To appear in *Advanced Robotics*
Vol. 00, No. 00, January 2013, 1–14

FULL PAPER

Virtual Merge and Split at Intersection for Vehicle Platooning Based on Self-Triggered Pinning Consensus Control

Ayaka Tanaka[a], Koichi Kobayashi[a]*, and Yuh Yamashita[a]

[a] *Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan*

(*v1.0 released January 2013*)

In this paper, a new method of vehicle platooning at an intersection is proposed based on self-triggered pinning consensus control. Using the proposed method, collision avoidance is achieved with no vehicles stopping/backing. First, the outline of self-triggered pinning consensus control is explained. Next, the problem setting of vehicle platooning is given, and virtual merge and split of vehicle groups are proposed. Furthermore, performance analysis of self-triggered pinning consensus control for vehicle platooning at an intersection is conducted. Finally, a numerical simulation is presented to demonstrate the proposed method.

Keywords: consensus; pinning control; self-triggered control; vehicle platooning; virtual merge and split

1. Introduction

The consensus problem of multi-agent systems is to find a control input such that each agent reaches a particular ordered state by using only neighborhood agents' information (see, e.g., [1–4]). In the conventional consensus problem, the state of each agent converges to the average of the initial states of all agents. To achieve consensus on the target value, it is important to consider external inputs. From this viewpoint, pinning consensus control has been proposed (see, e.g., [4–6]). Pinning control is a method that the external control input is added to some agents (pinning agents), e.g., leaders [7, 8]. In [4, 5], pinning consensus control using model predictive control (MPC) has been proposed. MPC is a control method that the control input is generated by solving the finite-time optimal control problem at each discrete time (see, e.g., [9, 10]).

In the case of networked control systems, pinning consensus control has technical issues on the amount of communication. This is because the external control input in pinning consensus control using MPC is frequently calculated by a centralized controller, and is sent to some agents [4, 5]. Event-triggered and self-triggered control methods have been proposed as a method to avoid congestion in communication networks (see, e.g., [11]). In event-triggered control, the control input is updated only when a certain condition on measured values is satisfied. In self-triggered control, both the control input and the next update time are calculated using the current measured values. Event-triggered consensus control has been studied in e.g., [2, 12, 13]. Self-triggered consensus control has been studied in e.g., [2, 14–16]. Moreover, event-triggered pinning consensus control has been studied in e.g., [17, 18]. Self-triggered pinning consensus control has been studied in e.g., [19].

*Corresponding author. Email: k-kobaya@ssi.ist.hokudai.ac.jp

On the other hand, vehicle platooning has attracted much attention as an application of control theory (see, e.g., [6, 20–28]). In vehicle platooning, consensus on the inter-vehicular distance between vehicles is important. Several methods on applications of consensus control to vehicle platooning have been studied (see, e.g., [6, 21–23, 27, 28]). Furthermore, in [6], merge and split of vehicle platoons have also been studied.

In this paper, as an application of self-triggered pinning consensus control, a new method of vehicle platooning at an intersection is proposed. To the best of our knowledge, vehicle platooning at an intersection using self-triggered pinning consensus control has not been studied. To avoid congestion in communication networks, it is appropriate to apply the external control input, which is calculated by e.g., cloud systems, to only some vehicles. Furthermore, it is also significant to adjust the communication intervals using self-triggered control.

Here, we use the problem setting in [21]. In [21], only a simple circular course has been considered. We suppose that there are n one-way traffic lanes and one intersection. In each lane, there is one vehicle group consisting of multiple vehicles. The inter-vehicular distance of each vehicle is modeled as a linear state equation. Then, virtual merge and split of vehicle groups are proposed. If the lead vehicle in some group reach a certain location on brink of an intersection, then merge starts, that is, n groups are regarded as one group. For one group, self-triggered pinning consensus control is performed. As a result, collision avoidance at an intersection is achieved. After all vehicles pass an intersection, this group is split to n groups. Thus, collision avoidance at an intersection with no vehicles stopping/backing is achieved. Furthermore, we analyze performance of vehicle platooning at an intersection. From the viewpoint of optimal control, we estimate the appropriate position where virtual merge should start for the case of two lanes.

The conference paper [29] is a preliminary version of this paper. In [29], only the outline of the proposed method has been explained. In this paper, details of the proposed method are explained. Furthermore, a new result on performance analysis (Section 4) and a numerical example are provided.

Notation: Let $\mathcal{R}$ denote the set of real numbers. Let I_n and $0_{m\times n}$ denote the $n\times n$ identity matrix and the $m\times n$ zero matrix, respectively. Let 1_n denote the n-dimensional column vector whose elements are all one. For the finite set A, let $|A|$ denote the number of elements in A. Let $M \succ 0$ ($M \succeq 0$) denote that the matrix M is positive-definite (positive-semidefinite).

2. Preliminaries

In this section, the outline of self-triggered pinning consensus control is explained. See, e.g., [19] for further details.

2.1 *Pinning Consensus Control*

First, pinning consensus control is summarized.

The dynamics of the agent $i \in \mathcal{V}, \mathcal{V} = \{1, 2, \dots, \bar{n}\}$ are defined as the following discrete-time integrator:

$$x_i(k+1) = x_i(k) + u_i(k), \tag{1}$$

where $x_i \in \mathcal{R}$ and $u_i \in \mathcal{R}$ are the state and the control input of the agent i, respectively. Communication links between agents are represented by an undirected connected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{V}\times\mathcal{V}$ is the set of edges. Let $A \in \{0,1\}^{\bar{n}\times\bar{n}}$ denote the adjacency matrix of G. We assume that there is no self-loop, that is, the (i,i)-th element of A is zero. Let $\mathcal{N}_i \subset \mathcal{V}$ denote the set of agents that are adjacent to the node i. The degree matrix D is defined by $D := \mathrm{diag}(|\mathcal{N}_1|, |\mathcal{N}_2|, \dots, |\mathcal{N}_{\bar{n}}|)$.

