Predictive tracking control of constrained nonlinear systems

L. Chisci, P. Falugi and G. Zappa

Dipartimento di Sistemi e Informatica
Università di Firenze
via Santa Marta 3, 50139 Firenze, Italy
e-mail: chisci,falugi,zappa@dsi.unifi.it

Abstract

Combining predictive control, LPV (Linear Parameter Varying) embedding and gain-scheduling ideas, new computationally efficient algorithms for tracking control of constrained nonlinear systems are proposed. Simulation experiments demonstrate the good tracking properties of such algorithms.

1 Introduction

Tracking control of constrained nonlinear systems is a challenging problem which has recently attracted considerable attention. In particular, a successful approach is based on the so called Reference Governor (RG) [1]-[4]. The RG is essentially a nonlinear device which manipulates on-line a command input to the suitably pre-compensated closed-loop system so as to impose constraint satisfaction. In [5], an alternative approach called Dual Mode (DM) predictive tracking has been proposed. The DM predictive controller oper-
ates as a normal regulator in a suitable neighborhood of the desired equilibrium, wherein constraints are feasible, while aims at recovering feasibility as quickly as possible whenever this is lost due to a set-point change. In the feasibility recovery mode, the controller directly synthesizes the plant control input and, hence, has more freedom than the RG which can only synthesize a command input to a pre-compensated closed-loop. In fact, simulation results demonstrated, for linear systems, the superior tracking speed of the DM controller over the RG. Another common approach to tracking control of nonlinear systems is Gain-Scheduling (GS) [6]-[9]. In particular [8] presented a GS control design based on set-valued methods which, unlike heuristic GS techniques, provides guarantees of stability and constraint satisfaction. This approach relies on embedding the original nonlinear system into an LPV (Linear Parameter Varying) model wherein the “varying parameter” is interpreted as a discrete scheduling variable. To this end, the continuous parameter space is partitioned into a finite number of disjoint regions, a discrete parameter is assigned to each region, and the discrete parameter time-evolution is modeled via a “diffusive” dynamics which allows the parameter to take all possible discrete values in finite time. The present paper improves the existing work on tracking control of nonlinear systems in several directions. First, LPV embedding techniques are used in order to design a GS pre-compensator and to compute off-line a family of admissible sets, parametrized by the set-point, which are useful for on-line reference management. This gives a systematic way to obtain the level sets required for the implementation of the generalized RG in [4]. Secondly, the DM approach of [5] is extended to nonlinear systems thus providing an enhancement of tracking speed with respect to the RG. Finally, an overlapping partition of the parameter space along with a “constant” parameter dynamics are used in place of the disjoint partition and “diffusive” dynamics adopted in [6, 7], implying a further significant improvement of tracking speed.
The paper is organized as follows. Section 2 formulates the tracking control problem of interest. Section 3 describes the embedding of the nonlinear system into an LPV model. Section 4 describes the proposed tracking algorithms and analyzes their properties. Section 5 compares RG and DM strategies under “diffusive” and “constant” parameter dynamics, by means of a simulation example. Finally Section 6 draws some conclusions.

2 Tracking control problem formulation

Consider a continuous-time nonlinear system

\begin{align*}
\dot{x}(t) &= f(x(t), u(t)) \\
y(t) &= h(x(t))
\end{align*}

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^p$, $y(t) \in \mathbb{R}^p$. The control objective is that

1. the output $y(t)$ track a piecewise constant reference $r(t)$, i.e. a signal switching among different constant set-points;

2. the state $x(t)$ and input $u(t)$ satisfy the linear inequality constraints

\begin{equation}
L x(t) + M u(t) \leq b
\end{equation}

It is assumed that to each constant set-point $r$ there is associated an unique (state,input) equilibrium pair $(x_{eq}(r), u_{eq}(r))$ such that

\begin{equation}
0 = f(x_{eq}(r), u_{eq}(r)), \quad r = h(x_{eq}(r))
\end{equation}

Clearly the constraints (2) restrict the statically admissible set-points $r$ to those which satisfy $L x_{eq}(r) + M u_{eq}(r) \leq b$. In order to ensure viability in finite time from one set-point to another [3, 4], the reference $r(t)$ is further restricted to belong to the set

\[ R_\delta = \{ r : L x_{eq}(r) + M u_{eq}(r) \leq b - \delta 1 \}, \]

where $1$ is a vector of ones and $\delta > 0$ an arbitrarily small number.
3 From “nonlinear” to “LPV” model

The approach pursued in this paper will make use of constraint-admissible invariant sets [10]. In this respect, note that nonlinearity of (1) makes difficult the construction of such sets. A possible approach to avoid this difficulty is to embed (1) into an LPV model. There are several ways to get an LPV model of a nonlinear plant. Here we shall focus our attention on a special class of systems [8] which admit a suitably simple LPV representation with useful properties for the subsequent developments. Let us therefore assume the system (1) to be of the form

\[ f(x, u) = \phi(x_1) + \tilde{A}(x_1)x + \tilde{B}u \]

\[ h(x) = x_1 \]

where the state vector is partitioned as \( x = [x'_1, x'_2]' \), \( x_1 \) being the controlled output. According to (3) the equilibria of (4) can be parametrized by the scheduling variable \( y = x_1 \), i.e for a given value of \( x_1 \) there exist functions \( u_{eq}(x_1) \) and \( x_{2,eq}(x_1) \) such that

\[ \phi(x_1) + \tilde{A}(x_1) \begin{bmatrix} x_1 \\ x_{2,eq}(x_1) \end{bmatrix} + \tilde{B}u_{eq}(x_1) = 0 \]

