Article

Event-Triggered Containment Control of Multi-Agent Systems With High-Order Dynamics and Input Delay

Jianzhen Li *, Chenglong Li, Xiaofei Yang and Wei Chen

School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China; 18260636507@163.com (C.L.); yxf0809@just.edu.cn (X.Y.); cwchenwei@aliyun.com (W.C.)

* Correspondence: jzli@just.edu.cn

Received: 1 October 2018; Accepted: 19 November 2018; Published: 22 November 2018

Abstract: This paper studies the problem of distributed containment control for multi-agent systems with high-order dynamics and input delays. Two event-triggered control algorithms are proposed for multi-agent systems without and with input delay, respectively. The communication instants between two linked followers are determined by the event-triggering condition, and every follower can detect the event based on its own control input. For the followers, edge-based estimators are adopted to predict state differences to neighbors. Control inputs of the followers are calculated based on the predicted values of the state differences. To deal with the input delay, a delay comprehension approach is developed. It is proved that for arbitrarily large but bounded input delays, the followers can move into the convex hull spanned by the leaders asymptotically. Simulation results show the effectiveness of the proposed algorithms.

Keywords: containment control; input delay; event-triggered control

1. Introduction

Distributed cooperative control of multi-agent systems has been studied extensively in the past decade. Compared with its centralized counterpart, distributed control has many advantages such as low cost, easy maintenance, and high adaptivity. As a fundamental problem in distributed cooperative control, the consensus problem has attracted much attention. For a group of agents, consensus means reaching an agreement on a quantity of interest. The consensus problem can be classified into leaderless consensus and leader-following consensus, according to the absence or presence of a leader. Recently, much progress has been made on both leaderless and leader-following consensus problems (see [1–5] and references therein).

Although the leader-following consensus problem for multi-agent systems with single leader is interesting, it is sometimes more meaningful to study the leader-following problem for multi-agent systems with multiple leaders. This problem is called the containment problem, the objective of which is to steer all the followers into the convex hull spanned by the leaders. In recent years, researchers have paid much attention to the containment control problem owing to its wide applications in swarm robotics [6–16]. In [17], the containment control problem is investigated in both continuous-time and discrete-time domain. A $PI^n$-Type containment control algorithm is proposed, which allows the followers to be of any-order integral dynamics.

In the aforementioned results, it is assumed that the agents can access their neighbors’ states continuously or with a fixed frequency. This is unnecessary or unrealistic in many applications. On one hand, it is a waste of communication resource to access the neighbors’ states frequently when the system states nearly approach their equilibriums or there are no disturbances imposed on the system [18–20]. On the other hand, too much communication may lead to rising power costs and network congestion, which may degrade the performance of the system [21–23]. The event-triggered control is a good
strategy to solve this problem [24–26]. Compared with the traditional sampled-data-based control,
where the measurement or control is triggered by time, the main feature of the event-triggered control is
that the measurement or control action is triggered by an event condition. The event-triggered control
strategy was applied in both leaderless and leader-following consensus problems for multi-agent
systems in [27–29]. For more details, one can refer to [30] and references therein. More recently,
the authors in [31–33] have proposed some event-triggered containment control algorithms for
multi-agent systems with multiple leaders.

Time-delay exists in most practical systems. So, it is very meaningful to investigate the stability
of time-delay systems [34–40]. For multi-agent systems, the effect of time-delay is also an important
problem to be considered. There are two sources of time-delay in multi-agent systems. One source
is the communication among agents, which is called communication delays. The other source is the
processing time for the information arrived at each agent, which is called input delays. The consensus
problem for multi-agent system with communication/input delays has been extensively investigated in
the literature (see [41] and references therein). For second-order multi-agent systems with time-varying
delays, the containment control problem is considered in [42]. Recently, Miao et al. addressed
the containment problem for second-order multi-agent systems with constant input delays [43].
An event-triggered containment control algorithm was proposed, together with conditions under
which the followers will converge to the convex hull spanned by the leaders.

In this paper, we consider the event-triggered containment control problem for multi-agent
systems with high-order dynamics and input delay. Motivated by [29], model-based approach is
developed to avoid continuous communications among followers, and edge-based estimators are
designed to predict state differences to neighbors. New event-triggered containment control algorithms
are proposed for multi-agent systems without and with input delay, respectively. Sufficient conditions
are derived under which the followers will move into the convex hull formed by the leaders.
Compared with existing literatures on the containment control problem for multi-agent systems,
our contribution is summarized as follows.

- A delay compensation-based event-triggered containment controller is developed. It is proved
that for arbitrarily large but bounded constant input delays, the proposed controller can drive all
the followers into the convex hull formed by the leaders. In contrast, the controller in [43] can
deal with input delays below an upper bound only.

- The proposed algorithms are distributed, while the algorithms in [31–33,43] are centralized.
Using the algorithms proposed in this paper, every follower can decide whether the event should
be triggered based on its own control input. However, using algorithms in [31–33,43], all agents
need their neighbors’ states to trigger the next communication.

- Compared with the containment control algorithms with continuous [6–16] or periodic
communications [17], the proposed event-triggered containment control algorithms has the
potential to reduce the communication burden of multi-agent systems.

**Notations:** Throughout this paper, $\mathbb{R}^l$ denotes the $l$ dimensional Euclidean space, $0_{p \times q}$ denotes a
$p \times q$ matrix with all the elements to be zero. Given a vector $\mu$, $\|\mu\|$ denotes the Euclidean norm of
$\mu$. Given a matrix $A$, $A > 0$ means that $A$ is a positive definite matrix, $\|A\|$ is the induced norm of $A$.
The superscript $T$ denotes the transpose of a vector or matrix. We use $\text{diag}(\mu_1, \ldots, \mu_p)$ to denotes the
diagonal matrix of all $\mu_1, \ldots, \mu_p$.

2. Problem Formulation

2.1. Algebraic Graph Theory

Consider a group of agents consisted of $M$ leaders and $N$ followers. The leaders do not need
to access information from the followers, while the followers are guided by the leaders. We assume
that only a part of the followers can access information from some leaders. These followers are called
informed followers. The rest of the followers cannot receive information from the leaders directly.
We use graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to denote the communication topology among the leaders and the followers, where $\mathcal{V} = \{1, \cdots, M+N\}$ is the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. A directed edge $(j, i) \in \mathcal{E}$ denotes that node $i$ can receive information from node $j$ but not necessarily vice versa. An undirected edge $(j, i) \in \mathcal{E}$ denotes that node $i$ and node $j$ can access information from each other. A graph is undirected if all the edges in the graph are undirected. The neighbor set $\mathcal{N}_i = \{ j \mid (j, i) \in \mathcal{E} \}$ denotes the set of nodes from which node $i$ can access information. A directed path is a sequence of directed edges of the form $(i_1, i_2), (i_2, i_3), \ldots$, where $i_j \in \mathcal{V}$.

The adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{(M+N) \times (M+N)}$ is defined as $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. The Laplacian matrix $L = [l_{ij}] \in \mathbb{R}^{(M+N) \times (M+N)}$ is defined as $l_{ii} = \sum_{j=1}^{M+N} a_{ij}$ and $l_{ij} = -a_{ij}, i \neq j$. We use $\mathcal{L} = \{1, 2, \cdots, M\}$ and $\mathcal{F} = \{M+1, M+2, \cdots, M+N\}$ to denote node set of the leaders and the followers, respectively. Notice that $a_{ij} = 0$ for all $i = 1, \cdots, M$ and $j = 1, \cdots, M+N$, because the leaders have no neighbors. Therefore, the Laplacian matrix $L$ can be rewritten as

$$ L = \begin{bmatrix} 0_{M \times M} & 0_{M \times N} \\ L_1 & L_2 \end{bmatrix}. \quad (1) $$

In this paper, for the sake of convenience, we assume that $a_{ij} = 1$, if $(j, i) \in \mathcal{E}$.