In pinning control, pinning agents are introduced. Pinning agents may be set as "leaders". Pinning control is a method of controlling only pinning agents by the external signal. By communications between agents, the state of each agent converges to a target value different from the average of the initial states of all agents. The set of pinning agents is defined by $\mathcal{V}_p = \{i_1, i_2, \ldots, i_m\} \subset \mathcal{V}$, $m \ll \bar{n}$. The external control input $v_i(k) \in \mathcal{R}$ is added to pinning agents as follows:

$$u_i(k) = -\varepsilon \sum_{j \in \mathcal{N}_i} (x_i(k) - x_j(k)) + v_i(k), \quad i \in \mathcal{V}_p, \tag{2}$$

$$u_i(k) = -\varepsilon \sum_{j \in \mathcal{N}_i} (x_i(k) - x_j(k)), \quad i \in \mathcal{V} \setminus \mathcal{V}_p, \tag{3}$$

where $\varepsilon \in (0, 1/\max_i |\mathcal{N}_i|)$ is a given parameter. In the case of $v_i(k) = 0$, $\lim_{k\to\infty} x_i(k) = \sum_{i=1}^{\bar{n}} x_i(0)/\bar{n}$ holds [3].

The system consisting of (1), (2), and (3) is represented as

$$x(k+1) = Px(k) + Bv(k), \tag{4}$$

where $P = I_{\bar{n}} - \varepsilon(D - A)$ and $v = [v_1 \ v_2 \ \cdots \ v_m]^\top$. In the matrix $B \in \{0,1\}^{\bar{n}\times m}$, the (i_j, j)-th element is given by 1, and other elements are given by 0.

2.2 *Self-Triggered Pinning Consensus Control*

Next, we consider combing pinning consensus control with self-triggered control. In the case of networked control systems, through a communication network, the external control input is sent from the centralized controller to each pinning agent. It is desirable that the number of times for sending of the external control input is low from the viewpoint of reduction of the amount of communication. Using self-triggered control, the number of times for sending can be reduced, because the next update time of the external control input is also optimized. Hence, we consider self-triggered pinning consensus control.

Here, we consider utilizing MPC [19, 30]. In the conventional MPC, the finite-time optimal control problem is solved at each time. In the finite-time optimal control problem of self-triggered control, both the control input and the next update time are calculated. For (4), we calculate $v(k)$, $k \in [t, t+T-1]$, where t is the current time, $t+T$ is the next update time (i.e., T is the sampling interval), and $v(t) = v(t+1) = \cdots = v(t+T-1)$. Both $v(k)$ and T are decision variables. It is not necessary to calculate all update times in the prediction horizon, because only the first external control input and the first update time are applied to the system.

To derive both $v(k)$ and T, we consider the following finite-time optimal control problem for the system (4), where $\tilde{x}(k) := x(k) - x_d 1_{\bar{n}}$ and $x_d \in \mathcal{R}$ is the target state given in advance.

Problem 1.

$$\begin{aligned}
\text{given} \quad & x(t) = x_t \\
\text{find} \quad & T \in \{1, 2, \ldots, \bar{T}\}, \\
& \bar{v} = [v^\top(t) \ v^\top(t+1) \ \cdots \ v^\top(t+N-1)]^\top \in \mathcal{R}^{mN} \\
\text{min} \quad & J = \frac{\alpha}{T} + J_s, \\
& J_s = \sum_{k=t}^{t+N-1} \left\{ \tilde{x}^\top(k) Q \tilde{x}(k) + v^\top(k) R v(k) \right\} + \tilde{x}^\top(t+N) Q_f \tilde{x}(t+N)
\end{aligned}$$

$$\begin{aligned}
\text{s.t.} \quad & \text{System (4)}, \\
& v(t) = v(t+1) = \cdots = v(t+T-1), && (5) \\
& v(t+T+r\bar{T}) = v(t+T+r\bar{T}+1) = \cdots = v(t+T+r\bar{T}+\bar{T}-1), \\
& \qquad\qquad\qquad\qquad r \in \{0,1,\ldots,f-1\}, && (6) \\
& v(t+T+f\bar{T}) = \cdots = v(t+N-1). && (7)
\end{aligned}$$

In this problem, $\alpha \geq 0$, $Q \succeq 0$, $R \succ 0$, and $Q_f \succeq 0$ are given weights, and N is a given prediction horizon. For a given N, the relation $N = T + f\bar{T} + s$, i.e., $N - T \equiv s \mod \bar{T}$ is satisfied ($\bar{T}$ is a given upper bound of T), where $f \geq 0$ and $s \geq 0$ can be determined by fixing T.

The control input is constrained by (5), (6), and (7). In the first time interval $[t, t+T-1]$, the control input is a constant, and both T and $v(t)$ in (5) are decision variables. The scalar T is chosen from the finite set $\{1, 2, \ldots, \bar{T}\}$. In the time interval $[t+T+r\bar{T}, t+T+r\bar{T}+\bar{T}-1]$, the control input is a constant. Also in the time interval $[t+T+f\bar{T}, t+N-1]$, the control input is a constant. Hence, in the time interval $[t+T, t+N-1]$, only $v(t+T+r\bar{T})$ in (6) and $v(t+T+f\bar{T})$ in (7) are decision variables. To consider the worst performance in the time interval $[t+T, t+N]$, the sampling interval in $[t+T, t+f\bar{T}]$ is given by $\bar{T}$.

We briefly summarize a solution method for Problem 1. First, the optimal value of J_s and the optimal control input sequence can be analytically derived under the assumption that x_t and T are given. Let $J_s^*(x_t, T)$ and $\bar{v}^*(x_t, T)$ denote the optimal value of J_s and the optimal control input sequence, respectively. For a given x_t, the optimal T for Problem 1 can be derived by

$$T^* = \underset{T \in \{1,2,\ldots,\bar{T}\}}{\arg\min} \left\{ \frac{\alpha}{T} + J_s^*(x_t, T) \right\}.$$

Using T^*, the optimal control input sequence for Problem 1 can be derived by $\bar{v}^*(x_t, T^*)$.

