Interpolation-based Off-line MPC for LPV systems

P. Bumroongsri^{a,b}, S. Kheawhom^{a,*}

 ^aComputational Process Engineering, Department of Chemical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, 10330, Thailand (e-mail: soorathep.k@chula.ac.th).
 ^bDepartment of Chemical Engineering, Faculty of Engineering, Mahidol University, Nakhon Pathom, 73170 Thailand (e-mail: pornchai.bum@mahidol.ac.th).

Abstract: Interpolation-based off-line MPC for LPV systems is presented in this work. The on-line computational time is reduced by pre-computing off-line the sequences of state feedback gains corresponding to the sequences of ellipsoidal invariant sets. At each sampling time, the real-time state feedback gain is calculated by linear interpolation between the pre-computed state feedback gains. Four interpolation techniques are presented. In the first technique, the smallest ellipsoid containing the current state measured is approximated and the corresponding real-time state feedback gain is calculated. In the second technique, the pre-computed state feedback gains are interpolated in order to get the largest possible real-time state feedback gain while robust stability is still guaranteed. In the third technique, the real-time state feedback gain is calculated by minimizing the violation of the constraints of the adjacent inner ellipsoids so the real-time state feedback gain calculated has to regulate the state from the current ellipsoids to the adjacent inner ellipsoids as fast as possible. In the last technique, the real-time state feedback gain is calculated by minimizing the one-step cost function so the real-time state feedback gain calculated has to regulate the next predicted state to the origin as fast as possible. A case study of nonlinear CSTR is presented to illustrate the implementation of the proposed techniques. The results show that the proposed interpolation techniques 2, 3 and 4 tend to produce less sluggish responses than the technique 1.

Keywords: Off-line MPC, LPV systems, Interpolation techniques.

1. INTRODUCTION

Model predictive control (MPC) has originated in the industries as an effective control algorithm to solve multivariable control problem. Although MPC based on a linear model has been successfully implemented in many industrial applications, it is well-known that the stability of MPC based on a linear model cannot be guaranteed in the presence of process nonlinearity (Morari and Lee, 1999). This has motivated the synthesis of MPC using linear parameter varying (LPV) model whose dynamics depend on the scheduling parameter that can be measured on-line (Lu and Arkun, 2000).

Wada et al. (2006) proposed on-line MPC for LPV systems using parameter-dependent Lyapunov function. At each sampling instant, the ellipsoidal invariant set containing the measured state is constructed so robust stability is guaranteed. Since the optimization problem has to be solved on-line at each sampling instant, the algorithm requires a relatively high computational effort.

Some researchers have proposed a dual-mode MPC for LPV systems (Casavola et al., 2002; Bumroongsri and Kheawhom, 2012a). The control law has the form u = Kx + c for the first *N* steps and u = Kx for the rest of the infinite horizon. Although the degrees of freedom are increased, larger on-line

computational time is required because the size of on-line optimization problem grows significantly with respect to *N*.

In order to reduce on-line computational time, off-line formulation of MPC have been proposed (Wan and Kothare, 2003; Bumroongsri and Kheawhom, 2012c). A sequence of state feedback gains corresponding to a sequence of invariant sets is pre-computed off-line. At each sampling instant, the real-time state feedback gain is calculated by linear interpolation between the pre-computed state feedback gains. Although the on-line computational time is significantly reduced, the conservativeness can be obtained in control of LPV systems because the scheduling parameter is not included in the controller design.

Off-line MPC for LPV systems was proposed by Bumroongsri and Kheawhom (2012b). The sequences of state feedback gains corresponding to the sequences of ellipsoids are pre-computed off-line. At each sampling instant, the scheduling parameter is measured and the smallest ellipsoid containing the measured state is approximated. The corresponding real-time state feedback gain is then calculated by linear interpolation between the pre-computed state feedback gains. The ellipsoid computed at each sampling instant is only an approximation so the algorithm sacrifices optimality in order to reduce on-line computational time. To improve the control performances of off-line MPC algorithm, an interpolation technique has been introduced (Kheawhom and Bumroongsri, 2013; Bumroongsri and Kheawhom, 2013.)

