IET Control Theory & Applications

Research Article

ISSN 1751-8644
Received on 9th September 2014
Accepted on 2nd March 2015
doi: 10.1049/iet-cta.2014.1357
www.ietdl.org

Distributed model predictive control for consensus of sampled-data multi-agent systems with double-integrator dynamics

Lifeng Zhou, Shaoyuan Li ✉

Department of Automation, Shanghai JiaoTong University and Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai 200240, People's Republic of China

✉ *E-mail: syli@sjtu.edu.cn*

Abstract: This study proposes a distributed model predictive control (MPC) strategy to achieve consensus of sampled-data multi-agent systems with double-integrator dynamics. On the basis of the error of state between each agent and the centre of its subsystem, a novel distributed MPC strategy (Algorithm 1) is obtained with the exchange of current states only. Then, a reverse iterative algorithm (Algorithm 2) is specially designed for the receding horizon optimisation of sampled-data double-integrator dynamics. Illustrative examples are finally displayed to verify the effectiveness and advantage of the distributed MPC consensus strategy and the impact of sampling period on consensus.

1 Introduction

In recent years, many brilliant achievements have been obtained for the collective behaviours of multi-agent systems (MASs), which have vivid examples in biology, physics, computer science, automatic control and distributed cooperative systems (see, e.g. [1–5]). The effective way to solve problems of MASs is to explore decentralised control strategies arising from local interactions among agents to guarantee that all agents agree upon certain quantities of interest, called consensus [6–10]. Such cooperative consensus strategies have potential impacts on many fields, such as flocking/swarming, sensor network, collaborative robots, underwater vehicles, unmanned aerial vehicles (UAVs), formation control and congestion control in communication networks.

So far, in most of previous consensus schemes, each agent is only available to observe the behaviours of its neighbours currently and then takes a timely decision without consideration of the prediction intelligence of each individual. However, abundant references in the biology literature have presented that almost all living creatures apply the prediction intelligence allowing them to predict the future state of their neighbours and themselves in cooperative work [11, 12]. Examples about this prediction intelligence are vividly reflected in bee swarm formation [11], bio-eyesight systems [12] and so on. Such predictive mechanisms with the capabilities of optimising system energy and control cost have attracted many researchers to apply these alluring features in the study of MASs.

For agents have a discrete-time single- or double-integrator dynamics, Ferrai-Trecate *et al*. [13] proposed decentralised model predictive control (MPC) schemes with control input constraints and shown that such a MAS asymptotically achieves consensus under mild assumptions. For the proof of convergence, the geometric properties of the optimal paths are utilised instead of a Lyapunov optimal value function. Following this line, Zhang *et al*. [14] proposed an MPC strategy to achieve consensus and numerical simulation has verified the performance of consensus. Furthermore, inspired by some efficient schemes in pinning control, they have presented an improved control strategy with faster convergence speed in [15]. Afterwards, for a sampled-data MAS with both fixed and switching network topologies, Zhan and Li in [16] have introduced a distributed MPC weighted-average consensus protocol and proved such a MAS asymptotically reaches the weighted-average consensus. Although such an outstanding work introduces the sampled-data network, it focuses on one-order integrator dynamics which are easier and less common than double-integrator dynamics in the industrial community. Moreover, the cost of communication and computation increases because of the additional exchange of computed inputs and the final computation in this distributed MPC weighted-average consensus protocol. In summary, discrete dynamics have been adopted directly in [13–15], which have lower applicability than sampled-data dynamics where sampling period has potential impacts on the performances of these systems. In [16], only one-order integrator dynamic has been discussed and more cost of communication and computation has been brought.

In short, designing a distributed MPC strategy with optimal energy for consensus of sampled-data MASs with double-integrator dynamics is still a challengeable work. In this paper, our main contributions include: (a) a novel distributed MPC strategy (Algorithm 1), based on the error of state between each agent and the centre of its subsystem, is obtained by merely exchanging the current states only; (b) in order to find the solution of receding horizon optimisation of sampled-data double-integrator dynamics, a reverse iterative algorithm (Algorithm 2) is proposed; and (c) the performance of the distributed MPC consensus strategy and the impact of sampling period on consensus are illustratively displayed.

The rest of this paper is organised as follows. Section 2 introduces some necessary preliminaries. In Section 3, we propose a distributed MPC strategy for sampled-data MAS with double-integrator dynamics and prove the asymptotical convergence of the MAS. Illustrative examples showing the performance of our proposed distributed MPC strategy are presented in Section 4. Finally, conclusions are drawn in Section 5.

2 Preliminaries

We first introduce the following notations and definitions to use throughout this paper. $\mathbf{1} = [1, 1, \ldots, 1]^{\mathrm{T}}$, $\mathbf{0} = [0, 0, \ldots, 0]^{\mathrm{T}}$. Matrix $\boldsymbol{I_m}$ represents the identify matrix with dimension m. The notations $\mathbb{N}^+$, $\mathbb{R}^+$, $\mathbb{N}$ and $\mathbb{R}$ denote the sets of positive integers, positive numbers, natural numbers and real numbers, respectively. The definitions $\mathbb{R}^m$, $\mathbb{R}^{mn}$ and $\mathbb{R}^{n\times n}$ denote the sets of m-dimensional real column vectors, mn-dimensional real column vectors and $n \times n$-dimensional real matrices, respectively. $\|\cdot\|$ indicates the Euclidean norm and $\otimes$ indicates the Kronecker product. The definition

$*(k+t|k)$ denotes the prediction value $*$ at instant $k+t$ based on the currently available information at instant k.

Consider a MAS composed of n agents. It can be abstractly described as a weighted graph denoted by $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathcal{A})$, consisting of a set of vertices $\mathcal{V}=\{1,2,\ldots,n\}$, edges $\mathcal{E}\subseteq\{(i,j): i,j\in\mathcal{V}, j\neq i\}$, and a weighted adjacency matrix $\mathcal{A}=[a_{ij}]_{n\times n}$ with non-negative adjacency elements a_{ij}. If $(i,j)\in\mathcal{E} \iff (j,i)\in\mathcal{E}$, we call the graph $\mathcal{G}$ is undirected and we only consider undirected graph in this paper. If there exist a sequence of edges $(i,k_1),(k_1,k_2),\ldots,(k_{s-1},k_s),(k_s,j)$, $k_t\in\mathcal{V}, t=1,\ldots,s$ between any two vertices $i,j\in\mathcal{V}$, the undirected graph $\mathcal{G}$ is connected. The neighbour set of agent i is denoted by $N_i=\{j\in\mathcal{V}:(i,j)\in\mathcal{E}\}$ and $|N_i|$ denotes the number of the neighbours of agent i. The Laplacian matrix $\boldsymbol{L_n}=[l_{ij}]_{n\times n}$ associated with $\mathcal{G}$ is defined as, $l_{ij}=-a_{ij}, i\neq j$, and $l_{ii}=\sum_{j=1,j\neq i}^{n} a_{ij}$.

