Automatica 49 (2013) 3695–3699

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Technical communique

Improved model prediction and RMPC design for LPV systems with bounded parameter changes[☆]

Pengyuan Zheng[a,b,c], Dewei Li[a,c], Yugeng Xi[a,c], Jun Zhang[b,c,1]

[a] Department of Automation, Shanghai Jiao Tong University, Shanghai, 200240, China
[b] Joint Institute of UMich-SJTU, Shanghai Jiao Tong University, Shanghai, 200240, China
[c] Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai, 200240, China

ARTICLE INFO

Article history:
Received 2 June 2013
Received in revised form
18 August 2013
Accepted 9 September 2013
Available online 15 October 2013

Keywords:
Linear parameter varying system
Bounded change
Model prediction
Robust model predictive control

ABSTRACT

This paper studies the future model prediction and robust model predictive control (RMPC) design for linear parameter varying systems with bounded parameter changes. By developing tight bound estimations for varying parameters, we construct a set-valued map as the predicted family of future models. This construction attains accurate estimations and thus reduces conservativeness. Based on model predictions, we use a parameter-dependent feedback to design RMPC that achieves an enhanced performance with guaranteed robust and stability properties.

1. Introduction

In the past 15 years, robust model predictive control (RMPC) has attracted extensive research interests (Alamo, Ramirez, Muñoz de la Peña, & Camacho, 2007; Limon, Alvarado, Alamo, & Camacho, 2010; Mayne, Rawlings, Rao, & Scokaert, 2000). In particular, significant progresses have been made in applying RMPC to linear parameter varying (LPV) systems (Kothare, Balakrishnan, & Morari, 1996; Wan & Kothare, 2003), especially to those with bounded parameter changes (Amato, Mattei, & Pironti, 2005; Casavola, Famularo, & Franze, 2002, 2003). Many of these results are focused on two critical issues, *i.e.,* the future model prediction and the RMPC design.

For the future model prediction, Casavola et al. (2002), Casavola et al. (2003) and Garone, Casavola, Franze, and Famularo (2007) considered a general case when the system parameters are in a unit simplex and then calculated the future families. Jungers, Oliveira, and Peres (2011) and Oliveira and Peres (2009) used the geometry of an uncertainty region and represented the parameters and their changes as a polytope. For the RMPC design, Lu and Arkun (2000) developed quasi-min–max MPC algorithms with 1-step control horizon. An extension to an arbitrary control horizon was studied in Casavola et al. (2002), and the stability condition with a reduced computational cost was generalized in Casavola et al. (2003). Li and Xi (2010) applied a sequence of feedback controls to ensure the recursive feasibility and stability.

This paper develops an accurate prediction method of the future model for LPV systems with bounded parameter changes and then designs the parameter-dependent RMPC algorithm with enhanced performance. By deriving tight bounds of varying parameters, we construct a set-valued map to predict future models. This method provides an elegant physical interpretation about the evolution of the future system models, and it also achieves better estimations and thus reduces the design conservativeness than those reported in Li and Xi (2010). Applying parameter-dependent feedback, we design an RMPC algorithm that solves for a set of state feedback controls based on future models. The resulting controller achieves better overall performance with guaranteed robust and stability properties.

Notations: For vectors $\alpha = [\alpha_1, \ldots, \alpha_n]^T$ and $\beta = [\beta_1, \ldots, \beta_n]^T$, $\alpha \preceq \beta$ (or $\beta \succeq \alpha$) denotes $\alpha_l \leq \beta_l$ (or $\beta_l \geq \alpha_l$) for $l = 1, \ldots, n$. The symbols $\mathbf{0}$ and $\mathbf{1}$ denote the vectors with all entries being 0 and 1, respectively.

☆ The authors thank the financial support from NSF China No. 61374110, 61333009, 60934007, 61174086, 61221003 and 61203073. The material in this paper was not presented at any IFAC conference. This paper was recommended for publication in revised form by Associate Editor Andrey V. Savkin under the direction of Editor André L. Tits.

E-mail addresses: pyzheng@sjtu.edu.cn (P. Zheng), dwli@sjtu.edu.cn (D. Li), ygxi@sjtu.edu.cn (Y. Xi), zhangjun12@sjtu.edu.cn (J. Zhang).

