Distributed Model Predictive Consensus With Self-triggered Mechanism in General Linear Multi-agent Systems

Zhan, J.; Jiang, Z.-P.; Wang, Y.; Li, X.

TR2018-182 December 29, 2018

Abstract

This paper investigates the consensus problem of general linear discrete-time multi-agent systems by using distributed model predictive control (DMPC) with self-triggered mechanism. First, a novel DMPC based consensus algorithm is proposed, where each agent only needs to obtain its neighbors' predicted state sequences once at each time step. We prove that the resultant DMPC optimization problem is feasible, and the proposed algorithm guarantees the dynamic consensus of agents. Then, to further reduce the communication cost and the energy consumption of control updates, a self-triggered DMPC based consensus algorithm is proposed with the control input and the triggering interval jointly optimized. Numerical examples including the benchmark problem with platooning vehicles are provided to verify the effectiveness and advantages of the proposed algorithms.

IEEE Transactions on Industrial Informatics

Distributed Model Predictive Consensus With Self-triggered Mechanism in General Linear Multi-agent Systems

Jingyuan Zhan, *Member, IEEE*, Zhong-Ping Jiang, *Fellow, IEEE*, Yebin Wang, *Senior Member, IEEE*, and Xiang Li, *Senior Member, IEEE*

***Abstract*—This paper investigates the consensus problem of general linear discrete-time multi-agent systems by using distributed model predictive control (DMPC) with self-triggered mechanism. First, a novel DMPC based consensus algorithm is proposed, where each agent only needs to obtain its neighbors' predicted state sequences once at each time step. We prove that the resultant DMPC optimization problem is feasible, and the proposed algorithm guarantees the dynamic consensus of agents. Then, to further reduce the communication cost and the energy consumption of control updates, a self-triggered DMPC based consensus algorithm is proposed with the control input and the triggering interval jointly optimized. Numerical examples including the benchmark problem with platooning vehicles are provided to verify the effectiveness and advantages of the proposed algorithms.**

***Index Terms*—Consensus, Multi-agent system, Distributed model predictive control, Self-triggered control.**

I. Introduction

Cooperative control of multi-agent systems has been an important area of research for decades due to its high efficiency and operational capability in completing special tasks. Among the extensive investigations, consensus, where all agents reach an agreement on certain quantities of interest, is one of the most fundamental and widely studied problems, e.g. [1]–[5]. Consensus of multi-agent systems has a wide range of industrial applications, such as intelligent transportation systems [6], [7], wireless sensor networks [8], [9], and power systems [10]–[12].

This work was supported by the National Natural Science Foundation of China (Nos. 61751303, 71731004, 61803007), the National Science Fund for Distinguished Young Scholars of China (No. 61425019), the Rail Transit Joint Funds of Beijing Natural Science Foundation and Traffic Control Technology (No. L171001), the U.S. National Science Foundation grant ECCS-1501044, and Mitsubishi Electric Research Laboratories.

Jingyuan Zhan was with the Adaptive Networks and Control Lab, Department of Electronic Engineering, the Research Center of Smart Networks and Systems, School of Information Science and Engineering, Fudan University, Shanghai 200433, and is with Beijing Key Laboratory of Transportation Engineering, College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China (e-mail: jzhan13@fudan.edu.cn).

Zhong-Ping Jiang is with Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201, USA (e-mail: zjiang@nyu.edu).

Yebin Wang is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: yebinwang@ieee.org).

Xiang Li (corresponding author) is with the Adaptive Networks and Control Lab, Department of Electronic Engineering, and with the Research Center of Smart Networks and Systems, School of Information Science and Engineering, Fudan University, Shanghai 200433, China (e-mail: lix@fudan.edu.cn).

Model predictive control (MPC), being able to treat constraints, multi-variables, and performance criteria, has become one of the most successful control strategies. Distributed MPC (DMPC) algorithms have been proposed for multi-agent systems (see [13]–[16] and the references therein). The majority of the existing DMPC algorithms consider the stabilization of a priori known set point. In contrast, consensus requires the agents to agree on a common trajectory online. Till now, there are still few results considering DMPC for consensus. Decentralized MPC based consensus schemes for first-order and second-order multi-agent systems were presented in [17] with sufficient conditions derived by exploiting the geometric properties of the optimal path. A fast consensus algorithm was proposed in [18] for the same class of multi-agent systems, where only a few pinned agents were equipped with the model predictive controllers. More recently, reference [19] proposed a DMPC based consensus algorithm for multi-agent systems with first-order dynamics, where not only the state but also the control input information need to be exchanged. As for general multi-agent systems with linear time invariant (LTI) dynamics, an iterative algorithm was proposed in [20] to reach the optimal consensus point by implementing the primal decomposition and incremental sub-gradient methods. A general DMPC framework for cooperative control of multi-agent systems was presented in [21], where the agents were required to optimize a local cost function in a sequential order. Moreover, a novel distributed receding horizon control algorithm was proposed in [22] for ensuring consensus under necessary and sufficient conditions.

Algorithms developed in [20], [21] required iterative or sequential communication and computation at each time step, which would be time-consuming and lead to heavy communication cost inevitably. Reference [22] overcame such defect, but could only achieve static consensus. Therefore, the first objective of this paper is to come up with a novel DMPC based consensus algorithm for general multi-agent systems with LTI dynamics, where iterative or sequential communication and computation at each time step can be avoided, as well as dynamic consensus remains achievable.

It is worth noting that existing DMPC based consensus algorithms [17]–[22] required each agent to solve a local optimization problem at each time step. Such a treatment would result in unnecessary communication cost and control updates. Hence, the second objective of this paper is to study the DMPC based consensus problem by introducing the self-triggered mechanism, where information transmissions and controller updates are executed at certain triggering time steps rather than at every time step. The next triggering time step is determined based on the information at the current triggering time step.

As an alternative to the periodic sampled-data control, event-triggered/self-triggered control is an effective approach in saving energy consumption, and it has gained popularity in networked control systems [23] and many industrial applications, such as smart home temperature control systems [24] and smart grids [25]. Existing research on event-triggered and self-triggered consensus problems focused on multi-agent systems with single- or double-integrator dynamics [26]–[31] and on general multi-agent systems with LTI dynamics [32]–[36]. All these results were based on continuous-time systems, while this paper considers the self-triggered distributed model predictive consensus problem of discrete-time multi-agent systems. To the best of our knowledge, results on event- or self-triggered model predictive consensus are scarce. Specifically, event-triggered MPC [37]–[39] and self-triggered MPC [40] were developed for single agent with linear or nonlinear dynamics. References [41]–[43] considered event- and self-triggered MPC of distributed agents with nonlinear dynamics. However, they merely achieved ultimate boundedness properties. This serves another motive to study the self-triggered DMPC based consensus problem in this paper.

The main contribution of this paper is two-fold. 1) We propose a novel DMPC based consensus algorithm for multi-agent systems with general LTI dynamics. The agents only need to solve their respective local optimization problem synchronously once at each time step, so as to avoid the iterative or sequential communication and computation in [20], [21]. Besides, the proposed algorithm overcomes the limitation of [22] and achieves dynamic consensus, where the final consensus state can be time-varying or even divergent. 2) We further develop a self-triggered DMPC based consensus algorithm, which effectively reduces the communication cost and the energy consumption of control updates. Each agent solves a local MPC problem to optimize not only the control input but also the triggering interval. Similar idea can be found in [40]. Nevertheless, reference [40] proposed a centralized self-triggered MPC approach only. This study partially extends the result in [40] to distributed control of multi-agent systems.

The remainder of this paper is organized as follows. Preliminaries and problem formulation are presented in Section II. A DMPC based consensus algorithm is proposed in Section III, along with the corresponding feasibility and consensus analyses. Section IV presents a self-triggered DMPC based consensus algorithm, followed by the corresponding feasibility and consensus analyses. As two potential applications, synchronization of linear oscillators and platoon of vehicles are numerically studied in Section V to demonstrate the effectiveness and the advantages of the proposed distributed model predictive consensus algorithms. Finally, Section VI concludes the whole paper.

II. Preliminaries and Problem Formulation

Mathematical notations used throughout this paper are defined as follows. Denote $\mathbb{R}$ the set of real numbers. $\mathbb{N} \triangleq \{0, 1, 2, ...\}$ and $\mathbb{N}_+$ is the set of positive integers. $\mathbb{R}^n$ denotes the set of n-dimensional real column vectors and $\mathbb{R}^{n\times m}$ the set of $n \times m$-dimensional real matrices. For $A \in \mathbb{R}^{n\times n}$, $A > 0$ means A is positive definite, A^T denotes the transpose of A, and $\|A\| = \sqrt{\max_i \lambda_i(A^TA)}$. Given a column vector x, the Euclidean norm of x is denoted by $\|x\| = (x^Tx)^{1/2}$. $\overline{m,n} \triangleq \{m, m+1, ..., n\}$ with $m \in \mathbb{N}$, $n \in \mathbb{N}$ and $m < n$. The subscript i indicates that the variable is associated with the ith agent.

Consider a multi-agent system consisting of N agents with the ith agent dynamics described by the following discrete-time equation

$$x_i(k+1) = Ax_i(k) + Bu_i(k), \tag{1}$$

where $x_i \in \mathbb{X}_i \subseteq \mathbb{R}^n$ is the state of agent i, $u_i \in \mathbb{U}_i \subseteq \mathbb{R}^m$ is the control input of agent i, and $(A, B) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m}$. The state constraint set $\mathbb{X}_i$ is assumed to be a closed set, and the input constraint set $\mathbb{U}_i$ is assumed to be a compact set containing the origin in its interior.

