Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-4 (2016) 019–024

Self-triggered consensus of multi-agent systems via model predictive control ⋆

Jingyuan Zhan, Xiang Li

Adaptive Networks and Control Lab, and Research Center of Smart Networks and Systems, School of Information Science and Technology, Fudan University, Shanghai 200433, P. R. China (e-mail: jzhan13,lix@fudan.edu.cn)

Abstract: This paper proposes a self-triggered consensus algorithm for multi-agent systems by using model predictive control (MPC), where the self-triggering rule and the control algorithm are optimized jointly. The proposed self-triggered MPC consensus algorithm drives the system to reach consensus asymptotically under mild assumptions, if the communication topology is connected. Numerical examples are finally presented to verify the effectiveness and advantages of the self-triggered MPC consensus algorithm. By comparing with the conventional time-triggered and event-triggered consensus algorithms, the self-triggered MPC consensus algorithm is shown to achieve equivalent performance with significant reduction of the numbers of controller updates and information transmission.

Keywords: Multi-agent systems, consensus, event-triggered control, self-triggered control, model predictive control.

1. 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, and it has broad applications in various fields, such as formation control of multi-robotic systems, congestion control in communication networks, and distributed sensor networks (see Fax and Murray (2004); Antoniou et al. (2009); Kar and Moura (2010); Zhan and Li (2013b)). Among the extensive investigations concerning multi-agent systems, consensus which means that all agents reach an agreement on certain quantities of interest, is one of the most fundamental and widely studied problems, e.g. Vicsek et al. (1995); Jadbabaie et al. (2003); Ren and Beard (2005); Olfati-Saber et al. (2007); Zhan and Li (2015).

Over the past few years, there have been emerging studies on solving consensus problems with the involvement of model predictive control (MPC). Decentralized MPC consensus schemes with constraints on every agent's input were presented in Ferrari-Trecate et al. (2009), and another fast consensus algorithm was proposed in Zhang et al. (2011), where only a few pinned agents were equipped with the model predictive controllers. Recently, we proposed a novel consensus algorithm in Zhan and Li (2013a) based on distributed model predictive control, and found it was beneficial to accelerating the convergence speed and enlarging the feasible range of the sampling period. All the existing literatures on MPC consensus assume uniform sampling intervals such that the associated MPC problems, which are known to be computational demanding, are required to be solved periodically. Using such periodic MPC would result in unnecessary energy consumption waste and communication burden. Especially in practical multi-agent systems, where digital platforms of individual agents may have limited onboard energy resources and low-level communication and actuating capabilities, the lifespan of the system would be consequently shortened by applying the periodic MPC. Therefore, it is of practical significance to come up with novel MPC consensus algorithms with avoiding the unnecessary usage of computation and communication resources.

As an alternative to the periodic time-triggered control, event-triggered control, where controller updates are determined by certain events that are triggered depending on the agents' behaviors, is an efficient approach in reducing the waste of computation and communication resources while guaranteeing satisfactory levels of performance. Event-triggered control has gained large popularity in recent decades (see Draper et al. (1960); Otanez et al. (2002); Kofman and Braslavsky (2006); Miskowicz (2006); Tabuada (2007); Astrom (2008); Anta and Tabuada (2010); Jr. et al. (2010); Wang and Lemmon (2011); Persis et al. (2013); Liu and Jiang (2015); Postoyan et al. (2015) and the references therein), and it has been applied to addressing consensus problems of multi-agent systems lately, e.g. Dimarogonas et al. (2012); Seyboth et al. (2013); Fan et al. (2013); Meng and Chen (2013); Guo et al. (2014); Zhu et al. (2014). Earlier results on event-triggered consensus algorithms require continuous checking of the prescribed triggering conditions. To avoid this, a novel event-triggered technique, termed self-triggered control, has been developed with addressing consensus problems in Dimarogonas et al. (2012); Fan et al. (2013, 2015). With the self-triggered control, each agent decides the next event

⋆ This work was supported by the National Natural Science Foundation (No. 61273223), and the National Science Fund for Distinguished Young Scholars of China (No. 61425019).

10.1016/j.ifacol.2016.05.004

time instant by using the current and past received states from its neighbors.

Considering the heavy computation cost of MPC, there have been developments for MPC with event- or self-triggered mechanisms. Most of the results focused on single agent with linear/nonlinear dynamics, such as Eqtami et al. (2010); Lehmann et al. (2013); Li and Shi (2014), the key idea for which was to derive the triggering rules based on the difference between predicted and measured states. Furthermore, event-triggered MPC was developed for distributed agents with nonlinear dynamics in Eqtami et al. (2012), and it was extended to deriving self-triggered conditions in Eqtami et al. (2013). More recently, a self-triggering controller based on model predictive control was proposed in Henriksson et al. (2015), where the cost function was used to jointly determine the control input and the next event time instant. All the existing results on event- or self-triggered MPC addressed ultimate bounded stable problems such that they can not be directly applied to solving multi-agent consensus problems.

