Event-Triggered Switched Pinning Control for Merging or Splitting Vehicle Platoons

Takuma Wakasa*, Kenji Sawada*, Seiichi Shin*

**The University of Electro-Communications, Chofu, Tokyo 182-8585 Japan (Tel:042-443-5891; e-mail: knj.sawada@uec.ac.jp).*

Abstract: This paper proposes a switched pinning control algorithm with an event-triggering mechanism. The target vehicle platoons are expressed by the multi-agent systems in which pinning agents receive target velocities from external devices. We construct a model predictive control (MPC) algorithm that switches pinning agents via mixed-integer quadratic programming (MIQP). The frequency of the switching, i.e., the solving of the MIQP problems is determined according to the convergence rate of vehicles to the target velocities. This event-triggering mechanism can reduce the calculation cost of the MPC in the steady. Moreover, our algorithm regroups platoons to the arbitrary ones by controlling the adjacent vector that expresses the adjacency between vehicles. As a result, our algorithm allows the external devices to form arbitrary platoons and control the velocity of each platoon while considering the calculation cost.

Keywords: Multi-Agent Systems, Model Predictive Control Consensus Control, Pinning Control, Event-Triggering mechanism, Intelligent Transport Systems, Mixed Dynamical System Model.

1. INTRODUCTION

This paper considers a velocity control problem for autonomous vehicle platoons by Intelligent Transport Systems (ITS). In the development of connected vehicles, platoon control methods have been studied [Serban, S. (2016)]. In the platoon control, each vehicle adjusts its velocity based on the information by sensing or vehicle to vehicle (V2V) communications. On the other hand, when the length of the platoon increases, the disturbances applied to a preceding vehicle propagate to the following vehicles [Richard, H.M. (2010)]. As a solution to this problem, vehicle control methods communicating with ITS via V2X communications have been studied [Yongfu, L. (2019)], [Yang, S. (2018)]. Using a wider range of information, we can control vehicles effectively.

The recent vehicle control has been studied in terms of multi-agent-systems (MASs), where each agent determines its behavior autonomously communicating with others, and then the group of agents achieves formation control. Typical platoon control is leader-follower control, which is a piece of the consensus control [Shun-ichi, A. (2015)], [Jiahu, Q. (2017)]. Sharing velocity information by V2V communication, vehicles follow the velocity of the leader. A method that combines V2X communication with consensus control becomes the pinning control method [Xu, D. (2018)], [Akinori, S. (2017)]. In the pinning control, an external device (ex. Intelligent traffic signal) applies the velocity command to certain vehicles (pinning agents) and lets the velocity of the platoon converge to the target values. Selecting the pinning agent is important because the consensus speed of the whole of the platoon does not become faster even if the pinning agent follows the target value quickly. For example, an optimal node selection method is proposed [Weng, Y. (2016)] for the invariant graph structure. However, we have to consider the variant graph structure because the vehicle platoons merge or split. Also, it is important to consider the pinning control of multiple independent MASs that do not communicate with each other before the merge of the platoons.

Motivated by the above, the previous work of the current authors studies the switched pinning control (SPC) method [Takuma, W. (2020)]. The SPC method selects and switches the pinning agents from the given MASs. The switching of the pinning agents is expressed by the Mixed Logical Dynamical System Model (MLD system model) [Jun-ichi, I. (2014)], [Alberto, B. (1999)]. The controller selects the pinning agents that minimize the consensus speed in finite time. This process is expressed by the MIQP (Mixed Integer Quadratic Programing). Solving this MIQP problem every step according to Model Predictive Control (MPC) strategy [Di, C. (2019)], the controller can let the platoon consensus to the target value faster. Moreover, this method is applicable for the situation that some platoons merge, that is, for the situation that the graph structure of MASs is variant.

On the other hand, this method solves the optimization problem every step, and the computational load becomes high as the number of vehicles increases. In this paper, we propose an event-triggered switched pinning control (event-triggered SPC) algorithm for the MASs. This proposed algorithm consists of (i) the optimization method of the pinning agents (ii) the representation method of the dynamical graph for the model predictive control (iii) the event-triggering mechanism to determine the switching frequency of the pinning agents. Switching frequency is equal to the solving frequency of the MIQP problem. As a result, the proposed method reduces the calculation time by adjusting the solving frequency according to the convergence rate to the target value. Also, our method expresses the merging and splinting of the platoons as the grouping and enables the external devices to regroup the platoons into the arbitrary platoons.

2. PRELIMINARIES

2.1 Dynamics of Each Agent

Suppose that the i-th agent is a spring-mass-damper system:

$$\frac{d}{dt}\begin{bmatrix} x^i(t) \\ v^i(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_i & -c_i \end{bmatrix}\begin{bmatrix} x^i(t) \\ v^i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u^i(t). \quad (1)$$

When n agents are assembled, the resultant state-space equation is given by

$$\dot{\boldsymbol{\xi}}(t) = \boldsymbol{A}_c\boldsymbol{\xi}(t) + \boldsymbol{B}_c\boldsymbol{u}(t) \quad (2)$$

where

$$\boldsymbol{A}_c = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ -\boldsymbol{K} & -\boldsymbol{C} \end{bmatrix},\ \boldsymbol{B}_c = \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{I} \end{bmatrix},\ \boldsymbol{\xi}(t) = [\boldsymbol{x}^{\mathrm{T}}(t)\ \boldsymbol{v}^{\mathrm{T}}(t)]^{\mathrm{T}},$$
$$\boldsymbol{x}(t) = [x^1(t)\ \cdots\ x^n(t)]^{\mathrm{T}}, \qquad \boldsymbol{v}(t) = [v^1(t)\ \cdots\ v^n(t)]^{\mathrm{T}},$$
$$\boldsymbol{K} = \mathrm{diag}\{k_1, \cdots, k_n\}, \qquad \boldsymbol{C} = \mathrm{diag}\{c_1, \cdots, c_n\}.$$