The communication topology of multi-agent system (1) is denoted by a digraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ with a vertex set $\mathcal{V} = \{1, 2, ..., N\}$, an edge set $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}, j \neq i\}$ and an adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$. If agent i can receive the information from agent j, then $a_{ij} = 1$; otherwise, $a_{ij} = 0$. We assume there is no self-edge in $\mathcal{G}$. The neighbors of agent i are denoted by $N_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. The digraph $\mathcal{G}$ contains a directed spanning tree if and only if there exists a root vertex such that any other vertex of the digraph can be reached by at least one path starting from the root.

Definition 1: Multi-agent system (1) with a given communication topology $\mathcal{G}$ is said to achieve consensus if and only if $\lim_{k\to+\infty} \|x_i(k) - x_j(k)\| = 0$ for all $i, j \in \{1, 2, ..., N\}$.

In this paper, we are going to design the control law $u_i(k)$ for each agent i to achieve consensus by proposing a novel DMPC method, and further propose a self-triggered DMPC based consensus algorithm. Noth that though we consider the discrete-time linear multi-agent system (1) only, the consensus problem of a continuous-time linear multi-agent system in a periodic sampled-data setting can also be solved by treating it as the consensus problem of (1) equivalently.

III. Distributed Model Predictive Consensus

In this section, we will propose a novel DMPC based consensus algorithm for multi-agent system (1) under a directed communication graph, and then present the feasibility and consensus analysis for the algorithm. We first make the following assumption.

Assumption 1: The matrix pair (A, B) in (1) is controllable, and the digraph $\mathcal{G}$ contains a directed spanning tree.

A. DMPC Based Consensus Algorithm

Let the MPC cost function for agent i at time step k be

$$\begin{aligned}&J_i\left(x_i(k), \bar{\mathbf{x}}_{-i}(k), \mathbf{u}_i(k)\right)\\&=\sum_{l=0}^{H-1}\left(\|x_i(k+l|k)-\bar{x}_{-i}(k+l|k)\|+\lambda\|u_i(k+l|k)\|\right)\\&\quad+V_i^f(k)\end{aligned} \tag{2}$$

where $H \geq 1$ is the prediction horizon, $\mathbf{u}_i(k) = [u_i^T(k|k), u_i^T(k+1|k), ..., u_i^T(k+H-1|k)]^T$ the future control vector of agent i to be determined, and $\lambda > 0$ the weight on the future control vector. Given $\mathbf{u}_i(k)$ and state at the current step k, the predicted state at $k+l$, denoted by $x_i(k+l|k)$, can be iteratively computed from the following formula

$$x_i(k+l+1|k) = Ax_i(k+l|k) + Bu_i(k+l|k).$$

With $\mathbf{u}_i$ being optimal control $\mathbf{u}_i^*$, the corresponding predicted state is also optimal and denoted by $x^*(k+l|k)$. $\bar{\mathbf{x}}_{-i}(k) = [\bar{x}_{-i}^T(k|k), \bar{x}_{-i}^T(k+1|k), ..., \bar{x}_{-i}^T(k+H|k)]^T$ denotes the averaged state trajectory of agent i's neighbors with

$$\bar{x}_{-i}(k+l|k) = \sum_{j\in N_i}\frac{x_j^a(k+l|k)}{|N_i|}.$$

$x_j^a(k+l|k)$ represents the assumed state trajectory of agent j at k, and it is obtained based on the optimal state trajectory of agent j determined at the previous time step:

$$x_j^a(k+l|k) = \begin{cases} x_j^*(k+l|k-1), l \in \overline{0, H-1};\\ Ax_j^*(k+H-1|k-1), l = H.\end{cases} \tag{3}$$

$V_i^f(k) = F_i\left(x_i(k+H|k), \bar{x}_{-i}(k+H|k)\right) = \beta_i\|x_i(k+H|k) - \bar{x}_{-i}(k+H|k)\|$ is the terminal cost with $\beta_i > 0$.

Remark 1: The term $\|x_i(k+l|k) - \bar{x}_{-i}(k+l|k)\|$ in the MPC cost function (2) differs from the consensus term $\sum_{j\in N_i} a_{ij}\|x_i(k+l|k) - x_j(k)\|_Q^2$ in [22] in two aspects: 1) the averaged state trajectory $\bar{\mathbf{x}}_{-i}(k)$ is used as the reference trajectory rather than utilizing the static neighboring states $\{x_j(k)\}_{j\in N_i}$ over the whole prediction horizon; 2) the Euclidean norm is designed instead of the commonly used quadratic function. Such differences guarantee that all agents in (1) reach dynamic consensus and facilitate the consensus analysis, respectively.

Having defined the MPC cost function (2), each agent i solves the following optimization problem $\mathcal{P}_i$:

$$\mathbf{u}_i^*(k) = \arg\min_{\mathbf{u}_i(k)} J_i\left(x_i(k), \bar{\mathbf{x}}_{-i}(k), \mathbf{u}_i(k)\right) \tag{4}$$

subject to

$$x_i(k|k) = x_i(k), \tag{5}$$

$$x_i(k+l|k) \in \mathbb{X}_i, \tag{6}$$

$$u_i(k+l|k) \in \mathbb{U}_i, \tag{7}$$

$$\begin{aligned}&\|x_i(k+l|k) - x_i^a(k+l|k)\|\\&\leq \frac{\gamma}{H-1}\min_{j\in N_i}\|x_j(k) - \bar{x}_{-j}(k|k)\|,\end{aligned} \tag{8}$$

$$x_i(k+H|k) \in \mathcal{X}_i^f(k) \tag{9}$$

for any $l \in \overline{0, H-1}$.

The novelty of problem $\mathcal{P}_i$ lies in constraints (8) and (9) with respect to the state prediction sequence, where $\gamma \in (0,1)$. (8) enforces a degree of consistency between what an agent plans to do and what neighbors believe the agent will do. The terminal region $\mathcal{X}_i^f(k)$ in (9) is defined as

$$\begin{aligned}\mathcal{X}_i^f(k) \triangleq \{x \in \mathbb{X}_i | \|x - x_i^a(k+H|k)\|\\ \leq (1-\gamma)v/\beta_i \min_{j\in N_i}\|x_j(k) - \bar{x}_{-j}(k|k)\|\}\end{aligned}$$

with $v \in (0,1)$. As a crucial element to establish stability, we make an assumption with respect to the terminal region and the terminal cost hereinafter. For the sake of notational simplicity, we first denote $b_i^l \triangleq x_i^*(k+l|k)$ and $\Delta b_i^l \triangleq b_i^l - \sum_{j\in N_i} b_j^l/|N_i|, l = 1, H$ for all $i \in \overline{1, N}$.

Assumption 2: For an arbitrary time step k, there exists an auxiliary local controller $\bar{u}_i = \kappa_i\left(b_i^H, \{b_j^H\}_{j\in N_i}\right) \in \mathbb{U}_i$ such that

$$Ab_i^H + B\bar{u}_i \in \mathcal{X}_i^f(k+1), \tag{10}$$

and

$$\begin{aligned}&F_i\left(Ab_i^H + B\bar{u}_i, \sum_{j\in N_i}\frac{Ab_j^H}{|N_i|}\right) - F_i\left(b_i^H, \sum_{j\in N_i}\frac{b_j^H}{|N_i|}\right)\\&\leq -\left\|\Delta b_i^H\right\| - \lambda\|\bar{u}_i\|.\end{aligned} \tag{11}$$

In the following proposition, we design a linear controller of $\kappa_i\left(b_i^H, \{b_j^H\}_{j\in N_i}\right)$, and derive sufficient conditions on the controller and β_i to ensure Assumption 2.

Proposition 1: Let the auxiliary local controller be $\bar{u}_i = K_i\Delta b_i^H \in \mathbb{U}_i$. If

$$\|BK_i\Delta b_i^H\| \leq \frac{(1-\gamma)v}{\beta_i}\min_{j\in N_i}\|\Delta b_j^1\| \tag{12}$$

and

$$\beta_i\left(\|A + BK_i\| - 1\right) + 1 + \lambda\|K_i\| < 0, \tag{13}$$

then Assumption 2 holds.

Proof: We easily obtain

$$F_i\left(Ab_i^H + B\bar{u}_i, \sum_{j\in N_i}\frac{Ab_j^H}{|N_i|}\right) = \beta_i\|(A+BK_i)\Delta b_i^H\|.$$

Inequality (12) directly implies that (10) in Assumption 2 holds. Then we have

$$\begin{aligned}&F_i\left(Ab_i^H + B\bar{u}_i, \sum_{j\in N_i}\frac{Ab_j^H}{|N_i|}\right) - F_i\left(b_i^H, \sum_{j\in N_i}\frac{b_j^H}{|N_i|}\right)\\&\quad+\left\|\Delta b_i^H\right\| + \lambda\|\bar{u}_i\|\\&=\beta_i\|(A+BK_i)\Delta b_i^H\| - \beta_i\left\|\Delta b_i^H\right\| + \left\|\Delta b_i^H\right\| + \lambda\|\bar{u}_i\|\\&\leq\left(\beta_i\|A+BK_i\| - \beta_i + 1 + \lambda\|K_i\|\right)\left\|\Delta b_i^H\right\|\\&\leq 0.\end{aligned}$$

The first inequality follows from the compatibility of vector norms, and the last inequality from (13). Till now, inequality (11) in Assumption 2 is proved. ■

Remark 2: According to inequalities (12)-(13), K_i and β_i depend on time varying Δb_i^H and $\Delta b_j^1, j \in N_i$, which requires the design of K_i and β_i at each time step. However, it's inefficient in real implementation. Considering the fact that $\|\Delta b_i^H\| \ll \|\Delta b_j^1\|, \forall j \in N_i$, inequality (12) holds itself, and we can always find a sufficiently large β_i satisfying inequality (13) for a given λ if $\|A + BK_i\| < 1$.

