An Improved Robust Distributed Model Predictive Control Based on Adaptive Feedback Weight

Wu Tongyan*, Wang Liang*, Wan Jiangwen*, Li Yayu**

** Beijing University of Aeronautics and Astronautics, Beijing, China (e-mail: zy1917301@buaa.edu.cn).*
*** No. 704 Research Institute of China Shipbuilding Industry Corporation, Shanghai, China.*

Abstract: This paper proposes an improved distributed robust model predictive control algorithm based on adaptive feedback weight for the polyhedral uncertain system with input saturation. This strategy decomposes the system into several subsystems but retains independent states, and assigns robust model predictive controllers to these subsystems of which the control goals are global optimization. The feedbacks of these controllers are formulated with the subsystems' independent states and the global state. In the presence of the control strategy, the weight changing the proportion of these states adaptively in the feedbacks can adjust the degree of coupling among the subsystems and hysteresis, so the subsystems transition from fast response to optimal stability. In a one-step iteration, the advanced strategy can not only make the steady-state value unaffected but also respond faster than the traditional distributed model predictive control algorithm. Finally, two simulation cases serve for study in the characteristics and advantages of this algorithm.

Keywords: Distributed model predictive control; Adaptive feedback weights; Full-dimensional subsystem state; Polytopic uncertain system

1. INTRODUCTION

Model predictive control (MPC) is an important control strategy in complex control systems and play a significant role in industrial processes (Morato et al., 2020; Saltık et al., 2018). The existing body of research on MPC suggests that the MPC algorithm can not only address the issue of the polyhedral uncertain system but also solve the problem with input saturation (Oravec and Bakošová, 2015). And experts used the nominal system for algorithm optimization (Ding et al., 2007). Generally, the term 'robust model predictive control (RMPC)' is interpreted as the MPC for uncertain systems. Researches, such as that conducted by Cao (Cao and Lin, 2005), had shown that the design of the RMPC with input saturation is transformed into a linear matrix inequality (LMI) optimization problem by rewriting the input saturation function. Huang (Huang et al., 2011) chose the weight of feedback control to improve Cao's algorithm. Wan (Wan and Kothare, 2003) and Ding (Ding et al., 2007) proposed the offline nominal system optimization to improve the speed of online computing. Oravec (Oravec and Bakošová, 2015) proposed a methodology for combining nominal system optimization and additional control input saturation, which was performed in the case study of the continuous stirred-tank reactor successfully.

Although extensive research has been carried out on RMPC, the algorithm did not perform as stable and real-time as robust distributed model predictive control (RDMPC) on large-scale systems (Scattolini, 2009). Recently, RDMPC has received remarkable attention in the literature. The aim of RDMPC is to decompose an uncertain system into several subsystems. By designing RMPC strategy for each subsystem, researchers obtained the subsystems control inputs holding for the centralized system control input (Y. Zhang and Li, 2007). Wang studied the amount of information exchanged between subsystem controllers and classified the RDMPC algorithm (Wang and Wen, 2008). The subsystems controllers operating in a completely independent fashion is called decentralized control. Some information is transmitted among subsystem controllers. Hence these controllers can use the behaviour of others to make decisions, which is called distributed control. The distributed control can be divided into two types based on different optimization objectives (Scattolini, 2009; Y. Zhang and Li, 2007): Nash-based or cooperative RDMPC. The cooperative RDMPC, proposed by Gherwi (Al-Gherwi et al., 2011), can solve subsystem optimization problems and achieved global optimization similar to RMPC. Zhang (L. Zhang et al., 2013) adopted the saturation-dependent Lyapunov function to reduced conservatism of the cooperative RDMPC. Shalmani (Shalmani et al., 2020) proposed an algorithm for an iterative Nash-based to achieve the overall optimal solution of the whole system in a partially distributed fashion. In order to deal with the problem of centralized target optimization under multiple constraints of subsystems, the researchers (Necoara et al., 2010) proposed a primal-dual decomposition method for solving convex optimization problems, which can increase the calculation speed. Besides, Gherwi (Al-Gherwi et al., 2013) used a closed-loop dual-mode approach to reduce the demanding computations of on-line. Such approaches, however, had failed to address the iteration problem. In view of the subsystem control solution, multiple iterations are required in each step of the control, so that the system performance can converge to that of the centralized RMPC. The insufficient iterative RDMPC algorithm account for performance degradation, and the control effectiveness is not as good as RMPC algorithm. Iterations can be reduced only if computational error is acceptable. Some iterations of these algorithms were inevitable for the case of the negotiation among subsystems is non-real-time. Multiple iterations will make the RDMPC algorithm lose its real-time feature. Therefore, the major goal of this work is to get better performance in one-step iteration.

This paper attempts to add some theories of RMPC and decentralized control to improve RDMPC algorithm. We preserve the full dimensional states of subsystems and assume that these states can be estimated from the partial states of the centralized system. The feedbacks of subsystems in our RDMPC are formed from the centralized system states and the subsystems' states. The weights belonging to these states adapt themselves for optimal control effects over time. Besides, we simplify the algorithm by nominal system optimization. Finally, two simulation cases presented demonstrate our strategy, indicating that these improvements can achieve a satisfactory level of performance in one-step iteration.

This work is organized as follows: In Section 2 the model is presented. The preliminaries of input saturation are discussed in this section. In Section 3, review traditional algorithms and introduce a new algorithm. The analysis and stability proof of the algorithm are in Section 4. In Section 5, the application of the algorithm is illustrated using two polyhedral uncertain systems cases. And conclusions are presented in Section 6.

2. PROBLEM FORMULATION AND PRELIMINARIES

2.1 Problem Formulation

It is assumed that Ω can be described as the convex hull of the family of linear polytopic uncertain system, Consider the linear time-variant discrete system is

$$x(k+1) = A(k)x(k) + B(k)u(k), x(0) = x_0 \tag{1}$$

$$[A(k), B(k)] \in \Omega, \Omega = convhull(\{[A_l, B_l], \forall l \in [1,2,\cdots,L]\}) \tag{2}$$

where $x(k) \in R^n$, $u(k) \in R^m$are the state vectors and control inputs, respectively, k is the discrete-time index. $A(k) \in R^{n\times n}$means the matrix of system, $B(k) \in R^{n\times m}$denotes the matrix of system inputs. Parameter l is the number of vertices of a family of uncertain matrices, L is the sum of l. According to the property of convex hull, the uncertainty of polyhedral system can be derived as

$$[A(k), B(k)] = \sum_{l=1}^{L} \lambda_l [A_l, B_l], \sum_{l=1}^{L} \lambda_l = 1 \tag{3}$$

where λ_l is a scalar and $0 < \lambda_l < 1$.

Divide up the input matrix $B(k)$ and control input $u(k)$ of the system by the method in (Al-Gherwi et al., 2011).

$$x(k+1) = A(k)x(k) + \begin{bmatrix} B_1(k) & \dots & B_M(k) \end{bmatrix}\begin{bmatrix} u_1(k) & \dots & u_M(k) \end{bmatrix}^T \tag{4}$$

In this method, the control input is decomposed into M parts. Each of the parts satisfies $u_i(k) \in R^{m_i}$, $m = \sum_{i=1}^{M} m_i$.

Similarly, $B(k)$ is decomposed into M parts and $B_i(k) \in R^{n\times m_i}$. The system matrix $A(k)$ and system states $x(k)$ are retained.

