded error. The admissible reference signals are generated as output trajectories of a reference generator, which is a linear system driven by unknown bounded inputs. The controller has to track the reference signals and to never violate a given tracking error bound, while satisfying state and input constraints, for any admissible reference. The design is based on a model predictive controller (MPC) enforcing a polyhedral robust control invariant set defined by the system and reference generator models and constraints. We describe an algorithm to compute the robust control invariant set and how to design the tracking MPC law that guarantees satisfaction of the tracking error bound and of the system constraints, and achieves persistent feasibility. We demonstrate the proposed method in two case studies.**

I. Introduction

Several control applications require a manipulated variable to track a time-varying reference signal with complicated behavior within given error bounds. For instance, in dual-stage processing machines in manufacturing [1], a slow stage with large workspace (also called operating range) moves a stage with much faster dynamics and limited workspace that actuates the work tool. Thus, the tool workspace is the sum of the stages' workspace. Because of the timescale separation of the stages (often few orders of magnitude), to ensure tool trajectory tracking it is enough to control the slow stage so that its distance from the tool position reference is always within the operating range of the fast stage. The automotive and aerospace industries are also rich of applications. The engine torque in spark-ignition engines is obtained by controlling airflow and spark timing [2]. Due to the limited torque generated from spark timing, the airflow-generated torque must be controlled in a bounded range around the requested torque. In hybrid electric vehicles the combustion engine power needs to be controlled so that the difference with the driver-requested power can be achieved by electric power [3]. Vehicle cornering control and attitude control with redundant actuators [4], [5] are also potential applications.

The design of reference tracking controllers enforcing input and state constraints has mainly followed two approaches. In [6]–[12] reference tracking model predictive control (MPC) was proposed, usually focused on guaranteeing *asymptotic* offset-free control for references within a specific class of signals when the constraints are inactive. Commonly, the class of reference signals consists of the outputs of an *autonomous* linear system. Instead, in [13], [14] a reference governor (RG) is used. RG modifies the reference based on the system state to generate a virtual reference, so that the closed-loop system satisfies state constraints. RG achieves finite-time convergence of the virtual reference to a constant feasible reference, and hence *asymptotic* offset-free output tracking of constant feasible references.

In this paper we design a reference tracking controller that satisfies state and input constraints and guarantees a given bound on tracking error during both *steady-state and transient*, without modifying the reference signal. We consider a class of reference signals which is more general than the outputs of a linear autonomous system. The reference signal is the output of a "reference generator", which is a constrained linear system *driven by unknown bounded inputs*. The simplest class of references that can be modeled with this approach [15, p. 159] is a bounded signal with bounded first derivative. Clearly, the class of references that can be produced is significantly richer than those produced by an autonomous linear system. Since the reference signal cannot be modified, the virtual setpoint-augmented MPC in [11], [16], which optimizes the reference jointly with the control commands, and the method in [17], which modifies the reference to keep the state trajectories in a specified region, are not applicable. The case of non-modifiable references is motivated by practical applications, such as in [1], [2], where the reference is generated by complex algorithms that have to satisfy requirements in multiple domains. There, the admissible reference manipulations are not described by simple functions and hence integrating reference manipulation with tracking control is not a viable option for implementation. Rather, the reference is generated offline or by supervisory algorithms to satisfy the specifications and to belong to a class of signals, and the control algorithm needs to guarantee that any reference within such class of signals can be tracked within the desired error bound.

The control design proposed here is based on computing offline a robust control invariant (RCI) set of plant and reference states for which there exists a control law that guarantees satisfaction of the constraints and of the bound on the tracking error at all times and for all admissible references. Such RCI is then enforced by MPC. Thus, the contributions of the papers are, (i) an algorithm for computing a polyhedral RCI set for the bounded tracking problem, and (ii), an MPC design that uses such RCI set to guarantee that the tracking error bound and the system constraints are persistently satisfied.

The paper[1] is organized as follows. In Section II we review preliminary results and formalize the problem. In Section III we describe the algorithm to compute the RCI set for bounded tracking, which is used in Section IV to design the MPC controller for bounded tracking. In Section V we present case studies validating the approach, and in Section VI we summarize our conclusions.

Notation: $\mathbb{R}$, $\mathbb{R}_{0+}$, $\mathbb{R}_+$ and $\mathbb{Z}$, $\mathbb{Z}_{0+}$, $\mathbb{Z}_+$ are the sets of real, nonnegative real, positive real, and integer, nonnegative integer, positive integer numbers. By $[a]_i$ we denote the i-th component of a, for $a \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $(a,b) = [a'\ b']' \in \mathbb{R}^{n+m}$ is the stacked vector, and I and 0 are the identity and the zero matrices of appropriate size. By $\|\cdot\|$ we denote the ∞-norm (or 1-norm), and $\mathcal{B}(\rho)$ where $\rho \in \mathbb{R}_+$ denotes the ∞-norm (or 1-norm) ball of appropriate dimension of radius ρ. For sets $\mathcal{A}$, $\mathcal{D}$, $\mathrm{proj}_a(\mathcal{A})$ denotes the projection of $\mathcal{A}$ onto the subspace that contains the vector a, $\mathcal{A} \oplus \mathcal{D}$ is the Minkowski (set) sum. For a discrete-time signal $x \in \mathbb{R}^n$ with sampling period T_s, x_t is the value at sampling instant t, i.e., at time $T_s t$, and $x_{k|t}$ denotes the predicted value of x at sample $t+k$, i.e., x_{t+k}, based on data at sample t, where $x_{0|t} = x_t$.

II. Preliminaries and Problem Definition

First we review results on invariant sets, see [19], [20] for details. Then, we formalize the problem tackled in this paper.

S. Di Cairano is with Mitsubishi Electric Research Laboratories, Cambridge, MA, email: dicairano@ieee.org

F. Borrelli is with the dept. Mechanical Eng., the University of California Berkeley, CA, email: fborrelli@berkeley.edu

[1]Preliminary results were presented in [18].

A. Preliminaries on Invariant Sets

Consider the system

$$x_{t+1} = f(x_t, u_t, w_t), \tag{1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $w \in \mathbb{R}^d$ are the state, input and disturbance vectors, respectively, subject to the constraints

$$x_t \in \mathcal{X},\ u_t \in \mathcal{U},\ w_t \in \mathcal{W},\ \forall t \in \mathbb{Z}_{0+}. \tag{2}$$

A robust control invariant set in $\mathcal{X}$ is a set of states for which there exists a control law such that (1) never violates (2) for any sequence of disturbances such that $w_t \in \mathcal{W}$ for all $t \in \mathbb{Z}_{0+}$.

Definition 1 (Robust control invariant set): A set $\mathcal{C} \subseteq \mathcal{X}$ is said to be a robust control invariant (RCI) set for (1), (2) if

$$x_t \in \mathcal{C} \Rightarrow \exists u_t \in \mathcal{U} : f(x_t, u_t, w_t) \subseteq \mathcal{C},\ \forall w_t \in \mathcal{W},\ \forall t \in \mathbb{Z}_{0+}.$$

The *maximal* RCI set $\mathcal{C}^\infty$ contains all other RCI sets in $\mathcal{X}$. □

The computation of RCI sets relies on the **Pre**-set operator

$$\begin{aligned}\textbf{Pre}(\mathcal{S}, \mathcal{W}) \quad \triangleq \quad & \{x \in \mathbb{R}^n \ :\ \exists u \in \mathcal{U} \\ & f(x, u, w) \subseteq \mathcal{S},\ \forall w \in \mathcal{W}\},\end{aligned} \tag{3}$$

which computes the set of states of (1) that can be robustly driven to the target set $\mathcal{S} \in \mathbb{R}^n$ in one step.