The DMPC based consensus algorithm is specified as follows.

Algorithm 1 (DMPC based consensus algorithm):

1) Initialization: Set $k = 0$. Each agent i transmits its state sequence $\{x_i^a(l|0)\}_{l=0}^H$ with $x_i^a(l|0) = A^l x_i(0)$ to all $j \in N_i$, and then receives $\{x_j^a(l|0)\}_{l=0}^H$ from all $j \in N_i$. Each agent i solves problem $\mathcal{P}_i$ by removing constraints (8)-(9) to obtain $\mathbf{u}_i^*(0)$. Go to Step 3).
2) Each agent i solves problem $\mathcal{P}_i$ to obtain $\mathbf{u}_i^*(k)$.
3) Each agent i applies $u_i(k) = u_i^*(k|k)$.
4) Each agent i transmits the optimal state trajectory $\{x_i^*(k+l|k)\}_{l=1}^H$ to all $j \in N_i$, and then receives $\{x_j^*(k+l|k)\}_{l=1}^H$ from all $j \in N_i$.
5) By using $\bar{x}_{-i}(k+1|k+1) = \sum_{j \in N_i} x_j^*(k+1|k)/|N_i|$, each agent i transmits $\|x_i^*(k+1|k) - \bar{x}_{-i}(k+1|k+1)\|$ to all $j \in N_i$, and then receives $\|x_j^*(k+1|k) - \bar{x}_{-j}(k+1|k+1)\|$ from all $j \in N_i$.
6) Set $k = k+1$, and go to Step 2).

Note that at the initialization step of Algorithm 1, each agent i solves problem $\mathcal{P}_i$ without constraints (8)-(9), by assuming that every neighbor applies zero control over the prediction horizon. A similar idea can be found in [14]. When $k \geq 1$, constraints (8)-(9) are enforced. For an arbitrary agent i without incoming neighbors, we let $u_i^*(k+l|k) = 0$ directly for all $l \in \overline{0, H-1}$ and all k.

Remark 3: A key feature of the DMPC based consensus algorithm is that each agent only needs to solve problem $\mathcal{P}_i$ relying on the information of its own and its neighbors. Problem $\mathcal{P}_i$ can be reformulated as a standard form of second-order cone programs (SOCPs), which have been widely studied and readily solved by the interior-point method (see Chapter 11 of [44]) with good theoretical convergence properties and efficient computational performance. Furthermore, numerous recent contributions address the real-time implementation of interior-point method solvers for SOCP on multiple platforms, such as an embedded conic solver (ECOS) proposed in [45] to solve SOCP with hundreds of decision variables within 10 msec. To better apply the state-of-the-art SOCP solvers in the DMPC based consensus algorithm, we will mainly focus on the reformulation of problem $\mathcal{P}_i$ as an SOCP in this paper.

In the remainder part of this subsection, we show how to reformulate problem $\mathcal{P}_i$ as a standard form of SOCPs. Problem $\mathcal{P}_i$ is a constrained problem of minimizing a sum of Euclidean norms, which can be recast in the following canonical form

$$\min_{\mathbf{u}_i \in \mathbb{R}^{Hm}} \sum_{l=1}^{2H} \|C_l \mathbf{u}_i + d_l\| \tag{14}$$

subject to

$$\|E_l \mathbf{u}_i + f_l\| \leq g_l, l = 1, 2, ..., (n+m+1)H, \tag{15}$$

where $\|C_l \mathbf{u}_i + d_l\|$ with $l = 1, ..., H$ and $l = H+1, ..., 2H$ correspond to the terms w.r.t. x and u respectively in (2) such that $C_l \in \mathbb{R}^{n \times Hm}, d_l \in \mathbb{R}^n$ for $l = 1, ..., H$ and $C_l \in \mathbb{R}^{m \times Hm}, d_l \in \mathbb{R}^m$ for $l = H+1, ..., 2H$, $E_l \in \mathbb{R}^{n_l \times Hm}$, $f_l \in \mathbb{R}^{n_l}$, and $g_l \in \mathbb{R}$. Constraints (6) and (7) can be easily recast in nH and mH inequalities of (15) respectively, and constraints (8)-(9) are recast in H inequalities of (15). Note that the objective function is not differentiable at any point $\mathbf{u}_i$ when $C_l \mathbf{u}_i + d_l = 0$. Then we transform problem (14)-(15) into the following SOCP by introducing new variable $t \in \mathbb{R}^{2H}$:

$$\min_{\mathbf{u}_i, t} \mathbf{1}^T t \tag{16}$$

subject to

$$\|C_l \mathbf{u}_i + d_l\| \leq e_l^T t, l = 1, ..., 2H \tag{17}$$

$$\|C_l \mathbf{u}_i + f_l\| \leq g_l, l = 1, 2, ..., (n+m+1)H, \tag{18}$$

where $\mathbf{1}$ is a column vector with all entries equal to 1, and $e_l \in \mathbb{R}^{2H}$ is with all zero entries except the l-th equal to 1. By putting $\mathbf{u}_i$ and t in a concatenated vector, we can easily write problem (16)-(18) into a standard form of SOCPs (see Chapter 4 of [44]) such that the interior-point method can be applied.

B. Feasibility and Consensus Analysis

Before the consensus analysis of Algorithm 1, we prove its iterative feasibility by the induction principle in the following lemma.

Lemma 1: For each agent i in multi-agent system (1), if problem $\mathcal{P}_i$ is feasible at time step k, then it is feasible at time step $k+1$ for all $k \geq 0$.

Proof: Define

$$\bar{\mathbf{u}}_i(k+1) = [u_i^{*T}(k+1|k), ..., u_i^{*T}(k+H-1|k), \bar{u}_i^T]^T \tag{19}$$

with $\bar{u}_i = \kappa_i \left(b_i^H, \{b_j^H\}_{j \in N_i}\right)$ satisfying Assumption 2. Then $x_i(k+l|k+1)$ rendered by $\bar{\mathbf{u}}_i(k+1)$ is equal to $x_i^*(k+l|k)$ for all $l \in \overline{1, H}$ such that (6)-(8) is easily fulfilled, and (9) is also fulfilled due to (10). Then we conclude that $\bar{\mathbf{u}}_i(k+1)$ is a feasible solution of problem $\mathcal{P}_i$ at time step $k+1$. ■

Let $x = [x_1^T, x_2^T, ..., x_N^T]^T$, and define $\mathcal{X}_0 \subseteq \mathbb{R}^{nN}$ as the set of all states for which a feasible solution can be found in step 1) of Algorithm 1. Then according to Lemma 1, we are ready to present the feasibility result in the following theorem whose proof is omitted for want of space.

Theorem 1: The DMPC based consensus algorithm (Algorithm 1) is feasible if the initial state $x(0) \in \mathcal{X}_0$ and Assumption 2 holds.

We are now in a position to state the main result.

Theorem 2: Consider multi-agent system (1) with communication topology $\mathcal{G}$. Under Algorithm 1, system (1) reaches consensus asymptotically if the initial state $x(0) \in \mathcal{X}_0$ and Assumptions 1-2 hold.

Proof: Denote

$$J_i^*(k) = \min_{\mathbf{u}_i(k)} J_i(x_i(k), \bar{\mathbf{x}}_{-i}(k), \mathbf{u}_i(k)),$$

and

$$\bar{J}_i(k+1) = J_i(x_i(k+1), \bar{\mathbf{x}}_{-i}(k+1), \bar{\mathbf{u}}_i(k+1))$$

with $\bar{\mathbf{u}}_i(k+1)$ defined in (19). Following the arguments in the proof of Lemma 1, $\bar{\mathbf{u}}_i(k+1)$ is a feasible solution of problem $\mathcal{P}_i$ at time step $k+1$. Then we have

$$\begin{aligned}&J_i^*(k+1)-J_i^*(k)\\ \leq&\bar{J}_i(k+1)-J_i^*(k)\\ =&\sum_{l=1}^{H-1}(\|x_i^*(k+l|k)-\bar{x}_{-i}(k+l|k+1)\|\\ &\quad-\|x_i^*(k+l|k)-\bar{x}_{-i}(k+l|k)\|)\\ &-\|x_i(k)-\bar{x}_{-i}(k|k)\|-\lambda\|u_i^*(k|k)\|\\ &+\|x_i^*(k+H|k)-\bar{x}_{-i}(k+H|k+1)\|\\ &+\lambda\|\bar{u}_i\|+V_i^f(k+1)-V_i^f(k)\end{aligned} \tag{20}$$

Therein,

$$\begin{aligned}&V_i^f(k+1)-V_i^f(k)\\ \leq&F_i\left(Ab_i^H+B\bar{u}_i,\sum_{j\in N_i}\frac{Ab_j^H}{|N_i|}\right)-F_i\left(b_i^H,\sum_{j\in N_i}\frac{b_j^H}{|N_i|}\right)\\ &+\beta_i\left\|\sum_{j\in N_i}\frac{x_j^*(k+H|k)-x_j^a(k+H|k)}{|N_i|}\right\|\\ \leq&-\|x_i^*(k+H|k)-\bar{x}_{-i}(k+H|k+1)\|-\lambda\|\bar{u}_i\|\\ &+\beta_i\left\|\sum_{j\in N_i}\frac{x_j^*(k+H|k)-x_j^a(k+H|k)}{|N_i|}\right\|\\ \leq&-\|x_i^*(k+H|k)-\bar{x}_{-i}(k+H|k+1)\|\\ &-\lambda\|\bar{u}_i\|+(1-\gamma)v\|x_i(k)-\bar{x}_{-i}(k|k)\|,\end{aligned} \tag{21}$$

the first inequality of which follows from the triangle inequality of vector norms, the second inequality from (11), and the last inequality from (9).

