Consensus of Multi-Agent Systems with Input Constraints Based on Distributed Predictive Control Scheme

Yueqi Hou[1], Xiaolong Liang[1, 2], Lyulong He[1], Jiaqiang Zhang[1, *], Jie Zhu[3] and Baoxiang Ren[3]

Abstract: Consensus control of multi-agent systems has attracted compelling attentions from various scientific communities for its promising applications. This paper presents a discrete-time consensus protocol for a class of multi-agent systems with switching topologies and input constraints based on distributed predictive control scheme. The consensus protocol is not only distributed but also depends on the errors of states between agent and its neighbors. We focus mainly on dealing with the input constraints and a distributed model predictive control scheme is developed to achieve stable consensus under the condition that both velocity and acceleration constraints are included simultaneously. The acceleration constraint is regarded as the changing rate of velocity based on some reasonable assumptions so as to simplify the analysis. Theoretical analysis shows that the constrained system steered by the proposed protocol achieves consensus asymptotically if the switching interaction graphs always have a spanning tree. Numerical examples are also provided to illustrate the validity of the algorithm.

Keywords: Multi-agent systems, consensus, input constraints, model predictive control, distributed control, switching interaction graphs.

1 Introduction

Consensus means that a group of dynamic agents agree upon a certain quantity of interests such as position and orientation, and which is one of the most fundamental problems in multi-agent systems (MASs) [Olfati-Saber, Fax and Murray (2007); Zhan and Li (2013)]. Due to the promising applications both in military and civil areas, especially in fields such as multi-robot systems and sensor networks, consensus control for multi-agent systems has attracted great attention from various domains [Dong and Geng (2015); Olfati-Saber and Murray (2004); Sahin (2005); Brambilla, Ferrante, Birattari et al. (2013)]. Consensus-seeking problems should be addressed using distributed protocols based on local information since these systems are very large-scale while the included individuals only have limited situational awareness [Zhang, Cheng, Chen et al. (2015); Kia, Cortés and Martínez (2015); Liu, Dou and Sun (2016)].

Many distributed consensus control algorithms have been put forward in robotics and

[1] National Key Laboratory of Air Traffic Collision Prevention, Air Force Engineering University, Xi’an, 710051, China.
[2] Shanxi Province Laboratory of Meta-Synthesis for Electronic and Information System, Xi’an, 710051, China.
[3] Air Traffic Control and Navigation College, Air Force Engineering University, Xi’an, 710051, China.
[*] Corresponding Author: Jiaqiang Zhang. Email: jiaqiang-z@163.com

control communities. In the 1980s, Reynolds [Reynolds (1987)] introduced three heuristic rules for biological swarms, which can be summarized as cohesion, separation, and alignment. In 1995, Vicsek et al. [Vicsek, Czirók, Benjacob et al. (2006)] proposed a flocking model for self-driven particle systems based on inter-agent velocity alignments. Olfati-Saber and Murray [Olfati-Saber and Murray (2003)] provided a pioneering theoretical framework for an analysis of consensus protocols for multi-agent networked systems. Jadbabaie, Lin et al. [Jadbabaie, Lin and Morse (2002)] explicitly took into account possible changes in nearest neighbors over time and provided a theoretical explanation for behavior observed in the simulation studies by Vicsek et al. [Vicsek, Czirók, Benjacob et al. (2006)]. Ren et al. [Ren and Beard (2005)] extended the results to the presence of limited and unreliable information exchange with dynamically changing interaction topologies. In the latest literatures, discussions have been focused on the directed information flow, switching network topologies, time-delays, and performance guarantees when disturbance exists [Li, Chen, Dong et al. (2016); Thunberg, Goncalves and Hu (2016); Cao, Xiao and Wang (2016); Sun and Ruan (2008)].

From the aforementioned references, we find that most of the existing consensus protocols only use the current information to design distributed control input. However, abundant evidence have shown that many creatures have the ability to predict the future motions of their neighbors [Montague, Dayan, Person et al. (1995)]. Inspired by these biological clues, model predictive control (MPC) was introduced to investigate the consensus problems of engineering multi-agent systems [Zhang, Cheng, Fan et al. (2008); Galbusera, Ferrari-Trecate and Scattolini (2013)]. An advantage of MPC is the capability to handle large-scale control problems, to cope with constraints on input variables and states easily, and that the control update rates are relatively low [Ferrari-Trecate (2008)]. Zhang et al. [Zhang, Chen and Stan (2011)] have proved that the convergence speed can be substantially increased while the total communication cost can be reduced if agents had predictive intelligence. Ferrari-Trecate et al. [Ferrari-Trecate, Galbusera, Marciandi et al. (2007)] proposed an innovative solution for consensus to time-varying and undirected communication graphs based on MPC and these results were extended to the case of directed graphs [Ferrari-Trecate, Galbusera, Marciandi et al. (2009)]. Trodden et al. [Trodden and Richards (2013)] developed a cooperative, distributed form of MPC for linear systems with persistent, bounded disturbances. Zhan et al. [Zhan and Li (2013)] proposed a weighted-average consensus protocol based on model predictive control and analyzed the stability for networks with fixed and switching topologies. Kumar et al. [Kumar and Kothare (2013)] designed a novel broadcast stochastic receding horizon control architecture using the only available feedback information and broadcast it to all agents to achieve the desired system behavior. Zhong et al. [Zhong, Sun, Wang et al. (2015)] concerned the consensus problems for first- and second-order discrete-time multi-agent systems with delays based on MPC Schemes. Cheng et al. [Cheng, Fan and Zhang (2015)] applied model predictive control schemes to consensus control in MASs with single-integrator dynamics under switching directed interaction graphs and derived the requirements for sampling period to achieve consensus.

It is a remarkable fact that most of the existing works do not account for input constraints, which in many cases have to be included in engineering networked systems due to actuators limitations. Cheng et al. [Cheng, Zhang, Fan et al. (2015)] extended their previous results to discrete-time double-integrator consensus problems with directed switching interaction topologies and acceleration constraints. However, owing to the limitation of available power and safety reasons, which is often encountered in practical applications, constraints on velocity should also be included. Therefore, we are trying to develop a discrete-time consensus protocol for a class of MASs with single-integrator dynamics and switching topologies, and both the velocity and acceleration constraints are included simultaneously under the MPC framework.

The remainder of this paper is organized as follows. In Section 2, some necessary preliminary results and lemmas are described together with problem description. Section 3 gives an MPC protocol for MASs with single-integrator dynamics and constraints. Thereafter, the corresponding stability analysis is provided in Section 4. Numerical examples are provided in Section 5 to illustrate the validity of the algorithm and Section 6 summarizes this paper.