The procedure to compute the maximal RCI set for (1) subject to (2) based on (3) is summarized by the following algorithm.

Algorithm 1 (Computation of $\mathcal{C}^\infty$):

1) $\Omega_0 \leftarrow \mathcal{X}$
2) $\Omega_{k+1} \leftarrow \textbf{Pre}(\Omega_k, \mathcal{W}) \cap \Omega_k$
3) `If` $\Omega_{k+1} = \Omega_k$, $\mathcal{C}^\infty \leftarrow \Omega_k$, `return`
4) $k \leftarrow k + 1$, `goto` 2.

Algorithm 1 generates the sequence of sets $\{\Omega_k\}_k$, satisfying $\Omega_{k+1} \subseteq \Omega_k$, for all $k \in \mathbb{Z}_{0+}$. In general, Algorithm 1 may not terminate. If the algorithm terminates in a finite number of iterations k^*, Ω_{k^*} is the maximal RCI set $\mathcal{C}^\infty$ for (1) subject to (2). If for some $\tilde{k}$, $\Omega_{\tilde{k}} = \emptyset$, no RCI set exists. See [19] for details on the termination of Algorithm 1.

Definition 2 (Input admissible set of $\mathcal{C}$): Given a RCI set $\mathcal{C}$ for (1), (2), the input admissible set for $x \in \mathcal{C}$ is

$$\mathcal{C}^u(x) = \{u \in \mathcal{U} : f(x, u, w) \in \mathcal{C},\ \forall w \in \mathcal{W}\},$$

where $\mathcal{C}^u(x) \neq \emptyset$ if $x \in \mathcal{C}$, and $\mathcal{C}^u(x) = \emptyset$ if $x \notin \mathcal{C}$.

Definition 3: When $\mathcal{W} = \{0\}$, i.e., (1) is not subject to disturbances, $\mathcal{C}$ in Definition 1 is called control invariant (CI) set.

B. Problem definition

Consider the discrete-time linear system

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t, \end{aligned} \tag{4}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$ are the state, input and output vectors, respectively. System (4) is subject to the constraints

$$x_t \in \mathcal{X},\ u_t \in \mathcal{U},\ \forall t \in \mathbb{Z}_{0+}. \tag{5}$$

We want (4) to track within a given error bound the time-varying reference signal y^r_t generated by the reference model

$$\begin{aligned} r_{t+1} &= A^r r_t + B^r \gamma_t \\ y^r_t &= C^r r_t, \end{aligned} \tag{6}$$

where $r \in \mathbb{R}^{n_r}$, $\gamma \in \mathbb{R}^{m_r}$ and $y^r \in \mathbb{R}^p$ are the reference model state, input, and output vectors, respectively, subject to the constraints

$$r_t \in \mathcal{R},\ \gamma_t \in \Gamma. \tag{7}$$

We assume that the input to (6) is selected by a *reference generator algorithm* (RGA). At every time instant $t \in \mathbb{Z}_{0+}$, RGA computes $\gamma_t \in \Gamma$ based on $r_t \in \mathcal{R}$ such that $r_{t+1} \in \mathcal{R}$. Thus, $r_t \in \mathcal{C}^r$ where $\mathcal{C}^r$ is a CI set for (6), (7), and RGA chooses γ_t such that $r_t \in \mathcal{C}^r$. By the definitions in Section II-A, $\mathcal{C}^r$ for (6), (7) is such that

$$r_t \in \mathcal{C}^r \Rightarrow\ \exists \gamma_t \in \Gamma :\ r_{t+1} \in \mathcal{C}^r,\ \forall t \in \mathbb{Z}_{0+}, \tag{8}$$

and $\mathcal{C}^\gamma(r)$ is the input admissible set associated to $\mathcal{C}^r$ in (8). The properties of the RGA are summarized in the following assumption.

Assumption 1: At every $t \in \mathbb{Z}_{0+}$, given $r_t \in \mathcal{C}^r$, where $\mathcal{C}^r$ is a known CI set of (6), (7), the RGA enforces $r_{t+1} \in \mathcal{C}^r \subseteq \mathcal{R}$ by selecting $\gamma_t \in \mathcal{C}^\gamma(r_t) \subseteq \Gamma$. □

If (6) is an integrator ($A^r = 1$, $B^r = 1$, $C^r = 1$), then $\mathcal{C}^{r,\infty} = \mathcal{R}$, $\mathcal{C}^\gamma(r) = \{\gamma \in \Gamma\ |r + \gamma \in \mathcal{R}\}$. In general, the bounds on r induce state-dependent bounds on γ. For instance, when at/close to the border of $\mathcal{C}^r$, not all the values in Γ are admissible for γ.

The RGA is separated from the controller and, as opposed to the approach in [11], [16], [17], its output cannot be modified by the controller. Also, as opposed to the reference governor [13], [14], the RGA does not guarantees the enforcement of system constraints. The RGA only guarantees that the reference signal will satisfy (7). The tracking control design proposed in this paper does not depend on a specific RGA, but only requires Assumption 1 to be satisfied.

The problem we address in this paper is formalized as follows.

Problem 1: Consider system (4) subject to (5), reference model (6) subject to (7), and a given tracking error bound $\epsilon \in \mathbb{R}_+$. Let $R^N_t = [r'_{0|t}, \ldots, r'_{N|t}]'$, $N \in \mathbb{Z}_{0+}$ be a predicted reference profile satisfying Assumption 1. Design a control law $u_t = \kappa(x_t, R^N_t)$ and a set $\mathcal{X}_0 \subseteq \mathbb{R}^n \times \mathbb{R}^{n_r}$ of initial states and references $(x_0, r_0) \in \mathcal{X}_0$ from which (4) in closed-loop with $\kappa(x_t, R^N_t)$ satisfies (5) and

$$\|y_t - y^r_t\| \leq \epsilon, \tag{9}$$

for all $r_t \in \mathcal{R}$ that satisfy Assumption 1, at every $t \in \mathbb{Z}_{0+}$. □

Remark 1: Problem 1 considers predicted future reference profiles of length $N \in \mathbb{Z}_{0+}$. However, it is not guaranteed that such reference preview is *reliable*. That is, the reference predicted k steps ahead from time t may not be the actual reference at time $t+k$, i.e., $r_{t+k} \neq r_{k|t}$. Thus, a solution to Problem 1 needs to allow for the reference preview to change unexpectedly during operation.