Then, applying as in [8] the following nonlinear change of coordinates (parametrized by \( r \))

\[ \tilde{x} = \begin{bmatrix} x_1 - r \\ x_2 - x_{2,eq}(x_1) \end{bmatrix}, \quad \tilde{u} = u - u_{eq}(x_1) \]

we obtain

\[ \dot{\tilde{x}}(t) = A_c(x_1(t))\tilde{x}(t) + B_c(x_1(t))\tilde{u}(t) \]

where

\[ A_c(x_1) = \begin{bmatrix} 0 & \tilde{A}_{12}(x_1) \\ 0 & \tilde{A}_{22}(x_1) - \frac{\partial x_{eq}(x_1)}{\partial x_1} \tilde{A}_{12}(x_1) \end{bmatrix}, \quad B_c(x_1) = \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 - \frac{\partial x_{eq}(x_1)}{\partial x_1} \tilde{B}_1 \end{bmatrix} \]
\( \hat{A}_{ij}(x_1) \) and \( \hat{B}_i \) being the partitioned blocks of \( \hat{A}(x_1) \) and, respectively, \( \hat{B} \). Notice that the model (6) enjoys the following useful properties:

1. it is quasi-linear in \( \tilde{x}_1 = x_1 - r \) and linear in \( \tilde{x}_2 \);
2. the matrices \( A_c(\cdot) \) and \( B_c(\cdot) \) only depend on the controlled output \( y = x_1 \);
3. the state variable \( \tilde{x} \) depends in a simple and linear way on the set-point \( r \).

Finally, we apply Euler discretization with a sampling time \( T_s \) in order to bring the system into discrete-time form, convenient for digital control implementation,

\[
\tilde{x}(t+1) = (I + T_sA_c(x_1(t)))\tilde{x}(t) + T_sB_c(x_1(t))\tilde{u}(t) = A(x_1(t))\tilde{x}(t) + B(x_1(t))\tilde{u}(t) \tag{7}
\]

In order to embed the quasi-linear dynamics (7) into an LPV model, it is assumed, without loss of generality, that the scheduling variable \( x_1 \) evolves in a compact set \( P \) which has a finite, possibly overlapping, partition

\[
P = \bigcup_{i=1}^{\ell} P_i \tag{8}
\]

which induces a corresponding partition of the state space into the regions \( X_i = \{ x : x_1 \in P_i \} \).

Denote by \( R_{\delta}^i = \{ r \in R_{\delta} \mid r + v \in P_i, \forall v : \| v \| \leq \delta \} \). Moreover let \( \mathcal{I} \triangleq \{ 1, 2, \ldots, \ell \} \) be the index set and \( \mathcal{I}(x) = \{ i \mid x_1 \in P_i \} \) the index subset indicating the regions to which \( x_1 \) belongs. For each region \( X_i \), the quasi-linear dynamics (7) is embedded into a linear polytopic dynamics

\[
\tilde{x}(t+1) \in \mathcal{F}(i(t)) \left[ \begin{array}{c} \tilde{x}(t) \\ \tilde{u}(t) \end{array} \right]
\]

i.e. \( \mathcal{F}(i) \) is a polytope of matrices in \( \mathbb{R}^{n \times (n+p)} \) such that \( [A(x_1), B(x_1)] \in \mathcal{F}(i) \) if \( x_1 \in P_i \).

Let us introduce a set-valued dynamics for the index \( i \)

\[
i(t+1) \in \mathcal{Q}(i(t)) \tag{9}
\]

in such a way that

\[
\mathcal{I}(x(t+1)) \subset \mathcal{Q}(i) \quad \text{if} \quad x(t) \in X_i \tag{10}
\]
Remark - Notice that (10) implies restrictions on \( x(t + 1) \) and, hence, an additional constraint on the control action \( \tilde{u}(t) \). In particular two kinds of dynamics will be adopted in the subsequent developments.

- **Diffusive dynamics**: It is assumed that each index \( j \) is reachable from any index \( i \) by iterating the dynamics \( Q \), i.e. the graph representing the dynamics \( Q \) is strongly connected.

- **Constant dynamics**: It is assumed that \( Q(i) = \{i\} \) and that for each pair of indices \( i, j \) there exists a sequence of sets \( \{P_i, \cdots, P_j\} \) such that any two consecutive elements of this sequence have intersection with nonempty interior.

Remark - To clarify the above definitions consider the SISO case \((p = 1)\) in which the sets \( P_i \) \((i = 1, 2, \cdots, \ell)\) are usually ordered intervals. In this case, a typical diffusive map is \( Q(i) = \{i - 1, i, i + 1\} \) and for the constant dynamics the necessary assumption is \( P_i \cap P_{i+1} \neq \emptyset \).

In the case of diffusive dynamics, the constraint (10) is guaranteed by a rate constraint of the form \( \|x_1(t+1) - x_1(t)\|_{\infty} \leq \delta y \) for an appropriate \( \delta y > 0 \), i.e.

\[
\|T_s A_{c1}(x_1(t))\tilde{x}(t) + T_s B_{c1}(x_1(t))\tilde{u}(t)\|_{\infty} \leq \delta y \tag{11}
\]

where \( A_{c1}, B_{c1} \) are the upper submatrices of \( A_c, B_c \) corresponding to the partitioning \( x = [x'_1, x'_2]' \). Conversely, in the case of constant dynamics, (10) is guaranteed by the invariance constraint \( x_1(t+1) \in P_i \), i.e.

\[
\tilde{x}_1(t) + T_s A_{c1}(x_1(t))\tilde{x}(t) + T_s B_{c1}(x_1(t))\tilde{u}(t) + r \in P_i \tag{12}
\]

The key idea underlying the constant dynamics approach is the use of overlapping invariant sets in order to allow fast tracking for the nonlinear system. A similar idea has been
adopted in [11, 12] in order to enlarge the domain of attraction for nonlinear stabilization.