Assuming that the overall system is controllable. For design RDMPC, the pth subsystem model can be defined by rewriting (4) as

$$x_p(k+1) = A(k)x_p(k) + B_p(k)u_p(k) + \sum_{q=1,q\neq p}^{M} B_q(k)u_q(k) \tag{5}$$

where $x_p(k) \in R^n$, $u_p(k)$ is the manipulated input of the pth subsystem obtained through subsystem p control optimization, i.e., u_{else} . $B_p(k)$ denotes the input matrix for $u_p(k)$. $B_q(k)$ and $u_q(k)$ are the input matrices and manipulated inputs belonging to the other subsystems. The centralized system is decomposed into M subsystems. Designing RMPC algorithm for each subsystem and solving the LMIs optimization problems enable the subsystems to figure out the control inputs. Combining these inputs into a centralized system input reaches the centralized system control.

The purpose of this strategy is to reduce the dimension of the input matrix, so that it can relieve the burden of the computation. As mentioned in the literature (Copp et al., 2019), the number of inequalities involved in the optimal solution by typical RMPC is $(m + 1 + L \times 2m)$. While each subsystem has only $(m_i + 1 + L \times 2m_i)$ inequality conditions to solve in the RDMPC algorithm, where m_i is the dimension of $u_i(k)$, and $m_i < m$. Accordingly, the strategy can provide high solving speed in optimization problem.

2.2 Input Saturation

In our method, the saturation characteristic of the control input is considered, which is $|u_{sat}| < |u_{max}|$.

According to (Cao and Lin, 2005; Casavola et al., 2000), the saturation function can be shown as (6). $F, H \in R^{m_p\times n}$ are two feedback gains of subsystem p. And

$\forall x_p(k) \in \chi(H_p(k)) = \{x_p(k): |H_p(k)x_p(k)|_\infty \leq u_{max}$

We have

$$u_{sat} \in convhull(E_j F_p(k)x_p(k) + E_j^- H_p(k)x_p(k)), j = 1,\cdots,2^{m_p} \tag{6}$$

where m_p is the input vector dimension of subsystem p. Suppose that E_j is all the matrices whose diagonal elements are either 1 or 0, with amount is 2^{m_p}. Considering $E_j^- = I - E_j$, formula (6) can be represented as:

$$u_{sat} = \sum_{i=j}^{2^{m_p}} \lambda_j (E_j F_p(k)x_p(k) + E_j^- H_p(k)x_p(k)) \qquad \forall x_p(k) \in \chi(H_p(k)) \tag{7}$$

where $\sum_{j=1}^{2^{m_p}} \eta_j = 1, \eta_j > 0$. By formula (7), we can add the saturation function into LMI.

3. AN IMPROVED RDMPC ALGORITHM DESIGN

3.1 Traditional Robust Distributed Model Predictive Control

Designing the RMPC algorithm solves the output u_i of each subsystem based on (5), in the sense that solving the min-max problem shown in (8) for subsystem p.

$$\min_{u_p(k+n|k)} \max_{[A(k+n),B_p(k+n),B_q(k+n)]\in\Omega} J_p(k), n \geq 0 \tag{8}$$

Suppose that the diagonal weighting matrix of the states in subsystem p is W_x>0 and the manipulated input diagonal weighting matrix for all subsystems is W_{ui}>0, i=1…, M. The subsystem cost index $J_p(k)$ can be defined as

$$J_p(k) = \sum_{n=0}^{\infty}(\|x_p(k+n|k)\|_{Wx}^2 + \sum_{i=1}^{M}\|u_p(k+n|k)\|_{Wu_i}^2) \tag{9}$$

Based on (9), each subsystem has a close cost index to that of the centralized system mentioned in the literature (Kothare et al., 1996; Schuurmans and Rossiter, 2000), as well as the optimal solution obtained by solving (8) of each subsystem is similar to that obtained by solving the RMPC problem of the centralized system. Considering input saturation and $l \geq 0$, formula (5) can be rewritten as

$$\begin{aligned} x_p(k+l+1|k) &= A(k+l)x_p(k+l|k) \\ &+ B_p(k+l)u_{sat,p}(k+l) \\ &+ \sum_{q=1,q\neq p}^{M} B_q(k+l)u_{sat,q}(k+l|k) \end{aligned} \tag{10}$$

According to formula (3), B_p and B_q also satisfy that $B_q = \sum_1^L \lambda_i B_{q,l}$, $B_p = \sum_1^L \lambda_i B_{p,l}$, where $B_{p,l}$ and $B_{q,l}$ are obtained by decomposing the vertices of the input matrix B_l in (2). Substitute (7) into (10), and substitute x_q by x_p in the presence of system p solution,

$$\begin{aligned} x_p(k+l+1|k) &= A(k+l)x_p(k+l|k) \\ &+ B_p(k+l)(E_{p,j}F_p(k) + E_{p,j}^{-}H_p(k))x_p(k+l|k) \\ &+ \sum_{q=1,q\neq p}^{M} B_q(k+l)\hat{F}_q(k)x_q(k+l|k), j = 1,\cdots,2^{m_p} \end{aligned} \tag{11}$$

where $A_p(k+l) = A(k+l) + \sum_{q=1,q\neq p}^{M} B_q(k+l)F_q(k)$, and F_q is other subsystems' feedback gains received from subsystem q via communication.

For the purpose of simplifying the min-max problem, define the quadratic function as

$$V_p(x_p(k+l|k)) = x_p^T(k+l|k)P_{p,k}x_p(k+l|k) \tag{12}$$

Suppose that V_p meets the following condition to keep the system stable (L. Zhang et al., 2013).

$$\begin{aligned} V_p(x_p(k+l+1|k)) - V_p(x_p(k+l|k)) \leq \\ -x_p^T(k+l|k)W_x x_p(k+l|k) \\ -u_p^T(k+l|k)W_{up}u_p(k+l|k) \\ -\sum_{q=1,q\neq p}^{M} u_q^T(k+l|k)W_{uq}u_q(k+l|k) \end{aligned} \tag{13}$$

A stable system must satisfy $V_p(x_p(\infty|k)) = 0$. We thus get (14) by adding both sides of (13) from l =0 to l =∞.

$$J_p(k) \leq V_p(x_p(k|k)) \leq \gamma_p(k) \tag{14}$$

where γ is the upper bound of J, We can give the following Semidefinite Program (SDP) to get γ for each subsystem based on nominal system optimization (Wan and Kothare, 2003) and additional control input saturation (Cao and Lin, 2005).

$$\min_{Q_p(k)>0,\gamma_p(k),Y_p(k),Z_p(k)} \gamma_p(k) \tag{15}$$

s.t.

$$\begin{bmatrix} I & x_p(k|k)^T \\ x_p(k|k) & Q_p(k) \end{bmatrix} \geq 0 \tag{16}$$

$$\begin{bmatrix} I & Z_p^T(k) \\ Z_p(k) & Q_p(k) \end{bmatrix} > 0 \tag{17}$$

$$\begin{bmatrix} Q_p(k) & * & * & * \\ A_p^{(0)}Q_p(k) + B_p^{(0)}(E_{p,j}Y_p(k) + E_{p,j}^{-}Z_p(k)) & Q_p(k) & * & * \\ (Q + \sum_{q=1,q\neq p}^{M} \hat{F}_q^T(k)W_{uq}\hat{F}_q(k))^{\frac{1}{2}}Q_p(k) & 0 & I_{\gamma_p(k)} & * \\ W_{up}^{\frac{1}{2}}(E_{p,j}Y_p(k) + E_{p,j}^{-}Z_p(k)) & 0 & 0 & I_{\gamma_p(k)} \end{bmatrix} > 0 \tag{18}$$

$$\begin{bmatrix} Q_p(k) & * \\ A_p^{(i)}Q_p(k) + B_p^{(i)}(E_{p,j}Y_p(k) + E_{p,j}^{-}Z_p(k)) & Q_p(k) \end{bmatrix} > 0 \tag{19}$$

$$i = 1,\cdots,L;\ j = 1,\cdots,2^{m_p}$$

where $Q_p(k) = \gamma_p(k)P_P^{-1}(k)$ is the invariant set of state variables, $F_p(k) = Y_p(k)Q_P^{-1}(k)$ is the control input at t=k, optimal control inputs can be formulated with $u_p(k) = F_p(k)x_p(k)$, $Z_p(k) = H_p(k)Q_P^{-1}(k)$, $I_{\gamma_p(k)}$ is $\gamma_p(k)$ multiplied by the unit matrix of the appropriate dimension. $B_p^{(0)}, B_q^{(0)}$derive from $B^{(0)}$. The $B^{(0)}$ and $A^{(0)}$ are the analytic centre of the polytope Ω.