For $l\in\overline{1,H-1}$,

$$\begin{aligned}&\|x_i^*(k+l|k)-\bar{x}_{-i}(k+l|k+1)\|\\ &-\|x_i^*(k+l|k)-\bar{x}_{-i}(k+l|k)\|\\ \leq&\|\bar{x}_{-i}(k+l|k+1)-\bar{x}_{-i}(k+l|k)\|\\ =&\left\|\sum_{j\in N_i}\frac{x_j^*(k+l|k)-x_j^a(k+l|k)}{|N_i|}\right\|\\ \leq&\frac{\gamma}{H-1}\|x_i(k)-\bar{x}_{-i}(k|k)\|.\end{aligned}$$

The first inequality follows from the triangle inequality, and the last inequality from (8). Therefore,

$$\begin{aligned}J_i^*(k+1)-J_i^*(k)\leq&-(1-\gamma)(1-v)\|x_i(k)-\bar{x}_{-i}(k|k)\|\\ &-\lambda\|u_i^*(k|k)\|,\end{aligned}$$

which, combined with $J_i^*(k)\geq 0$ for any k, gives that $\lim_{k\to\infty}\|x_i(k)-\bar{x}_{-i}(k|k)\|=0$ and $\lim_{k\to\infty}\|u_i^*(k|k)\|=0$ for all i according to LaSalle's invariance principle [46]. Since $\bar{x}_{-i}(k|k)=\sum_{j\in N_i}x_j^*(k|k-1)/|N_i|=\sum_{j\in N_i}x_j(k)/|N_i|$, we obtain $\lim_{k\to\infty}\|x_i(k)-\sum_{j\in N_i}x_j(k)/|N_i|\|=0$ for all i, which further implies that consensus is reached when $\mathcal{G}$ contains a directed spanning tree. ■

IV. Distributed Model Predictive Consensus With Self-triggered Mechanism

This section investigates the distributed model predictive consensus problem with self-triggered mechanism in order to reduce communication cost and energy consumption of control updates. We propose a self-triggered DMPC based consensus algorithm with control inputs and triggering time steps jointly determined.

A. Self-triggered DMPC Based Consensus Algorithm

Let k_l^i denote the lth triggering time step of agent i with $l\in\overline{0,+\infty}$. The triggering interval $h_l^i=k_{l+1}^i-k_l^i$ satisfies $h_l^i\in\overline{1,p}$ with $p\geq 1$ denoting the maximum allowable interval. Let the MPC cost function for agent i at triggering time step k_l^i be defined by

$$\begin{aligned}&J_i(x_i(k_l^i),\bar{\mathbf{x}}_{-i}(k_l^i),\mathbf{u}_i(k_l^i),h_l^i)\\ =&\|x_i(k_l^i|k_l^i)-\bar{x}_{-i}(k_l^i|k_l^i)\|+\lambda\|u_i(k_l^i|k_l^i)\|\\ &+\sum_{q=0}^{r-1}\big(\|x_i(k_l^i+h_l^i+q\cdot p|k_l^i)-\bar{x}_{-i}(k_l^i+h_l^i+q\cdot p|k_l^i)\|\\ &+\lambda\|u_i(k_l^i+h_l^i+q\cdot p|k_l^i)\|\big)+V_i^f(k_l^i)+\frac{\alpha}{h_l^i}\end{aligned} \tag{22}$$

where $r\in\mathbb{N}_+$, $h_l^i+r\cdot p$ is the prediction horizon denoted by H_l^i, and $\mathbf{u}_i(k_l^i)=[u_i^T(k_l^i|k_l^i),...,u_i^T(k_l^i+H_l^i-1|k_l^i)]^T$ is the future control vector of agent i to be determined. Control $\mathbf{u}_i$ is piecewise, i.e.,

$$\begin{cases}u_i(k_l^i|k_l^i)=u_i(k_l^i+1|k_l^i)=...=u_i(k_l^i+h_l^i-1|k_l^i);\\ u_i(k_l^i+h_l^i+j\cdot p|k_l^i)=u_i(k_l^i+h_l^i+j\cdot p+1|k_l^i)=...\\ \quad=u_i(k_l^i+h_l^i+j\cdot p+p-1|k_l^i),j\in\overline{0,r-1}\end{cases} \tag{23}$$

which is also illustrated in Fig. 1. $x_i(k_l^i+q|k_l^i)$ denotes agent i's state prediction of future step k_l^i+q at time step k_l^i, $x_i(k_l^i+q+1|k_l^i)=Ax_i(k_l^i+q|k_l^i)+Bu_i(k_l^i+q|k_l^i)$, $\bar{\mathbf{x}}_{-i}(k_l^i)=[\bar{x}_{-i}^T(k_l^i|k_l^i),...,\bar{x}_{-i}^T(k_l^i+(r+2)p|k_l^i)]^T$ denotes the averaged state trajectory of agent i's neighbors with

$$\bar{x}_{-i}(k_l^i+q|k_l^i)=\sum_{j\in N_i}\frac{x_j^a(k_l^i+q|k_l^i)}{|N_i|}. \tag{24}$$

$x_j^a(k_l^i+q|k_l^i)$ represents the assumed state trajectory of agent j, $j\in N_i\cup\{i\}$ by agent i at k_l^i, and it is obtained based on the optimal state trajectory of agent j determined at its latest triggering time step $\Gamma_j(k_l^i)<k_l^i$:

$$x_j^a(k_l^i+q|k_l^i)=\begin{cases}x_j^*(k_l^i+q|\Gamma_j(k_l^i)),q\in\overline{0,r\cdot p};\\ Ax_j^*(k_l^i+q-1|\Gamma_j(k_l^i)),q\in\overline{r\cdot p+1,(r+2)p}.\end{cases} \tag{25}$$

$V_i^f(k_l^i)=F_i(x_i(k_l^i+H_l^i|k_l^i),\bar{x}_{-i}(k_l^i+H_l^i|k_l^i))=\beta_i\|x_i(k_l^i+H_l^i|k_l^i)-\bar{x}_{-i}(k_l^i+H_l^i|k_l^i)\|$ is the terminal cost with $\beta_i>0$. $\alpha\in\mathbb{R}^+$ is a design variable that is used to trade off the cost of sampling against the cost of control. Similar idea is employed in [40].

Denote $B^{(p)}=\sum_{q=0}^{p-1}A^qB$ and $A^{(p)}\triangleq A^p+B^{(p)}K_i$. Then we make the following assumption.

Assumption 3: The matrix pair $(A^p,B^{(p)})$ is controllable, and the digraph $\mathcal{G}$ contains a directed spanning tree.

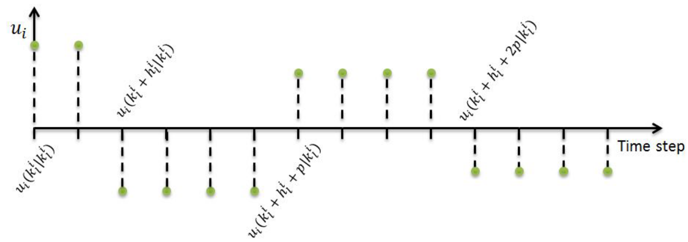

Fig. 1. Illustration of the control sequence $\{u_i(k_l^i+q|k_l^i)\}_{q=0}^{H_l^i-1}$ of agent i at triggering time step k_l^i with $h_l^i=2$, $p=4$, $r=3$ and $H_l^i=14$.

Given the MPC cost function (22), each agent solves the following optimization problem $\mathcal{SP}_i$:

$$\{\mathbf{u}_i^*(k_l^i), h_l^{i*}\} = \arg\min_{\mathbf{u}_i(k_l^i),h_l^i} J_i\left(x_i(k_l^i), \bar{\mathbf{x}}_{-i}(k_l^i), \mathbf{u}_i(k_l^i), h_l^i\right) \tag{26}$$

subject to (23) and

$$x_i(k_l^i|k_l^i) = x_i(k_l^i), \tag{27}$$

$$x_i(k_l^i|k_l^i), x_i(k_l^i+h_l^i+q\cdot p|k_l^i) \in \mathbb{X}_i, \tag{28}$$

$$u_i(k_l^i|k_l^i), u_i(k_l^i+h_l^i+q\cdot p|k_l^i) \in \mathbb{U}_i, \tag{29}$$

$$\begin{aligned}&\|x_i(k_l^i+h_l^i+q\cdot p|k_l^i) - x_i^a(k_l^i+h_l^i+q\cdot p|k_l^i)\|\\ &\leq \frac{\gamma}{r}\min_{j\in N_i}\|x_j(\Gamma_j(k_l^i)) - \bar{x}_{-j}(\Gamma_j(k_l^i)|\Gamma_j(k_l^i))\|,\end{aligned} \tag{30}$$

$$x_i(k_l^i+H_l^i|k_l^i) \in \mathcal{X}_i^f(k_l^i) \tag{31}$$

for any $q \in \overline{0, r-1}$.