**Remark** - It is possible to consider intermediate configurations between the diffusive and the constant index dynamics, namely configurations wherein the graph associated to \( Q \) is composed by a certain number of disjoint strongly connected components; however these have not been investigated in the present paper.

Summing up, the nonlinear dynamics (7) can be embedded into the LPV model

\[
\begin{cases}
i(t+1) \in Q(i(t)) \\
\bar{x}(t+1) \in F(i(t)) \begin{bmatrix} \bar{x}(t) \\ \bar{u}(t) \end{bmatrix}
\end{cases}
\]

(13)

subject to constraints (2) and (11) or (12).

**Remark** - The model (13) can be used to *safely* predict the behaviour of the system (7) in the sense that trajectories of (7) are also trajectories of (13) (the viceversa is not true, which makes the embedding *conservative* to some extent). In particular, each polytopic submodel \( F(i) \) is used to predict the future state \( x(t+1) \) whenever the scheduling variable \( x_1(t) \) belongs to a suitable region \( P_i \). Clearly, since the regions \( P_i \) need not be disjoint, all polytopic models \( F(i), i \in I(x(t)) \), are valid for a given \( x(t) \).

4 Tracking control algorithms

The LPV model (13) constitutes the basis for the design of a gain-scheduling controller

\[
\bar{u}(t) = F_{i(t)} \: \bar{x}(t), \quad i(t) \in I(x(t))
\]

(14)
providing asymptotic tracking and constraint satisfaction for the system (1). In fact, a sufficient condition to guarantee such properties for the system (1) is to guarantee the same properties for (13) provided that a sufficiently small sampling time \( T_s \) is selected [13]. The design of (14) is naturally accomplished by resorting to the theory of admissible sets [10, 14]. Consider the closed-loop system obtained by the feedback connection of (13) and (14)

\[
\begin{align*}
\begin{cases}
i(t+1) & \in Q(i(t)) \\
\tilde{x}(t+1) & \in F(i(t))
\end{cases}
\end{align*}
\]  

(15)

and replace the constraints (2) and (11) or (12) with the linear constraints

\[
L_i(t)\tilde{x}(t) + M_i(t)\bar{u}(t) + H_i(t)r \leq b_i(t)
\]

(16)

**Remark** - Note that by virtue of (5), the linear constraints (2) become nonlinear in \( x_1 \) like the constraints (11) and (12). In order to construct admissible sets, the linear constraints with \( i \)-dependent matrices in (16) are obtained combining (2), (5) and (11) or (12) after a local linearization with respect to \( x_1 \in P_i \).

**Definition 1** - Given \( \mu, 0 < \mu < 1 \), a set \( \Sigma \subset I \times \mathbb{R}^n \times \mathbb{R}^p \) is said \( \mu \)-contractive under the closed-loop dynamics (15) if

\[
\begin{align*}
\begin{bmatrix}
i(t) \\
\tilde{x}(t) \\
r
\end{bmatrix} & \in \Sigma \implies \\
\begin{bmatrix}
i(t+1) \\
\mu^{-1}\tilde{x}(t+1) \\
r
\end{bmatrix} & \in \Sigma
\end{align*}
\]  

(17)
If in addition

\[
\begin{bmatrix}
i \\
\tilde{x} \\
r
\end{bmatrix} \in \Sigma \implies (L_i + M_i F_i) \tilde{x} + H_i r \leq b_i, \quad r \in R_{\delta} \text{ (diffusive map)} \text{ or } r \in R_{\delta}^i \text{ (constant map)}
\]

(18)

\( \Sigma \) is said \textit{constraint admissible}

Notice that considering invariant sets in the extended space allows to compactly represent invariant sets in the state space for any values of \( i \) and \( r \). Exploiting techniques similar to the ones described in [10], it is possible to iteratively construct a constraint-admissible \( \mu \)-contractive set. To be more specific, a decreasing sequence of sets \( \Sigma_k \) is computed starting from the set \( \Sigma_0 \) of all elements \((i, \tilde{x}, r)\) that satisfy (18). If at some iteration \( k \), the interior of \( \Sigma_k \) becomes empty, the closed-loop system (15) is not exponentially stable with convergence rate \( \mu \); this means that \( \mu \) must be increased and/or the gains \( F_i \) must be changed. Otherwise, the construction procedure terminates after a finite number of iterations as soon as \( \Sigma_k = \Sigma_{k+1} \) yielding the largest constraint-admissible \( \mu \)-contractive set \( \Sigma_{\text{max}} = \Sigma_k \). For the detailed construction procedure, the reader is referred to [9, 15]. It is important to highlight the structure and the properties of \( \Sigma_{\text{max}} \). \( \Sigma_{\text{max}} \) is a collection of sets \( \Sigma_{\text{max}}^i \subset \mathbb{R}^{n+p}, \ i = 1, \ldots, \ell \), which, due to linearity of (13) and (16), are clearly polytopic (i.e. represented by a finite number of linear inequalities). Each \( \Sigma_{\text{max}}^i \) can be sliced for a given \( r \), yielding

\[
\Sigma_{\text{max}}^i (r) = \left\{ \tilde{x} : \begin{bmatrix} \tilde{x} \\ r \end{bmatrix} \in \Sigma_{\text{max}}^i \right\}
\]

(19)

\textit{Definition 2} - A state \( x \) is feasible for a set-point \( r \) or equivalently \( r \) is admissible for
the state $x$ if
\[
\begin{bmatrix}
  x_1 - r \\
  \bar{x}_2
\end{bmatrix}
\in \Sigma_{max}(r)
\]
for some $i \in \mathcal{I}(x)$.

