Event-triggered Formation Control of a Generalized Multi-agent System

Ryo Toyota and Toru Namerikawa
Department of System Design Engineering, Keio University, Kanagawa, Japan
(Tel: +81-45-563-1151; E-mail: ryot@nl.sd.keio.ac.jp, namerikawa@sd.keio.ac.jp)

Abstract: In this paper, we propose an event-triggered formation control algorithm for a multi-agent system consisting of linear discrete agents in order to reduce energy consumption due to unnecessary calculation and communication of agents. By using the event-triggered protocol, which updates the control input aperiodically only when certain triggering condition is satisfied, calculation and communication frequency are reduced. Also the multi-agent system is guaranteed an almost sure stability by determining the control input update timing by the triggering condition which is based on Lyapunov’s stability theorem. Finally we verify the effectiveness of the proposed algorithm by numerical simulation.

Keywords: Multi-agent system, formation control, event-triggered control

1. INTRODUCTION

In recent years, research on cooperative control of multi-agent systems, systems dealing with multiple control objects, has attracted many attention[1]. In a multi-agent system, each agent cooperates autonomously with other agents in order to achieve a certain task [2]. Research on cooperative control of multi-agent system is useful not only for understanding the mechanism of collective phenomena of living organisms but it is also known that it can be applied to various controlled objects such as UAV (Unmanned Aerial Vehicle)[3, 4], UUV (Unmanned Underwater Vehicle)[5], etc. Controls such as consensus control, formation control, enclosure control, flocking control, and swarming control are applied and are expected to be applied to various tasks such as monitoring, surveillance and transportation.

Now it can be said that the formation control is one of the most important problems in cooperative control of a multi-agent system. Formation is achieved when each agent forms a desired shape as a whole group by maintaining a certain relative distance with each other. In fact, formation of a multi-agent system is used in two wheel vehicle mobile robot[6], UAV[3, 4], artificial satellite[7], etc.

In order to realize formation of a multi-agent system, it is necessary to feedback the state, including the position of each agent, to the controller. When an autonomous vehicle, or an UAV is the object to be controlled, it is common to connect the controller and the agent by wireless communication. Here it is obvious that unnecessary communication and calculation between agents leads to waste of energy. Also, continuous communication causes consumption of communication resources between agents. Particularly, power consumption by communication and calculation can be a serious problem in multi-agent systems with limited battery capacity. Therefore, in order to reduce communication cost and calculation cost as much as possible, research on event-triggered control is attracting attention recently[8, 9].

Conventional research [10, 11] dealing with event-triggered formation of a multi-agent system, deals with agents represented by linear first order systems. Also, [12] deals with event triggered formation control of a multi-agent system using an agent represented by a general continuous linear system. However, with this method, it is only possible to achieve a formation shape that satisfies some specific condition, which is the target deviation from the position of other agents converges to 0 over time. Therefore, as far as we know, the paper handling event-triggered formation control of multi-agent system whose agent is represented by a general discrete linear system does not exist. Thus, in this paper, we aim to design an event-triggered formation control algorithm of multi-agent system represented by a general discrete linear system.

This paper is organized as follows. First as a problem description, the model of agents that constitutes the multi-agent system and the model of the whole multi-agent system is described. Next, an event-triggered formation control algorithm applied to the multi-agent system will be described. After examining the stability of the system based on Lyapunov’s stability theorem, we verify the effectiveness of the proposed control algorithm by numerical simulation.

2. PROBLEM DESCRIPTION

In this paper, we consider a multi-agent system consisting of N agents, considering a control rule that each agent $i \in \mathcal{N} = \{1, 2, \cdots, N\}$ achieves a formation based on a certain reference value.

In this paper, we consider a situation like Fig.1 where each agent is centrally controlled by a certain controller. Assume that each agent measures its own state and transmits back to each agent. When an autonomous vehicle, or an UAV is the object to be controlled, it is common to connect the controller and the agent by wireless communication. Here it is obvious that unnecessary communication and calculation between agents leads to waste of energy. Also, continuous communication causes consumption of communication resources between agents. Particularly, power consumption by communication and calculation can be a serious problem in multi-agent systems with limited battery capacity. Therefore, in order to reduce communication cost and calculation cost as much as possible, research on event-triggered control is attracting attention recently[8, 9].