2.2. System Models and Control Objectives

Consider a network of $M$ leaders and $N$ followers with the following linear dynamics:

$$ \dot{x}_i(t) = Ax_i(t), \quad i \in \mathcal{L}, \quad (2) $$

$$ \dot{\bar{x}}_i(t) = Ax_i(t) + Bu_i(t), \quad i \in \mathcal{F}, \quad (3) $$

where $x_i(t) \in \mathbb{R}^n$ and $u_i(t) \in \mathbb{R}^m$ are the state and the control input of agent $i$, respectively; $A$ and $B$ are constant matrices with appropriate dimensions.

When there exists a constant input delay $\tau$, (3) becomes

$$ \dot{x}_i(t) = Ax_i(t) + Bu_i(t-\tau), \quad i \in \mathcal{F}. \quad (4) $$

**Definition 1.** Let $\mathcal{C} = \mathbb{R}^p$. The set $\mathcal{C}$ is said to be convex if for any $x$ and $y$ in $\mathcal{C}$, the point $(1-a)x + ay$ is in $\mathcal{C}$ for any $a \in [0, 1]$. The convex hull $\text{Co}(X)$ of a set of points $X = \{x_1, \cdots, x_q\}$ is the minimal convex set containing all points in $X$.

**Definition 2.** We say algorithm $u_i(t)$ asymptotically solves the containment problem if, under algorithm $u_i(t)$, the followers move into the convex hull formed by the leaders asymptotically.

The following assumptions and lemmas are used later.

**Assumption 1.** The communication graph among the followers is undirected. For each follower, there exists at least one leader that has a path to it.

**Lemma 1** ([13]). Under assumption 1, the matrix $L_2$ defined in (1) is symmetric positive definite.

From Lemma 1 we have all the eigenvalues of $L_2$ are positive. Assume that the eigenvalues of $L_2$ are $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_N$.

**Lemma 2.** Each entry of $-L_2^{-1}L_1$ is nonnegative and each row sum of $-L_2^{-1}L_1$ is equal to one.
**Lemma 3.** Assume that the matrix pair \((A, B)\) is controllable, and all the poles of \(A\) are on the imaginary axis. For any constant scalar \(\gamma > 0\), the parametric Riccati equation

\[
A^TP + PA - PBB^TP = -\gamma P
\]

has a unique positive definite matrix \(P(\gamma) = W^{-1}(\gamma)\), where \(W(\gamma)\) is the unique positive definite solution to the following Lyapunov equation \(W(A + \frac{1}{2} I_n)^T + (A + \frac{1}{2} I_n)W = BB^T\). Moreover, \(\lim_{\gamma \to 0^+} P(\gamma) = 0\), \(\frac{d}{d\gamma} P(\gamma) > 0\), \(\forall \gamma > 0\), \(tr(B^TP(\gamma)B) = n\gamma\), \(P(\gamma)BB^TP(\gamma) \leq n\gamma P(\gamma)\), \(e^{At}P(\gamma)e^{At} \leq e^{(n-1)\gamma t}P(\gamma)\). Moreover, if all the eigenvalues of \(A\) are zero, then \(A^TP(\gamma)A \leq 3(n\gamma)^2P(\gamma)\).

### 2.3. Event-Triggered Communication Mechanisms

As will be explained later, since control inputs of the leaders are zero, the informed followers can estimate states of the leaders based on their initial states. Therefore, the event-triggered mechanism is not needed for the communication between the leaders and the informed followers.

Assume that followers \(i\) and \(j\) are two linked agents, and they exchange information at event instants \(\{t^k_{ij}, t'_{ij}, t^k_{ji}, t'_{ji}\} \cdots\). To exchange information based on the event-triggering communication mechanism, event triggering functions \(f_{ij}(t)\) and \(f_{ji}(t)\) should be designed for each edge \((j,i)\). The event instants between followers \(i\) and \(j\) are determined according to

\[
t_{ij}^{k+1} = t_{ij}^{k+1} = \inf\{t \mid t > t_{ij}^k, f_{ij}(t) \leq 0 \text{ or } f_{ji}(t) \leq 0\}, j \in \mathcal{N}_i
\]

where \(t_{ij}^0 = t_{ji}^0 = 0\). Once \(f_{ij}(t)\) reaches zero, an event is triggered. Follower \(i\) will send \(x_i(t)\) to follower \(j\), and then receive \(x_j(t)\) from follower \(j\). Conversely, if \(f_{ji}(t)\) reach zero first, follower \(j\) will send \(x_j(t)\) to follower \(i\), and then receive \(x_i(t)\) from follower \(i\).

The objective of this paper is to design the event-triggering functions and the control algorithms for system (2) and (3) without input delay, and the input delay system (2) and (4), respectively, such that all the followers can move to the convex hull formed by the leaders.

### 3. Event-Triggered Containment Control without Input Delay

In this section, we consider the containment control problem without input delay. Define \(x_F(t) = \begin{bmatrix} x_{M+1}^T(t), \ldots, x_{M+N}^T(t) \end{bmatrix}^T\), and \(x_L(t) = \begin{bmatrix} x_1^T(t), \ldots, x_N^T(t) \end{bmatrix}^T\). Note from Lemma 2 that if

\[
x_F(t) = -(L_2^{-1} L_1 \otimes I_n)x_L(t),
\]

then all the followers are in the convex hull formed by the leaders. For follower \(i\), define

\[
\varphi_i(t) = \sum_{j \in \mathcal{L} \cup \mathcal{F}} a_{ij}[x_i(t) - x_j(t)].
\]

If \(\varphi_i(t) = 0\) for \(i = M + 1, \ldots, M + N\), one has

\[
(L_2 \otimes I_n)x_F(t) + (L_1 \otimes I_n)x_L(t) = 0,
\]

which implies that (7) holds. Therefore, all the followers lie in the convex hull formed by the leaders if \(\varphi_i(t) = 0\) holds for \(i = M + 1, \ldots, M + N\).

Let \(e_{ij}(t) = x_i(t) - x_j(t)\) be the state difference between follower \(i\) and leader \(j\), and \(z_{ij}(t) = x_i(t) - x_j(t)\) be the state difference between followers \(i\) and \(j\). Equation (8) can be rewritten as

\[
\varphi_i(t) = \sum_{j \in \mathcal{L}} a_{ij}e_{ij}(t) + \sum_{j \in \mathcal{F}} a_{ij}z_{ij}(t), \quad i = M + 1, \ldots, M + N.
\]
For informed followers, to drive \( q_i(t) \) to zero, \( e_{ij}(t) \) and \( z_{ij}(t) \) are needed. For other followers, \( z_{ij}(t) \) is needed. However, when the event-triggering communication mechanism is adopted, such information is available only at the event instants. To solve this problem, we need the following state difference estimators to estimate \( e_{ij}(t) \) and \( z_{ij}(t) \) during the inter-event intervals.

### 3.1. Leader Edge State Difference Estimators

Suppose that informed follower \( i \) can access leader \( j \)'s state. In this case, edge \((j, i)\) is called leader edge of follower \( i \). From (2) and (3) and the definition of \( \hat{e}_{ij}(t) \) we have

\[
\hat{e}_{ij}(t) = Ae_{ij}(t) + Bu_i(t).
\]

Follower \( i \) can construct the following estimator to estimate \( e_{ij}(t) \)

\[
\hat{e}_{ij}(t) = A\hat{e}_{ij}(t) + Bu_i(t),
\]

where \( \hat{e}_{ij}(0) = e_{ij}(0) = x_i(0) - x_j(0) \). The solution of (12) is

\[
e_{ij}(t) = e^{At}e_{ij}(0) + \int_0^t e^{A(t-s)}Bu_i(s)ds.
\]

**Remark 1.** It is trivial to prove that \( \hat{e}_{ij}(t) = e_{ij}(t) \), for all \( t > 0 \). So, the informed followers can estimate \( e_{ij}(t) \) based on the initial state \( x_j(0) \), i.e., they only need to communicate with the leader one time. This is because the leaders’ control inputs are assumed to be zero. If the leaders’ control inputs are nonzero, the informed followers need to communicate with their leader neighbors frequently. Tracking dynamic leaders with an event-triggered controller is a tough problem, which will be considered in our future work.