The following theorem holds.

**Theorem 1** - If $x$ is feasible for $r$, then the gain-scheduling control law $u = u_{eq}(x_1) + F_i \begin{bmatrix}
  x_1 - r \\
  \bar{x}_2
\end{bmatrix}$, where $\bar{x} \in \Sigma_i(r)$, guarantees that, starting from $x(0) = x$, the constraints (2) are satisfied for all $t \geq 0$ and $x(t)$ converges to $x_{eq}(r) = [r', x_{2,eq}(r')]'$.

Hereafter, the set $\Sigma_{max}$ will be more simply denoted by $\Sigma$, and $\psi(\Sigma_i(r))$ will denote the image of $\Sigma_i(r)$ in the original coordinate space. Notice that, due to the nonlinear change of coordinates, $\psi(\Sigma_i(r))$ is not a polytope. The following facts can be easily proved.

1. Assume that $Q$ is diffusive. Then
   - $\Sigma_i(r)$ contains the origin $\bar{x} = 0$, for any $i \in \mathcal{I}$ and any $r \in R_\delta$, this means that $\psi(\Sigma_i(r))$ contains $x_{eq}(r)$.
   - If $\Sigma_i(r)$ has non empty interior, then $\Sigma_j(r)$ has non empty interior for all $j \in \mathcal{I}$.
   - If $\Sigma_i(r)$ has non empty interior, then $\Sigma_i(r')$ has non empty interior for all $r' \in R_\delta$.

2. Conversely assume that $Q$ is constant. Then
   - $\Sigma_i(r)$ is non empty and contains the origin only for $r \in R_{\delta_i}$.
   - $\psi(\Sigma_i(r)) \subset X_i$. 

10
• If $\Sigma^i(r)$ has non empty interior, then $\Sigma^i(r')$ has non empty interior for all $r' \in R^i_\delta$.

It is clear that by moving the state $x(t)$ in a safe tube around the equilibrium characteristic $x_{eq}(r)$, with an input belonging to an allowable tube around $u_{eq}(r)$, one can achieve asymptotic tracking for any $r \in R_\delta$. The tracking speed clearly increases with the width of the tube. Since in our case, the tube is provided by the sets $\psi(\Sigma^i(r))$, their size is relevant for the tracking speed. Therefore the computation of $\Sigma$ has two important aims.

(1) First, it allows to check whether (14) ensures constrained stabilization; hence it’s an useful tool for trial-and-error design of a gain-scheduling controller.

(2) Secondly the polytope $\Sigma$ carries valuable information for the design of an effective control law which, on one hand, avoids constraint violations and, on the other hand, yields as fast as possible tracking.

A simple tracking strategy can be obtained by adapting the reference governor policy to the present setup. This yields the following algorithm.

**Reference Governor Gain-Scheduling Tracking (RGGST) algorithm** - At time $t$, given the state $x(t)$ and the desired reference $r(t) = r$, let $\mathcal{I}(t) \overset{\Delta}{=} \mathcal{I}(x(t)) = \{i : x(t) \in X_i\}$ and assume that a set-point $\bar{r}(t)$ is admissible for $x(t)$. Then:

(1) **Solve** the optimization problem

$$
\begin{bmatrix}
i(t)
\lambda(t)
\end{bmatrix} = \arg \min_{i \in \mathcal{I}(t), \lambda} \lambda \quad \text{subject to} \quad \begin{bmatrix}
x_1(t) - \bar{r}
x_2(t)
\end{bmatrix} \in \Sigma^i(r), \quad \begin{bmatrix}
x_1(t) - \bar{r}
\end{bmatrix} 
$$

(2) **Set** $\bar{r}(t+1) = r + \lambda(t)[\bar{r}(t) - r]$
(3) Set $u(t) = F_i[x'_1(t) - \bar{r}'(t + 1), \bar{x}'_2(t)] + u_{eq}(x_1(t))$

The rationale of the above algorithm is, provided that $x(t)$ is feasible for $r(t)$, to investigate if $x(t)$ is feasible for a reference $\bar{r}(t + 1)$ which is closer to the desired set-point $r$. Notice that the optimization (20) amounts to linear programming problems, one for each value of $i \in \mathcal{I}(t)$, in the single scalar variable $\lambda$.

A faster tracking strategy can be obtained by exploiting additional degrees of freedom and predictive control ideas. This strategy, referred to as Dual Mode (DM) predictive tracking, consists of two different modes of operation.

- **Regulation Mode** - If the current state is feasible for the desired set-point, use the gain-scheduling feedback control law (14).

- **Feasibility Recovery Mode** - If conversely the current state is infeasible for the desired set-point, choose an input that will make the future state feasible for a set-point as close as possible to the desired one.

This Dual Mode tracking strategy is formalized by the following algorithm.