Conventional research [10, 11] dealing with event-triggered formation of a multi-agent system, deals with agents represented by linear first order systems. Also, [12] deals with event triggered formation control of a multi-agent system using an agent represented by a general continuous linear system. However, with this method, it is only possible to achieve a formation shape that satisfies some specific condition, which is the target deviation from the position of other agents converges to 0 over time. Therefore, as far as we know, the paper handling event-triggered formation control of multi-agent system whose agent is represented by a general discrete linear system does not exist. Thus, in this paper, we aim to design an event-triggered formation control algorithm of multi-agent system represented by a general discrete linear system.

This paper is organized as follows. First as a problem description, the model of agents that constitutes the multi-agent system and the model of the whole multi-agent system is described. Next, an event-triggered formation control algorithm applied to the multi-agent system will be described. After examining the stability of the system based on Lyapunov’s stability theorem, we verify the effectiveness of the proposed control algorithm by numerical simulation.

2.1. Model of a Multi-agent System

The model of agent $i$ in this paper is defined as follows.

$$x_i(t+1) = A_ax_i(t) + B_au_i(t)$$

(1)
Here, \( x_i(t) \in \mathbb{R}^n \), \( u_i(t) \in \mathbb{R} \) represents the state and control input of agent \( i \) respectively. Also, \( A_a \in \mathbb{R}^{n \times n} \), \( B_a \in \mathbb{R}^n \) represents agent’s state matrix and input matrix respectively.

Also, the following is defined as a model of the multi-agent system.

\[
x(t + 1) = Ax(t) + Bu(t)
\]

Here, \( x(t) \triangleq [x_1^T(t), x_2^T(t), \ldots, x_N^T(t)]^T \in \mathbb{R}^{nN} \), \( u(t) \triangleq [u_1^T(t), u_2^T(t), \ldots, u_N^T(t)]^T \in \mathbb{R}^n \) are vectors that summarize the states and control inputs of the \( N \) agents existing in the multi-agent system respectively, and \( A \triangleq I_N \otimes A_a \in \mathbb{R}^{nN \times nN} \), \( B \triangleq I_N \otimes B_a \in \mathbb{R}^{nN} \) represents the state matrix and the input matrix of the multi-agent system respectively.

Furthermore, the network of the multi-agent system is represented by graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \), the vertex set of the graph \( \mathcal{V} = \{v_1, v_2, \ldots, v_N\} \) and the edge set of the graph \( \mathcal{E} = \{e_1, e_2, \ldots, e_N\} \). Also, an adjacency matrix \( \mathcal{A} \in \mathbb{R}^{N \times N} \), an order matrix \( \mathcal{D} \in \mathbb{R}^{nN \times nN} \), and a graph Laplacian \( L \in \mathbb{R}^{nN \times nN} \) are defined. Here, each element of the adjacency matrix \( A = [a_{ij}] \) is defined as follows.

\[
a_{ij} = \begin{cases} 1, & \text{for } (v_i, v_j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}
\]

Also, the order matrix \( \mathcal{D} \) is a diagonal matrix having the number of inputs \( \text{deg}(v_1), \text{deg}(v_2), \ldots, \text{deg}(v_N) \) to each vertex as diagonal elements, and is defined as follows.

\[
\mathcal{D} = \begin{bmatrix}
\text{deg}(v_1) & 0 & \cdots & 0 \\
0 & \text{deg}(v_2) & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & \text{deg}(v_N)
\end{bmatrix}
\]

Further, the graph Laplacian is defined as follows.

\[
L = \mathcal{D} - \mathcal{A} = \begin{bmatrix}
\sum_{j=1}^{N} a_{1j} & -a_{12} & \cdots & -a_{1N} \\
-a_{21} & \sum_{j=1}^{N} a_{2j} & \cdots & -a_{2N} \\
\vdots & \ddots & \ddots & \vdots \\
-a_{N1} & -a_{N2} & \cdots & \sum_{j=1}^{N} a_{Nj}
\end{bmatrix}
\]

Also consider a virtual agent which is not part of the multi-agent system, and define the state of the agent as the reference state \( x_{\text{ref}}(t) \in \mathbb{R}^n \). We consider formation in this paper based on this reference state. To achieve this formation, we define the target deviation \( d_i(t) \in \mathbb{R}^n \) of each agent \( i \) of the multi-agent system for the reference state. Each agent achieves a formation by maintaining this target deviation from the reference state.

2.2. Control Objective

The control objective in this paper is as follows.

