PAPER

Model Predictive Control of Traffic Flow Based on Hybrid System Modeling

Tatsuya KATO[†], *Member*, **YoungWoo KIM**[†a)], *Nonmember*, **Tatsuya SUZUKI**[†], *Member*, *and* **Shigeru OKUMA**[†], *Nonmember*

SUMMARY This paper presents a new framework for traffic flow control based on an integrated model description by means of Hybrid Dynamical System (HDS). The geometrical information on the traffic network is characterized by Hybrid Petri Net (HPN). Then, the algebraic behavior of traffic flow is transformed into Mixed Logical Dynamical Systems (MLDS) form in order to introduce an optimization technique. These expressions involve both continuous evolution of traffic flow and event driven behavior of traffic signal. HPN allows us to easily formulate the problem for complicated and large-scale traffic network due to its graphical understanding. MLDS enables us to optimize the control policy for traffic signal by means of its algebraic manipulability and use of model predictive control framework. Since the behavior represented by HPN can be directly transformed into corresponding MLDS form, the seamless incorporation of two different modeling schemes provide a systematic design scenario for traffic flow control.

key words: *HDS, HPN, MLDS, traffic flow control*

1. Introduction

With the increasing number of automobile and complication of traffic network, the traffic flow control becomes one of significant economic and social issues in urban life. Many researchers have been involved in related researches in order to alleviate traffic congestion. From viewpoint of modeling, the existing scenarios can be categorized into the following two approaches:

(A1) Microscopic approach; and

(A2) Macroscopic approach.

The basic idea of Microscopic approach (A1) is that the behavior of each vehicle is affected by neighboring vehicles, and the entire traffic flow is represented as statistical occurrences. The Cellular Automaton (CA) based model [3], [4] and the Follow-the-Leader (FL) model are widely known ideas to represent the behavior of each vehicle. In the CA model, the road is discretized into many small cells. Each cell can be either empty or occupied by only one vehicle. The behavior of each vehicle in each cell is specified by the geometrical relationship with other vehicles together with some stochastic parameters. Also, in the FL model, each vehicle is supposed to have a tracking response to the preceding vehicle, which is described by first order or second order differential equation. Although many simulation results based on these microscopic models showed high similarity to the measured real data, these approaches are not suitable for the large-scale traffic network modeling available to the controller design because it requires enormous computational efforts to find all vehicles' behavior. Furthermore, the precise information on initial positions and speeds of all vehicles are usually not available in advance.

On the other hand, it has been a common strategy in the macroscopic approach (A2) that the designer uses a fluid approximation model where the behavior of traffic flow is regarded as a continuous fluid with density $k(x,t)$ and volume $q(x,t)$ at location x and time t. In this case, $k(x,t)$ and $q(x,t)$ must satisfy the following law of mass conservation;

$$\frac{\partial k(x,t)}{\partial t} + \frac{\partial q(x,t)}{\partial x} = 0. \tag{1}$$

Also, some relationship among q, k and v, which is usually described by

$$q(x,t) = k(x,t)v(x,t), \tag{2}$$

is introduced together with the appropriate model of the $v(x,t)$, where $v(x,t)$ denotes the velocity of the traffic flow. By incorporating these two equations, the macroscopic behavior of the traffic flow is uniquely decided. This model, however, is applicable only when the density of the traffic flow $k(x,t)$ is continuous. Although this model expresses well the behavior of the flow on the freeway, it is unlikely that this model can be applied to the urban traffic network which involves many discontinuities of the density coming from the existence of intersections controlled by traffic signals. In order to treat the discontinuity of the density in the macroscopic model, the idea of 'shock wave,' which represents the progress of the boundary of two neighboring different density area, has been introduced in [5]–[8]. Although these approaches included judicious use of theoretical ideas as for the flow dynamics, it is not straightforward to exploit them for the design of real-time traffic signal control since the flow model results in complicated nonlinear dynamics.

This paper presents a new method for the real-time traffic signal control based on integrated model descriptions by means of Hybrid Dynamical System (HDS). The geometrical information on the traffic network is characterized by using Hybrid Petri Net (HPN) by both graphical and algebraic descriptions. Then, the algebraic behavior of traffic flow is transformed into Mixed Logical Dynamical Systems

Manuscript received August 6, 2004.
Final manuscript received October 22, 2004.

[†]The authors are with Nagoya University, Nagoya-shi, 464-8603 Japan.
a) E-mail: kim@toyota-ti.ac.jp

(MLDS) form in order to introduce the optimization technique.

In the HPN model, each continuous place represents discretized section of the road, and the continuous marking represents the amount of vehicles (density multiplied by length of the section) in the corresponding section. Also, each discrete place represents the corresponding traffic signal, and the discrete marking (binary valued) represents the state of the traffic signal. Thus, the HPN model can be regarded as one of the discretized macroscopic model of the traffic flow that consists of the event driven behavior of the traffic signal. This implies that the HPN model can be a good model for the urban traffic network which includes many intersections controlled by traffic signals, and also be a core tool for a human-machine interface for the traffic network design due to its graphical understanding.

When we look at the control problem for the traffic signal, it is natural to introduce some algebraic representation of the traffic network. Although the HPN has some algebraic description, it is not a suitable form to formulate the optimization problem. Therefore, the MLDS form, which involves both continuous and logical (binary) evolutions, is introduced to formulate the Model Predictive Control (MPC) scheme for the traffic flow. The MPC for the traffic flow results in the Mixed Integer Quadratic Programming (MIQP), and can be solved by using commercial solver.

The behavior represented by the HPN can be directly transformed into the corresponding MLDS form. The seamless incorporation of two different modeling schemes provides the systematic design scenario for the traffic flow control. Also, the discontinuities of the traffic flow can be easily taken into account due to its discretized modeling fashion in the HPN. Moreover, the discretized modeling in the HPN enables us to control the number of installed sensors according to the required control performance.

In this paper, all traffic signals are supposed to have just two states 'green (go)' and 'red (stop).' No intermediate state (i.e. yellow) is considered to simplify the problem. Also, all signals do not always operate periodically, i.e. all signals can change the state at any time when the controller decides to do so. By removing the constraints of 'periodical operation' of traffic signals, the further optimization as for the traffic flow becomes possible.

2. Modeling of Traffic Flow Control System (TFCS) Based on HPN

The Traffic Flow Control System (TFCS) is the collective entity of traffic network, traffic flow and traffic signals. Although some of them have been fully considered by the previous studies, most of the previous studies did not simultaneously consider all of them. In this section, the HPN model is developed, which provides both graphical and algebraic descriptions for the TFCS.

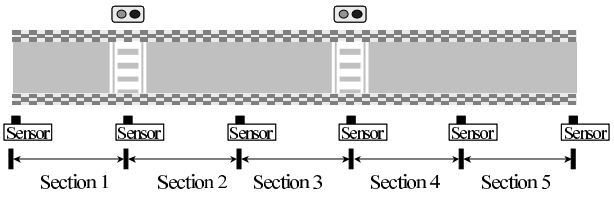

Fig. 1 Straight road.

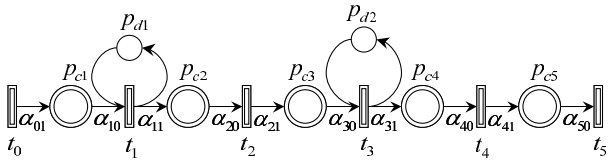

Fig. 2 HPN model of straight road.