**Dual Mode Gain-Scheduling Tracking (DMGST) algorithm** - At time $t$, given the state $x(t)$ and the desired reference $r(t) = r$, assume that a set-point $\bar{r}(t)$ is admissible for $x(t)$. Let $\Sigma^{\mathcal{I}(t)}(r) \triangleq \bigcup_{i \in \mathcal{I}(t)} \Sigma^i(r)$. Then the DMGST algorithm operates in dual-mode as follows.

- **Regulation Mode**

  If

  $\begin{bmatrix} x_1(t) - r \\ \bar{x}_2(t) \end{bmatrix} \in \Sigma^{\mathcal{I}(t)}(r)$ then

  $u(t) = u_{eq}(x_1(t)) + F_i \begin{bmatrix} x_1(t) - r \\ \bar{x}_2(t) \end{bmatrix}$

  for some $i \in \mathcal{I}(t)$
• Feasibility Recovery Mode

If \[
\begin{bmatrix}
x_1(t) - r \\
\tilde{x}_2(t)
\end{bmatrix}
\notin \Sigma^{I(t)}(r)
\]
then

\[
\begin{bmatrix}
j(t) \\
\lambda(t) \\
\tilde{u}(t)
\end{bmatrix} = \arg \min_{j \in \mathbb{I}, \lambda, \tilde{u}} \lambda \text{ subject to }
\begin{align*}
0 &\leq \lambda \leq 1 \\
Lx(t) + Mu_{eq}(x_1(t)) + M\tilde{u} &\leq b \\
\tilde{x}(t + 1) &\in \Sigma^j \\
\tilde{x}(t + 1) = A(x_1(t)) \begin{bmatrix}
x_1(t) - r \\
\tilde{x}_2(t)
\end{bmatrix} + B(x_1(t))\tilde{u} \\
\tau = r + \lambda[\tau(t) - r]
\end{align*}
\]

Set \(\tau(t + 1) = \lambda(t)\tau(t) + (1 - \lambda(t))r\)

Set \(u(t) =
\begin{cases}
F_k \begin{bmatrix}
x_1(t) - \tau(t) \\
\tilde{x}_2(t)
\end{bmatrix} + u_{eq}(x_1(t)) &\text{for } k \in I(t) : \begin{bmatrix}
x_1(t) - \tau(t) \\
\tilde{x}_2(t)
\end{bmatrix} \in \Sigma^k(\tau(t)), \text{ if } \lambda(t) = 1 \\
\tilde{u}(t) + u_{eq}(x_1(t)) &\text{if } \lambda(t) < 1
\end{cases}\)

Remarks

• The above DMGST algorithm is a non trivial extension to nonlinear systems of the strategy proposed in [5] for linear systems.

• The DMGST approach requires the determination of \(\Sigma\) which certainly is a computationally expensive task but, luckily, can be carried out off-line. As far as on-line computation is concerned, the most expensive part is the solution of the optimization problem (21). Note that (21) amounts to a number of linear programming
problems, one for each value of $i$, in $p + 1$ scalar variables $\lambda \in [0, 1]$ and $\tilde{u} \in \mathbb{R}^p$.

- The Dual-Mode approach of this paper differs from the Reference Governor (RG) approach [3]. In fact, to recover feasibility the DMGST algorithm fully exploits the plant control input $u$ as degree of freedom, while the RG uses a command input $\tilde{r}$.

- When $Q(i) = i$, the transition of the state $x(t)$ from $\Sigma^i(r)$ to $\Sigma^j(r)$ can occur only if the regions $X_i$ and $X_j$ have non void intersection and $r \in R^i_\delta \cap R^j_\delta$.

- The use of the linear GS feedback when $\lambda(t) = 1$ (no reference improvement) is just a technical condition which ensures finite recovery time (see the theorem below and its proof in the appendix). In practice the condition $\lambda(t) = 1$ is very unlikely to occur.

- Notice that the linear constraints (2) become nonlinear with respect to $\tilde{x}_1$ after the change of coordinates (5), and therefore must be linearized as in (11) in order to construct the invariant sets $\Sigma^i(r)$. However, in the control optimization step, where $\tilde{x}$ is fixed and $\tilde{u}$ only must be selected, the constraints (2) are linear with respect to $\tilde{u}$ and there is no need of linearization.

- Notice that, in the feasibility recovery mode, the DMGST algorithm operates as a predictive controller with prediction horizon (equal to the control horizon) equal to one. It is possible, in principle, to consider a prediction horizon $N > 1$ by imposing $\tilde{x}(t + N) \in \Sigma^j(\tilde{r})$, $j \in \mathcal{I}$, together with input and state constraints along the prediction horizon. However, due to the nonlinearity of the model, the convexity of the optimization problem is lost. In alternative, one could use the (uncertain) LPV model for prediction in order to keep linearity of the optimization problem at the price of an exponential growth of the number of constraints and a higher degree of conservatism.
• It is worth pointing out that other, more generic, classes of nonlinear systems can be embedded into the LPV model (13), using different linearization approaches [8], and then the proposed techniques are still applicable. However, in this general case, \( x_{eq}(r) \) and \( u_{eq}(r) \) may depend nonlinearly on \( r \) and, as a consequence, the DMGST algorithm may require a nonlinear, possibly non convex, optimization problem. Moreover, in general, the LPV description can be more complicated since the polytopic embedding may not only depend on the scheduling variable.

The proposed DMGST algorithm enjoys the following property.

