Multi-Agent Rendezvous Control Based on Event-Triggered Mechanism

Yang Zhang[1], Yuan Fan[1], Cheng Song[2], Chuanbao Dong[1], Lun Wang[1]
1.Electrical Engineering and Automation, Anhui University, Hefei 230601, P. R. China
E-mail: zhangyang910@foxmail.com;yuanf@ahu.edu.cn;dcbsucceed@foxmail.com;wanglun2016@foxmail.com
2.School of Automation, Nanjing University of Science and Technology, Nanjing 210094, P. R. China
E-mail: chengsong@njust.edu.cn

Abstract: In this paper, we apply the multi-agent system event-triggered control to the multi-agent rendezvous control, and design the multi-agent event-triggered controller. Meanwhile, we utilize the multi-agent system model that the state of all agents converge to the specified point based on the proportional transformation of the balance point of the original state of the agents. By selecting a suitable Lyapunov function, we prove that all the agents' position can converge to the desired rendezvous position. After that, an event-triggered control strategy for multi-agent rendezvous control is introduced. Simulation experiment is designed to test the rationality of the control strategy.

Key Words: Multi-Agent, Rendezvous, Event-Triggered Control

1 Introduction

Since multi-agent system has been proposed, it has been used widely in various industries. R. E. Mirollo and S. H. Strogatz try to solve the synchronization problem of coupled oscillators in [1],[3] et al and J. Cortes handle the cluster control problem in [4] with multi-agent system. Subsequently, Xiao and H. Ando apply the multi-agent system to the small world of rapid convergence and spacecraft rendezvous in [2]. Multi-agent system is an important branch of DAI (Distributed Artificial Intelligence) and composed of many agents, which aims to make all the agents cooperate with each other and accomplish the complex tasks that the individual agent can not accomplish. In multi-agent systems, all agents have their own laws of motion. The multi-agent system is an open system, which the agent is free and the agent can freely join or leave. Intelligence refers to the ability of the system to apply reasoning, learning and other techniques to analyze the data. The agent responds to the information according to the design method.
The event-trigger mechanism is a kind of control strategy, which is used to reduce the amount of data processing and the bandwidth of the system based on a certain condition. The problem how to effective reduce the amount of information of the system and ensure the stability of the system is the focus of research at home and abroad. The researchers study results published in [5],[6],[7],[8],[16],[17] for the event-triggered control.
In recent years, with the research of agent's rendezvous becomes a hot spot, more and more researchers pay attention to the control and application of agent. Jiang and Romain Postoyan et al design an agent rendezvous control using backstepping method in [9] and [10], separately. J. Alexander Fax and Suiyang Khoo give the corresponding algorithm for the master-slave consistency of a class of nonholonomic systems with multiple agents in [11], and the leader follower of the multi-agent system with unknown leader in [12]. Y. Dong, and J. Huang in [13] presents leader following connectivity preservation rendezvous based only on position measurement for multiple double integrator systems. W. Wang and J. S. Huang in [14] gives the proof that all the closed-loop signals of the output has the global uniform boundedness and consistent rendezvous asymptotically in multiple subsystems. In [15], Y. Q. Zhang and Y. G. Hong considered a leader in linear active multi-agent event-triggered rendezvous control problem, and given the proof of system stability based on Lyapunov function. In [18] and [19], C. Song gives the coverage control for heterogeneous mobile sensor networks on a circle and the optimal control algorithm for multi-agent persistent monitoring, respectively.
In this paper, the multi-agent system is used to control the rendezvous problem of swarm agents, where the event-triggered mechanism is introduced. The goal of multi-agent rendezvous is to bring all agents converge to the specified point. Furthermore, the zero-order holder model is introduced for the control input. By choosing a suitable Lyapunov function, the state error of the multi-agent system is satisfied by the first derivative of the Lyapunov function, and then the event-triggered control strategy is designed.
The contents of this thesis are as follows. In Section II, we introduce the preliminaries of this paper, including the knowledge of multi-agent graph theory, the basic model of the system and the meaning of some basic symbols used in this paper. In Section III, the feasibility of multi-agent

This work was supported by National Nature Science Foundation of China (No. 61573021; No. 61403203), the State Key Laboratory of Robotics and System (HIT) (No. SKLRS-2017-KF-02) and the Natural Science Foundation of Jiangsu Province (No. BK20140798).

rendezvous control is discussed. In Section IV, the event triggering mechanism of the multi-agent rendezvous is designed by using the first derivative of the Lyapunov function. In Section V, the simulation examples of the agents rendezvous event triggering control is given to verify the effectiveness of the control algorithm. Finally, the conclusion of this paper is given.

2 Preliminaries

2.1 Algebraic Graph Theory

In this section, we introduce some knowledge of algebraic graph theory. Consider an n-dimensional space consisting of N agents, where the number of agents is marked as $\{1, 2, 3, ..., N\}$. The information interaction topology between agents is represented by graph $G = (V, E)$, which the set of nodes $V = \{1, 2, ..., N\}$ and edges $E \subseteq V \times V$. Each agent i can communicate with its neighboring agent. Assume that the information exchange on edges E of the graph G is bidirectional. The adjacency matrix of graph G is expressed by A, in which elements given by either $a_{ij} = 1$ for $(i, j) \in E$ or $a_{ij} = 0$. a_{ij} shows that if agents between i and j can exchange information. The degree matrix of graph G is indicated by $D = diag\{d_1, d_2, ..., d_N\}$ with $d_i = card\{j : (i, j) \in E\}$. Define the Laplace matrix of graph G is $L = D - A$.