2.1 Representation of TFCS as HPN

HPN is one of the useful tools to model and visualize the system behavior with both continuous and discrete variables. Figure 2 shows the HPN model for the road of Fig. 1. In Fig. 2, each section i of l_i-meters long constitutes the straight road, and two traffic lights are installed at the point of crosswalk. HPN has a structure of $N = (P, T, q, I_+, I_-, M^0)$. The set of places P is partitioned into a subset of discrete places P_d and a subset of continuous places P_c. $p_c \in P_c$ represents each section of the road, and has maximum capacity (maximum number of vehicles). Also, P_d represents the traffic signal where green signal is indicated by a token in the corresponding discrete place $p_d \in P_d$. The marking $M = [\boldsymbol{m_C}|\boldsymbol{m_D}]$ has both continuous (m dimension) and discrete (n dimension) parts where $\boldsymbol{m_C}$ represents the number of vehicles in the corresponding continuous places, and $\boldsymbol{m_D}$ denotes the state of the corresponding traffic signal (i.e. takes binary value). Note that each signal is supposed to have only two states 'go (green)' or 'stop (red)' for simplicity. T is the set of continuous transitions which represent the boundary of two successive sections. The function $q_j(\tau)$ specifies the firing speeds assigned to transition $t_j \in T$ at time τ. $q_j(\tau)$ represents the number of vehicles passing through the boundary of two successive sections (measuring position) at time τ. Note that sensors to capture the number of the vehicles are supposed to be installed at every boundaries of the section as show in Fig. 1. Also, we do not consider any measurement error for sensors in this paper. The function $I_+(p,t)$ and $I_-(p,t)$ are forward and backward incidence relationships between transition t and place p which follows and precedes the transition. The element of $I(p,t)$ is always 0 or α_{ij}. α_{ij} is the number of traffic lanes in each section. Finally, M^0 is specified as the initial marking of the place $p \in P$. The net dynamics of HPN is represented by a simple first order differential equation for each continuous place $p_{c_i} \in P_c$ as follows:

$$\frac{dm_{C,i}(\tau)}{dt} = \sum_{t_j \in p_{c_i}{}^{\bullet} \cup {}^{\bullet} p_{c_i}} I(p_{c_i}, t_j) \cdot q_j(\tau) \cdot m_{D,j}(\tau), \tag{3}$$

where $m_{C,i}(\tau)$ is the marking for the place $p_{c_i}(\in P_c)$ at time τ, $m_{D,j}(\tau)$ is the marking for the place $p_{d_j}(\in P_d)$, and $I(p,t) = I_+(p,t) - I_-(p,t)$. The equation (3) is transformed to its discrete-time version supposing that $q_j(\tau)$ is constant during two successive sampling instants as follows:

$$m_{C,i}((\kappa+1)T_s) = m_{C,i}(\kappa T_s) + \sum_{t_j \in p_{c_i}{}^{\bullet} \cup {}^{\bullet} p_{c_i}} I(p_{c_i}, t_j) \cdot q_j(\kappa T_s) \cdot m_{D,j}(\kappa T_s) \cdot T_s. \tag{4}$$

where κ is sampling index, and T_s is sampling period.

Note that the transition t is *enabled* at the sampling instant κT_s if the marking of its preceding discrete place $p_{d_j} \in P_d$ satisfies $m_{D,j}(\kappa) \geq I_+(p_{d_j}, t)$. Also if t does not have any input (discrete) place, t is always *enabled*.

2.2 Definition of Flow q_i

In order to derive the flow behavior, the relationship among $q_i(\tau)$, $k_i(\tau)$ and $v_i(\tau)$ must be specified. One of the simple ideas is to use the well-known model

$$q_i(\tau) = \frac{(k_i(\tau) + k_{i+1}(\tau))}{2} \frac{v_i(k_i(\tau)) + v_{i+1}(k_{i+1}(\tau))}{2} \tag{5}$$

supposing that the density $k_i(\tau)$ and $k_{i+1}(\tau)$, and average velocity $v_i(\tau)$ and $v_{i+1}(\tau)$ of the flow in i and $(i+1)$th sections are almost identical. Then, by incorporating the velocity model

$$v_i(\tau) = v_{f_i} \cdot \left(1 - \frac{k_i(\tau)}{k_{jam}}\right), \tag{6}$$

with (5), the flow dynamics can be uniquely defined. Here, k_{jam} is the density in which the vehicles on the roadway are spaced at minimum intervals (traffic-jammed), and v_{f_i} is the maximum speed, that is, the velocity of the vehicle when no other vehicle exists in the same section.

If there exist no abrupt change in the density on the road, this model is expected to work well. However, in the urban traffic network, this is not the case due to the existence of the intersections controlled by the traffic signals. In order to treat the discontinuities of the density among neighboring sections (i.e. neighboring continuous places), the idea of 'shock wave' [11] is introduced as follows. We consider the case as shown in Fig. 3 where the traffic density of ith section is lower than that of $(i+1)$th section in which the boundary of density difference designated by the dotted line is moving forward. Here, the movement of this boundary is called shock wave and the moving velocity of the shock wave $c_i(\tau)$ depends on the densities and average velocities of ith and $(i+1)$th sections as follows [11]:

$$c_i(\tau) = \frac{v_i(\tau)k_i(\tau) - v_{i+1}(\tau)k_{i+1}(\tau)}{k_i(\tau) - k_{i+1}(\tau)}. \tag{7}$$

The traffic situation can be categorized into the following four types taking into account the density and shock wave.

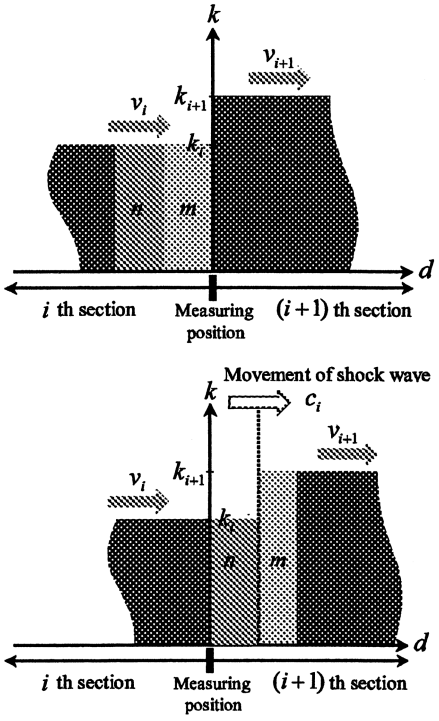

Fig. 3 Movement of shock wave in the case of $k_i(\tau) < k_{i+1}(\tau)$ and $c_i(\tau) > 0$.

(C1) $k_i(\tau) < k_{i+1}(\tau)$, and $c_i(\tau) > 0$,

(C2) $k_i(\tau) < k_{i+1}(\tau)$, and $c_i(\tau) \leq 0$,

(C3) $k_i(\tau) > k_{i+1}(\tau)$,

(C4) $k_i(\tau) = k_{i+1}(\tau)$ (no shock wave).

Firstly, in both cases of (C1) and (C2) where $k_i(\tau)$ is smaller than $k_{i+1}(\tau)$, the vehicles passing through the density boundary (dotted line) reduce their speeds. The movement of the shock wave is illustrated in Fig. 3 ($c_i(\tau) > 0$) and Fig. 4 ($c_i(\tau) \leq 0$). In Figs. 3 and 4, the 'measuring position' implies the position where transition t_i is assigned. Since the traffic flow $q_i(\tau)$ represents the number of vehicle passing through the measuring position per unit time, in the case of (C1), it can be represented by $n+m$ in Fig. 3, where n and m represent the area of the corresponding rectangular, i.e. the product of the $v_i(\tau)$ and $k_i(\tau)$. Similarly, in the case of (C2), $q_i(\tau)$ can be represented by m in Fig. 4. These considerations lead to the following models:

in the case of (C1)

$$q_i(\tau) = v_i(\tau)k_i(\tau) \tag{8}$$

$$= v_{f_i}\left(1 - \frac{k_i(\tau)}{k_{jam}}\right)k_i(\tau), \tag{9}$$

in the case of (C2)

$$q_i(\tau) = v_{i+1}(\tau)k_{i+1}(\tau) \tag{10}$$

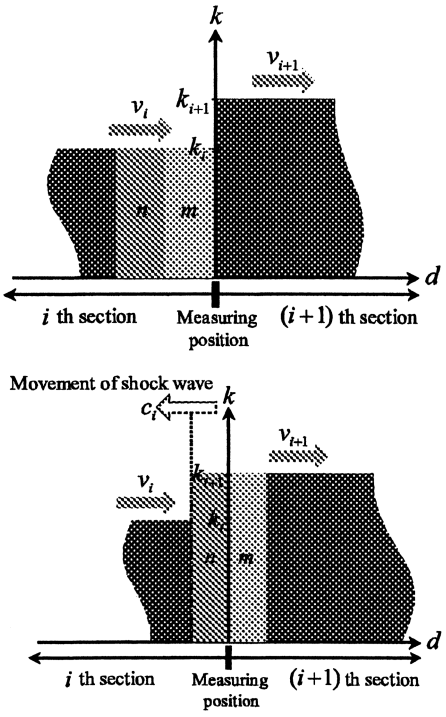

Fig. 4 Movement of shock wave in the case of $k_i(\tau) < k_{i+1}(\tau)$ and $c_i(\tau) \leq 0$.