**Theorem 2** - If \( x(0) \) is feasible for some \( r(0) \in R_\delta \) and \( r(t) = r \in R_\delta \) for all \( t \geq 0 \), the DMGST algorithm guarantees a *finite recovery time* (FRT) i.e. the existence of \( \bar{t} < \infty \) such that \( x(\bar{t}) \) is feasible for the desired set-point \( r \). Further, under the same conditions, the constraints (2) are satisfied for all \( t \geq 0 \) and the system asymptotically reaches the desired equilibrium i.e. \( \lim_{t \to \infty} x(t) = x_{eq}(r), \lim_{t \to \infty} u(t) = u_{eq}(r) \) and \( \lim_{t \to \infty} y(t) = r \).

*Proof* - see the Appendix.

**5 Simulation example**

In this section, we consider the application of our approach to the strongly nonlinear model of a *continuous stirred tank reactor* (CSTR) [16]. Assuming constant liquid volume, the CSTR for an exothermic, irreversible reaction, \( A \rightarrow B \), is described by the following model

\[
\dot{T} = \frac{q}{V}(T_f - T) - \frac{\Delta H}{\rho c_p} k_a e^{-\frac{E}{RT}} C_A + \frac{UA}{V \rho c_p} (T_c - T)
\]

\[
\dot{C_A} = \frac{q}{V}(C_{AF} - C_A) - k_a e^{\left(-\frac{E}{RT}\right)} C_A
\]

(22)
where \( C_A \) is the concentration of \( A \) in the reactor, \( T \) is the reactor temperature and \( T_c \) is the temperature of the coolant stream. The objective is to regulate \( y = x_1 = T \) and \( x_2 = C_A \) by manipulating \( u = T_c \). The constraints are \( 330^\circ K \leq T \leq 400^\circ K \) and \( 240^\circ K \leq T_c \leq 340^\circ K \). In this example the adopted parameter values are those reported in [16]. It is possible to describe a parametrized family of equilibrium points for (22) through the choice of \( x_1 = T \) as scheduling variable, i.e.

\[
C_{A_{eq}}(T) = \frac{q}{V} A F + \frac{k_o e^{-E_R}}{RT},
\]

\[
T_{c_{eq}}(T) = \frac{V \rho C_p}{U A} \left( -\frac{q}{V} (T_f - T) + \frac{\Delta H k_o}{\rho C_p} C_{A_{eq}}(T) \right) + T.
\]  

(23)

Then defining the new state and control variables according to (5)

\[
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2
\end{bmatrix} =
\begin{bmatrix}
T - r \\
C_A - C_{A_{eq}}(T)
\end{bmatrix}, \quad \tilde{u} = T_c - T_{c_{eq}}(T) 
\]

(24)

we obtain the quasi-linear description

\[
\begin{bmatrix}
\dot{\tilde{x}}_1 \\
\dot{\tilde{x}}_2
\end{bmatrix} =
\begin{bmatrix}
0 & -\frac{\Delta H}{\rho C_p} p_1 \\
-\frac{q}{V} & p_3
\end{bmatrix}
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2
\end{bmatrix} +
\frac{U A}{V \rho C_p}
\begin{bmatrix}
1 \\
p_2
\end{bmatrix}
\tilde{u} 
\]

(25)

with

\[
p_1 = k_o e^{-E_R/T_1}, \quad p_2 = -\frac{\partial C_{A_{eq}}(x_1)}{\partial x_1} = \frac{q E A F}{V R x_1^2} \frac{k_o e^{-E_R/T_1}}{\frac{q}{V} + k_o e^{-E_R/T_1}}, \quad p_3 = -p_1 - \frac{\Delta H}{\rho C_p} p_1 p_2
\]

In the new coordinates, the input and state constraints become

\[
240 \leq \tilde{u} + T_{c_{eq}}(x_1) \leq 340, \quad 330 \leq \tilde{x}_1 + r \leq 400 
\]

(26)

Notice that the first constraint exhibits a nonlinear dependence on \( x_1 \). From (26) the set of admissible set-points \( r \) turns out to be:

\[
R_\delta = [330 + \delta, 400 - \delta], \quad \delta = 0.001.
\]

(27)

We selected 17 regions \( P_i \) as indicated in Table 1. For each \( i \), a polytopic embedding of \([A(x_1), B(x_1)]\) with a minimum number of vertices \( (n_v = 4) \) has been found. Next,
Table 1: Control parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_s$</td>
<td>0.03 min</td>
</tr>
<tr>
<td>$\mu$</td>
<td>0.99999</td>
</tr>
<tr>
<td>$\delta$</td>
<td>0.001</td>
</tr>
<tr>
<td>$Q(i)$</td>
<td>${\max{1, i - 1}, i, \min{i + 1, \ell}}$ with $\delta y = 2.75$ or $Q(i) = i$ for $1 \leq i \leq \ell$.</td>
</tr>
<tr>
<td>$\ell$</td>
<td>17</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Overlapping 45%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$P_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[330.00, 340.00], [337.75, 342.75], [340.50, 345.50], [343.25, 348.25], [346.00, 351.00], [348.75, 353.75], [351.50, 356.50], [354.25, 359.25], [357.00, 362.00], [359.75, 364.75], [360.25, 370.25], [365.75, 375.75], [371.25, 381.25], [376.75, 386.75], [382.25, 392.25], [387.75, 397.75], [393.25, 400.00]</td>
</tr>
</tbody>
</table>