Let $\boldsymbol{x_i}\in\mathbb{R}^m$ and $\boldsymbol{x}=\mathrm{col}[\boldsymbol{x_1},\boldsymbol{x_2},\ldots,\boldsymbol{x_n}]\in\mathbb{R}^{mn}$ indicate the state of agent (node) i and the state of the MAS, respectively. The consensus of the MAS is reached if and only if

$$\boldsymbol{x_i}=\boldsymbol{x_j},\quad \forall i,j\in\{1,2,\ldots,n\} \tag{1}$$

3 Consensus via distributed MPC method

Consider that a MAS with undirected topology $\mathcal{G}$ is composed of n agents. Each agent of the MAS, as each vertex of $\mathcal{G}$, follows the same double-integrator dynamic as below

$$\begin{aligned}\dot{\boldsymbol{q}}_{\boldsymbol{i}}(\boldsymbol{t})&=\boldsymbol{p_i}(\boldsymbol{t})\\ \dot{\boldsymbol{p}}_{\boldsymbol{i}}(\boldsymbol{t})&=\boldsymbol{u_i}(\boldsymbol{t}),\quad i=1,2,\ldots,n\end{aligned} \tag{2}$$

where $\boldsymbol{q_i}(\boldsymbol{t})\in\mathbb{R}^m$ and $\boldsymbol{p_i}(\boldsymbol{t})\in\mathbb{R}^m$ denote the position and velocity of the ith agent at time t, respectively. And $\boldsymbol{u_i}(\boldsymbol{t})\in\mathbb{R}^m$ is the corresponding control input. Using the forward-difference approximation, we discretise (2) as

$$\begin{aligned}\boldsymbol{q_i}(\boldsymbol{k}+\boldsymbol{1})&=\boldsymbol{q_i}(\boldsymbol{k})+T\boldsymbol{p_i}(\boldsymbol{k})\\ \boldsymbol{p_i}(\boldsymbol{k}+\boldsymbol{1})&=\boldsymbol{p_i}(\boldsymbol{k})+T\boldsymbol{u_i}(\boldsymbol{k})\end{aligned} \tag{3}$$

where T indicating the sampling period, is a small positive constant. k indicates the discrete-time index. $\boldsymbol{q_i}(\boldsymbol{k})\in\mathbb{R}^m$, $\boldsymbol{p_i}(\boldsymbol{k})\in\mathbb{R}^m$ and $\boldsymbol{u_i}(\boldsymbol{k})\in\mathbb{R}^m$ denote, respectively, the position, velocity and the control input of agent i at $t=kT$. Following (1), we obtain consensus is reached if and only if

$$\begin{aligned}&\lim_{k\to\infty}\|\boldsymbol{q_i}(\boldsymbol{k})-\boldsymbol{q_j}(\boldsymbol{k})\|=0\\ &\lim_{k\to\infty}\|\boldsymbol{p_i}(\boldsymbol{k})-\boldsymbol{p_j}(\boldsymbol{k})\|=0,\quad \forall i,j\in\{1,2,\ldots,n\}\end{aligned} \tag{4}$$

Next, we first introduce the common method to obtain MPC input $\boldsymbol{u_i^*}(\boldsymbol{k})$ for agent i at instant k. Rewrite the discrete-time double-integrator model (3) in a compact way

$$\boldsymbol{x_i}(\boldsymbol{k}+\boldsymbol{1})=\boldsymbol{A}\boldsymbol{x_i}(\boldsymbol{k})+\boldsymbol{B}\boldsymbol{u_i}(\boldsymbol{k}),\quad i=1,2,\ldots,n \tag{5}$$

with $\boldsymbol{x_i}(\boldsymbol{k})=\mathrm{col}[\boldsymbol{q_i}(\boldsymbol{k}),\boldsymbol{p_i}(\boldsymbol{k})]$ and

$$\boldsymbol{A}=I_m\otimes\begin{pmatrix}1 & T\\ 0 & 1\end{pmatrix},\quad \boldsymbol{B}=I_m\otimes\begin{pmatrix}0\\ T\end{pmatrix}$$

Following some basic rules from MPC, we compute the prediction states $\boldsymbol{x_i}(\boldsymbol{k}+\boldsymbol{t}|\boldsymbol{k}), (t=1,2,\ldots,H_{\mathrm{p}})$:

$$\begin{aligned}\boldsymbol{x_i}(\boldsymbol{k}+\boldsymbol{t}|\boldsymbol{k})&=\boldsymbol{A}\boldsymbol{x_i}(\boldsymbol{k}+\boldsymbol{t}-\boldsymbol{1}|\boldsymbol{k})\\ &\quad+\boldsymbol{B}\boldsymbol{u_i}(\boldsymbol{k}+\boldsymbol{t}-\boldsymbol{1}|\boldsymbol{k}),\quad t<H_u\end{aligned} \tag{6}$$

$$\begin{aligned}\boldsymbol{x_i}(\boldsymbol{k}+\boldsymbol{t}|\boldsymbol{k})&=\boldsymbol{A}\boldsymbol{x_i}(\boldsymbol{k}+\boldsymbol{t}-\boldsymbol{1}|\boldsymbol{k})\\ &\quad+\boldsymbol{B}\boldsymbol{u_i}(\boldsymbol{k}+\boldsymbol{H_u}-\boldsymbol{1}|\boldsymbol{k}),\quad t\geq H_u\end{aligned} \tag{7}$$

with $\boldsymbol{X_i}(\boldsymbol{k})=\mathrm{col}[\boldsymbol{x_i}(\boldsymbol{k}+\boldsymbol{1}|\boldsymbol{k}),\boldsymbol{x_i}(\boldsymbol{k}+\boldsymbol{2}|\boldsymbol{k}),\ldots,\boldsymbol{x_i}(\boldsymbol{k}+\boldsymbol{H}_{\mathrm{p}}|\boldsymbol{k})]$, $\boldsymbol{U_i}(\boldsymbol{k})=\mathrm{col}[\boldsymbol{u_i}(\boldsymbol{k}|\boldsymbol{k})$ and $\boldsymbol{u_i}(\boldsymbol{k}+\boldsymbol{1}|\boldsymbol{k}),\ldots,\boldsymbol{u_i}(\boldsymbol{k}+\boldsymbol{H_u}-\boldsymbol{1}|\boldsymbol{k})]$. Here, integers H_{p}, H_u denote prediction horizon and control horizon, respectively, with $1\leq H_u\leq H_{\mathrm{p}}$. Thus (6) and (7) can be rewritten in a compact way as

$$\boldsymbol{X_i}(\boldsymbol{k})=\boldsymbol{P_x}\boldsymbol{x_i}(\boldsymbol{k})+\boldsymbol{P_u}\boldsymbol{U_i}(\boldsymbol{k})$$

with $\boldsymbol{P_x}=\mathrm{col}[\boldsymbol{A},\boldsymbol{A^2},\ldots,\boldsymbol{A}^{\boldsymbol{H}_{\mathrm{p}}}]$ and

$$\boldsymbol{P_u}=\begin{bmatrix}\boldsymbol{B} & & & \\ \boldsymbol{AB} & \ddots & & \\ \vdots & \vdots & \boldsymbol{B} & \\ \boldsymbol{A}^{\boldsymbol{H_u}-1}\boldsymbol{B} & \cdots & \boldsymbol{AB} & \boldsymbol{B}\\ \vdots & \cdots & \vdots & \vdots\\ \boldsymbol{A}^{\boldsymbol{H}_{\mathrm{p}}-1}\boldsymbol{B} & \cdots & \boldsymbol{A}^{\boldsymbol{H}_{\mathrm{p}}-\boldsymbol{H_u}+1}\boldsymbol{B} & \sum_{i=0}^{H_{\mathrm{p}}-H_u}\boldsymbol{A^i}\boldsymbol{B}\end{bmatrix}$$