The feedback gains $F_i(k)$ of subsystems are obtained by solving SDP of (15). The subsystem control objectives are close to the centralized system control objectives, so that the $F_i(k)$ obtained by solving for each subsystem is similar to

$F(k)$ obtained by solving RMPC for the centralized system. Meanwhile the centralized control input $u(k)$ is formulated with the optimal control inputs $u_i(k)$ got from SDP of all subsystems. The initial values for all subsystems' state are x_0. Due to different B_i and the delay of the u_{else} defined in (5), subsystems retain divergent states at the beginning of control. The gaps among the states decrease gradually with the control process since all subsystems will convert their states into the centralized system stable state eventually.

Literature (Al-Gherwi et al., 2011) proposed an algorithm in which the subsystems take into account the same state. The value of this state is determined by the global state $x(k)$ solved relying on (4) for the centralized system. The researchers simplified the processing of states and proved that the algorithm is stable. Besides, they had established that multiple iterations are used to obtain a state-feedback law until the error was tolerable in the current control cycle, but it may cost plenty of time. The main reasons for iterations in the previous algorithm are

a. x_q in formula (11) is substituted by x_p, which causes errors in the coupling term of the subsystem. According to the literature (Al-Gherwi et al., 2011), multiple iterations will eliminate this error.

b. In the act of solving the feedback gain $F_p(k)$ of the subsystem p, the algorithm needs to use the feedback gains $F_q(k)$ of other systems. In order to ensure the feasibility of the algorithm, generally, $F_q(k-1)$ is used instead of the $F_q(k)$.

Literature (Siljak, 2011) defined the strategy using independent states to be referred as decentralized model predictive control. This algorithm can realize the local optimization for the subsystems but not the global optimization. It should be pointed out that it responds faster than the cooperative distribution model predictive control algorithm.

3.2 Cooperative Adaptation Robust Distributed Model Predictive Control

Based on the rapidity of decentralized model predictive control algorithm, this work reserved a set of states for each subsystem that had the same dimensions as the centralized state, and propose a method applying adaptive feedback weights to subsystems states feedback to obtain the global optimization. we select a set of adaptive weights and the improved subsystems feedback states are formulated as

$$\tilde{x}_p(k) = \zeta_p(k)x_p(k) + (1\text{-}\zeta_p(k))x(k)\text{=}g(k, x_p, x, \zeta_p) \tag{20}$$

where $\zeta_p \in R^M$ is the weight of states belong to subsystem p, $1-\zeta_p$ is the weight acting on the state of the centralized system, $\zeta_p(k) \in [0,1]$ is a factor used to weaken the influence of x_p. The function g represents the input-output relationship of formula (20).

Considering (20), and imitating the SDP of the previous algorithm, the optimization problem of the new control algorithm for subsystem p will be transformed into (21).

$$\min_{Q_p(k)>0,\gamma_p(k),Y_p(k),Z_p(k),\zeta_p} \gamma_p(k) \tag{21}$$

s.t.

$$\begin{bmatrix} I & \tilde{x}_p(k\mid k)^T \\ \tilde{x}_p(k\mid k) & Q_p(k) \end{bmatrix} \geq 0 \tag{22}$$

Algorithm 1. Cooperative adaptation robust distributed model predictive control (CARDMPC) algorithm.

Step 0: Select a set of initial feasible control inputs $F_i(0)$ and same initial states $x(0)$ for subsystems.

Step 1: When control interval t=*k*, directly collect or indirectly estimate $x_i(k)$ of each subsystem i based on the centralized system state $x(k)$, and solve SDP for (21) to determine $Q_i(k), \gamma_i(k), Y_i(k), Z_i(k), \zeta_i(k)$. Get the state feedback gain matrices $F_i(k)$ of subsystems by $F_i(k) = Y_i(k)Q_i^{-1}(k)$ and the improved subsystems feedback states$\tilde{x}_i(k)$ by (20). The subsystem control inputs are formulated as $u_i(k) = F_i(k)\tilde{x}_i(k)$.

Step 2: Apply $u_i(k) = F_i(k)x_i(k)$, $u_{else}(k) = u_{else}(k-1)$ to the subsystems by (5) and record the state $x_i(k+1)$ of these subsystems at next interval.

Step 3: Combine the $u_i(k)$ obtained in Step 1 into the centralized system control input $u(k)$. Then update the state $x(k+1)$ of the centralized system according to (1).

Step 4: Increase the control interval k = k + 1, return to step 1 and repeat the procedure.

This method is proposed based on the characteristics of decentralized and distributed algorithms. Firstly, by decentralized algorithm, we adopt the state $x_p(k)$ of the subsystem p itself to replace the states $x_{q,q\neq p}(k)$ of the other subsystems in (11), so as to reduce the interaction among the subsystems (Siljak, 2011). But the algorithm could only achieve the local subsystem optimization if feedback states are always $x_p(k)$. Feedback states of subsystems need to be corrected and converge to the centralized states defined in distributed algorithms during the control process gradually. In this case, there is a reduced demand in the impact of $x_p(k)$ on state feedback and an increase demand in the weight of $x(k)$ on the state feedback during this process which means adjust weight adaptively in the control process.

4. MECHANISM AND ROBUST STABILITY ANALYSIS

In this part, The CARDMPC algorithm designed in 3.2 will be analysed further, including the working principle of adaptive algorithm, the approach to improve response speed, and algorithm stability analysis.

4.1 Mechanism Analysis Of The Improved Algorithm

Theorem 1. For a controllable system, when $\zeta_p \in (0,1)$, CARDMPC algorithm retains the other subsystems feedback gains F_q with hysteresis, and puts the gains into the state update equation of the subsystem. As the control process goes by, the hysteresis of F_q decreases gradually. Finally, the CARDMPC algorithm becomes the traditional collaborative RDMPC algorithm.

Proof: The subsystem state update equation of the improved distributed robust model predictive control is (23).

$$\begin{aligned} x_p(k+1) &= h(\tilde{x}_p(k)) = A(k)\tilde{x}_p(k) + B_p(k)F_p(k)\tilde{x}_p(k) \\ &+ \sum_{q=1,q\neq p}^{M} B_q(k)F_q(k-1)\tilde{x}_p(k) \end{aligned} \tag{23}$$

h is a function describing the iteration of the subsystem state. Algorithm 1 is used to solve $F_p(k)$, which can be regarded as a function f. $F_p(k) = f(\tilde{x}_p(k), F_q(k-1))$, where $F_q(k-1)$ is obtained by communication. Suppose the initial feedback gain and initial state of each subsystem are same, as $F_1(0) = F_2(0) = \cdots = F_M(0), x(0) = x_1(0) = x_2(0) = \cdots = x_M(0)$.