$$= v_{f_{i+1}}\left(1 - \frac{k_{i+1}(\tau)}{k_{jam}}\right)k_{i+1}(\tau). \tag{11}$$

In the cases of (C3) and (C4) where $k_i(\tau)$ is greater than $k_{i+1}(\tau)$, the vehicles passing through the density boundary come to accelerate. In this case, the flow can be well approximated by taking into account the average density of neighboring two sections. This is intuitively because the difference of the traffic density is going down. Then in the cases of (C3) and (C4), the traffic flow can be formulated as follows:

in the cases of (C3) and (C4),

$$q_i(\tau) = \left(\frac{k_i(\tau)+k_{i+1}(\tau)}{2}\right)v_f(\tau)\left(1 - \frac{k_i(\tau)+k_{i+1}(\tau)}{2k_{jam}}\right). \tag{12}$$

As the results, the flow model (8) ∼ (12) taking into account the discontinuity of the density can be summarized as follows:

$$q_i(\tau) = \begin{cases} \left(\frac{k_i(\tau)+k_{i+1}(\tau)}{2}\right)v_f\left(1 - \frac{k_i(\tau)+k_{i+1}(\tau)}{2k_{jam}}\right) \\ \quad if\ \ k_i(\tau) \geq k_{i+1}(\tau) \\ v_{f_i}\left(1 - \frac{k_i(\tau)}{k_{jam}}\right)k_i(\tau) \\ \quad if\ \ k_i(\tau) < k_{i+1}(\tau) \text{ and } c(\tau) > 0 \\ v_{f_{i+1}}\left(1 - \frac{k_{i+1}(\tau)}{k_{jam}}\right)k_{i+1}(\tau) \\ \quad if\ \ k_i(\tau) < k_{i+1}(\tau) \text{ and } c(\tau) \leq 0 \end{cases}. \tag{13}$$

2.3 Verification of the Derived Flow Model

In this subsection, we verify the effectiveness of the proposed traffic flow model developed in the previous subsection by comparing it with the microscopic model. The usefulness of Cellular Automaton (CA) in representing the traffic flow behavior was investigated in [3]. Some of well-known traffic flow simulators such as, TRANSIMS and MICROSIM are based on CA model.

The essential property of CA is characterized by its lattice structure where each cell represents a small section on the road. Each cell may include one vehicle or not. The evolution of CA is described by some rules which describe the evolution of the state of each cell depending on the states of its adjacent cells.

The evolution of the state of each cell in CA model can be expressed by

$$n_j(\tau + 1) = n_j^{in}(\tau)(1 - n_j(\tau)) - n_j^{out}(\tau), \tag{14}$$

where $n_j(\tau)$ is the state of each cell which represents the occupation by the vehicle in the jth cell ($n_j(\tau) = 0$ implies that a jth cell is empty, and $n_j(\tau) = 1$ implies that a vehicle is present in the jth cell at τ). $n_j^{in}(\tau)$ represents the state of the cell from which the vehicle moves to the jth cell, and $n_j^{out}(\tau)$ indicates the state of the destination cell leaving from the jth cell. In order to find $n_j^{in}(\tau)$ and $n_j^{out}(\tau)$, some rules are adopted as follows:

Acceleration rule : All vehicles, that have not reached at the speed of maximum speed v_f, accelerate its speed $v_{\langle j\rangle}(\tau)$ by one unit velocity v_{unit} as follows:

$$v_{\langle j\rangle}(\tau + \Delta\tau) \equiv v_{\langle j\rangle}(\tau) + v_{unit}. \tag{15}$$

Safety distance rule : If a vehicle has e empty cells in front of it, then the velocity at the next time instant $v_{\langle j\rangle}(\tau + \Delta\tau)$ is restricted as follows:

$$v_{\langle j\rangle}(\tau + \Delta\tau) \equiv \min\{e, v_{\langle j\rangle}(\tau + \Delta\tau)\}. \tag{16}$$

Randomization rule : With probability p, the velocity is reduced by one unit velocity as follows:

$$v_{\langle j\rangle}(\tau + \Delta\tau) \equiv v_{\langle j\rangle}(\tau + \Delta\tau) - p \cdot v_{unit}. \tag{17}$$

Figure 5 shows the behavior of traffic flow obtained by applying the CA model to the two successive sections which is 450 [m] long. The parameters used in the simulation are as follows: computational interval $\Delta\tau$ is 1 [sec], each cell in the CA is assigned to 4.5 [m]-long interval on the road, maximum speed v_f is 5 (cells/$\Delta\tau$), which is equivalent to 81 [km/h] (=4.5 [m/cell] · 5 [cells/$\Delta\tau$] · 3600 [sec]/1000). The left figure of Fig. 5 shows the obtained relationship among normalized flow $q_i(\tau)$ and densities $k_i(\tau)$ and $k_{i+1}(\tau)$. The right small figure is the abstracted illustration of the real

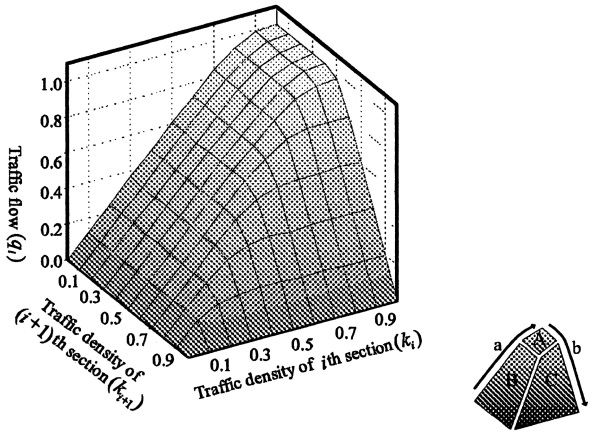

Fig. 5 Traffic flow behavior obtained from CA model.

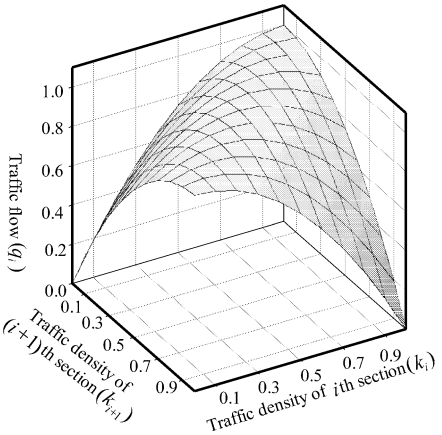

Fig. 7 Traffic flow behavior obtained by averaging k_i and k_{i+1}.

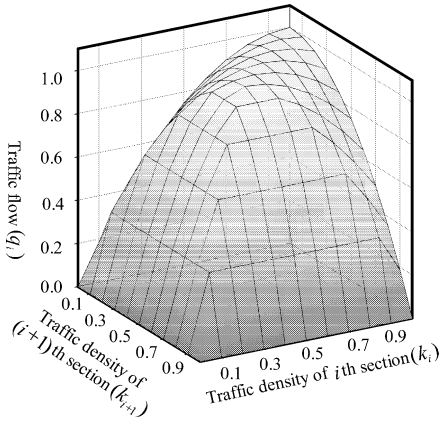

Fig. 6 Traffic flow behavior obtained from the proposed traffic flow model.

behavior.

First of all, we look at the behavior along the edge a in the right figure which implies the case that the traffic signal is changed from red to green. At the point of $k_i(\tau) = 0$ and $k_{i+1}(\tau) = 0$, the traffic flow $q_i(\tau)$ becomes zero since there is no vehicle in both ith and $(i + 1)$th section. Then, $q_i(\tau)$ is proportionally increased as $k_i(\tau)$ increases, and reaches at the saturation point ($k_i(\tau) = 0.9$). Next, we look at the behavior along the edge b which implies that the ith section is fully occupied. In this case, the maximum flow is measured until the density of the $(i + 1)$th section is reduced by 50% (i.e. $k_{i+1}(\tau) = 0.5$), and after that the flow goes down according to the increase of $k_{i+1}(\tau)$. Although CA model consists of quite simple procedures, it can show quite natural traffic flow behavior.