In this paper, interpolation-based off-line MPC for LPV systems is presented. Four interpolation techniques based on different ideas are proposed. The aim is to develop new interpolation techniques that can achieve good control performance while robust stability is still guaranteed.

The paper is organized as follows. In section 2, the problem description is presented. In section 3, interpolation-based offline MPC for LPV systems is presented. In section 4, we present an example to illustrate the implementation of the proposed techniques. Finally, in section 5, we conclude the paper.

Notation: For a matrix A, A^T denotes its transpose, A^{-1} denotes its inverse. I denotes the identity matrix. For a vector x, x(k/k) denotes the state measured at real time k, x(k+i/k) denotes the state at prediction time k+i predicted at real time k. The symbol * denotes the corresponding transpose of the lower block part of symmetric matrices.

2. PLOBLEM DESCRIPTION

The model considered here is the following discrete-time LPV system:

$$x(k+1) = A(p(k))x(k) + Bu(k)$$

$$y(k) = Cx(k)$$
(1)

where x(k) is the state of the plant and u(k) is the control input. We assume that the scheduling parameter p(k) is measurable on-line at each sampling time. Moreover, we assume that

$$A(p(k)) \in \Omega, \ \Omega = Co\{A_1, A_2, \dots, A_L\}$$
(2)

where Ω is the polytope, *Co* denotes the convex hull, A_j are the vertices of the convex hull. Any A(p(k)) within the polytope Ω is a linear combination of the vertices such that

$$A(p(k)) = \sum_{j=1}^{L} p_j(k) A_j, \sum_{j=1}^{L} p_j(k) = 1, 0 \le p_j(k) \le 1$$
(3)

The aim of this research is to find the state feedback control law

$$u(k) = K(p(k))x(k)$$
(4)

that stabilizes (1) and satisfies the input and output constraints

$$|u_h(k+i/k)| \le u_{h,\max}, h = 1, 2, 3, \dots, n_u$$
(5)

$$|y_r(k+i/k)| \le y_{r,\max}, r = 1,2,3,...,n_y$$

(6)

Wada et al. (2006) proposed on-line MPC for LPV systems using parameter-dependent Lyapunov function. At each sampling instant, the state feedback control law which minimizes the upper bound γ on the following worst-case performance cost

$$\min_{u(k+i/k)} \max_{A(p(k+i))\in\Omega, i\geq 0} J_{\infty}(k)$$

$$J_{\infty}(k) = \sum_{i=0}^{\infty} \begin{bmatrix} x(k+i/k) \\ u(k+i/k) \end{bmatrix}^{T} \begin{bmatrix} \Theta & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} x(k+i/k) \\ u(k+i/k) \end{bmatrix}$$
(7)

where Θ and *R* are weighting matrices, and asymptotically stabilizes the discrete-time LPV system (1) is given by $u(k) = K(p(k))x(k), K(p(k)) = \sum_{j=1}^{L} p_j(k)K_j, K_j = Y_jG_j^{-1}$ where Y_j and G_j are obtained by solving the following problem

$$\min_{\gamma, Y_j, G_j, Q_j} \gamma \tag{8}$$

s.t.
$$\begin{bmatrix} 1 & x(k/k)^T \\ x(k/k) & Q_j \end{bmatrix} \ge 0, \forall j = 1, 2, 3, \dots, L$$
(9)

$$\begin{bmatrix} G_{j} + G_{j}^{T} - Q_{j} & * & * & * \\ A_{j}G_{j} + BY_{j} & Q_{l} & * & * \\ \Theta^{\frac{1}{2}}G_{j} & 0 & \gamma I & * \\ \frac{1}{R^{\frac{1}{2}}Y_{j}} & 0 & 0 & \gamma I \end{bmatrix} \geq 0,$$
(10)
$$\forall j = 1, 2, 3, \dots, L, \forall l = 1, 2, 3, \dots, L$$