According to the receding horizon policy, the control input is persistently generated by the following procedure:

Procedure of self-triggered pinning consensus control:

Step 1: Problem 1 is solved at the current time t.

Step 2: Apply $[I_m \ 0_{m \times m(N-1)}]\bar{v}^*(x_t, T^*)$ to the system (4) in the time interval $[t, t+T^*-1]$.

Step 3: Update $t \leftarrow t + T^*$, and go to Step 1.

By this procedure, not only the external control input but also the next update time of the external control input are calculated. Thus, self-triggered pinning consensus control can be realized.

3. Virtual Merge and Split of Vehicle Groups at Intersection

Based on self-triggered pinning consensus control, we consider virtual merge and split of vehicle groups to avoid collisions of vehicles at an intersection. First, a mathematical model of vehicle platooning is derived based on [21]. In [21], only a simple circular course was considered. In this paper, we consider a more complicated case. Next, we propose virtual merge and split of vehicle groups.

3.1 *Mathematical Model of Vehicle Platooning*

Fig. 1 shows an intersection where n one-way traffic lanes intersect. Suppose that in each lane, there are p vehicles. The p vehicles over the lane i is called a group. That is, there are n vehicle groups. Suppose also that all vehicles do not turn left/right at an intersection. Hence, the number

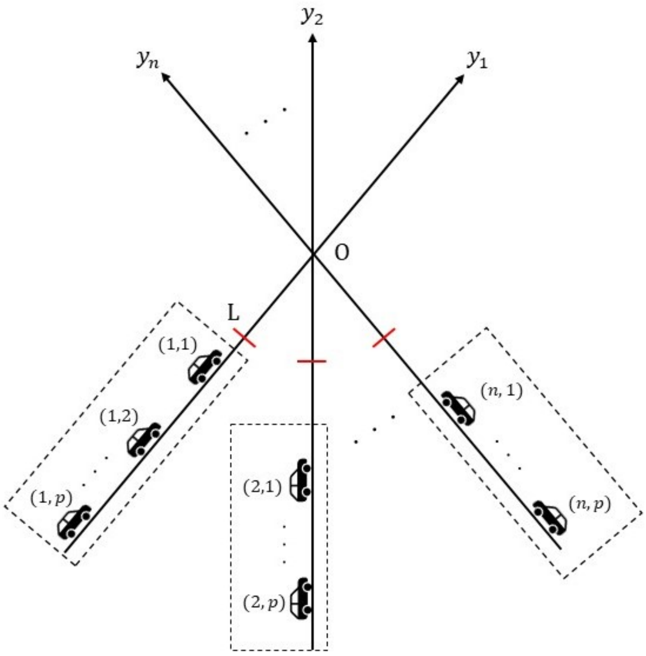

Figure 1. Intersection with n lanes before merge.

of vehicles of each group is not changed physically. For the lane $i \in \{1, 2, \dots, n\}$, the coordinate axis y_i is introduced. Set the position of the intersection O as $y_i = 0$.

The j-th vehicle in the lane i is labeled by (i, j). Let $l_{i,j}(k)$, $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2, \dots, p\}$ denote the position of the vehicle (i, j) along the coordinate axis y_i at time k, where $k = 0, 1, 2, \dots$ is the discrete time. The speed and the inter-vehicular distance corresponding to the state in the consensus problem are given by

$$s_{i,j}(k) = l_{i,j}(k+1) - l_{i,j}(k),\ j = 1, 2, \dots, p, \tag{8}$$

$$x_{i,j}(k) = l_{i,j}(k) - l_{i,j+1}(k),\ j = 1, 2, \dots, p-1. \tag{9}$$

Suppose that a part of $x_{i,j}(k)$ is regarded as a pinning agent. Let $\gamma(i) \in \{1, 2, \dots, p-1\}$ denote a pinning agent for the lane i. For simplicity of discussion, assume that the number of pinning agents in each vehicle group (i.e., each lane) is one. In addition, $s_{i,j}(k) - s_{i,j+1}(k)$ is regarded as a control input in (1). Hereafter, $s_{i,j}(k) - s_{i,j+1}(k)$ is denoted by $u_{i,j}(k)$.

Thus, the state equation of the inter-vehicular distance is given by

$$\begin{aligned} x_{i,\gamma(i)}(k+1) &= x_{i,\gamma(i)}(k) + u_{i,\gamma(i)}(k) + v_{i,\gamma(i)}(k), \\ x_{i,j}(k+1) &= x_{i,j}(k) + u_{i,j}(k), \quad j \in \{1, 2, \dots, p-1\} \backslash \{\gamma(i)\}, \end{aligned}$$

where $v_{i,\gamma(i)}(k)$ is the external control input applied to the pinning agent. These equations correspond to the system (4), where $\bar{n} = n(p-1)$ and $m = n$.

3.2 *Proposed Virtual Merge and Split*

Assume that the speed $s_{i,1}(k)$ of the lead vehicle in each group is constant and is given in advance. Using self-triggered pinning consensus control, it is expected that the inter-vehicular distance of each vehicle in the group approaches the target value x_d. Hence, it is also expected

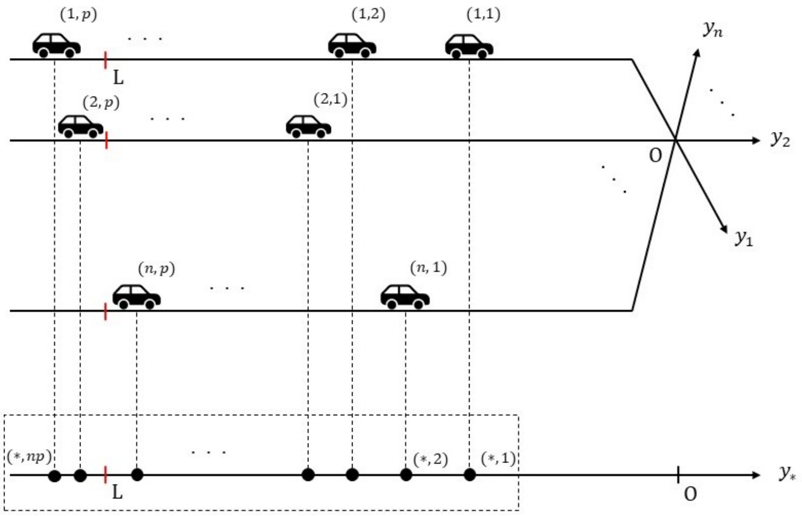

Figure 2. Illustration of merge in vehicle groups.

that consensus is practically achieved in each group. Noting that $s_{i,1}(k)$ is given, the speed of each vehicle is given by $s_{i,j}(k) = s_{i,j-1}(k) - u_{i,j-1}(k)$, $j = 2, 3, \ldots, p$. It is also expected that the speeds of all vehicles in the group approach a constant.