III. RCI Set for Bounded Error Tracking

Consider (4) subject to (5), and (6) subject to (7). By Assumption 1 $r_t \in \mathcal{C}^r$, for every $t \in \mathbb{Z}_{0+}$, and given $r_t \in \mathcal{C}^r$, $r_{t+1} \in \mathcal{C}^r$ if and only if $\gamma \in \mathcal{C}^\gamma(r_t)$. We define

$$\mathcal{X}^{x,r} = \{(x, r) : x \in \mathcal{X},\ r \in \mathcal{C}^r, (Cx - C^r r) \in \mathcal{B}(\epsilon)\}. \tag{10}$$

At any time $t \in \mathbb{Z}_{0+}$, given $(x_t, r_t) \in \mathcal{X}_0 \subseteq \mathcal{X}^{x,r}$, the control law that solves Problem 1 must guarantee $(x_{t+1}, r_{t+1}) \in \mathcal{X}^{x,r}$ for every admissible r_{t+1}. Thus, we look for a RCI set $\mathcal{C}^{x,r} \subseteq \mathcal{X}^{x,r}$ for (4), (6) subject to (5), (7) such that

$$\begin{aligned} & (x, r) \in \mathcal{C}^{x,r} \Rightarrow \exists u \in \mathcal{U} : \\ & \quad (Ax + Bu, A^r r + B^r \gamma) \in \mathcal{C}^{x,r},\ \forall \gamma \in \mathcal{C}^\gamma(r). \end{aligned} \tag{11}$$

The maximal RCI set for (11) by Algorithm 1 requires computing

$$\begin{aligned} \textbf{Pre}(\Omega_k, \mathcal{C}^\gamma(r)) = \{ & (x, r) :\ r \in \mathcal{C}^r,\ \exists u \in \mathcal{U}, \\ & (Ax + Bu, A^r r + B^r \gamma) \in \Omega_k, \forall \gamma \in \mathcal{C}^\gamma(r)\}, \end{aligned} \tag{12}$$

where $\Omega_0 = \mathcal{X}^{x,r}$ and $\Omega_k \subseteq \mathcal{X}^{x,r}$. The computation of (12) is challenging even if Ω_k, $\mathcal{C}^\gamma(r)$, $\mathcal{X}^{x,r}$ and $\mathcal{U}$ are polyhedral, because $\gamma \in \mathcal{C}^\gamma(r)$, i.e., the unknown disturbance belongs to a state-dependent set. The algorithm in [21], [22] computes (12) as a union of polyhedra, which is in general non-convex and whose complexity, in terms of convex elements in the union, grows exponentially with the number of iterations. Next, we propose a method to compute a *polyhedral* $\mathcal{C}^{x,r}$. The resulting $\mathcal{C}^{x,r}$ is a subset of the maximal RCI obtained from [21], [22], but is convex and polyhedral, and thus significantly easier to compute and use in a control algorithm.

Algorithm 2 (Computation of $\mathcal{C}^{x,r}$):

1) `Initialization:`
$k=0$,
$$\bar{\Omega}_0 = (\mathbb{R}^n \times \mathcal{C}^r) \cap \{(x,r): \ (x,r) \in \bar{\Omega}_0^{x,r}\}. \tag{13}$$
where $\bar{\Omega}_0^{x,r} = \{(x,r): \ x \in \mathcal{X}, (Cx - C^r r) \in \mathcal{B}(\epsilon)\}$

2) $\bar{\Omega}_{k+1} = \mathbf{Pre}_f(\bar{\Omega}_k, \Gamma) \cap \bar{\Omega}_k^{x,r}$ where
$$\begin{aligned} \mathbf{Pre}_f(\bar{\Omega}_k, \Gamma) &= (\mathbb{R}^n \times \mathcal{C}^r) \cap \tilde{\Omega}_{k+1}^{x,r} \\ \tilde{\Omega}_{k+1}^{x,r} &= \{(x,r): \ \exists u \in \mathcal{U}, \\ &\quad (Ax+Bu, A^r r + B^r \gamma) \in \bar{\Omega}_k^{x,r}, \forall \gamma \in \Gamma\}, \\ \bar{\Omega}_{k+1}^{x,r} &= \tilde{\Omega}_{k+1}^{x,r} \cap \bar{\Omega}_k^{x,r}. \end{aligned} \tag{14}$$

3) `if` $\bar{\Omega}_{k+1} = \bar{\Omega}_k$, `then` $\mathcal{C}^{x,r} \leftarrow \bar{\Omega}_k$, `return`

4) $k \leftarrow k+1$, `goto 2`

Algorithm 2 may terminate with an empty set, where if $\bar{\Omega}_k = \emptyset$, $\bar{\Omega}_{k+1} = \bar{\Omega}_k$ and hence the termination condition at Step 3 is met. This indicates that a (non-maximal) RCI set could not be found. Also, Algorithm 1 may never terminate because the RCI set may not be finitely determined [19], [20]. In practice, a maximum number of iterations $k_{\max}$ is used after which the termination of Algorithm 1 is enforced with $\mathcal{C}^{x,r} = \emptyset$. Thus, if Algorithm 1 terminates with $\mathcal{C}^{x,r} = \emptyset$, a polyhedral RCI set is not found, either because no polyhedral RCI set exists, or because no polyhedral RCI could be found within the iteration limit. In these cases the problem specifications must be changed by shrinking Γ, and/or increasing $k_{\max}$.

Theorem 1: Let Assumption 1 hold, and let Algorithm 2 converge in a finite number of iterations to $\mathcal{C}^{x,r} \neq \emptyset$. Then, $\mathcal{C}^{x,r}$ computed by Algorithm 2 is a polyhedral RCI set for (4), (6) subject to (5), (7), (9) and robust to any $\gamma^r \in \mathcal{C}^\gamma(r)$ and $\mathcal{C}^{x,r} \subseteq \mathcal{C}_\infty^{x,r}$.

Proof: Consider Algorithm 1 and its set sequence $\{\Omega_0, \Omega_1, \ldots\}$ computed by (12) with $\Omega_0 = \mathcal{X}^{x,r}$, and Algorithm 2 and its set sequence $\{\bar{\Omega}_0, \bar{\Omega}_1, \ldots\}$. The **Pre** in (12) can be rewritten as

$$\begin{aligned} \mathbf{Pre}(\Omega_k, \mathcal{C}^\gamma(r)) = \{(x,r): \ &\exists \gamma^r \in \mathcal{C}^\gamma(r), \ A^r r + B^r \gamma^r \in \mathcal{C}_r \\ &\exists u \in \mathcal{U}, \ (Ax+Bu, A^r r + B^r \gamma) \in \Omega_k, \forall \gamma \in \mathcal{C}^\gamma(r)\}, \end{aligned}$$

since $\{r \in \mathcal{C}^r : \exists \gamma^r \in \mathcal{C}^\gamma(r), \ A^r r + B^r \gamma^r \in \mathcal{C}^r\} = \mathcal{C}^r$ by the definition of CI set, and since $\mathcal{C}^\gamma(r) = \emptyset$ if $r \notin \mathcal{C}^r$.

We prove the theorem by induction, starting from $\bar{\Omega}_0 = \Omega_0 = \mathcal{X}^{x,r}$, where $\bar{\Omega}_0$ is polyhedral. Assume that at step k, $\bar{\Omega}_k \subseteq \Omega_k$, $\bar{\Omega}_k = (\mathbb{R}^n \times \mathcal{C}^r) \cap \bar{\Omega}_k^{x,r}$, and $\bar{\Omega}_k^{x,r}$ is polyhedral, and consider Algorithm 2. Indeed, the assumptions hold for $k=0$.