### 3.2. Neighbor Edge State Difference Estimators

Suppose that follower \( i \) can access follower \( j \)'s state. In this case, edge \((j, i)\) is called neighbor edge of follower \( i \). From (3) we have

\[
z_{ij}(t) = Az_{ij}(t) + B[u_i(t) - u_j(t)].
\]

In the inter-event interval \((t_{ij}^k, t_{ij}^{k+1})\), \( u_j(t) \) is not available for follower \( i \). Follower \( i \) can adopt the following estimator to estimate \( z_{ij}(t) \)

\[
\hat{z}_{ij}(t) = Az_{ij}(t),
\]

where \( \hat{z}_{ij}(0) = z_{ij}(0) \) and \( \hat{z}_{ij}(t_{ij}^k) = z_{ij}(t_{ij}^k) \). The solution of (15) in inter-event interval \((t_{ij}^k, t_{ij}^{k+1})\) is

\[
\hat{z}_{ij}(t) = e^{A(t-t_{ij}^k)}z_{ij}(t_{ij}^k).
\]

Define \( \hat{z}_{ij}(t) = z_{ij}(t) - z_{ij}(t) \). Notice that followers \( i \) can correcting \( \hat{z}_{ij}(t) \) at the event instants. Therefore, \( z_{ij}(t) = 0 \) at the event instants. Denote \( t_{ij}^l \) the last event instant of edge \((i, j)\) before time \( t \). If \( t \) is the event instant, then \( t_{ij}^l = t \). From (14) and (15) we have

\[
\hat{z}_{ij}(t) = Az_{ij}(t) - B[u_i(t) - u_j(t)].
\]

It follows that

\[
\hat{z}_{ij}(t) = -\int_{t_{ij}^l}^t e^{-A(t-s)}B[u_i(s) - u_j(s)]ds.
\]
3.3. Event-Triggering Functions

Same to [29], we use the following event trigging function for edge \((j,i)\)

\[
f_{ij}(t) = b(t) - h_{ij}(t),
\]

where \(h_{ij}(t) = \| \int_{t}^{t+1} e^{A(t-s)} Bu_i(s) ds \| \) for \(t \in [t_k^{i}, t_{k+1}^{i}]\), \(b(t) > 0\) is a continuous threshold function.

In the following of this paper, we set the threshold function as \(b(t) = ae^{-ct}\), where \(a\) and \(c\) are constant positive scalars.

Remark 2. Notice that \(h_{ij}(t)\) and \(f_{ij}(t)\) are not related to any information about other agents. So, follower \(i\) can detect the event dependently. In contrast, triggering functions in [31–33] are related to neighbors’ states. When these triggering functions are used, a centralized detector is needed.

3.4. Event-Triggered Containment Control Algorithms

Based on the aforementioned state difference estimators and event-triggering function, we consider the following containment control algorithm

\[
u_i(t) = K \left\{ \sum_{j \in L} a_{ij} \hat{x}_{ij}(t) + \sum_{j \in F} a_{ij} \hat{z}_{ij}(t) \right\}
\]

\[
= K \left\{ \sum_{j \in L} a_{ij} \hat{x}_{ij}(t) + \sum_{j \in F} a_{ij} \hat{z}_{ij}(t) \right\}
\]

\[
= K \left\{ \sum_{j \in L} a_{ij} \hat{x}_{ij}(t) + \sum_{j \in F} a_{ij} \hat{z}_{ij}(t) \right\}
\]

\[
= K [\varphi_i(t) + \sum_{j \in F} a_{ij} \hat{z}_{ij}(t)]
\]

\[
= K [\varphi_i(t) + e_i(t)], i = M + 1, \cdots, M + N,
\]

where \(e_i(t) = \sum_{j \in F} a_{ij} \hat{z}_{ij}(t)\), and \(K\) is a parametric matrix to be designed. The derivative of \(\varphi_i(t)\) along the solution of (2) and (3) is

\[
\dot{\varphi}_i(t) = \sum_{j \in L, j \notin F} a_{ij} [\dot{x}_i(t) - \dot{x}_j(t)]
\]

\[
= \sum_{j \in L} a_{ij} [Ax_i(t) + Bu_i(t) - Ax_j(t)] + \sum_{j \in F} a_{ij} [Ax_i(t) + Bu_i(t) - Ax_j(t) - Bu_j(t)]
\]

\[
= A \varphi_i(t) + B \left\{ \sum_{j \in L} a_{ij} u_i(t) + \sum_{j \in F} a_{ij} u_i(t) - u_j(t) \right\}
\]

\[
= A \varphi_i(t) + BK \left\{ \sum_{j \in L} a_{ij} [\varphi_i(t) + e_i(t)] + \sum_{j \in F} a_{ij} [\varphi_i(t) + e_i(t) - \varphi_j(t) - e_j(t)] \right\}
\]

\[
= A \varphi_i(t) + BK \left\{ \sum_{j \in L} a_{ij} \varphi_i(t) + \sum_{j \in F} a_{ij} \varphi_i(t) - \varphi_j(t) \right\}
\]

\[
+ BK \left\{ \sum_{j \in L \cup F} a_{ij} e_i(t) - \sum_{j \in F} a_{ij} e_j(t) \right\}, i = M + 1, \cdots, M + N.
\]

Define \(\varphi(t) = \begin{bmatrix} \varphi_{M+1}^T(t), \cdots, \varphi_{M+N}^T(t) \end{bmatrix}^T, \hat{e}(t) = \begin{bmatrix} e_{M+1}^T(t), \cdots, e_{M+N}^T(t) \end{bmatrix}^T\).

Equation (21) can be rewritten in a compact form as

\[
\dot{\varphi}(t) = \hat{A} \varphi(t) + L_2 \otimes BK \hat{e}(t),
\]

(21)
where \( \bar{A} = I_n \otimes A + L_2 \otimes BK \). From (6), (17) and (18) we have

\[
\| \hat{z}_{ij}(t) \| = \| \int_0^t e^{A(t-s)} B[u_i(s) - u_j(s)] ds \| \\
\leq \| \int_0^t e^{A(t-s)} Bu_i(s) ds \| + \| \int_0^t e^{A(t-s)} Bu_j(s) ds \| \\
\leq 2b(t) = 2ae^{-ct}.
\]

From (21) and (23) we have

\[
\| \varphi(t) \| \leq \| \sum_{j \in F} a_{ij} \hat{z}_{ij}(t) \| \leq (N - 1) \| \bar{z}_{ij}(t) \| \leq 2(N - 1)e^{-ct}.
\]

The proposed containment control algorithm is summarized in Algorithm 1, where \( T \) is the lifespan of the system.