However, in vehicle platooning for each lane, the positions of vehicles in other lanes are not considered. Hence, there is a possibility that collisions between vehicles occur at an intersection. In this paper, we consider collision avoidance by performing both virtual merge and split.

The outline of virtual merge and split of vehicle groups is given as follows.

Merge: Before approaching an intersection, n vehicle groups are regarded as one group. For one group consisting of np vehicles, self-triggered pinning consensus control is performed.

Split: After all vehicles pass an intersection, one group is regarded as n groups, where each group consists of p vehicles. For each group, self-triggered pinning consensus control is performed.

Fig. 1 shows the situation before merge and Fig. 2 an illustration of merge. In Fig. 1 and Fig. 2, $L(< 0)$ is the position where merge starts. When vehicle groups merge, we introduce the coordinate axis y_* and all vehicles in all lanes are regarded as one vehicle group. Let $l_{*,j'}(k), j' = 1, 2, \ldots, np$ denote the position of the vehicle over the coordinate axis y_*. From (8) and (9), the speed and the inter-vehicular distance are respectively given by

$$s_{*,j'}(k) = l_{*,j'}(k+1) - l_{*,j'}(k), \quad j' = 2, 3, \ldots, np, \tag{10}$$

$$x_{*,j'}(k) = l_{*,j'}(k) - l_{*,j'+1}(k), \quad j' = 1, 2, \ldots, np-1, \tag{11}$$

where $s_{*,1}(k)$ (the speed of the lead vehicle) is a constant. In addition, for the vehicle group over the coordinate axis y_*, the inter-vehicular distance $x_{*,\gamma'(i)}(k), \gamma'(i) \in \{1, 2, \ldots, np-1\}$ corresponding to the pinning agent is chosen from the group. For simplicity of discussion, the number of pinning agents in the vehicle group is limited to one. The state equation of the inter-vehicular distance is given by

$$\begin{aligned} x_{*,\gamma'(i)}(k+1) &= x_{*,\gamma'(i)}(k) + u_{*,\gamma'(i)}(k) + v_{*,\gamma'(i)}(k), \\ x_{*,j'}(k+1) &= x_{*,j'}(k) + u_{*,j'}(k), \quad j' \in \{1, 2, \ldots, np-1\} \backslash \{\gamma'(i)\}. \end{aligned}$$

Since $s_{*,1}(k)$ is given as a constant, the speed of each vehicle is given by $s_{*,j'}(k) = s_{*,j'-1}(k) -$

$u_{*,j'-1}$, $j' = 2, 3, \ldots, np$.

The inter-vehicular distances over the coordinate axis y_* are defined based on the location of each vehicle from the intersection in different lanes. These distances approaches the target value by performing self-triggered pinning consensus control for virtual one group. The speeds also approaches a constant. As a result, collision avoidance at an intersection can be achieved.

4. Performance analysis: Estimation of L

In this section, performance analysis of vehicle platooning at the intersection using the proposed method "virtual merge and split" is conducted. In order to avoid a collision at the intersection, it is sufficient that the inter-vehicular distances are larger than a given threshold. It may not be necessary to utilize consensus control. However, when vehicle platooning is performed before and after the intersection, a smaller threshold is desirable. In such cases, consensus control is useful as one of the control methods. Moreover, the shorter time during a merge, the smoother driving around the intersection. That is, it is desirable that L is as close to the intersection as possible. Therefore, we propose a method for estimating L from the viewpoint of optimal control. By assuming the situation that takes the longest to reach consensus, effective L for collision avoidance in any vehicle placement is calculated.

Here, L is estimated for the case where the number of lanes n is two. As a preparation, consider the conditions that require virtual merging. The following assumption is made for the proposed method.

Assumption 1. *The number of lanes n is given by $n = 2$. At the start of merge, the following relations hold:*

$$s_{1,1}(k) = s_{1,2}(k) = \cdots = s_{1,p}(k) = s_{2,1}(k) = s_{2,2}(k) = \cdots = s_{2,p}(k),$$
$$x_{1,1}(k) = x_{1,2}(k) = \cdots = x_{1,p-1}(k) = x_{2,1}(k) = x_{2,2}(k) = \cdots = x_{2,p-1}(k) = x_d.$$

This assumption implies that at the start of merge, vehicle groups in each lane are traveling in a state of converging to the same speed and the same inter-vehicular distance. Then, we can derive the following lemma.