First, $\bar{\Omega}_{k+1} = \mathbf{Pre}_f(\bar{\Omega}_k, \Gamma) \cap \bar{\Omega}_k^{x,r} = (\mathbb{R}^n \times \mathcal{C}^r) \cap \tilde{\Omega}_{k+1}^{x,r} \cap \bar{\Omega}_k^{x,r} = (\mathbb{R}^n \times \mathcal{C}^r) \cap \bar{\Omega}_{k+1}^{x,r}$ since $\tilde{\Omega}_{k+1}^{x,r} \cap \bar{\Omega}_k^{x,r} = \bar{\Omega}_{k+1}^{x,r}$ from Step 2 in Algorithm 2. Also, $\bar{\Omega}_{k+1} = \mathbf{Pre}_f(\bar{\Omega}_k, \Gamma) \cap \bar{\Omega}_k^{x,r} = (\mathbb{R}^n \times \mathcal{C}^r) \cap \tilde{\Omega}_{k+1}^{x,r} \cap \bar{\Omega}_k^{x,r} \cap (\mathbb{R}^n \times \mathcal{C}^r) = \mathbf{Pre}_f(\bar{\Omega}_k, \Gamma) \cap \bar{\Omega}_k$, by the intersection being idempotent, by the definition of $\mathbf{Pre}_f$ in (14), and by $\bar{\Omega}_k^{x,r} \cap (\mathbb{R}^n \times \mathcal{C}^r) = \bar{\Omega}_k$ being an assumption of the inductive step.

Thus, the backward iteration in Algorithm 2 is in the form of that in Algorithm 1, the formula is recursive under the inductive step assumption, which is satisfied for $k=0$. Hence, Step 2 in Algorithm 2 can be executed iteratively, and Step 3 is a geometric condition for invariance, which, if satisfied, indicates that an RCI set is found.

Consider now the $\mathbf{Pre}_f$ operation in (14), and rewrite it as

$$\begin{aligned} \mathbf{Pre}_f(\bar{\Omega}_k, \Gamma) = \{(x,r): \ &r \in \mathcal{C}^r, \ \exists u \in \mathcal{U}, \\ &(Ax+Bu, A^r r + B^r \gamma) \in \bar{\Omega}_k^{x,r}, \forall \gamma \in \Gamma\} \end{aligned} \tag{15}$$

where the order of projection and intersection can be inverted because the set $(\mathbb{R}^n \times \mathcal{C}^r)$ is independent of the projection variable u. With considerations similar to the ones for (12), and since $\bar{\Omega}_k = (\mathbb{R}^n \times \mathcal{C}^r) \cap \bar{\Omega}_k^{x,r}$ and we already impose $r \in \mathcal{C}^r$, which is CI,

$$\begin{aligned} \mathbf{Pre}_f(\bar{\Omega}_k, \Gamma) = \{(x,r): \ &\exists \gamma^r \in \mathcal{C}^\gamma(r), \ A_r r + B_r \gamma^r \in \mathcal{C}^r, \\ &\exists u \in \mathcal{U}, \ (Ax+Bu, A^r r + B^r \gamma) \in \bar{\Omega}_k, \forall \gamma \in \Gamma\}. \end{aligned} \tag{16}$$

By (16) we can compare $\mathbf{Pre}_f$ in (14) and **Pre** in (12) and obtain $\mathbf{Pre}(\Omega_k, \mathcal{C}^\gamma(r)) \supseteq \mathbf{Pre}_f(\bar{\Omega}_k, \Gamma)$, since $\Gamma \supseteq \mathcal{C}^\gamma(r)$ for all $r \in \mathcal{C}^r$ and $\bar{\Omega}_k \subseteq \Omega_k$ by the assumption of the induction step. Thus, $\mathcal{C}^{x,r} \subseteq \mathcal{C}_\infty^{x,r}$.

Since the disturbance set $\mathcal{C}^\gamma(r)$ is state dependent, Ω_{k+1} is usually nonconvex [21]. Instead, by the assumption of the inductive step $\bar{\Omega}_k^{x,r}$ is polyhedral and since Γ is polyhedral and $\mathcal{C}^r$ are polyhedral, $\bar{\Omega}_{k+1}^{x,r}$ and $\bar{\Omega}_{k+1}$ are polyhedral, because all the performed operations when acting on polyhedra, return polyhedra [19]. ■

From Theorem 1, if Algorithm 2 terminates at step k^* and $\Omega_{k^*} \neq \emptyset$, we obtain a polyhedral RCI $\mathcal{C}^{x,r} = \Omega_{k^*} = \{x \in \mathbb{R}^n, r \in \mathbb{R}^{n_r} : \ H_{k^*}[x' \ r']' \leq K_{k^*}\}$ and the associated input admissible set $\mathcal{C}^u((x,r)) = \{u \in \mathbb{R}^m : \ [x' \ r' \ u']' \in \mathcal{P}_{k^*}^{(x,r,u)}\} = \{u \in \mathbb{R}^m : \ H_{k^*}^{(u)}[x' \ r']' + J_{k^*}^{(u)} u \leq K_{k^*}^{(u)}\}$ is also polyhedral. The computationally expensive step of Algorithm 2 is the evaluation of $\mathbf{Pre}_f$, which requires the "robustification" of the tracking constraints with respect to $\gamma \in \Gamma$, i.e., the computation of the worst case value of $\gamma \in \Gamma$ for the enforcement of the tracking constraints, and the projection to eliminate u. The latter tends to be computationally expensive due to the projection algorithm. The former requires the solution of linear programs, and hence is computationally cheap. Algorithm 2 can be straightforwardly extended to the cases where (4) is subject to additive disturbances $w \in \mathcal{W} \subseteq \mathbb{R}^n$ by enforcing the constraints in $\tilde{\Omega}_{k+1}^{x,r}$ for all $w \in \mathcal{W}$, which again can be achieved by solving linear programs when $\mathcal{W}$ is a polytope [19].

Remark 2: Algorithm 2 may converge to $\mathcal{C}^{x,r}$ such that $\mathrm{proj}_r(\mathcal{C}^{x,r}) \subset \mathcal{C}^r$. In this case, it follows trivially from (14) that $\mathrm{proj}_r(\mathcal{C}^{x,r})$ is RCI for (6) for all $\gamma \in \Gamma \supseteq \mathcal{C}^\gamma(r)$, i.e., for all inputs selected by RGA according to Assumption 1. Thus, if $r_t \in \mathrm{proj}_r(\mathcal{C}^{x,r})$, $r_{t+\tau} \in \mathrm{proj}_r(\mathcal{C}^{x,r})$ for all $\tau \in \mathbb{Z}_{0+}$ for any input sequence selected by the RGA. Hence, $\mathrm{proj}_r(\mathcal{C}^{x,r})$ is the (invariant) set of initial reference states from which any trajectory satisfying Assumption 1 is properly tracked. In fact, by (14), if $(x,r) \in \mathcal{C}^{x,r}$, there exists $u \in \mathcal{U}$ such $(Ax+Bu, A^r r + B^r \gamma) \in \mathcal{C}^{x,r}$, and hence $A^r r + B^r \gamma \in \mathrm{proj}_r(\mathcal{C}^{x,r})$, for all $\gamma \in \Gamma \supseteq \mathcal{C}^\gamma(r)$.