**Algorithm 1 Event-Triggered Containment Control Algorithm for follower i: without input delay.**

**Initialization:**

\[ k \leftarrow 0; \]

\[ \text{for } j \in L \cup N_i \text{ do} \]

\[ \quad \text{receive } x_j(0) \text{ from leader } j; \hat{e}_{ij}(0) \leftarrow x_i(0) - x_j(0); \]

\[ \text{end for} \]

\[ \text{for } j \in F \cup N_i \text{ do} \]

\[ \quad \text{send } x_i(0) \text{ to follower } j \text{ and receive } x_j(0) \text{ from follower } j; \]

\[ \quad t_{ij}^k \leftarrow 0; \hat{e}_{ij}(0) \leftarrow x_i(0) - x_j(0); f_{ij}(0) \leftarrow \alpha; \]

\[ \text{end for} \]

\[ \text{compute the controller } u_i(0) \text{ as in (20)}; \]

**Iteration:**

\[ \text{while } t < T \text{ do} \]

\[ \quad \text{for } j \in L \cup N_i \text{ do} \]

\[ \quad \quad \text{compute } \hat{e}_{ij}(t) \text{ as in (13)}; \]

\[ \quad \text{end for} \]

\[ \quad \text{for } j \in F \cup N_i \text{ do} \]

\[ \quad \quad \text{compute } f_{ij}(t) \text{ as in (18)}; \]

\[ \quad \quad \text{if } f_{ij}(t) \leq 0 \text{ then} \]

\[ \quad \quad \quad \text{send } x_i(t) \text{ to follower } j \text{ and receive } x_j(t) \text{ from follower } j; t_{ij}^k \leftarrow t; \]

\[ \quad \quad \text{end if} \]

\[ \quad \quad \text{compute } \hat{z}_{ij}(t) \text{ as in (16)}; \]

\[ \text{end for} \]

\[ \text{compute the controller } u_i(t) \text{ as in (20)}; \]

\[ \text{end while} \]
**Theorem 1.** Consider a network with M leaders (2) and N followers (3). Suppose that $(A, B)$ is controllable, the communication graph satisfies Assumption 1, and the event-triggered communication mechanism is adopted with triggering function (18). Let $K = -B^TP$, where $P$ is the solution of the following Riccati equality

$$A^TP + PA - 2\lambda_1PBB^TP + aI = 0,$$

(25)

where $\lambda_1$ is the smallest eigenvalue of $L_2$, $a > 0$ is a constant scalar. With control algorithm (20), all the followers will converge to the convex hull formed by the leaders.

**Proof.** Assume that the communication graph satisfied Assumption 1. From Lemma 1 we have $\lambda_1 > 0$. Suppose that $(A, B)$ is controllable. For any $a > 0$, Riccati equality (27) has a solution $P$. Let $K = -B^TP$, then $\dot{A}$ in (25) is equal to $I_n \otimes A - L_2 \otimes BB^T$, where $L_2$ is a symmetric positive definite matrix, we can find an orthogonal matrix $U$ such that $U^{-1}L_2U = \text{diag}\{\lambda_1, \cdots, \lambda_N\}$. It follows that $(U^{-1} \otimes I_n)A(U \otimes I_n) = I_n \otimes A - \text{diag}\{\lambda_1, \cdots, \lambda_N\} \otimes BB^T$, which implies that $A = (U^{-1} \otimes I_n)(I_n \otimes A - \text{diag}\{\lambda_1, \cdots, \lambda_N\} \otimes BB^T)(U \otimes I_n)$. Then it follows that

$$e^{\dot{A}t} = (U \otimes I_n)\text{diag}\{e^{A-\lambda_1BB^T}t, \cdots, e^{A-\lambda_NBB^T}t\}(U^{-1} \otimes I_n).$$

(26)

Because $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$, we have

$$A^TP + PA - 2\lambda_1PBB^TP + aI \leq 0, i = 1, \cdots, N.$$

By a similar process with Section 4 in [29] we can find positive scalars $\beta$ and $\delta$ such that

$$\| e^{\dot{A}t} \| \leq \beta e^{-\delta t}.$$  

(27)

From (25) and (27) we have

$$\| \varphi(t) \| \leq \beta e^{-\delta t} \| \varphi(0) \| + 2\alpha \sqrt{N\lambda_N}(N - 1) \| BK \| \int_0^t \beta e^{-\delta(t-s)} e^{-\delta s} ds$$

$$\leq \beta e^{-\delta t} \left[ \| \varphi(0) \| + 2\alpha \sqrt{N\lambda_N}(N - 1) \| BK \| \int_0^t e^{(\delta-c)s} ds \right]$$

$$\leq \beta e^{-\delta t} \left[ \| \varphi(0) \| + 2\alpha \sqrt{N\lambda_N}(N - 1) \| BK \| \frac{e^{(\delta-c)t-1}}{\delta-c} \right].$$

(28)

Therefore, $\| \varphi(t) \|$ converge to zero exponentially. Based on the analysis above we have that all the followers will converge to the convex hull formed by the leaders, which complete the proof.

Next, we show that the Zeno behavior can be excluded by the control algorithm (20) and triggering function (8).

**Theorem 2.** Suppose that $0 < c < \delta$. The inter-event intervals are lower bounded by

$$t_{ij}^{k+1} - t_{ij}^k \geq \frac{1}{\| A \| + c} \ln(1 + \| A \| + c) \| BK \|,$$

(29)

where $\theta = \| \varphi(0) \| + 2\alpha \sqrt{N\lambda_N}(N - 1) + 2(N - 1)a$.

**Proof.** With control algorithm (20), from (23) and (29) we have

$$\| Bu(t) \| = \| BK[\varphi(t) + e_i(t)] \|$$

$$\leq \| BK \| (\| \varphi(t) \| + \| e_i(t) \|)$$

$$\leq \| BK \| [\eta(t) + 2(N - 1)a e^{-ct}],$$

(30)
where \( \eta(t) = \beta e^{-\delta t} \left[ \| \phi(0) \| + 2\alpha \sqrt{N\lambda N(N - 1)} \| BK \| \frac{e^{(c-\delta)t}}{\delta - c} \right] \).

Suppose that \( 0 < c < \delta \). For any linked followers \( i \) and \( j \), assume that the communication is triggered at event instant \( t_{ij}^k \). Then, \( t_{ij}^{k+1} \) is the next event instant that satisfies \( h_{ij}(t) = ae^{-ct} \) or \( h_{ij}(t) = ae^{-ct_i^j} \). Without loss of generality, we assume that \( h_{ij}(t_{ij}^{k+1}) = ae^{-ct_{ij}^{k+1}} \).

Define \( p_{ij}(t) = \frac{\theta|BK|}{\|A\| + c} e^{-c[t_e^{ij}(\|A\|+c)(t_{ij}^{k+1}) - 1]} \). For any \( t \in [t_{ij}^k, t_{ij}^{k+1}] \), we can obtain that \( h_{ij}(t) \) satisfies

\[
h_{ij}(t) = \left\| \int_{t_{ij}^k}^t e^{At-s} Bu_i(s) ds \right\|
\leq \|BK\| e^{\|A\|t} \int_{t_{ij}^k}^t e^{-\|A\|s} [\eta(s) + 2(N - 1)ae^{-cs}] ds
= \|BK\| e^{\|A\|t} \int_{t_{ij}^k}^t e^{-\|A\|s} [\beta e^{-\delta s} \| \phi(0) \| + 2\alpha \beta \sqrt{N\lambda N(N - 1)} \frac{e^{-cs} - e^{-\delta s}}{\delta - c} + 2(N - 1)ae^{-cs}] ds
\leq \|BK\| e^{\|A\|t} \int_{t_{ij}^k}^t e^{-\|A\|s} \{\beta e^{-cs} \| \phi(0) \| + 2\alpha \beta \sqrt{N\lambda N(N - 1)} \frac{e^{-cs}}{\delta - c} + 2(N - 1)ae^{-cs}\} ds
= \theta \|BK\| e^{\|A\|t} \int_{t_{ij}^k}^t e^{-([\|A\|+c])s} ds
= p_{ij}(t).
\]

It follows that \( p_{ij}(t_{ij}^{k+1}) \geq ae^{-ct_{ij}^{k+1}} \). Notice that \( p_{ij}(t_{ij}^k) = 0 < e^{-ct_{ij}^k} \). We can obtain that there exists a time instant \( t_p \in (t_{ij}^k, t_{ij}^{k+1}) \) such that \( p_{ij}(t_p) = e^{-ct_p} \). From the definition of \( p_{ij}(t) \) we have

\[
\frac{\theta}{\|A\| + c} e^{-ct_p} [e^{([\|A\|+c])t_p-\beta t_{ij}^k} - 1] = e^{-ct_p},
\]
which implies that

\[
t_{ij}^{k+1} - t_{ij}^k \geq t_p - t_{ij}^k = \frac{1}{\|A\| + c} \ln(1 + \frac{\|A\| + c}{\theta \|BK\|}).
\]

4. Containment Control with Input Delay

In this section, we consider the containment control problem for system (2) and (4). Same with the delay-free case that is considered in the last section, the following state difference estimators are needed.