The discretized system of (2) with zero-order-holder and ideal sampler is given by

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{A}_d\boldsymbol{\xi}_k + \boldsymbol{B}_d\boldsymbol{u}_k, \quad (3)$$
$$\boldsymbol{A}_d = e^{\boldsymbol{A}_c T_s}, \qquad \boldsymbol{B}_d = \int_0^{T_s} e^{\boldsymbol{A}_c\tau} d\tau\, \boldsymbol{B}_c, \qquad \boldsymbol{\xi}_k = \boldsymbol{\xi}(kT_s)$$

where $T_s > 0$ is a sampling time.

2.2 Graph Theory

Consider an agent set $\mathcal{A} = \{a_1, \cdots, a_n\}$, in the graph theory, the situation that agent a_i can get information of agent a_j is translated into the following expression: agent a_i is adjacent to agent a_j. This agent relation is expressed by a graph (Fig. 1). The graph consists of nodes and edges. Each node denotes each agent, and each edge denotes each transmission path of information.

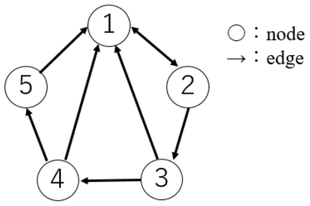


Fig. 1: an example of a graph.

The number of edges that enters node i is called in-degree and expressed by D_i. When the number of nodes is n and the in-degrees $D_1, \cdots, D_n$ are given, the in-degree matrix is given by

$$\boldsymbol{D} \stackrel{\mathrm{def}}{=} \mathrm{diag}\{D_1, \cdots, D_n\}. \quad (4)$$

The adjacency between the nodes is expressed by the adjacent matrix

$$\boldsymbol{A} = [A_{ij}] \in \mathbb{R}^{n\times n}, \quad (5)$$
$$A_{ij} = \begin{cases} 1 & \text{node } i \text{ is adjacent to node } j \\ 0 & \text{otherwise} \end{cases}.$$

The Graph Laplacian $\boldsymbol{L}$ is defined using $\boldsymbol{D}$ and $\boldsymbol{A}$ as follows:

$$\boldsymbol{L} \stackrel{\mathrm{def}}{=} \boldsymbol{D} - \boldsymbol{A}. \quad (6)$$

2.3 Grouping of Platoon Sets

We discuss the merging or splitting of the platoons in terms of grouping. We set platoon set $\mathcal{V}_k^i \subseteq \mathcal{A}\ (i = 1, \cdots, n)$ whose leader is vehicle a_i at step k. $\mathcal{A}$ and $\mathcal{V}_k^1, \cdots, \mathcal{V}_k^n$ satisfy the following equations every step:

$$\mathcal{A} = \mathcal{V}_k^1 \cup \cdots \cup \mathcal{V}_k^n,$$
$$\mathcal{V}_k^i \cap \mathcal{V}_k^j = \emptyset, \qquad i, j \in \{1, \cdots, n\}\ (i \neq j) \quad (7)$$

where $\emptyset$ is the empty set. Our grouping rule of $\mathcal{V}_k^i$ is according to an adjacent vector $\boldsymbol{d}_k = [d_k^1\ \cdots\ d_k^n]^{\mathrm{T}}$. d_k^i expresses whether vehicle a_i is the leader or follower as follows

$$d_k^i = \begin{cases} 0 & \text{if vehicle } a_i \text{ is the leader} \\ 1 & \text{otherwise} \end{cases}. \quad (8)$$

Using the adjacent vector, in this paper, each vehicle belongs to any one of the platoon sets $\mathcal{V}_k^1, \cdots, \mathcal{V}_k^n$ according to the following grouping rule:

$$a_i \in \mathcal{V}_k^l, \qquad l = \max\{j \mid d_k^j = 0,\ 1 \leq j \leq i\}, \quad (9)$$

where $1 \leq j \leq i$ is the index range of the leading candidates of vehicle a_i. For the straight course, the indexes of vehicles traveling in front of vehicle a_i are $i, i-1 \cdots, 2, 1$ (include i).

Here, we show an example of the grouping of 5 vehicles in the case of $\boldsymbol{d}_k = [0\ 1\ 0\ 1\ 1]^{\mathrm{T}}$.

Example) We focus on vehicle a_3. The rule (9) for vehicle a_3 becomes

$$a_3 \in \mathcal{V}_k^l, \qquad l = \max\{j \mid d_k^j = 0,\ \ 1 \leq j \leq 3\}.$$

The index range of the leading candidates is $1 \leq j \leq 3$. In this range, $j = \{1, 3\}$ satisfies $d_k^j = 0$. Then, l becomes 3 and vehicle a_3 is belongs to $\mathcal{V}_k^3$.