Throughout this paper, $\boldsymbol{I}_n$ and $\mathbf{1}_n$ denote identity matrix and the column vector of all ones of dimension n, respectively. $\boldsymbol{M}_n(R)$ represents square matrices of order n and the operator "$\otimes$" denotes the Kronecker product. Let $\mathrm{col}[\boldsymbol{x}_1(k),\cdots,\boldsymbol{x}_n(k)]=[\boldsymbol{x}_1^{\mathrm{T}}(k),\cdots,\boldsymbol{x}_n^{\mathrm{T}}(k)]^{\mathrm{T}}$ with vectors $\boldsymbol{x}_i(k)\in R^m$. The notation $\|\bullet\|$ denotes Euclidean norm.

2 Preliminary and problem description

Let $G=(W,E,\boldsymbol{A})$ be a weighted directed graph with vertices set $W=\{w_1,\ldots,w_n\}$, edge set $E\subseteq\{(w_i,w_j):w_i,w_j\in W\}$ and weighted adjacency matrix $\boldsymbol{A}=[a_{ij}]_{n\times n}$, which describes the interaction topology among agents. If there exists a directed edge $e_{ij}\in E$ between w_i and w_j, agent j is called a neighbor of agent i, i.e., agent i has information of agent j. $N_i=\{w_j\mid w_j\in W:e_{ij}\in E\}$ is defined as the neighbor set of agent i. The adjacency matrix $\boldsymbol{A}$ satisfies $a_{ij}=1$ if and only if $w_j\in N_i(j\neq i)$, otherwise $a_{ij}=0$. A directed graph G is said to have a spanning tree if there is at least one agent having directed paths to all the other agents. Laplacian matrix $\boldsymbol{L}=[l_{ij}]_{n\times n}$ plays an important role in the description of neighbor relationship of a graph and it is defined by

$$l_{ij}=\begin{cases}\sum_{k=1,\mathrm{k}\neq i}^{n}a_{ik}, & i=j\\ -a_{ij}, & i\neq j\end{cases} \tag{1}$$

Before discussing the main problem addressed in this note, we also need some necessary preliminary results and lemmas on graph and matrix theory first.

Lemma 2.1: Let $f(\boldsymbol{V})$ be a quadratic function of $\boldsymbol{V}=\mathrm{col}[\boldsymbol{v}_1,\cdots,\boldsymbol{v}_n]$, $\boldsymbol{v}_i=[v_{i,1},\cdots,v_{i,m}]^{\mathrm{T}}$, $v_{i,j}$ is jth element of $\boldsymbol{v}_i$. Suppose $f(\boldsymbol{V})$ can be rewritten as

$$f(\boldsymbol{V})=\sum_{l=1}^{m} f_l(\boldsymbol{V}_l), \tag{2}$$

with $\boldsymbol{V}_l=\text{col}[v_{1,l},\cdots,v_{n,l}]$, $f_l(\boldsymbol{V}_l)=(1/2)\boldsymbol{V}_l^{\text{T}}\boldsymbol{H}_l\boldsymbol{V}_l+\boldsymbol{V}_l^{\text{T}}\boldsymbol{\zeta}_l+c_l$. $\boldsymbol{H}_l=[h_{ij,l}]\in\boldsymbol{M}_n(R)$ is a nonsingular symmetric matrix with $h_{ij,l}>0$, $\boldsymbol{\zeta}_l=[\zeta_{1,l},\cdots,\zeta_{n,l}]^{\text{T}}$ is a column vector and c_l is a constant, $l=1,\cdots,m$, $i,j=1,\cdots,n$. $f(\boldsymbol{V})$ achieves its unique minimal point at $\boldsymbol{V}^*=\text{col}[\boldsymbol{v}_1^*,\cdots,\boldsymbol{v}_n^*]$, then it holds that

1). $\boldsymbol{V}^*$ can be calculated by stacking each minimum-value point of $f_l(\boldsymbol{V}_l)$, i.e., one can obtain the minimum-value point of $f(\boldsymbol{V})$ by seeking for the minimum-value point of $f_l(\boldsymbol{V}_l)$.

2). Considering the following constrained optimization problem,

$$\min f(\boldsymbol{V}), \text{s.t.} \|\boldsymbol{v}_i\|\le\overline{v},$$

whose unique optimal point is denoted by $\hat{\boldsymbol{V}}=\text{col}[\hat{\boldsymbol{v}}_1,\cdots,\hat{\boldsymbol{v}}_n]$, if $\boldsymbol{H}_l\equiv\boldsymbol{H}$ for all $l=1,\cdots,m$, then

$$\hat{\boldsymbol{v}}_i(k)=\begin{cases}\boldsymbol{v}_i^*(k), & \text{if } \|\boldsymbol{v}_i^*(k)\|\le\overline{v},\\ \dfrac{\overline{v}}{\|\boldsymbol{v}_i^*(k)\|}\boldsymbol{v}_i^*(k), & \text{otherwise.}\end{cases} \tag{3}$$

3). Considering the following constrained optimization problem,

$$\min f(\boldsymbol{V}), \text{s.t.} \|\boldsymbol{v}_i\|\ge\underline{v},$$

whose unique optimal point is denoted by $\hat{\boldsymbol{V}}=\text{col}[\hat{\boldsymbol{v}}_1,\cdots,\hat{\boldsymbol{v}}_n]$, if $\boldsymbol{H}_l\equiv\boldsymbol{H}$ for all $l=1,\cdots,m$, then

$$\hat{\boldsymbol{v}}_i(k)=\begin{cases}\boldsymbol{v}_i^*(k), & \text{if } \|\boldsymbol{v}_i^*(k)\|\ge\underline{v},\\ \dfrac{\underline{v}}{\|\boldsymbol{v}_i^*(k)\|}\boldsymbol{v}_i^*(k), & \text{otherwise.}\end{cases} \tag{4}$$

4). Denote $\boldsymbol{V}=\boldsymbol{V}_0+\boldsymbol{K}\boldsymbol{U}$, $g(\boldsymbol{U})=f(\boldsymbol{V}_0+\boldsymbol{K}\boldsymbol{U})$, where $\boldsymbol{V}_0$ is a constant matrix and $\boldsymbol{K}$ is a nonsingular and compatible matrix. Suppose the quadratic function $g(\boldsymbol{U})$ achieves its unique minimal point at $\boldsymbol{U}^*$, then it follows:

$$\boldsymbol{V}^*=\boldsymbol{V}_0+\boldsymbol{K}\boldsymbol{U}^*. \tag{5}$$

Proof:

The proof of part 1) and 2) can be found in Cheng et al. [Cheng, Zhang, Fan et al. (2015)], part 3) can be obtained analogously as part 2). Hence, only the proof of part 4) is given in this note.