4.1. Leader Edge State Difference Estimators

Suppose that follower \( i \) can access leader \( j \)'s state. Follower \( i \) can adopt the following estimator to estimate \( e_{ij}(t) \)

\[
\hat{e}_{ij}(t) = A\hat{e}_{ij}(t) + Bu_i(t - \tau),
\]
where \( \hat{e}_{ij}(0) = e_{ij}(0), u_i(t) = 0, t \in [-\tau, 0) \). The solution of (34) is

\[
\hat{e}_{ij}(t) = e^{At}\hat{e}_{ij}(0) + \int_0^t e^{A(t-s)} Bu_i(s - \tau) ds.
\]
4.2. Neighbor Edge State Difference Estimators

Suppose that follower $i$ can access follower $j$’s state. Note from (4) that

$$z_{ij}(t) = Az_{ij}(t) + B[u_i(t - \tau) - u_j(t - \tau)].$$

(36)

In the inter-event interval $(t^k_{ij}, t^{k+1}_{ij})$, $u_j(t)$ is not available for follower $i$. Follower $i$ can use the following estimator to estimate $z_{ij}(t)$

$$\hat{z}_{ij}(t) = A\hat{z}_{ij}(t),$$

(37)

where $\hat{z}_{ij}(0) = z_{ij}(0)$ and $\hat{z}_{ij}(t^k_{ij}) = z_{ij}(t^k_{ij})$. From (37) we have

$$\hat{z}_{ij}(t) = e^{A(t-t^k_{ij})}z_{ij}(t^k_{ij}), \ t \in [t^k_{ij}, t^{k+1}_{ij}).$$

(38)

From (36) we have

$$z_{ij}(t) = e^{A(t-t^k_{ij})}z_{ij}(t^k_{ij}) + \int_{t^k_{ij}}^t e^{A(t-s)}B[u_i(s - \tau) - u_j(s - \tau)]ds.$$  

(39)

It follows that

$$\hat{z}_{ij}(t) = z_{ij}(t) - z_{ij}(t) = \int_{t^k_{ij}}^t e^{A(t-s)}B[u_i(s - \tau) - u_j(s - \tau)]ds.$$  

(40)

4.3. Event-Triggering Functions

Notice that when there exists a constant input delay $\tau$, the estimating error $\hat{z}_{ij}(t)$ depends on $u_i(s - \tau) - u_j(s - \tau)$, which is different from the delay-free case. To achieve the control objective, the event trigging function should be adjusted accordingly. We use the following event trigging function for the edge $(j, i)$

$$f_{ij}(t) = b(t) - h_{ij}(t),$$

(41)

where $h_{ij}(t) = \| \int_{t^k_{ij}}^t e^{A(t-s)}Bu_i(s - \tau)ds \| \text{ for } t \in [t^k_{ij}, t^{k+1}_{ij})$, $b(t)$ is given in the last section.

Remark 3. Notice that the event triggering function $f_{ij}(t)$ depends on the input of agent $i$ only, which is different from the event trigging function in [43]. In [43], the event-triggered condition (19) is related to $\sum_{j \in L \cup F} a_{ij} |x_j(t) - x_i(t)|$. Since agent $i$ cannot access $x_j(t)$ during the inter-event interval, this signal is not available for agent $i$. So, the event trigging function in this paper is more feasible than that in [43].

4.4. Event-Triggered Containment Control Algorithms

Based on the state difference estimators and the event-triggering function, we consider the following containment algorithm

$$u_i(t) = Ke^{At} \left[ \sum_{j \in L} a_{ij} \hat{e}_{ij}(t) + \sum_{j \in F} a_{ij} \hat{z}_{ij}(t) \right], \ i = M + 1, \ldots, M + N.$$  

(42)
With the input delay $\tau$, (42) becomes

$$u_i(t - \tau) = Ke^{At}\left[\sum_{j \in L} a_{ij}e_{ij}(t - \tau) + \sum_{j \in F} a_{ij}e_{ij}(t - \tau)\right]$$

$$= Ke^{At}\left[\sum_{j \in L} a_{ij}e_{ij}(t - \tau) + \sum_{j \in F} a_{ij}e_{ij}(t - \tau)\right] - Ke^{At}\sum_{j \in F} a_{ij}e_{ij}(t - \tau)$$

$$= Ke^{At}q_i(t - \tau) - Ke^{At}\sum_{j \in F} a_{ij}e_{ij}(t - \tau)$$

$$= K\left\{q_i(t) - \int_{t-\tau}^{t} \left\{ \sum_{j \in L} a_{ij}e^{A(t-s)}Bu_j(s)\right\} ds - \sum_{j \in F} a_{ij}e^{A(t-s)}Bu_j(s)ds,\right\}$$

$$-Ke^{At}\sum_{j \in F} a_{ij}\int_{t_{ij}^-}^{t-\tau} e^{A(t-t_{ij}^-)}Bu_j(s)ds,$$

where $t_{ij}^-\tau$ is the last event instant on edge $(i, j)$ before $t - \tau$, and $t_{ij}^-\tau = t - \tau$ if $t - \tau$ is an event instant.

The proposed containment control algorithm is summarized in Algorithm 2.

**Algorithm 2 Event-Triggered Containment Control Algorithm for Agent i: with input delay.**

**Initiation:**

1. $k \leftarrow 0$;
2. for $j \in L \cup N_i$ do
   a. receive $x_j(0)$ from leader $j$; $\hat{e}_{ij}(0) \leftarrow x_i(0) - x_j(0)$;
3. end for
4. for $j \in F \cup N_i$ do
   a. send $x_i(0)$ to follower $j$; $t_k^j \leftarrow 0$; $\hat{e}_{ij}(0) \leftarrow x_i(0) - x_j(0)$; $f_{ij}(0) \leftarrow \alpha$;
5. end for
6. compute the controller $u_i(0)$ as in (42);

**Iteration:**

1. while $t < T$ do
2. for $j \in L \cup N_i$ do
3. compute $\hat{e}_{ij}(t)$ as in (35);
4. end for
5. for $j \in F \cup N_i$ do
6. compute $f_{ij}(t)$ as in (41);
7. if $f_{ij}(t) \leq 0$ then
   a. send $x_i(t)$ to follower $j$ and receive $x_j(t)$ from follower $j$; $t_k^j \leftarrow t$;
8. end if
9. compute $\hat{z}_{ij}(t)$ as in (38);
10. end for
11. compute the controller $u_i(t)$ as in (42).
12. end while

**Theorem 3.** Consider a network with $M$ leaders (2) and $N$ followers (4), with a communication graph satisfies Assumption 1. Suppose that $(A, B)$ is controllable, and all the poles of $A$ are on the imaginary axis. Suppose that the event-triggered communication mechanism is adopted with triggering function (41). Let $K = -\mu B^TP(\gamma)$, where $P(\gamma)$ is the solution of the Riccati inequality (5), $2\mu \lambda_1 > 1$. For any $\tau > 0$, there exists a scalar $\gamma^*$ such that for any $\gamma \in [0, \gamma^*)$, with control algorithm (42), all the followers will converge to the convex hull formed by the leaders.
Proof. Suppose that \((A, B)\) is controllable, and all the poles of \(A\) are on the imaginary axis. For any \(\gamma > 0\), from Lemma 3 we have there exist a solution \(P(\gamma)\) satisfies the Riccati inequality (5). In the rest of this paper, we omit \(\gamma\) for simplicity. Let \(K = -\mu B^TP\). Consider the following function
\[
V_1(t) = \sum_{i \in F} \phi_i^T(t)P\phi_i(t). \tag{44}
\]