1 Tel.: +86 21 34206731; fax: +86 21 3420 6525.

0005-1098/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.automatica.2013.09.024

2. Prediction of future system models

Let us consider an LPV system

$$x(k+1) = A(k)x(k) + B(k)u(k), \quad x \in \mathbb{R}^n, u \in \mathbb{R}^{n_i}, \tag{1}$$

where $x(k)$ can be measured, $|u(k)| \preceq \bar{u} \in \mathbb{R}^{n_i}$, and $|Fx(k)| \preceq \bar{x} \in \mathbb{R}^{n_x}$. Here $|\cdot|$ denotes the elementwise absolute value of a vector, and $\bar{u}, \bar{x}$ the bound vectors. Juxtapose $A(k)$ and $B(k)$ to form a matrix $[A(k)|B(k)]$ that represents the system model. Assume that $[A(k)|B(k)] \in \mathcal{P}_0$, where $\mathcal{P}_0$ is the convex hull spanned by $\{[A_l|B_l]\}_{l=1}^{n_p}$, *i.e.*, there exist $\lambda_l(k) \geq 0$ and $\sum_{l=1}^{n_p} \lambda_l(k) = 1$ such that

$$[A(k)|B(k)] = \sum_{l=1}^{n_p} \lambda_l(k)[A_l|B_l]. \tag{2}$$

Assume that the values of $\lambda_l(k)$ are known at time k, and their rates of changing are bounded by

$$|\lambda_l(k+1) - \lambda_l(k)| \leq \Delta_l, \quad l = 1, \ldots, n_p. \tag{3}$$

For the LPV system (1)–(3), we want to find the polytopic families that contain the future system models. To reduce the burden of online computation, it is desired to fix the number of vertices of the polytopic families.

To predict the future model, we first develop a tight iterative procedure to estimate the lower and upper bounds of the varying parameter λ.

Lemma 1. *Consider $\lambda = [\lambda_1, \ldots, \lambda_{n_p}]^T \succeq \mathbf{0}$ and $\sum_{l=1}^{n_p} \lambda_l = 1$. If $\mathbf{0} \preceq b^1 \preceq \lambda \preceq d^1 \preceq \mathbf{1}$, a series of nondecreasing lower bounds $\{b^n\}$ and non-increasing upper bounds $\{d^n\}$ of λ can be obtained from the iteration:*

$$\begin{aligned} b^{n+1} &= \max\left\{\left(1 - \sum_{l=1}^{n_p} d_l^n\right)\mathbf{1} + d^n, b^n\right\} \succeq \mathbf{0}, \\ d^{n+1} &= \min\left\{\left(1 - \sum_{l=1}^{n_p} b_l^n\right)\mathbf{1} + b^n, d^n\right\} \preceq \mathbf{1}. \end{aligned} \tag{4}$$

Furthermore, this iteration terminates when

$$d^n - b^n \preceq \min\left\{1 - \sum_{l=1}^{n_p} b_l^n, \sum_{l=1}^{n_p} d_l^n - 1\right\}\mathbf{1}. \tag{5}$$

Here min and max operations are taken elementwise.

Proof. From the assumption, we have that for any l, $0 \leq b_l^1 \leq \lambda_l \leq d_l^1 \leq 1$. Sum this inequality from $l = 1$ to n_p except i, and take into account that $\lambda_i = 1 - \sum_{l=1, l\neq i}^{n_p} \lambda_l$:

$$1 - \sum_{l=1}^{n_p} d_l^1 + d_i^1 \leq \lambda_i \leq 1 - \sum_{l=1}^{n_p} b_l^1 + b_i^1.$$

Since $b_i^1 \leq \lambda_i \leq d_i^1$, it yields that $\lambda_i \in [b_i^2, d_i^2]$, where

$$b_i^2 = \max\left\{1 - \sum_{l=1}^{n_p} d_l^1 + d_i^1, b_i^1\right\},$$

$$d_i^2 = \min\left\{1 - \sum_{l=1}^{n_p} b_l^1 + b_i^1, d_i^1\right\}.$$

Repeating this procedure leads to the iteration (4). It is clear that $\{b^n\}$ is nondecreasing and $\{d^n\}$ is non-increasing. When Eq. (5) is satisfied, the iteration will not generate any tighter bound and thus terminates. ■

From an early assumption, $\lambda(k)$ is known at time k. To estimate $\lambda(k+1)$, take the initial bounds as

$$b^1(1) = \max\{\lambda(k) - \Delta, \mathbf{0}\}, d^1(1) = \min\{\lambda(k) + \Delta, \mathbf{1}\}, \tag{6}$$

where $\Delta = [\Delta_1, \ldots, \Delta_{n_p}]^T$. By iterating (4), we can obtain the lower bound $b(1)$ and upper bound $d(1)$ for $\lambda(k+1)$. In real time implementation, we can specify a maximal iteration number to reduce the computational time. Similarly, the bounds for $\lambda(k+2)$ can be obtained by taking the initial bounds as

$$b^1(2) = \max\{b(1) - \Delta, \mathbf{0}\}, \qquad d^1(2) = \min\{d(1) + \Delta, \mathbf{1}\},$$

and then iterating (4). Continuing this procedure leads to a lower bound $b(j)$ and an upper bound $d(j)$ for $\lambda(k+j)$. It is clear that $b(j+1) \preceq b(j)$ and $d(j+1) \succeq d(j)$. We then define a set-valued map $\mathcal{T}(b)$ as

$$\mathcal{T}(b) := \sum_{l=1}^{n_p} b_l[A_l|B_l] + \left(1 - \sum_{l=1}^{n_p} b_l\right)\mathcal{P}_0. \tag{7}$$