After the feedback gains $F_i(k)(i = 1,2\cdots,M)$ of all subsystems have been fixed, the system state of the centralized system can be updated by the formula (24).

$$\begin{aligned} x(k+1) &= h_1(x(k)) \\ &= A(k)x(k) + [B_1(k)\ldots B_M(k)][F_1(k)\ldots F_M(k)]^T x(k) \end{aligned} \tag{24}$$

$\tilde{x}_p(k)$ of the formula (23) is defined in (20). When $\zeta_p = 0$ for subsystem p, the new algorithm becomes the traditional collaborative RDMPC. $\tilde{x}_p(k) = x(k)$, $x(k)$ is only determined by formula (24). The control function is $F_p(k) = f(x(k), F_q(k-1))$. In this case, the algorithm does not involve formula (23). x_p is neither employed nor calculated.

When $\zeta_p = 1$, the algorithm becomes a decentralized RDMPC. $\tilde{x}_p(k) = x_p(k)$, $x_p(k)$ is only determined by formula (23). The control function is $F_p(k) = f(x_p(k), F_q(k-1))$. In this case, the algorithm does not involve formula (24). The formula (23) can be rewritten as formula (25). The interaction between subsystems is incomplete because of F_q, which has a one-step delay. The assumption cannot obtain the global optimization.

$$\begin{aligned} x_p(k+1) &= A(k)x_p(k) + B_p(k)F_p(k)x_p(k) \\ &+ \sum_{q=1,q\neq p}^{M} B_q(k)F_q(k-1)x_p(k) \end{aligned} \tag{25}$$

When $\zeta_p \in (0,1)$, the CARDMPC algorithm will perform normally. Under the same initial conditions, and assuming $F_q(-1) = F_q(0)$, the state of traditional cooperative RDMPC and CARDMPC algorithms can be determined by (23) and (24) at k=1. The result is $x(1) = x_p(1)$. Then update the state at k=2, as shown in formula (26) and (27).

$$\begin{aligned} x(2) &= h_1(x(1)) \\ &= A(1)x(1) + [B_1(1)\ldots B_M(1)][F_1(1)\ldots F_M(1)]^T x(1) \end{aligned} \tag{26}$$

$$\begin{aligned} x_p(2) &= h(g(x_p(1), x(1), \xi_p(1))) \\ &= A(1)\tilde{x}_p(1) + B_p(1)F_p(1)\tilde{x}_p(1) \\ &+ \sum_{q=1,q\neq p}^{M} B_q(1)F_q(0)\tilde{x}_p(1) \end{aligned} \tag{27}$$

where F_q used in (26) delay one step behind that in (27). At the early stage of control, generally, the feedback gain F promotes the system to reach stability gradually, while $|F|$ itself tends to decrease from large to small. Accordingly, $|F_q(0)| > |F_q(1)|$. This hysteresis may lower the system performance γ temporarily. But a larger $|F|$ is good for the regulation of x i.e., $|x_p(2)| < |x(2)|$. At the middle and later stages of control, the hysteresis may cause overshooting of state or long-setting-time. Therefore, it's necessary to eliminate the hysteresis of feedback gain gradually in the control process. Finally, the system can restore the global optimization defined by traditional collaborative DRMPC algorithm.

When $k>1$, by the functions $g$and h, $\tilde{x}_p(k+1)$ can be further expressed as

$$\begin{aligned} \tilde{x}_p(k+1) &= g(x_p(k+1), x(k+1), \zeta_p(k+1)) \\ &= g(h(\tilde{x}_p(k)), h_1(x(k)), \zeta_p(k+1)) \end{aligned} \tag{28}$$

As long as $0 < \zeta_p < 1$, based on the formulation of h, $\tilde{x}_p$ will converge to x and the hysteresis effect will be smaller over time. In other words, the system will achieve the global optimization as defined in equation (8) eventually. A constant ζ_p will also satisfy the definition in (3.2). The variation of ζ_p does not need to be take into account. Based on the above analysis, the algorithm only needs to determine a constant value of ζ_p in application. As long as the magnitude of ζ_p is fixed, the system can automatically eliminate the bad influence of hysteresis in the control process.

Case 1 noted in section 5 has proved this part in practice setting different constant values for ζ. The algorithm in section 3.2 can solve the dynamic value ζ, which not only increases the calculation burden, but also the weight optimization result will quickly drop to 0 in the calculation process. It is worth noting that the maximum ζ can be obtained at the beginning of solving the optimization. Because $\forall\zeta \in [0,1]$ can gradually attenuate the influence of $x_p(k)$, this article chooses the average of the ζ optimization results of the first three-time intervals as the constant value of ζ. This method will be applied to the weight solution in Case 2.

According to the analysis, the initial value selection of ζ is related to multiple parameters such as the uncertainty stability, setting time and Initial state of system. In the future, the relationship between the optimal value of ζ and system parameters will be further studied.

4.2 Convergence and Stability of the CARDMPC

The stability analysis of the system is mainly to verify the convergence of the system error. Roman (Roman et al., 2020) studied the stability of nonlinear algorithms by analysing limit cycle. Rigatos (Rigatos et al., 2017) proved the stability of H_∞ algorithm in application by Lyapunov stability analysis. The new algorithm uses positive robust invariant set (22), which can construct the Lyapunov function conveniently.

Theorem 2. (Kerrigan, 2001) For subsystem p, a set of solutions, $Y_p(k)$,$Q_p(k)$,$\gamma_p(k)$, are obtained by solving the problem (15) when t=k without feedback weight ζ. As a result, the ellipsoid $E_p = \{x_p: x_p^T Q_p^{-1} x_p \le 1\}$ is the robust invariant set of the subsystem p concerning the controller $u_p(k) = Y_p(k)Q_p^{-1}(k)x_p(k)$.

Proof: If the problem (15) is solvable, we can get (29) from (15).

$$x_p^T(k)Q_p^{-1}(k)x_p(k) \le 1 \tag{29}$$

When $l > 0$, the right term of (13) must be negative, so the left term of (13) as:

$$V_p(x_p(k+l+1\,|\,k)) - V_p(x_p(k+l\,|\,k)) \le 0 \tag{30}$$

Inequality (30) can be rewritten as following in combination with (12).

$$\begin{aligned} &x_p^T(k+l+1)Q_p^{-1}(k)x_p(k+l+1) \\ &\quad \le x_p^T(k+l)Q_p^{-1}(k)x_p(k+l) \end{aligned} \tag{31}$$

It means that:

$$x_p^T(k+l)Q_p^{-1}(k)x_p(k+l) \le 1 \tag{32}$$

and proves that E_p is the robust invariant set of the controller $u_p(k) = Y_p(k)Q_p^{-1}(k)x_p(k)$ got by solving (21).

Lemma 1. (Al-Gherwi et al., 2011) In the solution process, when the states of the subsystems are the same as the centralized system, the invariant set E_c of the centralized system is the intersection of the invariant sets E_i of the subsystems.

By lemma 1, the centralized system state is used for the optimization of all subsystems. But our strategy retains subsystems states. Observing (4) and (5), subsystem state equations are reformulated from the centralized system state equation. And the optimization goal (8) of each subsystem includes the overall state and all control variables. The subsystem control inputs are calculated each other via communication. It means that the control variables obtained by each subsystem through the algorithm is close to the global optimization in the control process but are only manipulated by the corresponding subsystems. In addition, the combination of all subsystem solutions is a globally optimal solution. The reserved state of the subsystem also converges to the centralized system state with the control process. So new algorithm corresponds to lemma 1. We get:

Inference 1. For CARDMPC algorithm, the states of each subsystem converge to the states in RDMPC algorithm and have the following properties: the intersection of the robust invariant sets E_i obtained by the solution of the subsystems contains the robust invariant set E_c of the centralized system finally.