From (4) and (8) we have
\[
\dot{\phi}_i(t) = A\phi_i(t) + B\left(\sum_{j \in \mathcal{L}} a_{ij}u_i(t - \tau) + \sum_{j \in \mathcal{F}} a_{ij}[u_i(t - \tau) - u_j(t - \tau)]\right). \tag{45}
\]

The derivative of \(V_1(t)\) along the solution of (45) satisfies
\[
\dot{V}_1(t) = 2 \sum_{i \in \mathcal{F}} \phi_i^T(t)P\phi_i(t)
= 2 \sum_{i \in \mathcal{F}} \phi_i^T(t)P\left\{A\phi_i(t) + B\left(\sum_{j \in \mathcal{L}} a_{ij}u_i(t - \tau) + \sum_{j \in \mathcal{F}} a_{ij}[u_i(t - \tau) - u_j(t - \tau)]\right)\right\},
\]
where \(u(t - \tau) = \begin{bmatrix} u_{M+1}(t - \tau), & \cdots, & u_{M+N}(t - \tau) \end{bmatrix}^T\). From (44) we have
\[
u(t - \tau) = (\mu I_N \otimes B^TP)[-\varphi(t) + \int_{t-\tau}^t (L_2 \otimes e^{A(t-s)}B)u(s - \tau)ds + \int_{t_{ij}-\tau}^{t_{ij}} (L_2 \otimes e^{A(t-s)}B)u(s - \tau)ds], \tag{47}\]
which, together with (47), yielding that
\[
\dot{V}_1(t) = \varphi^T(t)(I_N \otimes (PA + A^TP))\varphi(t) - 2(\mu L_2 \otimes PB^TP)\varphi(t)
+ 2\varphi^T(t)(\mu L_2 \otimes PB^TP)\int_{t-\tau}^t (L_2 \otimes e^{A(t-s)}B)u(s - \tau)ds
+ 2\varphi^T(t)(\mu L_2 \otimes PB^TP)\int_{t_{ij}-\tau}^{t_{ij}} (L_2 \otimes e^{A(t-s)}B)u(s - \tau)ds
\]
\[
= \varphi^T(t)(I_N \otimes (PA + A^TP))\varphi(t)
+ 2\varphi^T(t)(\mu L_2 \otimes PB^TP)\int_{t-\tau}^t (I_N \otimes e^{A(t-s)}B)u(s - \tau)ds
+ 2\varphi^T(t)(\mu L_2 \otimes PB^TP)\int_{t_{ij}-\tau}^{t_{ij}} (I_N \otimes e^{A(t-s)}B)u(s - \tau)ds. \tag{48}\]

Also, from (44) we have
\[
u(t - \tau) = (-\mu I_N \otimes B^TP\varphi(t - \tau) - \varphi(t - \tau)]. \tag{49}\]
It follows that

\[
\dot{V}_1(t) = \varphi^T(t)[I_N \otimes (PA + A^T P) - 2(\mu L_2 \otimes PBB^T P)]\varphi(t) \\
-2\varphi^T(t)(\mu^2 L_2^2 \otimes PBB^T P) \int_{t-\tau}^{t} (I_N \otimes e^{A(t-s)}BB^T Pe^{At})[\varphi(s - \tau) - e(s - \tau)] ds \\
+2\varphi^T(t)(\mu L_2 L_2 \otimes PBB^T P) \int_{t-\tau}^{t} (I_N \otimes e^{A(t-s)}B)u(s - \tau) ds \tag{50}
\]

\[
\leq \varphi^T(t)[I_N \otimes (PA + A^T P) - 2(\mu L_2 \otimes PBB^T P)]\varphi(t) \\
+\varphi^T(t)[(2k_2^2 L_2^2 + k_3^2 L_2^2 L_2) \otimes PBB^T P]\varphi(t) \\
+\frac{m\gamma}{k} \{\pi_1^T(t)(I_N \otimes P)\pi_1(t) + \pi_2^T(t)(I_N \otimes P)\pi_2(t) + \pi_3^T(t)(I_N \otimes P)\pi_3(t)\},
\]

where we have used the fact that \(PBB^T P \leq n\gamma P\), which is obtained from Lemma 3, \(I_2\) is the Laplacian matrix corresponding to the communication topology among the followers, and

\[
\pi_1(t) = \int_{t-\tau}^{t} (I_N \otimes e^{A(t-s)}BB^T Pe^{At})\varphi(s - \tau) ds,
\]

\[
\pi_2(t) = \int_{t-\tau}^{t} (I_N \otimes e^{A(t-s)}BB^T Pe^{At})e(s - \tau) ds,
\]

\[
\pi_3(t) = \int_{t-\tau}^{t} (I_N \otimes e^{A(t-s)}B)u(s - \tau) ds.
\]

Define \(\xi(t) = (U^{-1} \otimes I_n)\varphi(t)\). It follows from (51) and (5) that

\[
V_1(t) \leq \xi^T(t)[I_N \otimes (PA + A^T P) - 2(\mu U^T L_2 U \otimes PBB^T P)]\xi(t) \\
+ \xi^T(t)[(2k_2^2 U^T L_2^2 U + k_3^2 U^T L_2^2 L_2) \otimes PBB^T P]\xi(t) \\
+ \frac{m\gamma}{k} \{\tilde{\pi}_1^T(t)(I_N \otimes P)\tilde{\pi}_1(t) + \tilde{\pi}_2^T(t)(I_N \otimes P)\tilde{\pi}_2(t) + \tilde{\pi}_3^T(t)(I_N \otimes P)\tilde{\pi}_3(t)\} \tag{51}
\]

\[
\leq \xi^T(t)[I_N \otimes (-\gamma)P - (2\lambda_1 \mu - 1)I_N \otimes PBB^T P]\xi(t) \\
+ \xi^T(t)[(2k_2^2 U^T L_2^2 U + k_3^2 U^T L_2^2 L_2) \otimes PBB^T P]\xi(t) \\
+ \frac{m\gamma}{k} \{\tilde{\pi}_1^T(t)(I_N \otimes P)\tilde{\pi}_1(t) + \tilde{\pi}_2^T(t)(I_N \otimes P)\tilde{\pi}_2(t) + \tilde{\pi}_3^T(t)(I_N \otimes P)\tilde{\pi}_3(t)\}
\]

where

\[
\tilde{\pi}_1(t) = \int_{t-\tau}^{t} (U \otimes e^{A(t-s)}BB^T Pe^{At})\xi(s - \tau) ds.
\]