On the other hand, Fig. 6 shows the behavior in case of using HPN where proposed flow model given by (13) is embedded. we can see that Fig. 6 shows the similar characteristics to Fig. 5, especially, the saturation characteristic is well represented despite of the use of macroscopic model. As another simple modeling strategy, we consider the case that the average of two $k_i(\tau)$ and $k_{i+1}(\tau)$ are used to decide the flow $q_i(\tau)$ (i.e. use (12)) for all cases. Figure 7 shows the behavior in case of using HPN where the flow model is supposed to be given by (12) for all cases. Although the $q_i(\tau)$ shows similar characteristics in the region of $k_i(\tau) \geq k_{i+1}(\tau)$, at the point of $k_i(\tau) = 0$ and $k_{i+1}(\tau) = k_{jam}$, $q_i(\tau)$ takes its maximum value. This obviously contradicts to the natural flow behavior.

Before concluding this subsection, it is worthwhile to compare the computational amount. In the case of using CA,we need 140 seconds. On the other hand, in case of using HPN and (13), we need only 0.06 seconds.

3. Transformation to MLDS Form

Although the HPN can represent the hybrid dynamical behavior of TFCS including both continuous traffic flow and discrete traffic signal control, it is still not well formulated when some optimization problem is addressed. In this section, the MLDS form is introduced to formulate the Model Predictive Control (MPC) stated in the next section.

The MLDS form can generally be formalized as follows [13]:

$$\begin{aligned} \boldsymbol{x}(\kappa+1) &= \boldsymbol{A}_{\kappa}\boldsymbol{x}(\kappa) + \boldsymbol{B}_{1\kappa}\boldsymbol{u}(\kappa) \\ &\quad + \boldsymbol{B}_{2\kappa}\boldsymbol{\delta}(\kappa) + \boldsymbol{B}_{3\kappa}\boldsymbol{z}(\kappa), \end{aligned} \tag{18}$$

$$\begin{aligned} \boldsymbol{y}(\kappa) &= \boldsymbol{C}_{\kappa}\boldsymbol{x}(\kappa) + \boldsymbol{D}_{1\kappa}\boldsymbol{u}(\kappa) \\ &\quad + \boldsymbol{D}_{2\kappa}\boldsymbol{\delta}(\kappa) + \boldsymbol{D}_{3\kappa}(\kappa), \end{aligned} \tag{19}$$

$$\begin{aligned} &\boldsymbol{E}_{2\kappa}\boldsymbol{\delta}(\kappa) + \boldsymbol{E}_{3\kappa}\boldsymbol{z}(\kappa) \leq \\ &\quad \boldsymbol{E}_{1\kappa}\boldsymbol{u}(\kappa) + \boldsymbol{E}_{4\kappa}\boldsymbol{x}(\kappa) + \boldsymbol{E}_{5\kappa}. \end{aligned} \tag{20}$$

In the MLDS form,κ represents the sampling index. Note that sampling period T_s is eliminated the following. Equations (18), (19) and (20) are state equation, output equation and constraint inequality, respectively, where $x(\kappa)$, $y(\kappa)$ and $u(\kappa)$ are the state, output and input variable, whose components are constituted by continuous and/or 0-1 binary variables, $\delta(\kappa) \in \{0, 1\}$ and $z(\kappa) \in \Re$ represent auxiliary logical (binary) and continuous variables. The MLDS is known to be able to represent other forms of HDS such as Piece-Wise Affine (PWA), Hybrid Automaton (HA) and so on.

In the TFCS represented by HPN, equation (4) is directly transformed to the state equation in MLDS form by regarding the continuous marking as the state variable. Also, the TFSC only has binary input variable which denotes the state of the traffic signal (i.e. green or red). The output variable is not specified in our problem setting since all states are supposed to be measurable in this work.

The constraint inequality of (20) often plays an essential role to represent some nonlinearity which exists in the original system. In the TFCS, the nonlinearity appears in (13). In the following, this nonlinear constraint is transformed to the set of linear inequality constraints. The flow model developed in the previous section (shown in Fig. 5) can be approximated by the Piece-Wise Affine (PWA) model shown in the right figure of Fig. 5, which consists of three planes as follows:

Plane A: The traffic flow $q_i(\kappa)$ is saturated ($k_i(\kappa) \geq a$ and $k_{i+1}(\kappa) \leq (k_{jam} - a)$).

Plane B: The traffic flow $q_i(\kappa)$ is mainly affected by the quantity of traffic density $k_i(\kappa)$ ($k_i(\kappa) < a$ and $k_i(\kappa) + k_{i+1}(\kappa) \leq k_{jam}$).

Plane C: The traffic flow q_i is mainly affected by the quantity of traffic density $k_{i+1}(\kappa)$ ($k_{i+1}(\kappa) > k_{jam} - a$ and $k_i(\kappa) + k_{i+1}(\kappa) > k_{jam}$).

Here, a is the threshold value to specify the region of saturation characteristic of the traffic flow, that is, if $k_i(\kappa) \geq a$ and $k_{i+1}(\kappa) < k_{jam} - a$, the $q_i(\kappa)$ takes almost its maximum value q_{max}.

Figure 8 shows these partitions on $k_{i+1}(\kappa) - k_i(\kappa)$ plane. In order to derive the linear inequalities expression of the flow model, three auxiliary variables $\delta_{P,i,1}(\kappa)$, $\delta_{P,i,2}(\kappa)$ and $\delta_{P,i,3}(\kappa)$ are introduced, and are defined as follows:

$$[\delta_{P,i,1}(\kappa) = 1] \leftrightarrow \begin{cases} k_i(\kappa) & \geq & a \\ k_{i+1}(\kappa) & \leq & k_{jam} - a \end{cases}, \quad [\delta_{P,i,2}(\kappa) = 1] \tag{21}$$

$$[\delta_{P,i,1}(\kappa) = 1] \leftrightarrow \begin{cases} k_i(\kappa) & \leq & a - \varepsilon \\ k_i(\kappa) + k_{i+1}(\kappa) & \leq & k_{jam} \end{cases}, \quad [\delta_{P,i,3}(\kappa) = 1] \tag{22}$$

$$\leftrightarrow \begin{cases} k_{i+1}(\kappa) & \geq & k_{jam} - a + \varepsilon \\ k_i(\kappa) + k_{i+1}(\kappa) & \geq & k_{jam} + \varepsilon \end{cases}, \tag{23}$$

$$\delta_{P,i,1}(\kappa) + \delta_{P,i,2}(\kappa) + \delta_{P,i,3}(\kappa) = 1, \tag{24}$$

where ε is a small tolerance.

By using these binary variables, the flow model $q_i(\kappa)$ given by (13) can be rewritten in a compact linear form as follows:

$$q_i(\kappa) = q_{max}\delta_{P,i,1}(\kappa) + \frac{q_{max}k_i(\kappa)}{a}\delta_{P,i,2}(\kappa) + \frac{q_{max}(1 - k_{i+1}(\kappa))}{a}\delta_{P,i,3}(\kappa), \tag{25}$$

$$\sum_{i=1}^{3} \delta_{P,i,j}(\kappa) = 1,$$

where $0 \leq k_i(\kappa) \leq k_{jam}$, $0 \leq k_{i+1}(\kappa) \leq k_{jam}$ (= 1), and q_{max} is the maximum value of the traffic flow.