Remark 3: The differences of Algorithm 2 with Algorithm 1 are as follows. First, by Assumption 1, the reference state is constrained within $\mathcal{C}^r$ from the beginning. Pre-computing $\mathcal{C}^r$ reduces the computational burden because the sets for reference states and inputs do not change with the iterations. Second, Algorithm 2 robustifies with respect to Γ instead of $\mathcal{C}^\gamma(r)$, thus removing the dependency of γ from the reference state. Third, in Algorithm 2 the set determined by the enforcement of the state and tracking constraints, and the set determined by the admissible reference states are kept separated. This allows the more conservative robustification to affect only the tracking constraints. In fact, while $A^r r + B^r \gamma \in \mathcal{C}^r$, for all $\gamma \in \mathcal{C}^\gamma(r)$, the same is not true if $\gamma \in \Gamma$. If we substitute $\mathcal{C}^\gamma(r)$ with Γ in (12), the set of admissible reference states may shrink and even become empty, causing $\mathcal{C}^{x,r}$ to be empty. For instance, when (6), (7) model a constrained integrator, substituting $\mathcal{C}^\gamma(r)$ with Γ in (12) causes $\mathcal{C}^{x,r} = \emptyset$ because from any initial reference state, the reference constraints can eventually be violated. Instead, in Algorithm 2 the set of admissible reference states remains $\mathcal{C}_r$ and the effect of using Γ rather than $\mathcal{C}_r(\gamma)$ results in requiring (9) to be enforced for $r \in (\mathcal{C}^r \oplus \Gamma)$, rather than for $r \in \mathcal{C}^r$. Thus, by neglecting that close to the boundary of $\mathcal{C}^r$ not all $\gamma \in \Gamma$ can occur, we introduce conservativeness in the RCI set. However, such condition can still be satisfied, and the obtained RCI is a polyhedron, and hence simpler to compute and use in a control law than a union of polyhedra. □

Based on Remark 3, the following proposition is immediate.

Proposition 1: Let $\mathcal{C}^{x,r}$ be the RCI set obtained by Algorithm 2, and let $\hat{\mathcal{C}}^r \subseteq \mathcal{C}^r \subseteq \mathcal{R}$ be a RCI set for (6) such that $\hat{\mathcal{C}}^\gamma(r) = \Gamma$ for all $r \in \hat{\mathcal{C}}^r$. Then, for the maximal RCI set $\hat{\mathcal{C}}^{x,r}_\infty$ obtained by Algorithm 1 using (12) where $\hat{\mathcal{C}}^r$ and $\hat{\mathcal{C}}^\gamma(r)$ are used in place of $\mathcal{C}^r$ and $\mathcal{C}^\gamma(r)$ it holds $\hat{\mathcal{C}}^{x,r}_\infty \subseteq \mathcal{C}^{x,r}$. Also, if $\mathcal{C}^\gamma(r) = \Gamma$ for all $r \in \mathcal{C}^r$, $\mathcal{C}^{x,r} = \mathcal{C}^{x,r}_\infty$.

IV. MPC with Guaranteed Tracking Error Bound

The RCI set $\mathcal{C}^{x,r}$ is a set of states and references satisfying the constraints and for which there exists at least an input that tracks any future admissible reference signal within the allowed error bound and enforces constraints. Given $(x, r) \in \mathcal{C}^{x,r} \subseteq \mathcal{X} \times \mathcal{C}^r$, the set of inputs that guarantees $(x_{t+1}, r_{t+1}) \in \mathcal{C}^{x,r}$ for any admissible r_{t+1} is the input admissible set $\mathcal{C}^u((x_t, r_t))$. Next, we propose an approach based on MPC for computing $u_t \in \mathcal{U}$ such that $(x_{t+1}, r_{t+1}) \in \mathcal{C}^{x,r}$.

At $t \in \mathbb{Z}_{0+}$, let $x_t \in \mathcal{X}$ and $R^N_t = [r'_{0|t}, \dots, r'_{N|t}]'$, the reference state trajectory along a future horizon of length $N \in \mathbb{Z}_+$, be given, where R^N_t is generated by RGA according to Assumption 1. Let $\bar{k} \in \mathbb{Z}_{0+}$, $\bar{k} \le N$, be such that $r_{t+k} = r_{t|k}$, and hence $r_{k-1|t+1} = r_{k|t}$, for $k = 1, \dots, \bar{k}$ for all $t \in \mathbb{Z}_{0+}$, i.e., $\bar{k}$ is the number of steps of reliable preview. Consider the finite horizon optimal control problem,

$$
\begin{aligned}
& g_U(x_t, R^N_t) = && \text{(17a)}\\
& \quad \arg\min_{U_t} \sum_{k=0}^{N-1} q(y_{k|t}, y^r_{k|t}, x_{k|t}, u_{k|t}) && \text{(17b)}\\
& \quad \text{s.t.} \quad x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, && \text{(17c)}\\
& \quad \phantom{\text{s.t.}} \quad y_{k|t} = Cx_{k|t}, && \text{(17d)}\\
& \quad \phantom{\text{s.t.}} \quad y^r_{k|t} = C^r r_{k|t}, && \text{(17e)}\\
& \quad \phantom{\text{s.t.}} \quad (x_{k|t}, r_{k|t}) \in \mathcal{X}^{x,r},\ k = 0, \dots, \bar{k}-1, && \text{(17f)}\\
& \quad \phantom{\text{s.t.}} \quad u_{k|t} \in \mathcal{U},\ k = 0, \dots, \bar{k}-1, && \text{(17g)}\\
& \quad \phantom{\text{s.t.}} \quad u_{k|t} \in \mathcal{C}^u((x_{k|t}, r_{k|t})),\ k = \bar{k}, \dots, N-1, && \text{(17h)}\\
& \quad \phantom{\text{s.t.}} \quad x_{0|t} = x_t, && \text{(17i)}
\end{aligned}
$$

where $U_t = [u'_{0|t} \dots u'_{N-1|t}]'$, $q(\cdot)$ is a nonnegative stage cost, with a little abuse of notation (17f), (17g) are ignored if $\bar{k} = 0$, and let $U^*_t = [u^*_{0|t}{}' \dots u^*_{N-1|t}{}']'$ be the optimal solution of (17). The first element of U^*_t is applied to (4), hence obtaining the control law

$$u_t = g_{\mathrm{RHC}}(x_t, R^N_t) = [I\ 0 \cdots 0] g_U(x_t, R^N_t). \tag{18}$$

At time $t + 1$, (17) is solved based on the new state $x_{0|t+1} = x_{t+1}$, and R^N_{t+1}. The next theorem presents sufficient conditions guaranteeing persistent feasibility of the MPC control law (17), (18).

Theorem 2: Consider system (4) and reference model (6) subject to (5), (7) and tracking constraint (9). Let Assumption 1 hold, i.e., $\gamma_t \in \mathcal{C}^r(r_t)$ for all $t \in \mathbb{Z}_{0+}$. Let $\mathcal{C}^{x,r} \subseteq \mathcal{X}^{x,r}$ be a RCI set and $\mathcal{C}^u((x, r))$ be the associated input admissible set for (4), (5), (6), (7), (9), robust to any $\gamma \in \mathcal{C}^\gamma(r) \subseteq \Gamma$. Let $\bar{k} \in \mathbb{Z}_{0+}$ be given such that $\bar{k} \le N$ and $r_{k-1|t+1} = r_{k|t}$ for $k = 1, \dots, \bar{k}$ for all $t \in \mathbb{Z}_{0+}$, R^N_t be generated by RGA according to Assumption 1, and $u_t = g_{\mathrm{RHC}}(x_t, R^N_t)$. If (17) is feasible at time $t \in \mathbb{Z}_{0+}$, then it is feasible for all $\bar{t} \in \mathbb{Z}_{0+}$, $\bar{t} \ge t$, and the closed-loop satisfies (5) and (9).