1. All agents \( i \in \mathcal{N} \) asymptotically converge to the target relative state \( d_i(t) \in \mathbb{R}^n \) viewed from the reference state \( x_{\text{ref}}(t) \in \mathbb{R}^n \) (Fig.2). In other words, achieve formation.

\[
\lim_{t \to \infty} \|x_i(t) - (x_{\text{ref}}(t) + d_i(t))\| = 0, \quad \forall i \in \mathcal{N}
\]

2. Reduce the number of control input updates.

3. EVENT-Triggered FORMATION CONTROL ALGORITHM

Here, event-triggered formation control algorithm applied to the multi-agent system (2) is defined as follows.

\[
u(t) \triangleq -(\tilde{L} \otimes K)(\tilde{x}(\tau_k) - \tilde{d}(\tau_k)),
\]

\[
t \in \{\tau_k, \tau_k + 1, \tau_k + 2, \ldots, \tau_{k+1} - 1\}
\]

Since this paper considers centralized control, the control input for each agent \( i \in \mathcal{N} \) is an element of \( u(t) \) in (2).
Now $K \in \mathbb{R}^{1 \times n}$ is the feedback gain and $L \in \mathbb{R}^{N \times N}$ is the graph Laplacian of the system. Also, $t_k, t_{k+1}, \ldots$ is the event time when the input is updated, and $t_k + 1, t_{k} + 2, \ldots$ is the time ahead of the event time $t_k$ by 1, 2, \ldots steps. Therefore, from $t_k$ to $t_{k+1} - 1$, the same input value is applied to the system. We further define the following.

$$
\bar{L} = [L + \text{diag}(l), -l] \in \mathbb{R}^{N \times (N+1)}
$$

$$
l = [l_1, l_2, \ldots, l_N]^T \in \mathbb{R}^N
$$

Here, the input $l$ is obtained by adding information on the reference state to the target relative state from the reference state. If information of the reference state is not added, a for-stabilizing the state $x_i(t) \in \mathbb{R}^n$ of a single agent $i$ represented by the following equation to $0_i$ by event-triggered control.

$$
x_i(t+1) = A_{x_i} x_i(t) + B_a u_i(t)
$$

Here, the input $u_i(t) \in \mathbb{R}$ is given in the form of state feedback using the feedback gain $K \in \mathbb{R}^{1 \times n}$ and it is supposed to be updated aperiodically.

$$
u_i(t) = -K x_i(t_k), \quad t \in \{t_k, t_k + 1, t_k + 2, \ldots, t_{k+1} - 1\}
$$

Let us consider setting the Lyapunov function $V$ as follows and derive a triggering condition in which $V$ decreases over time.

$$
V(x_i) = x_i^T P x_i
$$

The state $x_i(t_{k} + 1)$ at $t_{k} + 1$ which is one step ahead from the most recent event time $t_k$ can be expressed as follows using the latest input update value $u_i(t_k)$.

$$
x_i(t_{k} + 1) = A_{x_i} x_i(t_k) + B_a u_i(t_k)
$$

$$
= A_{x_i} x_i(t_k) + B_a (-K x_i(t_k))
$$

$$
= (A_a - B_a K) x_i(t_k)
$$

Therefore, the value of the Lyapunov function $V(x_i(t_{k} + 1))$ at time $t_{k} + 1$ is expressed as follows.

$$
V(x_i(t_{k} + 1))
$$

Here, if

$$
(A_a - B_a K)^T P (A_a - B_a K) - \beta P \leq 0
$$

holds, the following expression holds.

$$
V(x_i(t_{k} + 1))
$$

$$
\leq \beta x_i^T (t_k) P x_i(t_k)
$$

Here, since $x_i^T (t_k) P x_i(t_k)$ is the value of the Lyapunov function $V(x_i(t_k))$ at the latest event time, the following holds.

$$
V(x_i(t_{k} + 1)) \leq \beta V(x_i(t_k))
$$

This indicates that the value of the Lyapunov function $V(x_i(t_{k} + 1))$ at $t_{k} + 1$, which is one step after the most recent event time $t_k$, is smaller than the value obtained by multiplying by the value of the Lyapunov function $V(x_i(t_k))$ at the latest event time $t_k$ by scalar $\beta \in (0, 1)$ (Fig.3).