Lemma 1. *Let J_s^* denote the optimal value of the cost function J_s of the finite-time optimal control problem in Problem 1. Under Assumption 1, at the start of merge, the candidates of the current state $x(t) = x_t$ maximizing J_s^* are given by the following p vehicle placements (see also Fig. 3):*

$$\begin{aligned} &1) \quad l_{1,1}(t) = l_{2,1}(t), \\ &2) \quad l_{1,2}(t) = l_{2,1}(t), \\ &\qquad \vdots \\ &p) \quad l_{1,p}(t) = l_{2,1}(t). \end{aligned}$$

Proof. Without loss of generality, consider the case of $l_{1,1} \geq l_{2,1}$. In this case, when the vehicle (1,1) reaches L, and $l_{2,1}$ (the position of the lead vehicle of another group) is in the range $L - x_d(p-1) \leq l_{2,1} \leq L$, merging is needed to avoid a collision (see also Fig. 4). At the time of merge, J_s^* is a downward convex function with respect to the inter-vehicular distance of the virtual group $x_{*,j}, j = 1, 2, \ldots, 2p-1$, thus it takes the maximum value at the end point. When merge is required, the possible range of $x_{*,j}$ at the start of merging is $0 \leq x_{*,j} \leq x_d$. Therefore, there are 2^{2p-1} combinations of 0 and x_d as the end points of J_s^*. The end points that can be taken as the inter-vehicular distance of a virtual group are only p combinations (see also Fig. 3)

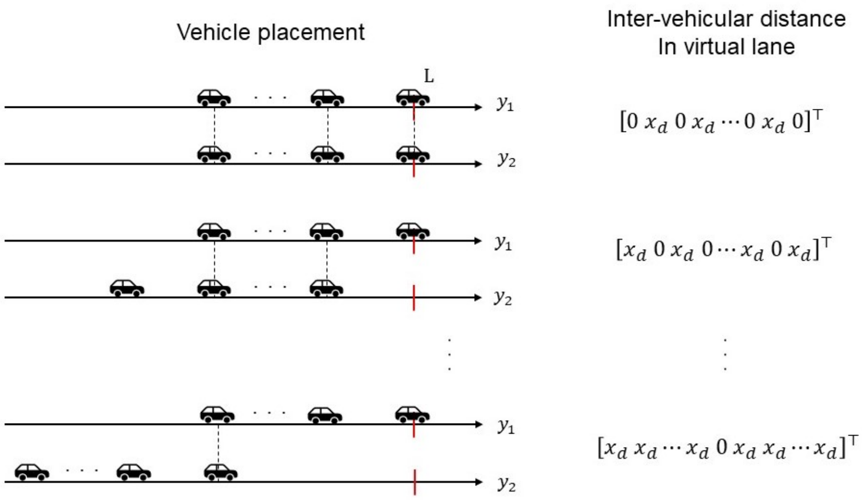

Figure 3. Vehicle placement at end points.

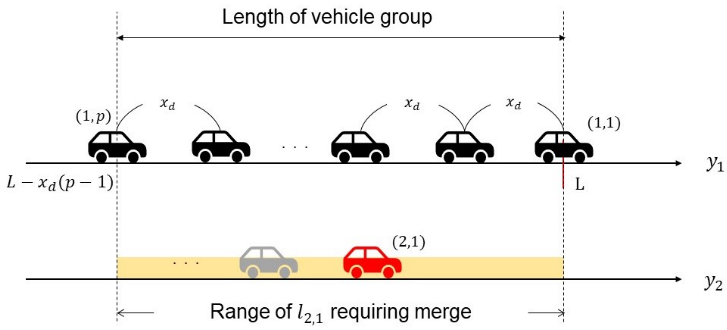

Figure 4. Example when merge is required.

This completes the proof. □

Based on this lemma, the procedure for estimating L is shown below.

Procedure for estimating L:

Step 1 (Find the worst vehicle placement):

Given each vehicle placement obtained in Lemma 1 as the initial state, solve the finite-time optimal control problem once at time $t = 0$. In self-triggered control, the optimal communication interval T is obtained along with the input by solving the problem. Since the convergence of the state becomes slower as T becomes larger, the calculation is performed by fixing T to its upper limit value $\bar{T}$. Set a sufficiently long prediction horizon N. For each vehicle placement, find the optimal value of the cost function J by solving Problem 1. Find a vehicle placement such that the optimal value is maximum.

Step 2 (Calculate L):

Calculate the optimal predicated state trajectory in Problem 1 when the worst vehicle placement obtained in Step 1 is used. From the obtained trajectory, find the time $t_{\max}$ that approaches the target inter-vehicular distance. Using $t_{\max}$, derive L, which is as close to the intersection as possible, as $L = -s_{*,1} \cdot t_{\max}$ ($s_{*,1}$ is the speed of the lead vehicle of the virtual group).

By applying L estimated in the above procedure for "virtual merge and split", it is considered possible to merge so as to avoid collision for any vehicle placement. In addition, with the proposed estimation method, efficient L can be calculated offline in advance.

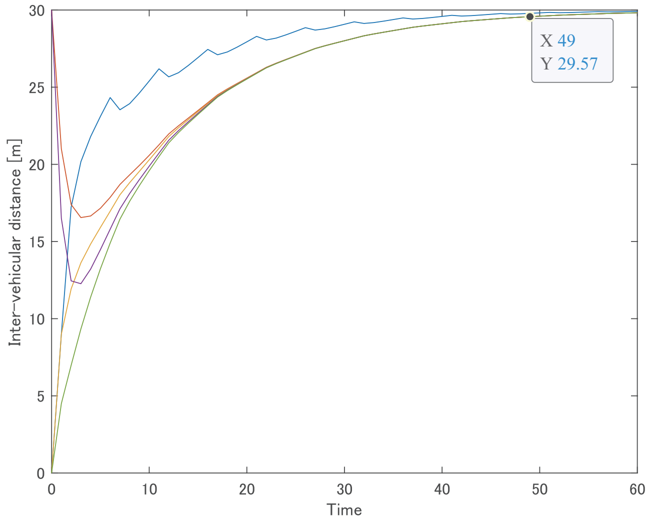

Figure 5. Time-series graph of the inter-vehicular distance.

5. Numerical example

We present a numerical example.

