28

# Traffic Network Control Based on Hybrid System Modeling

Youngwoo Kim
*Nagoya University, Nagoya*
*Japan*

## 1. Introduction

With the increasing number of automobiles 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) (2)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 (11) is widely known idea 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. 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 and its traffic light controller design. This is because they require 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) (9) 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 are 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 literature (6) (5) (7) (8). Although these approaches included judicious use of theoretical ideas 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 network control based on an integrated Hybrid Dynamical System (HDS) framework. The proposed method characterizes its synthetic modeling description. The information on geometrical traffic network is modeled by using Hybrid Petri Net (HPN), whereas the information on the behavior of traffic flow is modeled by means of Mixed Logical Dynamical Systems (MLDS) description. The former allows us to easily apply our method to complicated and wide range of traffic network due to its graphical understanding and algebraic manipulability. The latter allows us to represent physical features governing the dynamics of traffic flow and control mechanism for traffic congestion control employing the model predictive control policy (13).

Note that current traffic flow away from the signaler affects future traffic flow behavior. Through the model predictive control policy, we can construct the decentralized controller in a manner that each traffic outflow from the intersection or crosswalk is controlled and the information is shared with neighboring traffic controllers. A large-scale centralized traffic network controller is not appropriate because of the increased computational effort, synchronization in information processes and so on. In this case, the decentralized controller with model predictive control policy could be a realistic method.

In order to control large-scale traffic network with nonlinear dynamics, we formulate the traffic network control system based on the Mixed Integer NonLinear Programming (MINLP) problem. Generally, it is difficult to find the global optimal solution to the nonlinear programming problem. However, if the problem can be recast to the convex programming problem, the global optimal solution is easily found by applying an efficient method such as Steepest Descent Method (SDM). We use in this paper general performance criteria for traffic network control and show that although the problem contains non-convex constraint functions as a whole, the generated sub-problems are always included in the class of convex programming problem. In order to achieve high control performance of the traffic network with dynamically changing traffic flow, we adopt Model Predictive Control (MPC) policy. Note that MLDS formulation often encounters multiplication of two decision variables, and that without modification, it cannot be directly applied to MPC scheme. One way to avoid the multiplication is to introduce a new auxiliary variable to represent it. And then it becomes a linear system formally. However, as we described before, the introduction of discrete variables causes substantial computational amounts. A new method for this type of control problem is proposed. Although the system representation is nonlinear, MPC policy is successfully applied by means of the proposed Branch and Bound strategy.

After verification of the solution optimality, PWARX classifier is applied which describes a nonlinear feedback control law of the traffic control system. This implies we don't need a time-consuming searching process of a solver such as a Branch-and-Bound algorithm to solve

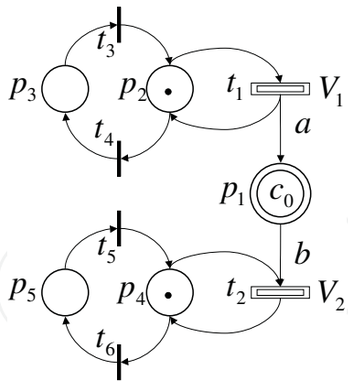


Fig. 1. Example of Hybrid Petri Net model

a mixed integer nonlinear programming (MINLP) problem, and furthermore the exactly same solutions are obtained in a very short time.

The problem we address in this paper is a special classification problem where the output $y$ is a 0-1 binary variable, and very good classification performance is desirable even with very large number of the introduced clusters. If we plot the observational data in a same cluster in the $x$-$y$ space, it will show always zero inclination, since we have a binary output, i.e., all components of $\theta$, $a$ and $b$ except for $f$ will be zeros. This implies we need consideration for a binary output. A new performance criterion is presented in this paper to consider not only a covariance of $\theta$, but also a covariance of $y$. The proposed method is a hierarchical classification procedure, where the cluster splitting process is introduced to the cluster with the worst classification performance (which includes 0-1 mixed values of $y$). The cluster splitting process is follows by the piecewise fitting process to compute the cluster guard and dynamics, and the cluster updating process to find new center points of the clusters. The usefulness of the proposed method is verified through some numerical experiments.