The robust invariant set of centralized systems will gradually converge proved in (Kerrigan, 2001). According to Inference 1, the new algorithm will gradually approach the traditional algorithm, and the intersection of the invariant sets E_i of subsystems will be the robust invariant set E_c of centralized systems finally. Therefore, the final result of the subsystem obtained by the new algorithm will also reach convergence.

Theorem 3. For the CARDMPC algorithm, the initial states of the subsystems are all x_0. And these subsystems are solvable for the optimization problem (21) at k=0. Accordingly, the system using this control algorithm is closed-loop asymptotic stable.

Proof: For $\forall i \ge 0$, when the control input is $u_p(k)$, the robust invariant set of the subsystem p's state x_p is E_p as:

$$x_p(k+i) \in E_p = \left\{x_p : x_p^T Q_p^{-1}(k)x_p \le 1\right\} \tag{33}$$

Due to the centralized system control input is the combination of all subsystem control inputs, the robust invariant set E_c of the centralized state x is

$$x(k+i) \in E_c \tag{34}$$

By Inference 1, we can get:

$$x(k+i) \in E_p \tag{35}$$

Based on the definition of (20), $\tilde{x}_p(k+1)$ satisfies the inequation as:

$$\tilde{x}_p(k+i) \le \max(x(k+i), x_p(k+i)) \tag{36}$$

Combining (33) and (35), we can get (37) as:

$$\tilde{x}_p(k+i) \in E_p \tag{37}$$

According to theorem 2, E_p is the robust invariant set of the controller $u_p(k)$. Then the (31) can be reformulated as:

$$\tilde{x}_p^T(k+1)Q_p^{-1}(k)\tilde{x}_p(k+1) \le \tilde{x}_p^T(k)Q_p^{-1}(k)\tilde{x}_p(k) \le 1 \tag{38}$$

In the process of solving the problem (21), only invariant set constraint takes system state update into account. Therefore, a set of optimal solutions of the subsystem p $Y_p(k)$,$Q_p(k)$ and $\gamma_p(k)$ obtained at t=k are still feasible solutions of the subsystem when t=k+1. We have:

$$\tilde{x}_p^T(k+1)Q_p^{-1}(k+1)\tilde{x}_p(k+1) \le \tilde{x}_p^T(k+1)Q_p^{-1}(k)\tilde{x}_p(k+1) \tag{39}$$

Substituting (39) into (38), we get

$$\tilde{x}_p^T(k+1)Q_p^{-1}(k+1)\tilde{x}_p(k+1) \le \tilde{x}_p^T(k)Q_p^{-1}(k)\tilde{x}_p(k) \tag{40}$$

which can be rewritten as:

$$\tilde{x}_p^T(k+1)P_p(k+1)\tilde{x}_p(k+1) \le \tilde{x}_p^T(k)P_p(k)\tilde{x}_p(k) \qquad (41)$$

Suppose the Lyapunov function of subsystem is $V(\tilde{x}_p) = \tilde{x}_p^T(k)P_p(k)\tilde{x}_p(k) > 0$, It shows that the Lyapunov function of the algorithm is monotonically decreasing, i.e., $\dot{V}(\tilde{x}_p) \le 0$. At the same time, $V(\tilde{x}_p)$ satisfies radially unbounded. When k=0, all subsystems are solvable for the problem (21), all subsystems are closed-loop asymptotically stable(JJE Slotine, 1991). The state of subsystems will gradually converge to the equilibrium point $x = 0$. In addition, the control goal of the algorithm is global optimization, so the centralized system is also progressively stable.

5. CASE STUDIES

In this part, we modified the basic functions of the MUP toolbox (Version 20200224) (Bakošov'a and Oravec, 2014) to implement our algorithm for the cases. In the toolbox, we choose the YALMIP toolbox (Version R20200116) for optimal planning of (21) and solve via solver SeDuMi (Version 1.3). These cases were simulated by MATLAB R2017b under the condition of CPU I7-8550U, 1.8GHz computer. By comparing the control performance of different cases, we can analyse the characteristics of our CARDMPC algorithm.

5.1 Case Study 1

Consider a third-order uncertain system that has been used for traditional RDMPC (Wang and Wen, 2008). In the system, the vertices of the system matrix and the input matrix are

$$A^{(1)} = \begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, A^{(2)} = \begin{bmatrix} 1 & 1.5 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B^{(1)} = \begin{bmatrix} 0.5 & 1.67 \\ 1 & 0.5 \\ 0 & 1 \end{bmatrix}, B^{(2)} = \begin{bmatrix} 0.6 & 1.7 \\ 1 & 0.6 \\ 0 & 1 \end{bmatrix}$$

Suppose $u(k) \in [-1,1]$ is the saturation limited input, $x(k) \in [-2,2]$ is the saturation limited state. $x(0) = [2,0.5,1.5]^T$ is the initial state of the system. Based on (L. Zhang et al., 2013), we decompose the centralized system into two subsystems of which the initial states are both $x(0)$, and use the collaborative adaptive RDMPC to design the controllers for the two subsystems respectively. We set $W_x = I_{3\times3}$, $W_{u1} = W_{u2} = I_{2\times2}$ and substitute the following uncertainty coefficients into the system by (3).

$$\lambda_1 = 0.2 + 0.2\cos(0.2k\pi), \lambda_2 = 0.8 - 0.2\cos(0.2k\pi) \qquad (42)$$

In order to study the influence of the feedback weights on the system, we substitute $\zeta = 0.5$, $\zeta = 0.3$, and $\zeta = 0.1$ into the control algorithm respectively. The results of the centralized system are shown in the figures below.

According to figure 1-5, it shows that the feedback weight affects the control performance apparently. A large weight will cause states overshoot and control inputs fluctuation in the system. It means that the ζ is so large that the time-delay effect cannot be attenuated in time, which exceeds the adaptive capacity of the algorithm. It means the large weight affects the robustness of the system. On the contrary, the final values of the system state under different weight are the same, which is consistent with the analysis of Inference 1. While choosing the appropriate feedback weight will make the system respond faster without overshooting and loss of robustness. In essence the algorithm with appropriate ζ can adjust the weight adaptively to eliminate the adverse effects caused because of hysteresis.

In order to express the cost of different algorithms conveniently, we define the current cost index function as:

$$J_c(k) = \sum_{n=0}^{k}(\|x(k+n|k)\|_{W_x}^2 + \sum_{i=1}^{M}\| u_p(k+n|k)\|_{W_{ui}}^2) \qquad (43)$$

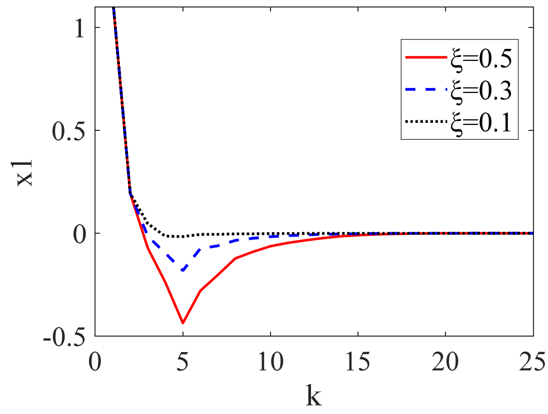

Fig. 1. Dynamic responses of x_1 with different ζ.