By Jensen inequality and Lemma 3 we have

\[
\tilde{\pi}_1^T(t)(I_N \otimes P)\tilde{\pi}_1(t) \\
\leq \tau \int_{t-\tau}^{t} \xi^T(s - \tau)(U^T U \otimes e^{A^T PBB^T e^{A(t-s)}Pe^{A(t-s)}BB^T Pe^{A^T}})\xi(s - \tau) ds \\
\leq \tau \int_{t-\tau}^{t} \xi^T(s - \tau)(I_N \otimes e^{(n-1)\gamma(t-s)}e^{A(t-s)}BB^T PBB^T Pe^{A^T})\xi(s - \tau) ds \\
\leq \tau \int_{t-\tau}^{t} \xi^T(s - \tau)(I_N \otimes n^2 \gamma^2 e^{(n-1)\gamma(t-s)}e^{A(t-s)}Pe^{A^T})\xi(s - \tau) ds \\
\leq \tau \int_{t-\tau}^{t} \xi^T(s - \tau)(I_N \otimes n^2 \gamma^2 e^{(n-1)\gamma(t-s)}e^{A(t-s)}Pe^{A^T})\xi(s - \tau) ds \\
\leq \tau n^2 \gamma^2 e^{(n-1)\gamma(t-s)} \int_{t-\tau}^{t} \xi^T(s - \tau)(I_N \otimes P)\xi(s - \tau) ds \\
\leq \tau n^2 \gamma^2 e^{(n-1)\gamma(t-s)} \int_{t-\tau}^{t} \xi^T(s)(I_N \otimes P)\xi(s) ds,
\]

where
Therefore it follows that
\[
\begin{align*}
\pi^T_2(t)(I_N \otimes P)\pi_2(t) & \leq \tau n^2 \gamma^2 e^{2(n-1)\gamma t} \int_{t-2\tau}^t e^T(s) (I_N \otimes P)e(s) ds \\
& \leq \lambda_{\max}(P) \tau n^2 \gamma^2 e^{2(n-1)\gamma t} \int_{t-2\tau}^t e^T(s) (I_N \otimes P)e(s) ds \\
& \leq \lambda_{\max}(P) \tau n^2 \gamma^2 e^{2(n-1)\gamma t} \int_{t-2\tau}^t \alpha NN e^{-2\alpha s} ds \\
& \leq 2\alpha NN_1 \lambda_{\max}(P) \tau n^2 \gamma^2 e^{2(n-1)\gamma t} \cdot e^{-2\gamma(t-2\tau)}.
\end{align*}
\]  
From the definition of the event-triggering function (41) we have
\[
\pi^T_2(t)\pi_3(t) \leq \alpha^2 e^{-2\gamma(t-\tau)}.
\]  
Define
\[
V_2(t) = \frac{n^2 \gamma^2 e^{2(n-1)\gamma t}}{k} \int_{0}^{2\tau} \int_{t-s}^{t} \xi^T(l)(I_N \otimes P)\xi(l)dl ds.
\]
We have
\[
V_2(t) = 2\tau \frac{n^2 \gamma^2 e^{2(n-1)\gamma t}}{k} \xi^T(t)(I_N \otimes P)\xi(t) - \frac{n^2 \gamma^2 e^{2(n-1)\gamma t}}{k} \int_{t-2\tau}^{t} \xi^T(s)(I_N \otimes P)\xi(s) ds.
\]  
Consider the following Lyapunov function
\[
V(t) = V_1(t) + V_2(t).
\]  
From (52)–(55) we have
\[
\dot{V}(t) \leq \xi^T(t)[I_N \otimes (-\gamma P) - (2\lambda_1 \mu - 1)I_N \otimes PBB^TP]\xi(t) + \xi^T(t)[2\mu^2 U^TL_2^2U + k^2 U^TL_2L_2U] \otimes PBB^T]\xi(t) + 2\gamma n^2 \lambda_{\max}(P) \tau^2 \gamma^2 e^{2(n-1)\gamma t} \cdot e^{-2\gamma(t-2\tau)} + \frac{n^2 \gamma^2}{k} \lambda_{\max}(P) \alpha^2 \cdot e^{-2\gamma(t-2\tau)}.
\]  
Because $2\lambda_1 \mu > 1$, there exists a small enough $k$ such that
\[
2\mu^2 U^TL_2^2U + k^2 U^TL_2L_2U \leq (2\lambda_1 \mu - 1)I_N.
\]  
Then, we have
\[
\dot{V}(t) \leq -(1 - 2\tau \frac{n^2 \gamma^2 e^{2(n-1)\gamma t}}{k} \cdot \gamma^2 T(t)[I_N \otimes P]\xi(t) + \frac{n^2 \gamma^2}{k} 2\alpha NN_1 \lambda_{\max}(P) \tau^2 \gamma^2 e^{2(n-1)\gamma t} \cdot e^{-2\gamma(t-2\tau)} + \frac{n^2 \gamma^2}{k} \lambda_{\max}(P) \alpha^2 \cdot e^{-2\gamma(t-2\tau)}.
\]  
For any $\tau > 0$, there exists a $\gamma^*$ such that
\[
1 - 2\tau \frac{n^2 \gamma^2 e^{2(n-1)\gamma t}}{k} > \frac{1}{2}
\]  
for any $\gamma \in [0, \gamma^*)$. Therefore it follows that
\[
\dot{V}(t) \leq -\frac{\gamma^*}{2} \xi^T(t)[I_N \otimes P]\xi(t) + \frac{n^2 \gamma^2}{k} 2\alpha NN_1 \lambda_{\max}(P) \tau^2 \gamma^2 e^{2(n-1)\gamma t} \cdot e^{-2\gamma(t-2\tau)} + \frac{n^2 \gamma^2}{k} \lambda_{\max}(P) \alpha^2 \cdot e^{-2\gamma(t-2\tau)}.
\]
Notice that the last two terms converge to zero exponentially, we conclude that $\xi(t)$ converge to zero asymptotically, which implies that the followers converge to the convex hull formed by the leaders. \hfill $\Box$

**Remark 4.** From Theorem 3 we can see that for arbitrarily large but bounded delays, all the followers will converge to the convex hull formed by the leaders. In contrast, the algorithms in [43] require that the input delay is smaller than an upper bound.

5. Simulation Examples

This section gives a numerical example to show the effectiveness of the proposed algorithms. Consider a network with 10 followers and 4 leaders in the two-dimensional space. The parameter matrices $A$ and $B$ are

$$
A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},
B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.
$$

Suppose that $x_1(t) = [1 + 0.8t, 1 + 0.15t]^T$, $x_2(t) = [5 + 0.75t, 1 + 0.15t]^T$, $x_3(t) = [4 + 0.8t, 6 + 0.1t]^T$ and $x_4(t) = [8 + 0.75t, 6 + 0.1t]^T$. Initial positions of the followers are chosen as $x_5(0) = [1, -1]^T$, $x_6(0) = [2, -1]^T$, $x_7(0) = [3, -1]^T$, $x_8(0) = [-3, 3]^T$, $x_9(0) = [0, -3]^T$, $x_{10}(0) = [5, -3]^T$, $x_{11}(0) = [8, -3]^T$, $x_{12}(0) = [2, -7]^T$, $x_{13}(0) = [5, -7]^T$, and $x_{14}(0) = [9, -8]^T$. Initial velocities of the followers are chosen as zero.

The network graph associated with the 14 agents is shown by Figure 1, and the corresponding Laplacian matrices are as following

$$
L_1 = \begin{bmatrix} -1 & -1 & -1 & 0 \\ 0 & -1 & -1 & -1 \\ 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},
L_2 = \begin{bmatrix} 5 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 6 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 6 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & -1 & 0 & -1 & 5 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 3 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \end{bmatrix}.
$$

It is easy to obtain that the smallest eigenvalue of $L_2$ is $\lambda_1 = 0.2845$.

**Containment control without input delay.** We verify the effectiveness of containment control algorithm (20) and trigger function (18) first. Chose $a = 1$. By solving matrix equality (27) we can obtain $P = \begin{bmatrix} 1.5839 & 0 & 0.7543 & 0 \\ 0 & 1.5839 & 0 & 0.7543 \\ 0.7543 & 0 & 1.1947 & 0 \\ 0 & 0.7543 & 0 & 1.1947 \end{bmatrix}$. According to Theorem 1, we can set $K$ as

$$
K = -B^T P = \begin{bmatrix} 0.7543 & 0 & 1.1947 & 0 \\ 0 & 0.7543 & 0 & 1.1947 \end{bmatrix}.
$$
Figure 1. The network topology associated with vehicles 1 to 14. Here $i$ denotes vehicle $i, i = 1, \ldots, 14$.