Consider the case of $n = 2$ and $p = 3$. In each lane, communications between agents are given by a complete graph. The pinning agent for the lane i is given by $x_{i,1}$ (in other words, the control signal from the external is sent to the vehicle $(i, 2)$). When virtual merge is performed, the pinning agent is given by $x_{*,1}$ (i.e., the number of pinning agents is given by one). The parameter ε is given by $\varepsilon = 0.15$. To represent communication delays, instead of (4), we consider the input-delay system $x(k+1) = Px(k) + Bv(k-1)$, where $v(-1)$ is given by $v(-1) = 0$. Also for this system, Problem 1 can be solved[1]. In Problem 1, Q, R, α, and $\bar{T}$ are given by $Q = 500I_n$, $R = 15000I_m$, $\alpha = 100$, and $\bar{T} = 5$, respectively. The target value x_d is given by 30m (when virtual merge is performed) and 25m (otherwise). The initial locations of vehicles are given by $l_{1,1}(0) = -1005$m, $l_{1,2}(0) = -1055$m, $l_{1,3}(0) = -1090$m, $l_{2,1}(0) = -1030$m, $l_{2,2}(0) = -1070$m, and $l_{2,3}(0) = -1100$m. Suppose that the speed of the lead vehicle is always 10m/s.

First, we consider estimating L. In Step 1 of the procedure for estimating L, the worst vehicle placement can be derived as the first placement in Fig. 3. In this case, the optimal value of the cost function is 1.20×10^7. In Step 2, if we suppose that a tolerable error is 1.5%, then $t_{\max}$ can be derived as $t_{\max} = 49$s (see Fig. 5). Thus, L can be estimated as $L = -s_{*,1} \cdot t_{\max} = -10 \cdot 49$m $= -490$m. In the numerical simulation below, considering margins, L is set as $L = -500$m.

Next, we present the computation result in the case where virtual merge and split are not performed. Fig. 6 and Fig. 7 show the inter-vehicular distance and the speed, respectively. From these figures, we see that consensus about the inter-vehicular distance and the speed is achieved. Fig. 8 shows the position of each vehicle. From this figure, we see that some vehicles collide at the intersection.

Finally, we present the computation result using the proposed method. Fig. 9 shows the inter-vehicular distance. From this figure, we see that before virtual merge is performed, the inter-vehicular distance converges to 25m. We also see that when virtual merge is performed, the inter-vehicular distance converges to 30m. Fig. 10 shows the speed of each vehicle. From this figure, we see that the speed converges to 10m/s (the speed of the lead vehicle). Fig. 11 shows the position of each vehicle. From this figure, we see that all vehicles pass the origin at different

[1] We suppose that collections of measurements in vehicles, calculations of Problem 1, and transmissions of external control inputs to vehicles are performed by utilizing ITS (Intelligent Transport Systems).

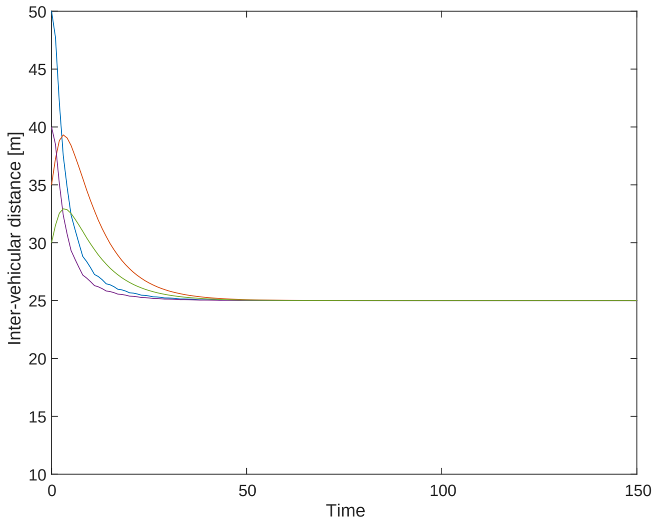

Figure 6. Inter-vehicular distance in the case where virtual merge and split are not performed.

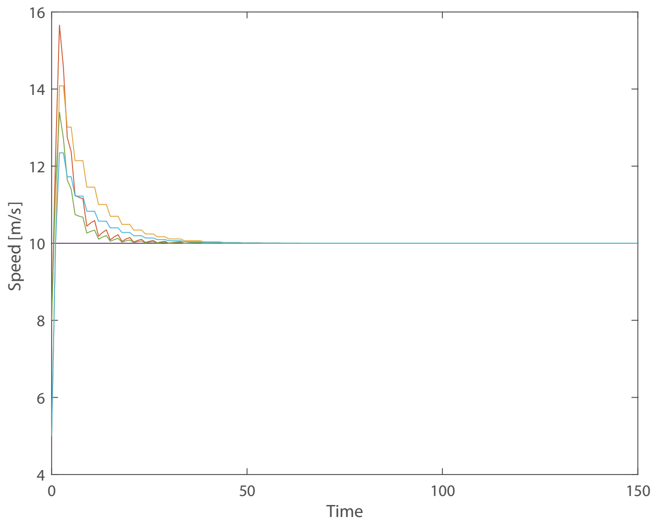

Figure 7. Speed in the case where virtual merge and split are not performed.

times. That is, collision avoidance at the intersection is achieved. Fig. 12 shows the external control input applied to pinning agents. We remark that when a virtual merge is performed, the number of pinning agents is one. Thus, using the proposed method, vehicle platooning with collision avoidance at the intersection can be achieved.

6. Conclusion

In this paper, we proposed a new method of vehicle platooning at an intersection using self-triggered pinning consensus control. To avoid a collision at an intersection, virtual merge and split of vehicle groups were proposed. In addition, we analyzed the performance of the proposed method for the specific case of two lanes. Finally, a numerical simulation was presented to demonstrate the proposed method.

In future work, it is important to improve the estimation method of L from the viewpoint of

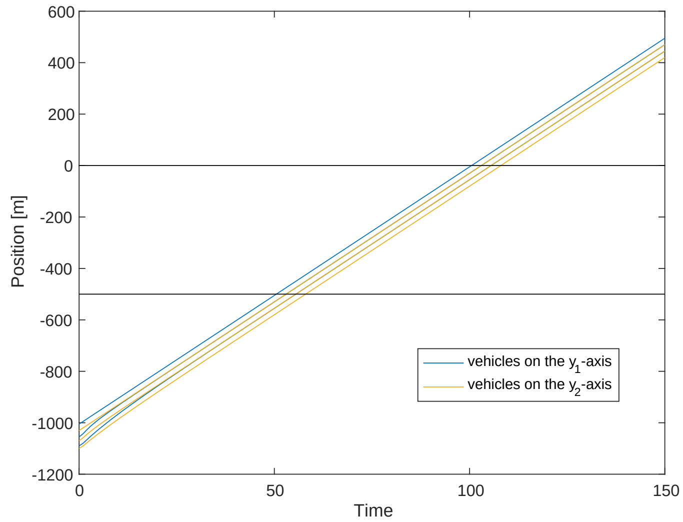

Figure 8. Position in the case where virtual merge and split are not performed.