Then, we definite the MPC cost function for agent i as

$$J_i(k)=\|\boldsymbol{P_x}\boldsymbol{x_i}(\boldsymbol{k})+\boldsymbol{P_u}\boldsymbol{U_i}(\boldsymbol{k})-\boldsymbol{r_i}(\boldsymbol{k})\|^2_{\boldsymbol{\Theta}_i}+\|\boldsymbol{U_i}(\boldsymbol{k})\|^2_{\boldsymbol{\Lambda}_i} \tag{8}$$

where $\boldsymbol{\Theta_i}$ and $\boldsymbol{\Lambda_i}$ represent associated state-weighted matrix with an appropriate dimension and control-weighted matrix with an appropriate dimension, respectively. $\boldsymbol{r_i}(\boldsymbol{k})$ denotes the reference signal. Optimising the MPC cost function by using $[\{\partial J_i(k)\}/\{\partial \boldsymbol{U_i}(\boldsymbol{k})\}]$, we obtain the optimal input sequence: $\boldsymbol{U_i^*}(\boldsymbol{k})=-(\boldsymbol{P_u^{\mathrm{T}}}\boldsymbol{\Theta_i}\boldsymbol{P_u}+\boldsymbol{\Lambda_i})^{-1}\boldsymbol{P_u^{\mathrm{T}}}\boldsymbol{\Theta_i}(\boldsymbol{P_x}\boldsymbol{x_i}(\boldsymbol{k})-\boldsymbol{r_i}(\boldsymbol{k}))$. Usually, we use the first entry of $\boldsymbol{U_i^*}$ as the actual control signal, that is, $\boldsymbol{u_i^*}(\boldsymbol{k})=-[\underbrace{1,1,\ldots,1}_{m},0,0,\ldots,0]\times\boldsymbol{U_i^*}(\boldsymbol{k})$. Denoting $\boldsymbol{\varphi_i}=[\underbrace{1,1,\ldots,1}_{m},0,0,\ldots,0]\times(\boldsymbol{P_u^{\mathrm{T}}}\boldsymbol{\Theta_i}\boldsymbol{P_u}+\boldsymbol{\Lambda_i})^{-1}\boldsymbol{P_u^{\mathrm{T}}}\boldsymbol{\Theta_i}$, we have the optimal MPC input

$$\boldsymbol{u_i^*}(\boldsymbol{k})=-\boldsymbol{\varphi_i}(\boldsymbol{P_x}\boldsymbol{x_i}(\boldsymbol{k})-\boldsymbol{r_i}(\boldsymbol{k})) \tag{9}$$

where the vector $\boldsymbol{\varphi_i}$ is independent of $\boldsymbol{x_i}(\boldsymbol{k})$ and thus can be calculated offline.

Remark 1: By optimising the MPC cost function (8) associated with prediction and control horizons, we have obtained the optimal MPC input. Inspired by this rolling or receding horizon optimisation strategy, we will design the distributed MPC input to better adapt to the MAS (3) afterwards.

Note that each agent can only get information from its neighbours, we decompose the whole network consisting of n agents into n subsystems accordingly. For agent i, its subsystem consists of itself and its neighbours, indicated by $\delta_i=\{n_1^i,n_2^i,\ldots,n^i_{|N_i|+1}\}$, $n_1^i<n_2^i<\cdots<n^i_{|N_i|+1}$, $n_j^i\in N_i\cup i$, $j=1,2,\ldots,|N_i|+1$. Considering the interactions among agents in each subsystem, we define the distributed MPC cost function for agent i with position and velocity state separated in the following form

$$\begin{aligned}J_i(k)&=\alpha\sum_{t=1}^{H_{\mathrm{p}}}\|(\boldsymbol{q_i}(\boldsymbol{k}+\boldsymbol{t}|\boldsymbol{k})-\boldsymbol{r_i^q}(\boldsymbol{k}+\boldsymbol{t}|\boldsymbol{k})\|^2\\ &\quad+\beta\sum_{t=1}^{H_{\mathrm{p}}}\|(\boldsymbol{p_i}(\boldsymbol{k}+\boldsymbol{t}|\boldsymbol{k})-\boldsymbol{r_i^p}(\boldsymbol{k}+\boldsymbol{t}|\boldsymbol{k})\|^2\\ &\quad+\lambda\sum_{t=0}^{H_u-1}\|(\boldsymbol{u_i}(\boldsymbol{k}+\boldsymbol{t}|\boldsymbol{k})\|^2\end{aligned} \tag{10}$$

with $\alpha,\beta,\lambda\in\mathbb{R}^+$, $t\in\mathbb{N}$. $\boldsymbol{r_i^q}(\boldsymbol{k}+\boldsymbol{t}|\boldsymbol{k})\in\mathbb{R}^m$ and $\boldsymbol{r_i^p}(\boldsymbol{k}+\boldsymbol{t}|\boldsymbol{k})\in\mathbb{R}^m$ denoting, respectively, the position and velocity vector references, are defined as below

$$r_i^q(k+t|k) = \frac{1}{|N_i|+1}\sum_{j\in\delta_i} q_j(k) + t\times T\times r_i^p(k+t|k)$$
$$r_i^p(k+t|k) = \frac{1}{|N_i|+1}\sum_{j\in\delta_i} p_j(k) \quad \forall i,j\in\mathcal{V},$$
$$t = 0,1,2,\dots,H_{\rm p} \tag{11}$$

In fact, the references $r_i^q(k+t|k)$ and $r_i^p(k+t|k)$ describe the centre of the state of i' subsystem. Moreover, agent i, resolves its actual control input $u_i(k)$ following the distributed MPC consensus algorithm described as below:

Algorithm 1: Step 1: Agent i receives feasible states $q_j(k)$, $p_j(k)$ for agent j which has not yet calculated its optimal input or optimal states $q_j^*(k)$, $p_j^*(k)$ for agent j which already has calculated its optimal input from its neighbours at instant k. Then agent i computes its references $r_i^q(k)$ and $r_i^p(k)$ at instant k.

Step 2: Agent i optimises the distributed MPC cost function (10), and denotes its optimal control input by $u_i^*(k)$ and the corresponding states by $q_i^*(k)$, $p_i^*(k)$.

Step 3: Agent i sends its optimal states $q_i^*(k)$, $p_i^*(k)$ to its neighbours, facilitating them to calculate the references and solve the optimisation problems of their own.

Step 4: Agent i applies $u_i(k) := u_i^*(k)$ as its actual control input at instant k.

Remark 2: Compared with previous distributed MPC consensus in [16], where each agent receives the information of its neighbours directly as well as the information of its neighbours' neighbours indirectly, the distributed protocol proposed in this paper, similar to the protocols in [13, 15], requires the exchange of current states only, which brings less cost of communication and computation.

Before giving the optimal distributed MPC control input, we need some necessary lemmas below.

Lemma 1: Given a polynomial with respect to parameters e, f, d in the following form

$$\sum_{i=1}^{h}(\alpha(e+iTf)^2+\beta(f+iTd)^2)+\lambda d^2 \tag{12}$$

with scalars $\alpha,\beta,\lambda\in\mathbb{R}^+$, $h\in\mathbb{N}^+$. Moreover, T is a small positive constant. Then, by polynomial transform, polynomial (12) can be written as

$$b(d+ce+\check{c}f)^2+\breve{\alpha}e^2+\breve{\beta}f^2+\breve{\gamma}ef \tag{13}$$

where $b>0$, $c=0$, $\check{c}>0$, $\breve{\alpha}>0$, $\breve{\beta}>0$ and $\breve{\gamma}>0$.

Proof: As for polynomial equation

$$\sum_{i=1}^{h}(\alpha(e+iTf)^2+\beta(f+iTd)^2)+\lambda d^2$$
$$= b(d+ce+\check{c}f)^2+\breve{\alpha}e^2+\breve{\beta}f^2+\breve{\gamma}ef$$

Using coefficient comparison to calculate the related scalars, we obtain $b=(\beta T^2\sum_{i=1}^h i^2+\lambda)>0$, $c=0$, $\check{c}=\beta T/b\sum_{i=1}^h i>0$,

$\breve{\alpha}=\alpha>0$

$$\begin{aligned}\breve{\beta} &= \alpha T^2\sum_{i=1}^h i^2+\beta-b\check{c}^2\\ &= \alpha T^2\sum_{i=1}^h i^2+\beta-\frac{(\beta T\sum_{i=1}^h i)^2}{b}\\ &= \beta+\frac{\alpha\lambda T^2\sum_{i=1}^h i^2}{b}\\ &\quad+\frac{\alpha\beta T^4(\sum_{i=1}^h i^2)^2-\beta^2T^2(\sum_{i=1}^h i)^2}{b}\end{aligned}$$

Considering T is a small positive constant, we obtain $\breve{\beta}\simeq\beta>0$ and $\breve{\gamma}=2\alpha T\sum_{i=1}^h i>0$. Thus the proof is simply completed. □

Lemma 2: Given a polynomial with respect to parameters e, f, d in the following form

$$\alpha(e+Tf)^2+\beta(f+Td)^2+\lambda d^2+\alpha_*(e+Tf)^2$$
$$+\beta_*(f+Td)^2+\gamma_*(e+Tf)(f+Td) \tag{14}$$

with scalars α, β, λ, α_*, β_*, $\gamma_*\in\mathbb{R}^+$. Moreover, T is a small positive constant. Then, by polynomial transform, polynomial (14) can be written as

$$b(d+ce+\check{c}f)^2+\breve{\alpha}_*e^2+\breve{\beta}_*f^2+\breve{\gamma}_*ef \tag{15}$$

where $b>0$, $c>0$, $\check{c}>0$, $\breve{\alpha}_*>0$, $\breve{\beta}_*>0$, $\breve{\gamma}_*>0$.

