Journal of Process Control 14 (2004) 231–247

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

A new robust model predictive control method I: theory and computation

Yiyang Jenny Wang , James B. Rawlings*

Department of Chemical Engineering, University of Wisconsin-Madison, WI 53706-1691, USA

Received 28 January 2002; received in revised form 14 November 2002; accepted 14 November 2002

Abstract

In this paper, we propose a new robust MPC method that guarantees stability and offset-free set point tracking in the presence of model uncertainty. A *min-max* optimization problem that explicitly accounts for model uncertainty is used to determine the optimal control action subject to the input and output constraints. The robust regulator uses a tree trajectory to forecast the time-varying model uncertainty. The controller design procedure uses integrators to reject non-zero disturbances and maintain the process at the optimal operating conditions (set points). Constraints may cause offset, which occurs when the set points are unreachable. In the feasible region where constraints are not active, the robust MPC theory we propose achieves offset-free non-zero set point tracking if there exists a control policy that robustly stabilizes all models in the uncertainty set.

Keywords: Robust model predictive control; Tracking; Offset-free control; LMI

1. Introduction

Model predictive control (MPC), also known as receding horizon control or moving horizon control, is a class of model-based control theories that use linear or nonlinear process models to forecast system behavior. The success of the MPC control performance depends on the accuracy of the open-loop predictions, which in turn depend on the accuracy of the process models. It is possible for the predicted trajectory to differ, perhaps considerably, from the actual plant behavior [1]. The difference between the plant and the model is known as plant-model mismatch, which can cause the control performance to be sluggish, overly conservative or, in the worst-case scenario, unstable. Because process identification is difficult at best, process modeling errors occur frequently.

Although the feedback in MPC reduces the impact of the discrepancy between the plant and the forecasted behavior, MPC is not designed to explicitly handle plant-model mismatch. Robust model predictive control (RMPC) is the class of predictive control methods that increases the effectiveness of the controller by explicitly accounting for the modeling errors in the control design procedure. Instead of forecasting system behavior using one process model as in MPC, RMPC forecasts system behavior for every model in the uncertainty set. Two common descriptions used to account for model uncertainty are parameter uncertainties and bounded exogenous disturbances. The optimal control actions are determined by a *min–max* optimization that minimizes the deviations of the forecasted behavior from the desired behavior for the model with the largest deviation. The *min–max* optimization originated from research in dynamic game theory, and the terminology of ‘‘minimax controller’’ was adopted from the statistical decision theory of the 1950s [2]. Campo and Morari [3], Zafiriou [4], Genceli and Nikolaou [5], and Zheng and Morari [6] proposed using the *min–max* optimization in MPC with finite impulse response models. Lee and Cooley [7,8] proposed using the min–max optimization in MPC on state-space models with time-varying parametric uncertainty in the B matrix. Lee and Yu [9] proposed using the *min–max* optimization on discrete state-space models with polytopic model uncertainty. The on-line *min–max* optimization is computationally intensive.

Kothare et al. [10], Casavola et al. [11], and Lu and Arkun [12,13] proposed using linear matrix inequalities (LMI) to solve for the gain in the state feedback control policy $u(t) = Kx(t)$ that minimizes the model in the

* Corresponding author. Fax: +1-608-265-8794.
E-mail address: jbraw@bevo.che.wisc.edu (J.B. Rawlings).

0959-1524/03/$ - see front matter
doi:10.1016/S0959-1524(02)00133-6

polytope with the largest deviation from the origin. The LMI-based optimization is performed on-line to determine a new K at every time step.

Badgwell [14,15] proposed using a nominal model to determine the optimal control action. Robust stability is achieved by adding the following constraint to the sum of the forecasted behavior's deviations from the origin:

$$\Phi(t+1) \leqslant \Phi(t) \tag{1}$$

for all models in the polytope. Similarly, Slupphaug and Foss [16] used a nominal model to determine the optimal control actions while guaranteeing closed-loop stability for all models in the polytope by adding the constraint

$$\|x(t+1)\|^2 \leqslant \|x(t)\|^2. \tag{2}$$

Pannocchia and Semino [17,18] proposed modifying the nominal model to robustly stabilize the system for the given model uncertainty description. Chisci et al. [19,20], Lee and Kouvaritakis [21] and Kouvaritakis et al. [22] proposed using the linear state feedback control policy in an invariant terminal region. A nominal model is used to determine the N control moves needed to move the forecasted plant behavior into the terminal region.

When the output set point is non-zero, besides determining if a robustly stabilizing control policy exists for convergence to the origin, RMPC needs to determine if the control policy can achieve non-zero set point tracking. Khammash and Zou [23] and Bemporad and Mosca [24] proposed using an input reference trajectory to achieve offset-free set point tracking. If the input reference trajectory is unknown, Bemporad [25], Christofides [26], Freeman and Kokotovic [27], and Blanchini [28] showed that if the model uncertainty is described by a nominal model with bounded disturbances, offset-free non-zero set point tracking can be achieved if the disturbances converge to zero. Rossiter and Kouvaritakis [29], Lee et al. [30], Rodrigues and Odloak [31], and Megias et al. [32] proposed using models in the velocity form and minimizing the sum of $y(t) - y_t$ and $u(t+1) - u(t)$ with the terminal constraint $u(t+j+1) - u(t+j) = 0$ for $j \geqslant N$. Pannocchia and Rawlings [33] showed the steady state achieved by the above controller depends on the horizon length N when using process models in the velocity form.

Marconi and Isidori [34] stated that correct steady-state x_s and u_s values are needed to achieve "perfect tracking" of the set point. Kassmann et al. [35] proposed using a nominal model in the velocity form to compute the steady-state x_s and u_s values. Ralhan and Badgwell [36] proposed adding an estimated output disturbance to the process model to achieve offset-free control to the zero set point but do not discuss non-zero set point tracking.

In this paper, we propose a new RMPC method that achieves non-zero set point tracking for a process with time-varying model uncertainty and input and output constraints. Scokaert and Mayne [37] proposed using a time-varying sequences of bounded disturbances to model the uncertainty. We propose using time-varying sequences of models in a polytope to forecast the model uncertainties. Integral control is achieved by adding state disturbances to all models on the vertices of the polytope. Offset-free non-zero set point tracking is achieved by computing the steady-state x_s and u_s values that minimize the sum of the deviations of the predicted steady-state output from the set point for all models in the uncertainty description. The values of the steady-state biases are chosen so that all models in the uncertainty description have the same x_s and u_s values. Because every model in the polytope has the same x_s and u_s values, offset-free control can be accomplished by designing a robust control policy that controls the state to the same x_s for all models in the uncertainty description.

2. Problem statement

The uncertain dynamic system is described by the following discrete state-space model:

$$\begin{aligned} x(t+1) &= A(t)x(t) + B(t)u(t) \\ y(t+1) &= Cx(t+1) \end{aligned} \tag{3}$$

in which $(A(t), B(t))$ are the time-varying state-space model matrices; $C \in \mathbb{R}^{p\times n}$ describes the relationship between the output and the state; $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are the state, the manipulated input, also known as the control decision variable, and the controlled output vectors, respectively. The controlled output and the manipulated input are subject to the following constraints:

$$u(t) \in \mathbb{U}, \quad x(t) \in \mathbb{X}, \quad y(t) \in \mathbb{Y} \tag{4}$$

in which $\mathbb{U}$, $\mathbb{X}$, and $\mathbb{Y}$ are convex and closed subsets of $\mathbb{R}^m$, $\mathbb{R}^n$, and $\mathbb{R}^p$, respectively.

When model uncertainty is present, the exact plant model $(A(t), B(t))$ is not known.

The model uncertainty region is described by Ω, the convex hull of $\Pi = \{(A_1, B_1), (A_2, B_2), \ldots, (A_1, B_1)\}$. The convex hull is defined as the linear convex combination of all models in Π. Let $i \in \mathbb{I}$ be the model index. $(A(t), B(t))$ is in Ω if and only if there exist $\mu_1(t), \mu_2(t), \ldots, \mu_1(t) \in \mathbb{R}$ such that

$$A(t) = \sum_{i=1}^{\mathcal{I}} \mu_i(t) A_i \quad \text{and} \quad B(t) = \sum_{i=1}^{\mathcal{I}} \mu_i(t) B_i \tag{5}$$

for any

$$0 \leqslant \mu_i(t) \leqslant 1 \quad \text{and} \quad \sum_{i=1}^{\mathcal{I}} \mu_i(t) = 1 \tag{6}$$

The system is said to be at steady state at time T if

$$\begin{aligned} u_s &= u(s+1) = u(s) \quad \text{for all } s \geqslant T \\ x_s &= x(s+1) = x(s) \quad \text{for all } s \geqslant T \\ y_s &= y(s+1) = y(s) \quad \text{for all } s \geqslant T \end{aligned} \tag{7}$$

in which $u_s \in \mathbb{R}^m$, $x_s \in \mathbb{R}^n$, and $y_s \in \mathbb{R}^p$ are the steady-state control, state, and controlled output vectors that satisfy the hard constraints

$$u_s \in \mathbb{U}, \;\; x_s \in \mathbb{X}, \;\; y_s \in \mathbb{Y} \tag{8}$$

Because there is no uncertainty in the C matrix,

$$y_s = Cx_s \tag{9}$$

at steady state.

Of the many model uncertainty descriptions used for robust control, we choose the polytopic model uncertainty description because it is one of the more versatile model uncertainty descriptions. It can be used to approximate ellipsoidal parametric model uncertainty as well as bounded disturbance uncertainties. See [38] for more details.

3. Definitions and notations

3.1. System theory

D1. A function $\alpha : \mathbb{X} \to \mathbb{R}$ in which $\mathbb{X} \subset \mathbb{R}^n$ is positive definite if $\alpha(x) > 0$ for all $x \in \mathbb{X}$, $x \neq 0$ and $\alpha(0) = 0$.

D2. A function $\alpha : \mathbb{X} \to \mathbb{R}_+$ in which $\mathrm{X} \subset \mathrm{R}^n$ is said to be proper if it is continuous, positive definite, strictly increasing, and $\alpha(s) \to \infty$ as $s \to \infty$.

D3. The symbol $x^{(r)}$ is denotes the rth element of the vector $x \in \mathbb{R}^n$.

D4. A control law is a function $h(\cdot) : \mathbb{X} \to \mathbb{U}$. A control policy is a sequence of control laws.

D5. The system $x(s+1) = A_i x(s) + B_i u(s)$, $y(s) = Cx(s)$ is said to be admissible if it satisfies the state and control constraints $x(s) \in \mathbb{X}$ and $u(s) \in \mathbb{U}$ for all $s \geqslant 0$ and for all models $i \in \{1, \ldots, \mathcal{I}\}$. A control sequence $\{u(s), u(s+1), \ldots\}$ is said to be admissible (for a given initial state) if the associated system is admissible and is said to be feasible if it satisfies all the constraints in an associated optimal control problem.