Here the addition + is in the sense of the Minkowski sum. The predicted polytopic family at time $k+j$ can be written as $\mathcal{P}(k+j) = \mathcal{T}(b(j))$, and the algorithm to predict the future system models is obtained as follows.

Algorithm 1. *Step* 1. Calculate $d(m)$ and $b(m)$ for $m = 1, 2, \ldots$ until $b(m) = \mathbf{0}$;
Step 2. For $j = 1, \ldots, m-1$, let $\mathcal{P}(k+j) = \mathcal{T}(b(j))$; for $j \geq m$, let $b(j) = 0$ and thus $\mathcal{P}(k+j) = \mathcal{P}_0$.

The following theorem ensures that the polytopic family $\mathcal{P}(k+j)$ from Algorithm 1 indeed contains the future system model $[A(k+j)|B(k+j)]$.

Theorem 2. *For the future model $[A(k+j)|B(k+j)]$ and $\mathcal{P}(k+j)$ obtained from Algorithm 1, we have for all $h \geq 1$,*

$$[A(k+j)|B(k+j)] \in \mathcal{P}(k+j) \subseteq \mathcal{P}(k+j+h). \tag{8}$$

To prove Theorem 2, we need the following two lemmata.

Lemma 3. *Consider two vectors $\lambda = [\lambda_1, \ldots, \lambda_{n_p}]^T$ and $b = [b_1, \ldots, b_{n_p}]^T$. If $\lambda \succeq b \succeq \mathbf{0}$, $\sum_{l=1}^{n_p} b_l \leq 1$, and $\sum_{l=1}^{n_p} \lambda_l = 1$, then there exists a vector $p = [p_1, \ldots, p_{n_p}]^T \succeq \mathbf{0}$ with $\sum_{l=1}^{n_p} p_l = 1$ such that*

$$\lambda = \left(1 - \sum_{l=1}^{n_p} b_l\right)p + b. \tag{9}$$

The proof is easy. If $\sum_{l=1}^{n_p} b_l = 1$, it implies $\lambda = b$, and Eq. (9) is trivially satisfied; if $\sum_{l=1}^{n_p} b_l < 1$, let $p = (\lambda - b)/\left(1 - \sum_{l=1}^{n_p} b_l\right) \succeq \mathbf{0}$, then p satisfies Eq. (9).

Next lemma shows that the set-valued map $\mathcal{T}$ is non-increasing in the sense of set inclusion.

Lemma 4. *Consider two vectors $\alpha = [\alpha_1, \ldots, \alpha_{n_p}]^T \succeq \mathbf{0}$ and $\beta = [\beta_1, \ldots, \beta_{n_p}]^T \succeq \mathbf{0}$ with $\sum_{l=1}^{n_p} \alpha_l \leq 1$ and $\sum_{l=1}^{n_p} \beta_l \leq 1$. If $\alpha \preceq \beta$, then $\mathcal{T}(\beta) \subseteq \mathcal{T}(\alpha)$.*

Proof. Denote the i-th vertex of $\mathcal{T}(\beta)$ as Vrtx_i:

$$\text{Vrtx}_i = \sum_{l=1}^{n_p} \beta_l[A_l|B_l] + \left(1 - \sum_{l=1}^{n_p} \beta_l\right)[A_i|B_i].$$

It suffices to show that for an arbitrary i, Vrtx_i lies in $\mathcal{T}(\alpha)$. Let $\theta = [\theta_1, \ldots, \theta_{n_p}]^T$, where

$$\theta_l = \begin{cases} \beta_l, & \text{if } l \neq i; \\ \beta_i + 1 - \sum_{m=1}^{n_p} \beta_m & \text{if } l = i. \end{cases}$$