2.4 Consensus Control and Pinning Control

We divide the control input u_k^i into a distributed control input $u_k^{con,i}$ and an external control input $u_k^{pin,i}$ as follows:

$$u_k^i = u_k^{con,i} + u_k^{pin,i}. \quad (10)$$

We give the distributed control input for vehicle a_i as follows:

$$u_k^{con,i} = g_{11}x_k^i + g_{12}v_k^i$$
$$+ g_{21}(x_k^i - x_k^j) + g_{22}(v_k^i - v_k^j) \quad (11)$$

where g_{11}, g_{12}, g_{21}, g_{22} are gains. Assembling (11) for $i = 1, \cdots, n$, we get the distributed control input for platoons [Feng, X. (2007)]

$$\boldsymbol{u}_k^{con} = \boldsymbol{A}_{con}\boldsymbol{\xi}_k, \quad (12)$$
$$\boldsymbol{A}_{con} = [\boldsymbol{G}_{11} - \boldsymbol{G}_{21}\boldsymbol{L} \quad \boldsymbol{G}_{12} - \boldsymbol{G}_{22}\boldsymbol{L}].$$

The external input is applied to only some vehicles called pinning agents. We express an index set of pinning agents and the number of pinning agents by $\mathcal{P}$ and $n_p = |\mathcal{P}|$, respectively. The external control input is given by

$$u_k^{pin,i} = \begin{cases} g_p(v_r^i - v_k^i) & i \in \mathcal{P} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where g_p is a gain and v_r^i is a target velocity of vehicle a_i. Assembling (13) for $i = 1, \cdots, n$, the following equation holds:

$$\boldsymbol{u}_k^{pin} = \boldsymbol{G}_p^i(\boldsymbol{\xi}_r - \boldsymbol{\xi}_k) \quad (14)$$

where

$$\boldsymbol{G}_p^i = [\boldsymbol{O} \quad g_p\boldsymbol{A}_p^i], \qquad a_p^i = \begin{cases} 1 & i \in \mathcal{P} \\ 0 & \text{otherwise} \end{cases},$$
$$\boldsymbol{A}_p^i = \mathrm{diag}\{a_p^1, \cdots, a_p^n\}. \quad (15)$$

The target value vector is given by

$$\boldsymbol{\xi}_r = [\boldsymbol{x}_r^{\mathrm{T}} \ \boldsymbol{v}_r^{\mathrm{T}}]^{\mathrm{T}}, \quad (16)$$
$$\boldsymbol{x}_r = [x_r^1 \cdots x_r^n]^{\mathrm{T}}, \qquad \boldsymbol{v}_r = [v_r^1 \cdots v_r^n]^{\mathrm{T}}.$$

Assigning (12) and (14) to (3), we get a state-space equation of the pinning control

$$\boldsymbol{\xi}_{k+1} = \overline{\boldsymbol{A}}\boldsymbol{\xi}_k + \overline{\boldsymbol{B}}\boldsymbol{G}_p^i(\boldsymbol{\xi}_r - \boldsymbol{\xi}_k) \quad (17)$$

where $\overline{\boldsymbol{A}} = \boldsymbol{A}_d + \boldsymbol{B}_d\boldsymbol{A}_{con}$ and $\overline{\boldsymbol{B}} = \boldsymbol{B}_d$.

3. EVENT-TRIGGERED SWITCHED PINNING CONTROL

For the optimal velocity consensus of the vehicle platoons, it is important to select the pinning agents optimally. At the same time, we have to consider variant graph structure and multiple independent platoons which do not communicate with each other because the vehicle platoons merge or split. This situation is corresponding to the case where matrix $\overline{\boldsymbol{A}}$ and matrix $\boldsymbol{G}_p^i$ in (17) become time-variant matrices $\overline{\boldsymbol{A}}_k$ and $\boldsymbol{G}_{p,k}^i$. For this problem, the switched pinning control (SPC) method is proposed [Takuma, W. (2020)]. The SPC switches the pinning agents by solving the MIQP problem according to MPC strategy. The SPC updates network information, i.e., $\overline{\boldsymbol{A}}_k$ and switches the pinning agent, i.e., $\boldsymbol{G}_{p,k}^i$ every step. However, every step computation is far from a low computational task. Also, the SPC does not allow the external device to regroup the platoon structure into the arbitrary one.

Motivated by the above, we propose an event-triggered switched pinning control (event-triggered SPC) algorithm. Its calculation cost is switching frequency M, which equals to the solving frequency of MIQP. Observing the states at step k, the event-triggering mechanism selects the next step $\boldsymbol{G}_{p,k+1}^i$ and switches the pinning agent to vehicle a_i once per $1/M$ steps. $\boldsymbol{G}_{p,k+1}^i$ is continuously used until the next optimization, i.e., the following equation holds for $h = 0, 1, 2, \cdots$:

$$\boldsymbol{G}^i_{p,k+\frac{1}{M}h+1} = \boldsymbol{G}^i_{p,k+\frac{1}{M}h+2} = \cdots = \boldsymbol{G}^i_{p,k+\frac{1}{M}h+\frac{1}{M}}. \quad (18)$$

When we predict the N times switching, i.e., $h = 0, 1, \cdots, N-1$, the prediction horizon becomes N/M. Also, we consider the merging and splitting of the platoons as the grouping under the rule (9). Moreover, we consider controlling the adjacent vector $\boldsymbol{d}_k$ to regroup the platoons into the arbitrary platoons.

From the above, we propose an MPC algorithm that (i) updates $\boldsymbol{L}_k$, $\overline{\boldsymbol{A}}_k$, and M according to the trigger-condition, (ii) regroups n vehicles according to the rule (9), and (iii) solves following Problem 1. Our proposed algorithm regroups the vehicles into the arbitrary platoons and controls the velocity of each platoon with the low computational load.