Proof: As for polynomial equation

$$\alpha(e+Tf)^2+\beta(f+Td)^2+\lambda d^2$$
$$+\alpha_*(e+Tf)^2+\beta_*(f+Td)^2+\gamma_*(e+Tf)(f+Td)$$
$$= b(d+ce+\check{c}f)^2+\breve{\alpha}_*e^2+\breve{\beta}_*f^2+\breve{\gamma}_*ef$$

Using coefficient comparison to calculate the related scalars, we obtain $b=(T^2(\beta+\beta_*)+\lambda)>0$, $c=T\gamma_*/2b>0$, $\check{c}=(2T(\beta+\beta_*)+T^2\gamma_*)/2b>0$

$$\begin{aligned}\breve{\alpha}_* &= \alpha+\alpha_*-bc^2\\ &= \alpha+\alpha_*-\frac{T^2\gamma_*^2}{4b}\end{aligned}$$

Considering T is a small positive constant, we obtain $\breve{\alpha}_*\simeq\alpha+\alpha_*>0$. Similarly, $\breve{\beta}_*=(\alpha+\alpha_*+T^2(\beta+\beta_*)+T\gamma_*-b\check{c}^2)>0$ and $\breve{\gamma}_*=(2T(\alpha+\alpha_*)+\gamma_*-2bc\check{c})>0$. Thus the proof is simply completed. □

Lemma 3: Given an equation with respect to parameters e_t, f_t, $u_t\in\mathbb{R}$:

$$J=\alpha\sum_{t=1}^{H_{\rm p}}e_t^2+\beta\sum_{t=1}^{H_{\rm p}}f_t^2+\lambda\sum_{t=0}^{H_u-1}u_t^2 \tag{16}$$

with $\alpha,\beta,\lambda\in\mathbb{R}^+$, H_u, $H_{\rm p}\in\mathbb{N}^+$, $t\in N$ and $1\le H_u\le H_{\rm p}$. If there exists a small positive constant T and e_t, f_t, $u_t\in\mathbb{R}$ fulfil:

(1) $e_t=e_{t-1}+Tf_{t-1}$, $t=1,2,\dots,H_{\rm p}$;
(2) $f_t=f_{t-1}+Tu_{t-1}$, $t=1,2,\dots,H_{\rm p}$;
(3) $u_t=u_{H_u-1}$, $H_u\le t\le H_{\rm p}$;
(4) $e_0,f_0\in\mathbb{R}$.

then by optimising J, u_t has the concise form

$$u_t=-c_te_t-\check{c}_tf_t,\quad t=0,1,\dots,H_u-1 \tag{17}$$

where c_t, $\check{c}_t\in\mathbb{R}^+$ and their values depend on $H_{\rm p}$, H_u and the small positive constant T.

Proof: On the basis of conditions (1)–(4) above, it is obvious that

$$e_{t+1} = e_t + Tf_t \tag{18}$$
$$f_{t+1} = f_t + Tu_t \tag{19}$$

Considering $H_u \le t \le H_{\rm p}$, we also have

$$e_{H_u+\tau-1} = e_{H_u-1} + \tau T f_{H_u-1} \tag{20}$$
$$f_{H_u+\tau-1} = f_{H_u-1} + \tau T u_{H_u-1} \tag{21}$$

with $\tau = t - H_u + 1$.

Then, a reverse iterative algorithm described as below is utilised to accomplish the proof.

Algorithm 2: Step 1: Rewrite (16) as

$$J = J_1 + J_2 + \cdots + J_{H_u-1} + J_{H_u} \tag{22}$$

where

$$J_t = \alpha e_t^2 + \beta f_t^2 + \lambda u_{t-1}^2, \quad t = 1, 2, \ldots, H_u - 1 \tag{23}$$
$$J_{H_u} = \alpha \sum_{\tau=1}^{h} e_{H_u+\tau-1}^2 + \beta \sum_{\tau=1}^{h} f_{H_u+\tau-1}^2 + \lambda u_{H_u-1}^2 \tag{24}$$

with $h = H_{\rm p} - H_u + 1$.

Step 2: By Lemma 1, substitute (20) and (21) into (24) to obtain that

$$J_{H_u} = b_{H_u}(u_{H_u-1} + c_{H_u-1}e_{H_u-1} + \breve{c}_{H_u-1}f_{H_u-1})^2 + \breve{\alpha}_{H_u-1}e_{H_u-1}^2 + \breve{\beta}_{H_u-1}f_{H_u-1}^2 + \breve{\gamma}_{H_u-1}e_{H_u-1}f_{H_u-1}$$

with $b_{H_u} = (\beta T^2 \sum_{\tau=1}^{h} \tau^2 + \lambda) > 0$, $c_{H_u-1} = 0$, $\breve{c}_{H_u-1} = \beta T / b_{H_u} \sum_{\tau=1}^{h} \tau > 0$, $\breve{\alpha}_{H_u-1} = \alpha > 0$, $\breve{\beta}_{H_u-1} = (\alpha T^2 \sum_{\tau=1}^{h} \tau^2 + \beta - b_{H_u} \breve{c}_{H_u-1}^2) > 0$ and $\breve{\gamma}_{H_u-1} = 2\alpha T \sum_{\tau=1}^{h} \tau > 0$.

Step 3: By Lemma 2, substitute (18) and (19) into (23) and the equation $\breve{\alpha} e_t^2 + \breve{\beta} f_t^2 + \breve{\gamma}_t e_t f_t$ to obtain that

$$J_t + \breve{\alpha} e_t^2 + \breve{\beta} f_t^2 + \breve{\gamma}_t e_t f_t = b_t(u_{t-1} + c_{t-1}e_{t-1} + \breve{c}_{t-1}f_{t-1})^2 + \breve{\alpha}_{t-1}e_{t-1}^2 + \breve{\beta}_{t-1}f_{t-1}^2 + \breve{\gamma}_{t-1}e_{t-1}f_{t-1}$$

with $b_t = (T^2(\beta + \breve{\beta}_t) + \lambda) > 0$, $c_{t-1} = T\breve{\gamma}_t / 2b_t > 0$, $\breve{c}_{t-1} = (2T(\beta + \breve{\beta}_t) + T^2 \breve{\gamma}_t)/2b_t > 0$, $\breve{\alpha}_{t-1} = (\alpha + \breve{\alpha}_t - b_t c_{t-1}^2) > 0$, $\breve{\beta}_{t-1} = (\alpha + \breve{\alpha}_t + T^2(\beta + \breve{\beta}_t) + T\breve{\gamma}_t - b_t \breve{c}_{t-1}^2) > 0$ and $\breve{\gamma}_{t-1} = (2T(\alpha + \breve{\alpha}_t) + \breve{\gamma}_t - 2b_t c_{t-1}\breve{c}_{t-1}) > 0$.

Moreover, implement this process from $t = H_u - 1$ to $t = 1$.