## 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.

### 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. HPN is a structure of $N = (P, T, I_+, I_-, C, D)$. The set of places $P$ is partitioned into a subset of discrete places $P_d$ and a subset of continuous places $P_c$. The set of transition $T$ is partitioned into a subset of discrete transitions $T_d$ and a subset of continuous transitions $T_c$. The incidence matrix of the net is defined as $I(p,t) = I_-(p,t) - I_+(p,t)$, where $I_+(p,t)$ and $I_-(p,t)$ are the forward and backward incidence relationships between the transition $t$ and the place $p$ which follows and precedes the transition. We denote the preset (postset) of transition t as $^\bullet t$ ($t^\bullet$) and its restriction to continuous or discrete places as $^{(d)}t =$

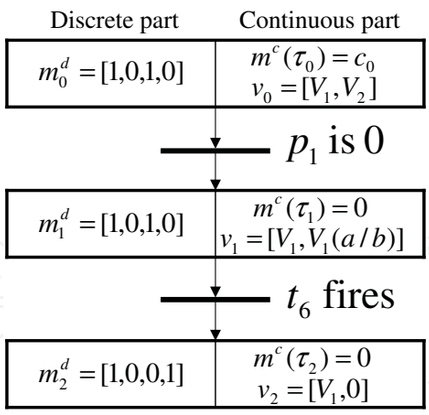


Fig. 2. Phase diagram of Hybrid Petri Net model

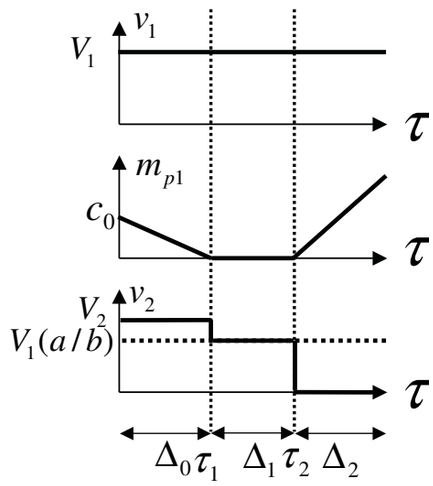


Fig. 3. Behavior of Hybrid Petri Net model

$^\bullet t \cap P_d$ or $^{(c)}\ t = {}^\bullet t \cap P_c$. Similar notation may be used for presets and postsets of places. The function $C$ and $D$ specify the firing speeds associated to the continuous transitions and the timing associated to the (timed) discrete transitions. For any continuous transition $t_i$, we let $C(t_i) = (v_i, V_i)$, where $v_i$ and $V_i$ represent the minimum and maximum firing speed of transition $t_i$. We associate to the timed discrete transition its firing delay, where the firing delay is short enough and the state is preserved until next sampling instant. The acquisition of firing sequence of the discrete transition at every sampling instant is applied to a variety of scheduling and control problems. The marking $M = [M_C|M_D]$ has both continuous ($m$ dimension) and discrete ($n$ dimension) parts.

Consider a simple example of First-Order Hybrid Petri Net model, Fig.1, where the control switch is represented with two discrete transitions and two discrete places connected to the continuous transition. In Fig.1, $p_1$ is the continuous place with the initial marking $m_c(\tau_0) = m_{p1} = c_0$, and $p_2$, $p_3$, $p_4$ and $p_5$ are the discrete places with the initial marking $m_d(\tau_0) = [m_{p2}, m_{p3}, m_{p4}, m_{p5}] = [1, 0, 1, 0]$. We assume $V_1 a < V_2 b$, where $V_1$ and $V_2$ are firing speed of $t_1$ and