The terminal region $\mathcal{X}_i^f(k_l^i)$ in (31) is defined as

$$\begin{aligned}\mathcal{X}_i^f(k_l^i) \triangleq \{&x\in\mathbb{X}_i|\|x - x_i^a(k_l^i+H_l^i|k_l^i)\|\\ &\leq (1-\gamma)v/\beta_i\min_{j\in N_i}\|x_j(\Gamma_j(k_l^i)) - \bar{x}_{-j}(\Gamma_j(k_l^i)|\Gamma_j(k_l^i))\|\}\end{aligned}$$

with $\gamma\in(0,1)$ and $v\in(0,1)$. Before giving the following assumption with respect to $V_i^f(k_l^i)$ and $\mathcal{X}_i^f(k_l^i)$, we first denote $b_{ii} \triangleq x_i^*(k_l^i+H_l^i|k_l^i), b_{ij} \triangleq x_j^a(k_l^i+H_l^i|k_{l+1}^i), j\in N_i$, $\Delta b_i \triangleq b_{ii} - \sum_{j\in N_i} b_{ij}/|N_i|$ and $c_j \triangleq x_j(\Gamma_j(k_{l+1}^i)) - \bar{x}_{-j}(\Gamma_j(k_{l+1}^i)|\Gamma_j(k_{l+1}^i))$ for all $i\in\overline{1,N}$ and all $j\in N_i$.

Assumption 4: For an arbitrary triggering time step k_l^i, there exists an auxiliary local controller $\bar{u}_i = \kappa_i\left(b_{ii}, \{b_{ij}\}_{j\in N_i}\right) \in \mathbb{U}_i$ such that

$$A^p b_{ii} + B^{(p)}\bar{u}_i \in \mathcal{X}_i^f(k_{l+1}^i), \tag{32}$$

and

$$\begin{aligned}&F_i\left(A^p b_{ii} + B^{(p)}\bar{u}_i, \sum_{j\in N_i}\frac{A^p b_{ij}}{|N_i|}\right) - F_i\left(b_{ii}, \sum_{j\in N_i}\frac{b_{ij}}{|N_i|}\right)\\ &\leq -\|\Delta b_i\| - \lambda\|\bar{u}_i\|.\end{aligned} \tag{33}$$

Similar as Proposition 1, we design a linear controller of $\kappa_i\left(b_{ii}, \{b_{ij}\}_{j\in N_i}\right)$ in the following proposition to validate Assumption 4.

Proposition 2: Let the auxiliary local controller be $\bar{u}_i = K_i\Delta b_i \in \mathbb{U}_i$. If

$$\|B^{(p)}K_i\Delta b_i\| \leq \frac{(1-\gamma)v}{\beta_i}\min_{j\in N_i}\|c_j\| \tag{34}$$

and

$$\beta_i\left(\|A^{(p)}\| - 1\right) + 1 + \lambda\|K_i\| < 0, \tag{35}$$

then Assumption 4 holds.

Proof: Inequality (34) directly implies that (32) in Assumption 4 holds. We easily obtain that $A^p b_{ii} + B^{(p)}u_i - \sum_{j\in N_i} A^p b_{ij}/|N_i| = A^{(p)}\Delta b_i$. Then

$$\begin{aligned}&F_i\left(A^p b_{ii} + B^{(p)}\bar{u}_i, \sum_{j\in N_i}\frac{A^p b_{ij}}{|N_i|}\right) - F_i\left(b_{ii}, \sum_{j\in N_i}\frac{b_{ij}}{|N_i|}\right)\\ &\quad + \|\Delta b_i\| + \lambda\|\bar{u}_i\|\\ &\leq \beta_i\|A^{(p)}\Delta b_i\| - \beta_i\|\Delta b_i\| + \|\Delta b_i\| + \lambda\|\bar{u}_i\|\\ &\leq \left(\beta_i\|A^{(p)}\| - \beta_i + 1 + \lambda\|K_i\|\right)\|\Delta b_i\|\\ &\leq 0.\end{aligned}$$

The last inequality follows from (35). Till now, inequality (33) is proved. ■

Remark 4: According to inequalities (34)-(35), K_i and β_i depend on time varying Δb_i and $c_j, j\in N_i$, which requires the design of K_i and β_i at each triggering time step. However, it's inefficient in real implementation. Considering the fact that $\|\Delta b_i\| \ll \|c_j\|, \forall j\in N_i$, inequality (34) holds itself, and we can always find a sufficiently large β_i satisfying inequality (35) for a given λ if $\|A^p + B^{(p)}K_i\| < 1$.

The self-triggered DMPC based consensus algorithm for agent i is specified as follows.

Algorithm 2 (self-triggered DMPC based consensus algorithm for agent i):

1) Initialization: Each agent i set $k=0$ as the first triggering time step, i.e., $k_l^i=0$ with $l=0$. Each agent i transmits its state sequence $\{x_i^a(q|0)\}_{q=0}^{p+r\cdot p}$ with $x_i^a(q|0) = A^q x_i(0)$ to all $j\in N_i$, and then receives $\{x_j^a(q|0)\}_{q=0}^{p+r\cdot p}$ from all $j\in N_i$. Each agent i solves problem $\mathcal{SP}_i$ by removing constraints (30)-(31) to obtain h_l^{i*} and $\mathbf{u}_i^*(0)$. Go to Step 3).
2) Agent i solves problem $\mathcal{SP}_i$ to obtain h_l^{i*} and $\mathbf{u}_i^*(k_l^i)$.
3) Agent i transmits $\|x_i(k_l^i) - \bar{x}_{-i}(k_l^i|k_l^i)\|$, $\{x_j^a(k_l^i+q|k_l^i)\}_{q=0}^{(r+2)p}$ and the optimal state trajectory $\{x_i^*(k_l^i+q|k_l^i)\}_{q=1}^{H_l^i+p}$ to all $j\in N_i$.
4) Agent i applies $u_i(k) = u_i^*(k_l^i|k_l^i)$.
5) Set $k=k+1$, and check whether $k = k_l^i + h_l^{i*}$ or not. If $k = k_l^i + h_l^{i*}$, set $l = l+1$ and go to Step 2); otherwise, go to step 4).

Similar as Algorithm 1, we let $u_i^*(k_l^i+l|k_l^i) = 0, \forall l\in\overline{0, r\cdot p+p-1}, \forall k$ directly for any arbitrary agent i without incoming neighbors.

Remark 5: To solve problem $\mathcal{SP}_i$, we may solve

$$\min_{\mathbf{u}_i(k_l^i)} J_i(x_i(k_l^i), \bar{\mathbf{x}}_{-i}(k_l^i), \mathbf{u}_i(k_l^i), h_l^i) \tag{36}$$

subject to (23) and (27)-(31), by assuming $h_l^i = 1,2,...,p$, and then obtain h_l^{i*} which gives the lowest value of the cost. The computation method to solve (36) is the same as that to solve problem $\mathcal{P}_i$, which has been described in Remark 3. From this point of view, the self-triggered DMPC based consensus algorithm reduces the communication and control updating times at the cost of increased computational complexity. It would be still interesting to develop more efficient methods to solve problem $\mathcal{SP}_i$ in our future research. A possible alternative is to establish the connection between problem $\mathcal{SP}_i$ and mixed-integer second-order cone program (MISOCP), which is readily solved by combining existing solution methods for SOCPs with extensions of mixed-integer linear or nonlinear programming methods (see [47]).

B. Feasibility and Consensus Analysis

Before presenting the consensus analysis of the self-triggered DMPC based consensus algorithm (Algorithm 2), we prove its iterative feasibility by the induction principle in the following lemma.

Lemma 2: For each agent i in multi-agent system (1) under Assumption 4, if Problem $\mathcal{SP}_i$ is feasible at time step k_l^i, then it is feasible at time step k_{l+1}^i for all $l \geq 0$.

Proof: Define

$$\begin{aligned}\bar{\mathbf{u}}_i(k_{l+1}^i) =&[u_i^{*T}(k_{l+1}^i|k_l^i), u_i^{*T}(k_{l+1}^i+1|k_l^i), ...,\\ &u_i^{*T}(k_{l+1}^i+r\cdot p-1|k_l^i), \bar{u}_i^T(k_{l+1}^i+r\cdot p|k_{l+1}^i), ...,\\ &\bar{u}_i^T(k_{l+1}^i+r\cdot p+p-1|k_{l+1}^i)]^T\end{aligned} \tag{37}$$

with $\bar{u}_i(k_{l+1}^i+q|k_{l+1}^i) = \kappa_i(b_{ii}, \{b_{ij}\}_{j\in N_i})$ satisfying Assumption 4 for any $q \in \overline{r\cdot p, r\cdot p+p-1}$. Then $x_i(k_{l+1}^i+h_l^i+q\cdot p|k_{l+1}^i)$ rendered by $\bar{\mathbf{u}}_i(k_{l+1}^i)$ is equal to $x_i^*(k_{l+1}^i+h_l^i+q\cdot p|k_l^i)$ for all $q \in \overline{0,r}$. Constraints (23) and (28)-(29) are easily fulfilled. We also obtain $x_i^a(k_{l+1}^i+h_l^i+q\cdot p|k_{l+1}^i) = x_i^*(k_{l+1}^i+h_l^i+q\cdot p|k_l^i)$ such that (30) with $q \in \overline{0,r-1}$ is fulfilled. Besides, (32) directly gives that $x_i(k_{l+1}^i+r\cdot p+p|k_{l+1}^i)$ rendered by $\bar{\mathbf{u}}_i(k_{l+1}^i)$ satisfies (31). Then we conclude that $\{\bar{\mathbf{u}}_i(k_{l+1}^i), p\}$ is a feasible solution of problem $\mathcal{SP}_i$ at time step k_{l+1}^i. ∎

Define $\mathcal{X}_0' \subseteq \mathbb{R}^{nN}$ as the set of all states for which a feasible solution can be found in step 1) of Algorithm 2. Then according to Lemma 2, we are ready to present the feasibility result in the following theorem with proof omitted.