According to the definition of $g(\boldsymbol{U})$, it can be obtained that

$$g(\boldsymbol{U})=\sum_{l=1}^{m} g_l(\boldsymbol{U}_l)=\sum_{l=1}^{m} f_l(\boldsymbol{V}_{0,l}+\boldsymbol{K}\boldsymbol{U}_l),$$

with $f_l(\boldsymbol{V}_{0,l}+\boldsymbol{K}\boldsymbol{U}_l)=(1/2)(\boldsymbol{V}_{0,l}+\boldsymbol{K}\boldsymbol{U}_l)^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{V}_{0,l}+\boldsymbol{K}\boldsymbol{U}_l)+(\boldsymbol{V}_{0,l}+\boldsymbol{K}\boldsymbol{U}_l)^{\mathrm{T}}\boldsymbol{\zeta}_l+c_l$, where $\boldsymbol{V}_{0,l}$ is the lth line of $\boldsymbol{V}_0$.

One can directly calculate the minimal point $\boldsymbol{U}_l^*$ of $g_l(\boldsymbol{U}_l)$ by $\partial g_l(\boldsymbol{U}_l)/\partial \boldsymbol{U}_l=0$, $l=1,\cdots,m$, then it follows:

$$\boldsymbol{K}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{K}\boldsymbol{U}_l^*+\boldsymbol{K}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{V}_{0,l}+\boldsymbol{\zeta}_l)=0. \tag{6}$$

As we have known that

$$\boldsymbol{H}\boldsymbol{V}_l^*+\boldsymbol{\zeta}_l=0. \tag{7}$$

By virtue of part 1) and substituting (5) into (7), then (6) is obtained, thus the conclusion of part 4) can be verified.

The proof is thus completed.

Lemma 2.2 [Ren and Beard (2005)]: Let $\boldsymbol{A}=[a_{ij}]\in\boldsymbol{M}_n(R)$ be a stochastic matrix with $a_{ii}>0$. If the graph associated with $\boldsymbol{A}$ has a spanning tree, then $\boldsymbol{A}$ is SIA (stochastic, indecomposable and aperiodic). That is $\lim_{k\to\infty}\boldsymbol{A}^k=\boldsymbol{1}_n\boldsymbol{y}^{\mathrm{T}}$, where $\boldsymbol{y}$ is a nonnegative column vector satisfing $\boldsymbol{A}\boldsymbol{y}=\boldsymbol{y}$ and $\boldsymbol{1}^{\mathrm{T}}\boldsymbol{y}=\boldsymbol{y}$.

Lemma 2.3 [Cheng, Fan and Zhang (2015)]: Let $\boldsymbol{A}_1,\cdots,\boldsymbol{A}_k$ be a finite set of SIA matrices with the property for each sequence $\boldsymbol{A}_{i1},\cdots,\boldsymbol{A}_{ij}$ of positive length, the matrix product $\boldsymbol{A}_{ij}\boldsymbol{A}_{i(j-1)}\cdots\boldsymbol{A}_{i1}$ is SIA. Then for each infinite sequence $\boldsymbol{A}_{i1},\boldsymbol{A}_{i2},\cdots$, there exists a column vector $\boldsymbol{y}$ such that

$$\lim_{k\to\infty}\boldsymbol{A}_{ik}\boldsymbol{A}_{i(k-1)}\cdots\boldsymbol{A}_{i1}=\boldsymbol{1}_n\boldsymbol{y}^{\mathrm{T}}.$$

3 Predictive control-based on consensus with input constraints

3.1 Consensus with velocity constraint only

Consider that a system consists of *n* agents with discrete-time single-integrator dynamics and velocity constraints given as below.

$$\boldsymbol{x}_i(k+1)=\boldsymbol{x}_i(k)+T\boldsymbol{v}_i(k), s.t.\ \|\boldsymbol{v}_i\|\le\bar{v}, i=1,\cdots,n, \tag{8}$$

where $\boldsymbol{v}_i(k)\in R^m$, $\boldsymbol{x}_i(k)\in R^m$ are the velocity and position of agent *i*, respectively. $T\in R^+$ is the sampling period.

Denote

$$\boldsymbol{X}_i(k)=\mathrm{col}\,[\boldsymbol{x}_i(k+1\,|\,k),\boldsymbol{x}_i(k+2\,|\,k),\cdots,\boldsymbol{x}_i(k+Np\,|\,k)],$$

$$\boldsymbol{V}_i(k)=\mathrm{col}\,[\boldsymbol{v}_i(k),\boldsymbol{v}_i(k+1),\cdots,\boldsymbol{v}_i(k+Nc-1)].$$

Notations Np and Nc are the prediction and control horizon, respectively, fulfilling $Nc\le Np$.

Based on the nominal model (8) and velocity $\boldsymbol{V}_i(k)$ to be designed, the position of agents for the instants $k+t$, $t=1,\cdots,Np$ can be obtained as follows

$$\boldsymbol{x}_i(k+\mathrm{t}\,|\,k)=\boldsymbol{x}_i(k+t-1\,|\,k)+T\boldsymbol{v}_i(k+t-1),1\le t\le Nc-1,$$

$$\boldsymbol{x}_i(k+\mathrm{t}\,|\,k)=\boldsymbol{x}_i(k+t-1\,|\,k)+T\boldsymbol{v}_i(k+Nc-1),Nc\le t\le Np.$$

Then the above iteration can be rewritten in a compact form as:

$$\boldsymbol{X}_i(k)=\boldsymbol{P}_x\boldsymbol{x}_i(k\,|\,k)+\boldsymbol{Q}_x\boldsymbol{V}_i, \tag{9}$$

where $\boldsymbol{P}_x=[1,\cdots,1]^{\mathrm{T}}_{1\times Np}\otimes\boldsymbol{I}_m$,

$$\boldsymbol{Q}_x=\begin{bmatrix}1 & 0 & 0 & 0\\ \vdots & \ddots & \cdots & \vdots\\ 1 & \cdots & 1 & 0\\ 1 & \cdots & 1 & 1\\ 1 & \cdots & 1 & 2\\ \vdots & \cdots & \vdots & \vdots\\ 1 & \cdots & 1 & (Np-Nc+1)\end{bmatrix}_{Np\times Nc}\otimes(T\boldsymbol{I}_m).$$

As prepared above, the optimization problem with velocity constraints in the MPC scheme designed for agent *i* in a finite time horizon Np is described as follows:

$$\min J_i(k)=\frac{1}{2}\sum_{t=1}^{Np}\|\boldsymbol{x}_i(k+t\,|\,k)-\boldsymbol{r}_{i,x}(k+t)\|^2+\frac{\alpha}{2}\sum_{t=0}^{Nc-1}\|\boldsymbol{v}_i(k+t)\|^2, \tag{10}$$

with $\boldsymbol{r}_{i,x}(k+t)=\frac{1}{1+|N_i(k)|}\sum_{j\in n_i(k)\cup\{i\}}\boldsymbol{x}_j(k)$, and α is a positive weight coefficient.