$$\begin{bmatrix} X & Y_j \\ Y_j^T & G_j + G_j^T - Q_j \end{bmatrix} \ge 0,$$
(11)
$$\forall j = 1, 2, 3, \dots, L, \ X_{hh} \le u_{h, \max}^2, \ h = 1, 2, 3, \dots, n_u$$

$$\begin{bmatrix} S & * \\ (A_j G_j + BY_j)^T C^T & G_j + G_j^T - Q_j \end{bmatrix} \ge 0,$$

$$\forall j = 1, 2, 3, \dots, L, \ S_{rr} \le y_{r, \max}^2, r = 1, 2, 3, \dots, n_y$$
(12)

Since the optimization problem has to be solved on-line at each sampling instant, the algorithm requires a relatively high computational effort.

3. INTERPOLATION-BASED OFF-LINE MPC

In this section, interpolation-based off-line MPC for LPV systems is presented. The sequences of state feedback gains

 $K_{i,j}$ corresponding to the sequences of ellipsoids $\varepsilon_{i,j} = \{x/x^T Q_{i,j}^{-1} x \le 1\}$ are pre-computed off-line where i = 1, 2, 3, ..., N denote the number of ellipsoids in each sequence and j = 1, 2, 3, ..., L denote the vertices of the polytope Ω . At each sampling time, the real-time state feedback gain is calculated by linear interpolation between the pre-computed state feedback gains.

3.1 Interpolation-based off-line MPC

Off-line: Choose a sequence of states x_i , i = 1, 2, 3, ..., N. For each x_i , substitute x(k/k) in (9) by x_i and solve the optimization problem (8) to obtain the corresponding state feedback gain $K_{i,j} = Y_{i,j}G_{i,j}^{-1}$ and ellipsoids $\varepsilon_{i,j} = \left\{x/x^T Q_{i,j}^{-1}x \le 1\right\}$. Note that x_i should be chosen such that $\varepsilon_{i+1,j} \subset \varepsilon_{i,j}$. Moreover, for each $i \ne N$, the inequality $Q_{i,j}^{-1} - (A_j + BK_{i+1,j})^T Q_{i,l}^{-1}(A_j + BK_{i+1,j}) > 0, \forall j = 1,2,3,...,L$, $\forall l = 1,2,3,...,L$ must be satisfied.

On-line: The real-time state feedback gain is calculated by linear interpolation between the pre-computed state feedback gains. Four interpolation techniques are proposed as follows

Technique 1: (Bumroongsri and Kheawhon, 2012b) The first technique is based on an approximation of the smallest ellipsoid containing the measured state. At each sampling time, when x(k) satisfies $x(k) \in \varepsilon_{i,j}$, $x(k) \notin \varepsilon_{i+1,j}$, $\forall j = 1,2,3,...,L$, $i \neq N$, the real-time state feedback gain $K(\lambda(k)) = \lambda(k) [\sum_{i=1}^{L} p_j(k) K_{i,j}] + (1 - \lambda(k)) [\sum_{i=1}^{L} p_j(k) K_{i+1,j}]$

can be calculated from $\lambda(k) \in (0,1]$ obtained by solving

$$x(k)^{T}(\lambda(k)[\sum_{j=1}^{L}p_{j}(k)Q_{i,j}^{-1}] + (1 - \lambda(k))[\sum_{j=1}^{L}p_{j}(k)Q_{i+1,j}^{-1}])x(k) = 1$$
(13)

It is seen that $\lambda(k) = 0$ and $\lambda(k) = 1$ correspond to the ellipsoids $\varepsilon_{i+1,j}$ and $\varepsilon_{i,j}$, respectively. In this technique, no optimization problem is needed to be solved on-line. Figure 1 shows the graphical representation of the state feedback gain in each prediction horizon. It is seen that the state feedback gain $K(\lambda(k))$ is implemented throughout the prediction horizon. Thus, the state must be restricted to lie in the smallest ellipsoid approximated by (13) and robust stability is guaranteed.