D6. The system $x(s+1) = Ax(s) + Bu(s)$ in which (A, B) is time invariant is said to be controllable if there exists a positive integer n_c and a proper function W_c such that for every system satisfying $x(0) = x$ and $(x(s), u(s)) = (0, 0)$ for all $s \geqslant n_c$.

$$\sum_{s=0}^{n_c-1} \|(x(s), u(s))\| \leqslant W_c(\|x\|) \tag{10}$$

3.2. The new robust regulator

We require some terminology to describe the level, node, and branch of the tree trajectory used by the robust regulator to forecast the system behavior with time-varying model uncertainty.

D7. The integer $N \in \mathbb{I}$ denotes the finite prediction horizon length.

D8. The integer $j \in \mathbb{I}$ denotes the tree trajectory level which is equal to the number of steps into the forecast horizon, $j = 0, 1, \ldots, N$.

D9. The integer string $\tau \in \mathbb{I}^j$ is the node index at level j that keeps track of the sequence of models needed to arrive at the node τ.

D10. The branch index $l \in \mathbb{I}$ denotes the collection of nodes for one possible time-varying combination of (A_i, B_i) in Π.

D11. Let $a \in \mathbb{I}$ and $i \in \mathbb{I}$ be integers and $\tau \in \mathbb{I}^a$ be a node. The operator $\mathbf{P}$ is defined as

$$\mathbf{P}(\tau, i) = \tau \tag{11}$$

in which $(\tau, i) \in \mathbb{I}^{a+1}$ is a node index of length $a+1$. The operator $\mathbf{P}$ removes the last integer in the node index (τ, i).

D12. The unit integer string $1_N \in \mathbb{I}^N$ is defined as

$$1_N = \underbrace{1, 1, 1, \ldots, 1}_{N \text{ times}} \tag{12}$$

D13. If $a \in \mathbb{I}$ and $b \in \mathbb{I}$, then $(a)_b$ is defined as the number a in base b.

D14. Let $a \in \mathbb{I}^n$ and $b \in \mathbb{I}^m$ be integer strings of length n and m, respectively, with $n \geqslant m$. If $a = a_n, a_{n-1}, \ldots, a_2, a_1$ and $b = b_m, b_{m-1}, \ldots, b_2, b_1$, then the operator $\oplus$ is defined as

$$a \oplus b = a \oplus \bar{b} = a_n + \bar{b}_n, a_{n-1} + \bar{b}_{n-1}, \ldots, a_2 + \bar{b}_2, a_1 + \bar{b}_1 \tag{13}$$

in which $\bar{b} = 0, \ldots, 0, b_m, \ldots, b_1$ such that $\bar{b} \in \mathbb{I}^n$.

D15. If $a \in \mathbb{I}^n$ and $b \in \mathbb{I}^m$ are the node indices of length n and m, respectively, then the operator ; separates node a from node b in the set $\{a; b\}$.

D16. The node index $\tau_l \in \mathbb{I}^N$ denotes the terminal node of branch l and is defined as

$$\tau_l = (l-1)_{\mathcal{I}} \oplus \mathbf{1}_N \tag{14}$$

D17. The set T_l contains all the nodes on branch l and is defined as

$$T_l = \left\{\tau_l; \mathbf{P}(\tau_l); \mathbf{P}^2(\tau_l); \ldots; \mathbf{P}^{N-1}(\tau_l)\right\} \tag{15}$$

D18. The set $\bar{T}_l$ is a subset of T_l and is defined as

$$\bar{T}_l = T_l / \tau_l \tag{16}$$

which implies $\tau_l \not\subseteq T_l \cap \bar{T}_l$. The set $\bar{T}_l$ contains all the nodes in T_l except for the terminal node τ_l.

4. The new robust regulator design

The regulator determines the optimal manipulated input trajectory based on the system behavior predictions of a model. To correctly model the time-varying uncertainty, we propose a method similar to the one proposed by Scokaert and Mayne [37] that produces a tree trajectory for all possible time-varying combinations of the models in Π. Scokaert and Mayne [37] proposed using a disturbance model to model uncertainty. This proposal uses parametric uncertainty in the state-space models to generate the tree trajectory. The components of the tree trajectory are: level, node, and branch. The tree trajectory is rooted at $x(t)$ and $u(t)$ in the state and input space, respectively. If τ is the node index at level j, then (τ, i) is the node index at level $j+1$ under the ith model forecast.

$$\begin{aligned} x_{\tau,i}(t) &= A_i x_\tau(t) + B_i u_\tau(t) \ \text{ for } i = 1, \ldots, \mathcal{I} \\ y_{\tau,i}(t) &= C x_{\tau,i}(t) \end{aligned} \tag{17}$$

in which $x_\tau(t) \in \mathbb{R}^n$, $u_\tau(t) \in \mathbb{R}^m$, and $y_\tau(t) \in \mathbb{R}^p$ are the state, the control decision variable, and the controlled output vectors at node τ, respectively. The total number of nodes η depends on $\mathcal{I}$ and the prediction horizon length N, with N having the larger effect (see Fig. 1).

$$\eta = \begin{cases} N+1 & \text{if } \mathcal{I} = 1 \\ \dfrac{\mathcal{I}^{N+1}-1}{\mathcal{I}-1} & \text{if } \mathcal{I} > 1 \end{cases}$$

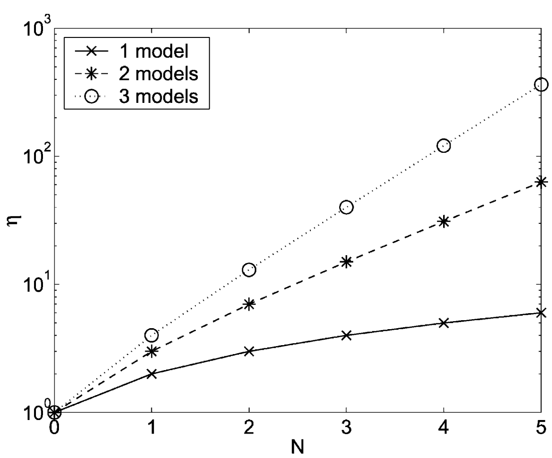

Fig. 1. η as a function of N and $\mathcal{I}$.

Each branch of the tree trajectory is formed by one time-varying sequence of A_i, B_i in Π. Fig. 2 illustrates all the branches in the $\mathcal{I} = 3$ and $N = 2$ tree trajectory. The total number of branches $\mathcal{L}$ is dependent on the number of models $\mathcal{I}$ and the prediction horizon length N.

$$\mathcal{L} = \mathcal{I}^N \tag{18}$$

Table 1 lists the branches, node sets T_l and $\bar{T}_l$, and the terminal nodes τ_l for the $\mathcal{I} = 3$ and $N = 2$ tree trajectory shown in Fig. 2.

Table 1
Set definitions for $\mathcal{I} = 3$ and $N = 2$

Branch l	Terminal node τ_l	Set T_l	Set $\bar{T}_l$
1	1,1	{1,1;1}	{1}
2	1,2	{1,2;1}	{1}
3	1,3	{1,3;1}	{1}
4	2,1	{2,1;2}	{2}
5	2,2	{2,2;2}	{2}
6	2,3	{2,3;2}	{2}
7	3,1	{3,1;3}	{3}
8	3,2	{3,2;3}	{3}
9	3,3	{3,3;3}	{3}

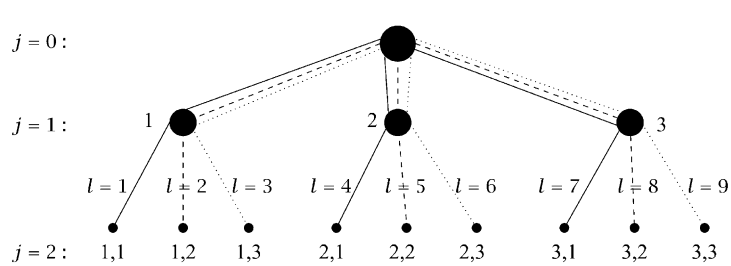

Fig. 2. Graphical illustration of the branches for $\mathcal{I} = 3$ and $N = 2$.

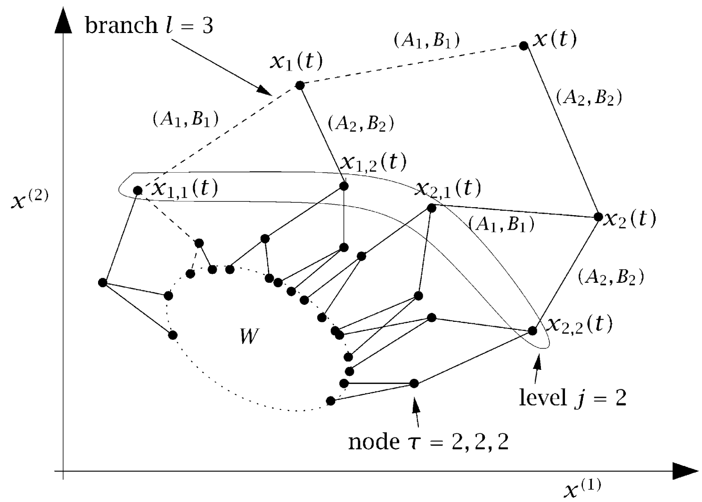

Fig. 3. Tree trajectory with $\mathcal{I} = 2$ and $N = 4$ in the $x \in \mathbb{R}^2$ state space.

Fig. 3 is a graphical illustration of the tree trajectory for $\mathcal{I} = 2$ and $N = 4$ in the $x \in \mathbb{R}^2$ state space. Each node has a control decision variable. The total number of decision variables is equal to the total number of nodes η. For the tree trajectory in Fig. 3, there are

$$\mathcal{L} = 2^4 = 16 \tag{19}$$

branches and

$$\eta = \frac{2^5 - 1}{2 - 1} = 31 \tag{20}$$

decision variables. The forecasted state and output at node τ on level $j = 2$ and branch $l = 3$ as shown in Fig. 3 are:

$$x_{1,1}(t) = A_1 x_1(t) + B_1 u_1(t)$$
$$y_{1,1}(t) = C x_{1,1}(t)$$

5. The open-loop control problem

The objective of the control problem is to find the control actions that, once implemented, cause all branches in the tree trajectory to converge to x_s and u_s. In nominal MPC, the regulator computes the optimal input trajectory by minimizing a performance objective that is the sum of the deviations of the state and input from the steady-state x_s and u_s values. The same concept is extended to the robust MPC regulator. We define the following deviation variables used to compute the performance objective.

$$\bar{u}_\tau(t) = u_\tau(t) - u_s$$
$$\bar{x}_\tau(t) = x_\tau(t) - x_s$$
$$\bar{y}_\tau(t) = y_\tau(t) - Cx_s.$$

