ISA Transactions 61 (2016) 287–296

Contents lists available at ScienceDirect

ISA Transactions

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

Formation control and collision avoidance for multi-agent systems based on position estimation

Yuanqing Xia*, Xitai Na, Zhongqi Sun, Jing Chen

School of Automation, Beijing Institute of Technology, Beijing 100081, PR China

ARTICLE INFO

Article history:
Received 13 March 2015
Received in revised form
11 November 2015
Accepted 15 December 2015
Available online 16 January 2016

Keywords:
Formation control
Second-order systems
Optimal control
Collision avoidance

ABSTRACT

In this paper, formation control strategies based on position estimation for double-integrator systems are investigated. Firstly, an optimal control formation control strategy is derived based on the estimator. It is proven that the control inputs are able to drive the agents to the predefined formation and the controller is optimal even based on the estimation law if the estimator has converged to stable. Secondly, a consensus law based on the estimator is presented, which enables the agents converge to the formation in a cooperative manner. The stability can be guaranteed by proper parameters. Thirdly, extra control input for inter collision avoidance is added into the derived consensus control strategy, and efficacy analysis are provided in detail. Finally, the effectiveness of the strategies proposed are shown by simulation and experiment results.

1. Introduction

With the widely use of multi-agent, scholars are also growing their interest in agent's algorithm study. The research of multi-agent mainly focus on agents' formation control, and obstacle avoidance [1,3]. The cooperative control of multi-agent systems can be categorized as single-integrator system [4], double-integrator system [5], high-order system [6] and some more complex dynamics such as distributed-order fractional damping system [7] and nonholonomic robotic system [8,9].

The essence of formation control is the strict geometry coalescence control [1,4]. The purpose of formation control is to realize specified configuration or to reach to desired target point by adjusting individual's behavior. Formation control has many research methods now, we can be roughly divided into three categories: leader–follower, virtual structure and behavior based method. The leader–follower approach plus the Lyapunov and sliding mode methods are used in [11] and [12] to design cooperative controllers for a group of underactuated vessels. A combination of line-of-sight path-following and nonlinear synchronization strategies is studied in [13] to make a group of underactuated vessels asymptotically follow a given straight-line path with a given forward speed profile. Virtual structure is a wildly used formation control method [14], whereas, it is only suitable for formation control with less individuals. In recent years, Beard et al. [15] have done a lot of fruitful work by using distributed frame to realize virtual structure formation. Actually, the formation control based on behavior method is not often used, yet the typical literature include [16] and [17].

As nodes move to achieve a desired configuration they must avoid obstacles and remain connected. Navigation functions are originally developed in the seminal work in [18] to enable a single point-mass agent to move in an environment with spherical obstacles. The navigation function developed in [18] is designed to be a real-valued function that is designed so that the negated gradient field does not have a local minima and the agents' performance converges to a desired destination. Results such as [19] and [20] are motivated by the need to prevent the graph partitioning.

In most of these applications, strong ability of communication is needed by each individuals, in other words, the absolute position of each agent is required to carry out the following work such as formation control, consensus control and rendezvous control. But in order to get the exact location, it has to build a localization platform, which costs highly. Even if the localization platform is ready, in many cases we also cannot obtain the absolute position of agents, for instance the transmission fault, the limitations of the hardware and the factor of the energy costs make it difficult to get the absolute position. However, the real-time relative position of the multi-agents can be easily measured by using infrared detection or bluetooth devices. The velocity of each individual can also obtained from calculating the distance of adjacent two detection times by the photosensitive elements.

The works presented in this paper are based on our precious work [2] and the estimator proposed in [24]. However, the

* Corresponding author.
E-mail address: xia_yuanqing@bit.edu.cn (Y. Xia).

http://dx.doi.org/10.1016/j.isatra.2015.12.010
0019-0578/

controllers designed in this paper are quite different from [24], in which the estimation law is presented based on single-integrator system, and use unicycle systems to test its efficacy. The linear velocity of the unicycle is designed as a PD-like feedback law, which in fact is assumed to be a single integrator system, while the angular velocity is designed as a periodic time-varying cosinoidal function and is independent of states information. While in this paper, we expend its application into double-integrator systems. The main contribution can be summarized as follows: (a) an optimal control law based on the maximum principle is designed based on the estimator for the second order systems. It also has been proven that the control inputs are able to drive the agents to the predefined formation, and is optimal even based on the estimation law if the estimator converges to stable; (b) the consensus law based on the estimator is presented, which enables the agents converge to the formation in a cooperative manner. The stability can be guaranteed by proper parameters; (c) we further studied the inter collision avoidance scene based on the estimator and the consensus law presented, and provided the efficacy analysis in detail.

The structure of this paper is organized as follows. In Section 2, some basic knowledge of graph theories and some necessary preliminary results for formation background are presented. Section 3 states the model and problems we studied in this paper. Section 4 proposes an derived formation control strategy for the agents to realize cost limited while converging to the desired position puts forward the collision avoidance method used in this paper. Section 5 shows the results of the proposed control strategy to verify the proposed strategy and result of adding the extra control input which can realize obstacle avoidance effectively. In Section 6, we carried out experiments with the proposed algorithm. Finally, concluding remarks are given in Section 7.

2. Preliminaries

In this section, we introduce several notations in graph theory and list some useful notations for later reference.