Problem 1: Suppose that observed states $\boldsymbol{\xi}_k \in \mathbb{R}^n$, matrix $\boldsymbol{G}_{p,k}^i \in \mathbb{R}^{n\times 2n}$, $\overline{\boldsymbol{A}}_k \in \mathbb{R}^{2n\times 2n}$, the number of pinning agents $n_p \in \mathbb{N}$, the predictive horizon $N \in \mathbb{N}$, the switching frequency $M \in \mathbb{N}$, and the adjacent vector $\boldsymbol{d}_k \in \mathbb{R}^n$ are given. Find the series $\widehat{\boldsymbol{G}}_{pN,k} = [\boldsymbol{G}_{p,k+1}^i \cdots \boldsymbol{G}_{p,k+N}^i] \in \mathbb{R}^{2n\times 2nN}$ $(i = 1, \cdots, n)$ that minimize

$$J(\widehat{\boldsymbol{G}}_{pN,k}) = \sum_{j=1}^{N/M} \left(\boldsymbol{\xi}_r - \widehat{\boldsymbol{\xi}}_{k+j|k}\right)^{\mathrm{T}} \left(\boldsymbol{\xi}_r - \widehat{\boldsymbol{\xi}}_{k+j|k}\right) \quad (19)$$
$$\text{s.t.} \ \widehat{\boldsymbol{\xi}}_{k+j+1|k} = \overline{\boldsymbol{A}}_k\widehat{\boldsymbol{\xi}}_{k+j|k} + \overline{\boldsymbol{B}}\widehat{\boldsymbol{G}}_{p,k+j|k}^i\left(\boldsymbol{\xi}_r - \widehat{\boldsymbol{\xi}}_{k+j1|k}\right),$$
$$\widehat{\boldsymbol{G}}^i_{p,k+\frac{1}{M}h+1|k} = \widehat{\boldsymbol{G}}^i_{p,k+\frac{1}{M}h+2|k} = \cdots = \widehat{\boldsymbol{G}}^i_{p,k+\frac{1}{M}h+\frac{1}{M}|k}$$
$$(h = 0, 1, \cdots, N-1)$$

where $\widehat{\boldsymbol{\xi}}_{k+j|k}$ and $\widehat{\boldsymbol{G}}_{p,k+j|k}^i$ are the j-th prediction states and the matrix $\boldsymbol{G}_{p,k}^i$ at step k.

Our proposed method consists of the following three sub-methods.

3.1 Optimal Switching Method of Pinning Agents

When $n_p = 1$, the state-space equation (17) has n modes according to the index of the pinning agent. The state-space equation in mode i is expressed by (17). We assemble (17) of n modes into one equation

$$\boldsymbol{\xi}_{k+1} = \overline{\boldsymbol{A}}\boldsymbol{\xi}_k + \sum_{i=1}^{n} \delta_k^i\{\overline{\boldsymbol{B}}\boldsymbol{G}_p^i(\boldsymbol{\xi}_r - \boldsymbol{\xi}_k)\}. \quad (20)$$

δ_k^i is the i-th element of the mode vector at step k given by

$$\delta_k^i = \begin{cases} 1 & i \in \mathcal{P} \\ 0 & \text{otherwise} \end{cases}, \quad (21)$$
$$\boldsymbol{\delta}_k = [\delta_k^1 \cdots \delta_k^n]^{\mathrm{T}}. \quad (22)$$

We get the state-space equation in mode i. Since the state-space equation (20) is non-linear, we covert (20) into the MLD system model

$$\begin{cases} \boldsymbol{\xi}_{k+1} = \widehat{\boldsymbol{A}}\boldsymbol{\xi}_k + \widehat{\boldsymbol{B}}\boldsymbol{z}_k \\ \widehat{\boldsymbol{C}}\boldsymbol{\xi}_k + \widehat{\boldsymbol{D}}\boldsymbol{z}_k + \widehat{\boldsymbol{E}}\boldsymbol{\delta}_k \le \widehat{\boldsymbol{F}} \end{cases} \quad (23)$$