Note that the definition of $\boldsymbol{r}_{i,x}(k+t)$ only depends on the position of agent *i* and its neighbors, which indicates that this control scheme is distributed.

For simplicity, we rewrite the MPC cost function (10) into a compact form and substituting (9) in, it derives:

$$\min J_i(k)=\frac{1}{2}\|\boldsymbol{P}_x\boldsymbol{x}_i(k\,|\,k)+\boldsymbol{Q}_x\boldsymbol{V}_i(k)-\boldsymbol{R}_{i,x}(k)\|^2+\frac{\alpha}{2}\|\boldsymbol{V}_i(k)\|^2, \tag{11}$$

where $\boldsymbol{R}_{i,x}(k)=col\,[\boldsymbol{r}_{i,x}(k),\cdots,\boldsymbol{r}_{i,x}(k)]$.

Eq. (11) is a quadratic function, whose minimum-value point $\boldsymbol{V}_i^*(k)$ can be calculated by using $\partial J_i(k)/\partial\boldsymbol{V}_i(k)=0$. Since $(\boldsymbol{Q}_x^{\mathrm{T}}\boldsymbol{Q}_x+\alpha\boldsymbol{I}_{Nc}\otimes\boldsymbol{I}_m)$ is always positive definite and invertible, $\boldsymbol{V}_i^*(k)$ is obtained as

$$\boldsymbol{V}_i^*(k)=-(\boldsymbol{Q}_x^{\mathrm{T}}\boldsymbol{Q}_x+\alpha\boldsymbol{I}_{Nc}\otimes\boldsymbol{I}_m)^{-1}\boldsymbol{Q}_x^{\mathrm{T}}[\boldsymbol{P}_x\boldsymbol{x}_i(k\,|\,k)-\boldsymbol{R}_{i,x}(k)], \tag{12}$$

whose first *m* entry will be actually implemented as the control input at sampling instant *k*, i.e., $\boldsymbol{v}_i^*(k)=\{[1,0,\cdots,0\,]_{1\times Np}\otimes\boldsymbol{I}_m\}\boldsymbol{V}_i^*(k)$.

If we take the velocity constraints $\| \boldsymbol{v}_i(k+t-1) \| \le \bar{v}$ into consideration, the control input can be obtained as (13) based on Lemma 2.1:

$$\hat{\boldsymbol{v}}_i(k) = \begin{cases} \boldsymbol{v}_i^*(k), & \text{if } \| \boldsymbol{v}_i^*(k) \| \le \bar{v}, \\ \dfrac{\bar{v}}{\| \boldsymbol{v}_i^*(k) \|} \boldsymbol{v}_i^*(k), & \text{otherwise.} \end{cases} \tag{13}$$

3.2 Consensus with velocity and acceleration constraints

Due to the limitation of structural strength or available overload, which is often encountered in practical networked systems, constraints on acceleration should also be considered. Therefore, both the velocity and acceleration constraints are considered simultaneously in this subsection.

Theorem 1: Assume each agent of the system has discrete-time second-order dynamics with input constraints given as follows:

$$\begin{aligned} \boldsymbol{x}_i(k+1) &= \boldsymbol{x}_i(k) + T\boldsymbol{v}_i(k), \\ \boldsymbol{v}_i(k) &= \boldsymbol{v}_i(k-1) + T\boldsymbol{u}_i(k). \end{aligned} \tag{14}$$

$$s.t. \| \boldsymbol{v}_i(k) \| \le \bar{v}, \| \boldsymbol{u}_i(k) \| \le \bar{u},\ \bar{\boldsymbol{v}} > 0, \bar{\boldsymbol{u}} > 0, \tag{15}$$

where $\boldsymbol{u}_i(k) \in R^m$ is the acceleration of i.

Suppose the constraint on turning rate is ignored and coordinated turn can always be implemented, that is, the velocity and acceleration are exactly in line. One can derive the actual control input as

$$\hat{\boldsymbol{v}}_i(k) = \eta_i \boldsymbol{v}_i^*(k), \tag{16}$$

with $\eta_i = \begin{cases} \dfrac{\bar{v}^*}{\| \boldsymbol{v}_i^*(k) \|}, & \| \boldsymbol{v}_i^*(k) \| > \bar{v}^* \\ 1, & \underline{v}^* \le \| \boldsymbol{v}_i^*(k) \| \le \bar{v}^* \\ \dfrac{\underline{v}^*}{\| \boldsymbol{v}_i^*(k) \|}, & \| \boldsymbol{v}_i^*(k) \| < \underline{v}^* \end{cases}$. $\boldsymbol{v}_i^*(k)$ is the optimal velocity when no input constraint exists, $\bar{v}^* = \min\{\bar{v}, \| \boldsymbol{v}_i(k-1) \| + T\bar{u}\}$, $\underline{v}^* = \max\{0, \| \boldsymbol{v}_i(k-1) \| - T\bar{u}\}$.

Proof:

Based on (14) and $\boldsymbol{u}_i(k)$ to be designed, the velocity of agent i for the instant $k+t$, $t = 1, \cdots, Nc$, can be obtained as follows

$$\boldsymbol{v}_i(k+t \mid k) = \boldsymbol{v}_i(k+t-1 \mid k) + T\boldsymbol{u}_i(k+t), 0 \le t \le Nc-1.$$

Denote

$$\boldsymbol{U}_i(k) = \text{col}\,[\boldsymbol{u}_i(k), \boldsymbol{u}_i(k+1), \cdots, \boldsymbol{u}_i(k+Nc-1)].$$

Then the above iteration can be rewritten in a compact form as:

$$\boldsymbol{V}_i(k) = \boldsymbol{P}_v \boldsymbol{v}_i(k-1) + \boldsymbol{Q}_v \boldsymbol{U}_i(k), \tag{17}$$

where $\boldsymbol{P}_v = [1,\cdots,1]^{\mathrm{T}}_{1\times Nc} \otimes \boldsymbol{I}_m, \boldsymbol{Q}_v = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \ddots & 0 \\ 1 & \cdots & 1 \end{bmatrix}_{Nc\times Nc} \otimes (T\boldsymbol{I}_m).$

Thus by virtue of (14) and (17), the position of agent i can be rewritten in a compact form as:

$$\boldsymbol{X}_i(k) = \boldsymbol{P}_x \boldsymbol{x}_i(k \mid k) + \boldsymbol{Q}_x \boldsymbol{P}_v \boldsymbol{v}_i(k-1) + \boldsymbol{Q}_x \boldsymbol{Q}_v \boldsymbol{U}_i(k). \tag{18}$$