Proof: Since $\mathcal{C}^{x,r} \subseteq \mathcal{X}^{x,r}$ and $\mathcal{C}^u((x, r)) \neq \emptyset$ iff $(x, r) \in \mathcal{C}^{x,r}$, any feasible solution of (17) ensures that the closed-loop satisfies (5), (9). Next, we prove the persistent feasibility of (17), i.e., given the optimal solution $U^*_t = [u^*_{0|t}{}' \dots u^*_{N-1|t}{}']'$ at time $t \in \mathbb{Z}_{0+}$ for x_t and R^N_t, we show that at $t+1$ there exists a feasible solution $\tilde{U}_{t+1} = [\tilde{u}'_{0|t+1} \dots \tilde{u}'_{N-1|t+1}]'$. Consider first $\bar{k} \ge 1$. For $k = 1, \dots, \bar{k}-1$, let $\tilde{u}_{k-1|t+1} = u^*_{k|t}$. By the feasibility of U^*_t and since $r_{k-1|t+1} = r_{k|t}$ for $k = 1, \dots, \bar{k}$, this ensures satisfaction of the constraints for the first $\bar{k}$ steps, and by applying $\tilde{u}_{k-1|t+1} = u^*_{k|t}$ for $k = 1, \dots, \bar{k} - 1$ we obtain $x_{\bar{k}-1|t+1} = x_{\bar{k}|t}$. Hence, by the feasibility of U^*_t and since $\mathcal{C}^u((x, r)) \neq \emptyset$ iff $(x, r) \in \mathcal{C}^{x,r}$, $(x_{\bar{k}-1|t+1}, r_{\bar{k}-1|t+1}) \in \mathcal{C}^{x,r}$. Since $\mathcal{C}^{x,r}$ is RCI, there exists $\tilde{u}_{\bar{k}-1|t+1} \in \mathcal{C}^{x,r}(x_{\bar{k}-1|t+1}, r_{\bar{k}-1|t+1})$ such that $(x_{\bar{k}|t+1}, r_{\bar{k}|t+1}) \in \mathcal{C}^{x,r}$, for every admissible $r_{\bar{k}|t+1}$, because $\mathcal{C}^{x,r}$ is RCI for every $\gamma \in \Gamma$ and Assumption 1 guarantees that $\gamma_{\bar{k}-1|t+1} \in \mathcal{C}^\gamma(r_{\bar{k}-1|t+1}) \subseteq \Gamma$. The same argument can be repeated for constructing $\tilde{u}_{k|t+1}$, for $k = \bar{k}, \dots, N-1$, thus obtaining $\tilde{U}_{t+1}$.

For the case $\bar{k} = 0$, i.e., the entire reference preview is unreliable, (17h) is the only constraint in (17). Since $u_t = u^*_{0|t}$, $(x_{1|t}, r_{1|t}) \in \mathcal{C}^{x,r}$ for every $\gamma_{0|t} \in \Gamma \supseteq \mathcal{C}^\gamma(r_{0|t})$. By Assumption 1, $\gamma_t \in \mathcal{C}^\gamma(r_t) \subseteq \Gamma$ and hence $(x_{t+1}, r_{t+1}) \in \mathcal{C}^{x,r}$ and $\mathcal{C}^u(x_{t+1}, r_{t+1}) \neq \emptyset$. Thus, there exists $\tilde{u}_{0|t+1}$ for which $(x_{1|t+1}, r_{1|t+1}) \in \mathcal{C}^{x,r}$ for every $\gamma_{0|t+1} \in \Gamma \supseteq \mathcal{C}^\gamma(r_{0|t+1})$. Since the RGA computes $r_{k|t+1}$, $k = 0, \dots, N$ according to Assumption 1 and $\mathcal{C}^{x,r}$ is RCI for any $\gamma \in \Gamma \supseteq \mathcal{C}^\gamma(r)$, the reasoning can be repeated to construct $\tilde{U}_{t+1}$ such that $\tilde{u}_{k|t+1} \in \mathcal{C}(x_{k|t+1}, r_{k|t+1})$, and hence $\tilde{U}_{t+1}$ is feasible. ■

In Theorem 2 it is not necessary that $r_{k-1|t+1} = r_{k|t}$ for $k = \bar{k}+1, \dots, N$, as long as $\gamma_{k|t+1} \in \mathcal{C}^\gamma(r_{k|t+1})$. However $\bar{k} > 0$, i.e., some steps of reliable preview in (17), may increase the controller performance. Since $\mathcal{C}^u(x, r) \neq \emptyset$ iff $(x, r) \in \mathcal{C}^{x,r} \subseteq \mathcal{X}^{x,r}$ and $\mathcal{C}^u(x, r) \subseteq \mathcal{U}$, a larger $\bar{k}$, i.e., a larger number of reference steps fixed in advance, will usually provide a larger feasible set for (17) .

Remark 4: The results of Theorem 2 also hold for a controller selecting a feasible, yet non-optimal, solution of (17). □

The next corollary follows immediately from Theorem 2 and summarizes the proposed method as a solution for Problem 1.

Corollary 1: Under the assumptions of Theorem 2, $\mathcal{X}_0 = \mathcal{C}^{x,r}$ and $\kappa(x, R^N) = g_{\mathrm{RHC}}(x, R^N)$ solve Problem 1. □

In Corollary 1, $\mathcal{X}_0 = \mathcal{C}^{x,r}$, i.e., an RCI, guarantees initial feasibility, and recursive feasibility follows from Theorem 2. For $\kappa(x, R^N) = g_{\mathrm{RHC}}(x, R^N)$, $\mathcal{C}^{x,r}$ may not be the largest set for solving Problem 1, i.e., there may exist $\mathcal{X}_0 \supset \mathcal{C}^{x,r}$ that solves Problem 1, however $\mathcal{C}^{x,r}$ is polyhedral. Also, it may happen that $\mathrm{proj}_r(\mathcal{X}^{x,r}_0) \subset \mathcal{C}^r$. Thus, the RGA can initialize the reference state only in $\mathrm{proj}_r(\mathcal{X}^{x,r}_0)$. However, after initialization, the RGA can select any $\gamma \in \mathcal{C}^\gamma(r)$ and feasibility is preserved according to Theorem 2, because $\mathrm{proj}_r(\mathcal{X}^{x,r}_0)$ will be RCI for (6) for $\gamma \in \mathcal{C}^\gamma(r)$ as discussed in Remark 2.

Remark 5: Practical examples of stage cost $q(\cdot)$ are $q(\cdot) = (y_{k|t} - y^r_{k|t})' Q_y (y_{k|t} - y^r_{k|t}) + u'_{k|t} Q_u u_{k|t}$, or $q(\cdot) = (y_{k|t} - y^r_{k|t})' Q_y (y_{k|t}, y^r_{k|t}) + (u_{k|t} - u^r_{k|t})' Q_u (u_{k|t} - u^r_{k|t})$, where Q_y, Q_u are positive definite matrices, and $u^r_{k|t} = M^r_u y^r_{k|t}$ is the steady state input associated to $y^r_{k|t}$ by the inverse dc-gain matrix M^r_u. The satisfaction of (5), (9) is guaranteed by the RCI set, and hence any stage cost $q(\cdot)$ encoding additional control objectives, such as energy minimization and smooth motion, can be used. A nonnegative terminal cost $q_f(y, y^r, x)$ can also be included in (17).