The state and output deviation variables are defined as

$$\bar{x}_i(t) = A_i \bar{x}(t) + B_i \bar{u}(t) \quad \text{for} \quad i = 1, \ldots, \mathcal{I}$$
$$\bar{x}_{\tau,i}(t) = A_i \bar{x}_\tau(t) + B_i \bar{u}_\tau(t) \quad \text{for} \quad \tau \in \bar{T}_l \quad \text{and} \quad i = 1, \ldots, \mathcal{I}$$
$$\bar{y}_\tau(t) = C \bar{x}_\tau(t) \tag{21}$$

subject to

$$\bar{u}_\tau(t) \in \mathbb{U}_d \quad \text{for} \quad \tau \in T_l \quad \text{and} \quad l = 1, \ldots, \mathcal{L}$$
$$\bar{x}_\tau(t) \in \mathbb{X}_d \quad \text{for} \quad \tau \in T_l \quad \text{and} \quad l = 1, \ldots, \mathcal{L}$$
$$\bar{y}_\tau(t) \in \mathbb{Y}_d \quad \text{for} \quad \tau \in T_l \quad \text{and} \quad l = 1, \ldots, \mathcal{L}$$
$$\bar{x}_{\tau_l}(t) \in W_d \quad \text{for} \quad l = 1, \ldots, \mathcal{L} \tag{22}$$

in which

$$\mathbb{U}_d = \mathbb{U} - u_s, \quad \mathbb{X}_d = \mathbb{X} - x_s, \quad \mathbb{Y}_d = \mathbb{Y} - y_s, \quad W_d = W - x_s \tag{23}$$

The subsets $\mathbb{U}_d$, $\mathbb{X}_d$, and $\mathbb{Y}_d$ are convex, closed subsets of $\mathbb{R}^m$, $\mathbb{R}^n$, and $\mathbb{R}^p$, respectively, and contain the origin in their interiors. The state admissible set W_d is compact, contains the origin in its interior, and is positively invariant for

$$\bar{x}_{\tau,i}(t) = (A_i + B_i K)\bar{x}_\tau(t) \tag{24}$$

for any $i = 1, \ldots, \mathcal{I}$, $\bar{x}_\tau(t) \in W_d$, and $W_d \subset \mathbb{X}_d$. The control law in W_d is

$$h_W(\bar{x}_\tau(t)) = \bar{u}_\tau(t) = K\bar{x}_\tau(t) \quad \text{for any} \quad \bar{x}_\tau(t) \in W_d \tag{25}$$

which is admissible because it satisfies the control constraint $h_W(\cdot) \subset \mathbb{U}_d$. Let

$$U_l(t) = \left\{ \bar{u}(t), \left\{ \bar{u}_\tau(t) \mid \tau \in \bar{T}_l \right\} \right\} \quad \text{for} \quad l = 1, \ldots, \mathcal{L}$$

denote the sequence of control actions for the lth branch of the tree trajectory. The open-loop control problem in the robust regulator is

$$\min_{U_l(t)} \max_{l=1,\dots,\mathcal{I}} \Phi_l(t;\, U_l(t)) \tag{26}$$

subject to Eqs. (21) and (22) in which $\Phi_l(t;\ U_l(t))$ is the performance objective for the lth branch using the control sequence $U_l(t)$ and is defined as

$$\Phi_l(t;\, U_l(t)) = \phi(\bar{x}(t), \bar{u}(t)) + \sum_{\tau\in\bar{T}_l} \phi(\bar{x}_\tau(t), \bar{u}_\tau(t)) + \phi_N\big(\bar{x}_{\tau_l}(t)\big). \tag{27}$$

To reduce the notation complexity, let

$$\Phi_l(t) = \Phi_l(t;\, U_l(t)) \tag{28}$$

unless we need to stress the functional dependence of $\Phi_l(t)$ on $U_l(t)$. The solution to Eq. (26) is denoted as $U^*(t)$, which is the sequence of control actions that minimizes the deviations of the state and the control from x_s and u_s for the branch with the "worst" or the "largest" performance objective. The control action implemented at time t is $u(t) = \bar{u}^* + u_s$, the first control action in $U^*(t)$ which is in the deviation variable form plus the steady-state control u_s.

5.1. Remarks

R1. *The function*

$$\phi(x, u) = x^T Q x + u^T R u \tag{29}$$

in which $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are positive definite is continuous, positive definite, and proper.

R2. *The function*

$$\phi_N(x) = x^T F x \tag{30}$$

in which $F \in \mathbb{R}^{n\times n}$ is symmetric and positive definite is continuous, positive definite, and proper.

R3. *Because both $\phi(x, u)$ and $\phi_N(x)$ are continuous, positive definite and proper, $\Phi_l(t)$ is continuous, positive definite, and proper.*

5.2. Assumption

A1. The control constraint sets $\mathbb{U}_d$, $\mathbb{X}_d$, and $\mathbb{Y}_d$ are compact and convex, and contain the origin in their interiors.

Theorem 5.1. *Suppose A1 holds. If there exist K and $F = F^T > 0$ such that*

$$F - Q - K^T R K - (A_i + B_i K)^T F (A_i + B_i K) \geqslant 0 \quad \text{for } i = 1, \dots, \mathcal{I} \tag{31}$$

then Eq. (26) generates a control policy that guarantees convergence of $\bar{x}(t)$ and $\bar{u}(t)$ to the origin for all time-varying sequences of (A_i, B_i) in Π.

Proof. See Appendix B. □

5.3. Remarks

R4. *Theorem 5.1 guarantees convergence of $\bar{x}(t)$ and $\bar{u}(t)$ to the origin for time-varying sequences of (A_i, B_i) in Π and not time-varying sequences of $(A(t), B(t))$ in Ω.*

R5. *The terminal region W_d is time-invariant and contains the origin in its interior.*

R6. *If there exist K and F such that Theorem 5.1 is true, then Eq. (27) reduces to a quadratic equation.*

Theorem 5.2. *If there exist K and $F = F^T > 0$ such that Theorem 5.1 holds, then Eq. 26 generates a control policy that guarantees convergence of $x(t)$ and $u(t)$ to x_s and u_s, respectively, for any time-varying sequence of $(A(t), B(t))$ in Ω.*

Proof. See Appendix C. □

Both Kothare et al. [10] and Lu and Arkun [13] proposed control theories with a robust stability constraint similar to Eq. (31). Kothare et al. [10] proposed a finite prediction horizon length of $N = 0$ in which K and F are recalculated at each time step. Lu and Arkun [13] used $N = 1$ with knowledge of the correct model at time t and the linear state feedback policy, which is recalculated after every state measurement. Our regulator formulation has the following features:

- K and F are computed off-line and are not recalculated during the controller operation.
- The control actions implemented are determined by the tree trajectory forecasts and not the control policy $\bar{u}(t) = K\bar{x}(t)$. This control policy is used only to approximate the infinite-horizon cost, and it is not used as a control policy.

6. Existence of K and F

The state feedback gain K in the control law $\bar{u}_\tau(t) = K\bar{x}_\tau(t)$ for all $\bar{x}_\tau(t) \in W_d$ that guarantees decreasing robust performance objective can be transformed to

$$K = YP^{-1} \quad \text{and} \quad F = P^{-1} \tag{32}$$

in which $P = P^T > 0$ and Y are obtained from the solution (if it exists) of the following LMI-based optimization problem.

$$\min_{\gamma, Y, P} \gamma \tag{33}$$

subject to

$$\gamma \geqslant 0 \tag{34}$$

$$P > 0 \tag{35}$$

$$\begin{bmatrix} P & (A_iP + B_iY)^T & (Q^{1/2}P)^T & (R^{1/2}Y)^T \\ (A_iP + B_iY) & P & 0 & 0 \\ Q^{1/2}P & 0 & I & 0 \\ R^{1/2}Y & 0 & 0 & I \end{bmatrix} \geqslant \gamma I \, \forall i = 1, \ldots, \mathcal{I} \tag{36}$$

Eq. (36) is an LMI and is equal to the Schur complement of Eq. (31) after the variable transformation [Eq. (32)].

Effective algorithms are available for finding the solutions of the LMI-based optimization problems. These algorithms converge to the global optimum with non-heuristic stopping criteria. Boyd and El Ghaoui [39], Alizadeh et al. [40], Nesterov and Nemirovsky [41] and Vandenberghe and Boyd [42] showed that upon termination, the algorithms arrive at a solution that is within some pre-specified numerical tolerance of the global optimum. Numerical experience shows that these algorithms solve LMI problems efficiently.

Lemma 6.1. *If there exist Y and $P = P^T > 0$ such that $K = YP^{-1}$, $F = P^{-1}$, and*

$$F - Q - K^TRK - (A_i + B_iK)^TF(A_i + B_iK) \geqslant 0 \, \forall i = 1, \ldots, \mathcal{I}$$

then

$$F - Q - K^TRK - (A + BK)^TF(A + BK) \geqslant 0$$

for all (A, B) in Ω.

Proof. The Schur complement of $F - Q - K^TRK - (A + BK)^TF(A + BK)$ is a linear convex combination of the Schur complements of $F - Q - K^TRK - (A_i + B_iK)^TF(A_i + B_iK)$. If there exist Y and P such that Eq. (36) holds, then the linear convex combination of the Schur complement in Eq. (36) is positive semi-definite. □

7. Output feedback

Feedback is an essential part of any control theory and is used in predictive control to determine the difference, if any, between the predicted and the measured state. In the absence of plant-model mismatch, there is no discrepancy between the nominal MPC's predicted state trajectory and the measured state trajectory. Feedback is not necessary when there is no model uncertainty, neither parametric nor disturbance uncertainties. However, the RMPC method we are proposing is designed specifically to handle model uncertainties. Feedback is therefore an essential part of our theory.

We propose using the input output ARMAX (Auto Regressive Moving Average Exogenous Input) model to describe the system behavior. The state is formed by storing all past inputs and outputs that affect the output at the next time step. If the ARMAX model is

$$\begin{aligned} y(t+1) &= a_1y(t) + a_2y(t-1) + \cdots + a_{n_y}y(t - n_y + 1) \\ &\quad + b_1u(t) + b_2u(t-1) + \cdots + b_{n_u}u(t - n_u + 1), \end{aligned}$$

the state is defined as

$$x(t) = \left[y(t) \ldots y(t - n_y + 1)u(t-1) \ldots u(t - n_u + 1)\right]^T, \tag{37}$$

and the state-space model is

$$x(t+1) = \begin{bmatrix} a_1 & a_2 & \cdots & a_{n_y} & b_2 & \cdots & b_{n_u} \\ I & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} x(t) + \begin{bmatrix} b_1 \\ 0 \\ \vdots \\ 0 \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} I & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} x(t).$$

In principle, because all $y(t)$ and $u(t)$ are measurable, the state $x(t)$ is measurable. Measurement noise can be removed by filtering the signals $y(t)$ and $u(t)$ before constructing $x(t)$.

When model uncertainty is present, we propose using the integrating disturbance models to reconcile the difference between the model predictions and the measured states. Because the correct model is not known, a different integrating state is added to each (A_i, B_i).

$$\begin{aligned} x_{\tau,i}(t) &= A_ix_\tau(t) + B_iu_\tau(t) + p_{i,s} \\ y_{\tau,i}(t) &= Cx_{\tau,i}(t) \end{aligned} \tag{38}$$

in which $p_{i,s} \in \mathbb{R}^n$ is the steady-state bias for model i, and $p_{i,s}$ remains constant until there is a new measurement.