Figure 9 shows the PWA model of the flow in the case of $a = 0.3$ and $q_{max} = 1$, which approximates the nonlinear flow model developed in the previous section,

The equations (21) to (23) can be generalized as follows:

$$[\delta_{P,i,j}(\kappa) = 1] \leftrightarrow \left[\begin{bmatrix} k_i(\kappa) \\ k_{i+1}(\kappa) \end{bmatrix} \in \mathfrak{R}_j\right], \tag{26}$$

$$\mathfrak{R}_j = \left\{\begin{bmatrix} k_i(\kappa) \\ k_{i+1}(\kappa) \end{bmatrix} : \boldsymbol{S_j}\boldsymbol{k_i}(\kappa) \leq \boldsymbol{T_j}\right\}, \tag{27}$$

where $\boldsymbol{k_i}(\kappa) = [k_i(\kappa)\ k_{i+1}(\kappa)]^T$ and $\boldsymbol{S_j}$ and $\boldsymbol{T_j}$ are the matrices with suitable dimensions. Also, these logical conditions can be transformed to following inequalities.

$$\boldsymbol{S_j}\boldsymbol{k_i}(\kappa) - \boldsymbol{T_j} \leq \boldsymbol{M_j}^*[1 - \delta_{P,i,j}(\kappa)] \tag{28}$$

$$\boldsymbol{M_j^*} \triangleq \max_{\boldsymbol{k_i} \in \mathfrak{R}_j} \boldsymbol{S_j}\boldsymbol{k_i}(\kappa) - \boldsymbol{T_j} \tag{29}$$

The flow $q_i(\kappa)$ of (25) can be represented by the vector form by using

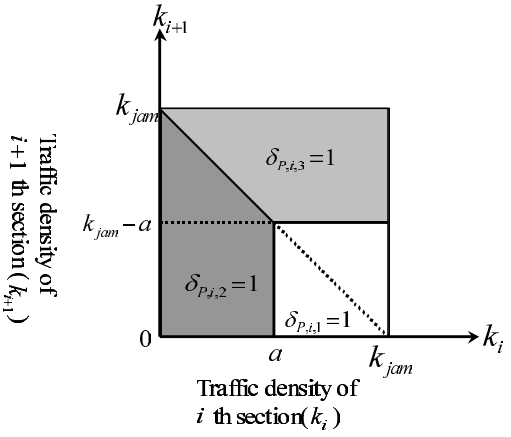

Fig. 8 Division of flow model by introducing auxiliary variables.

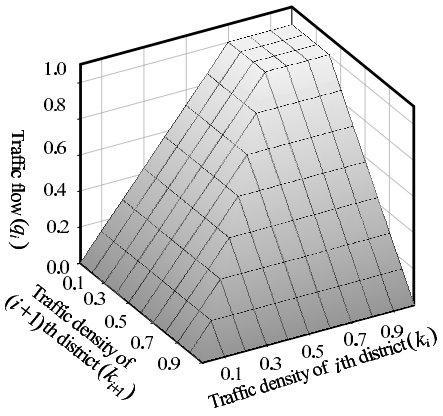

Fig. 9 Traffic flow behavior obtained by PWA model.

$\boldsymbol{\delta_{P,i}}(\kappa) = [\delta_{P,i,1}(\kappa)\ \delta_{P,i,2}(\kappa)\ \delta_{P,i,3}(\kappa)]$ as follows:

$$q_i(\kappa) = f(\boldsymbol{\delta_{P,i}}(\kappa),\ \boldsymbol{k_{i+1}}(\kappa)) \tag{30}$$

$$= \sum_{j=1}^{3} (\boldsymbol{f_i^j}(\kappa)\boldsymbol{k_i}(\kappa) + h_i^j)\delta_{P,i,j}(\kappa), \tag{31}$$

where $\boldsymbol{f_i^j}$ and h_i^j are given as follows (see Fig. 9):

$$\boldsymbol{f_i^1} = [\ 0 \quad 0\ ], \tag{32}$$

$$h_i^1 = q_{max}, \tag{33}$$

$$\boldsymbol{f_i^2} = [\ \tfrac{q_{max}}{a} \quad 0\ ], \tag{34}$$

$$h_i^2 = 0, \tag{35}$$

$$\boldsymbol{f_i^3} = [\ 0 \quad -\tfrac{q_{max}}{a}\ ], \tag{36}$$

$$h_i^3 = \frac{q_{max}}{a}. \tag{37}$$

Next, we introduce an auxiliary variable 'controlled traffic flow' $\boldsymbol{z_i}(\kappa) = [z_{i,1}(\kappa)\ z_{i,2}(\kappa)\ z_{i,3}(\kappa)]$ which implies the flow under the traffic signal control. $z_{i,j}(\kappa)$ is defined by

$$z_{i,j}(\kappa) = (\boldsymbol{f_i^j}(\kappa)\boldsymbol{k_i}(\kappa) + h_i^j)u_i(\kappa)\delta_{P,i,j}(\kappa). \tag{38}$$

That is,

$$q_i(\kappa)u_i(\kappa) = \sum_{j=1}^{3} z_{i,j}(\kappa), \tag{39}$$

where $u_i(\kappa) \in \{0, 1\}$ denotes the binary control input which represents the state of traffic signal. Then the equivalent inequalities to (38) are given as follows:

$$z_{i,j}(\kappa) \le M_i u_i(\kappa)\delta_{P,i,j}(\kappa), \tag{40}$$

$$z_{i,j}(\kappa) \ge m_i u_i(\kappa)\delta_{P,i,j}(\kappa), \tag{41}$$

$$z_{i,j}(\kappa) \le \boldsymbol{f_i^j}\boldsymbol{k_i}(\kappa) + h_i^j - m_i(1 - u_i(\kappa)\delta_{P,i,j}(\kappa)), \tag{42}$$

$$z_{i,j}(\kappa) \ge \boldsymbol{f_i^j}\boldsymbol{k_i}(\kappa) + h_i^j - M_i(1 - u_i(\kappa)\delta_{P,i,j}(\kappa)), \tag{43}$$

where M_i and m_i are

$$M_i = \max_{\boldsymbol{k_i}(\kappa)\in\Re_j} \left\{\boldsymbol{f_i^j}\boldsymbol{k_i}(\kappa) + h_i^j\right\}, \tag{44}$$

$$m_i = \min_{\boldsymbol{k_i}(\kappa)\in\Re_j} \left\{\boldsymbol{f_i^j}\boldsymbol{k_i}(\kappa) + h_i^j\right\}. \tag{45}$$

The product term $u_i(\kappa)\ \delta_{P,i,j}(\kappa)$ can also be replaced by another auxiliary logical variable $\delta_{M,i,j}(\kappa) = u_i(\kappa)\delta_{P,i,j}(\kappa)$ in order to linearize the constraints. This constraint can be transformed to the equivalent inequalities as follows:

$$-u_i(\kappa) + \delta_{M,i,j}(\kappa) \le 0, \tag{46}$$

$$-\delta_{P,i,j}(\kappa) + \delta_{M,i,j}(\kappa) \le 0, \tag{47}$$

$$u_i(\kappa) + \delta_{P,i,j}(\kappa) + \delta_{M,i,j}(\kappa) \le 1. \tag{48}$$

As the results, the MLDS form for the TFCS can be formalized as follows:

$$\boldsymbol{x}(\kappa + 1) = \boldsymbol{A}\boldsymbol{x}(\kappa) + \boldsymbol{B}\boldsymbol{z}(\kappa), \tag{49}$$

$$\boldsymbol{z}(\kappa) = diag(\boldsymbol{C}\boldsymbol{u}(\kappa))\boldsymbol{\delta}(\kappa), \tag{50}$$

$$\boldsymbol{E_2}\boldsymbol{\delta}(\kappa) + \boldsymbol{E_3}\boldsymbol{z}(\kappa) \le \boldsymbol{E_1}\boldsymbol{u}(\kappa) + \boldsymbol{E_4}\boldsymbol{x}(\kappa) + \boldsymbol{E_5}, \tag{51}$$

where the element $x_i(\kappa)$ of $\boldsymbol{x}(\kappa) \in \Re^{|P|}$, is the marking of the place p_{c_i} at the sampling index κ, the element $u_i(\kappa)(\in \{0, 1\})$ of $\boldsymbol{u}(\kappa) \in Z^{|T|}$, is the state of the traffic signal installed at ith section and $\boldsymbol{\delta}(\kappa)=[\boldsymbol{\delta_P}(\kappa), \boldsymbol{\delta_M}(\kappa)]'$. Note that if there is no traffic signal installed at ith section, $u_i(\kappa)$ is always set to be one. The example of matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{E_1}$, $\boldsymbol{E_2}$, $\boldsymbol{E_3}$, $\boldsymbol{E_4}$ and $\boldsymbol{E_5}$ of Fig. 1 are described in the appendix.