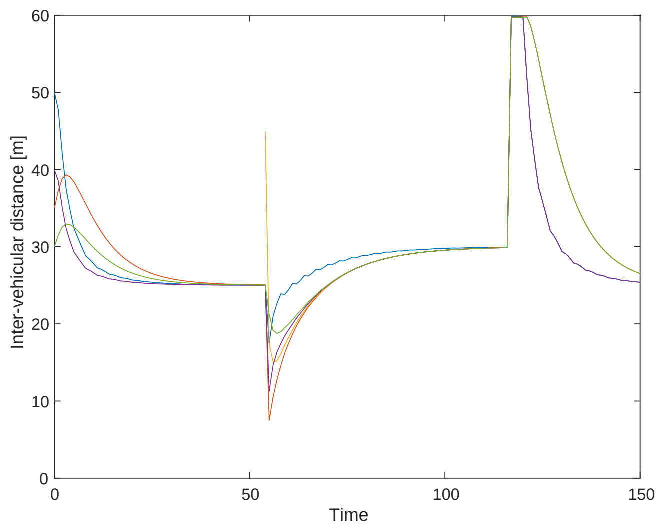

Figure 9. Inter-vehicular distance in the case where the proposed method is applied.

smooth intersection driving. The start position of virtual merge L is determined only based on the vehicle placement with the longest state convergence time. For placements that are easier to achieve consensus, we consider that a collision can be avoided in a shorter merging time. In other words, L may be set closer to the intersection. We consider that smoother merging at an intersection will be possible by making L variable according to the situation of vehicle groups. Furthermore, it is important to develop the estimation method of L for any number of lanes. It is also significant to develop a method of vehicle platooning in consideration of the existence of disturbances. Finally, in this paper, we considered one-way streets. Also in the cases of undivided streets and streets with double track, we consider that the problem setting in this paper is useful. It is future work to consider further details.

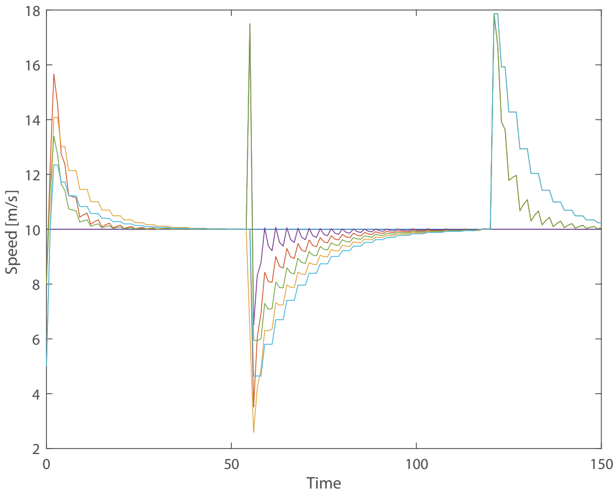

Figure 10. Speed in the case where the proposed method is applied.

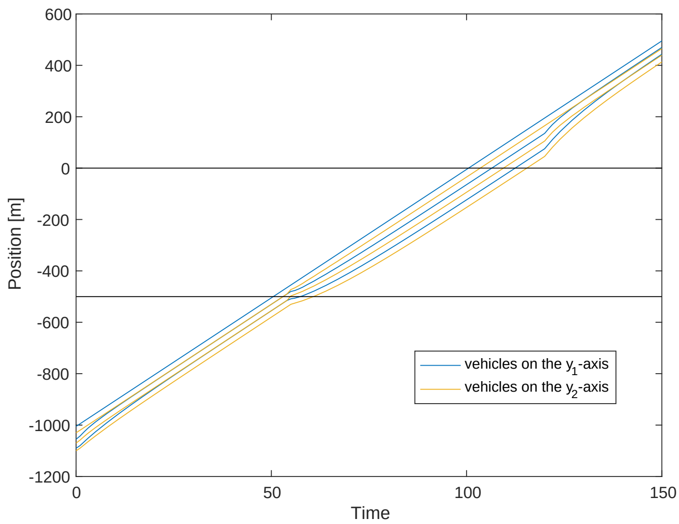

Figure 11. Position in the case where the proposed method is applied.

Acknowledgments

This work was partly supported by JSPS KAKENHI Grant Numbers JP19H02158, JP21H04558, JP22K04163.

References

[1] Ferrari-Trecate G, Galbusera L, Marciandi MPE, Scattolini R. Model predictive control schemes for consensus in multi-agent systems with single-and double-integrator dynamics. IEEE Transactions on Automatic Control. 2009;54(11):2560–2572.
[2] Hamada K, Hayashi N, Takai S. Event-triggered and self-triggered control for discrete-time average

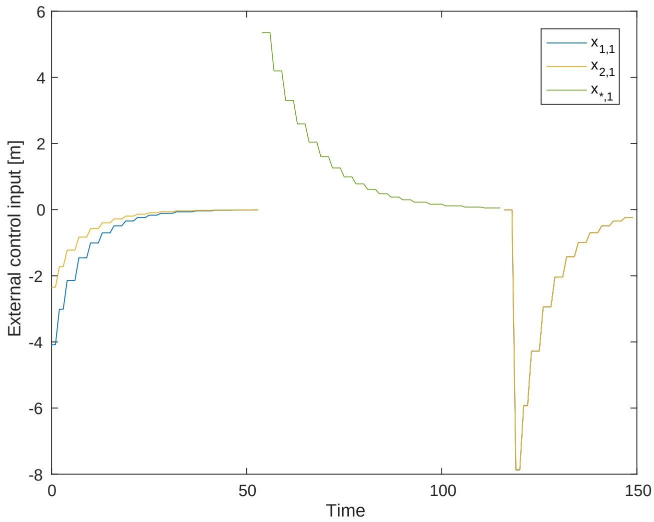

Figure 12. External control input in the case where the proposed method is applied.

consensus problems. SICE Journal of Control, Measurement, and System Integration. 2014;7(5):297–303.