Thus though these three steps in Algorithm 2, (16) can be rewritten as

$$J = \sum_{t=1}^{H_u} b_t(u_{t-1} + c_{t-1}e_{t-1} + \breve{c}_{t-1}f_{t-1})^2 + C \tag{25}$$

where $C = \breve{\alpha}_0 e_0^2 + \breve{\beta}_0 f_0^2 + \breve{\gamma}_0 e_0 f_0 \in \mathbb{R}$, and it is calculated by Step 3 in Algorithm 2. Thereby, it is clear that J attains its minimum when $u_t = -c_t e_t - \breve{c}_t f_t$, $t = 0, 1, \ldots, H_u - 1$, with scalars $c_t, \breve{c}_t \in \mathbb{R}^+$. Usually, we use the first step of u_t as actual control signal, that is, $u_0 = -c_0 e_0 - \breve{c}_0 f_0$. The proof is thus completed. □

Remark 3: In fact, we use reverse iterative algorithm (Algorithm 2) to obtain (25) starting from J_{H_u} and ending with J_1. In this iteration process, we can get those positive constant scalars $b_{t+1}, c_t, \breve{c}_t$ $(t = 0, 1, \ldots, H_u - 1)$ one by one. As the scalars c_t and $\breve{c}_t$ are obtained by the optimisation process, so the associated energy cost of the MAS are minimal, which will be illustratively verified in Section 4.

Lemma 4: By optimising the associated distributed MPC cost function $J_i(k)$, defined as (10), the distributed MPC input $\boldsymbol{u_i}(\boldsymbol{k})$ can be equivalently written as the following from

$$\boldsymbol{u_i}(\boldsymbol{k}) = -c_{i0}\Big(\boldsymbol{q_i}(\boldsymbol{k}) - \frac{1}{|N_i|+1}\sum_{j\in\delta_i}\boldsymbol{q_j}(\boldsymbol{k})\Big) - \breve{c}_{i0}\Big(\boldsymbol{p_i}(\boldsymbol{k}) - \frac{1}{|N_i|+1}\sum_{j\in\delta_i}\boldsymbol{p_j}(\boldsymbol{k})\Big) \tag{26}$$

where $c_{i0}, \breve{c}_{i0} \in \mathbb{R}^+$ are dependent on $H_{\rm p}$, H_u and the small positive constant T.

Proof: On the basis of the definition of references in (11) and the basic rules (6), (7) from MPC, it is clear that

$$\begin{aligned} &\boldsymbol{q_i}(\boldsymbol{k+t}|\boldsymbol{k}) - \boldsymbol{r_i^q}(\boldsymbol{k+t}|\boldsymbol{k}) \\ &\quad = \boldsymbol{q_i}(\boldsymbol{k+t-1}|\boldsymbol{k}) - \boldsymbol{r_i^q}(\boldsymbol{k+t-1}|\boldsymbol{k}) \\ &\qquad + T(\boldsymbol{p_i}(\boldsymbol{k+t-1}|\boldsymbol{k}) - \boldsymbol{r_i^p}(\boldsymbol{k+t-1}|\boldsymbol{k})), \\ &\qquad 1 \le t \le H_{\rm p} \end{aligned}$$

and

$$\begin{aligned} &\boldsymbol{p_i}(\boldsymbol{k+t}|\boldsymbol{k}) - \boldsymbol{r_i^p}(\boldsymbol{k+t}|\boldsymbol{k}) \\ &\quad = \boldsymbol{p_i}(\boldsymbol{k+t-1}|\boldsymbol{k}) - \boldsymbol{r_i^p}(\boldsymbol{k+t-1}|\boldsymbol{k}) \\ &\qquad + T\boldsymbol{u_i}(\boldsymbol{k+t-1}|\boldsymbol{k}), \ 1 \le t < H_u \\ &\boldsymbol{p_i}(\boldsymbol{k+t}|\boldsymbol{k}) - \boldsymbol{r_i^p}(\boldsymbol{k+t}|\boldsymbol{k}) \\ &\quad = \boldsymbol{p_i}(\boldsymbol{k+t-1}|\boldsymbol{k}) - \boldsymbol{r_i^p}(\boldsymbol{k+t-1}|\boldsymbol{k}) \\ &\qquad + T\boldsymbol{u_i}(\boldsymbol{k+H_u-1}|\boldsymbol{k}), \ H_u \le t \le H_{\rm p} \end{aligned}$$

Note that, as for a given vector, each element of which is independent. Thus the relations among the corresponding elements of vectors $\boldsymbol{q_i}(\boldsymbol{k+t}|\boldsymbol{k}) - \boldsymbol{r_i^q}(\boldsymbol{k+t}|\boldsymbol{k})$, $\boldsymbol{p_i}(\boldsymbol{k+t}|\boldsymbol{k}) - \boldsymbol{r_i^p}(\boldsymbol{k+t}|\boldsymbol{k})$ and $\boldsymbol{u_i}(\boldsymbol{k+t}|\boldsymbol{k})$ satisfy conditions (1)–(3) in Lemma 3. The elements of vectors $\boldsymbol{q_i}(\boldsymbol{k}|\boldsymbol{k}) - \boldsymbol{r_i^q}(\boldsymbol{k}|\boldsymbol{k})$ and $\boldsymbol{p_i}(\boldsymbol{k}|\boldsymbol{k}) - \boldsymbol{r_j^p}(\boldsymbol{k}|\boldsymbol{k})$ satisfy condition (4). Optimising the associated distributed MPC cost function $J_i(k)$ and following the reverse iterative algorithm (Algorithm 2) in Lemma 3, we obtain

$$\boldsymbol{u_i}(\boldsymbol{k+t}|\boldsymbol{k}) = -c_{it}(\boldsymbol{q_i}(\boldsymbol{k+t}|\boldsymbol{k}) - \boldsymbol{r_i^q}(\boldsymbol{k+t}|\boldsymbol{k})) - \breve{c}_{it}(\boldsymbol{p_i}(\boldsymbol{k+t}|\boldsymbol{k}) - \boldsymbol{r_i^p}(\boldsymbol{k+t}|\boldsymbol{k}))$$

where c_{it}, $\breve{c}_{it}(0 \le t \le H_u - 1)$, can be calculated in detail in Algorithm 2. Usually, we use the first step of $\boldsymbol{u_i}(\boldsymbol{k+t}|\boldsymbol{k})$ as the actual control signal

$$\boldsymbol{u_i}(\boldsymbol{k}) = -c_{i0}\Big(\boldsymbol{q_i}(\boldsymbol{k}) - \frac{1}{|N_i|+1}\sum_{j\in\delta_i}\boldsymbol{q_j}(\boldsymbol{k})\Big) - \breve{c}_{i0}\Big(\boldsymbol{p_i}(\boldsymbol{k}) - \frac{1}{|N_i|+1}\sum_{j\in\delta_i}\boldsymbol{p_j}(\boldsymbol{k})\Big)$$

where $c_{i0}, \breve{c}_{i0} \in \mathbb{R}^+$ are dependent on $H_{\rm p}$, H_u and the small positive constant T. The proof is thus completed. □

By far, as for each subsystem, we have obtained the distributed MPC input $\boldsymbol{u_i}(\boldsymbol{k})$. Next, we first add some necessary assumption to guarantee the global consensus of the MAS (3) with the distributed MPC input $\boldsymbol{u_i}(\boldsymbol{k})$.

Assumption 1: Assume that the undirected graph $\mathcal{G}$ used in this paper is connected. Moreover, if $(i,j) \in \mathcal{E}$, $a_{ij} = 1$, else $a_{ij} = 0$.

Then, we give the main results as below.

Theorem 1: Consider the MAS (3) with the network topology $\mathcal{G}$, the consensus is achieved asymptotically if the distributed MPC input $\boldsymbol{u_i}(\boldsymbol{k})$ is applied.