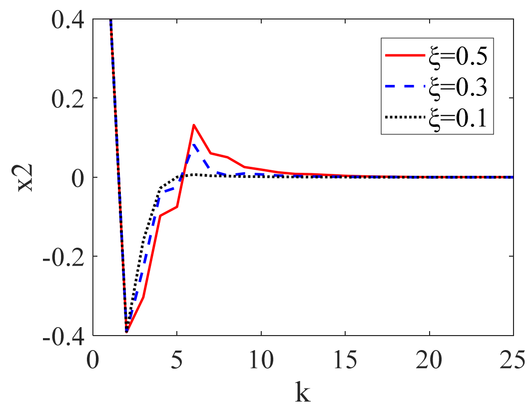

Fig. 2. Dynamic responses of x_2 with different ζ.

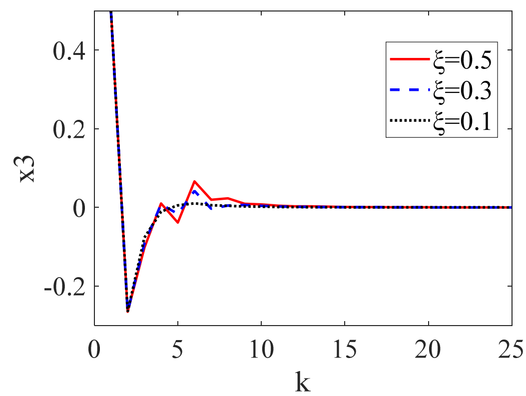

Fig. 3. Dynamic responses of x_3 with different ζ.

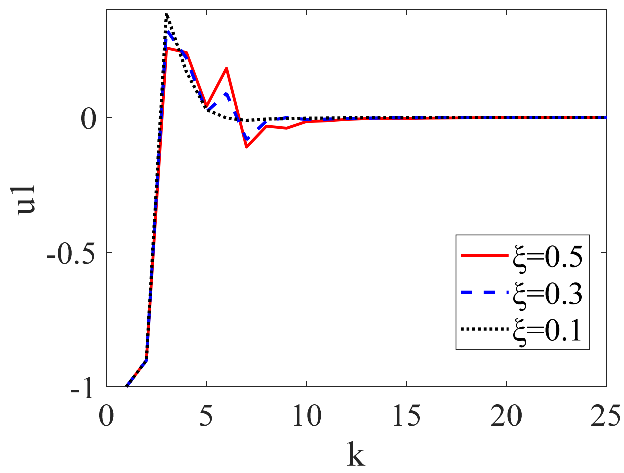

Fig. 4. Dynamic responses of u_1 with different ζ.

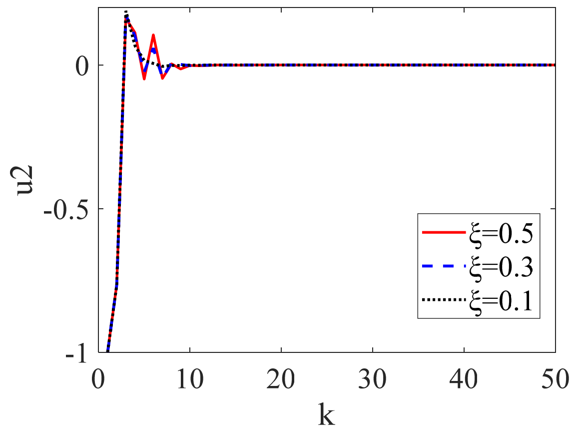

Fig. 5. Dynamic responses of u_2 with different ζ.

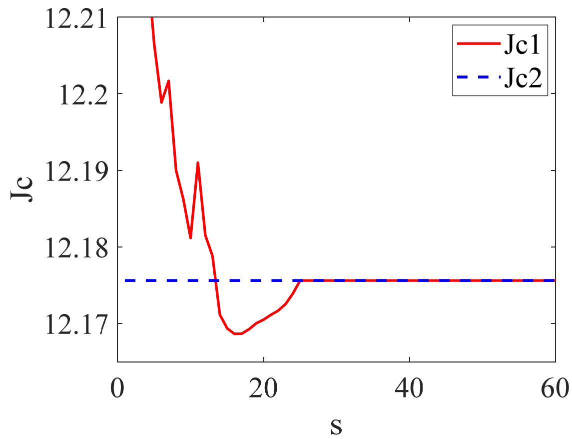

Fig. 6. Final value of J_c with different s

where $x(k)$ is the state of the centralized system. Compared with (43) and (9), the state in (43) is the centralized system state. J_c directly represents the global performance, the state in (9) belongs to the subsystem, J_p is the subsystem performance. Meanwhile, (9) involves the infinite time domain, and (43) is solved in the finite time domain. When the time k is long enough, J_p will be equal to J_c.

We designed an experiment to determine the optimal value of ζ. In the experiment, we set the maximum ζ to 0.3, reduce ζ according to formula (44), and recorded the cost index J_c at the end of control. The results are shown in figure 6.

$$\zeta_1 = \zeta_2 = 0.3 - 0.3 * s / 60 \tag{44}$$

In formula (44), s is the slope adjustment coefficient. The change step is 1, and $s \in [1,60]$.

In figure 6, J_{c1} is the current cost index with different s, J_{c2} is the current cost index of the traditional RDMPC algorithm. This experiment shows that different feedback weights will get different results. The appropriate value of ζ can achieve a lower cost index than that of the traditional algorithm.

Table 1 lists the final index of J_c for various values of ζ_i.

Table 1. Final value of J_c with different cases.

Cases	Final value of J_c
ζ_i=0 (traditional algorithm)	12.1756
ζ_i=0.1	12.1689
ζ_i=0.3	12. 2565
ζ_i=0.5	12.6949
ζ_i=0.21(optimum)	12.1681

It can be found from the table 1 that the new algorithm can improve the performance by selecting the appropriate feedback weight. Unsuitable coefficient ζ_i may have poor performance. Therefore, after determining a series of parameters of the system, we can also determine the appropriate weight ζ through experiments.

5.2 Case Study 2

A discrete state-space model proposed by Zhang to study distributed model predictive control of positive systems in literature (J. Zhang et al., 2020). The two-vertex uncertain model can be described as:

$$A^{(1)} = \begin{bmatrix} 0.34 & 0.36 & 0.35 \\ 0.35 & 0.33 & 0.36 \\ 0.32 & 0.35 & 0.34 \end{bmatrix}, A^{(2)} = \begin{bmatrix} 0.45 & 0.37 & 0.36 \\ 0.36 & 0.44 & 0.37 \\ 0.42 & 0.36 & 0.35 \end{bmatrix}$$

$$B^{(1)} = \begin{bmatrix} 0.01 & 0.02 \\ 0.02 & 0.01 \\ 0.02 & 0.03 \end{bmatrix}, B^{(2)} = \begin{bmatrix} 0.05 & 0.03 \\ 0.04 & 0.03 \\ 0.03 & 0.06 \end{bmatrix}$$

Suppose $u(k) \in [-192.6,192.6]$ is the saturation limited input, $x(k) \in [-21.4,21.4]$ is the saturation limited state $x(0) = [1,2,1.5]^T$ is the initial state of the system. we also decompose the centralized system into two subsystems of which the initial states are $x(0)$, and adopte centralized robust model predictive control (CRMPC), traditional RDMPC (TRDMPC) and collaborative adaptive RDMPC (CARDMPC) to design the controllers for the two subsystems respectively. For all of them, $W_x = I_{3\times3}$, $W_{u1} = W_{u2} = I_{2\times2}$. Similarly, we set the uncertain parameters of the system as (42).