Theorem 3: The self-triggered DMPC based consensus algorithm (Algorithm 2) is feasible if the initial state $x(0) \in \mathcal{X}_0'$ and Assumption 4 holds.

The main result concerning the self-triggered DMPC based consensus algorithm is carried out in the following theorem.

Theorem 4: Consider multi-agent system (1) with communication topology $\mathcal{G}$. Under Algorithm 2, system (1) reaches consensus asymptotically with $\lim_{l\to\infty} h_l^i = p$ if the initial state $x(0) \in \mathcal{X}_0'$ and Assumptions 3-4 hold.

Proof: Denote

$$J_i^*(k_l^i) = \min_{\mathbf{u}_i(k_l^i), h_l^i} J_i(x_i(k_l^i), \bar{\mathbf{x}}_{-i}(k_l^i), \mathbf{u}_i(k_l^i), h_l^i),$$

and

$$\bar{J}_i(k_{l+1}^i) = J_i(x_i(k_{l+1}^i), \bar{\mathbf{x}}_{-i}(k_{l+1}^i), \bar{\mathbf{u}}_i(k_{l+1}^i), p)$$

with $\bar{\mathbf{u}}_i(k_{l+1}^i)$ defined in (37). Following the arguments in the proof of Lemma 2, $\{\bar{\mathbf{u}}_i(k_{l+1}^i), p\}$ is a feasible solution of problem $\mathcal{SP}_i$ at time step k_{l+1}^i.

$$\begin{aligned}
&J_i^*(k_{l+1}^i) - J_i^*(k_l^i)\\
\leq\ & \bar{J}_i(k_{l+1}^i) - J_i^*(k_l^i)\\
=\ & \sum_{q=0}^{r-1}\Big(\|x_i^*(k_l^i+h_l^i+q\cdot p|k_l^i) - \bar{x}_{-i}(k_l^i+h_l^i+q\cdot p|k_{l+1}^i)\|\\
&\quad -\|x_i^*(k_l^i+h_l^i+q\cdot p|k_l^i) - \bar{x}_{-i}(k_l^i+h_l^i+q\cdot p|k_l^i)\|\Big)\\
&-\|x_i^*(k_l^i|k_l^i) - \bar{x}_{-i}(k_l^i|k_l^i)\| + \lambda\|u_i^*(k_l^i|k_l^i)\|\\
&+\|x_i(k_l^i+H_l^i|k_{l+1}^i) - \bar{x}_{-i}(k_l^i+H_l^i|k_{l+1}^i)\|\\
&+\lambda\|u_i(k_l^i+H_l^i|k_{l+1}^i)\| + V_i^f(k_{l+1}^i) - V_i^f(k_l^i)\\
&+\frac{\alpha}{p} - \frac{\alpha}{h_l^i}
\end{aligned} \tag{38}$$

Therein,

$$\begin{aligned}
&V_i^f(k_{l+1}^i) - V_i^f(k_l^i)\\
\leq& F_i\left(A^p b_{ii} + B^{(p)}\bar{u}_i, \sum_{j\in N_i}\frac{A^p b_{ij}}{|N_i|}\right) - F_i\left(b_{ii}, \sum_{j\in N_i}\frac{b_{ij}}{|N_i|}\right)\\
&+\beta_i\left\|\sum_{j\in N_i}\frac{x_j^a(k_l^i+H_l^i|k_{l+1}^i) - x_j^a(k_l^i+H_l^i|k_l^i)}{|N_i|}\right\|\\
\leq& -\|x_i(k_l^i+H_l^i|k_{l+1}^i) - \bar{x}_{-i}(k_l^i+H_l^i|k_{l+1}^i)\|\\
&\quad -\lambda\|u_i(k_l^i+H_l^i|k_{l+1}^i)\|\\
&+\beta_i\left\|\sum_{j\in N_i}\frac{x_j^a(k_l^i+H_l^i|k_{l+1}^i) - x_j^a(k_l^i+H_l^i|k_l^i)}{|N_i|}\right\|\\
\leq& -\|x_i(k_l^i+H_l^i|k_{l+1}^i) - \bar{x}_{-i}(k_l^i+H_l^i|k_{l+1}^i)\|\\
&-\lambda\|u_i(k_l^i+H_l^i|k_{l+1}^i)\| + (1-\gamma)v\|x_i(k_l^i) - \bar{x}_{-i}(k_l^i|k_l^i)\|,
\end{aligned}$$

the first inequality of which follows from the triangle inequality of vector norms, the second inequality from (33), and the last inequality from (31).

For any $q \in \overline{0,r-1}$, it holds:

$$\begin{aligned}
&\|x_i^*(k_l^i+h_l^i+q\cdot p|k_l^i) - \bar{x}_{-i}(k_l^i+h_l^i+q\cdot p|k_{l+1}^i)\|\\
&-\|x_i^*(k_l^i+h_l^i+q\cdot p|k_l^i) - \bar{x}_{-i}(k_l^i+h_l^i+q\cdot p|k_l^i)\|\\
\leq&\|\bar{x}_{-i}(k_l^i+h_l^i+q\cdot p|k_{l+1}^i) - \bar{x}_{-i}(k_l^i+h_l^i+q\cdot p|k_l^i)\|\\
=&\left\|\sum_{j\in N_i}\frac{x_j^a(k_l^i+h_l^i+q\cdot p|k_{l+1}^i) - x_j^a(k_l^i+h_l^i+q\cdot p|k_l^i)}{|N_i|}\right\|\\
\leq&\frac{\gamma}{r}\|x_i(k_l^i) - \bar{x}_{-i}(k_l^i|k_l^i)\|,
\end{aligned}$$

where the last inequality is due to (30). Therefore,

$$\begin{aligned}
&J_i^*(k_{l+1}^i) - J_i^*(k_l^i)\\
\leq& -(1-\gamma)(1-v)\|x_i(k_l^i) - \bar{x}_{-i}(k_l^i|k_l^i)\| - \lambda\|u_i^*(k_l^i|k_l^i)\|\\
&+\frac{\alpha}{p} - \frac{\alpha}{h_l^i},
\end{aligned}$$

which, combined with $J_i^*(k_l^i) \geq 0$ for any l, gives that $\lim_{l\to\infty} h_l^i = p$, $\lim_{l\to\infty}\|x_i(k_l^i) - \bar{x}_{-i}(k_l^i)\| = 0$ and $\lim_{l\to\infty}\|u_i^*(k_l^i|k_l^i)\| = 0$ for all i according to LaSalle's invariance principle [46]. Therefore, it follows that the triggering interval converges to p and consensus is reached asymptotically when $\mathcal{G}$ contains a directed spanning tree. ∎

V. Simulation Examples

As two potential applications, synchronization of linear oscillators and platoon of vehicles are numerically studied in this section to demonstrate the effectiveness of the DMPC based consensus algorithm (Algorithm 1) and the self-triggered DMPC based consensus algorithm (Algorithm 2). Throughout the simulation examples, we use 'fmincon' function based on the interior-point method in the MATLAB toolbox to solve problems $\mathcal{P}_i$ and $\mathcal{SP}_i$ in Algorithms 1 and 2 respectively.

Example 1: Consider a network of 5 identical linear oscillators with interconnection topology shown in Fig. 2. The dynamics of oscillator i is given by (1) with $A = [0.9762\ 0.2169\ 0;\ -0.2169\ 0.9762\ 0;\ 0\ 0.9762\ 0.2169]$ and $B = [1\ 0;\ 0\ 1;\ 0\ 1]$. The state and input constraint sets for each i are $\mathbb{X}_i = \{x \in \mathbb{R}^3 || x_j| \leq 18, j = 1,2,3\}$ and $\mathbb{U}_i = \{u \in \mathbb{R}^2 || u_j| \leq 10, j = 1,2\}$ respectively. The initial states of all oscillator are randomly chosen from the uniform distributions on $[-10, 10]^3$. We first visualize the performance of the DMPC based consensus algorithm with $H = 3$, $\lambda = 0.01$, $\gamma = 0.6$, $v = 0.9$ and $\beta_i = 3$ for all i. Note that K_i in (13) could be set as $[-0.8970\ -0.2002\ -0.0015;\ 0.1058\ -0.9329\ -0.1002]$ such that (13) is satisfied. State trajectories of the agents are shown in Fig. 3, where x_1, x_2 and x_3 correspond to the first, second and third dimension of states, respectively. It reveals that dynamic consensus is achieved. Then we assume the oscillators apply the self-triggered DMPC based consensus algorithm (Algorithm 2) with $\alpha = 2$, $r = 1$, $p = 3$, $\lambda = 0.01$, $\gamma = 0.6$, $v = 0.9$ and $\beta_i = 3$ for all i. We also notice that K_i in (35) could be set as $[-0.3024\ -0.1404\ 0.0001;\ 0.1273\ -0.3255\ -0.0017]$ such that (35) is satisfied. State trajectories of the agents are shown in Fig. 4. It reveals that dynamic consensus is achieved, and the convergence speed is comparable to that in Fig. 3. Furthermore, Fig. 5 shows the triggering time instants of the oscillators with the triggering intervals converging to 3, and the total number of triggering instants is 175. In contrast to the DMPC based consensus algorithm (Algorithm 1) with 300 triggering time instants, the self-triggered DMPC based consensus algorithm (Algorithm 2) reduces the number of triggering time instants significantly, although the computation complexity is increased.