Substituting (17) into the cost function (11) and the minimum-value point $\boldsymbol{U}_i^*(k)$ can be obtained by solving a quadratic problem. Then the first m entry of $\boldsymbol{U}_i^*(k)$ will be actually implemented at sampling instant k, i.e.,

$$\boldsymbol{u}_i^*(k) = \{[1,0,\cdots,0]_{1\times Nc} \otimes \boldsymbol{I}_m\}\boldsymbol{U}_i^*(k).$$

Then from part 4) of Lemma 2.1, the velocity at sampling instant k can be obtained as

$$\boldsymbol{v}_i^*(k) = \boldsymbol{v}_i(k-1) + T\boldsymbol{u}_i^*(k). \tag{19}$$

If we take the constraints $\| \boldsymbol{u}_i^*(k) \| \le \overline{u}$ into consideration and suppose that the velocity and acceleration are exactly in line, we have

$$\| \boldsymbol{v}_i(k-1) \| - T\overline{u} \le \| \hat{\boldsymbol{v}}_i(k) \| \le \| \boldsymbol{v}_i(k-1) \| + T\overline{u}, \tag{20}$$

where $\hat{\boldsymbol{v}}_i(k)$ denote the actually implemented control input.

On the other hand, $\| \hat{\boldsymbol{v}}_i(k) \|$ should fulfill the velocity constraint, i.e., $\| \hat{\boldsymbol{v}}_i(k) \| \le \overline{v}$. Denote $\overline{v}^* = \min\{\overline{v}, \| \boldsymbol{v}_i(k-1) \| + T\overline{u}\}$, $\underline{v}^* = \max\{0, \| \boldsymbol{v}_i(k-1) \| - T\overline{u}\}$, on basis of part 2) and 3) of Lemma 2.1, then the actually implemented control input can be rewritten as (16) with

$$\eta_i = \begin{cases} \dfrac{\overline{v}^*}{\| \boldsymbol{v}_i^*(k) \|}, & \| \boldsymbol{v}_i^*(k) \| > \overline{v}^* \\ 1, & \underline{v}^* \le \| \boldsymbol{v}_i^*(k) \| \le \overline{v}^* \\ \dfrac{\underline{v}^*}{\| \boldsymbol{v}_i^*(k) \|}, & \| \boldsymbol{v}_i^*(k) \| < \underline{v}^* \end{cases} \tag{21}$$

The proof is thus completed.

Remark 1: Note that although the double-integrator dynamics of the system is adopted, $\boldsymbol{v}_i(k)$ is still implemented as the control input, while $\boldsymbol{u}_i(k)$ is regarded as the changing rate of velocity input. No specific expression of $\boldsymbol{U}_i^*(k)$ or $\boldsymbol{u}_i^*(k)$ is given as it is indeed an intermediate variable used in the derivation process.

Remark 2: As a matter of fact, the control input under different constraints can be obtained by changing the value of $\overline{v}$ and $\overline{u}$. When $\overline{v}$ or $\overline{u}$ is set as positive infinity, it means that the corresponding constraints for $\overline{v}$ or $\overline{u}$ will not be activated.

4 Stability analysis

In this section, we will present the convergence analysis of the consensus protocol.

Before seeking to analytically solve the problem, we first give the solution with a much simpler form to simplify the analysis.

Theorem 2: If the quadratic optimization problem (11) is feasible with constraints described in (15), then the control input $\boldsymbol{v}_i^*(k)$ without constraints has an equivalent expression as below:

$$\boldsymbol{v}_i^*(k) = -\sigma_i[\boldsymbol{x}_i(k\,|\,k) - \boldsymbol{r}_{i,x}(k)], \tag{22}$$

with $\sigma_i > 0$.

Proof:

By the definition of $\boldsymbol{P}_x$ and $\boldsymbol{R}_{i,x}(k)$, we know that $\boldsymbol{P}_x\boldsymbol{x}_i(k\,|\,k) - \boldsymbol{R}_{i,x}$ is an *mNp*-dimensional column vector which can be rewritten as follows:

$$\boldsymbol{P}_x\boldsymbol{x}_i(k\,|\,k) - \boldsymbol{R}_{i,x} = \boldsymbol{P}_x[\boldsymbol{x}_i(k\,|\,k) - \boldsymbol{r}_{i,x}(k)]. \tag{23}$$

Recall (12) and substitute (23) in, it yields that

$$\boldsymbol{V}_i^*(k) = -\left(\boldsymbol{Q}_x^{\mathrm{T}}\boldsymbol{Q}_x + \alpha\boldsymbol{I}_{Nc} \otimes \boldsymbol{I}_m\right)^{-1}\boldsymbol{Q}_x^{\mathrm{T}}\boldsymbol{P}_x[\boldsymbol{x}_i(k\,|\,k) - \boldsymbol{r}_{i,x}(k)]. \tag{24}$$

Consider that only the first *m* entry of $\boldsymbol{V}_i^*(k)$ will be actually implemented as the control input, it follows that

$$\begin{aligned}\boldsymbol{v}_i^*(k) &= -\begin{bmatrix}\phi_1\\ \vdots\\ \phi_m\end{bmatrix}\begin{bmatrix}\boldsymbol{P}_{x,1}, \cdots, \boldsymbol{P}_{x,m}\end{bmatrix}[\boldsymbol{x}_i(k\,|\,k) - \boldsymbol{r}_{i,x}(k)]\\ &= -\begin{bmatrix}\phi_1\boldsymbol{P}_{x,1} & \cdots & \phi_1\boldsymbol{P}_{x,m}\\ \vdots & \ddots & \vdots\\ \phi_m\boldsymbol{P}_{x,1} & \cdots & \phi_m\boldsymbol{P}_{x,m}\end{bmatrix}[\boldsymbol{x}_i(k\,|\,k) - \boldsymbol{r}_{i,x}(k)],\end{aligned} \tag{25}$$

where ϕ_p is the *p*-th row of $\left(\boldsymbol{Q}_x^{\mathrm{T}}\boldsymbol{Q}_x + \alpha\boldsymbol{I}_{Nc} \otimes \boldsymbol{I}_m\right)^{-1}\boldsymbol{Q}_x^{\mathrm{T}}$ and $\boldsymbol{P}_{x,q}$ is the *q*-th column of $\boldsymbol{P}_x$. Denote

$$\sigma_{pq} = \phi_p\boldsymbol{P}_{x,q} = \sum_{l=1}^{mNp}\phi_{pl}\boldsymbol{P}_{x,lq},\; p,q = 1,\cdots,m.$$

By virtue of stacking matrices and Kronecker product, it is easy to verify that

$$\sigma_{pq} = \begin{cases}\sigma_{pp} & p = q\\ 0 & \text{otherwise}\end{cases} \tag{26}$$

with $\sigma_{11} = \cdots = \sigma_{mm} = \sigma_i$.