where

$$\boldsymbol{z}_k^i = \delta_k^i\boldsymbol{B}_c\boldsymbol{G}_p^i(\boldsymbol{\xi}_r - \boldsymbol{\xi}_k), \ \widehat{\boldsymbol{\xi}}_r = [\boldsymbol{\xi}_r \cdots \boldsymbol{\xi}_r]^{\mathrm{T}}, \quad (24)$$
$$\boldsymbol{z}_k = [\boldsymbol{z}_k^1 \cdots \boldsymbol{z}_k^n]^{\mathrm{T}}, \qquad \widehat{\boldsymbol{A}} = \overline{\boldsymbol{A}}, \qquad \widehat{\boldsymbol{B}} = [\boldsymbol{I} \cdots \boldsymbol{I}]^{\mathrm{T}},$$
$$\widehat{\boldsymbol{C}} = \left[\boldsymbol{0} \ \boldsymbol{0} \ -\overline{\boldsymbol{G}}_p \ \overline{\boldsymbol{G}}_p\right]^{\mathrm{T}}, \ \widehat{\boldsymbol{D}} = [-\boldsymbol{I} \ \boldsymbol{I} \ -\boldsymbol{I} \ \boldsymbol{I}]^{\mathrm{T}},$$
$$\widehat{\boldsymbol{E}} = \left[\widehat{\boldsymbol{f}}_{\mathrm{inf}} \ -\widehat{\boldsymbol{f}}_{\mathrm{sup}} \ \widehat{\boldsymbol{f}}_{\mathrm{sup}} \ -\widehat{\boldsymbol{f}}_{\mathrm{inf}}\right]^{\mathrm{T}},$$
$$\widehat{\boldsymbol{F}} = \left[\boldsymbol{0} \ \boldsymbol{0} \ \boldsymbol{f}_{\mathrm{sup}} - \widehat{\boldsymbol{G}}_p\widehat{\boldsymbol{\xi}}_r \ -\boldsymbol{f}_{\mathrm{inf}} + \widehat{\boldsymbol{G}}_p\widehat{\boldsymbol{\xi}}_r\right]^{\mathrm{T}},$$
$$\overline{\boldsymbol{G}}_p = \boldsymbol{B}_c\left[\boldsymbol{G}_p^1 \cdots \boldsymbol{G}_p^n\right]^{\mathrm{T}}, \qquad \widehat{\boldsymbol{G}}_p = \boldsymbol{B}_c\mathrm{diag}\{\boldsymbol{G}_p^1, \cdots, \boldsymbol{G}_p^n\},$$
$$\boldsymbol{f}_{\mathrm{inf}} = [\boldsymbol{f}_{\mathrm{inf}}^1 \cdots \boldsymbol{f}_{\mathrm{inf}}^n]^{\mathrm{T}}, \ \boldsymbol{f}_{\mathrm{sup}} = \left[\boldsymbol{f}_{\mathrm{sup}}^1 \cdots \boldsymbol{f}_{\mathrm{sup}}^n\right]^{\mathrm{T}},$$
$$\widehat{\boldsymbol{f}}_{\mathrm{inf}} = \mathrm{diag}\{\boldsymbol{f}_{\mathrm{inf}}^1, \cdots, \boldsymbol{f}_{\mathrm{inf}}^n\}, \ \widehat{\boldsymbol{f}}_{\mathrm{sup}} = \mathrm{diag}\{\boldsymbol{f}_{\mathrm{sup}}^1, \cdots, \boldsymbol{f}_{\mathrm{sup}}^n\}.$$

Constant vectors $\boldsymbol{f}_{\mathrm{inf}}^i = f_{\mathrm{inf}}^i[1 \cdots 1]^{\mathrm{T}}$, $\boldsymbol{f}_{\mathrm{sup}}^i = f_{\mathrm{sup}}^i[1 \cdots 1]^{\mathrm{T}}$ are supremum and infimum of $f^i(v_k^i) = g_pa_p^i(v_r^i - v_k^i)$.

From the above, to select the pinning agents is equal to design the mode vector $\boldsymbol{\delta}_k$ in (23).

Using the MLD system model (23), we find $\widehat{\boldsymbol{\delta}}_{k+i|k}$ $(i = 1, \cdots, N)$ that minimize the following cost function:

$$J(\widehat{\boldsymbol{\delta}}_{N,k}) = \sum_{i=1}^{N} \left(\boldsymbol{\xi}_r - \widehat{\boldsymbol{\xi}}_{k+i|k}\right)^{\mathrm{T}} \left(\boldsymbol{\xi}_r - \boldsymbol{\xi}_{k+i|k}\right) \quad (25)$$

and switches the pinning agents at step k as follows:

$$\boldsymbol{\delta}_{k+1} = \hat{\boldsymbol{\delta}}_{k+1|k}. \tag{26}$$

3.2 Representation Algorithm of Dynamical Graph for Application to MPC Algorithm

In this section, we propose a representation algorithm of the dynamical platoon structure for application to the MPC algorithm. Each vehicle decides which it is a leader or follower according to the inter-vehicular distance or commands from the external devices. This paper represents this process by determining the adjacent vector $\boldsymbol{d}_k$. Then we need to consider the dynamical graph (we call it $\mathcal{G}_k$) and update the Graph Laplacian in the MLD system model (23).

In this paper, we consider two types of adjacent vector $\boldsymbol{d}_k$ in Section 2.3. The first one is an internal-adjacent vector $\boldsymbol{d}_k^{in}$ that each vehicle sets according to the inter-vehicular distance. The internal-adjacent vector $\boldsymbol{d}_k^{in}$ is given as follows:

$$\boldsymbol{d}_k^{in} = \left[d_k^{in,1} \cdots d_k^{in,n}\right]^{\mathrm{T}}, \tag{27}$$

$$d_k^{in,i} = \begin{cases} 0 & \text{if } x_k^{i-1} - x_k^i > x_d \\ 1 & \text{otherwise} \end{cases}.$$

The second one is an external-adjacent vector $\boldsymbol{d}_k^{ex}$ that expresses the formation commands from the external devices. When the external devices command arbitrary platoon structures, they design the external-adjacent vector $\boldsymbol{d}_k^{ex}$ as follows:

$$\boldsymbol{d}_k^{ex} = \left[d_k^{ex,1} \cdots d_k^{ex,n}\right]^{\mathrm{T}}, \tag{28}$$

$$d_k^{ex,i} = \begin{cases} 0 & \text{if make vehicle } a_i \text{ a leader} \\ 1 & \text{if make vehicle } a_i \text{ a follower} \\ -1 & \text{otherwise} \end{cases}.$$

When changing the vehicle a_i to the leader or follower, the external device designs the i-th element of $\boldsymbol{d}_k^{ex}$ such as $d_k^{ex,i} = 0$ or $d_k^{ex,i} = 1$, respectively.