From references [13, 14], similarly, for the section $t \in \{t_k + 1, t_k + 2, \ldots, t_{k+1}\}$ up to the next event time, it can be described as follows.

$$
V(x_i(t)) \leq \beta x_i^T (t_k) P x_i(t_k)
$$

$$
\leq \beta V(x_i(t_k)),
$$

Here, by giving the next event time $t_{k+1}$ as follows using the Lyapunov function $V$, it is possible to decrease
at the time when the value of the Lyapunov function $V$ is likely to increase beyond a certain threshold value.

$$\tau_{k+1} = \min\{V(A_x x_i(t) + B_x u_i(t_k)) > \beta V(x_i(t_k))\}$$

(21)

Therefore, when the control input (12) and the triggering condition (21) are applied to agent $i$ represented by (11), the state of agent $i$ is guaranteed almost sure stability to $\theta_n$. Here, the reason for being almost sure stability is because $V(x(t))$ is not monotonically decreasing when the value of the Lyapunov function $V(x(t))$ is smaller than $\beta V(x(t))$ as in Fig.3.

Next, consider the following situation in which the reference input $r_i(t)$ is added to the input $u_i(t)$ of agent $i$. As a result, the state $x_i(t)$ of each agent can converge to an arbitrary state, not to $\theta_n$.

$$x_i(t + 1) = A_a x_i(t) + B_a u_i(t)$$

(22)

$$u_i(t) = r_i(t) - K x_i(t)$$

(23)

$$x_i(t + 1) = (A_a - B_a K)x_i(t) + B_a r_i(t)$$

(24)

Here, from [15] it is shown that $(A - BK, B)$ is also controllable if $(A, B)$ is controllable. Further, from [16], it is generally known that the value of the reference input $r_i(t)$ does not affect the stability of the system even if it affects the value of the steady state of the system. Therefore, if the reference input $r_i(t)$ is set as follows,

$$r_i(t) = K(x_{\text{ref}}(t) + d_i(t))$$

(25)

$$u_i(t) = r_i(t) - K x_i(t)$$

(26)

then $\lim_{t \to \infty} x_i(t) = x_{\text{ref}}(t) + d_i(t)$ holds. Therefore, the following holds and agent $i$ will be guaranteed almost sure stability to the target relative state from the reference state $x_{\text{ref}}(t) - d_i(t) \in \mathbb{R}^n$.

$$\lim_{t \to \infty} ||x_i(t) - (x_{\text{ref}}(t) + d_i(t))|| = 0$$

(27)

Thus, even in this case as well, when the control input is made aperiodic like the following, and by setting the Lyapunov function $V$ with input $u_i(t)$ in (28), and by giving the next event time $\tau_{k+1}$ using $P$ and $\beta$ as follows, the Lyapunov function $V$ can be decreased at the time when the Lyapunov function $V$ is likely to increase beyond a certain threshold.

$$u_i(t) = -K(x_i(t) + (x_{\text{ref}}(t_k) + d_i(t_k))),$$

(28)

$$t \in \{\tau_k, \tau_k + 1, \tau_k + 2, \cdots, \tau_{k+1} - 1\}$$

$$\tau_{k+1} = \min\{V(A_x x_i(t) + B_x u_i(t_k)) > \beta V(x_i(t_k))\}$$

(29)

Therefore, by applying the control input (28) and the triggering condition (29) to agent $i$ represented by (22), the state of agent $i$ is guaranteed almost sure stability to $x_{\text{ref}}(t) - d_i(t) \in \mathbb{R}^n$.

The stability remains the same even if the number of agents is increased and spreading the idea to a multi-agent system. (2) and (7) represent the states and control inputs of all agents $i \in \mathcal{N}$ at the same time. Therefore, if the feedback gain $K$, positive definite matrix $P$, and scalar $\beta \in (0, 1)$ satisfying (9), and the triggering condition (10) are used, the control objective will be achieved.

4. NUMERICAL SIMULATION

Here, we confirm the effectiveness of the proposed method by numerical simulation. In this section, we simulate using a quadrotor which is one kind of an UAV as an agent. Although the model of the quadrotor is usually expressed as a nonlinear model, if linearization is held around the equilibrium point based on [3, 4], the vertical model can be expressed as a 2nd order system and the horizontal model can be expressed as a 4th order system.