The model disturbances are updated whenever new measurements are available via

$$p_i(t+1) = p_{i,s} + L_i(x(t+1) - x_i(t)) \tag{39}$$

in which $p_i(t+1) \in \mathbb{R}^n$ is the disturbance that integrates the difference between the estimated state and the predicted state from the previous time step. There is a different disturbance filter gain $L_i \in \mathbb{R}^{n\times n}$ for each model i. The computation for L_i is non-trivial. Section 10 provides a detailed explanation.

8. Target calculation

The set point is the desired value of the controlled output $y(t)$, but the regulator performance objective penalizes deviations of the state $x(t)$ and input $u(t)$ from the steady-state x_s and u_s values, respectively. The target calculation therefore has two main objectives. First, it is necessary to determine if the set point is reachable for the given process constraints. Second, the target calculation determines the best steady-state x_s and u_s that minimize the sum of the difference between the steady-state controlled output y_s and the set point y_t for models $i = 1, \dots, \mathcal{I}$.

The steady-state u_s, x_s, and y_s values are defined by Eq. (7). In terms of the process models, there is an $x_{i,s} \in \mathbb{R}^n$ such that

$$x_{i,s} = A_i x_{i,s} + B_i u_s \tag{40}$$

for every $u_s \in \mathbb{R}^m$. The value of the $x_{i,s}$ depends on the model i (see Fig. 4). The steady-state biases $p_{i,s} \in \mathbb{R}^n$ are defined so that

$$\begin{aligned} x_{i,s} &= A_i x_{i,s} + B_i u_s + p_{i,s} \\ x_s &= x_{i,s} \end{aligned} \tag{41}$$

Fig. 5 shows that there are multiple steady-state input values that satisfy the steady-state equality constraint in Eq. (41) because of the extra degree of freedom provided by the steady-state bias $p_{i,s}$. Even though

$$u_s \neq u'_s, \quad p_{1,s} \neq p'_{1,s}, \quad p_{2,s} \neq p'_{2,s} \tag{42}$$

they satisfy the steady-state equality constraint with the same x_s value.

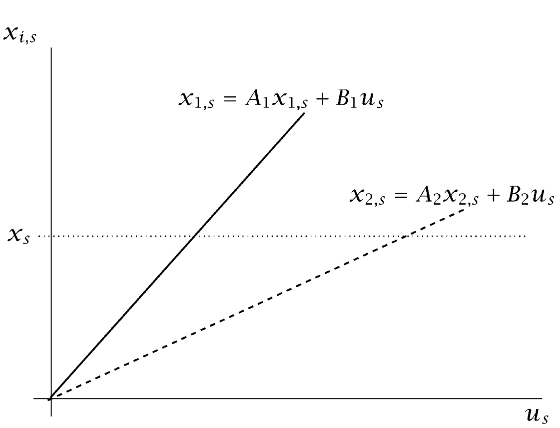

Fig. 4. Graphical illustration of the steady-state equality constraints without the steady-state bias $p_{i,s}$.

In single-model MPC (SMPC), the target calculation is used to determine the steady-state x_s and u_s values. We modify the RMPC target calculation so that it determines not only the x_s and u_s values but the unique $p_{i,s}$ values that satisfy Eq. (41).

Together, the steady-state biases $p_{i,s}$ and the model disturbances $p_i(t)$ act as the integrator needed to achieve offset-free set point tracking. The model disturbance $p_i(t)$ integrates the difference between the plant and the forecasted state using model i. The steady-state bias $p_{i,s}$ are chosen so that the difference between $p_{i,s}$ and $p_i(t)$ is as small as possible while satisfying the steady-state equality constraints in Eq. (41). With integral control, the model forecast is Eq. (38). Once the new state measurement is available, the model disturbances are updated by Eq. (39).

8.1. Assumption

A2. The A_i matrices are non-integrating for all $i = 1, \dots, \mathcal{I}$.

With the new updated model disturbances, the optimal steady-state u_s, x_s, and y_s values are determined by the following optimization problem if assumption A2 holds.

$$\min_{u_s, p_{1,s}, \dots, p_{1,s}} \quad \Phi_s \tag{43}$$

subject to

$$\begin{aligned} x_s = (I - A_i)^{-1}\left(B_i u_s + p_{i,s}\right) \quad &\text{for} \;\; i = 1, \dots, \mathcal{I} \\ u_s \in \mathbb{U}& \\ (I - A_i)^{-1}\left(B_i u_s + p_{i,s}\right) \in \mathbb{X} \quad &\text{for} \;\; i = 1, \dots, \mathcal{I} \\ G_i u_s + C(I - A_i)^{-1} p_{i,s} \in \mathbb{Y} \quad &\text{for} \;\; i = 1, \dots, \mathcal{I} \end{aligned} \tag{44}$$

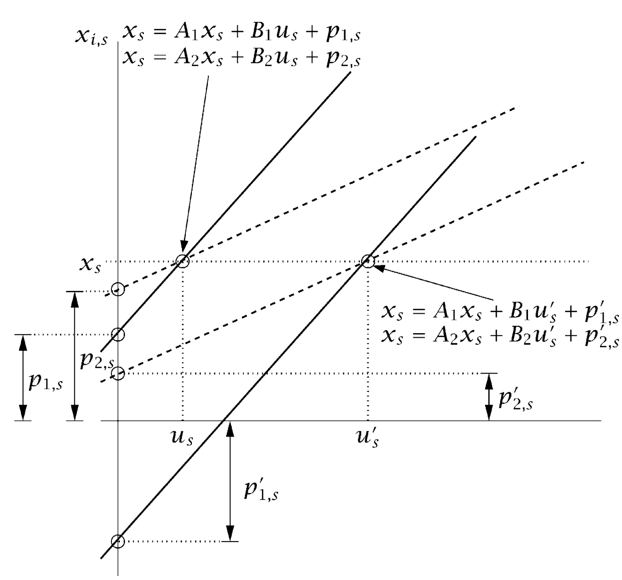

Fig. 5. Graphical illustration of the steady-state bias $p_{i,s}$.

in which the steady-state gain matrix G_i is defined as

$$G_i = C(I - A_i)^{-1} B_i \tag{45}$$

and the steady-state performance objective Φ_s is defined as

$$\Phi_s = \sum_{i=1}^{\mathcal{I}} \left(G_i u_s + C(I - A_i)^{-1} p_{i,s} - y_t\right)^T \times Q_s\left(G_i u_s + C(I - A_i)^{-1} p_{i,s} - y_t\right) + \sum_{i=1}^{\mathcal{I}} \left(p_{i,s} - p_i(t)\right)^T P_i \left(p_{i,s} - p_i(t)\right)$$

The positive definite matrix $Q_s \in \mathbb{R}^{p\times p}$ is the penalty matrix for the difference between the predicted steady-state output $y_s = G_i u_s + C(I - A_i)^{-1} p_{i,s}$ and the set point $y_t \in \mathbb{R}^p$. The symmetric positive definite matrices $P_i \in \mathbb{R}^{n\times n}$ penalize the difference between the steady-state biases $p_{i,s}$ and the model disturbance estimates $p_i(t)$. The minimization of the difference between $p_{i,s}$ and $p_i(t)$ allows Eq. (43) to determine the unique steady-state biases so that Eq. (41) holds for all i.

Eq. (43) minimizes the deviation of the predicted y_s from y_t for all models in Π and the difference between the steady-state bias $p_{i,s}$ and the model disturbance estimates $p_i(t)$. The purpose of minimizing $y_s - y_t$ is to find the y_s value that is as close to the set point as possible. The minimization of $p_{i,s} - p_i(t)$ allows us to find the unique steady-state biases $p_{i,s}$ that are as close as possible to the model disturbance estimates.

Eq. (43) is equivalent to the following optimization problem.

$$\min_{X_s} \; X_s^T H_1 X_s - 2(H_2 y_t + H_3 X(t))^T X_s \tag{46}$$

subject to

$$G X_s = 0$$

$$L_u X_s \in \mathbb{U},\; L_x X_s \in \mathbb{X},\; L_y X_s \in \mathbb{Y}$$

in which

$$X_s = \begin{bmatrix} u_s \\ p_{1,s} \\ \vdots \\ p_{\mathcal{I},s} \end{bmatrix}, \quad X(t) = \begin{bmatrix} x(t) \\ p_1(t) \\ \vdots \\ p_{\mathcal{I}}(t) \end{bmatrix},$$

$$H_2 = \begin{bmatrix} -\sum_{i=1}^{\mathcal{I}} G_i^T Q_s \\ (I - A_1)^{-T} C^T Q_s \\ \vdots \\ (I - A_{\mathcal{I}})^{-T} C^T Q_s \end{bmatrix}, \quad H_3 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & P_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_{\mathcal{I}} \end{bmatrix},$$

$$H_1 = \begin{bmatrix} \sum_{i=1}^{\mathcal{I}} G_i^T Q_s G_i & G_1^T Q_s C(I - A_1)^{-1} & \cdots & G_{\mathcal{I}}^T Q_s C(I - A_{\mathcal{I}})^{-1} \\ (I - A_1)^{-T} C^T Q_s G_1 & (I - A_1)^{-T} C^T Q_s C(I - A_1)^{-1} + P_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (I - A_{\mathcal{I}})^{-T} C^T Q_s G_{\mathcal{I}} & 0 & \cdots & (I - A_{\mathcal{I}})^{-T} C^T Q_s C(I - A_{\mathcal{I}})^{-1} + P_{\mathcal{I}} \end{bmatrix},$$

$$G = \begin{bmatrix} (I - A_1)^{-1} B_1 - (I - A_2)^{-1} B_2 & (I - A_1)^{-1} & -(I - A_2)^{-1} & 0 & \cdots & 0 \\ (I - A_1)^{-1} B_1 - (I - A_3)^{-1} B_3 & (I - A_1)^{-1} B_1 & 0 & (I - A_3)^{-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (I - A_1)^{-1} B_1 - (I - A_{\mathcal{I}})^{-1} B_{\mathcal{I}} & (I - A_1)^{-1} & 0 & 0 & \cdots & (I - A_{\mathcal{I}})^{-1} \end{bmatrix},$$

$$L_u = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix},$$

$$L_x = \begin{bmatrix} (I - A_1)^{-1} B_1 & (I - A_1)^{-1} & 0 & \cdots & 0 \\ (I - A_2)^{-1} B_2 & 0 & (I - A_2)^{-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (I - A_{\mathcal{I}})^{-1} B_{\mathcal{I}} & 0 & 0 & \cdots & (I - A_{\mathcal{I}})^{-1} \end{bmatrix},$$

$$L_y = \begin{bmatrix} G_1 & C(I - A_1)^{-1} & 0 & \cdots & 0 \\ G_2 & 0 & C(I - A_2)^{-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{\mathcal{I}} & 0 & 0 & \cdots & C(I - A_{\mathcal{I}})^{-1} \end{bmatrix}$$

9. Closed-loop stability

Fig. 6 is the closed-loop block diagram of the robust MPC method we are proposing, which includes all the elements that affect closed-loop stability. The target calculation computes the steady-state x_s and u_s values before the regulator computes the optimal manipulated input trajectory. If there is no model uncertainty or unmodeled disturbances, the steady-state values do not change over time, and the target calculation does not affect the regulator stability. However, when model uncertainties are present, the model disturbances $p_i(t)$ do not remain constant. They change whenever the estimated state is not equal to the predicted state. When $p_i(t)$ changes, the target calculation needs to be recomputed, which changes the u_s, x_s, y_s, and $p_{i,s}$ values. Because robust MPC considers model uncertainty explicitly in the controller design procedure, the target calculation directly affects the closed-loop stability.