Technique 2: In the second technique, the pre-computed state feedback gains $K_{i,j}$ are interpolated in order to get the largest possible real-time state feedback gain while robust stability is still guaranteed. At each sampling time, when x(k) satisfies $x(k) \in \varepsilon_{i,j}$, $x(k) \notin \varepsilon_{i+1,j}$, $\forall j = 1,2,3,...,L$,



Fig.1. The graphical representation of the state feedback gain in each prediction horizon of technique 1.



Fig.2. The graphical representation of the state feedback gain in each prediction horizon of technique 2.

$$i \neq N$$
, the real-time state feedback gain
 $K(\alpha(k)) = \alpha(k) \left[\sum_{j=1}^{L} p_j(k) K_{i,j}\right] + (1 - \alpha(k)) \left[\sum_{j=1}^{L} p_j(k) K_{i+1,j}\right]$

can be calculated from $\alpha(k)$ obtained by solving the following problem

$$\min \alpha(k) \tag{14}$$

s.t.

$$\begin{bmatrix} 1 & \left(\left(A(p(k)) + BK(\alpha(k))\right)x(k)\right)^{T} \\ \left(A(p(k)) + BK(\alpha(k))\right)x(k) & Q_{i,j} \end{bmatrix} \ge 0,$$

$$j = 1, 2, 3, \dots, L$$
(15)

$$\begin{bmatrix} u_{h,\max}^2 & *\\ (K(\alpha(k))x(k))_h & 1 \end{bmatrix} \ge 0$$
(16)

$$0 \le \alpha(k) \le 1 \tag{17}$$

 $K_{i+1,j}$ is always larger than $K_{i,j}$ because input and output constraints impose less limit on the state feedback gain as *i* increases. Thus, the largest possible real-time state feedback gain $K(\alpha(k))$ can be calculated by minimizing $\alpha(k)$ in (14) while robust stability is still guaranteed by (15). The input constraint is guaranteed by (16).

Figure 2 shows the graphical representation of the state feedback gain in each prediction horizon. It is seen that the largest possible real-time state feedback gain $K(\alpha(k))$ is only implemented at each sampling time k. At time k+1 and so on, the state feedback gain $K_i = \sum_{j=1}^{L} p_j(k) K_{i,j}$ is implemented. Thus, the state must be restricted to lie in the

ellipsoids $\varepsilon_{i,j}$ and robust stability is guaranteed.

Technique 3: In the third technique, the real-time state feedback gain is calculated by minimizing the violation of the constraints of the adjacent inner ellipsoids so the real-time state feedback gain calculated has to regulate the state from the current ellipsoids $\varepsilon_{i,j}$ to the adjacent inner ellipsoids

$$\begin{split} \varepsilon_{i+1,j} & \text{as fast as possible. At each sampling time, when } x(k) \\ \text{satisfies} & x(k) \in \varepsilon_{i,j}, \ x(k) \notin \varepsilon_{i+1,j}, \ \forall j = 1,2,3,...,L, \ i \neq N, \\ \text{the real-time} & \text{state} & \text{feedback} \\ \text{gain} \\ & K(\delta(k)) = \delta(k) [\sum_{j=1}^{L} p_j(k) K_{i,j}] + (1 - \delta(k)) [\sum_{j=1}^{L} p_j(k) K_{i+1,j}] \end{split}$$

can be calculated from $\delta(k)$ obtained by solving the following problem.

$$\min \sigma(k) \tag{18}$$

s.t.
$$\begin{bmatrix} 1 + \sigma(k) & ((A(p(k)) + BK(\delta(k)))x(k))^T \\ (A(p(k)) + BK(\delta(k)))x(k) & Q_{i+1,j} \end{bmatrix} \ge 0,$$

$$j = 1,2,3,...,L \tag{19}$$

$$\begin{bmatrix} 1 & ((A(p(k)) + BK(\delta(k)))x(k))^T \\ (A(p(k)) + BK(\delta(k)))x(k) & Q_{i,j} \end{bmatrix} \ge 0, \\ j = 1, 2, 3, ..., L$$