Using the above adjacency vector, we give the time-variant Graph Laplacian

$$\boldsymbol{L}_k = \begin{bmatrix} d_k^1 & 0 & \cdots & 0 & -d_k^1 \\ -d_k^2 & d_k^2 & 0 & \cdots & 0 \\ 0 & -d_k^3 & d_k^3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -d_k^n & d_k^n \end{bmatrix}, \tag{29}$$

$$d_k^i = \begin{cases} d_k^{in,i} & \text{if } d_k^{ex,i} = -1 \\ d_k^{ex,i} & \text{otherwise} \end{cases}.$$

Also, using (29), we make $\boldsymbol{A}_{con}$ and $\bar{\boldsymbol{A}}$ time-variant as follows:

$$\boldsymbol{A}_{con,k} = [\boldsymbol{K}_{11} - \boldsymbol{K}_{21} \quad \boldsymbol{K}_{12} - \boldsymbol{K}_{22}\boldsymbol{L}_k], \tag{30}$$

$$\bar{\boldsymbol{A}}_k = \boldsymbol{A}_d + \boldsymbol{B}_d\boldsymbol{A}_{con,k}. \tag{31}$$

When the target velocity of the leader vehicle a_i is given by V_{ri}, the target velocity vector $\boldsymbol{v}_r$ is given by

$$\boldsymbol{v}_r = [v_r^1 \cdots v_r^n]^{\mathrm{T}}, \qquad v_r^i = \begin{cases} V_r^i & \text{if } d_k^i = 0 \\ v_r^{i-1} & \text{otherwise} \end{cases}. \tag{32}$$

3.3 Decision Method of the Switching Frequency by Event-Triggering Mechanism

When considering the calculation cost, we do not desire that the switching of the pinning agents continues in all steps. Therefore, we change the switching frequency according to the convergence rate to the target values. As mentioned in Section 3, we introduce the switching frequency M which is the number of switching per one step. Specifically, we propose an event-triggering mechanism that changes M according to the convergence rate to the target value as follows.

We evaluate the convergence rate by the sum of squares of the differences between target values and states of all vehicles

$$\xi_{e,k} = (\boldsymbol{\xi}_r - \boldsymbol{\xi}_k)^{\mathrm{T}}\boldsymbol{Q}_M(\boldsymbol{\xi}_r - \boldsymbol{\xi}_k) \tag{33}$$

where $\boldsymbol{Q}_M$ is the diagonal weight matrix.

We prepare m switching frequencies $M_1 > \cdots > M_m \in \mathbb{N}$ and express a set of them by $M_{sw} = \{M_1, \cdots, M_m\}$. Moreover, we give the switching frequency as follows:

$$M(\xi_{e,k}) = \begin{cases} M_1 & \xi_{th}^1 < \xi_{e,k} \\ M_i & \xi_{th}^i < \xi_{e,k} \le \xi_{th}^{i-1} \\ M_m & \xi_{e,k} \le \xi_{th}^{m-1} \end{cases} \tag{34}$$

where $\xi_{th} \in \mathbb{R}$ is a basic threshold and ξ_{th}^i is defined as follows:

$$\xi_{th}^i = (r_M)^{i-1}\xi_{th}, \tag{35}$$

where $r_M \in \mathbb{R}\,(0 < r_M < 1)$ is a design parameter of switching frequency. Hereafter, we write $M(\xi_{e,k})$ by M simply unless otherwise noted. The MPC algorithm switches the pinning agents according to the given switching frequency once per $1/M$ steps, as mentioned in Section 3.

Here, we define the interval time T_{M_i} as the time when M equals to M_i in the simulation time T_{sim}. T_{M_i} and T_{sim} satisfy the following equation

$$T_{sim} = \sum_{i=1}^{m} T_{M_i}. \tag{36}$$

Also, we define the approximation number of switching as

$$N_{sw}^{apr} = \sum_{i=1}^{m} M_i \frac{T_{M_i}}{T_s}. \tag{37}$$

N_{sw}^{apr} is the number of switching under the assumption that the mode switches to the different mode every step. On the other hand, in the actual situation, since it is possible that a certain mode is selected continuously, at least, the actual number of switching N_{sw}^{act} is less than or equal to N_{sw}^{apr} i.e.,

$$N_{sw}^{apr} \ge N_{sw}^{act}. \tag{38}$$

Dividing (36) by the sampling time T_s, we get the following equation:

$$\frac{T_{sim}}{T_s} = \sum_{i=1}^{m} \frac{T_{M_i}}{T_s}. \tag{39}$$

From (39), the following inequality holds:

$$M_1 \frac{T_{sim}}{T_s} \ge \sum_{i=1}^{m} M_i \frac{T_{M_i}}{T_s}$$

$$\Leftrightarrow N_{sw,M_1} \geq N_{sw}^{apr} \tag{40}$$

where N_{sw,M_1} is the number of switching when T_{M_1} equals to T_{sim}. Eq. (40) holds only if T_{M_1} equals to T_{sim}. Therefore, even once the switching frequency decreases from the initial frequency M_1, the approximation number of switching also decreases. Moreover, from (38), the actual number of switching also decreases.

3.4 MPC Algorithm

When formulating Problem 1, we consider the following constraint that limits the number of pinning agents to n_p:

$$[1 \cdots 1] \cdot \widehat{\boldsymbol{\delta}}_{k+1|k} = n_p \quad (i = 1, \cdots, N). \tag{41}$$

From the above, Problem 1 is formulated as follows:

Problem (OP): At step k, suppose that the graph $\mathcal{G}_k$, the target value vector $\boldsymbol{\xi}_r \in \mathbb{R}^{2n}$, the states of vehicles $\boldsymbol{\xi}_k \in \mathbb{R}^{2n}$, the mode vector $\boldsymbol{\delta}_k \in \mathbb{R}^2$, the penalty matrix $\boldsymbol{Q}_k \in \mathbb{R}^{2n\times 2n}$, the natural numbers $N \in \mathbb{N}$, $M \in \mathbb{N}$, $n_p \in \mathbb{N}$ are given. Find the solution to the following optimization problem.