For an unconstrained system, Eq. (46) is equal to

$$\min_{X_s} \Phi_s = X_s^T H_1 X_s - 2(H_2 y_t + H_3 X(t))^T X_s \tag{47}$$

subject to

$$GX_s = 0. \tag{48}$$

The analytical solution for X_s is

$$\begin{aligned} X_s &= H_1^{-1}(H_2 y_t + H_3 X(t)) \\ &\quad - H_1^{-1} G^T \left(G H_1^{-1} G^T\right)^{-1} G H_1^{-1} (H_2 y_t + H_3 X(t)). \end{aligned} \tag{49}$$

Since the system is unconstrained, the control law

$$u(t) = K(x(t) - x_s) + u_s \tag{50}$$

is a feasible control action. Once $u(t)$ has been implemented, the plant is

$$x(t+1) = A(t)x(t) + B(t)u(t) \tag{51}$$

for any $(A(t), B(t))$ in Ω. The disturbance updates are

$$p_i(t+1) = p_{i,s} + L_i(x(t+1) - x_i(t)) \tag{52}$$

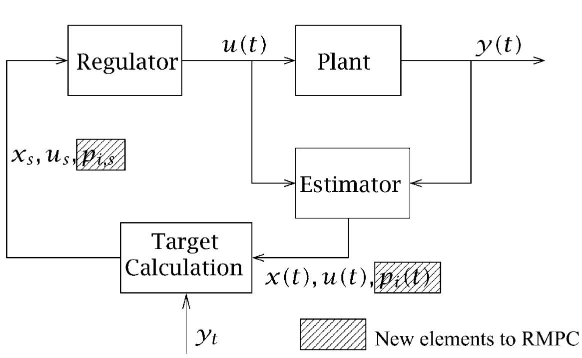

Fig. 6. Block diagram for the RMPC method.

In terms of the augmented state $X(t+1)$, the plant and disturbance updates are equal to

$$\begin{aligned} X(t+1) &= \left(\bar{A} + \tilde{L}\left(\bar{M} + \bar{B}\tilde{K}\right)\right) X(t) \\ &\quad + \left(\bar{N} + \tilde{L}\left(\bar{E} + \bar{B}\tilde{K}\bar{J}\right)\right) X_s \end{aligned} \tag{53}$$

in which

$$\tilde{L} = \begin{bmatrix} I & 0 & \cdots & 0 \\ 0 & L_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_{\mathcal{I}} \end{bmatrix} \quad \text{and} \quad \tilde{K} = \begin{bmatrix} K & 0 & \cdots & 0 \end{bmatrix} \tag{54}$$

are the augmented disturbance filter gain and the state feedback gain, respectively. The augmented model matrices are

$$\bar{A} = \begin{bmatrix} A & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B \\ B - B_1 \\ \vdots \\ B - B_{\mathcal{I}} \end{bmatrix},$$

$$\bar{M} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ A - A_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A - A_{\mathcal{I}} & 0 & \cdots & 0 \end{bmatrix}, \bar{N} = \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \end{bmatrix},$$

$$\bar{E} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ B - B_1 & -I & 0 & \cdots & 0 \\ B - B_2 & 0 & -I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B - B_{\mathcal{I}} & 0 & 0 & \cdots & -I \end{bmatrix},$$

$$\bar{J} = \begin{bmatrix} -(I - A_1)^{-1} B_1 & -(I - A_1)^{-1} & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Eq. (53) expresses $X(t+1)$ as a function of $X(t)$ and X_s. We substitute Eq. (49) into Eq. (53) to express $X(t+1)$ as a function of $X(t)$ and y_t.

$$X(t+1) = \left(\tilde{A} + \tilde{L}\left(\tilde{E} + \tilde{B}\tilde{K}\tilde{J}\right)\right) X(t) + \tilde{G} y_t \tag{55}$$

in which

$$\tilde{A} = \bar{A} + \bar{N}\left\{ H_1^{-1} H_3 - H_1^{-1} G^T \left(G H_1^{-1} G^T\right)^{-1} G H_1^{-1} H_3 \right\}$$

$$\tilde{B} = \bar{B}$$

$$\tilde{E} = \bar{M} + \bar{E}\left\{H_1^{-1}H_3 - H_1^{-1}G^T\left(GH_1^{-1}G^T\right)^{-1}GH_1^{-1}H_3\right\}$$

$$\tilde{J} = I + \bar{J}\left\{H_1^{-1}H_3 - H_1^{-1}G^T\left(GH_1^{-1}G^T\right)^{-1}GH_1^{-1}H_3\right\}$$

$$\tilde{G} = \left(\bar{N} + \tilde{L}\left(\bar{E} + \bar{B}\tilde{K}\bar{J}\right)\right)\left\{H_1^{-1}H_2 - H_1^{-1}G^T\left(GH_1^{-1}G^T\right)^{-1}GH_1^{-1}H_2\right\}$$

For the vertices of Ω, $A = A_i$ and $B = B_i$. The corresponding augmented matrices are $\tilde{A}_i$, $\tilde{B}_i$, $\tilde{E}_i$ and $\tilde{G}_i$.

Lemma 9.1. *If there exist $\tilde{L}$ such that the eigenvalues of $\tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J})$ are within the unit circle, then the eigenvalues of $\tilde{A} + \tilde{L}(\tilde{E} + \tilde{B}\tilde{K}\tilde{J})$ are within the unit circle for all (A, B) in Ω.*

Proof. See [38]. □

Theorem 9.2. *If there exist $\tilde{L}$ and $\tilde{K}$ such that the eigenvalues of $\tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J})$ are within the unit circle for all (A, B) in Ω, then there exists a control policy that robustly stabilizes the time-varying uncertain system in Eq. (51). If the plant converges to the time-invariant model (A_∞, B_∞), then the control policy achieves the steady state (x_s, u_s) that minimizes or eliminates the difference between the predicted steady-state output y_s and the output set point y_t.*

Proof. See [38]. □

10. Existence of $\tilde{L}$

Section 6 shows the robustly stabilizing K and F can be found by solving an LMI-based optimization problem. Numerical algorithms are available that find the optimal solution to Eq. (33). The optimization in Theorem 9.2 cannot be transformed into an LMI-based optimization problem because of the specific structures of $\tilde{L}$ and $\tilde{K}$. The following numerical algorithm determines if closed-loop stability can be achieve for all (A, B) in Ω. Closed-loop stability and offset-free non-zero set point tracking can be guaranteed if there exists a feasible solution to Eq. (56). Closed-loop stability does not depend on finding the optimal solution.

Algorithm 10.1

1. Find the robustly stabilizing K and F by solving Eq. (33) and forming $\tilde{K}$ as in Eq. (54)
2. Solve
$$\min_{L_i} \sum_{i=1}^{\mathcal{I}} \left\| \tilde{A}_i + \tilde{L}\left(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J}\right)\right\|^2 \quad (56)$$
subject to the eigenvalues of $\tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J})$ are within the unit circle for all $i = 1, \ldots, \mathcal{I}$.

The optimization in Step 2 of Algorithm 10.1 is a difficult non-convex problem. Nonlinear solvers may fail to find a feasible $\tilde{L}$ even though a feasible solution exists. In order to devise a solution strategy, we note that $L_i = 0$ for all $i = 1, \ldots, \mathcal{I}$ is a feasible solution if the integrators are pre-multiplied by a factor less than one.

$$p_i(t+1) = d_i p_{i,s} + L_i(x(t+1) - x_i(t)) \quad (57)$$

in which $d_i \in \mathbb{R}$ and $0 \leqslant d_i < 1$ for all i. The steady-state equality constraints for the target calculation are

$$x_s = A_i x_s + B_i u_s + d_i p_{i,s} \quad \text{for} \quad i = 1, \ldots, \mathcal{I},$$

and the state forecast is

$$x_{\tau,i}(t) = A_i x_\tau(t) + B_i u_\tau(t) + d_i p_{i,s} \quad (58)$$

With the above modifications, if $L_i = 0$ for all i, the eigenvalues of $\tilde{A}_i + \tilde{L}\left(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J}\right)$ that correspond to the integrating modes are equal to d_i. If d_i is less than one, $L_i = 0$ is a feasible solution. The nonlinear optimization solvers are more likely to find a solution to Eq. (56) if the initial guess is feasible. We propose, therefore, the following algorithm that involves an iterative procedure beginning with $0 \leqslant d_i < 1$. The $\tilde{L}$ that satisfies Theorem 9.2 is found when $d_i = 1$ for all $i = 1, \ldots, \mathcal{I}$.

Algorithm 10.2

1. Find the robustly stabilizing K and F by solving Eq. (33) and forming $\tilde{K}$ as in Eq. (54).
2. Let $d_i < 1$ and $L_i = 0$ for all $i = 1, \ldots, \mathcal{I}$. Let
$$\tilde{L}_0 = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ 0 & L_1 & 0 & \cdots & 0 \\ 0 & 0 & L_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & L_{\mathcal{I}} \end{bmatrix}$$
be the initial guess.
3. Solve
$$\min_{L_i} \sum_{i=1}^{\mathcal{I}} \left\| \tilde{A}_i + \tilde{L}\left(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J}\right)\right\|^2 + \rho\left\|\tilde{L} - \tilde{L}_0\right\|^2 \quad (59)$$
subject to the eigenvalues of $\tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J})$ are within the unit circle. The weighting factor $\rho \in \mathbb{R}$ is a user defined variable that decides how close the solution $\tilde{L}$ is to the initial guess $\tilde{L}_0$.
4. If the nonlinear optimizer returns an optimal solution, increase d_i and let $\tilde{L}_0$ be equal to the solution of Eq. (59). Go to Step 5. If the nonlinear optimizer cannot find a solution, decrease d_i and repeat Step 3.
5. Solve Eq. (59). If the solution is optimal and $d_i = 1$, STOP. If $d_i < 1$, go to Step 3 and repeat. If no solution exists or the solution is not optimal, decrease d_i and repeat Step 3.

Algorithm 10.2 successfully found solutions for systems with various numbers of states and models. The largest problem we solved had 12 states and three models. As the numbers of states and models increase, the time for Algorithm 10.2 to find a solution increases, with the number of states having a larger effect. The solution to Algorithm 10.2 is determined off-line. The computation time does not affect the RMPC method's on-line implementation feasibility.

11. Summary of the RMPC method

In this section, we provide a summary of the steps needed in the new RMPC method.