Then $\text{Vrtx}_i = \sum_{l=1}^{n_p} \theta_l [A_l|B_l]$. From $\sum_{l=1}^{n_p} \beta_l \le 1$ and $\beta \succeq \alpha$, we get $\theta \succeq \alpha$. Since $\sum_{l=1}^{n_p} \theta_l = 1$, from Lemma 3, there exists a vector $p = [p_1, \ldots, p_{n_p}]^T \succeq 0$ with $\sum_{l=1}^{n_p} p_l = 1$ such that $\theta = \left(1 - \sum_{l=1}^{n_p} \alpha_l\right) p + \alpha$. Hence,

$$\text{Vrtx}_i = \sum_{l=1}^{n_p} \alpha_l [A_l|B_l] + \left(1 - \sum_{l=1}^{n_p} \alpha_l\right) \sum_{l=1}^{n_p} p_l [A_l|B_l],$$

which implies that $\text{Vrtx}_i \in \mathcal{T}(\alpha)$. So $\mathcal{T}(\beta) \subseteq \mathcal{T}(\alpha)$. ■

Now we are ready to prove Theorem 2.

Proof of Theorem 2. The future system model for time $k + j$ predicted at time k can be written as

$$[A(k+j)|B(k+j)] = \sum_{l=1}^{n_p} \lambda_l(k+j)[A_l|B_l].$$

From Lemma 1, $\lambda(k+j) \succeq b(j) \succeq \mathbf{0}$. Also, since $b(j+1) \preceq b(j)$, it follows that

$$\sum_{l=1}^{n_p} b_l(j) \le \sum_{l=1}^{n_p} b_l(1) \le \sum_{l=1}^{n_p} \lambda_l(k+1) = 1.$$

Now from Lemma 3, there exists a vector $p(k+j) \succeq \mathbf{0}$ with $\sum_{l=1}^{n_p} p_l(k+j) = 1$ such that

$$\lambda(k+j) = \left(1 - \sum_{l=1}^{n_p} b_l(j)\right) p(k+j) + b(j). \tag{10}$$

Therefore,

$$[A(k+j)|B(k+j)] = \sum_{l=1}^{n_p} b_l(j)[A_l|B_l] + \left(1 - \sum_{l=1}^{n_p} b_l(j)\right) \times \sum_{l=1}^{n_p} p_l(k+j)[A_l|B_l]. \tag{11}$$

From Eq. (7), Eq. (11) implies that $[A(k+j)|B(k+j)] \in \mathcal{P}(k+j)$. Further, since $b(j+h) \preceq b(j)$, from Lemma 4, we obtain $\mathcal{P}(k+j) \subseteq \mathcal{P}(k+j+h)$. ■

Denote the i-th vertex of $\mathcal{P}(k+j)$ as $[A_i(j)|B_i(j)]$ (the dependence on k is dropped hereafter for brevity). Since $\mathcal{P}(k+j) = \mathcal{T}(b(j))$ and again from Eq. (7), we have

$$[A_i(j)|B_i(j)] = \sum_{l=1}^{n_p} b_l(j)[A_l|B_l] + \left(1 - \sum_{l=1}^{n_p} b_l(j)\right)[A_i|B_i]. \tag{12}$$

Eq. (11) can then be rewritten as

$$[A(k+j)|B(k+j)] = \sum_{l=1}^{n_p} p_l(k+j)[A_l(j)|B_l(j)], \tag{13}$$

i.e., the future model $[A(k+j)|B(k+j)]$ can be written as a convex combination of the vertices of $P(k+j)$ with combination coefficients $p_l(k+j)$ defined from Eq. (10).

Thanks to the set-valued map $\mathcal{T}$ defined in Eq. (7), the current prediction algorithm also has an elegant physical interpretation about the evolutions of future system models. From Eq. (7), it can be seen that predicting future polytopic family $\mathcal{P}(k+j)$ amounts to first shrinking $\mathcal{P}_0$ with a factor $1-\sum_{l=1}^{n_p} b_l(j)$ and then translating it by an amount of $\sum_{l=1}^{n_p} b_l(j)[A_l|B_l]$. This illustrates how the polytopic families evolve; it is also evident that $\mathcal{P}(k+j)$ is always similar to $\mathcal{P}_0$. Moreover, the scaling factor $1-\sum_{l=1}^{n_p} b_l(j)$ can be conveniently used as a measure of the uncertainty of the predicted model: when it is 1, we know nothing about the system model except that it is in $\mathcal{P}_0$; when it is 0, we exactly know what the system model is.

In Li and Xi (2010), two of the present authors have reported an earlier work on calculating the future model polytopic families. It is worth mentioning that Algorithm 1 in this paper (referred as A1 hereafter) always yields tighter polytopic families of future models than the algorithm in Li and Xi (2010) (referred as A1*). This is manifested by the following theorem showing that the future model polytope predicted by A1 is indeed a subset of that by A1*.