(20)

$$\begin{bmatrix} u_{h,\max}^2 & *\\ (K(\delta(k))x(k))_h & 1 \end{bmatrix} \ge 0$$
(21)

$$0 \le \delta(k) \le 1 \tag{22}$$

By applying Schur complement to (19), we obtain $x(k+1)^T Q_{i+1,j}^{-1} x(k+1) \le 1 + \sigma(k)$. By minimizing $\sigma(k)$ in (18), the real-time state feedback gain $K(\delta(k))$ calculated has to regulate the state from the current ellipsoids $\varepsilon_{i,j}$ to the adjacent inner ellipsoids $\varepsilon_{i+1,j}$ as fast as possible. Robust stability is guaranteed by (20). The input constraint is guaranteed by (21).

Figure 3 shows the graphical representation of the state feedback gain in each prediction horizon. It is seen that the real-time state feedback gain $K(\delta(k))$ calculated is only implemented at each sampling time k. At time k+1 and so on, the state feedback gain $K_i = \sum_{j=1}^{L} p_j(k) K_{i,j}$ is implemented. Thus, the state must be restricted to lie in the

ellipsoids $\varepsilon_{i,j}$ and robust stability is guaranteed.

Technique 4: In the last technique, the real-time state feedback gain is calculated by minimizing the one-step cost function so the real-time state feedback gain calculated has to regulate the next predicted state to the origin as fast as possible. At each sampling time, when the measured state x(k) satisfies $x(k) \in \varepsilon_{i,j}$, $x(k) \notin \varepsilon_{i+1,j}$, $\forall j = 1, 2, ..., L$, $i \neq N$, the real-time state feedback gain

$$K(\beta(k)) = \beta(k) \left[\sum_{j=1}^{L} p_j(k) K_{i,j}\right] + (1 - \beta(k)) \left[\sum_{j=1}^{L} p_j(k) K_{i+1,j}\right]$$

can be calculated from $\beta(k)$ obtained by solving the following problem.

$$\min J_k$$
 (23)

S.t.

$$\begin{bmatrix} J_k & \left(\left(A(p(k)) + BK(\beta(k)) \right) x(k) \right)^T \\ \left(A(p(k)) + BK(\beta(k)) \right) x(k) & \Theta^{-1} \end{bmatrix} \ge 0,$$
(24)

$$\begin{bmatrix} 1 & ((A(p(k)) + BK(\beta(k)))x(k))^T \\ (A(p(k)) + BK(\beta(k)))x(k) & Q_{i,j} \end{bmatrix} \ge 0,$$

 $j = 1, 2, \dots, L$
(25)

$$\begin{bmatrix} u_{h,\max}^2 & *\\ (K(\beta(k))x(k))_h & 1 \end{bmatrix} \ge 0$$
(26)

$$0 \le \beta(k) \le 1 \tag{27}$$

By applying Schur complement to (24), we obtain $J_k \ge x(k+1)^T \Theta x(k+1)$ where Θ is the weighting matrix. Thus, J_k in (23) is the one-step cost function. Robust stability is guaranteed by (25). The input constraint is guaranteed by (26).

Figure 4 shows the graphical representation of the state feedback gain in each prediction horizon. It is seen that the real-time state feedback gain $K(\beta(k))$ calculated is only implemented at each sampling time k. At time k+1 and so on, the state feedback gain $K_i = \sum_{i=1}^{L} p_j(k) K_{i,j}$ is

implemented. Thus, the state must be restricted to lie in the ellipsoids $\varepsilon_{i,i}$ and robust stability is guaranteed.