1. Find the robustly stabilizing K and F by solving Eq. (33), an LMI-based optimization.
2. Use Algorithm 10.2 to find the augmented disturbance filter gain $\tilde{L}$,which requires satisfaction of a set of inequalities.
3. Find the steady-state u_s, x_s, and $p_{i,s}$ values by solving Eq. (43), a quadratic program.
4. Find the sequence of control actions $U^*(t)$ by solving Eq. (26), a *min–max* optimization.
5. Inject the control action $u(t) = \bar{u}^*(t) + u_s$ from $U^*(t)$ into the plant.
6. Measure $y(t+1)$ and construct $x(t+1)$ as shown in Eq. (37).
7. Use Eq. (39) to update the model disturbances $p_i(t)$.
8. Go to Step 3 and repeat.

Steps 1 and 2 are performed off-line to establish closed-loop stability for offset-free control of the time-varying system described by Eqs. (3) and (4). Steps 3–7 are computed on-line at each time step.

12. Conclusion

In this paper, we propose a new RMPC method that includes a closed-loop stability condition that determines if offset-free control is possible given the model uncertainty description. The stability condition considers the effects of the robust regulator, target calculation, and state estimation on the closed-loop stability. Offset-free control in the presence of model uncertainty is accomplished by adding steady-state biases to the steady-state equality constraints so that for a single u_s value, all models in the model uncertainty description have the same steady-state x_s value. The target calculation determines the x_s and u_s values necessary to achieve offset-free control. The tree trajectory in the robust regulator is designed to forecast the state for a system with time-varying uncertainty. Offset-free control is achieved when all branches in the tree trajectory converge to the x_s and u_s values. In a companion paper [43] we apply this approach to a number of test examples and compare its performance to single-model MPC. We will also discuss the computational burden of finding the optimal tree trajectory control actions by solving the *min–max* optimization.

Of course we may consider many alternatives to the *min–max* formulation for treating model uncertainty. If an identification procedure produces a nominal model and an associated probability distribution for the model parameters, then one might choose to optimize the expectation of the plant behavior rather than the worst case behavior. We hope to see in the future a critical comparison of several alternative formulations of the robust MPC. Some of the critical issues for this comparison will likely be: ease of uncertainty identification, online computational burden, and achieved robustness properties of the controller.

Acknowledgements

The authors would like to thank David Mayne for providing much helpful feedback on this problem. The financial support of the National Science Foundation, through grants CTS-9708497 and CTS-0105360, and the industrial members of the Texas-Wisconsin Modeling and Control Consortium is gratefully acknowledged.

Appendix A. Additional definitions needed for theorem 5.1 and 5.2 proofs

D19. If $a \in \mathbb{I}$ and $b \in \mathbb{I}$, then the operation $\Re(a, b)$ finds the remainder of a divided by b.

D20. Let $\bar{\bar{T}}_l$ be a subset of $\bar{T}_l$ and be defined as

$$\bar{\bar{T}}_l = \bar{T}_l \backslash \mathbf{P}(\tau_l) \quad \text{for} \quad l = 1, \ldots, \mathcal{L} \tag{60}$$

The set $\bar{\bar{T}}_l$ contains all the nodes in T_l except for the terminal node τ_l and the second to last node $\mathbf{P}(\tau_l)$.

D21. The set Γ contains the nodes on all branches $l = 1, \ldots, \mathcal{L}$ and is defined as

$$\Gamma = \bigcup_{l=1}^{\mathcal{L}} T_l \tag{61}$$

D22. The subset Γ_N contains all the terminal nodes τ_l, the nodes on level N, and is defined as

$$\Gamma_N = \{\tau_l \mid l = 1, \ldots, \mathcal{L}\} \tag{62}$$

D23. The subset Γ_{N-1} contains the nodes on level $N-1$, the second to last level, and is defined as

$$\Gamma_{N-1} = \left\{ \mathbf{P}(\tau_l) \mid l = 1, \ldots, \mathcal{L} \right\} \tag{63}$$

D24. The subset $\bar{\Gamma}$ contains all nodes on the tree trajectory except for the terminal nodes and is defined as

$$\bar{\Gamma} = \Gamma / \Gamma_N \tag{64}$$

D25. The subset $\bar{\bar{\Gamma}}$ contains all nodes on the tree trajectory except for the nodes on levels N and $N-1$. The subset is defined as

$$\bar{\bar{\Gamma}} = \bar{\Gamma} / \Gamma_{N-1} \tag{65}$$

D26. If $i \in \mathbb{I}$ is the model index, then S_i denotes the set of branch indices such that

$$S_i = \left\{ l \mid \mathbf{P}^{N-1}(\tau_l) = i, l = 1, \ldots, \mathcal{L} \right\} \tag{66}$$

D27. If $\sigma \in \mathbb{I}$, $l \in \mathbb{I}$, $i = 1, \ldots, \mathcal{I}$, and $c = \Re(l, \mathcal{I}^{N-1})$ for $l \in S_i$, then the set of branch indices $\bar{S}_l$ is defined as

$$\bar{S}_l = \left\{ \sigma \middle| \begin{array}{ll} \mathcal{I}^N - 1 \leqslant \sigma \leqslant \mathcal{I}^N & \text{if} \quad c = 0 \\ \mathcal{I}(c-1) + 1 \leqslant \sigma \leqslant \mathcal{I}c & \text{if} \quad c \neq 0 \end{array} \right\} \tag{67}$$

D28. If $l \in \mathbb{I}$ and $i = 1, \ldots, \mathcal{I}$, the set S is defined as

$$S = \bigcup_{l \in S_i} \bar{S}_l = 1, \ldots, \mathcal{L} \tag{68}$$

Appendix B. Proof for theorem 5.1

At time t and state $\bar{x}^*(t)$, the solution to Eq. (26) is $\Phi^*(t)$. The optimal control sequence is $U^*(t)$. The optimal control action $\bar{u}(t) = \bar{u}^*(t)$ is implemented, and the state at time $t+1$ is

$$\begin{aligned} \bar{x}(t+1) \quad &= A_{i^*}\bar{x}^*(t) + B_{i^*}\bar{u}^*(t) \\ &= \bar{x}_{i^*}(t) \end{aligned} \tag{69}$$

when model i^* is the plant at time t. Consider the following feasible but non-optimal control actions at time $t+1$:

$$\bar{u}(t+1) = \bar{u}_{i^*}(t) \tag{70}$$

$$\bar{u}_\tau(t+1) = \bar{u}_{i^*,\tau}(t) \quad \text{for} \quad \tau \in \bar{\bar{\Gamma}} \tag{71}$$

$$\bar{u}_\tau(t+1) = K\bar{x}_{i^*,\tau}(t) \quad \text{for} \quad \tau \in \Gamma_{N-1} \tag{72}$$

The control sequence at time $t+1$ is

$$U_\sigma(t+1) = \left\{ \bar{u}(t+1), \left\{ \bar{u}_\tau(t+1) \mid \tau \in \bar{T}_\sigma \right\} \right\} \tag{73}$$

for $\sigma = 1, \ldots, \mathcal{L}$

Eq. (17) shows the state forecasts for the tree trajectory. The state at node $\tau = i$ and time $t+1$ is

$$\begin{aligned} \bar{x}_i(t+1) \quad &= A_i\bar{x}(t+1) + B_i\bar{u}(t+1) \\ &= A_i\bar{x}_{i^*}(t) + B_i\bar{u}_{i^*}(t) \\ &= \bar{x}_{i^*,i}(t) \end{aligned} \tag{74}$$

The state at node τ and time $t+1$ is

$$\bar{x}_\tau(t+1) = \bar{x}_{i^*,\tau}(t) \quad \text{for} \quad \tau \in \bar{\Gamma} \tag{75}$$

The performance objective for branch l at time t with $\bar{u}(t) = \bar{u}^*(t)$ is

$$\Phi_l^*(t) = \phi(\bar{x}^*(t), \bar{u}^*(t)) + \sum_{\tau \in \bar{T}_l} \phi(\bar{x}_\tau(t), \bar{u}_\tau(t)) + \phi_N\left(\bar{x}_{\tau_l}(t)\right) \quad \text{for} \quad l = 1, \ldots, \mathcal{L} \tag{76}$$

For the branches in S_{i^*}, Eq. (76) is equivalent to

$$\begin{aligned} \Phi_l^*(t) = {} & \phi(\bar{x}^*(t), \bar{u}^*(t)) + \phi(\bar{x}_{i^*}(t), \bar{u}_{i^*}(t)) \\ & + \sum_{\tau \in \bar{\bar{T}}_\sigma} \phi\left(\bar{x}_{i^*,\tau}(t), \bar{u}_{i^*,\tau}(t)\right) + \phi_N\left(\bar{x}_{i^*,\mathbf{P}(\tau_\sigma)}(t)\right) \end{aligned} \tag{77}$$

for $\sigma \in \bar{S}_l$ and $l \in S_{i^*}$

After substituting Eqs. (70), (71), and (75) into Eq. (77),

$$\begin{aligned} \Phi_l^*(t) = {} & \phi(\bar{x}^*(t), \bar{u}^*(t)) + \phi(\bar{x}(t+1), \bar{u}(t+1)) \\ & + \sum_{\tau \in \bar{\bar{T}}_\sigma} \phi(\bar{x}_\tau(t+1), \bar{u}_\tau(t+1)) + \phi_N\left(\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1)\right) \end{aligned}$$

for $\sigma \in \bar{S}_l$ and $l \in S_{i^*}$

(78)

The non-optimal performance objective for branch σ at time $t+1$ is

$$\begin{aligned} \Phi_\sigma(t+1) = {} & \phi(\bar{x}(t+1), \bar{u}(t+1)) \\ & + \sum_{\tau \in \bar{\bar{T}}_\sigma} \phi(\bar{x}_\tau(t+1), \bar{u}_\tau(t+1)) \\ & + \phi\left(\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1), \bar{u}_{\mathbf{P}(\tau_\sigma)}(t+1)\right) + \phi_N\left(\bar{x}_{\tau_\sigma}(t+1)\right) \end{aligned}$$

for $\sigma = 1, \ldots, \mathcal{I}$

(79)

For node $\tau \in \Gamma_{N-1}$ at time $t+1$, Eq. (72) is a feasible control action. The state at $\tau_\sigma \in \Gamma_N$ is

$$\begin{aligned} \bar{x}_{\tau_\sigma}(t+1) &= \bar{x}_{\mathbf{P}(\tau_\sigma),i}(t+1) \\ &= A_i\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1) + B_i\bar{u}_{\mathbf{P}(\tau_\sigma)}(t+1) \\ &= A_i\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1) + B_iK\bar{x}_{i^*,\mathbf{P}(\tau_\sigma)}(t) \\ &= (A_i + B_iK)\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1) \quad \text{for} \quad i = 1, \ldots, \mathcal{I} \end{aligned} \tag{80}$$

After substituting Eq. (80) into Eq. (79),

$$\begin{aligned}\Phi_\sigma(t+1) = {} & \phi(\bar{x}(t+1), \bar{u}(t+1)) \\ & + \sum_{\tau\in\bar{\bar{T}}_\sigma} \phi(\bar{x}_\tau(t+1), \bar{u}_\tau(t+1)) \\ & + \phi\big(\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1), K\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1)\big) \\ & + \phi_N\big((A_i + B_iK)\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1)\big)\end{aligned} \tag{81}$$