Since agent i will always move toward the desired position $\boldsymbol{r}_{i,x}(k)$, $\sigma_i > 0$ is guaranteed. Then it derives that

$$\boldsymbol{v}_i^*(k) = -\sigma_i[\boldsymbol{x}_i(k\,|\,k) - \boldsymbol{r}_{i,x}(k)]. \tag{27}$$

Thus the proof is completed.

Remark 3: One can see that the consensus protocol is not only distributed but also only depends on the errors of states between agent i and its neighbors. In fact, just as described in many of the literature, such errors are usually sufficient for consensus control. Substituting the result of Lemma 4.1 into (16), the actual implemented control input $\hat{\boldsymbol{v}}_i(k)$ can be rewritten as

$$\hat{\boldsymbol{v}}_i(k) = -\eta_i\sigma_i[\boldsymbol{x}_i(k \mid k) - \boldsymbol{r}_{i,x}(k)]. \tag{28}$$

Theorem 3: Consider a system consists of n agents, whose dynamics and constraints of each agent are described in (14) and (15) with control input given in (28). Then there exists a stochastic matrix $\boldsymbol{D}(k)$ such that the discrete time update scheme of the system can be written as

$$\boldsymbol{X}(k+1) = \boldsymbol{D}(k)\boldsymbol{X}(k),$$

with $\boldsymbol{X}(k) = \mathrm{col}[\boldsymbol{x}_1(k), \cdots, \boldsymbol{x}_n(k)]$.

Proof:

Recall the definition of $\boldsymbol{r}_{i,x}(k)$ and substitute into the control input (28), it gives

$$\begin{aligned}\hat{\boldsymbol{v}}_i(k) &= -\eta_i\sigma_i\left(\boldsymbol{x}_i(k \mid k) - \frac{1}{1+|N_i(k)|}\sum_{j\in N_i(k)\cup\{i\}}\boldsymbol{x}_j(k)\right)\\ &= -\frac{\eta_i\sigma_i}{1+|N_i(k)|}\boldsymbol{L}_i(k)\otimes\boldsymbol{I}_m\boldsymbol{X}(k),\end{aligned} \tag{29}$$

where $\boldsymbol{L}_i(k)$ is the i-th row of the Laplacian matrix associated with the interaction graph.

Hence, it follows from (29) that

$$\hat{\boldsymbol{V}}(k) = \mathrm{col}[\hat{\boldsymbol{v}}_1(k), \cdots, \hat{\boldsymbol{v}}_n(k)] = -\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{\Theta}_k\boldsymbol{L}(k)\otimes\boldsymbol{I}_m\boldsymbol{X}(k), \tag{30}$$

where $\boldsymbol{L}(k)$ is the Laplacian matrix, $\boldsymbol{H} = diag(\eta_1, \cdots, \eta_n)$, $\boldsymbol{\Sigma} = diag(\sigma_1, \cdots, \sigma_n)$, $\boldsymbol{\Theta}_k = diag(\frac{1}{|N_1(k)|+1}, \cdots, \frac{1}{|N_n(k)|+1})$.

Hence, it follows from (14) that

$$\boldsymbol{X}(k+1) = \boldsymbol{X}(k) + T\hat{\boldsymbol{V}}(k) = [\boldsymbol{I}_n - T\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{\Theta}_k\boldsymbol{L}(k)]\otimes\boldsymbol{I}_m\boldsymbol{X}(k). \tag{31}$$

Denote $\boldsymbol{D}(k) = [\boldsymbol{I}_n - T\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{\Theta}_k\boldsymbol{L}(k)]\otimes\boldsymbol{I}_m$. One can see that $\frac{|N_i(k)|}{|N_i(k)|+1} < 1$, thus enforcing $\eta_i\sigma_i T \le 1$ guarantees that $D_{ii}(k)$ is positive, that is, diagonal elements $D_{ii}(k) = 1 - T\sigma_i\eta_i\frac{|N_i(k)|}{|N_i(k)|+1}$ satisfies $D_{ii}(k) > 0$. Furthermore, it is not difficult to verify that $\boldsymbol{D}(k)$ satisfies the conditions that all off-diagonal elements are nonnegative and all its row sums are equal to 1, which indicates $\boldsymbol{D}(k)$ is a (row) stochastic matrix.

Thus the proof is completed.

Remark 4: One can see from Lemma 4.2 that the system achieves asymptotic consensus for any initial condition, if and only if there exists an infinite matrices sequence such that

$$\lim_{k\to\infty} \boldsymbol{X}(k) = \lim_{k\to\infty} \boldsymbol{D}(k-1)\cdots\boldsymbol{D}(0)\boldsymbol{X}(0) = \boldsymbol{X}_{SS},$$

where $\boldsymbol{X}_{SS}$ denotes the consensus state of the system.

Theorem 4: Suppose that the interaction graph, denoted as $G(k)$, changes at time $t = kT$, and keeps fixed across uniformly bounded and non-overlapping time interval $[kT, kT+T)$. Let a finite set G denote all the possible interaction topologies of the system, then $G(k) \in G$. The constrained system (14) with control input described in (16) achieves consensus asymptotically if the infinite sequence $G(k)$ always has a spanning tree.

Proof:

From Lemma 4.2, we know that $\boldsymbol{D}(k)$ describes the corresponding interaction topology of $G(k)$ and is a stochastic matrix with positive diagonal entries. The assumption that interaction graph $G(k)$ always has a spanning tree indicates that the graph associated with $\boldsymbol{D}(k)$ has a spanning tree. By virtue of Lemma 2.2, we know that $\boldsymbol{D}(k)$ is SIA. Then by applying Lemma 2.3, one gets that

$$\lim_{k\to\infty} \boldsymbol{X}(k) = \lim_{k\to\infty} \boldsymbol{D}(k-1)\cdots\boldsymbol{D}(0)\boldsymbol{X}(0) = \left(\mathbf{1}\boldsymbol{y}^{\mathrm{T}}\right)\boldsymbol{X}(0) = \boldsymbol{X}_{ss}.$$

Thus the consensus state is obtained.

5 Simulation study

In this section, numerical examples are presented to illustrate the feasibility of the distributed MPC consensus protocol by a planar problem.