Theorem 5. *Consider the nontrivial cases when $n_p \ge 2$. For the predicted future model $\mathcal{P}(k+j)$ obtained from A1 at time k, we have that*

$$\mathcal{P}(k+j) \subseteq tran(a_j, \mathcal{P}_0), \tag{14}$$

where $tran(a_j, \mathcal{P}_0)$ is the future model predicted by A1* *at time k.*

Proof. The procedure to calculate a_j in Eq. (14) is excerpted from Li and Xi (2010) for completeness[2]:

$$a_1 = \min\{\max\{\max(\lambda(k)) + \Delta_0, 1 - (n_p - 1)(\min(\lambda(k)) - \Delta_0)\}, 1\}, \tag{15}$$

$$a_j = \min\left\{\max\left\{a_{j-1} + \Delta_0, 1 - (n_p - 1)\left(\frac{1 - a_{j-1}}{n_p - 1} - \Delta_0\right)\right\}, 1\right\}. \tag{16}$$

We will first simplify Eqs. (15) and (16). Since $\sum_{l=1}^{n_p} \lambda_l(k) = 1$, it can be derived that

$$1 - (n_p - 1)\min \lambda(k) \ge \max \lambda(k).$$

Furthermore, because $n_p \ge 2$, we have

$$1 - (n_p - 1)(\min \lambda(k) - \Delta_0) \ge \max \lambda(k) + \Delta_0.$$

Thus, Eq. (15) can be simplified into

$$a_1 = \min\left\{1 - (n_p - 1)(\min \lambda(k) - \Delta_0), 1\right\}. \tag{17}$$

Letting

$$b_1 = \max\{\min \lambda(k) - \Delta_0, 0\}, \tag{18}$$

we can obtain $a_1 = 1 - (n_p - 1)b_1$. Similarly, we get

$$1 - (n_p - 1)\left(\frac{1 - a_{j-1}}{n_p - 1} - \Delta_0\right) \ge a_{j-1} + \Delta_0.$$

Thus a_j in Eq. (16) can be simplified into

$$a_j = \min\left\{1 - (n_p - 1)\left(\frac{1 - a_{j-1}}{n_p - 1} - \Delta_0\right), 1\right\}. \tag{19}$$

Letting $b_j = \max\left\{\frac{1-a_{j-1}}{n_p-1} - \Delta_0, 0\right\}$, we can obtain $a_j = 1 - (n_p - 1)b_j$. Then b_j can also be calculated from $b_j = \max\left\{b_{j-1} - \Delta_0, 0\right\}$. The best parameter Δ_0 for A1* when the rates of parameter changes are different is $\Delta_0 = \max\{\Delta_1, \ldots, \Delta_{n_p}\}$, and the corresponding polytopic set is

$$tran(a_j, \mathcal{P}_0) = \mathcal{T}(b_j\mathbf{1}). \tag{20}$$

From the definitions of $b^1(1)$ in Eq. (6) and b_1 in Eq. (18), it is clear that $b^1(1) \succeq b_1\mathbf{1}$. From Lemma 1, we have $b(1) \succeq b^1(1)$, and thus $b(1) \succeq b_1\mathbf{1}$. Similarly, we can prove that $b(j) \succeq b_j\mathbf{1}$, for all $j \ge 2$. From Lemma 4, it is immediate that $\mathcal{T}(b(j)) \subseteq \mathcal{T}(b_j\mathbf{1})$. Taking Eq. (20) into account, we can prove Eq. (14) as desired. ■

2 Note that Eq. (16) corrects some typographical errors in Li and Xi (2010). Also, we use Δ_0 to denote Δ in Li and Xi (2010) to avoid notation conflict, and Δ_0 is indeed a scalar.

From the proof, we see that A1* uses only the smallest element in $\lambda(k)$ to predict the future model polytopic families, which inevitably yields unnecessary slackness. In contrast, the proposed A1 utilizes all the elements to render more freedom and can thus reduce the conservativeness and deal with different rates of parameter changes effectively.

3. RMPC design

Taking advantage of the set-valued map of future model prediction, we design an RMPC by the parameter-dependent state feedback with more design variables.

We formulate the RMPC design at time k as a mini–max problem:

$$\min_{U(k)} \max_{[A(k+j)|B(k+j)]\in\mathcal{P}(k+j),j\geq 0} J(k) \quad \text{s.t. (1)–(3)}, \tag{21}$$

where

$$J(k) = \sum_{j=0}^{\infty}(\|x(k+j|k)\|_L^2 + \|u(k+j|k)\|_R^2), \tag{22}$$

and $L > 0, R > 0$ are symmetric weighting matrices.