4.1. Linear Quadrotor Model[3,4]

By discretizing the vertical and the horizontal model of the continuous linear quadratic system, the following is obtained where $\Delta t$ is the step time.
Table 1 Definition of symbols for UAV model

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>x, y</td>
<td>[m]</td>
<td>Position in ground coordinate</td>
</tr>
<tr>
<td>h</td>
<td>[m]</td>
<td>Altitude in ground coordinate</td>
</tr>
<tr>
<td>u, v, w</td>
<td>[m/s]</td>
<td>Velocity in body coordinate</td>
</tr>
<tr>
<td>φ, θ, ψ</td>
<td>[rad]</td>
<td>Attitude angle</td>
</tr>
<tr>
<td>p, q, r</td>
<td>[rad/s]</td>
<td>Angular velocity</td>
</tr>
<tr>
<td>M₁, M₂, M₃</td>
<td>[Nm]</td>
<td>Moment command</td>
</tr>
<tr>
<td>Tₜₜ</td>
<td>[N]</td>
<td>Total thrust command</td>
</tr>
<tr>
<td>Iₓₓ, Iᵧᵧ, I𝑧𝑧</td>
<td>[kgm²]</td>
<td>Inertia moment</td>
</tr>
<tr>
<td>rᵢ</td>
<td></td>
<td>Yawing moment coefficient</td>
</tr>
<tr>
<td>m</td>
<td>[kg]</td>
<td>Quadrotor mass</td>
</tr>
<tr>
<td>g</td>
<td>[m/s²]</td>
<td>Gravity constant</td>
</tr>
</tbody>
</table>

**Vertical Model:**

\[
x_{vi}(t+1) = A_{vi}x_{vi} + B_{vi}u_{vi}
\]

\[
A_{vi} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad B_{vi} = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \frac{\Delta t}{2} \end{bmatrix}
\]

**Horizontal Model:**

\[
x_{hi}(t+1) = A_{hi}x_{hi} + B_{hi}u_{hi}
\]

\[
A_{hi} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} & \frac{\Delta t^3}{6} \\ 0 & 1 & \frac{\Delta t}{2} & \frac{\Delta t^2}{4} \\ 0 & 0 & 1 & \frac{\Delta t}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_{hi} = \begin{bmatrix} \frac{\Delta t^4}{24} \\ \frac{\Delta t^3}{6} \\ \frac{\Delta t^2}{2} \end{bmatrix}
\]

4.2. Simulation Condition

In this simulation, we verify whether multiple agents can converge to a target state for a certain reference state under the proposed control algorithm. Simulation conditions are summarized in Table 2 below. Here, feedback gain \( K \) was taken so that \((A - BK)\) was stable, and \( P, \beta \) were taken to satisfy (9). Simulation time was set to 30 [s].

Table 2 Simulation Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta t )</td>
<td>0.01 [sec]</td>
</tr>
<tr>
<td>( K )</td>
<td>98.3, 3.03, 4.19, 3.05</td>
</tr>
<tr>
<td>( P )</td>
<td>308.27, 423.61, 307.78, 100.00</td>
</tr>
<tr>
<td>( \beta )</td>
<td>0.99</td>
</tr>
</tbody>
</table>

4.3. Simulation Results

The disks in Fig. 5 represents each agent, the lines are their trajectories respect to their color, and the black disk indicates the position of the virtual agent. Also, the orange lines represent the network structure. Fig. 6 is a graph representing the transition of \( V(t) \) over time. Also, Fig. 7 is a graph representing the formation error.

The multi-agent system successfully achieved a formation, and the number of control input updates were 420 out of 3000 steps as shown in Fig. 8.
5. CONCLUSION

In this paper, we proposed an event triggered algorithm to reduce energy consumption due to unnecessary calculation and communication of agents in formation control of multi-agent system represented by discrete linear system. In the proposed method, the control input is aperiodically updated only at the time when the specific triggering condition holds. Thereby reducing the number of calculations and the number of communications. Also, almost sure stability was guaranteed by determining the trigger condition based on Lyapunov’s stability theorem. Numerical simulation was carried out and the effectiveness of the proposed method was confirmed as well.

Future tasks include devising a method for optimum determination of parameters such as $K, \beta, P$, self-triggered control which does not require continuous calculation of the triggering condition until the next input update timing, decentralizing the algorithm, consideration of collision avoidance among agents, and actual experiments.

6. ACKNOWLEDGEMENTS

This work was supported by Technology Foundation of the R&D project “Design of Information and Communication Platform for Future Smart Community Services” by the Ministry of Internal Affairs and Communications of Japan.

REFERENCES