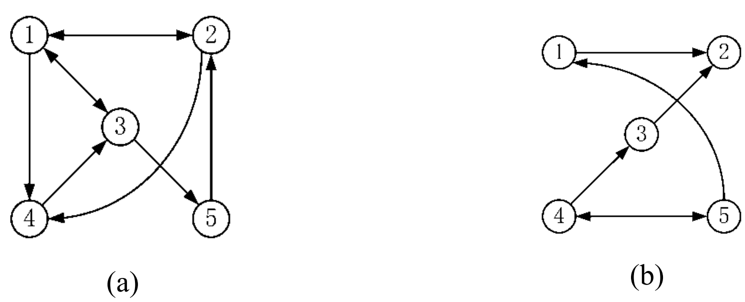

Figure 1: Interaction topologies: (a) G_1 and (b) G_2. An arrow from node j heading to node i implies agent i receives information of agent j; thus j is a neighbor of i. Both G_1 and G_2 have a directed spanning tree

Consider a system of $n = 5$ agents moving in two-dimensional plane, that is $m = 2$. The interaction topology switches from G_1 to G_2 periodically with period $T = 0.1$, which is sufficiently small and satisfies the constraint condition $\eta_i \sigma_i T < 1$. The graphs G_1, G_2 are defined in Fig. 1 and both have a directed spanning tree. The parameters are set as $Np = 10$, $Nc = 6$, $\alpha = 0.5$, $\overline{v} = 10$, $\overline{u} = 2$. The initial positions and velocities of all the

agents are set as $\boldsymbol{x}_1=[195,125]^{\mathrm{T}}$, $\boldsymbol{v}_1=[2,-8]^{\mathrm{T}}$, $\boldsymbol{x}_2=[25,80]^{\mathrm{T}}$, $\boldsymbol{v}_2=[-8,6]^{\mathrm{T}}$, $\boldsymbol{x}_3=[120,15]^{\mathrm{T}}$, $\boldsymbol{v}_3=[2,0]^{\mathrm{T}}$, $\boldsymbol{x}_4=[60,5]^{\mathrm{T}}$, $\boldsymbol{v}_4=[-3,6]^{\mathrm{T}}$, $\boldsymbol{x}_5=[20,185]^{\mathrm{T}}$, $\boldsymbol{v}_5=[-3,-4]^{\mathrm{T}}$.
The positional trajectories and velocities of all the agents are shown in Figs. 2 and 3. It can be observed that the control protocol (16) steers all the agents to an average consensus point asymptotically. The evolution of velocity and acceleration amplitude under different constraint conditions are shown in Fig. 4; one can see that both velocity and acceleration meet the condition of constraints, respectively. It can be observed from Figs. 4(a) and (b) that the velocity changes at a relatively gentle rate since the acceleration is constrained, while the results containing only velocity constraints in (c) and (d) are just the reverse. This implies the efficiency of our algorithm and also highlights the necessity of taking both velocity and acceleration constraints into consideration simultaneously. Thus the effectiveness of Theorem 1 is demonstrated.

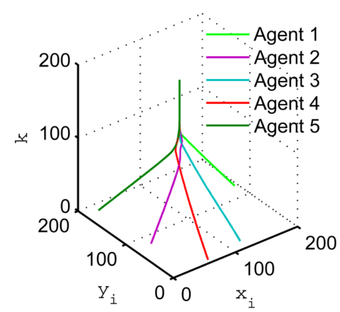

Figure 2: Evolution of the agent positions

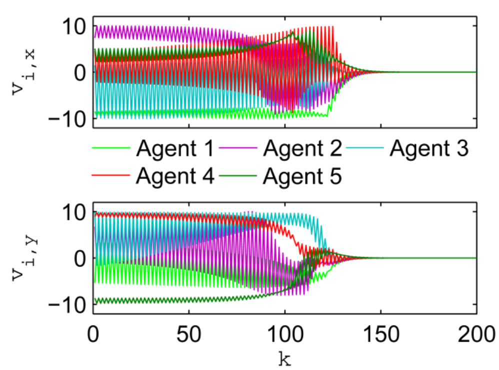

Figure 3: Evolution of the agent velocities with velocity and acceleration constraints

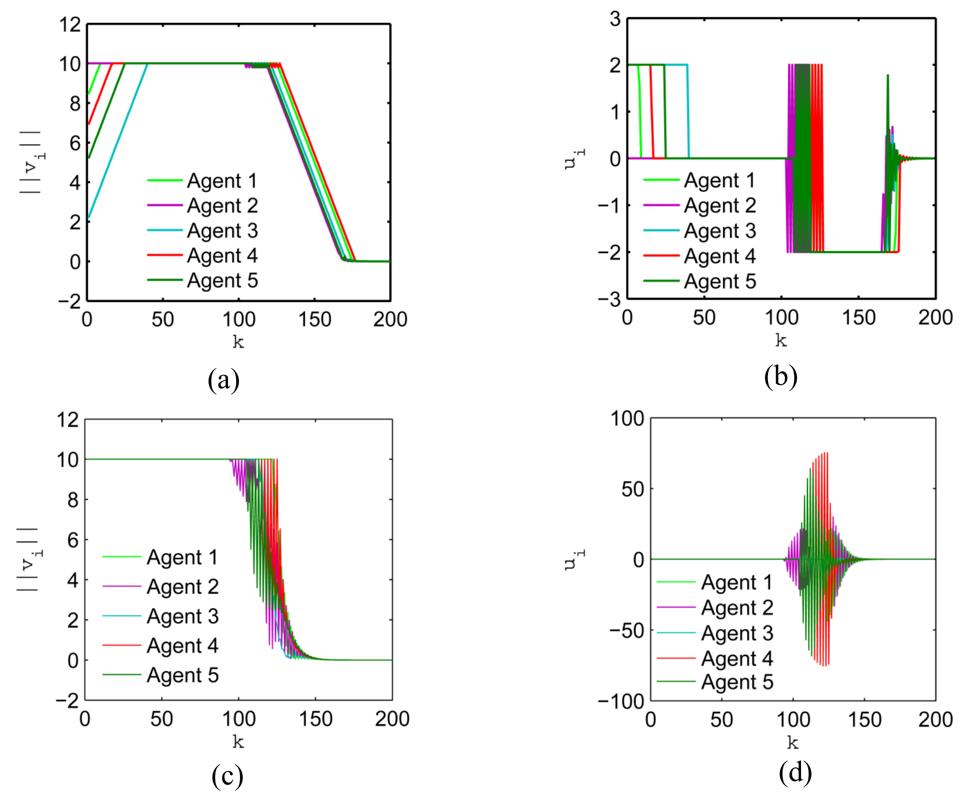

Figure 4: Evolution of velocity and acceleration amplitude under different constraint conditions: (a)-(b), both velocity and acceleration constraints are considered; (c)-(d) only velocity constraints are considered

It is worth mentioning that at least two factors are responsible for the large fluctuations in the control input $\boldsymbol{v}_{i,x}$ and $\boldsymbol{v}_{i,y}$ (Fig. 3). The first is the frequent graph switching over time. The second but the most is that there is no constraint on the turning rate of velocity vectors, which leads to sharp changes in velocity direction. This conclusion can be further confirmed by Fig. 4(a), in which one can see that the amplitude of velocities has no big fluctuations.