The difference between Eqs. (78) and (81) is

$$\begin{aligned}\Phi_l^*(t) - \Phi_\sigma(t+1) = {} & \phi(\bar{x}^*(t), \bar{u}^*(t)) \\ & + x_{\mathbf{P}(\tau_\sigma)}(t+1)^T\big(F - Q - K^TRK \\ & -(A_i + B_iK)^TF(A_i + B_iK)\big)x_{\mathbf{P}(\tau_\sigma)}(t+1)\end{aligned}$$

for $\sigma \in \bar{S}_l$ and $l \in S_{i^*}$

If there exist F and K such that

$$\begin{aligned}& F - Q - K^TRK - (A_i + B_iK)^TF(A_i + B_iK) \geqslant 0 \\ & \text{for } i = 1, \ldots, \mathcal{I}\end{aligned} \tag{82}$$

then

$$\Phi_l^*(t) - \Phi_\sigma(t+1) \geqslant 0 \quad \text{for } \sigma \in \bar{S}_l \text{ and } l \in S_{i^*} \tag{83}$$

The optimal performance objective at time t is the maximum of the $\Phi_l^*(t)$, so

$$\Phi_\sigma(t+1) \leqslant \Phi^*(t) \quad \text{for } \sigma \in S \tag{84}$$

Eq. (73) is a feasible but non-optimal control sequence. Recall Eq. (28). $\Phi_\sigma(t+1; U_\sigma(t+1))$ denotes the performance objective at time $t+1$ using the control sequence $U_\sigma(t+1)$ in Eq. (73). Let $\Phi_\sigma(t+1; U'_\sigma(t+1))$ denote the performance objective at time $t+1$ using some other control sequence $U'_\sigma(t+1)$. We define the function

$$\Psi\big(U'_\sigma(t+1)\big) = \max_{\sigma\in S} \Phi_\sigma\big(t+1; U'_\sigma(t+1)\big) \tag{85}$$

The value of $\Psi(U_\sigma(t+1))$ is

$$\Psi(U_\sigma(t+1)) = \max_{\sigma\in S} \Phi_\sigma(t+1; U_\sigma(t+1)) \tag{86}$$

The *min–max* optimization

$$\min_{U'_\sigma(t+1)} \max_{\sigma\in S} \Phi_\sigma\big(t+1; U'_\sigma(t+1)\big) = \min_{U'_\sigma(t+1)} \Psi\big(U'_\sigma(t+1)\big) \tag{87}$$

has a solution $U^*(t+1)$ which implies

$$\Psi(U^*(t+1)) \leqslant \Psi(U_\sigma(t+1)) \tag{88}$$

Therefore,

$$\min_{U'_\sigma(t+1)} \max_{\sigma\in S} \Phi_\sigma\big(t+1; U'_\sigma(t+1)\big) \leqslant \max_{\sigma\in S} \Phi_\sigma(t+1; U_\sigma(t+1)) \tag{89}$$

$$\Rightarrow \Phi^*(t+1; U^*(t+1)) \leqslant \Phi^*(t; U^*(t)) \tag{90}$$

in which $\Phi^*(t+1; U^*(t+1))$ is the optimal performance objective at time $t+1$. Eq. (90) indicates $\Phi^*(t; U^*(t))$ is a non-increasing sequence bounded below by zero and therefore converges.

Appendix C. Proof for theorem 5.2

At time $t+1$, the state is

$$\begin{aligned}\bar{x}(t+1) & = \sum_{i=1}^{\mathcal{I}} \mu_i(t)A_i\bar{x}^*(t) + \sum_{i=1}^{\mathcal{I}} \mu_i(t)B_i\bar{u}^*(t) \\ & = \sum_{i=1}^{\mathcal{I}} \mu_i(t)(A_i\bar{x}(t) + B_i\bar{u}^*(t)) \\ & = \sum_{i=1}^{\mathcal{I}} \mu_i(t)\bar{x}_i(t)\end{aligned} \tag{91}$$

if the convex combination $(\sum_{i=1}^{\mathcal{I}} \mu_i(t)A_i,\ \sum_{i=1}^{\mathcal{I}} \mu_i(t)B_i)$ is the plant. The control action $\bar{u}(t) = \bar{u}^*(t)$ implemented is from the optimal control sequence $U^*(t)$. $\bar{x}(t+1)$ is a linear convex combination of the states at node $\tau = i$ and time t. Consider the following feasible but non-optimal candidate control actions:

$$\bar{u}(t+1) = \sum_{i=1}^{\mathcal{I}} \mu_i(t)\bar{u}_i(t) \tag{92}$$

$$\bar{u}_\tau(t+1) = \sum_{i=1}^{\mathcal{I}} \mu_i(t)\bar{u}_{i,\tau}(t) \quad \text{for} \quad \tau \in \bar{\bar{\Gamma}} \tag{93}$$

$$\bar{u}_\tau(t+1) = \sum_{i=1}^{\mathcal{I}} \mu_i(t)K\bar{x}_{i,\tau}(t) \quad \text{for} \quad \tau \in \Gamma_{N-1} \tag{94}$$

A control sequence at time $t+1$ is

$$\begin{aligned}& U_\sigma(t+1) = \Big\{\bar{u}(t+1), \Big\{\bar{u}_\tau(t+1) \mid \tau \in \bar{T}_\sigma\Big\}\Big\} \\ & \text{for } \sigma = 1, \ldots, \mathcal{I}\end{aligned} \tag{95}$$

We claim the state at node τ and time $t+1$ is

$$\bar{x}_\tau(t+1) = \sum_{i=1}^{\mathcal{I}} \mu_i(t)\bar{x}_{i,\tau}(t) \tag{96}$$

Proof for the above claim is in [38]. The performance objective for branch σ is

$$\begin{aligned}\Phi_\sigma(t+1) &= \phi(\bar{x}(t+1), \bar{u}(t+1)) \\ &\quad + \sum_{\tau\in\bar{\bar{T}}_\sigma}\phi(\bar{x}_\tau(t+1), \bar{u}_\tau(t+1)) \\ &\quad + \phi\big(\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1), \bar{u}_{\mathbf{P}(\tau_\sigma)}(t+1)\big) \\ &\quad + \phi_N\big(\bar{x}_{\tau_\sigma}(t+1)\big)\end{aligned} \tag{97}$$

in which

$$\bar{u}_{\mathbf{P}(\tau_\sigma)}(t+1) = K\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1)$$

$$\bar{x}_{\tau_\sigma}(t+1) = (A_k + B_kK)\bar{x}_{\mathbf{P}(\tau_\sigma)}(t+1) \quad \text{for } k = 1,\ldots,\mathcal{I}$$

After substituting Eqs. (92)–(94) and (96) into Eq. (97),

$$\begin{aligned}\Phi_\sigma(t+1) &= \phi\left(\sum_{i=1}^{\mathcal{I}}\mu_i(t)\bar{x}_i(t), \sum_{i=1}^{\mathcal{I}}\mu_i(t)\bar{u}_i(t)\right) \\ &\quad + \sum_{\tau\in\bar{\bar{T}}_\sigma}\phi\left(\sum_{i=1}^{\mathcal{I}}\mu_i(t)\bar{x}_{i,\tau}(t), \sum_{i=1}^{\mathcal{I}}\mu_i(t)\bar{u}_{i,\tau}(t)\right) \\ &\quad + \phi\left(\sum_{i=1}^{\mathcal{I}}\mu_i(t)\bar{x}_{i,\mathbf{P}(\tau_\sigma)}(t), \sum_{i=1}^{\mathcal{I}}\mu_i(t)K\bar{x}_{i,\mathbf{P}(\tau_\sigma)}(t)\right) \\ &\quad + \phi_N\left((A_k + B_kK)\sum_{i=1}^{\mathcal{I}}\mu_i(t)\bar{x}_{i,\mathbf{P}(\tau_\sigma)}(t)\right)\end{aligned} \tag{98}$$

By the convexity of the function $\phi(x,u)$ and $0 \leqslant \mu_i(t) \leqslant 1$,

$$\begin{aligned}&\phi\left(\sum_{i=1}^{\mathcal{I}}\mu_i(t)\bar{x}_{i,\tau}(t), \sum_{i=1}^{\mathcal{I}}\mu_i(t)\bar{u}_{i,\tau}(t)\right) \\ &\quad\leqslant \sum_{i=1}^{\mathcal{I}}\phi\big(\mu_i(t)\bar{x}_{i,\tau}(t)\mu_i(t)\bar{u}_{i,\tau}(t)\big) \\ &\quad= \sum_{i=1}^{\mathcal{I}}\mu_i(t)^2\phi\big(\bar{x}_{i,\tau}(t), \bar{u}_{i,\tau}(t)\big) \\ &\quad\leqslant \sum_{i=1}^{\mathcal{I}}\mu_i(t)\phi\big(\bar{x}_{i,\tau}(t), \bar{u}_{i,\tau}(t)\big) \\ &\quad \text{for } \tau \in \bar{T}_\sigma \text{ and } \sigma = 1,\ldots,\mathcal{I}\end{aligned} \tag{99}$$

Let

$$\begin{aligned}\Phi_{\sigma,i}(t+1) &= \phi(\bar{x}_i(t), \bar{u}_i(t)) + \sum_{\tau\in\bar{\bar{T}}_\sigma}\phi\big(\bar{x}_{i,\tau}(t), \bar{u}_{i,\tau}(t)\big) \\ &\quad + \phi\big(\bar{x}_{i,\mathbf{P}(\tau_\sigma)}(t), K\bar{x}_{i,\mathbf{P}(\tau_\sigma)}(t)\big) \\ &\quad + \phi_N\big((A_k + B_kK)\bar{x}_{i,\mathbf{P}(\tau_\sigma)}(t)\big) \\ &\quad \text{for } \sigma \in \bar{S}_l, \quad l \in S_i, \quad i = 1,\ldots,\mathcal{I}\end{aligned} \tag{100}$$

in which $\Phi_{\sigma,i}$ is the performance objective at $t+1$ for branch σ with $\mathbf{P}^{N-1}(\tau_\sigma) = i$ and $i = 1,\ldots,\mathcal{I}$. After substituting Eqs. (99) and (100) into Eq. (98), the performance objective becomes

$$\begin{aligned}&\Phi_\sigma(t+1) \\ &\quad\leqslant \sum_{i=1}^{\mathcal{I}}\mu_i(t)\Bigg[\phi(\bar{x}_i(t), \bar{u}_i(t)) + \sum_{\tau\in\bar{\bar{T}}_\sigma}\phi\big(\bar{x}_{i,\tau}(t), \bar{u}_{i,\tau}(t)\big) \\ &\quad + \phi\big(\bar{x}_{i,\mathbf{P}(\tau_\sigma)}(t), K\bar{x}_{i,\mathbf{P}(\tau_\sigma)}(t)\big) + \phi_N\big((A_k + B_kK)\bar{x}_{i,\mathbf{P}(\tau_\sigma)}(t)\big)\Bigg] \\ &\quad= \sum_{i=1}^{\mathcal{I}}\mu_i(t)\Phi_{\sigma,i}(t+1)\end{aligned} \tag{101}$$

If there exists K and F such that Theorem 5.1 holds, then

$$\Phi_{\sigma,i}(t+1) \leqslant \Phi^*(t) \text{ for } \sigma \in \bar{S}_l, \ l \in \bar{S}_i \text{ and } i = 1,\ldots,\mathcal{I} \tag{102}$$

$$\Rightarrow \sum_{i=1}^{\mathcal{I}}\mu_i(t)\Phi_{\sigma,i}(t+1) \leqslant \sum_{i=1}^{\mathcal{I}}\mu_i(t)\Phi^*(t) \tag{103}$$

By combining Eqs. (101), (100) and (103),

$$\Phi_\sigma(t+1) \leqslant \sum_{i=1}^{\mathcal{I}}\mu_i(t)\Phi_{\sigma,i}(t+1) \leqslant \Phi^*(t) \quad \text{for } \sigma \in S \tag{104}$$

Eq. (95) is a feasible but non-optimal control sequence. Recall Eq. (28). $\Phi_\sigma(t+1; U_\sigma(t+1))$ denotes the performance objective at time $t+1$ using the control sequence.