In Eq. (22), the current control $u(k)$ is chosen as a variable to be optimized. All the future controls $u(k+j|k)$ are determined by state feedback controls:

$$u(k+j|k) = K(k+j)x(k+j|k), \quad j = 1, \ldots, \infty, \tag{23}$$

where the feedback gain $K(k+j)$ is chosen as

$$K(k+j) = \begin{cases} \sum_{l=1}^{n_p} p_l(k+j)K_l(j), & \text{if } j = 1, \ldots, N; \\ K(k+N), & \text{if } j > N. \end{cases}$$

Here $p_l(k+j)$ are the convex combination coefficients from Eq. (13), and each $K_l(j)$ is a state feedback gain to be designed for all the vertices of $\mathcal{P}(k+j)$. Eq. (23) thus defines a parameter-dependent state feedback control.

Similar to the techniques used in Casavola et al. (2003), Kothare et al. (1996) and Wang, Tanaka, and Griffin (1996), we can transform the mini–max problem (21) into a semidefinite programming (SDP) problem:

$$\min_{\gamma,\gamma_1,u(k),Q_j,Y_l(j),X_j,H_j} \gamma_1 + \gamma, \tag{24}$$

$$s.t. \quad -\bar{u}_i \leq u_i(k) \leq \bar{u}_i, \quad i = 1, \ldots, n_i;$$

$$\begin{bmatrix} X_j & \star \\ Y_l(j)^T & Q_j \end{bmatrix} \geq 0, \quad \begin{matrix} X_{j,ii} \leq \bar{u}_i^2, & i = 1, \ldots, n_i, \\ j = 1, \ldots, N, & l = 1, \ldots, n_p; \end{matrix}$$

$$\begin{bmatrix} H_j & \star \\ Q_jF^T & Q_j \end{bmatrix} \geq 0, \quad H_{j,ii} \leq \bar{x}_i^2, \quad \begin{matrix} j = 1, \ldots, N, \\ i = 1, \ldots, n_x; \end{matrix}$$

$$\begin{bmatrix} \gamma_1 & \star \\ u(k) & R^{-1} \end{bmatrix} \geq 0; \quad \begin{bmatrix} 1 & \star \\ A(k)x(k) + B(k)u(k) & Q_1 \end{bmatrix} \geq 0;$$

$$\begin{bmatrix} Q_j & \star & \star & \star \\ A_l(j)Q_j + B_l(j)Y_l(j) & Q_{j+1} & 0 & 0 \\ L^{1/2}Q_j & 0 & \gamma & 0 \\ R^{1/2}Y_l(j) & 0 & 0 & \gamma \end{bmatrix} \geq 0, \quad \begin{matrix} j = 1, \ldots, N, \\ l = 1, \ldots, n_p; \end{matrix}$$

$$\begin{bmatrix} Q_j & \star & \star & \star \\ \frac{(A_l(j) + A_i(j))Q_j + B_l(j)Y_i(j) + B_i(j)Y_l(j)}{2} & Q_{j+1} & 0 & 0 \\ L^{1/2}Q_j & 0 & \gamma & 0 \\ R^{1/2}\frac{Y_l(j) + Y_i(j)}{2} & 0 & 0 & \gamma \end{bmatrix} \geq 0,$$

$$j = 1, \ldots, N, l = 1, \ldots, n_p - 1, i = l+1, \ldots, n_p.$$

where $[A_l(j)|B_l(j)]$ are from Eq. (12), $[A_l(N)|B_l(N)]$ are chosen as vertices of $\mathcal{P}_0$, and $Q_{N+1} = Q_N$.

For all the vertices, the RMPC controller designed above applies a convex combination of state feedbacks, whereas Algorithm 2 in Li and Xi (2010) applies one single state feedback. It is clear that the design here has more design variables and thus can enhance control performance. Moreover, it can be shown that the closed-loop system is stable.

Note that this algorithm has more optimizing variables and more LMI conditions, which will result in heftier online computational burden. To alleviate it, we can further engage the efficient RMPC algorithm of offline invariant set construction and online synthesis developed in Wan and Kothare (2003).

4. Numerical studies

We now use numerical examples to show the efficacy of the proposed prediction and control algorithms. For convenience, denote Algorithm 2 from Li and Xi (2010) as A2*, and the RMPC algorithm (24) in this paper as A2. Also consider an algorithm mA2*, where Eq. (22) in A2* is replaced by the optimization design (24) in this paper.