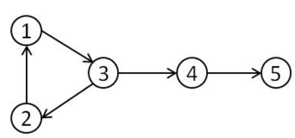

Fig. 2. Interconnection topology of the oscillators in Example 1.

Example 2: Consider a group of 5 vehicles moving along a single lane with information transmission flow shown in Fig. 6. The dynamics of each vehicle [7] is described by

$$\begin{cases} \dot{r}_i(t) = v_i(t) \\ \dot{v}_i(t) = \frac{1}{M_i} u_i(t) \end{cases}, \tag{39}$$

where $r_i \in \mathbb{R}$ and $v_i \in \mathbb{R}$ are the position and velocity of vehicle i respectively, $M_i > 0$ is the mass of vehicle i, and $u_i \in \mathbb{R}$ is the control input of vehicle i. The velocity constraint is 0 m/s $\leq v_i \leq 10$ m/s, and the input constraint is $|u_i| \leq 10$ N for all i. In order to effectively improve traffic safety and efficiency, the platooning of vehicles is to maintain

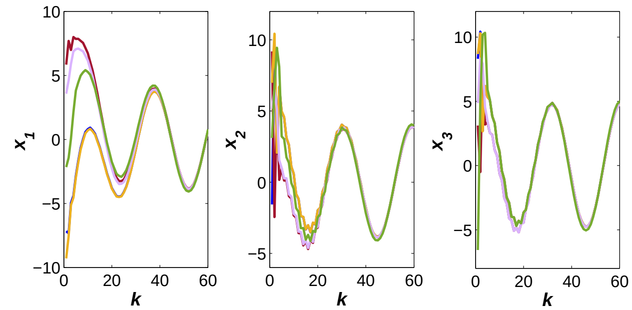

Fig. 3. State trajectories of the oscillators under the DMPC based consensus algorithm (Algorithm 1).

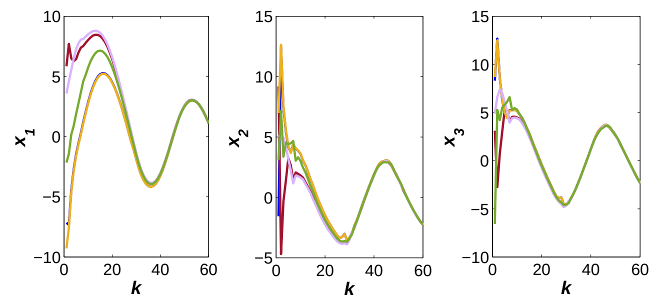

Fig. 4. State trajectories of the oscillators under the self-triggered DMPC based consensus algorithm (Algorithm 2).

a desired inter-vehicle spacing policy with a common velocity, i.e., $r_i(t) - h_i = r_j(t) - h_j$ and $v_i(t) = v_j(t)$ for all i and j as $t \to \infty$, where $h_i - h_j$ denotes the desired constant spacing between vehicle i and j.

As mentioned in Section II, a continuous-time linear multi-agent system in a periodic sampled-data setting can be transformed into discrete-time system (1) equivalently. We assume the sampled-data control is applied to system (39) with the sampling period equal to 1 s, let $M_i = 1$ kg for all i, and denote $x_i = [r_i - h_i;\ v_i]$. Then the platooning of system (39) can be solved by treating it as the consensus problem of discrete-time system (1) with $A = [1\ 1;\ 0\ 1]$ and $B = [0.5;\ 1]$. Assume the initial positions of vehicles are $r_1 = 0$ m, $r_2 = -40$ m, $r_3 = -50$ m, $r_4 = -100$ m, $r_5 = -140$ m, and the initial velocities of vehicles are $v_1 = 2.9$ m/s, $v_2 = 4.7$ m/s, $v_3 = 1.3$ m/s, $v_4 = 2$ m/s, $v_5 = 5$ m/s. Set $h_1 = 0$ m, $h_2 = -20$ m, $h_3 = -40$ m, $h_4 = -60$ m, $h_5 = -80$ m. We first visualize the performance of the DMPC based consensus algorithm with $H = 3$, $\lambda = 0.01$, $\gamma = 0.6$, $v = 0.9$ m/s and $\beta_i = 3$ for all i. Note that K_i in (13) could be set as $[-0.6167 - 1.2703]$ such that (13) is satisfied. Position and velocity trajectories of the vehicles are shown in Fig. 7, revealing that the desired inter-vehicle spacing policy with a common velocity is achieved. Then we assume the vehicles apply the self-triggered DMPC based consensus algorithm (Algorithm 2) with $\alpha = 2$, $r = 2$, $p = 4$, $\lambda = 0.01$, $\gamma = 0.6$, $v = 0.9$ and $\beta_i = 3$ for all i. We also notice that K_i in (35) could be set as

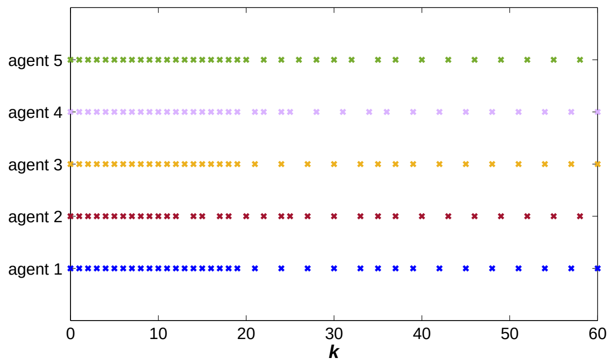

Fig. 5. Triggering time instants of the oscillators under the self-triggered DMPC based consensus algorithm (Algorithm 2).

$[-0.083 \ -0.4159]$ such that (35) is satisfied. Position and velocity trajectories of the vehicles are shown in Fig. 8, and it also reveals that the desired inter-vehicle spacing policy with a common velocity is achieved. Furthermore, Fig. 9 shows the triggering time instants of the vehicles with the triggering intervals converging to 4 s, the total number of which is 151. In contrast to the DMPC based consensus algorithm (Algorithm 1) with 300 triggering time instants, the number of triggering time instants determined by the self-triggered DMPC based consensus algorithm (Algorithm 2) is significantly reduced at the expense of increased computational complexity.

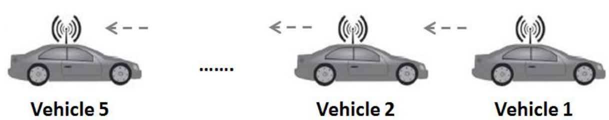

Fig. 6. Platoon configuration consisting of 5 vehicles, where the dashed arrows denote the information transmission flow among vehicles.

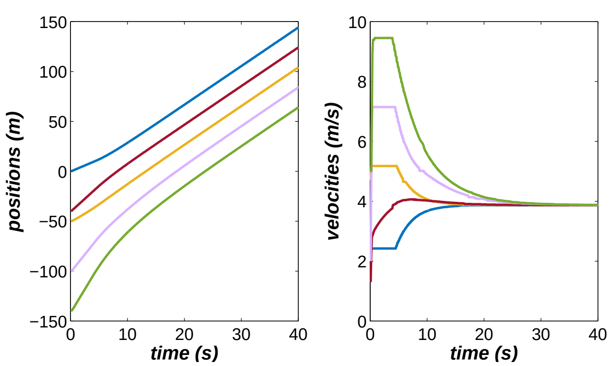

Fig. 7. Position and velocity trajectories of the vehicles under the DMPC based consensus algorithm (Algorithm 1).

VI. Conclusion

In this paper, we have studied the self-triggered DMPC based consensus problem in general linear discrete-time multi-agent systems with LTI dynamics. We have firstly proposed a DMPC based consensus algorithm, in which each agent needs to obtain its neighbors' predicted state sequences once to update its control input at each time step. Then we have presented a self-triggered DMPC based consensus algorithm, where the triggering interval is optimized together with the control input, and the information transmissions and control updates are executed at triggering time steps only. Both algorithms have been proved to be feasible and to drive the agents to achieve dynamic consensus. Two numerical examples have also been provided to validate the proposed algorithms. In contrast to the DMPC based consensus algorithm, the self-triggered DMPC based consensus algorithm can significantly reduce the information transmission and control update times without deteriorating the performance. An important and pressing topic for future research is to further investigate the self-triggered DMPC based consensus problem in heterogeneous multi-agent systems with disturbances.

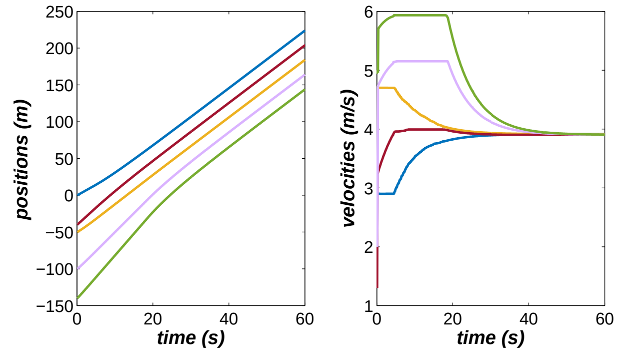

Fig. 8. Position and velocity trajectories of the vehicles under the self-triggered DMPC based consensus algorithm (Algorithm 2).

