Neurocomputing 000 (2017) 1–8

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Brief papers

Event-triggered circle formation control for second-order-agent system

Mei Yu[a,*], Hangfei Wang[a], Guangming Xie[b], Kaiqi Jin[a]

[a] School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, PR China
[b] College of Engineering, Peking University, Beijing 100871, PR China

ARTICLE INFO

Article history:
Received 1 April 2017
Revised 29 July 2017
Accepted 20 August 2017
Available online xxx

Communicated by Bo Shen

Keywords:
Multi-agent system
Second-order integrator
Event-triggered control
Lyapunov method

ABSTRACT

This paper studies the circle formation control for second-order integrator multi-agent systems with event-triggered control strategy. Centralized and decentralized event-triggered control laws are proposed and two corresponding event-trigger conditions are derived respectively using Lyapunov method. With the designed strategies and event-trigger conditions, the second-order system can achieve circle formation control. Finally, numerical simulations are given to illustrate the effectiveness of the proposed event-triggered control strategy.

1. Introduction

Numerous works concerning multi-agent systems (MAS) have appeared in the last decade [1–11]. Information consensus has attracted more and more attentions from many engineering application fields, such as formation control, flocking, artificial intelligence, automatic control, and so on [12].

A variety of team tasks can be carried out by cooperation of agents, such as environment monitoring, exploration, search, rescue and transportation, which are very difficult or inefficient to be accomplished by single agent. Meanwhile, the occurrence of forming formation for agents becomes a common phenomenon. For the formation control of multi-agent systems, one important topic is how to generate and maintain desired geometric pattern. Circle formations are the typical pattern, which have been investigated in some literatures [13–15]. As mentioned above, consensus problem can be applied in formation control. In [16], the circle formation control of multi-agent system is discussed in consensus framework. It is assumed that the agents move along a given circle in one-dimensional space, and each agent can only sense two agents immediately that are arranged in front of and behind itself. By introducing the distance variable which describes the angular distance from the agent to its immediate counterclockwise neighboring agent, the circle formation control of the system is transformed to ensure the introduced distance variables converging to the formation values.

Traditional control method of multi-agent systems is to design appropriate protocols, which are proposed according to time, such as continuous-time control, discrete-time control and sampled-data control. In practice, controllers are in general implemented on digital platforms. For multi-agent systems, an important aspect is the communication and controller actuation schemes. With the progress of computer and communication technologies, agents could be equipped with digital microprocessors. Thus it is necessary to implement protocols on digital platform, where control laws can be updated at discrete instances of time. Periodic samplings and control updates are the classical method. However in terms of the number of control updates, the periodic sampling control might be conservative, because the constant sampling period is obtained to guarantee stability in the worst case. Therefore, event-triggered control schedule has been proposed, where the control updates are not periodic, but determined by a series of certain events [17–19]. The event-triggered control schedule can decrease the energy consumption of systems by reducing control updates. Event-triggered control of multi-agent systems has been studied widely in [20–24].

As an extension of [16], sampled-data control strategies are designed to guide agents to reach the prescribed circle formation in literature [25]. Different from the literatures [16–25], this paper considers the event-triggered circle formation control of second-order integrator multi-agent systems. Based on the framework proposed in [16], we remain the basic assumptions unchanged, where the agents move in the one-dimensional space of a circle, and each

* Corresponding author.
E-mail addresses: meiyu@ncepu.edu.cn, yumei@pku.org.cn (M. Yu), wanghangfei@ncepu.edu.cn (H. Wang), xiegming@pku.edu.cn (G. Xie), jinkaiqi@ncepu.edu.cn (K. Jin).

http://dx.doi.org/10.1016/j.neucom.2017.08.061
0925-2312/

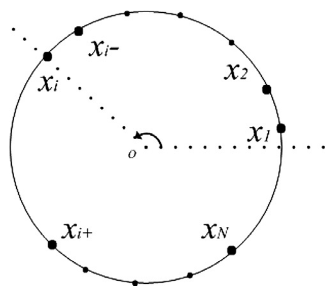

Fig. 1. Agents distributed on a circle.

agent can only sense its two neighboring agents. The circle formation of first-order multi-agent systems under event-triggered control schedule is studied in [26], while the velocity variable considered in the system of this paper seems more realistic.

Up to now, event-triggered strategy can be divided into centralized and decentralized control strategy according to the triggered method, respectively. When centralized event-triggered control strategy is adopted, the control input is updated using the global error, that is to say, each agent in MAS has the same triggered instants, numbers and triggered thresholds. Centralized structures can more easily address a crisis situation and can increase its resiliency by removing or reorganizing their upper management in response to a crisis while the rest of the organization remains mostly unaffected. However, each agent has its own event-triggered condition which is used to decide when to update its control input, and this may lead to asynchronous of the control update for each agent. On the other hand, decentralized control strategy means that each agent updates its own control input using its information and its neighbors' information only, which indicates that the triggered instants, triggered numbers and triggered thresholds are different for each agent. Most of the literatures do not pay attention to the above two different event-triggered strategies synthetically for circle formation control of second-order MAS.