Consider an LPV system where

$$A_1 = \begin{bmatrix} 1.15 & 0.05 \\ 0.05 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.15 & 0.05 \\ 3.5 & 1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1.15 & 0.05 \\ 3 & 0.9 \end{bmatrix},$$

$$A_4 = \begin{bmatrix} 1.15 & 0.05 \\ 1 & 0.9 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 & 0.5 \\ 0.02 & 0.2 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 1 & 0.5 \\ 0.1 & 0.2 \end{bmatrix},$$

$$B_3 = \begin{bmatrix} 1 & 0.01 \\ 0.02 & 0.2 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 1 & 0.01 \\ 0.01 & 0.2 \end{bmatrix}.$$

Choose L, R as 2×2 identity matrices, and $\bar{u} = [1, 1]^T$, $\Delta_1 = 0.05$, $\Delta_2 = 0.03$, $\Delta_3 = 0.024$, $\Delta_4 = 0.056$. The parameters vary as follows:

$$\lambda_1(k+1) = \begin{cases} 0, & \text{if } \lambda_1(k+1) \leq 0, \\ 0.5, & \text{if } \lambda_1(k+1) \geq 0.5, \\ \lambda_1(k) + (-1)^{\lfloor k/6 \rfloor}0.05, & \text{otherwise}, \end{cases}$$

$$\lambda_2(k+1) = \begin{cases} 0, & \text{if } \lambda_1(k+1) \leq 0, \\ 0.5, & \text{if } \lambda_2(k+1) \geq 0.5, \\ \lambda_2(k) + (-1)^{\lfloor k/4 \rfloor}0.03, & \text{otherwise}, \end{cases}$$

$$\lambda_3(k+1) = 0.3(1 - \lambda_1(k+1) - \lambda_2(k+1)),$$

$$\lambda_4(k+1) = 1 - \lambda_1(k+1) - \lambda_2(k+1) - \lambda_3(k+1),$$

where $\lfloor\cdot\rfloor$ denotes the floor operation. The initial parameters are chosen as $\lambda_1(k) = 0.3$, $\lambda_2(k) = 0.4$, $\lambda_3(k) = 0.09$, $\lambda_4(k) = 0.21$. Also set $x(0) = [2, 2]$ and $N = 3$ for A2*, A2, and mA2*.

Fig. 1 shows the polytopic families of future models from A1 and A1*. It is clear that A1 produces smaller polytopic families and thus is less conservative than A1* as theoretical analysis revealed earlier. The closed-loop state responses are plotted in Fig. 2. Both A2 and mA2* can drive the system faster than A2*. Further A2 has the least control cost, whereas mA2* is less than A2* as well. We have thus shown the efficacy of our future model predictions and the RMPC design algorithm.

The main reason that this RMPC controller can reduce the conservativeness and improve the overall performance is twofold. First, the prediction algorithm provides a more accurate estimation about the future system models. Second, the proposed RMPC algorithm in this paper applies a family of parameter-dependent feedback controllers for all the vertices; thus it has more design variables so as to achieve better performance.

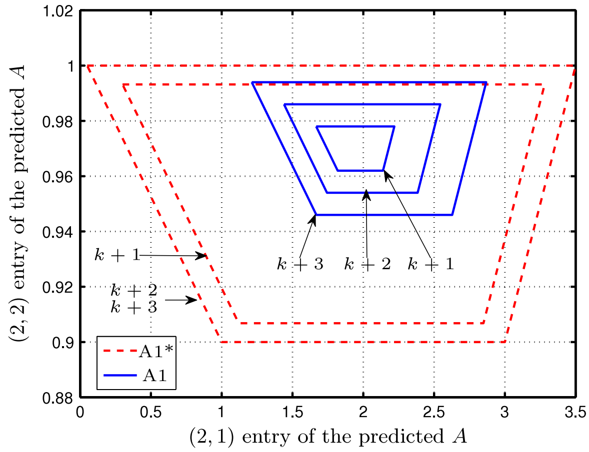

Fig. 1. Prediction of future model polytopic families.

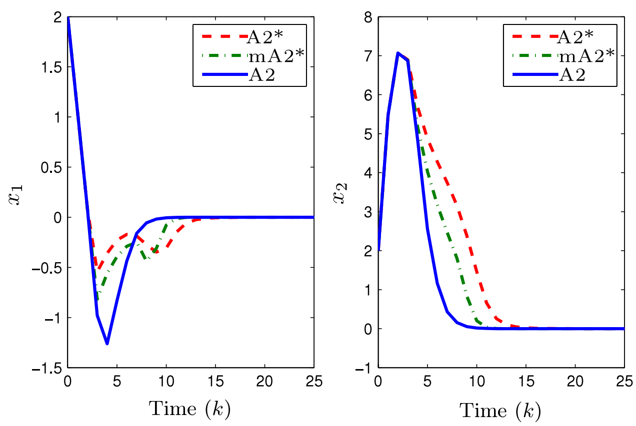

Fig. 2. State responses for the closed-loop system.