Fig.3. The graphical representation of the state feedback gain in each prediction horizon of technique 3.



Fig.4. The graphical representation of the state feedback gain in each prediction horizon of technique 4.

4. EXAMPLE

In this section, we present an example that illustrates the implementation of the proposed off-line MPC algorithm. The numerical simulations have been performed in Intel Core i-5 (2.4GHz), 2 GB RAM, using SeDuMi (Sturm, 1998) and YALMIP (Löfberg, 2004) within Matlab R2008a environment. We will consider the application of our approach to the following nonlinear model for CSTR where the consecutive reaction $A \longrightarrow B \longrightarrow C$ takes place

$$\begin{bmatrix} \vdots \\ x_1 \\ \vdots \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 - Da_1 & 0 \\ Da_1 & -1 - Da_2 x_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \quad (28)$$

where x_1 denotes the dimensionless concentration of A and x_2 denotes the dimensionless concentration of B. The control variable u corresponds to the inlet concentration of A. The operating parameters are $Da_1=1$ and $Da_2=2$.

Let $\overline{x_1} = x_1 - x_{1,eq}$, $\overline{x_2} = x_2 - x_{2,eq}$ and $\overline{u} = u - u_{eq}$ where subscript eq is used to denote the corresponding variable at the equilibrium condition, the input and output constraints are given as

$$\left| \overline{x}_1 \right| \le 0.5, \left| \overline{x}_2 \right| \le 0.5, \left| \overline{u} \right| \le 0.5$$
 (29)

By evaluating the Jacobian matrix of (28) along the vertices of the constraints set (29), we have that all the solutions of (28) are also the solutions of the following differential inclusion

$$\begin{bmatrix} \vdots \\ x_1 \\ \vdots \\ x_2 \end{bmatrix} \in \left(\sum_{j=1}^2 p_j A_j \right) \begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \overline{u}$$
(30)

where A_j , j = 1,2 are given by

$$A_{1} = \begin{bmatrix} -1 - Da_{1} & 0 \\ Da_{1} & -1 - Da_{2}x_{2,\min} \end{bmatrix}, A_{2} = \begin{bmatrix} -1 - Da_{1} & 0 \\ Da_{1} & -1 - Da_{2}x_{2,\max} \end{bmatrix}$$
(31)

and p_i , j = 1,2 are given by

$$p_1 = \frac{x_{2,\max} - x_2}{x_{2,\max} - x_{2,\min}}, p_2 = \frac{x_2 - x_{2,\min}}{x_{2,\max} - x_{2,\min}}$$
(32)

The discrete-time model is obtained by discretization of (30) using Euler first-order approximation with a sampling period of 0.1 min and it is omitted here for brevity. The weighting matrices are $O_{11} \begin{bmatrix} 1 & 0 \\ 0 \end{bmatrix}$ and B = 0.01

matrices are
$$\Theta = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
 and $R = 0.01$

Figure 5 shows two sequences of ellipsoids constructed offline. Each sequence has three ellipsoids ($\mathcal{E}_{i,j}$, i = 3, j = 2). In this example, two sequences of ellipsoids are constructed because the polytope Ω has two vertices.



Fig. 5. Two sequences of ellipsoids constructed off-line.

Figure 6 shows the closed-loop responses of the system. In technique 1, the smallest ellipsoid containing the measured state is approximated at each sampling instant and the corresponding real-time state feedback gain is calculated. Since the same real-time state feedback gain $K(\lambda(k))$ is implemented throughout the prediction horizon as shown in Fig.1, technique 1 tends to produce relatively slow responses compared to other techniques. In technique 2, the precomputed state feedback gains are interpolated in order to get the largest possible real-time state feedback gain $K(\alpha(k))$. At each sampling time, the largest possible real-time state feedback gain is implemented as shown in Fig. 2 so technique 2 tends to produce less sluggish responses than technique 1. In this example, technique 2 gives 0.5% better performance cost (7) compared to technique 1. In technique 3, the real-time state feedback gain is calculated by minimizing the violation of the constraints of the adjacent inner ellipsoids. At each sampling time, the real-time state feedback gain $K(\delta(k))$ is implemented as shown in Fig. 3 so the state has to be regulated from the current ellipsoids $\varepsilon_{i,i}$