[3] Olfati-Saber R, Murray RM. Consensus problems in networks of agents with switching topology and time-delays. IEEE Transactions on Automatic Control. 2004;49(9):1520–1533.

[4] Zhang HT, Chen MZ, Stan GB. Fast consensus via predictive pinning control. IEEE Transactions on Circuits and Systems I: Regular Papers. 2011;58(9):2247–2258.

[5] Kobayashi K. Predictive pinning control with communication delays for consensus of multi-agent systems. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. 2019;102(2):359–364.

[6] Wakasa T, Nagatani Y, Sawada K, Shin S. Switched pinning control for merging and splitting maneuvers of vehicle platoons. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. 2020;103(4):657–667.

[7] Su H, Wang X. Pinning control of complex networked systems: Synchronization, consensus and flocking of networked systems via pinning. Springer Science & Business Media. 2013.

[8] Wang X, Su H. Pinning control of complex networked systems: A decade after and beyond. Annual Reviews in Control. 2014;38(1):103–111.

[9] Camacho EF, Alba CB. Model predictive control. Springer science & business media. 2013.

[10] Mayne DQ, Rawlings JB, Rao CV, Scokaert PO. Constrained model predictive control: Stability and optimality. Automatica. 2000;36(6):789–814.

[11] Heemels WP, Johansson KH, Tabuada P. An introduction to event-triggered and self-triggered control. In: 51st IEEE Conference on Decision and Control. 2012. p. 3270–3285.

[12] Li XM, Zhou Q, Li P, Li H, Lu R. Event-triggered consensus control for multi-agent systems against false data-injection attacks. IEEE Transactions on Cybernetics. 2019;50(5):1856–1866.

[13] Liang J, Liu K, Ji Z, Wang X. Event-triggered consensus control for linear multi-agent systems. IEEE Access. 2019;7:144572–144579.

[14] Iwase S, Hayashi N, Takai S. A gradient-based approach for discrete-time average consensus with self-triggered control. SICE Journal of Control, Measurement, and System Integration. 2016;9(3):122–127.

[15] Su Y, Wang Q, Sun C. Self-triggered consensus control for linear multi-agent systems with input saturation. IEEE/CAA Journal of Automatica Sinica. 2019;7(1):150–157.

[16] Zhan J, Li X. Self-triggered consensus of multi-agent systems via model predictive control. IFAC-PapersOnLine. 2016;49(4):19–24.

[17] Adaldo A, Alderisio F, Liuzza D, Shi G, Dimarogonas DV, Di Bernardo M, Johansson KH. Event-triggered pinning control of switching networks. IEEE Transactions on Control of Network Systems. 2015;2(2):204–213.

[18] Wu ZG, Xu Y, Pan YJ, Shi P, Wang Q. Event-triggered pinning control for consensus of multiagent systems with quantized information. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2017;48(11):1929–1938.

[19] Andoh S, Kobayashi K, Yamashita Y. Self-triggered pinning consensus control for multi-agent systems. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. 2020;103(2):443–450.

[20] Alam A, Mårtensson J, Johansson KH. Look-ahead cruise control for heavy duty vehicle platooning. In: 16th International IEEE Conference on Intelligent Transportation Systems. 2013. p. 928–935.

[21] Andoh S, Kobayashi K, Yamashita Y. Hierarchical pinning control for consensus of multi-agent systems and its application to vehicle platooning. In: 2020 IEEE/SICE International Symposium on System Integration. 2020. p. 1064–1069.

[22] Chen J, Bai D, Liang H, Zhou Y. A third-order consensus approach for vehicle platoon with inter-vehicle communication. Journal of Advanced Transportation. 2018;2018(article ID 8963289).

[23] Di Bernardo M, Salvi A, Santini S. Distributed consensus strategy for platooning of vehicles in the presence of time-varying heterogeneous communication delays. IEEE Transactions on Intelligent Transportation Systems. 2014;16(1):102–112.

[24] Kianfar R, Falcone P, Fredriksson J. A control matching model predictive control approach to string stable vehicle platooning. Control Engineering Practice. 2015;45:163–173.

[25] Maruta I, Yoshikawa T, Azuma S, Sugie T. Traffic control at intersection with linear traffic lights. In: 43rd Annual Conference of the IEEE Industrial Electronics Society. 2017. p. 2923–2928.

[26] Shi S, Lazar M. On distributed model predictive control for vehicle platooning with a recursive feasibility guarantee. IFAC-PapersOnLine. 2017;50(1):7193–7198.

[27] Wu J, Wang Y, Wang L, Shen Z, Yin C. Consensus-based platoon forming for connected autonomous vehicles. IFAC-PapersOnLine. 2018;51(31):801–806.

[28] Zegers JC, Semsar-Kazerooni E, Ploeg J, van de Wouw N, Nijmeijer H. Consensus control for vehicular platooning with velocity constraints. IEEE Transactions on Control Systems Technology. 2017; 26(5):1592–1605.

[29] Tanaka A, Kobayashi K, Yamashita Y. Self-triggered consensus approach to vehicle platooning at intersection. In: 2022 IEEE International Conference on Consumer Electronics. IEEE. 2022. p. 1–2.

[30] Henriksson E, Quevedo DE, Peters EG, Sandberg H, Johansson KH. Multiple-loop self-triggered model predictive control for network scheduling and control. IEEE Transactions on Control Systems Technology. 2015;23(6):2167–2181.