4. Model Predictive Control (MPC) for TFCS

The Model Predictive Control (MPC) [14], [15] is one of the well-known paradigms for optimizing the systems with constraints and uncertainties. The Receding Horizon Control (RHC) policy is the key idea to realize the MPC. In RHC, finite-horizon optimization is carried out based on the measured state at each sampling instant, and only the first control input is applied to the controlled plant. In this section, firstly, the RHC policy is briefly reviewed, then the optimization problem for TFCS is formulated as the Mixed Integer Linear Programming (MILP). Finally, some idea to reduce the computational amount is described.

The proposed method does not explicitly optimize the conventional signal control parameters, such as 'Cycle,' 'Offset' and 'Split.' However, since all signals are supposed to be able to change the state at any time when the controller decides to do so, the conventional signal control parameters are optimized implicitly. By removing the constraints as for these parameters, further optimization becomes possible. Note that if the designer would like to impose some constraints on these parameters (for example, constant offset), it can be embedded in our problem setup in a straightforward manner.

4.1 RHC for TFCS

In RHC policy, the control input at each sampling instant is decided based on the prediction of the behavior for next several sampling periods called the prediction horizon.

In order to formulate the optimization procedure, firstly, equation (49) is modified to evaluate the state and input variables in the prediction horizon as follows:

$$\boldsymbol{x}(\kappa + \lambda|\kappa) = \boldsymbol{A^\kappa}x(\kappa) + \sum_{\eta=0}^{\lambda-1}\{\boldsymbol{A^\eta}(\boldsymbol{B}(diag(\boldsymbol{C}\boldsymbol{u}(\kappa + \lambda - 1 - \eta|\kappa))) \cdot\boldsymbol{\delta}(\kappa + \lambda - 1 - \eta|\kappa))\}, \tag{52}$$

where $x(\kappa + \lambda|\kappa)$ denotes the predicted state vector at sampling index $\kappa + \lambda$, which is obtained by applying the input sequence, $u(\kappa), \cdots, u(\kappa + \lambda)$ to (49) starting from the state

$x(\kappa|\kappa) = x(\kappa)$.

Now we consider the following control requirements that usually appear in TFCS.

(R1) Maximize the traffic flow over entire traffic network.

(R2) Avoid the frequent change of traffic signal.

(R3) Avoid the concentration of traffic mass in a certain section.

These requirements can be realized by minimizing the following objective function.

$$\begin{aligned} &J(\boldsymbol{u}(\kappa|\kappa), \cdots, \boldsymbol{u}(\kappa+N|\kappa) \\ &\quad, \boldsymbol{x}(\kappa|\kappa), \cdots, \boldsymbol{x}(\kappa+N|\kappa) \\ &\quad, \boldsymbol{\delta}(\kappa|\kappa), \cdots, \boldsymbol{\delta}(\kappa+N|\kappa)) \\ &= \sum_{\lambda=1}^{N} \Bigg\{ - \sum_i w_{1,i} \Bigg\{ \Bigg(\boldsymbol{\Theta_i} \begin{bmatrix} x_i(\kappa+\lambda|\kappa)/l_i \\ x_{i+1}(\kappa+\lambda|\kappa)/l_{i+1} \end{bmatrix} \\ &\quad + \boldsymbol{\Phi_i} \Bigg)' \boldsymbol{\delta_{M,i}}(\kappa+\lambda|\kappa) \Bigg\} \\ &\quad - \sum_i w_{2,i} \Big\{ 1 - \big| u_i(\kappa+\lambda|\kappa) - u_i(\kappa+\lambda+1|\kappa) \big| \Big\} \\ &\quad + \sum_i w_{3,i} \Bigg\{ \left| \frac{x_i(\kappa+\lambda|\kappa)}{l_i} - \frac{x_{i+1}(\kappa+\lambda|\kappa)}{l_{i+1}} \right| \Bigg\} \Bigg\}. \end{aligned} \tag{53}$$

where

$$\boldsymbol{\Theta_i} = \begin{bmatrix} 0 & 0 \\ \frac{q_{max}}{a} & 0 \\ 0 & -\frac{q_{max}}{a} \end{bmatrix}, \tag{54}$$

$$\boldsymbol{\Phi_i} = \begin{bmatrix} q_{max} \\ 0 \\ \frac{q_{max} x_i(\kappa)}{a l_i} \end{bmatrix}, \tag{55}$$

N denotes the prediction horizon. Also, $w_{1,i}$, $w_{2,i}$ and $w_{3,i}$ are positive weighting parameters for ith section which satisfy $w_{1,i} + w_{2,i} + w_{3,i} = 1$, and $0 \le w_{1,i} \le 1$, $0 \le w_{2,i} \le 1$ and $0 \le w_{3,i} \le 1$. The three terms in the left side of (53) correspond to the requirement (R1), (R2) and (R3), respectively.

As the results, the optimization problem can be formulated as follows:

find
$\boldsymbol{\delta}(\kappa+\lambda|\kappa) = [\boldsymbol{\delta_P}(\kappa+\lambda|\kappa), \boldsymbol{\delta_M}(\kappa+\lambda|\kappa)]'$
$(\lambda = 1, \cdots N)$
which minimizes (53)

subject to (28), (29), (32), (33), (34), (35), (36), (37), (39), (40), (41), (42), (43), (44), (45), (46), (47), (48), (49), (50), *and* (51)

The objective function (53) contains absolute functions. Although they are not directly tractable as the MILP formulation, the introduction of new variables makes it possible to handle as the MILP.

$$\begin{aligned} &J(\boldsymbol{u}(\kappa|\kappa), \cdots, \boldsymbol{u}(\kappa+N|\kappa) \\ &\quad, \boldsymbol{x}(\kappa|\kappa), \cdots, \boldsymbol{x}(\kappa+N|\kappa) \\ &\quad, \boldsymbol{\delta}(\kappa|\kappa), \cdots, \boldsymbol{\delta}(\kappa+N|\kappa)) \\ &= \sum_{\lambda=1}^{N} \Bigg\{ - \sum_i w_{1,i} \Bigg\{ \Bigg(\boldsymbol{\Theta_i} \begin{bmatrix} x_i(\kappa+\lambda|\kappa)/l_i \\ x_{i+1}(\kappa+\lambda|\kappa)/l_{i+1} \end{bmatrix} \\ &\quad + \boldsymbol{\Phi_i} \Bigg)' \boldsymbol{\delta_{M,i}}(\kappa+\lambda|\kappa) \Bigg\} \\ &\quad - \sum_i w_{2,i} \Big\{ 1 - \big(e^+_{u,i}(\kappa+\lambda|\kappa) + e^-_{u,i}(\kappa+\lambda|\kappa) \big) \Big\} \\ &\quad + \sum_i w_{3,i} \Big\{ \big(e^+_{x,i}(\kappa+\lambda|\kappa) + e^-_{x,i}(\kappa+\lambda|\kappa) \big) \Big\} \Bigg\}, \end{aligned} \tag{56}$$

where

$$\begin{aligned} e^+_{u,i}(\kappa+\lambda|\kappa) &- e^-_{u,i}(\kappa+\lambda|\kappa) \\ &= u_i(\kappa+\lambda|\kappa) - u_i(\kappa+\lambda+1|\kappa), \end{aligned} \tag{57}$$

$$\begin{aligned} e^+_{x,i}(\kappa+\lambda|\kappa) &- e^-_{x,i}(\kappa+\lambda|\kappa) \\ &= \frac{x_i(\kappa+\lambda|\kappa)}{l_i} - \frac{x_{i+1}(\kappa+\lambda|\kappa)}{l_{i+1}}, \end{aligned} \tag{58}$$

$$e^+_{u,i}(\kappa+\lambda|\kappa) \ge 0 \ , \ e^-_{u,i}(\kappa+\lambda|\kappa) \ge 0, \tag{59}$$

$$e^+_{x,i}(\kappa+\lambda|\kappa) \ge 0 \ , \ e^-_{x,i}(\kappa+\lambda|\kappa) \ge 0. \tag{60}$$

The MLDS formulation coupled with RHC scheme can be transformed to the canonical form of 0-1 MILP problem with the objective function of (56). As a solver for MILP, we have adopted the Branch-and-Bound (B&B) algorithm. The B&B algorithm alternately executes branching process and bounding process, starting by solving the relaxed problem without integer constraints as follows:

Branching Process: If a 0-1 variable does not meet 0-1 constraints, at the optimal solution in the subproblem, the algorithm constructs two new sub-problems, in which some variables are fixed at zero or one. Then, Linear Programming (LP) method is applied to the subproblem.