2.2 System Model

The system considered consists of N agents, with $x_i \in R$ denoting the state of agent i. We assume that the agents dynamics obey a single integrator model

$$\dot{x}_i = u_i(t), \qquad i = 1, 2, ..., N \tag{1}$$

where u_i denotes the control input for each agent. Each agent is assigned a subset $N_i \subset \{1, ..., N\}$ of the other agents, called agent i's communication set, that includes the agents with which it can communicate. The undirected communication graph $G = \{V, E\}$ of the multi-agent team consists of a set of vertices $V = \{1, ..., N\}$ indexed by the team members, and a set of edges, $E = \{(i, j) \in V \times V | i \in N_j\}$ containing pairs of vertices that correspond to communicating agents.
The agreement control laws were given by

$$u_i(t) = \sum_{j \in N_i} (x_i - x_j), \qquad i = 1, 2, ..., N \tag{2}$$

The closed-loop equations of the nominal system were $\dot{x}_i = \sum_{j \in N_i} (x_i - x_j), i \in \{1, ..., N\}$, so that $\dot{x} = -Lx$, where $x = [x_1, ..., x_N]^T$ is the stack vector of agents states and L is the Laplacian of the communication graph. We also denote by $u = [u_1, ..., u_N]^T$ the stack vector of control inputs. For a connected graph, all agents' states converge to a common point, called the agreement point, which coincides with the average $\frac{1}{N} \sum_i x_i(0)$ of the initial states.
For each agent i, and $t \geq 0$, introduce a time-varying error $e_i(t)$. Denote the vector $e(t) = [e_1(t), ..., e_N(t)]^T$. The sequence of event-triggered executions is denoted by: $t_0^i, t_1^i, ..., t_k^i, ...$ To the sequence of events $t_0^i, t_1^i, ..., t_k^i, ...$ corresponds a sequence of control updates $u_i(t_0^i), u_i(t_1^i), ..., u_i(t_k^i), ...$
Note the following symbols: x_i represents the two-dimensional coordinates (x_{i1}, x_{i2}) and $x_{i1} > 0, x_{i2} > 0$ for each agent i; x_o is the desired rendezvous point's two-dimensional coordinates (o_1, o_2), where $o_1, o_2 > 0$; $A \otimes B$ represents the Kronecker product of matrix A and matrix B; $\|C\|$ represents the norm of matrix C; I_n represents the unit matrix of order n.
The above control formulation is redefined here to integrate event-triggered strategies. We consider the system (1) of the multi-agent rendezvous problem in Cartesian coordinates. The control formulation for multi-agent rendezvous control is described in the following sections.

3 Multi-Agent Rendezvous Control

Rendezvous problem is described that all agents converge to a target point. In this paper, the desired rendezvous point $x_o(o_1, o_2)$ is a point in the coordinate system. Each agent follows the architecture of the previous design, and the agent interacts with each other. We assume that the graph of the multi-agent system is connected and undirected. The ultimate goal of all agents is to move to the desired rendezvous point x_o.
$\chi_i \in R^{2\times2}$ is the weight matrix $\chi_i = diag(\chi_{i1}, \chi_{i2})$ of agent i, where $i \in \{1, 2, ..., N\}$. Define the weighted state for each agent i :

$$\bar{x}_i(t) = \chi_i x_i(t) \tag{3}$$

where $\bar{x}_i(t)$ and $x_i(t)$ is the weighted state and the real state of the agent i, respectively.
For each agent i, we consider the following multi-agent dynamic system:

$$\dot{x}_i(t) = u_i(t) \tag{4}$$

Between control updates the value of the input u_i is held constant in a zero-order hold fashion, and is equal to the last control update, i.e.

$$u_i(t) = u_i(t_k^i), \quad t \in [t_k^i, t_{k+1}^i) \tag{5}$$

and thus the control law is piecewise constant between the event times $t_0^i, t_1^i, ...$
Let $q_i(t) \in R^{2\times1}$ be the measurement state of agent i, and defined by

$$q_i(t) = \sum_{j \in N_i} (\chi_j x_j - \chi_i x_i) \tag{6}$$

Define the measurement error $e_i(t) \in R^{2\times1}$:

$$e_i(t) = q_i(t_k^i) - q_i(t) \tag{7}$$

By definition (5), we propose the controller $u_i(t)$ for each agent as

$$u_i(t) = k_i q_i(t_k^i), \qquad t \in [t_k^i, t_{k+1}^i) \tag{8}$$

where $u_i \in R^{2\times1}$, and $k_i \in R^{2\times2}$ is the diagonal matrix $k_i = diag(\tilde{k}, \tilde{k})$. k_i is the feedback gain to be determined