Solving the optimization results of the first three-time intervals, and taking the average value of these results. Finally, we put $\zeta_1 = \zeta_2 = 0.8$ into collaborative adaptive RDMPC.

For the three algorithms, we adopt a one-step iteration method. The control results of the centralized system with different algorithms are compared as follows.

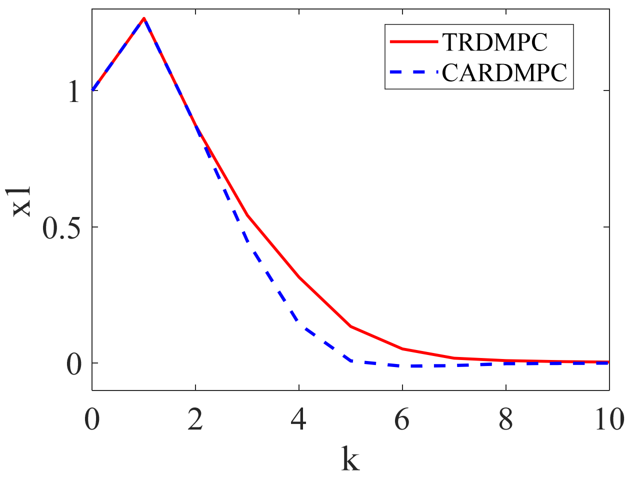

Fig. 7. Dynamic responses of x_1 with different algorithms.

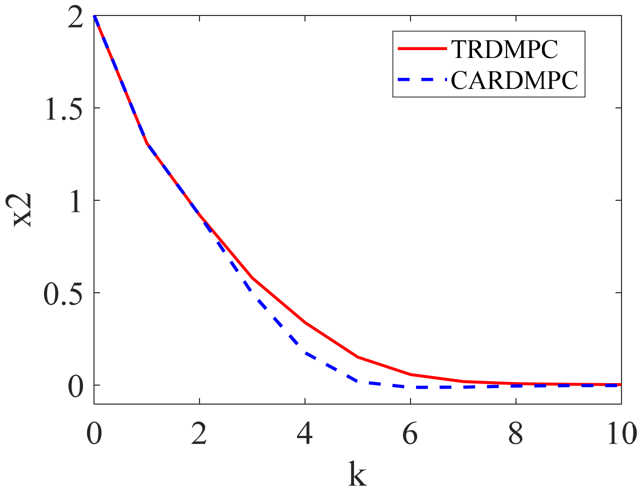

Fig. 8. Dynamic responses of x_2 with different algorithms.

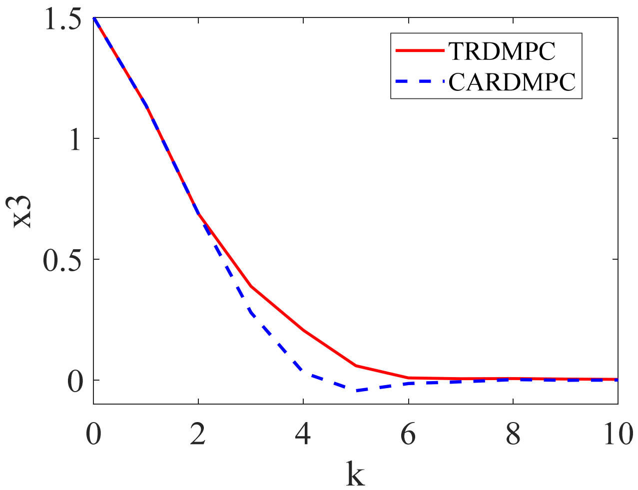

Fig. 9. Dynamic responses of x_3 with different algorithms.

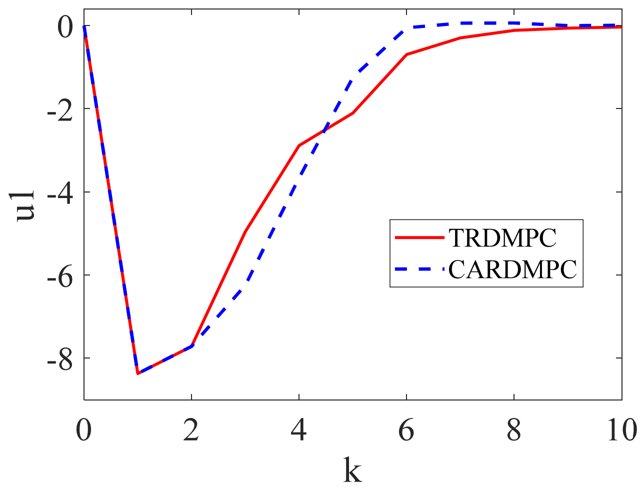

Fig. 10. Dynamic responses of u_1 with different algorithms.

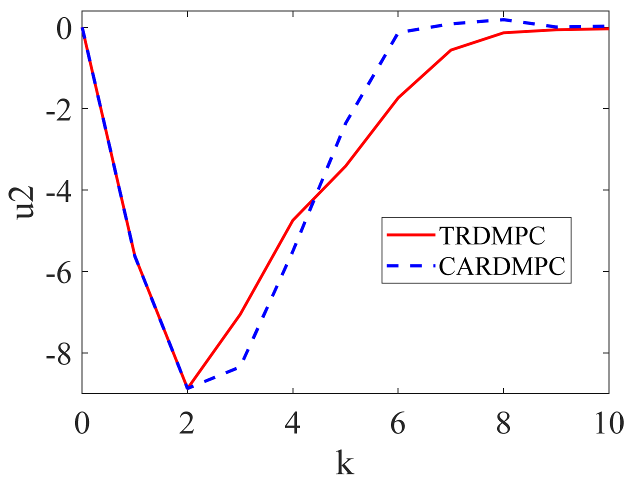

Fig. 11. Dynamic responses of u_2 with different algorithms.

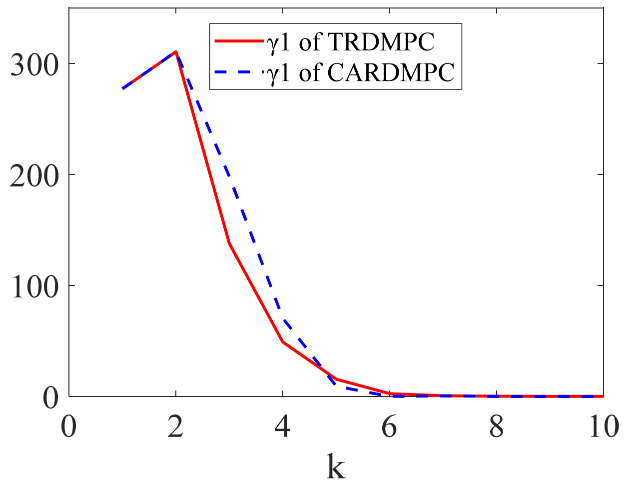

Fig. 12. Dynamic responses of γ_1 with different algorithms.