A directed graph $\mathcal{G}$ consists of a finite set of vertex $\mathcal{V}(\mathcal{G}) = \{v_1, v_2, \dots. v_n\}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where an edge is an ordered pair of vertices in $\mathcal{V}(\mathcal{G})$. If (v_i, v_j) is an edge of $\mathcal{G}$, v_i is defined as the parent vertex and v_j is defined as the child vertex. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $A = [a_{ij}] \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ with non-negative elements, $|\mathcal{V}|$ denotes the cardinality of $\mathcal{V}$. The element a_{ij} of A is positive if $(v_i, v_j) \in \mathcal{E}$, which means there is a directed path from i to j, and it is zero otherwise. It is assumed that $(v_i, v_i) \notin \mathcal{E}$ for any $v_i \in \mathcal{V}$. The set of neighbors of $v_i \in \mathcal{V}$ is denoted by $\mathcal{N}_i = \{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$, called i's communication set, which includes the agents with which agent i can communicate. A directed graph $\mathcal{G}$ is said to have a spanning tree if there exists a vertex, called the root, such that it can be connected to all other vertices through paths. The node is called the root of the spanning tree. Here we concentrate on the directed graph which is often used in modeling communication topologies among agents. The degree d_i of the vertex i is defined by $d_i = \sum_{j \in \mathcal{N}_i} a_{ij}$. Let Δ be the $N \times N$ diagonal matrix of d_i. Then Δ is defined as the degree matrix of $\mathcal{G}$. The (combinatorial) Laplacian of $\mathcal{G}$ is denoted by the positive semi-definite matrix $L = \Delta - A$. Here we have $L = [l_{ij}]$ of $\mathcal{G}$ is

$$l_{ij} = \begin{cases} \sum\limits_{k \in \mathcal{N}_i} a_{ik} & i = j \\ -a_{ij} & i \neq j. \end{cases}$$

The following result is recalled which will be used later.

Lemma 1 (*Ren et al.* [21]). *Consider the linear system described by*

$$\dot{x} = -Lx \tag{1}$$

where $x = (x_1, x_2, \dots, x_N) \in \mathbb{R}^N$ and L is the Laplacian matrix of a weighted directed graph $\mathcal{G}$. For (1), there exists a finite vector $x^\infty = (x_c^\infty, \dots, x_c^\infty) \in \mathbb{R}^N$ such that x exponentially converges to x^∞ if and only if $\mathcal{G}$ has a spanning tree.

Lemma 2 (*Ren and Beard* [22]). *Let*

$$\rho_\pm = \frac{\gamma\mu - \alpha \pm \sqrt{(\gamma\lambda - \alpha)^2 + 4\mu}}{2} \tag{2}$$

where $\rho, \mu \in \mathbb{C}$. If $\alpha \geq 0$, $\mathrm{Re}(\mu) < 0$, $\mathrm{Im}(\mu) > 0$ and

$$\gamma > \Gamma(\mu) \tag{3}$$

where $\Gamma(\mu) = \sqrt{\dfrac{2}{|\mu| \cos\left[\tan^{-1}\dfrac{\mathrm{Im}(\mu)}{-\mathrm{Re}(\mu)}\right]}}$, *then $\mathrm{Re}(\rho) < 0$, where $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ represent, respectively, the real and imaginary parts of a number.*

3. Problem formulation

3.1. System dynamics

Consider the second order linear system of the N-agents

$$\begin{cases} \dot{p}_i = v_i \\ \dot{v}_i = u_i \end{cases} \tag{4}$$

where $p_i \in \mathbb{R}^n$, $v_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^n$ denote the position, velocity and control input of agent i, respectively. The overall system is rewritten as

$$\begin{cases} \dot{p} = v \\ \dot{v} = u \end{cases}$$

where $p = (p_1, \dots, p_N)$, $v = (v_1, \dots, v_N)$ and $u = (u_1, \dots, u_N)$. The interaction topology among the agents is modeled by a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, which is referred to as the interaction graph for the agents. As we have mentioned in Section 1, it is difficult to get agents' exact position p_i sometimes. To solve this problem, we first give the following assumptions:

- Agent i is able to sense the relative position of its neighbors, which is denoted as

  $$p_{ij} = p_j^i = p_j - p_i, \quad \forall j \in \mathcal{N}_i \tag{5}$$

  where p_j^i is the relative position of agent j to agent i, and cannot obtain the absolute position of the all agents.
- Each of the agent can measure its own absolute velocity $v_i, i = 1, \dots, N$ which is easy to get by calculating, and can obtain its neighbors' velocity information by communication.
- All the agents estimate their own positions denoted as $\hat{p}_i, i = 1, \dots, N$, and receive the values of the estimated position of their neighbors by communication.
- We assume agents are equipped with infrared and bluetooth function, which means the robot can explore the around environment, and it can take the corresponding collision avoidance measures once it explores other objects.
- Once agents $j (j \neq i)$ come into i's communication region, they can communicate with each other (we use it only in obstacle avoidance).
- We assume there exists dynamic obstacles between agents themselves only, i.e. has no static obstacles in the environment.

For the agents modeled by (4), suppose that the interaction graph is given by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ and the desired positions is $p_d = (p_{1d}, \dots, p_{Nd})$.