V. Case Studies

Next we describe a numerical and an application case study.

A. Underdamped 2^{nd} order system with 2^{nd} order reference model

We consider the second order system

$$\dot{x}(t) = \begin{bmatrix} -2 & -2 \\ 2 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 4 \\ 0 \end{bmatrix} u(t) \quad (19a)$$

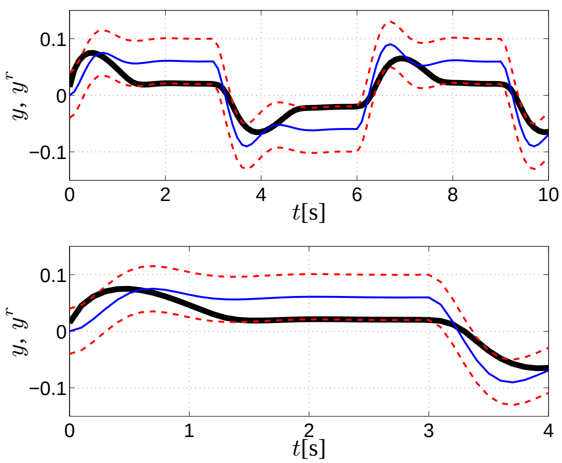
$$y(t) = \begin{bmatrix} 0 & 5 \end{bmatrix} x(t), \quad (19b)$$

subject to constraints $-0.1 \leq u(t) \leq 0.1$, $-\left[0.04\ 0.025\right]' \leq x(t) \leq \left[0.04\ 0.025\right]$. We sample (19) with period $T_s = 0.1$s to obtain (4). The reference model (6) is obtained by sampling with $T_s = 0.1$s

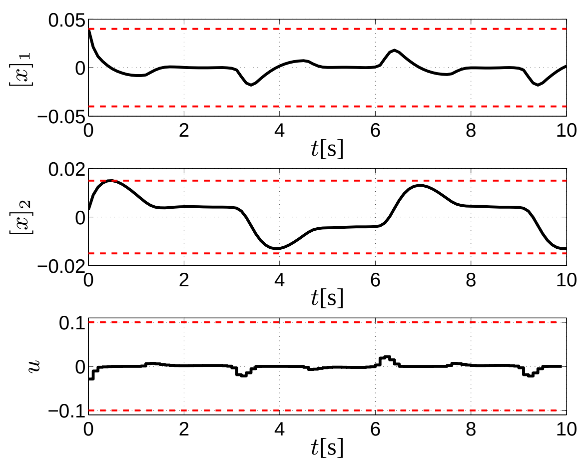
$$\dot{r}(t) = \begin{bmatrix} -4 & -6.25 \\ 4 & 0 \end{bmatrix} r(t) + \begin{bmatrix} 2 \\ 0 \end{bmatrix} \gamma(t) \quad (20a)$$

$$y^r(t) = \begin{bmatrix} 0 & 3.125 \end{bmatrix} r(t), \quad (20b)$$

where $\gamma \in \Gamma = \{\gamma \in \mathbb{R} :\ \gamma^- \leq \gamma \leq \gamma^+\}$, $\gamma^+ = -\gamma^- = 0.06$. For $\epsilon = 0.04$, we compute the RCI set $\mathcal{C}^{x,r}$ and the corresponding input admissible set $\mathcal{C}^u((x,r))$ using Algorithm 2, which converges after 12 iterations, and we design the MPC tracking controller (17), (18) with $\bar{k} = 0$, $q(y_{k|t}, y^r_{k|t}, x_{k|t}), u_{k|t}) = u^2_{k|t}$ and $N = 3$.



(a) Reference output (blue, thin), system output (black, thick) and tracking bounds (red, dash). Top: entire simulation. Bottom: initial part.



(b) System states and input (black), and constraints (red, dash).

Fig. 1. Second order system in closed-loop with the proposed controller, tracking a reference generated by a second order model.

Figure 1 depicts the closed loop system behavior when the initial state is $x_0 = [0.04\ 0.003]'$ for a reference profile which satisfies Assumption 1. Figure 1(a) shows that a non-trivial trajectory is obtained due to the conflicting objectives of minimizing input energy and enforcing the tracking error bound. As shown in Figure 1(b), the control algorithm enforces the state and input constraints and maintains the tracking error within the desired bound.

B. Control of a dual-stage dual-axis processing machine

We consider a dual-stage dual-axis processing machine, where a servo system actuates the main (slow) stage, while a much faster stage with limited operating range provides rapid and precise positioning of the work tool, see, e.g., [1]. The machine has to move the work tool along a given 2D target trajectory, where, due to the mechanical configuration, the tool position is the sum of the positions of the slow and fast stage, and the two axes are mechanically decoupled. Since the two stages have significant time scale separation (e.g., 3 orders of magnitude), we design a controller for the slow stage trajectory while assuming the fast stage to be infinitely fast. Thus, the controller has to maintain the difference between the slow stage trajectory and the target trajectory within the range of the fast stage, while enforcing the operating constraints and possibly minimizing the torque of the slow stage to minimize energy consumption and induced vibrations.

The dynamics of the main stage for each axis $i \in \{\mathbf{x}, \mathbf{y}\}$ are

$$\dot{x}_i(t) = \begin{bmatrix} 0 & 1 \\ 0 & a_i \end{bmatrix} x_i(t) + \begin{bmatrix} 0 \\ b_i \end{bmatrix} u_i(t), \quad (21a)$$

$$y_i(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x_i(t), \quad (21b)$$

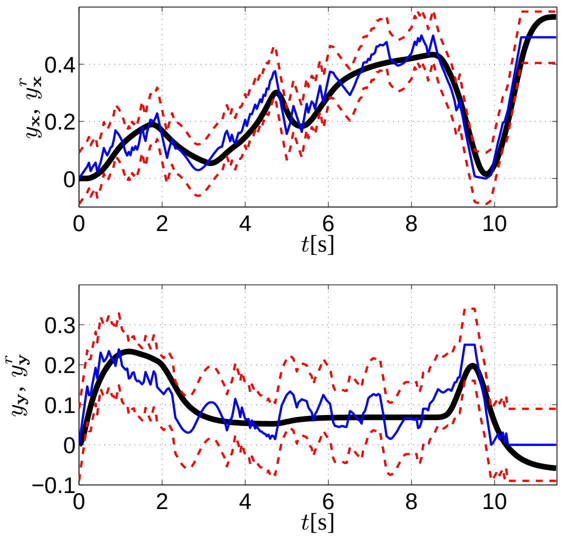
where, $[x_i]_1$[m] is the position, $[x_i]_2$[m/s] is the velocity, u_i[Nm] is the input torque, $i \in \{\mathbf{x}, \mathbf{y}\}$ is the axis index, and $(a_{\mathbf{x}}, b_{\mathbf{x}}) = (-1.08, 0.34)$, $(a_{\mathbf{y}}, b_{\mathbf{y}}) = (-2.29, 0.73)$. Systems (21) are subject to $-\bar{u}_i \leq u_i(t) \leq \bar{u}_i$, $-\left[1\ 0.8\right]' \leq x_i(t) \leq \left[1\ 0.8\right]'$, $i \in \{\mathbf{x}, \mathbf{y}\}$, where $\bar{u}_{\mathbf{x}} = 35$Nm and $\bar{u}_{\mathbf{y}} = 17.5$Nm. The discrete time model (4) is obtained by sampling (21) with period $T_s = 0.04$s, while the reference signal is obtained from an RGA that converts a 2D work path into a reference trajectory satisfying $\gamma_t \in \mathcal{C}^\gamma(r_t)$ for (6), (7), with $A^r = 1$, $B^r = 1$, $C^r = 1$ and $D^r = 0$, $\Gamma = \{\gamma \in \mathbb{R} :\ \gamma^- \leq \gamma \leq \gamma^+\}$, $\gamma^+ = -\gamma^- = 0.033$. The tracking error bound is the range of the fast actuator, $\epsilon = 0.09$. We compute the $\mathcal{C}^{x,r}$ using Algorithm 2, which converged after 8 iterations, for both axes[2].