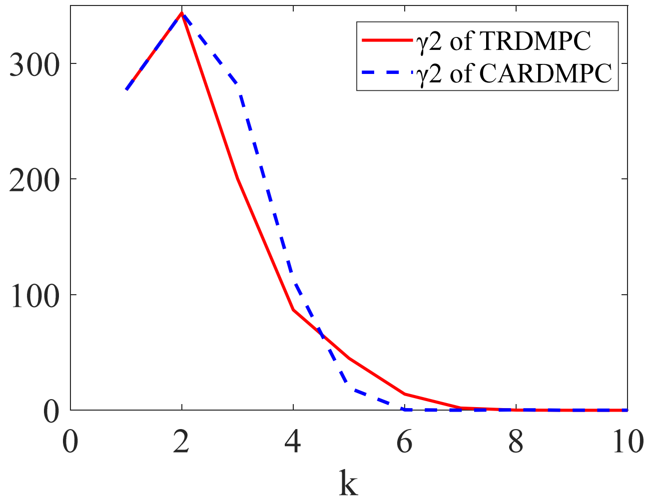

Fig. 13. Dynamic responses of γ_2 with different algorithms.

Table 2. System stabilization time with different algorithms.

State	*Ts* (CRMPC)	*Ts* (RDMPC)	*Ts* (CARDMPC)
x_1	21(27)	7(7)	5(5)
x_2	21(27)	7(7)	5(5)
x_3	21(26)	6(6)	4(6)

Table 2 lists the setting time belonging to the three state variables with different algorithms. The unit of time is an algorithm period. The value outside the brackets is the stability time under the condition of 5% error, and the inside is the stability time under the condition of 2% error. The table shows that the state setting time of the new algorithm reduces by 28.6% to 33.3% compared with that of the traditional cooperative RDMPC, and 76.2% to 80.9% compared with that of CRMPC under the condition of 5% error in this case.

Figure 7 to figure 11 is the curve of the system response with the traditional RDMPC and the collaborative adaptive RDMPC respectively. Our new algorithm can reach stability faster based on them. And both of two algorithms can achieve the same result, which indicates that CARDMPC converges to TRDMPC eventually. Analysing the upper bound of cost index for two algorithms, γ_1 of subsystem 1 and γ_2 of subsystem 2, in figure 12 and 13, we find that the γ of the new algorithm can be reduced to 0 more quickly.

6. CONCLUSIONS

In this paper, we propose an improved distributed model predictive control algorithm with adaptive feedback weight, which can also be called cooperative adaptive model predictive control. The algorithm preserves a set of subsystem's states. At the initial control stage, we use the independent full-dimension state of subsystems to delay the coupling terms of other subsystems, which maintains an effective control input and increases system control speed. It achieves an effect closing to decentralized control. By modifying the weight of hysteresis term about subsystem's states automatically with the control process, the state of the subsystems participating in the feedback is equal to the state of the centralized system, which means restoring the distributed control to make the system reach the performance of RDMPC. We have proved that CARDMPC can achieve better control performance than traditional RDMPC in a one-step iteration by the appropriate weights ζ through two cases. At present, ζ are obtained only by experiment. In the future, the method to determine the appropriate ζ will be studied in combination with a series of parameters of the controlled object.

REFERENCES

Al-Gherwi, W., Budman, H., & Elkamel, A. (2011). A robust distributed model predictive control algorithm. *Journal of Process Control*, *21*(8), 1127–1137. https://doi.org/10.1016/j.jprocont.2011.07.002

Al-Gherwi, W., Budman, H., & Elkamel, A. (2013). A robust distributed model predictive control based on a dual-mode approach. *Computers & Chemical Engineering*, *50*, 130–138.

Bakošov'a, M., & Oravec, J. (2014). Robust Model Predictive Control of Uncertain Linear Systems with Persistent Disturbances and Input Constraints. *American Control Conference*, 5242–5247. https://github.com/oravec-juraj/mup/wiki/

Cao, Y.-Y., & Lin, Z. (2005). Min--max MPC algorithm for LPV systems subject to input saturation. *IEE Proceedings-Control Theory and Applications*, *152*(3), 266–272.

Casavola, A., Giannelli, M., & Mosca, E. (2000). Min--max predictive control strategies for input-saturated polytopic uncertain systems. *Automatica*, *36*(1), 125–133.

Copp, D. A., Vamvoudakis, K. G., & Hespanha, J. P. (2019). Distributed output-feedback model predictive control for multi-agent consensus. *Systems & Control Letters*, *127*, 52–59.

Ding, B., Xi, Y., Cychowski, M. T., & Mahony, T. O.(2007). Improving off-line approach to robust MPC based-on nominal performance cost. *Automatica*, *43*(1), 158–163.

Huang, H., Li, D., Lin, Z., & Xi, Y. (2011). An improved robust model predictive control design in the presence of actuator saturation. *Automatica*, *47*(4), 861–864.

JJE Slotine, W. L. (1991). APPLIED NONLINEAR CONTROL. In *Prentice-Hall, Inc.* (Vol. 8, Issue 6). https://doi.org/10.1088/1741-2552/8/6/060201

Kerrigan, E. C. (2001). *Robust constraint satisfaction: Invariant sets and predictive control*. University of Cambridge.

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

Morato, M. M., Normey-Rico, J. E., & Sename, O. (2020). Model predictive control design for linear parameter varying systems: A survey. *Annual Reviews in Control*.

Necoara, I., Dumitrache, I., & Suykens, J. A. K. (2010). Fast primal-dual projected linear iterations for distributed consensus in constrained convex optimization. *Proceedings of the IEEE Conference on Decision and*

Control, 1366–1371. https://doi.org/10.1109/CDC.2010.5717991

Oravec, J., & Bakošová, M. (2015). Robust Model Predictive Control Based on Nominal System Optimization and Control Input Saturation. *IFAC-PapersOnLine*, *48*(14), 314–319.

Rigatos, G., Siano, P., Selisteanu, D., & Precup, R. E. (2017). Nonlinear Optimal Control of Oxygen and Carbon Dioxide Levels in Blood. *Intelligent Industrial Systems*, *3*(2), 61–75. https://doi.org/10.1007/s40903-016-0060-y

Roman, R. C., Precup, R. E., & Petriu, E. M. (2020). Hybrid data-driven fuzzy active disturbance rejection control for tower crane systems. *European Journal of Control*, *xxxx*. https://doi.org/10.1016/j.ejcon.2020.08.001

Saltık, M. B., Özkan, L., Ludlage, J. H. A., Weiland, S., & Van den Hof, P. M. J. (2018). An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects. *Journal of Process Control*, *61*, 77–102. https://doi.org/10.1016/j.jprocont.2017.10.006

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control--a review. *Journal of Process Control*, *19*(5), 723–731.

Schuurmans, J., & Rossiter, J. A. (2000). Robust predictive control using tight sets of predicted states. *IEE Proceedings-Control Theory and Applications*, *147*(1), 13–18.

Shalmani, R. A., Rahmani, M., & Bigdeli, N. (2020). Nash-based robust distributed model predictive control for large-scale systems. *Journal of Process Control*, *88*, 43–53.

Siljak, D. D. (2011). *Decentralized control of complex systems*. Courier Corporation.

Wan, Z., & Kothare, M. V. (2003). An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica*, *39*(5), 837–846.

Wang, L., & Wen, C. (2008). Distributed model predictive control based on cooperation. *2008 Chinese Control and Decision Conference*, 3526–3531.

Zhang, J., Zhang, L., & Raissi, T. (2020). A linear framework on the distributed model predictive control of positive systems. *Systems & Control Letters*, *138*, 104665.

Zhang, L., Wang, J., & Li, K. (2013). Gain-scheduling Distributed MPC for Polytopic Uncertain Systems subject to Actuator Saturation. *IFAC Proceedings Volumes*, *46*(13), 444–449.

Zhang, Y., & Li, S. (2007). Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes. *Journal of Process Control*, *17*(1), 37–50.