to the adjacent inner ellipsoids $\varepsilon_{i+1,j}$ as fast as possible. In this example, technique 2 and technique 3 behave almost identically in regulating the output. In technique 4, the real-time state feedback gain is calculated by minimizing the one-

step cost function so the real-time state feedback gain calculated has to regulate the next predicted state to the origin as fast as possible. As shown in Fig. 6, technique 4 tends to produce the fastest responses among all techniques. In this example, technique 4 gives 0.6% better performance cost (7) compared to technique 1.



Fig. 6. The closed-loop responses.

Table 1. The on-line computational time.

Algorithm	CPU time (s)
Technique 1	0.001
Technique 2	0.047
Technique 3	0.101
Technique 4	0.075

Table 1 shows the on-line computational time. It is seen that technique 1 has the smallest on-line computational time because no optimization problem is needed to be solved on-line. In comparison, technique 3 has the largest on-line computational time because many LMIs constraints are involved in the on-line optimization problem.

5. CONCLUSIONS

In this paper, we have presented interpolation-based off-line MPC for LPV systems. The sequences of state feedback gains are pre-computed off-line. The real-time state feedback gain is calculated by linear interpolation between the pre-

computed state feedback gains. Four interpolation techniques are presented. It is shown that the proposed techniques give better control performance than the old technique.

ACKNOWLEDGEMENTS

This work was supported by Rajadapisek Sompoj Fund, Chulalongkorn University Centenary Academic Development Project, and the Thailand Research Fund (TRF).

REFERENCES

- Bumroongsri, P. and Kheawhom, S. (2012a). MPC for LPV systems based on parameter-dependent Lyapunov function with perturbation on control input strategy. *Engineering Journal*, 16, 61-72. doi: 10.4186/ej.2012.16.2.61.
- Bumroongsri, P. and Kheawhom, S. (2012b). An ellipsoidal off-line model predictive control strategy for linear parameter varying systems with applications in chemical processes. *Syst. Control Lett.*, 61, 435-442.
- Bumroongsri, P. and Kheawhom, S. (2012c). An off-line robust mpc algorithm for uncertain polytopic discretetime systems using polyhedral invariant sets. J. Process Contr., 22, 975-983.
- Bumroongsri, P. and Kheawhom, S. (2013). An interpolationbased robust mpc algorithm using polyhedral invariant sets. In Proc. of the 2013 European Control Conference (ECC), 3161-3166.
- Casavola, A. Famularo, D., and Franze, G. (2002). A feedback min-max MPC algorithm for LPV systems subject to bounded rates of change of parameters. *IEEE T. Automat. Contr.*, 47, 1147-1152.
- Kheawhom, S. and Bumroongsri, P. (2013). Constrained robust model predictive control based on polyhedral invariant sets by off-line optimization. *Chemical Engineering Transactions*, 32, 1417-1442. doi:10.3303/CET1332237.
- Lu, Y., and Arkun, Y. (2000). Quasi-min-max MPC algorithms for LPV systems. *Automatica*, 36, 527-540.
- Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. *in Proc. IEEE international symposium on computer aided control systems design*, 284-289.
- Morari, M., and Lee, J.H. (1999). Model predictive control: past, present and future. *Comput. Chem. Eng.*, 23, 667-682.
- Sturm, J.F. (1998). Using Sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Method Softw.*, 11, 625-653.
- Wada, N., Saito, K., and Saeki, M. (2006). Model predictive control for linear parameter varying systems using parameter dependent lyapunov function. *IEEE T. Circuits Syst.*, 53, 1446-1450.
- Wan, Z., and Kothare, M.V., (2003). An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica*, 39, 837-846.