Proof: Denote the position and velocity of the centre of mass (COM) of all agents in the group as $\bar{\boldsymbol{q}}(\boldsymbol{k}) = (1/n)\sum_{i=1}^{n} \boldsymbol{q_i}(\boldsymbol{k})$ and $\bar{\boldsymbol{p}}(\boldsymbol{k}) = (1/n)\sum_{i=1}^{n} \boldsymbol{p_i}(\boldsymbol{k})$, respectively. Moreover, define the position difference and the velocity difference between agent i and the COM as $\tilde{\boldsymbol{q}}_i(\boldsymbol{k}) = \boldsymbol{q_i}(\boldsymbol{k}) - \bar{\boldsymbol{q}}(\boldsymbol{k})$ and $\tilde{\boldsymbol{p}}_i(\boldsymbol{k}) = \boldsymbol{p_i}(\boldsymbol{k}) - \bar{\boldsymbol{p}}(\boldsymbol{k})$, respectively. Then, substituting (26) into the MAS(3), we obtain

$$
\begin{aligned}
\tilde{\boldsymbol{q}}_i(\boldsymbol{k}+\boldsymbol{1}) &= \tilde{\boldsymbol{q}}_i(\boldsymbol{k}) + T\tilde{\boldsymbol{p}}_i(\boldsymbol{k}) \\
\tilde{\boldsymbol{p}}_i(\boldsymbol{k}+\boldsymbol{1}) &= \tilde{\boldsymbol{p}}_i(\boldsymbol{k}) - Tc_{i0}\Big(\tilde{\boldsymbol{q}}_i(\boldsymbol{k}) - \frac{1}{|N_i|+1}\sum_{j\in\delta_i}\tilde{\boldsymbol{q}}_j(\boldsymbol{k})\Big) \\
&\quad - T\check{c}_{i0}\Big(\tilde{\boldsymbol{p}}_i(\boldsymbol{k}) - \frac{1}{|N_i|+1}\sum_{j\in\delta_i}\tilde{\boldsymbol{p}}_j(\boldsymbol{k})\Big)
\end{aligned}
$$

For notational convenience, we also define

$$
\begin{aligned}
\tilde{\boldsymbol{Q}}(\boldsymbol{k}) &= \mathrm{col}[\tilde{\boldsymbol{q}}_1(\boldsymbol{k}), \tilde{\boldsymbol{q}}_2(\boldsymbol{k}), \ldots, \tilde{\boldsymbol{q}}_n(\boldsymbol{k})] \\
\tilde{\boldsymbol{P}}(\boldsymbol{k}) &= \mathrm{col}[\tilde{\boldsymbol{p}}_1(\boldsymbol{k}), \tilde{\boldsymbol{p}}_2(\boldsymbol{k}), \ldots, \tilde{\boldsymbol{p}}_n(\boldsymbol{k})], \\
\boldsymbol{C_0} &= \mathrm{diag}\left(\frac{c_{10}}{|N_1|+1}, \frac{c_{20}}{|N_2|+1}, \ldots, \frac{c_{n0}}{|N_n|+1}\right) \\
\check{\boldsymbol{C}}_0 &= \mathrm{diag}\left(\frac{\check{c}_{10}}{|N_1|+1}, \frac{\check{c}_{20}}{|N_2|+1}, \ldots, \frac{\check{c}_{n0}}{|N_n|+1}\right)
\end{aligned}
$$

Following Lemma 3, we obtain $c_{i0},\ \check{c}_{i0} \in \mathbb{R}^+,\ \ i = 1, 2, \ldots, n$. Therefore $\boldsymbol{C_0}$ and $\check{\boldsymbol{C}}_0$ are positive definite diagonal matrices. Consider the Lyapunov function candidate

$$
\begin{aligned}
\tilde{V}(k) &= \tilde{V}(\tilde{\boldsymbol{Q}}(\boldsymbol{k}), \tilde{\boldsymbol{P}}(\boldsymbol{k})) \\
&= \frac{1}{2}\tilde{\boldsymbol{Q}}(\boldsymbol{k})^{\mathrm{T}}(\boldsymbol{I_m} \otimes (\boldsymbol{C_0}\boldsymbol{L_n}))^{\mathrm{T}}\tilde{\boldsymbol{Q}}(\boldsymbol{k}) + \frac{1}{2}\tilde{\boldsymbol{P}}(\boldsymbol{k})^{\mathrm{T}}\tilde{\boldsymbol{P}}(\boldsymbol{k})
\end{aligned} \tag{27}
$$

with $\boldsymbol{L_n} \in \mathbb{R}^{n\times n}$ as the Laplacian matrix associated with undirected graph $\mathcal{G}$. By Assumption 1, the matrix $\boldsymbol{L_n}$ has exactly one zero eigenvalue with an associated eigenvector $\mathbf{1}$ and all other eigenvalues with positive real parts [17]. Therefore the Lyapunov function candidate is positive definite with respect to $\tilde{\boldsymbol{Q}}(\boldsymbol{k})$ and $\tilde{\boldsymbol{P}}(\boldsymbol{k})$.

The derivative of $\tilde{V}(k)$ along the trajectories of the agents is given by

$$
\begin{aligned}
&\tilde{V}(\tilde{\boldsymbol{Q}}(\boldsymbol{k}+\boldsymbol{1}), \tilde{\boldsymbol{P}}(\boldsymbol{k}+\boldsymbol{1})) - \tilde{V}(\tilde{\boldsymbol{Q}}(\boldsymbol{k}), \tilde{\boldsymbol{P}}(\boldsymbol{k})) \\
&= \tilde{\boldsymbol{Q}}(\boldsymbol{k})^{\mathrm{T}}(\boldsymbol{I_m} \otimes (\boldsymbol{C_0}\boldsymbol{L_n}))^{\mathrm{T}}(\tilde{\boldsymbol{Q}}(\boldsymbol{k}+\boldsymbol{1}) - \tilde{\boldsymbol{Q}}(\boldsymbol{k})) \\
&\quad + \tilde{\boldsymbol{P}}(\boldsymbol{k})^{\mathrm{T}}(\tilde{\boldsymbol{P}}(\boldsymbol{k}+\boldsymbol{1}) - \tilde{\boldsymbol{P}}(\boldsymbol{k})) \\
&= T\tilde{\boldsymbol{P}}(\boldsymbol{k})^{\mathrm{T}}(\boldsymbol{I_m} \otimes (\boldsymbol{C_0}\boldsymbol{L_n}))\tilde{\boldsymbol{Q}}(\boldsymbol{k}) \\
&\quad + \tilde{\boldsymbol{P}}(\boldsymbol{k})^{\mathrm{T}}(-T(\boldsymbol{I_m} \otimes (\boldsymbol{C_0}\boldsymbol{L_n}))\tilde{\boldsymbol{Q}}(\boldsymbol{k}) \\
&\quad - T(\boldsymbol{I_m} \otimes (\check{\boldsymbol{C}}_0\boldsymbol{L_n})\tilde{\boldsymbol{P}}(\boldsymbol{k})) \\
&= -T\tilde{\boldsymbol{P}}(\boldsymbol{k})^{\mathrm{T}}(\boldsymbol{I_m} \otimes (\check{\boldsymbol{C}}_0\boldsymbol{L_n}))\tilde{\boldsymbol{P}}(\boldsymbol{k}) \le 0.
\end{aligned}
$$

We thus obtain $\tilde{V}(k) \le \tilde{V}(0) < \infty$. So $\tilde{V}(k)$ is bounded. Therefore $\|\tilde{\boldsymbol{Q}}(\boldsymbol{k})\|$ and $\|\tilde{\boldsymbol{P}}(\boldsymbol{k})\|$ are also bounded. It then follows from the LaSalle Invariance Principle [18] that all trajectories of the agents converge to the largest invariant set inside the region

$$
\boldsymbol{S} = \{[\tilde{\boldsymbol{Q}}(\boldsymbol{k})^{\mathrm{T}}, \tilde{\boldsymbol{P}}(\boldsymbol{k})^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2mn} : \tilde{V}(k+1) - \tilde{V}(k) = 0\}
$$