Another interesting question to explore is the bounds of variations for the parameters. In reality, it is usually difficult to know precisely how the model parameters are varying. The bounds of variations can thus provide a useful means to estimate the possible ranges of these parameters.

In Fig. 3, we plot the lower bounds for the 1-step prediction of varying parameters for both A1 and A1*. It is clear that the lower bounds from A1 are constantly larger than those from A1*, which indicates that A1 yields tighter bounds as expected. Moreover, it can be seen that A1* essentially uses the same lower bound for all the different parameters and thus it cannot achieve good future prediction. As a contrast, A1 is a more adept algorithm since it uses different lower bounds for different parameters and therefore generates tighter parameter ranges. Different lower bound estimations also increase the prediction design variables and therefore make it possible for the future model prediction to cut down the conservativeness and to achieve more accurate system models.

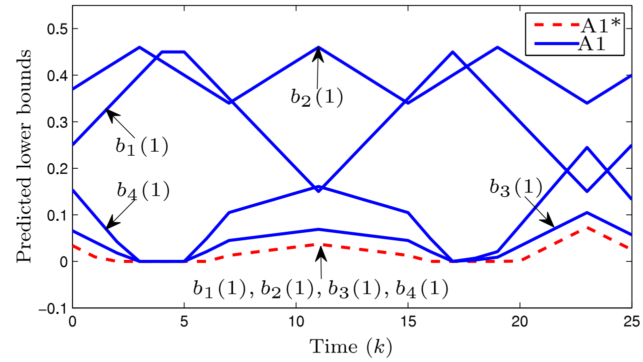

Fig. 3. Predicted lower bounds of varying parameters.

5. Conclusion

This paper developed a set-valued map to predict the future models of the LPV system with bounded parameter changes. This method can achieve more accurate results than previously reported. An RMPC algorithm taking advantage of this model prediction was developed to achieve better control performance. Numerical examples demonstrate the effectiveness of the algorithms.

References

Alamo, T., Ramirez, D. R., Muñoz de la Peña, D., & Camacho, E. F. (2007). Min–max MPC using a tractable QP problem. *Automatica*, *43*(4), 693–700.

Amato, F., Mattei, M., & Pironti, A. (2005). Gain scheduled control for discrete-time systems depending on bounded rate parameters. *International Journal of Robust and Nonlinear Control*, *15*(11), 473–494.

Casavola, A., Famularo, D., & Franze, G. (2002). A feedback min–max MPC algorithm for LPV systems subject to bounded rates of change of parameters. *IEEE Transactions on Automatic Control*, *47*(7), 1147–1153.

Casavola, A., Famularo, D., & Franze, G. (2003). Predictive control of constrained nonlinear systems via LPV linear embeddings. *International Journal of Robust and Nonlinear Control*, *13*(3-4), 281–294.

Garone, E., Casavola, A., Franze, G., & Famularo, D. (2007). New stabilizability conditions for discrete-time linear parameter varying systems. In *46th IEEE CDC* (pp. 2755–2760). IEEE.

Jungers, M., Oliveira, R. C. L. F., & Peres, P. L. D. (2011). MPC for LPV systems with bounded parameter variations. *International Journal of Control*, *84*(1), 24–36.

Kothare, M. V., Balakrishnan, V., & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, *32*(10), 1361–1379.

Li, D., & Xi, Y. (2010). The feedback robust MPC for LPV systems with bounded rates of parameter changes. *IEEE Transactions on Automatic Control*, *55*(2), 503–507.

Limon, D., Alvarado, I., Alamo, T., & Camacho, E. F. (2010). Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, *20*(3), 248–260.

Lu, Y., & Arkun, Y. (2000). Quasi-min–max MPC algorithms for LPV systems. *Automatica*, *36*(4), 527–540.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: stability and optimality. *Automatica*, *36*(6), 789–814.

Oliveira, R. C. L. F., & Peres, P. L. D. (2009). Time-varying discrete-time linear systems with bounded rates of variation: stability analysis and control design. *Automatica*, *45*(11), 2620–2626.

Wan, Z., & Kothare, M. V. (2003). Efficient robust constrained model predictive control with a time varying terminal constraint set. *Systems & Control Letters*, *48*(5), 375–383.

Wang, H. O., Tanaka, K., & Griffin, M. F. (1996). An approach to fuzzy control of nonlinear systems: stability and design issues. *IEEE Transactions on Fuzzy System*, *4*(1), 14–23.