For each polytope $\mathcal{F}(i)$ we designed a stabilizing gain $F_i$ with contraction factor 0.96 using LMI techniques; two parameter dynamics were considered: $Q(i) = \{\max\{1, i - 1\}, i, \min\{i + 1, \ell\}\}$ and $Q(i) = i$. The successful calculation of a non-empty $\Sigma$ proves that the corresponding GS controller (14) is stabilizing under the imposed constraints. The number of constraints $n_i$ defining the sets $\Sigma_i^j(r)$, for both parameter dynamics, ranges between 8 and 14. We compared, by simulation experiments, RG and DM tracking strategies under different parameter dynamics. Figs 1(a) and 1(b) compare the behaviours obtained with DMGST and RGGST assuming a diffusive dynamics for the parameter; it can be seen that the additional degree of freedom $\tilde{u}$ of DMGST allows to achieve a faster tracking after a set-point change. Conversely in Figs. 2(a) and 2(b) the state evolutions obtained with the DMGST, considering different evolutions of the parameter, are reported. The constant parameter dynamics $Q(i) = i$ allows a faster response due to the considerable increase of size of the invariant sets $\Sigma^j_i(r)$. This is illustrated.
Figure 1: DM vs RG strategy comparison

in Fig. 3 for \( r = 365 \). The different size of the invariant sets \( \Sigma^i(r) \), provided by the two parameter dynamics, should not surprise. In fact, when \( Q \) is diffusive, \( \Sigma^i(r) \) is an invariant set under all possible allowable sequences of the closed-loop polytopic dynamics \( A + BF_i, [A, B] \in \mathcal{F}(i) \). When the system is strongly nonlinear, the matrices \( A + BF_i \) vary significantly as \( i \) evolves and, therefore, \( \Sigma^i(r) \) reduces to a small neighborhood of the origin. Conversely, when \( Q \) is constant, a single polytopic model is considered. Even in the presence of the additional constraint \( \psi(\Sigma') \subset X_i \), it turns out that \( \Sigma^i(r) \) is much larger than the previous one.

6 Conclusions

The paper has addressed tracking control of a nonlinear system in the presence of state and/or control constraints, paying particular attention to the tracking speed and, at the same time, to the on-line computational complexity. It has been shown that LPV model-
Figure 2: Constant vs diffusive dynamics comparison

(a) Desired set-point (solid), temperature responses of DMGST algorithm for constant (dashed) and diffusive (dash-dotted) dynamics

(b) $C_A$ of DMGST algorithm for constant (dashed) and diffusive (dash-dotted) dynamics

Constant vs gain-scheduling control provide systematic tools for the calculation of constraint-admissible sets and, hence, for the design of an effective reference governor. Furthermore, it has been shown how to significantly improve the tracking response acting both at the optimization level and at the modeling level. More precisely, at the optimization level, the use of the plant control input as a further degree of freedom allowed the enhancement of tracking speed with respect to the reference governor. At the modeling level, an overlapping partition of the scheduling parameter space and the assumption of a constant parameter dynamics yielded a less conservative LPV model, from which larger constraint-admissible sets and, again, a faster tracking. All the proposed tracking algorithms involve the on-line solution of a linear program with a few (one or two, in the single-input case) degrees-of-freedom and can, therefore, be implemented even in control applications requiring high sampling rates.
Figure 3: Invariant sets $\Sigma^i(r)$, $r = 365$, $i = 11$, reported in the original coordinate space for constant (dashed) and diffusive (solid) parameter dynamics together with the family of equilibrium points $C_{A_{eq}}(T)$ (dotted)

References


[5] Chisci L, Zappa G. Dual mode predictive tracking of piecewise constant references


Appendix

Proof of Theorem 2 - The convergence to the equilibrium follows directly from the FRT property. In fact, since $x(t)$ is feasible for $r$, the gain-scheduling control law (15) will be activated at $t$ providing constraint satisfaction for all $t \geq 0$ and convergence of $x$ to $x_{eq}(r)$ and $y(t)$ to $r$. Let the sets $\Sigma_i$ admit the matrix description

$$
\Sigma^i = \left\{ \begin{bmatrix} \tilde{x} \\ r \end{bmatrix} : \Phi_i \begin{bmatrix} \tilde{x} \\ 0 \end{bmatrix} + \Psi_i \begin{bmatrix} r \\ \beta \end{bmatrix} \leq \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \right\}
$$

Since $\Sigma$ is, by construction, $\mu$-contractive $\begin{bmatrix} x_1 - r \\ \tilde{x}_2 \end{bmatrix} \in \Sigma^i(r)$ implies the existence of an admissible $\tilde{u}$ such that

$$
\Phi_i \left[ \begin{bmatrix} \mu^{-1} (A(x_1) \begin{bmatrix} x_1 - r \\ \tilde{x}_2 \end{bmatrix} + B(x_1)\tilde{u}) \end{bmatrix} + \Psi_i^j r \leq \beta_1 \quad \forall j \in Q(i) \right.
$$

from which, multiplying by $\mu$, we get

$$
\Phi_i \left[ A(x_1) \begin{bmatrix} x_1 - r \\ \tilde{x}_2 \end{bmatrix} + B(x_1)\tilde{u} \right] + \Psi_i^j r \leq \beta_1 - (1 - \mu)(\beta_1 - \Psi_i^j r) \quad \forall j \in Q(i) \quad (28)
$$

Let us first assume a diffusive dynamics; then it will be shown how the results carry over to the case of constant dynamics. From the definition of $R_\delta$, thanks to the “safety
margin \( \delta > 0 \), it follows that for any \( r \in R_\delta \) there is a neighborhood of \( \tilde{x} = 0 \) not depending on \( r \), wherein the constraints (2) are satisfied for the closed-loop system (15). Stating this more formally, there exists \( \varepsilon > 0 \) such that \( \delta x \in \Sigma_j(r) \) for all \( \delta x : \|\delta x\|_\infty < \varepsilon \).