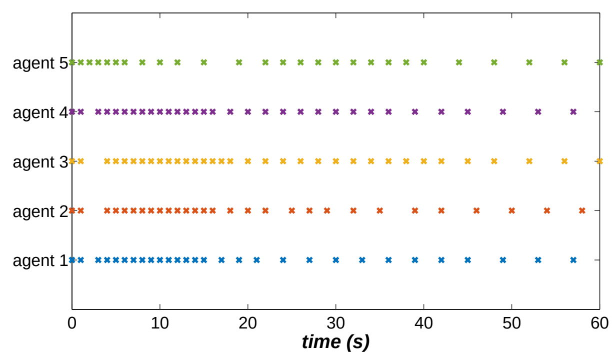

Fig. 9. Triggering time instants under the self-triggered DMPC based consensus algorithm (Algorithm 2).

References

[1] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-deriven particles," *Phys. Rev. Lett.*, vol. 75, no. 6, pp. 1226–1229, 1995.

[2] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[3] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, 2005.

[4] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in multi-agent networked systems," *Proc. of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[5] J. Zhan and X. Li, "Asynchronous consensus of multiple double-integrator agents with arbitrary sampling intervals and communication delays," *IEEE Trans. Circuits and Systems-I: Regular Papers*, vol. 62, no. 9, pp. 2301–2311, 2015.

[6] W. Ren, R. W. Beard, and E. M. Arkins, "Information consensus in multivehicle cooperative control," *IEEE Contr. Syst. Mag.*, vol. 71, no. 2, pp. 71–82, 2007.

[7] M. di Bernardo, A. Salvi, and S. Santini, "Distributed consensus strategy for platooning of vehicles in the presence of time-varying heterogeneous communication delays," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 1, pp. 102–112, 2015.

[8] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Quantized data and random link failures," *IEEE Trans. Signal Processing*, vol. 58, no. 3, pp. 1383–1400, 2010.

[9] M. Vazquez-Olguin, Y. Shmaliy, and O. Ibarra-Manzano, "Distributed unbiased FIR filtering with average consensus on measurements for WSNs," *IEEE Trans. Industrial Informatics*, vol. 13, no. 3, pp. 1440–1447, 2017.

[10] J. M. Solanki, S. Khushalani, and N. N. Schulz, "A multi-agent solution to distribution systems restoration," *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1026–1034, 2007.

[11] A. J. del Real, A. Arce, and C. Bordons, "An integrated framework for distributed model predictive control of large-scale power networks," *IEEE Trans. Industrial Informatics*, vol. 10, no. 1, pp. 197–209, 2014.

[12] G. Wen, G. Hu, J. Hu, X. Shi, and G. Chen, "Frequency regulation of source-grid-load systems: A compound control strategy," *IEEE Trans. Industrial Informatics*, vol. 12, no. 1, pp. 69–78, 2016.

[13] R. Scattolini, "Architectures for distributed and hierarchical model predictive control - a review," *Journal of Process Control*, vol. 19, pp. 723–731, 2009.

[14] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, pp. 549–558, 2006.

[15] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horzion control for large scale dynamically decoupled systems," *Automatica*, vol. 42, pp. 2105–2115, 2006.

[16] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, "Cooperative distributed model predictive control," *Systems and Control Letters*, vol. 59, pp. 460–469, 2010.

[17] G. Ferrari-Trecate, L. Galbusera, M. P. E. Marciandi, and R. Scattolini, "Model predictive control schemes for consensus in multi-agent systems with single and double integrator dynamics," *IEEE Trans. Circuits and Syst. I, Reg. Papers*, vol. 54, no. 11, pp. 2560–2572, 2009.

[18] H. Zhang, M. Z. Q. Chen, and G. B. Stan, "Fast consensus via predictive pinning control," *IEEE Trans. Circuits and Syst. I, Reg. Papers*, vol. 58, no. 9, pp. 2247–2258, 2011.

[19] J. Zhan and X. Li, "Consensus of sampled-data multi-agent networking systems via model predictive control," *Automatica*, vol. 49, no. 8, pp. 2502–2507, 2013.

[20] B. Johansson, A. Speranzon, M. Johansson, and K. H. Johansson, "On decentralized negotiation of optimal consensus," *Automatica*, vol. 44, no. 4, pp. 1175–1179, 2008.

[21] M. A. Müller, M. Reble, and F. Allgöwer, "Cooperative control of dynamically decoupled systems via distributed model predictive control," *Int. J. Robust. Nonlinear Control*, vol. 22, pp. 1376–1397, 2012.

[22] H. Li and W. Yan, "Receding horizon control based consensus scheme in general linear multi-agent systems," *Automatica*, vol. 56, pp. 12–18, 2015.

[23] D. Wu, X.-M. Sun, Y. Tan, and W. Wang, "On designing event-triggered schemes for networked control systems subject to one-step packet dropout," *IEEE Trans. Industrial Informatics*, vol. 12, no. 3, pp. 902–910, 2016.

[24] F. Li, B. Zheng, and H. Teng, "Design of smart home temperature control system based on event-triggered mechanism," *Modern Electron. Technique*, vol. 38, no. 2, pp. 158–162, 2015.

[25] C. Li, X. Yu, W. Yu, T. Huang, and Z. Liu, "Distributed event-triggered scheme for economic dispatch in smart grids," *IEEE Trans. Industrial Informatics*, vol. 12, no. 5, pp. 1775–1785, 2016.

[26] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[27] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, 2013.

[28] Y. Fan, G. Feng, Y. Wang, and C. Song, "Distributed event-triggered control of multi-agent systems with combinational measurements," *Automatica*, vol. 49, no. 2, pp. 671–675, 2013.

[29] X. Meng and T. Chen, "Event based agreement protocols for multi-agent networks," *Automatica*, vol. 49, no. 7, pp. 2125–2132, 2013.

[30] G. Guo, L. Ding, and Q. L. Han, "A distributed event-triggered transmission strategy for sampled-data consensus of multi-agent systems," *Automatica*, vol. 50, no. 5, pp. 1489–1496, 2014.

[31] Y. Fan, L. Liu, G. Feng, and Y. Wang, "Self-triggered consensus for multi-agent systems with zeno-free triggers," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2779–2784, 2015.

[32] E. Garcia, Y. Cao, A. Giua, and D. Casbeer, "Decentralized event-triggered consensus with general linear dynamics," *Automatica*, vol. 50, no. 10, pp. 2633–2640, 2014.

[33] H. Zhang, G. Feng, H. Yan, and Q. Chen, "Consensus of linear multi-agent systems via event-triggered control," *International Journal of Control*, vol. 87, no. 6, pp. 1243–1251, 2014.

[34] W. Zhu, Z. P. Jiang, and G. Feng, "Event-based consensus of multi-agent systems with general linear models," *Automatica*, vol. 50, no. 2, pp. 552–558, 2014.

[35] D. Yang, W. Ren, X. Liu, and W. Chen, "Decentralized event-triggered consensus for linear multi-agent systems under general directed graphs," *Automatica*, vol. 69, pp. 242–249, 2016.

[36] W. Hu, L. Liu, and G. Feng, "Consensus of linear multi-agent systems by distributed event-triggered strategy," *IEEE Trans. Cybernetics*, vol. 46, no. 1, pp. 148–157, 2016.

[37] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Event-triggered control for discrete-time systems," in *American Control Conf.*, 2010, pp. 4719–4724.

[38] D. Lehmann, E. Henriksson, and K. H. Johansson, "Event-triggered model predictive control of discrete-time linear systems subject to disturbances," in *European Control Conf.*, 2013, pp. 1156–1161.

[39] H. Li and Y. Shi, "Event-triggered robust model predictive control of continuous-time nonlinear systems," *Automatica*, vol. 50, no. 5, pp. 1507–1513, 2014.

[40] E. Henriksson, D. E. Quevedo, E. G. W. Peters, H. Sandberg, and K. H. Johansson, "Multiple-loop self-triggered model predictive control for network scheduling and control," *IEEE Trans. Control Systems Technology*, vol. 23, no. 6, pp. 2167–2181, 2015.

[41] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Event-based model predictive control for the cooperation of distributed agents," in *American Control Conf.*, 2012, pp. 6473–6478.

[42] A. Eqtami, S. Heshmati-Alamdari, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Self-triggered model predictive control for nonholonomic systems," in *European Control Conf.*, 2013, pp. 638–643.

[43] A. Eqtami, S. Heshmati-alamdari, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A self-triggered model predictive control framework for the cooperation of distributed nonholonomic agents," in *IEEE Conf. Decision and Control*, 2013, pp. 7384–7389.

[44] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.

[45] D. Dueri, J. Zhang, and B. Acikmese, "Automated custom code generation for embedded, real-time second order cone programming," in *19th IFAC World Congress*, 2014, pp. 1605–1612.

[46] J. Hurt, "Some stability theorems for ordinary difference equations," *SIAM Journal on Numerical Analysis*, vol. 4, no. 4, pp. 582–596, 1967.

[47] H. Y. Benson and U. Saglam, "Mixed-integer second-order cone programming: A survey," *In H. Topaloglu (Ed.), Tutorials in Operations Research (Catonsville, MD: INFORMS)*, pp. 13–36, 2013.