We designed the MPC tracking controller (17), (18) with $q(y_{k|t}, y^r_{k|t}, x_{k|t}), u_{k|t}) = u^2_{k|t}$, $N = 10$ (i.e., 0.4s), and $\bar{k} = N$. Thus, there is perfect preview along the entire prediction horizon and the RCI set guarantees persistent feasibility when the horizon recedes from step to step. Figure 2 shows that despite the "rich" time-varying nature of the reference signal, the tracking error is within the desired bound at every time step, and the input and state constraints are always satisfied, for both axes.

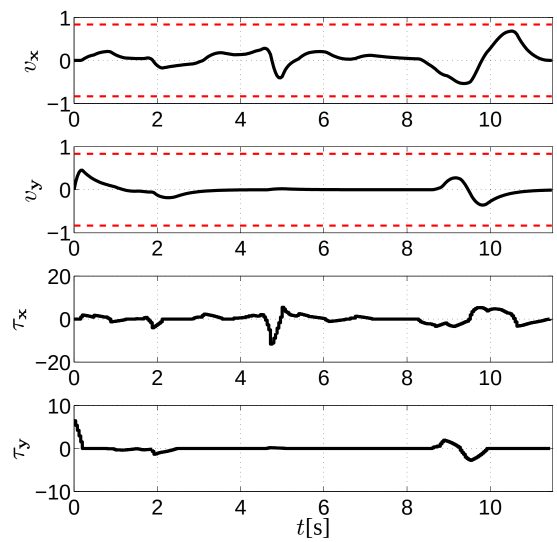
VI. Conclusions

We have proposed a tracking controller for constrained linear systems that guarantee satisfaction of system constraints and a given error bound on the tracking error, for all the reference trajectories generated by a constrained linear system driven by a bounded input. The control design is based on MPC enforcing a polyhedral RCI set computed by a specific algorithm, which guarantees the persistent feasibility of the optimal control problem, and hence the satisfaction of the system constraints and of the tracking bound. Future research will focus on design trade-offs such as the volume of the RCI set as function of the error bound and the reference model input range.

[2]In the case studies Algorithm 1 by [21] did not converge in 24 hours.



(a) Reference position (blue, thin), position (black, thick) and tracking bounds (red, dash). Top: x-axis. Bottom: y-axis.



(b) Velocities v_i and torques τ_i, $i \in \{\mathbf{x}, \mathbf{y}\}$, for x-axis and y-axis (black), and constraints (red, dash).

Fig. 2. Simulations of the dual-stage dual-axis processing machine.

References

[1] S. Haghighat, S. Di Cairano, D. Konobrytskyi, and S. Bortoff, "Coordinated control of a dual-stage positioning system using constrained model predictive control," in *Proc. ASME Dynamic Sys. Control Conf.*, 2014.

[2] S. Di Cairano, J. Doering, I. Kolmanovsky, and D. Hrovat, "Model predictive control of engine speed during vehicle deceleration," *IEEE Trans. Contr. Systems Technology*, vol. 22, no. 6, pp. 2205–2217, 2014.

[3] S. Di Cairano, W. Liang, I. Kolmanovsky, M. Kuang, and A. Phillips, "Power smoothing energy management and its application to a series hybrid powertrain," *IEEE Trans. Contr. Systems Technology*, vol. 21, no. 6, pp. 2091–2103, 2013.

[4] S. Di Cairano, H. Tseng, D. Bernardini, and A. Bemporad, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," *IEEE Trans. Contr. Systems Technology*, vol. 21, no. 4, pp. 1236–1248, 2013.

[5] S. Di Cairano and I. V. Kolmanovsky, "Constrained actuator coordination by virtual state governing," in *Proc. 50th IEEE Conf. on Decision and Control*, 2011, pp. 5491–5496.

[6] T. A. Badgwell and K. R. Muske, "Disturbance model design for linear model predictive control," in *Proc. American Contr. Conf.*, Anchorage, AK, 2002, pp. 1621–1626.

[7] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model predictive control," *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[8] E. Zafiriou and M. Morari, "A general controller synthesis methodology based on the IMC structure and the H_2, H_∞ and μ-optimal control theories," *Computers & Chemical Eng.*, vol. 12, no. 7, pp. 757–765, 1988.

[9] L. Magni, G. D. Nicolao, and R. Scattolini, "Output feedback and tracking of nonlinear systems with model predictive control," *Automatica*, vol. 37, no. 10, pp. 1601–1607, 2001.

[10] D. Limon, I. Alvarado, T. Alamo, and E. Camacho, "MPC for tracking piecewise constant references for constrained linear systems," *Automatica*, vol. 44, no. 9, pp. 2382 – 2387, 2008.

[11] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E. Camacho, "MPC for tracking of constrained nonlinear systems," in *Proc. 48th IEEE Conf. on Decision and Control*, Shangai, China, 2009, pp. 7978–7983.

[12] U. Maeder and M. Morari, "Offset-free reference tracking with model predictive control," *Automatica*, vol. 46, no. 9, pp. 1469–1476, 2010.

[13] A. Bemporad, A. Casavola, and E. Mosca, "Nonlinear control of constrained linear systems via predictive reference management," *IEEE Trans. Automatic Control*, vol. 42, no. 3, pp. 340–349, 1997.

[14] E. Gilbert and I. Kolmanovsky, "Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor," *Automatica*, vol. 38, no. 12, pp. 2063–2073, 2002.

[15] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill, Madison, Wisconsin, 2009.

[16] P. Falugi and D. Q. Mayne, "Tracking performance of Model Predictive Control," in *Proc. 51st IEEE Conf. on Decision and Control*, Maui, HI, 2012, pp. 2631 – 2636.

[17] P. Falugi, "Model predictive control for tracking randomly varying references," *Int. J. Control*, vol. 88, no. 4, pp. 745–753, 2015.

[18] S. Di Cairano and F. Borrelli, "Constrained tracking with guaranteed error bounds," in *Proc. 52nd IEEE Conf. on Decision and Control*, Florence, Italy, 2013, pp. 3800–3805.

[19] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer Science & Business Media, 2007.

[20] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Mathematical Problems in Egineering*, vol. 4, pp. 317–367, 1998.

[21] S. Raković, E. Kerrigan, D. Mayne, and J. Lygeros, "Reachability analysis of discrete-time systems with disturbances," *IEEE Trans. Automatic Control*, vol. 51, no. 4, pp. 546–561, 2006.

[22] M. Barić, "Constrained Control: Computations, Performance and Robustness," Ph.D. dissertation, ETH Zürich, 2008.