Hence

\[
\Psi^j_1 r \leq \beta_1 - \varepsilon \|\Phi^j_1\|_1 1 \tag{29}
\]

Combining (28) and (29), we get that \[
\Phi^j_1 \begin{bmatrix} x_1 - \overline{r} \\ \tilde{x}_2 \end{bmatrix} + B(x_1) \tilde{u} \leq \beta_1 - (1 - \mu)\varepsilon \|\Phi^j_1\|_1 1 \quad \forall j \in Q(i) \tag{30}
\]

For convenience let us rewrite the update of \( r(t) \) as follows

\[
r(t + 1) = r(t) + \rho(r - r(t)) \tag{31}
\]

where \( \rho \triangleq 1 - \lambda(t) \in [0, 1] \). Since the algorithm operates so that \[
\begin{bmatrix} i(t) \\ \tilde{x}(t) \end{bmatrix} \in \Sigma(r(t)) \]

at any time \( t \), there exists an admissible \( \tilde{u} \) such that for some \( j \in Q(i(t)) \)

\[
\Phi^j_1 \begin{bmatrix} x_1(t) - r(t) \\ \tilde{x}_2(t) \end{bmatrix} + B(x_1(t)) \tilde{u} \leq \beta_1 - (1 - \mu)\varepsilon \|\Phi^j_1\|_1 1 \tag{32}
\]

Recall that the objective at time \( t \) is to find the largest \( \rho(t) \in [0, 1] \) and an admissible \( \tilde{u} \) for which

\[
\Phi^j_1 \begin{bmatrix} x_1(t) - r(t + 1) \\ \tilde{x}_2(t) \end{bmatrix} + B(x_1(t)) \tilde{u} \leq \beta_1 \tag{33}
\]

Replacing (31) in (33), we get

\[
\Phi^j_1 \begin{bmatrix} x_1(t) - r(t) \\ \tilde{x}_2(t) \end{bmatrix} + B(x_1(t)) \tilde{u} \leq \beta_1 - \rho \Psi^j_1 (r - r(t)) - \rho \Phi^j_1 (r(t)) \tag{34}
\]

23
Comparing (32) and (34), it turns out that

\[
\bar{\rho}(t) = \begin{cases} 
1 & \text{if } \max \left[ (\Psi_1^j + \Phi_1^j)(r - \bar{r}(t)) \right] \leq 0 \\
\min \left( 1, \frac{(1 - \mu)e \| \Phi_1^j \|_1}{\max[ (\Psi_1^j + \Phi_1^j)(r - \bar{r}(t)) ]} \right) & \text{otherwise}
\end{cases}
\]

(35)

satisfies (34) for the same \( \hat{u} \) appearing in (32). Hence optimization at time \( t \) provides \( \rho(t) \geq \bar{\rho}(t) \). Note that \( \bar{\rho}(t) > 0 \). From (31) it is easy to derive the relationship

\[ r - \bar{r}(t) = \prod_{k=0}^{t-1} (1 - \rho(k))(r - \bar{r}(0)) \]

from which, in turn, exploiting (35)

\[ \bar{\rho}(t) = \frac{\bar{\rho}(0)}{\prod_{k=0}^{t-1} (1 - \rho(k))} \]

whenever \( \bar{\rho}(t) < 1 \). Thus \( \| \bar{r}(t+1) - \bar{r}(t) \| = \rho(t)\| r - \bar{r}(t) \| \geq \bar{\rho}(t)\| r - \bar{r}(t) \| = \bar{\rho}(0)\| r - \bar{r}(0) \| \)

for all \( t \geq 0 \). Taking \( \bar{t} = \frac{1}{\bar{\rho}(0)} \) either \( \rho(t) = 1 \) for some \( t < \bar{t} \) or \( \rho(t) < 1 \) for all \( t < \bar{t} \) and therefore \( \rho(\bar{t}) = 1 \); in both cases we get the result.

Consider now the constant dynamics \( Q(i) = i \). According to the previous reasoning, limited to the reference region \( R^i_\delta \), if \( x(t) \) is feasible for \( \bar{r}(t), \bar{r}(t) \in R^i_\delta \), then in a finite number of steps \( x(t) \) becomes feasible for any other reference \( r \in R^i_\delta \), i.e. \( x(t) \in \Sigma^i(r) \).

Assume now that \( R^i_\delta \) and \( R^j_\delta \) have non void intersection and that \( r \in R^i_\delta \cap R^j_\delta \). Clearly \( x_{eq}(r) \in \Sigma^i(r) \cap \Sigma^j(r) \) as well, then, repeating the procedure, \( x(t) \) becomes feasible in finite time for any other \( r' \in \Sigma^j(r) \). Otherwise applying the linear feedback \( \tilde{u} = F_i \tilde{x} \), \( x(t) \) converges exponentially fast to \( x(r) \), i.e. it enters, in a finite number of steps, into a neighborhood of \( x_{eq}(r) \), i.e. \( x(t) \in \Sigma^i(r) \cap \Sigma^j(r) \). Iterating this procedure the theorem follows. 

\[ \blacksquare \]