$$\text{minimize } J(\widehat{\boldsymbol{\delta}}_{N,k})$$
$$J(\widehat{\boldsymbol{\delta}}_{N,k}) = \sum_{i=1}^{N/M} \left(\boldsymbol{\xi}_r - \widehat{\boldsymbol{\xi}}_{k+i|k}\right)^{\mathrm{T}} \boldsymbol{Q}_k \left(\boldsymbol{\xi}_r - \widehat{\boldsymbol{\xi}}_{k+i|k}\right)$$
$$\boldsymbol{Q}_k = \mathrm{diag}\{0, \cdots, 0, q_k^1, \cdots, q_k^n\} \tag{43}$$
$$q_k^i = \begin{cases} 100 & \text{if } d_k^i = 0 \\ 1 & \text{otherwise} \end{cases}$$
$$\text{s.t. } (18),\ (23),\ \text{and } (41).$$

The penalty matrix $\boldsymbol{Q}_k$ converges the leader preferentially.

Problem (OP) is a Mixed Integer Quadratic Programing problems (MIQP). The external device solves this MIQP according to the following MPC algorithm.

Step 1: Set $k = 0$, $M = M_1$, $M_{step} = 1$, and **go to Step 2**.

Step 2: Observe the states $\boldsymbol{\xi}_k$, and update the mode vector $\boldsymbol{d}_k$, the Graph Laplacian $\boldsymbol{L}_k$, the penalty matrix $\boldsymbol{Q}_k$, and the switching frequency M. Also, update the platoon groups $\mathcal{V}_k^1, \cdots, \mathcal{V}_k^n$ according to rule (9). If $M_{step} \neq 1/M$ and M is not updated, **go to Step 3A**. If $M_{step} = 1/M$ or M is updated according to the event-triggering condition, **go to Step 3B-1**.

Step 3A: The external device does not switch the pinning agents and **go to Step 4**.

Step 3B-1: The external device solves the Problem (OP), and **go to Step 3B-2**

Step 3B-2: Based on (26), switch the pinning agents and set $M_{step} = 0$ and **go to Step 4**.

Step 4: Set $k = k + 1$ and $M_{step} = M_{step} + 1$ and **go back to Step 2**.

At **Step 3**, we assume that the graph $\mathcal{G}_k$ and the penalty matrix $\boldsymbol{Q}_k$ are constant during the prediction.

4. NUMERICAL EXPERIMENTS

In this section, we set parameters as follows:

$$k_i = 0.1,\ c_i = 0.1,\ g_{11} = 0.1, g_{12} = 0.1,\ g_{21} = 0,$$
$$r_M = 1/4,\ n_p = 1,\ M_{sw} = \{1, 1/2, 1/3, 1/5, 1/10\}.$$

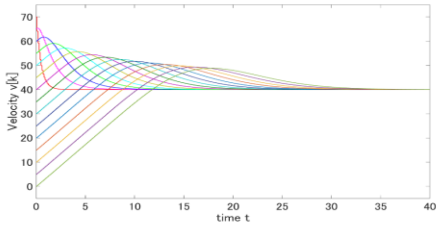
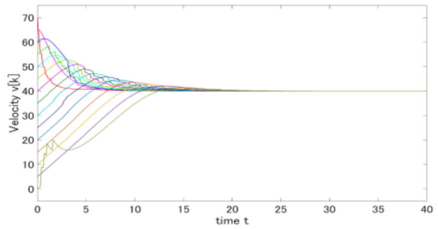
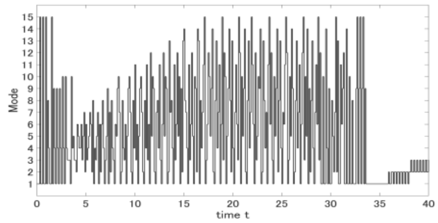


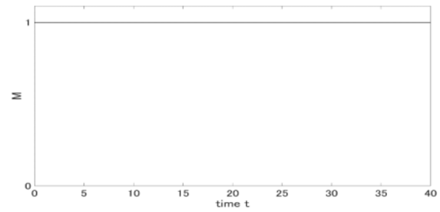
Fig. 2: Normal pinning control in 4.1.



(a) Velocity



(b) Mode



(c) Switching frequency

Fig. 3: SPC

Moreover, we solve Problem (OP) using commercial solver Gurobi 8.1 on MATLAB R2018b.

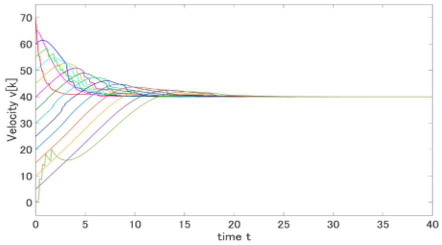
In this section, we apply the event-triggered SPC, the SPC, and the normal pinning control to a leader-follower type of vehicle platoon that consists of 15 vehicles. We assume that there are no formation commands from the external device. The sampling time is $T_s = 0.1$ (s).

Figs. 2~4 are the responses of the event-triggered SPC, the SPC, and the normal pinning control. In Fig. 3 and Fig. 4, we add the mode transition diagram and the switching frequency transition diagram.