Bounding Process: The sub-problem is pruned off from the enumeration tree if at least one of following conditions is met.

(1) The solution is infeasible.

(2) The solution to the sub-problem has a higher cost than best integer solution(s) discovered.

One of the important problems in B&B algorithm is how to choose the branching variable. In this work, if one of the $\delta_{P,i,j}(\kappa)$ variable is chosen as the branching variable, then the remaining $\delta_{P,i,j}(\kappa)$ variables may be specified by referring (26) and (27) automatically. This idea accelerates the B&B algorithm.

5. Numerical Experiments

5.1 Signal Control on Straight Road

In this section, we show some results on the numerical experiments to show the usefulness of our strategy.

First, we consider the straight road which has two signals and is divided into five sections as shown in Fig. 2. 50 vehicles are supposed to wait at the start section at the beginning of the simulation. The signals are controlled by our proposed method with the sampling period of thirty second, and the prediction horizon $N = 2$. Also, CA model is used to simulate the movement of each vehicle. In the CA model, max speed was set to be $v_{max} = 5$ [3].

The obtained signal patterns and distributions of vehicles are listed in Table 1. The P_{c_i} denotes the number of vehicles in each section. The P_{d_i} denotes the signal pattern of each signal (G is green, R is red). From Table 1, we can see that the signal turns red when its upstream section becomes empty. Moreover, the proposed method can generate a reasonable offset (time difference) between signals taking into account the movement of vehicles.

5.2 Signal Control in Intersections

In this section, we consider the signal control for the traffic network as shown in Figs. 10 and 11. This traffic network consists of four intersections where only single-way traffic flow is allowed on each road, but two-way road can be easily modeled by integrating two single-way roads in the way that each single-way road takes the opposite direction. We assume that one vehicle enters from left side of this network in every 5 seconds, and from upper side in every 40 seconds. This implies that the horizontal traffic flow is higher than the vertical traffic flow. We have examined following five methods as follows:

A: No control where traffic signal is changed every 30 seconds,

Table 1 Results of signal control on straight road.

	P_{c_1}	P_{d_1}	P_{c_2}	P_{c_3}	P_{d_2}	P_{c_4}	P_{c_5}
step1	50	G	0	0	R	0	0
step2	43	G	7	0	R	0	0
step3	35	G	15	0	R	0	0
step4	26	G	17	7	G	0	0
step5	18	G	17	15	G	0	0
step6	10	G	16	17	G	7	0
step7	1	G	17	17	G	15	0
step8	0	R	10	16	G	17	7
step9	0	R	1	17	G	17	15
step10	0	R	0	10	G	16	17
step11	0	R	0	1	G	17	17
step12	0	R	0	0	R	10	16
step13	0	R	0	0	R	1	17
step14	0	R	0	0	R	0	10
step15	0	R	0	0	R	0	1
step16	0	R	0	0	R	0	0

B: Conventional method with fixed cycling time, 100 seconds, and minimal length of each signal, 10 seconds. In this method, the splits of green signals of the ways in horizontal direction are set to be

$$\frac{q_{H,i}(\kappa)}{q_{H,i}(\kappa) + q_{V,j}(\kappa)},$$

where $q_{H,i}(\kappa)$ is traffic flow of the district i in horizontal direction, and $q_{V,i}(\kappa)$ is traffic flow of the district j in vertical direction (orthogonally adjoining the district i).

C: Proposed method with prediction horizon $N = 1$ without considering uniformity of traffic density,

D: Proposed method with prediction horizon $N = 4$ without

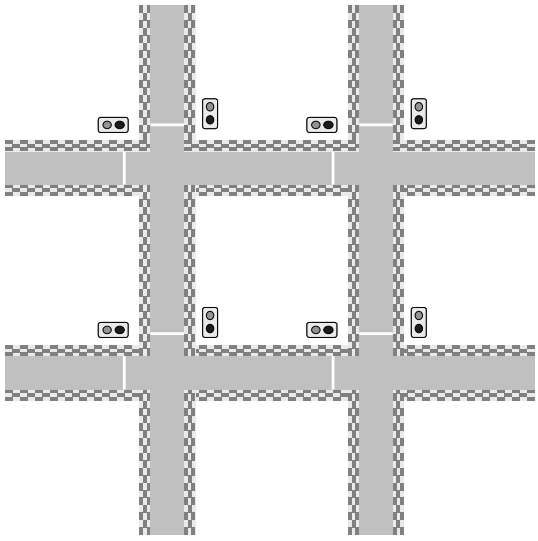

Fig. 10 Traffic network with four single-way intersections.

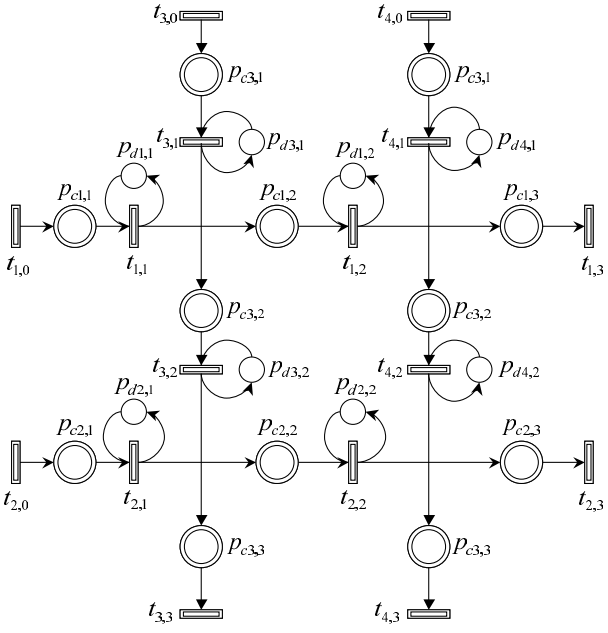

Fig. 11 HPN of traffic network with four single-way intersections.

Table 2 Results of the intersection control.

Method	A	B	C	D	E
Number of passing cars	762	940	960	952	950
Rate of green signal(%)	50	90	90	91.5	92

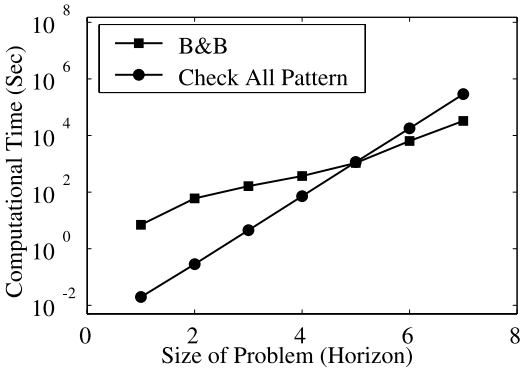

Fig. 12 Computational efforts.

consideration of uniformity of traffic density ($w_{3,i}(i = 1 \sim 8)$ were set to be zero and $w_{1,i} + w_{2,i} = 1$),

E: Proposed method with prediction horizon $N = 4$ considering uniformity of traffic density.

Table 2 shows the results by applying these methods where simulation time is 1000 seconds. From these results, we can see that the results by applying the proposed methods (C,D,E) have better solutions than those by (A,B). In our method, since the cycling time is variable it has higher degree of freedom in planning the signal pattern than conventional methods. This is especially desirable feature when vertical and horizontal traffic flows have significant difference.

Table 2 shows the total number of vehicles which pass through this traffic network in both horizontal and vertical directions. From these results, we can see that the MPC with longer prediction horizon enables more vehicles to get through this traffic network.

Also, evaluations of the computational efforts are shown in Fig. 12.

1. B&B method
2. Full search method

Here, the full search method means to check all patterns of m_D, and other variables are computed by (4). From Fig. 12, we can see that the difference of the computational efforts between two schemes becomes larger with the increase of the horizon.