There are two main contributions in this paper. The first one is circle formation control problem of second-order multi-agent systems under the proposed event-triggered control strategy. The second one is that both cases of centralized and decentralized event-triggered control strategies are considered respectively.

The remainder of this paper is organized as follows. In Section 2, we formulate the circle formation problem. The centralized and decentralized event-triggered circle formation control laws are proposed and analyzed in Section 3. Numerical simulations are given in Section 4, and the conclusions are given in Section 5.

2. Problem formulation

Suppose that the multi-agent system consists of N agents, and the agents move on a pre-given circle all the time. Obviously, the initial positions of the agents locate on one circle and no two agents occupy the same position. We establish coordinate system on the circle and denote the position of agent i by x_i which is measured by the angle in coordinate system. We label the agents counterclockwise by $1, 2, \ldots, N$ for analysis purpose, as shown in Fig. 1. Without loss of generality, we assume that the initial positions of agents satisfy

$$0 \le x_1(0) < \cdots < x_N(0) < 2\pi. \tag{1}$$

Each agent has the same dynamics described by a second-order integrator model:

$$\begin{cases} \dot{x}_i(t) = v_i(t) \\ \dot{v}_i(t) = u_i(t) \end{cases} \quad i = 1, 2, \ldots, N, \tag{2}$$

where u_i denotes the control input and v_i is the velocity of each agent.

Because each agent can only exchange information with its two neighboring agents, we denote agent i's two neighbors by i^+ and i^- by the following rules:

$$i^+ = \begin{cases} i+1 & i = 1, 2, \ldots, N-1, \\ 1 & i = N, \end{cases} \quad i^- = \begin{cases} N & i = 1, \\ i-1 & i = 2, 3, \ldots, N. \end{cases} \tag{3}$$

Further, we introduce two variables $y_i(t)$ and $\xi_i(t)$ to describe the angular distance and velocity difference respectively from agent i to its neighbor i^+. The mathematical expressions of $y_i(t)$ and $\xi_i(t)$ are defined as follows:

$$y_i = \begin{cases} x_{i^+} - x_i & i = 1, 2, ..., N-1, \\ x_{i^+} - x_i + 2\pi & i = N, \end{cases} \tag{4}$$

$$\xi_i = v_{i^+} - v_i. \tag{5}$$

Obviously $\sum_{i=1}^{N} y_i = 2\pi$ and $\sum_{i=1}^{N} \xi_i = 0$ always hold.

Let d_i denote the desired angular distance between agents i and i^+. The desired circle formation can be determined completely by the vectors

$$d = [d_1, d_2, ..., d_N]^T \text{and} \ \ y = [y_1, y_2, \ldots, y_N]^T. \tag{6}$$

We call the desired circle formation is admissible if $d_i > 0$ and $\sum_{i=1}^{N} d_i = 2\pi$. From the definitions (4) and (6), we can see that the angular distance y_i is varying over time while the value d_i is constant.

Definition 1 (Circle formation problem)**.** Given an admissible circle formation d, design event-triggered distributed control laws $u_i(t), i = 1, \ldots, N$, such that under any initial condition (1) the solution to system (2) converges to some equilibrium point x^*, which satisfies $y^* = d$.

Lemma 1 (Schur complement lemma [27])**.** *The symmetric linear matrix inequality*

$$\begin{bmatrix} Q(t) & S(t) \\ S^T(t) & R(t) \end{bmatrix} > 0$$

is equivalent to the following two conditions:

$$(1) Q(t) > 0, R(t) - S^T(t) Q^{-1}(t) S(t) > 0.$$

$$(2) R(t) > 0, Q(t) - S(t) R^{-1}(t) S^T(t) > 0.$$

In the following sections, both centralized and decentralized event-triggered circle-forming control laws are formulated.

3. Event-triggered circle formation

In this section, we present the event-triggered circle formation control strategies under centralized and decentralized cases respectively.

3.1. Centralized case

For the centralized event-triggered control strategy, each agent updates its controller at a sequence of event time synchronously. The event time sequence is determined by a common event-triggered condition and denoted by $t_0, t_1, \ldots,$ which corresponds to a sequence of control updates $u(t_0), u(t_1), \ldots$. Between a consecutive control updates, the value of input $u(t)$ is equal to the last control update, and is held constant by a zero-order holder, that is,

$$u_i(t) = u_i(t_k), t \in [t_k, t_{k+1}). \tag{7}$$