$U_\sigma(t+1)$ in Eq. (95). Let $\Phi_\sigma(t+1; U'_\sigma(t+1))$ denote the performance objective at time $t+1$ using some other control sequence $U'_\sigma(t+1)$. We define the function

$$\Psi\big(U'_\sigma(t+1)\big) = \max_{\sigma\in S}\Phi_\sigma\big(t+1; U'_\sigma(t+1)\big) \tag{105}$$

The value of $\Psi(U_\sigma(t+1))$ is

$$\Psi(U_\sigma(t+1)) = \max_{\sigma\in S}\Phi_\sigma(t+1; U_\sigma(t+1)) \tag{106}$$

The *min–max* optimization

$$\min_{U'_\sigma(t+1)} \max_{\sigma\in S} \Phi_\sigma\big(t+1;\, U'_\sigma(t+1)\big) = \min_{U'_\sigma(t+1)} \Psi\big(U'_\sigma(t+1)\big) \tag{107}$$

has a solution $U^*(t+1)$ which implies

$$\Psi(U^*(t+1)) \leqslant \Psi(U_\sigma(t+1)) \tag{108}$$

Therefore,

$$\min_{U'_\sigma(t+1)} \max_{\sigma\in S} \Phi_\sigma\big(t+1;\, U'_\sigma(t+1)\big) \leqslant \max_{\sigma\in S} \Phi_\sigma(t+1;\, U_\sigma(t+1)) \tag{109}$$

$$\Rightarrow\ \Phi^*(t+1;\, U^*(t+1)) \leqslant \Phi^*(t;\, U^*(t)) \tag{110}$$

which indicates $\Phi^*(t; U^*(t))$ is a non-increasing sequence bounded below by zero and therefore converges. In other words,

$$\begin{aligned} & \Phi^*(t;\, U^*(t)) \ \to\ 0 \quad \text{as} \quad t \to \infty \\ \Rightarrow\quad & \bar{x}_\tau(t) \ \to\ 0 \quad \text{and} \quad \bar{u}_\tau(t) \ \to\ 0 \\ \Rightarrow\quad & \bar{x}_\tau(t) \ \to\ x_s \quad \text{and} \quad u_\tau(t) \ \to\ u_s. \end{aligned}$$

References

[1] R. Berber, Methods of Model Based Process Control (NATO ASI Series. Series E, Applied Sciences, Vol. 293), Kluwer Academic, Dordrecht, The Netherlands, 1995.
[2] T. Başar, P. Bernhard, H^∞-Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach, Birkauser, Boston, 1995.
[3] P.J. Campo, M. Morari, Robust model predictive control, in: Proceedings of the 1987 American Control Conference, June 1987, pp. 1021–1026.
[4] E. Zafiriou, Robust model predictive control of processes with hard constraints, Comput. Chem. Eng. 14 (4/5) (1990) 359–371.
[5] H. Genceli, M. Nikolaou, Design of robust constrained model-predictive controllers with Volterra series, AIChE J. 41 (9) (1995) 2098–2107.
[6] Z.Q. Zheng, M. Morari, Robust stability of constrained model predictive control, in: Proceedings of the 1993 American Control Conference, 1993, pp. 379–383.
[7] J.H. Lee, B. Cooley, Recent advances in model predictive control and other related areas, in: J.C. Kantor, C.E. García, B. Carnahan (Eds.), Proceedings of Chemical Process Control—V, CACHE, AIChE, 1997, pp. 201–216.
[8] J.H. Lee, B.L. Cooley, Min-max predictive control techniques for a linear state-space system with a bounded set of input matrices, Automatica 36 (3) (2000) 463–473.
[9] J.H. Lee, Z. Yu, Worst-case formulations of model predictive control for systems with bounded parameters, Automatica 33 (5) (1997) 763–781.
[10] M.V. Kothare, V. Balakrishnan, M. Morari, Robust constrained model predictive control using linear matrix inequalities, Automatica 32 (10) (1996) 1361–1379.
[11] A. Casavola, M. Giannelli, E. Mosca, Min-max predictive control strategies for input-saturated polytopic uncertain systems, Automatica 36 (1) (2000) 125–133.
[12] Y. Lu, Y. Arkun, Polytope updating in quasi-min-max MPC algorithms, in: ADCHEM 2000 Proceedings, Pisa, Italy, 2000.
[13] Y. Lu, Y. Arkun, Quasi-min-max MPC algorithms for LPV systems, Automatica 36 (4) (2000) 527–540.
[14] T.A. Badgwell, Robust model predictive control of stable linear systems, Int. J. Control 68 (4) (1997) 797–818.
[15] T.A. Badgwell, A robust model predicitive algorithm for nonlinear plants. International Symposium on Advanced Control of Chemical Processes (ADCHEM), Banff, Canada, June 1997, pp. 477–481.
[16] O. Slupphaug, B.A. Foss, Bilinear matrix inequalities and robust stability of nonlinear multi-model MPC, in: Proceedings of the American Control Conference, Philadelphia, PA, June 1998.
[17] G. Pannocchia, D. Semino, Optimal modified models for robust predictive controllers, in: Proceedings of the 1999 IFAC World Congress, Beijing, China, 1999.
[18] D. Semino, G. Pannocchia, Use of different kinds of linear models in predictive control of distillation columns, in: ADCHEM 2000 Proceedings, Pisa, Italy, 2000.
[19] L. Chisci, J. Rossiter, G. Zappa, Robust predictive control with restricted constraints to cope with estimation errors, in: ADCHEM 2000 Proceedings, Pisa, Italy, 2000.
[20] L. Chisci, J.A. Rossiter, G. Zappa, Systems with persistent disturbances: predictive control with restriced constraints, Automatica 37 (7) (2001) 1019–1028.
[21] Y.I. Lee, B. Kouvaritakis, Robust receding horizon predictive control for systems with uncertain dynamics and input saturation, Automatica 36 (10) (2000) 1497–1504.
[22] B. Kouvaritakis, J.A. Rossiter, J. Schuurmans, Efficient robust predictive control, IEEE Trans. Auto. Cont. 45 (8) (2000) 1545–1549.
[23] M. Khammash, L. Zou, Analysis of steady-state tracking errors in sampled-data systems with uncertainty, Automatica 37 (6) (2001) 889–897.
[24] A. Bemporad, E. Mosca, Fulfilling hard constrains in uncertain linear systems by reference managing, Automatica 34 (4) (1998) 451–461.
[25] A. Bemporad, Reducing conservativeness in predictive control of constrained systems with disturbances, in: Proceedings of the 37th IEEE Conference on Decision and Control, Tampa, FL USA, 1998, pp. 1384–1389.
[26] P. Christofides, Robust control of nonlinear processes, in: Proceedings of the 1998 IFAC DYCOPS Symposium, Corfu, Greece, 1998, pp. 477–483.
[27] R. Freeman, P. Kokotovic, Robust integral control for a class of uncertain non-linear systems, in: Proceedings of the 34th Conference on Decision and Control, 1995, pp. 2245–2250.
[28] F. Blanchini, Control synthesis for discrete time systems with control and state bounds in the presence of disturbances, J. Optim. Theory Appl. 65 (1990) 29–40.
[29] J. Rossiter, B. Kouvaritakis, Youla parameter and robust predictive control with constraint handling, in: Workshop on Nonlinear Predictive Control, Ascona, Switzerland, 1998.
[30] J. Lee, K. Lee, W. Kim, Model-based iterative learning control with a quadratic criterion for time-varying linear systems, Automatica 36 (5) (2000) 641–657.
[31] M. Rodrigues, D. Odloak, Robust stable MPC via dynamic output feedback, in: ADCHEM 2000 Proceedings, Pisa, Italy, 2000.
[32] D. Megias, J. Serrano, G. de Prada, Min-max constrained quasi-infinite horizon model predictive control using linear programming, in: ADCHEM 2000 Proceedings, Pisa, Italy, 2000.
[33] G. Pannocchia, J.B. Rawlings, The Velocity Algorithm LQR: A Survey. Technical Report 2001-01, TWMCC, Department of Chemical Engineering, University of Wisconsin-Madison, May 2001.
[34] L. Marconi, A. Isidori, Mixed internal model-based and feedforward control for robust tracking in nonlinear systems, Automatica 36 (7) (2000) 993–1000.

[35] D. Kassmann, T. Badgwell, R. Hawkins, Robust steady-state target calculation for model predictive control, AIChE J. 46 (5) (2000) 1007–1024.

[36] S. Ralhan, T. Badgwell, Robust disturbance rejection for FIR systems with bounded parameters, in: ADCHEM 2000 Proceedings, Pisa, Italy, 2000.

[37] P.O. Scokaert, D.Q. Mayne, Min-max feedback model predictive control for constrained linear systems, IEEE Trans. Auto. Cont. 43 (8) (1998) 1136–1142.

[38] Y.J. Wang, Robust Model Predictive Control, PhD thesis, University of Wisconsin-Madison, 2002.

[39] S. Boyd, L.E. Ghaoui, Methods of centers for minimizing generalized eigenvalues, Linear Algebra Applications 188 (1993) 63–111.

[40] F. Alizadeh, J. Haeberly, M. Overton, A new primal-dual interior- point method for semidefinite programming, in: Proceedings of the 5th SIAM Conf. on Applied Linear Algebra, Snowbird, UT, 1994.

[41] Y. Nesterov, A. Nemirovsky, Interior-point polynomial methods in convex programming, in: Proceedings of the 5th SIAM Conf. on Applied Linear Algebra, Snowbird, UT, 1994.

[42] L. Vandenberghe, S. Boyd, A primal-dual potential reduction method for problems involving matrix inequalities, Mathematical Programming 69 (1) (1995) 205–236.

[43] Y.J. Wang, J.B. Rawlings, A new robust model predictive control method II: examples, Journal of Process Control, this issue.