6. Conclusions

In this paper, we have proposed a new method for traffic signal control based on hybrid dynamical system theory. First of all, the synthetic modeling method for the Traffic Flow Control System (TCCS) has been proposed where the information on geometrical traffic network was modeled by using Hybrid Petri Net (HPN), whereas the information on the behavior of traffic flow was modeled by means of Mixed Logical Dynamical Systems (MLDS) form. The former allows us to easily apply our method to complicated and wide range of traffic network due to its graphical understanding. The latter enables us to optimize the control policy for the traffic signal by means of its algebraic manipulability and use of model predictive control framework. Secondly, the shock wave model has been introduced in order to treat the discontinuity of the traffic flow. By approximating the derived flow model with piece-wise linear function, the flow model has been naturally coupled with the MLDS form. Finally, the model predictive control problem for the TFCS has been formulated. This formulation has been recast to the 0-1 Mixed Integer Linear Programming (MILP) problem. Some numerical experiments have been carried out, and have shown the usefulness of the proposed design framework. Our future works include the development of hierarchical modeling and planning schemes, and analytical consideration of stochastically changing traffic network dynamics.

References

[1] S. Darabha and K.R. Rahagopal, "Intelligent cruise control systems and traffic flow stability," Transp. Res. C, vol.7, pp.329–352, 1999.
[2] H. Hayakawa, "Let's regard traffic flow as powder," Parity, vol.13, no.5, pp.13–22, 1998. (in Japanese)
[3] Y. Kato, Traffic Flow Simulation by Celluar Automaton Method, JSIE, 15-2, 242/250, 2000.
[4] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," Journal de Physique I France, vol.2, p.2221, 1992.
[5] H.J. Payne, "Models of freeway traffic and control," in Mathematical Models of Public Systems, Simulation Council, vol.1, pp.51–61, 1971.
[6] M.J. Lighthill and G.B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," Proc. R. Soc. Lond. A, vol.229, p.281, 1955.
[7] I. Prigogine and R. Herman, Kinetic Theory of Vehicular Traffic, Elsevier, New York, 1971.
[8] M. Cremer and J. Ludwig, "A fast simulation model for traffic flow on the basis of Boolean operations," Math. Comput. Simul., vol.28, pp.297–303, 1986.
[9] F. Balduzzi, A. Giua, and G. Merga, "First-order hybrid Petri nets: A model for optimization and control," IEEE Trans. Robot. Autom., vol.16, no.4, pp.382–399, 2000.
[10] M. Aoyama, N. Uchihira, and K. Hiraishi, PetriNet no riron to zissen, ISCIE, 1995.
[11] Richard Haberman Mathematical Models.
[12] P.P. Chaudahuri, D.R. Chowdhury, S. Nandi, and S. Chattopadhyay, Additive Cellular Automata–Theory and Applications, IEEE Computer Society Press, 1997.
[13] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," Tech. Report, AUT98-04, ETH, Automatica, Special Issue on Hybrid Systems, vol.35, no.3, pp.407–427, 1999.
[14] M. Morari and E. Zafirou, Robust Process Control, Prentice Hall, 1989.
[15] E.F. Camacho and C. Bordons, Model predictive control in the process industry, Springer-Verlag, 1995.

Appendix: Matrices in MLDS form for Fig. 1

The matrices in MLDS form for the straight road illustrated in Fig. 1 are given as follows,

$$\boldsymbol{A} = \boldsymbol{I}, \tag{A·1}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \ddots & \ddots & \vdots \\ 0 & -1 & 1 & \ddots & \ddots & \vdots \\ 0 & -1 & 1 & \ddots & \ddots & \vdots \\ 0 & -1 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 \end{bmatrix}^T, \tag{A·2}$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & \vdots \\ 1 & 0 & 0 & \cdots & \vdots \\ 0 & 1 & 0 & \cdots & \vdots \\ 0 & 1 & 0 & \cdots & \vdots \\ 0 & 1 & 0 & \cdots & \vdots \\ 0 & 0 & 1 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 1 \end{bmatrix}, \tag{A·3}$$

$$\boldsymbol{E1} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}^T, \tag{A·4}$$

$$\boldsymbol{E2} = \begin{bmatrix} -\boldsymbol{\Gamma} & \boldsymbol{0} \\ \boldsymbol{\Lambda} & \boldsymbol{0} \\ -\boldsymbol{\Lambda} & \boldsymbol{0} \\ \boldsymbol{\Gamma} & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{I} & -\boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{I} \end{bmatrix}, \tag{A·5}$$

$$\boldsymbol{E3} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} & \boldsymbol{I} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}^T, \tag{A·6}$$

$$\boldsymbol{E4} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F} & -\boldsymbol{F} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{C} \end{bmatrix}^T, \tag{A·7}$$

$$\boldsymbol{E5} = \begin{bmatrix} 0 \\ 0 \\ h_0 - m \\ h_0 - m \\ h_0 - m \\ h_1 - m \\ \vdots \\ h_5 - m \\ M - h_0 \\ M - h_0 \\ M - h_0 \\ M - h_1 \\ \vdots \\ M - h_5 \\ 0 \\ 0 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{A·8}$$

where

$$\boldsymbol{\Gamma} = \begin{bmatrix} M_0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & M_0 & 0 & \ddots & \ddots & \vdots \\ \vdots & 0 & M_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & M_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & M_6 \end{bmatrix}, \tag{A·9}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} m_0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & m_0 & 0 & \ddots & \ddots & \vdots \\ \vdots & 0 & m_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & m_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & m_6 \end{bmatrix}, \tag{A·10}$$

$$\boldsymbol{F} = \begin{bmatrix} f_0^1 & 0 & \cdots & \cdots & \cdots & 0 \\ f_0^2 & 0 & \ddots & \ddots & \ddots & 0 \\ f_0^3 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & f_1^1 & 0 & \ddots & \ddots & 0 \\ 0 & f_1^2 & 0 & \ddots & \ddots & 0 \\ 0 & f_1^3 & 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & f_5^3 \end{bmatrix}, \tag{A·11}$$

and

$$\boldsymbol{x} = \begin{bmatrix} 0 & x_1 & x_2 & \cdots & x_5 & 0 \end{bmatrix}^T. \tag{A·12}$$

Tatsuya Kato was born in Aichi, Japan, in 1979. He received the B.E. degrees in Electrical Engineering from Aichi Institute of Technology, Japan in 2002 and the M.E degree in Electrical Engineering from Nagoya University, Japan in 2004. Now, he is candidate for the Ph.D. degree in Department of Mechanical Science and Engineering of Nagoya University.

YoungWoo Kim was born in TaeGu city, Korea, in 1973. He received the B.E. degree in Electrical Engineering from YeungNam University, Korea in 1997 and the M.S. and Ph.D. degrees in Electrical Engineering from Nagoya University, Japan in 2000 and 2004. Now, he is in post-doctoral program, working for Space Robotic Research Center of Toyota Technological Institute, Nagoya Japan. His research interests include operational researches, discrete event systems, networked systems, and hybrid dynamical systems.

Tatsuya Suzuki was born in Aichi, Japan, in 1964. He recieved the BE, ME and Ph.D. degrees in Electronic Mechanical Engineering from Nagoya University, Japan in 1986, 1988 and 1991, respectively. From 1998 to 1999, he was a visiting researcher of the Mechanical Engineering Department of U.C.Berkeley. Currently, he is an associate Professor of the Department of Mechanical Science and Engineering of Nagoya University. He won the paper award from IEEJ in 1995. His current research interests are in the areas of hybrid dynamical systems and intelligent systems. Dr.Suzuki is a member of the IEEJ, SICE, ISCIE, RSJ, JSME and IEEE.

Shigeru Okuma was born in Gifu, Japan, in 1948. He received the B.E., M.E. and Ph.D. degrees in Electronic-Mechanical Engineering from Nagoya University, Japan in 1970, 1972 and 1978, respectively. He also received the M.E. degree in systems engineering from Case Western Reserve University, Cleveland, OH, in 1974. Since 1990, he has been a Professor in the Department of Electrical Engineering and Computer Science, Nagoya University. His research interests are power electronics, robotics and evolutionary soft computing. Dr. Okuma was the recipient of the IEEE IECON'92 Best Paper Award, in addition to paper awards from the Japan Society for Precision Engineering and Institute of Electrical Engineering of Japan.