Let $\alpha = 1, c = 0.01$. Figure 2 shows positions of vehicles 1 to 14 at time instants 0 s, 5 s, 10 s, and 15 s. It can be seen that agents 5 to 14 move into the convex hull spanned by agents 1 to 4. Figure 3 shows the event instants on edges 1 to 12. Tables 1 and 2 show numbers of event instants counted by edge and agent, respectively. From Table 2 we can see that for most agents, the communication burden is light. However, for agents with many neighbor edges (agents 9–11), their communication burdens are heavy.

**Containment control with input delay.** Let $\tau = 0.1$ s. By solving parametric Riccati Equation (5)

with $\gamma = 0.001$, we can obtain $P = \begin{bmatrix} 0.0080 & 0 & 0.0316 & 0 \\ 0 & 0.0080 & 0 & 0.0316 \\ 0.0316 & 0 & 0.2535 & 0 \\ 0 & 0.0316 & 0 & 0.2535 \end{bmatrix}$. Let $\mu = 2$. According to Theorem 3, we can set $K$ as

$$K = -\mu B^T P = \begin{bmatrix} 0.0632 & 0 & 0.5069 & 0 \\ 0 & 0.0632 & 0 & 0.5069 \end{bmatrix}.$$

<table>
<thead>
<tr>
<th>edge</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of event instants</td>
<td>9</td>
<td>19</td>
<td>8</td>
<td>40</td>
<td>40</td>
<td>16</td>
<td>26</td>
<td>31</td>
<td>25</td>
<td>39</td>
<td>38</td>
<td>23</td>
</tr>
</tbody>
</table>

**Table 1. Number of event instants of each edge (\(\tau = 0, 15\) s).**

<table>
<thead>
<tr>
<th>agent</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of event instants</td>
<td>28</td>
<td>57</td>
<td>64</td>
<td>26</td>
<td>101</td>
<td>198</td>
<td>78</td>
<td>25</td>
<td>38</td>
<td>23</td>
</tr>
</tbody>
</table>

**Table 2. Number of event instants of each agent (\(\tau = 0, 15\) s).**
Figure 2. Positions of agents 1 to 14 using Algorithm 1, where squares denote leaders and circles denote followers.

Figure 3. Event instants on edges 1–12 ($\tau = 0, 15$ s).

Figure 4 shows positions of agents 1 to 14 using Algorithm 2. It can be seen that agents 5 to 14 move into the convex hull spanned by vehicles 1 to 4. Figure 5 shows the event instants on edges 1 to 12. Tables 3 and 4 show numbers of event instants counted by edge and by agent, respectively.
Figure 4. Positions of agents 1 to 14 using Algorithm 2 with $\tau = 0.1$, where squares denote leaders and circles denote followers.

(c) Positions of agents 1 to 14 at $t = 20$ s.

(d) Positions of agents 1 to 14 at $t = 30$ s.

Figure 5. Event instants on edges 1–12 ($\tau = 0.1$, 30 s).

Table 3. Number of event instants of each edge ($\tau = 0.1$, 30 s).

<table>
<thead>
<tr>
<th>edge</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of event instants</td>
<td>8</td>
<td>24</td>
<td>12</td>
<td>17</td>
<td>23</td>
<td>19</td>
<td>26</td>
<td>28</td>
<td>26</td>
<td>25</td>
<td>23</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 4. Number of event instants of each agent ($\tau = 0.1$, 30 s).

<table>
<thead>
<tr>
<th>agent</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of event instants</td>
<td>32</td>
<td>37</td>
<td>54</td>
<td>26</td>
<td>114</td>
<td>116</td>
<td>44</td>
<td>26</td>
<td>23</td>
<td>20</td>
</tr>
</tbody>
</table>
Suppose that $\tau = 0.5$ s. By solving parametric Riccati Equation (5) with $\gamma = 0.0002$, we can obtain

$$
P = \begin{bmatrix}
0.0024 & 0 & 0.0141 & 0 \\
0 & 0.0024 & 0 & 0.0141 \\
0.0141 & 0 & 0.1688 & 0 \\
0 & 0.0141 & 0 & 0.1688
\end{bmatrix}.
$$

Let $\mu = 2$. According to Theorem 3, we can set $K$ as

$$
K = -\mu B^T P = \begin{bmatrix}
0.0283 & 0 & 0.3375 & 0 \\
0 & 0.0283 & 0 & 0.3375
\end{bmatrix}.
$$

Figure 6 shows positions of agents 1 to 14. It can be seen that agents 5 to 14 move into the convex hull spanned by vehicles 1 to 4. Figure 7 shows the event instants on edges 1 to 12. Tables 5 and 6 show numbers of event instants counted by edge and by agent, respectively.

**Remark 5.** From Figures 4 and 6 we can see that the converge speed is lower when $\tau$ is larger. In fact, when the input delay $\tau$ is getting larger, according to Theorem 3, we should choose a smaller $\gamma$. From Lemma 3 we know that $\lim_{\gamma \to 0^+} P(\gamma) = 0$. So, when $\gamma$ is small, the gain matrix $K$ is small, which leads to low converge speed.
Figure 7. Event instants on edges 1–12 ($\tau = 0.5$, 50 s).

Table 5. Number of event instants of each edge ($\tau = 0.1$, 50 s).

<table>
<thead>
<tr>
<th>edge</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of event instants</td>
<td>10</td>
<td>25</td>
<td>10</td>
<td>77</td>
<td>80</td>
<td>23</td>
<td>24</td>
<td>81</td>
<td>24</td>
<td>81</td>
<td>80</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 6. Number of event instants of each agent ($\tau = 0.5$, 50 s).

<table>
<thead>
<tr>
<th>agent</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of event instants</td>
<td>35</td>
<td>97</td>
<td>113</td>
<td>24</td>
<td>154</td>
<td>399</td>
<td>104</td>
<td>24</td>
<td>80</td>
<td>22</td>
</tr>
</tbody>
</table>

6. Conclusions

The event-triggered containment control problem has been considered in this paper, and two control algorithms have been proposed for multi-agent systems without and with input delay, respectively. Continuous communication has been avoided in the proposed triggering conditions. With the proposed algorithms, we have proved that all the followers can be driven into the convex hull formed by the leaders. When there is not input delay, it has been proved that the Zeno behavior can be avoided. Compared with some existing algorithms for the containment control problem with input delay, the proposed algorithm can deal with arbitrarily large input delay. The proposed triggering conditions can be detected dependently by each agent, while existing triggering conditions in those literatures on event-triggered containment control need to be realized in a centralized way.

Author Contributions: Conceptualization, J.L., C.L., X.Y. and W.C.; methodology, J.L.; software, C.L.; validation, J.L., X.Y. and W.C.; formal analysis, J.L.; data curation, C.L.; writing—original draft preparation, J.L. and C.L.; writing—review and editing, J.L.; funding acquisition, J.L.

Funding: This work was supported by National Natural Science Foundation of China under grant 61503162, Natural Science Fundamental Research Project of Jiangsu Colleges and Universities under Grant 18KJB520010.

Conflicts of Interest: The authors declare no conflict of interest.

References


8. Cao, Y.; Ren, W.; Egerstedt, M. Distributed containment control with multiple stationary or dynamic leaders in fixed and switching directed networks. *Automatica* 2012, 48, 1586–1597. [CrossRef]


11. Meng, Z.; Ren, W.; You, Z. Distributed finite-time attitude containment control for multiple rigid bodies. *Automatica* 2010, 46, 2092–2099. [CrossRef]


15. Li, Z.; Ren, W.; Liu, X.; Fu, M. Distributed containment control of multi-agent systems with general linear dynamics in the presence of multiple leaders. *Int. J. Robust Nonlinear Control* 2013, 23, 534–547. [CrossRef]


41. Zhou, B.; Lin, Z. Consensus of high-order multi-agent systems with large input and communication delays. *Automatica* 2014, 50, 452–464. [CrossRef]