The proof is omitted because of the page limitation and its similarity with the proof of Theorem 4.1 in [19]. $\tilde{V}(k+1) - \tilde{V}(k) = 0$, if and only if $\tilde{\boldsymbol{P}}(\boldsymbol{k}) = \mathbf{0}$. That is $\boldsymbol{p_1}(\boldsymbol{k}) = \boldsymbol{p_2}(\boldsymbol{k}) = \cdots = \boldsymbol{p_n}(\boldsymbol{k}) = \bar{\boldsymbol{p}}(\boldsymbol{k})$, which implies that the velocities of all agents will converge to the velocity of the COM asymptotically. Obviously, the derivative of $\tilde{\boldsymbol{P}}(\boldsymbol{k})$ is also equal to $\mathbf{0}$. Then note that

$$
\begin{aligned}
\tilde{\boldsymbol{P}}(\boldsymbol{k}+\boldsymbol{1}) - \tilde{\boldsymbol{P}}(\boldsymbol{k}) &= -T(\boldsymbol{I_m} \otimes (\boldsymbol{C_0}\boldsymbol{L_n}))\tilde{\boldsymbol{Q}}(\boldsymbol{k}) \\
&\quad - T(\boldsymbol{I_m} \otimes (\check{\boldsymbol{C}}_0\boldsymbol{L_n}))\tilde{\boldsymbol{P}}(\boldsymbol{k})
\end{aligned}
$$

we thus have $\tilde{\boldsymbol{Q}}(\boldsymbol{k}) = \mathbf{0}$. That is $\boldsymbol{q_1}(\boldsymbol{k}) = \boldsymbol{q_2}(\boldsymbol{k}) = \cdots = \boldsymbol{q_n}(\boldsymbol{k}) = \bar{\boldsymbol{q}}(\boldsymbol{k})$, which suggests that the positions of all agents will also converge to the position of the COM asymptotically. So far, we have completed the proof. □

Remark 4: As for the directed topology, the consensus of the MAS with the distributed MPC input $\boldsymbol{u_i}(\boldsymbol{k})$ may not be guaranteed. However, the Laplacian matrix of the undirected topology which is connected and the non-symmetric Laplacian matrix of the directed topology which is strongly connected and balanced have the similar characteristics (see more details in [17]). Therefore the results in Theorem 1 can be extended to the directed graph case if it is strongly connected and balanced.

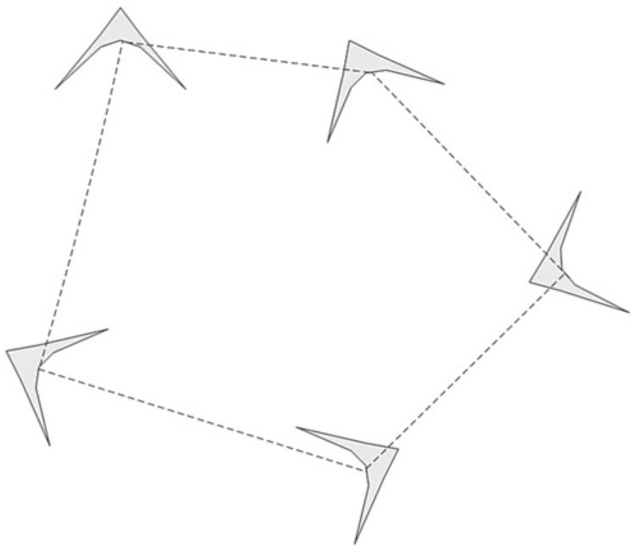

Fig. 1 *Illustration of the five-UAV MAS*

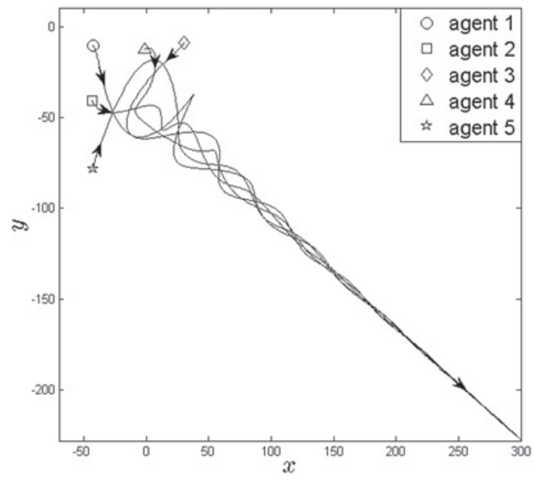

Fig. 2 *Moving trajectories (positions) evolution of the MAS*

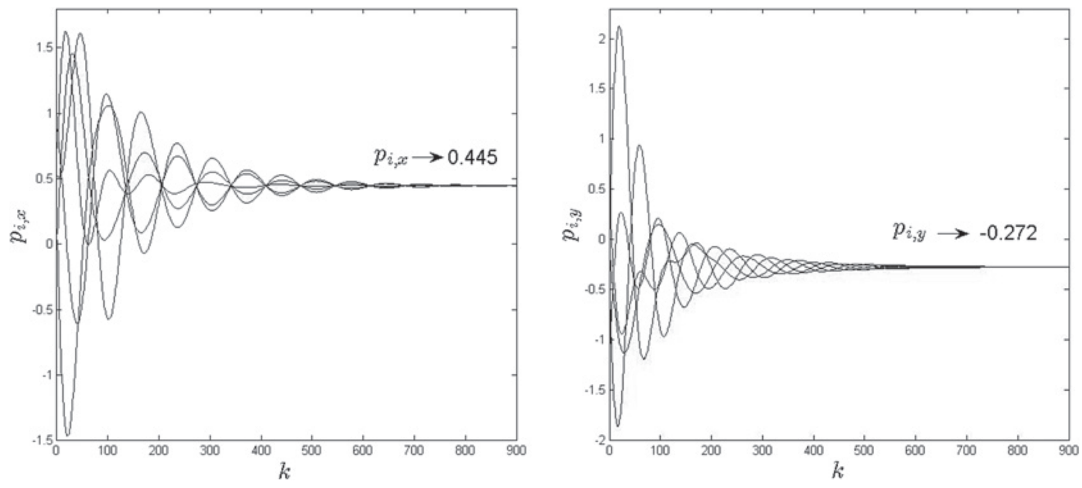

Fig. 3 *Velocities evolution of the MAS*

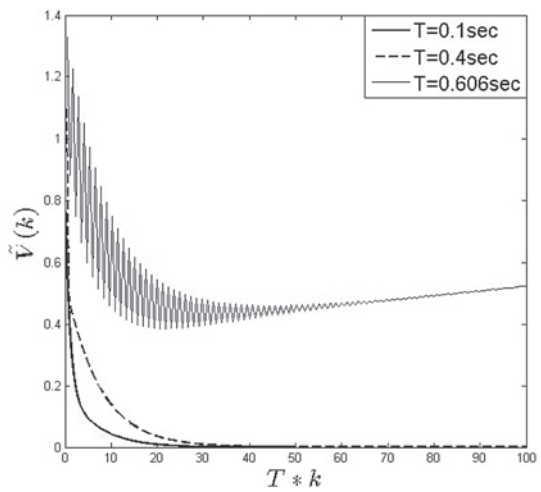

Fig. 4 *Impact of sampling period on consensus of the MAS*

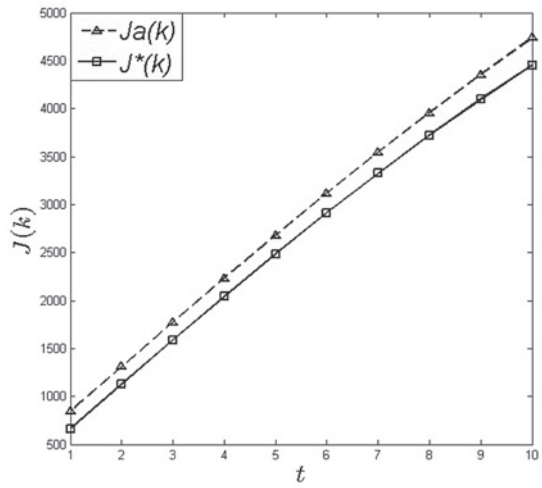

Fig. 5 *Total energy of the MAS in an optimisation process at instant k*

4 Illustrative examples

We present illustrative examples in this section to verify the effectiveness of the proposed distributed MPC consensus strategy. Consider a five-UAV MAS in Fig. 1 following dynamics (3) with distributed MPC input $\boldsymbol{u_i(k)}$. Five UAVs, each endowed with an engine and dynamically coupled in a two-nearest neighboured way (the dashed line represents the transmission of information between two UAVs in Fig. 1), are tasked to accomplish the space docking mission simultaneously. Obviously, the topology in Fig. 1 satisfies Assumption 1. The UAVs are all flying at distinct, preassigned altitudes to avoid collision as they perform docking mission. Despite its simplicity, this example is of significant interest since it contains almost all the ingredients of a distributed control problem: dynamic coupling between subsystems (each UAV and its neighbours), modularity and collective behaviours.