6 Conclusions

In this paper, a distributed MPC scheme has been developed to achieve consensus for MASs with single-integrator dynamics, input constraints and switching directed interaction topologies. The control inputs in analytic form are obtained under different constraints. Under the condition that the switching interaction graphs always have a spanning tree, we prove that the system containing velocity and acceleration constraints can achieve consensus asymptotically. Further extensions of this work will concern constraints on the turning rate of velocity vectors, which is a key factor to eliminate the large oscillations in control input.

Acknowledgement: This work was financially supported by the Major Program of National Natural Science Foundation of China [grant numbers is not public]; the National Natural Science Foundation of China [Grant No. 61703427].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Brambilla, M.; Ferrante, E.; Birattari, M.; Dorigo, M. (2013): Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, vol. 7, pp. 1-41.

Cheng, Z.; Fan, M.; Zhang, H. (2015): Distributed MPC based consensus for single-integrator multi-agent systems. *ISA Transactions*, vol. 58, pp. 112-120.

Cheng, Z.; Zhang, H.; Fan, M.; Chen, G. (2015): Distributed consensus of multi-agent systems with input constraints: a model predictive control approach. *IEEE Transactions on Circuits and Systems*, vol. 62, no. 3, pp. 825-834.

Cao, M.; Xiao, F.; Wang, L. (2016): Second-order consensus in time-delayed networks based on periodic edge-event driven control. *Systems & Control Letters*, vol. 96, pp. 37-44.

Dong, R.; Geng, Z. (2015): Consensus for formation control of multi-agent systems. *International Journal of Robust and Nonlinear Control*, vol. 25, no. 14, pp. 2481-2501.

Ferrari-Trecate, G.; Galbusera, L.; Marciandi, M. P. E.; Scattolini, R. (2007): A model predictive control scheme for consensus in multi-agent systems with single-integrator dynamics and input constraints. *IEEE Conference on Decision and Control*, pp. 1492-1497.

Ferrari-Trecate, G. (2008): Contractive distributed MPC for consensus in networks of single and double-integrators. *International Federation of Automatic Control*, vol. 41, no. 2, pp. 9033-9038.

Ferrari-Trecate, G.; Galbusera, L.; Marciandi, M. P. E.; Scattolini, R. (2009): Model predictive control schemes for consensus in multi-agent systems with single- and double-integrator dynamics. *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2560-2572.

Galbusera, L.; Ferrari-Trecate, G.; Scattolini, R. (2013): A hybrid model predictive control scheme for containment and distributed sensing in multi-agent systems. *Systems & Control Letters*, vol. 62, no. 5, pp. 413-419.

Jadbabaie, A.; Lin, J.; Morse, A. S. (2002): Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 2953-2958.

Kumar, G.; Kothare, M. V. (2013): Broadcast stochastic receding horizon control of multi-agent systems. *Automatica*, vol. 49, no. 12, pp. 3600-3606.

Kia, S. S.; Cortés, J.; Martínez, S. (2015): Distributed event-triggered communication for dynamic average consensus in networked systems. *Automatica*, vol. 59, pp. 112-119.

Liu, X.; Dou, L.; Sun, J. (2016): Consensus for networked multi-agent systems with unknown communication delays. *Journal of the Franklin Institute*, vol. 353, no. 16, pp. 4176-4190.

Li, H.; Chen, G.; Dong, Z.; Xia, D. (2016): Consensus analysis of multi-agent systems with second-order nonlinear dynamics and general directed topology: an event-triggered scheme. *Information Sciences*, vol. 370-371, pp. 598-622.

Montague, P. R.; Dayan, P.; Person, C.; Sejnowski, T. J. (1995): Bee foraging in uncertain environments using predictive hebbian learning. *Nature*, vol. 377, no. 6551, pp. 725-728.

Olfati-Saber, R.; Murray, R. M. (2003): Consensus protocols for networks of dynamic agents. *Proceedings of the American Control Conference*, pp. 951-956.

Olfati-Saber, R.; Murray, R. M. (2004): Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520-1533.

Olfati-Saber, R.; Fax, J. A.; Murray, R. M. (2007): Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215-233.

Reynolds, C. W. (1987): Flocks, herds, and schools: a distributed behavioral model. *Computer Graphics (ACM)*, vol. 21, no. 4, pp. 25-34.

Ren, W.; Beard, R. W. (2005): Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655-661.

Sahin, E. (2005): Swarm Robotics: from sources of inspiration to domains of application. *Lecture Notes in Computer Science*. pp. 10-20.

Sun, Y.; Ruan, J. (2008): Consensus problems of multi-agent systems with noise perturbation. *Chinese Physics B*, vol. 17, no. 11, pp. 4137-4141.

Trodden, P.; Richards, A. (2013): Cooperative distributed MPC of linear systems with coupled constraints. *Automatica*, vol. 49, no. 2, pp. 479-487.

Thunberg, J.; Goncalves, J.; Hu, X. (2016): Consensus and formation control on SE (3) for switching topologies. *Automatica*, vol. 66, pp. 109-121.

Vicsek, T.; Czirók, A.; Benjacob, E.; Cohen, I. I.; Shochet, O. (2006): Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, vol. 75, no. 6, pp. 1226-1229.

Zhang, H. T.; Cheng, Z.; Fan, M. C.; Wu, Y. (2008): Collective behavior coordination with predictive mechanisms. *IEEE Circuits and Systems Magazine*, vol. 8, no. 3, pp. 67-85.

Zhang, H. T.; Chen, M. Z. Q.; Stan, G. B. (2011): Fast consensus via predictive pinning control. *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 9, pp. 2247-2258.

Zhan, J.; Li, X. (2013): Consensus of sampled-data multi-agent networking systems via model predictive control. *Automatica*, vol. 49, no. 8, pp. 2502-2507.

Zhang, H.; Cheng, Z.; Chen, G.; Li, C. (2015): Model predictive flocking control for second-order multi-agent systems with input constraints. *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 6, pp. 1599-1606.

Zhong, Z.; Sun, L.; Wang, J.; Lv, P.; Zheng, H. (2015): Consensus for first- and second-order discrete-time multi-agent systems with delays based on model predictive control schemes. *Circuits Systems and Signal Processing*, vol. 34, no. 1, pp. 127-152.