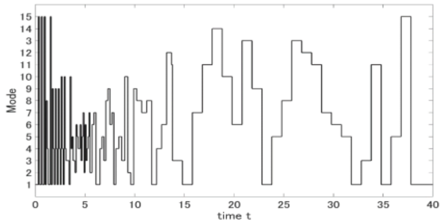
Comparing Figs. 2~ 4, we can see that the convergence speed of the two SPC methods is faster than the one of the normal pinning control method. Moreover, it is clear that the number of switching on the event-triggered SPC decreases as the velocities of platoons converge to the target.

Table 1 shows the settling time T_{st} and the approximation of the number of switching N_{sw} for each method. The settling time is the time elapsed to the time at which the velocities of the vehicles within $\pm 1\%$ of the target velocities.

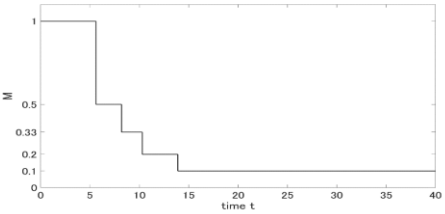
From Table 1, we can see that the settling time of the SPC is about 13 (s) shorter than one of the normal pinning control.



(a) Velocity



(b) Mode



(c) Switching frequency

Fig. 4: Event-Triggered SPC

Moreover, the settling time of the event-triggered SPC is about 1.5 (s) longer than one the SPC. On the other hand, as we state in Section 3.3, the number of switching of the event-triggered SPC is about 1/4 less than one of the SPC.

Table 1 Comparison of settling time and the number of switching

	Settling time T_{st} [s]	The number of switching [times] (approximation)
Normal pinning control	33.0	0
SPC	19.9	About 400
Event-triggered SPC	21.5	About 108

5. CONCLUSIONS

As a vehicle platoon formation method via ITS (Intelligent Transport System), this paper has proposed the event-triggered switched pinning control method. The proposed event triggering mechanism controls the switching frequency of pinning agents according to the convergence rate to the target velocity to reduce the calculation cost of the MPC algorithm. Also, our method regroups the platoons into the arbitrary groups by controlling the adjacent vector. As a result, the MPC algorithm to be installed in the ITS can regroup the platoons into the arbitrary platoons and control their velocity to the target values with a lower computational cost. In Section 4, we have confirmed the reduction of the number of the switching.

Since the class of our optimization problem is a MIQP problem, it may be effective to apply the MIQP solution method based on the path search algorithm [Yoshiki, N. 2019] to our proposed method. This is our future work.

This paper is funded by JSPS KAKENHI Grant Number JP19H02158, JP19K04444, JP19H02163, and JP17H06293.

REFERENCES

Akinori, S. and Toshimitu, U. (2017). Consensus speed of multi-agent systems via pinning control, *Information and Communication Engineers*, Vol. J100-A, No. 7, pp. 229-302.

Alberto, B. and Manfred, M. (1999). control of systems integrating logic, dynamics, and constraints, *Automatica*, Vol. 35, No. 3, pp. 407-427.

Di, C. and Huiping, L. (2019). Self-triggered model predictive control with adaptive selection of sampling number, *IEEE International Conference on Industrial Cyber Physical Systems*, pp. 802-807.

Feng, X. and Long, W. (2007). Consensus problems for high-dimensional multi-agent systems, *IET Control Theory & Applications*, Vol. 1, Issue 3, pp. 830-837.

Jiahu, Q., Qichao, M., Yang, and S., Long, W. (2017). Recent advances in consensus of multi-agent systems: a brief survey, *IEEE Trans. on Industrial Electronics*, Vol. 64, No. 6, pp. 4972-4983.

Jun-ichi, I., Shun-ichi, A., and Izumi, M. (2014). *Control of hybrid systems*, pp. 122-141 *Corona publishing co., ltd.*,

Richard H.M. and Julio, H.B. (2010). String instability in classes of liner time invariant formation control with limited communication range, *IEEE Transactions on Automatic Control*, Vol. 55, Issue. 7, pp. 1519-1530.

Serban, S., Cristian, O., Sean, W., and Ali, J. (2016). Optimal distributed control for platooning via sparse coprime factorizations, *American Control Conference,* pp. 2591-2598.

Shun-ichi, A., Masaaki, N., Hideaki, I., Naoki, H., Kazunori S., and Kenshi H. (2015). *Control of Multi-agent systems*, pp. 21-121, *Corona publishing co., ltd.*,

Takuma W., Yoshiki N., Kenji S., and Seiichi S., (2020). Switched pinning control for vehicle platoons via mixed logical dynamical modelling, *2020 IEEE/SICE International Symposium on system integration* , pp. 449-454.

Weng Y., Ying W., Xiaofan W., Hongbo S., (2016). Optimal Controlled Nodes Selection for Fast Consensus, *Asian Journal of Control*, Vol. 18, No. 3, pp. 932-944.

Xu D., Akinori S., and Toshimitu U. (2018). Stability analysis of pinning consensus control of discrete-time multi-agent systems, *Institute of Electronics, Information and Communication Engineers*, Vol. J101-A, No. 7, pp. 189-195.

Yan S., Wang J., and Wang J., (2018). Coordinated control of vehicle lane change and speed at intersection under V2X, *3rd International Conference on Mechanical, Control and Computer Engineering*, pp. 69-73.

Yongfu, L., Wenbo, C., Srinivas, P., and Yibing, W., (2019). Platoon control of connected multi-agent systems under V2X communications: design and experiments, *IEEE Trans. on Intelligent Transportation Systems*, pp. 1-12.

Yoshiki, N., Kenji, S. and Seiichi, S. (2019). Self-triggered optimal control based on path search algorithm, *SICE Journal of Control, Measurement, and System Integration*, Vol. 12, No. 3, pp. 085-093.