The initial planar positions and velocities of five UAVs: $\boldsymbol{q_i(0)}$, $\boldsymbol{p_i(0)} \in \mathbb{R}^2$, $i = 1, 2, 3, 4, 5$ are randomly chosen from the boxes: $[-100, 100] \times [-100, 100]$ and $[-2, 2] \times [-2, 2]$, respectively. The scalars $\alpha = \beta = \lambda = 1$. Moreover they all remain fixed throughout all simulations in this section. Then, we display Table 1 as below to show the values of c_t, $\check{c}_t$ $(t = 0, 1, \ldots, H_u - 1)$ which we will use later and the computation time t_c associated with prediction horizon H_p, control horizon H_u and sampling period T.

To visualise the performance of the five-UAV MAS with distributed MPC consensus strategy, we plot trajectories (positions) and velocities of all the agents with $\{H_\text{p} = 10, H_u = 1, T = 0.1\,\text{s}\}$ in Figs. 2 and 3, respectively. It is visualised that the velocities achieve consensus while positions converge to a same manifold, and hence the consensus of the MAS is guaranteed.

Moreover, the trajectory of Lyapunov function $\tilde{V}(k)$ is plotted in Fig. 4 to illustrate the impact of sampling period T on consensus. When we choose sampling period T from 0.1 to 0.606 s with $\{H_\text{p} = 20, H_u = 2\}$, the trajectories of $\tilde{V}(k)$ indicate that only a suitable range of sampling periods can guarantee the consensus of the MAS. If sampling period is not in the suitable range, for example, $T \geq 0.606$ s, the consensus of the MAS is totally broken. This is the reason why we have always emphasised that the sampling period T is a small positive constant before.

Next, we plot the total energy of the MAS $J(k) := \sum_{i=1}^{N} J_i(k)$ with $J_i(k)$ defined in (10) in an optimisation process at instant k to show the advantage of our distributed MPC strategy in Fig. 5. The

Table 1 Values of c_t, $\check{c}_t (t = 0, 1, \ldots, H_u - 1)$ and the computation time t_c

H_p	H_u	T, s	c_0	$\check{c}_0$	c_1	$\check{c}_1$	c_2	$\check{c}_2$	…	…	t_c, s
10	1	0.1	0.0156	0.1153	*	*	*	*	*	*	0.034
20	2	0.1	0.0070	0.1070	0.0081	0.1080	*	*	*	*	0.042

coefficient set $\{c_t, \check{c}_t\}$ with $\{H_p = 10, H_u = 1, T = 0.1\,\text{s}\}$ in Table 1, is applied to plot the path $J^*(k)$. On the other hand, $J_a(k)$ is plotted when an arbitrary coefficient set $\{c_0 = 0.02, \check{c}_0 = 0.2\}$ is selected. It is observed that the total energy consumption associated with the coefficient set in Table 1 is always less than it associated with the arbitrary coefficient set in an optimisation process. Thus the proposed distributed MPC consensus strategy has more advantages in terms of energy consumption.

5 Conclusion

In this paper, we have presented a distributed MPC strategy to analyse the consensus of sampled-data MAS with double-integrator dynamics, of which the convergence proof has been given for network with undirected topology. Such a distributed MPC strategy, based on the error of state between each agent and the centre of its subsystem, is acquired by reverse iterative algorithm which is specially designed for receding horizon optimisation of sampled-data double-integrator dynamics. The performance of our proposed distributed MPC strategy and the impact on system's consensus of the sampling period have also been verified illustratively.

6 Acknowledgments

This work was supported by the National Nature Science Foundation of China (61233004, 61221003, 61473184), the National Basic Research Program of China (973 Program - 2013CB035500), and partly sponsored by the Higher Education Research Fund for the Doctoral Program of China (20120073130006).

7 References

1 Olfati-saber, R., Fax, A., Murray, R.M.: 'Consensus and cooperation in multi-agent networked systems', *Proc. IEEE*, 2007, **95**, (1), pp. 215–233
2 Reynolds, C.W.: 'Flocks, herds, and schools: a distributed behavioral model'. ACM SIGGRAPH Conf. Proc. Comput. graph., Anaheim, California, July 1987, pp. 25–34
3 Vicsek, T., Czirók, A., Ben-Jacob, E. *et al.*: 'Novel type of phase transition in a system of self-driven particles', *Phys. Rev. Lett.*, 1995, **75**, (6), pp. 1226–1229
4 Jadbabaie, A., Lin, J., Morse, A.S.: 'Coordination of groups of mobile autonomous agents using nearest neighbor rules', *IEEE Trans. Autom. Control,* 2003, **48**, (6), pp. 988–1001
5 Fax, A., Murray, R.M.: 'Information flow and cooperative control of vehicle formations', *IEEE Trans. Autom. Control,* 2004, **49**, (9), pp. 1465–1476
6 Olfati-Saber, R.: 'Flocking for multi-agent dynamic systems: algorithms and theory', *IEEE Trans. Autom. Control,* 2006, **51**, (3), pp. 401–420
7 Olfati-Saber, R., Murray, R.M.: 'Consensus problems in networks of agents with switching topology and time-delays', *IEEE Trans. Autom. Control,* 2004, **49**, (9), pp. 1520–1533
8 Ren, W., Beard, R.W.: 'Consensus seeking in multiagent systems under dynamically changing interaction topologies', *IEEE Trans. Autom. Control,* 2005, **50**, (5), pp. 655–661
9 Ren, W., Beard, R.W., Arkins, E.M.: 'Information consensus in multivehicle cooperative control', *IEEE Control Syst. Mag.*, 2007, **71**, (2), pp. 71–82
10 Yu, W., Chen, G., Wang, Z. *et al.*: 'Distributed consensus filtering in sensor networks', *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 2009, **39**, (6), pp. 1568–1577
11 Woods, E.F.: 'Electronic prediction of swarming in bees', *Nature,* 1959, **184**, (4690), pp. 842–844
12 Melcher, D.: 'Predictive remapping of visual features precedes saccadic eye movements', *Nat. Neurosci.*, 2007, **10**, (7), pp. 903–907
13 Ferrari-Trecate, G., Galbusera, L., Marciandi, M.P.E.: 'Model predictive control schemes for consensus in multi-agent systems with single and double integrator dynamics', *IEEE Trans. Autom. Control,* 2009, **54**, (11), pp. 2560–2572
14 Zhang, H.T., Chen, M.Z.Q., Zhou, T.: 'Improve consensus via decentralized predictive mechanisms', *Europhys. Lett.*, 2009, **86**, (4), p. 40011
15 Zhang, H.T., Chen, M.Z.Q., Stan, G.B.: 'Fast consensus via predictive pinning control', *IEEE Trans. Circuits Syst. I, Regul. Pap.*, 2011, **58**, (9), pp. 2247–2258
16 Zhan, J., Li, X.: 'Consensus of sampled-data muti-agent networking systems via model predictive control', *Automatica,* 2013, **49**, (8), pp. 2502–2507
17 Godsil, C., Royle, G.: 'Algebraic graph theory' (Springer, New York, 2001)
18 Khalil, H.K.: 'Nonlinear systems' (Prentice-Hall, Upper Saddle River, 2002)
19 Su, H., Wang, X.: 'Pinning control of complex networked systems: synchronization, consensus and flocking of networked systems via pinning' (Springer, Berlin and Heidelberg GmbH & Co. K, 2013)