Therefore,the objective of this paper is to address consensus problems of multi-agent systems via self-triggered M-PC. Following the approach in Henriksson et al. (2015), we intend to propose a self-triggered MPC consensus algorithm with control input and triggering rule jointly optimized, aiming at improving the efficiency of the usage in computation and communication resources. The remainder of this paper is organized as follows. Preliminaries are presented in Section 2, and a self-triggered MPC consensus algorithm is proposed in Section 3 with the corresponding stability analysis given as well. Simulation results are provided in Section 4 to illustrate the effectiveness and advantages of the proposed self-triggered MPC consensus algorithm. Finally, Section 5 concludes the whole paper.

2. PRELIMINARIES

We first introduce the mathematical notations to use throughout this paper. $\mathbb{R}^n$ denotes the set of n-dimensional real column vectors. $\mathbf{1} = (1, 1, ..., 1)'$, $\mathbf{0} = (0, 0, ..., 0)'$, and I is the identity matrix with an appropriate dimension if no confusion arises. $\|x\| = (x'x)^{1/2}$ is the 2-norm of a column vector x.

Consider a networked multi-agent system consisting of N agents with the dynamics described as

$$\dot{x}_i(t) = u_i(t), i \in \mathcal{I}_N \tag{1}$$

where $\mathcal{I}_N = \{1, 2, ..., N\}$, $x_i \in \mathbb{R}^n$ denotes the state of agent i, and $u_i \in \mathbb{R}^n$ is the control input of agent i to be designed based on the state information received by agent i from its neighbors. The communication topology of the system is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ with the vertex set $\mathcal{V} = \{v_1, v_2, ..., v_N\}$, the edge set $\mathcal{E} \subseteq \{(v_i, v_j) : v_i, v_j \in \mathcal{V}, v_j \neq v_i\}$, and the weighted adjacency matrix $A = [a_{ij}]$. A graph $\mathcal{G}$ is undirected if $(v_i, v_j) \in \mathcal{E} \iff (v_j, v_i) \in \mathcal{E}$, and this paper only concerns undirected graphs. An undirected graph is connected if there exists a path, i.e., a sequence of distinct edges such that consecutive edges are joint, between any two vertices. The adjacency elements $\{a_{ij}\}$ associated with the edges are positive, and the others are zeros. The neighbors of agent i are denoted by $N_i = \{j \in \mathcal{I}_N : (v_i, v_j) \in \mathcal{E}\}$.

The network of agents reach a consensus if and only if $x_i = x_j$ for all $i, j \in \mathcal{I}_N, i \neq j$, and the classic consensus algorithm is given by

$$u_i = \sum_{j \in N_i} a_{ij}(x_j - x_i). \tag{2}$$

In an event-triggered setting, denote the triggering event time instants of agent i by a sequence $t_0^i, t_1^i, ..., t_k^i, ...$ for $i \in \mathcal{I}_N$. Then the control input of agent i is written as

$$u_i(t) = u_i(t_k^i), t_k^i \leq t < t_{k+1}^i. \tag{3}$$

In Dimarogonas et al. (2012), each agent determines its triggering event time instants by some prescribed event conditions established based on the state measurement of its own and its neighbors. The triggering condition in Seyboth et al. (2013) relies on the state measurement of each agent itself only. Event-triggered control can significantly reduce the number of controller updates such that plenty of energy can be saved and the lifespan of the system can be lengthened.

However, it is apparent that continuous monitoring of the state measurement is required in the event-triggered formulation. To further avoid this, self-triggered control has been developed in Dimarogonas et al. (2012); Fan et al. (2015, 2013), where each agent decides the next event time instant by using the current and past received states form its neighbors.

3. SELF-TRIGGERED MPC CONSENSUS

In this section, we propose a self-triggered consensus algorithm by using model predictive control. All the existing literatures on MPC consensus assume uniform sampling intervals, and using such time-triggered MPC not only results in unnecessary energy consumption waste and communication burden, but also requires a central synchronized clock. Distinguishing from the existing work concerning MPC consensus algorithms, model predictive control is used in this paper to jointly optimize the control inputs and the sampling intervals (or inter-event intervals) in order to reduce the numbers of controller updates and information transmission.

We assume $t_0^i = 0$ for any $i \in \mathcal{I}_N$. At each triggering event time instant t_k^i, agent i obtains sampled states of itself $x_i(t_k^i)$ and its neighbors $x_j(t_k^i)$. Agent i's control input is updated at its triggering event time instants only. Let $T(t_k^i)$ denote the inter-event interval, i.e. $t_{k+1}^i = t_k^i + T(t_k^i)$. Agent i solves the following MPC optimization problem with $u_i(t_k^i)$ and $T(t_k^i)$ determined jointly

$$\min_{u_i(t_k^i), T(t_k^i)} J_i\left(t_k^i, u_i(t_k^i), T(t_k^i)\right) = \min_{u_i(t_k^i), T(t_k^i)} \left(\left\|x_i(t_{k+1}^i|t_k^i) - \bar{x}_i(t_k^i)\right\|^2 + \gamma\|u_i(t_k^i)\|^2 + \frac{\lambda}{T(t_k^i)} \right) \tag{4}$$

subject to

$$x_i(t_{k+1}^i|t_k^i) = x_i(t_k^i) + T(t_k^i)u_i(t_k^i),$$
$$\bar{x}_i(t_k^i) = \frac{\sum_{j \in N_i} x_j(t_k^i)}{|N_i|},$$
$$T_{min} \leq T(t_k^i) \leq T_{max}. \tag{5}$$