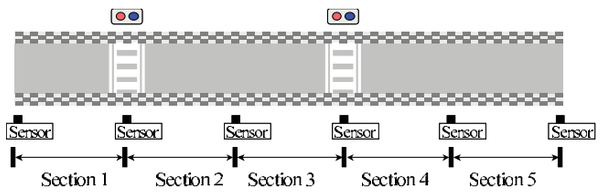


Fig. 4. Traffic network

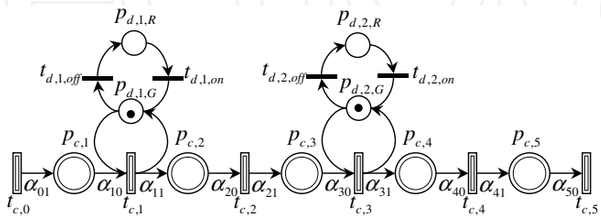


Fig. 5. Hybrid Petri Net model of traffic network

$t_2$ respectively, $a$ and $b$ are the arc weights given by the incidence relationship. The behavior is illustrated in Fig.2 and Fig.3.

Figure 5 shows the HPN model for the road of Fig.4. In Fig.5, each section $i$ of $l_i$-meters long constitutes the straight road, and two traffic lights are installed at the point of crosswalk. $p_c \in P_c$ represents each section of the road, and has maximum capacity (maximum number of vehicles). Also, $p_d \in P_d$ represents the traffic signal where green signal is indicated by an existence of a token. 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. $q_j(\tau)$ is the firing speeds assigned to transition $t_j \in T$ at time $\tau$. $q_j(\tau)$ represents the number of vehicles passing through the boundary per time unit of two successive sections(measuring position) at time $\tau$. The sensors to capture the number of the vehicles are supposed to be installed at every boundary of the section as show in Fig.4. The element of $I(p,t)$ is always 0 or $\alpha_{ij}$. $\alpha_{ij}$ 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:

if $p_{d,k} = {}^\bullet t_j$ is not null,

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

otherwise,

$$\frac{dm_{C,i}(\tau)}{dt} = \sum_{t_j \in p_{c_i}{}^\bullet \cup {}^\bullet p_{c_i}} I(p_{c_i}, t_j) \cdot q_j(\tau), \tag{4}$$

where $m_{C,i}(\tau)$ is the marking for the place $p_{c_i} (\in P_c)$ at time $\tau$, and $m_{D,k}(\tau)$ is the marking for the place $p_{d_k} (\in P_d)$. The equation (3) is transformed to its discrete-time version supposing

that $q_j(\tau)$ is constant during two successive sampling instants as follows:

$$m_{C,i}((\kappa+1)T_s) = m_{C,i}(\kappa T_s) + \sum_{t_j \in p_{c_i}{}^{\bullet} \cup {}^{\bullet} p_{c_i}} I(p_{c_i}, t_j) \cdot q_j(\kappa T_s) \cdot m_{D,j}(\kappa T_s) \cdot T_s. \quad (5)$$

where $\kappa$ is sampling index, and $T_s$ is sampling period.
Note that the transition $t$ is *enabled* at the sampling instant $\kappa T_s$ 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

$$q_i(\tau) = \frac{(k_i(\tau) + k_{i+1}(\tau))}{2} \frac{v_i(\tau) + v_{i+1}(\tau)}{2} \quad (6)$$

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

$$v_i(\tau) = v_{f_i} \cdot \left(1 - \frac{k_i(\tau)}{k_{jam}}\right), \quad (7)$$

with (6), 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 exists 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'(10) is introduced as follows. We consider the case as shown in Fig.6 where the traffic density of $i$th 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 $i$th and $(i+1)$th sections as follows:

$$c_i(\tau) = \frac{v_i(\tau)k_i(\tau) - v_{i+1}(\tau)k_{i+1}(\tau)}{k_i(\tau) - k_{i+1}(\tau)}. \quad (8)$$

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

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

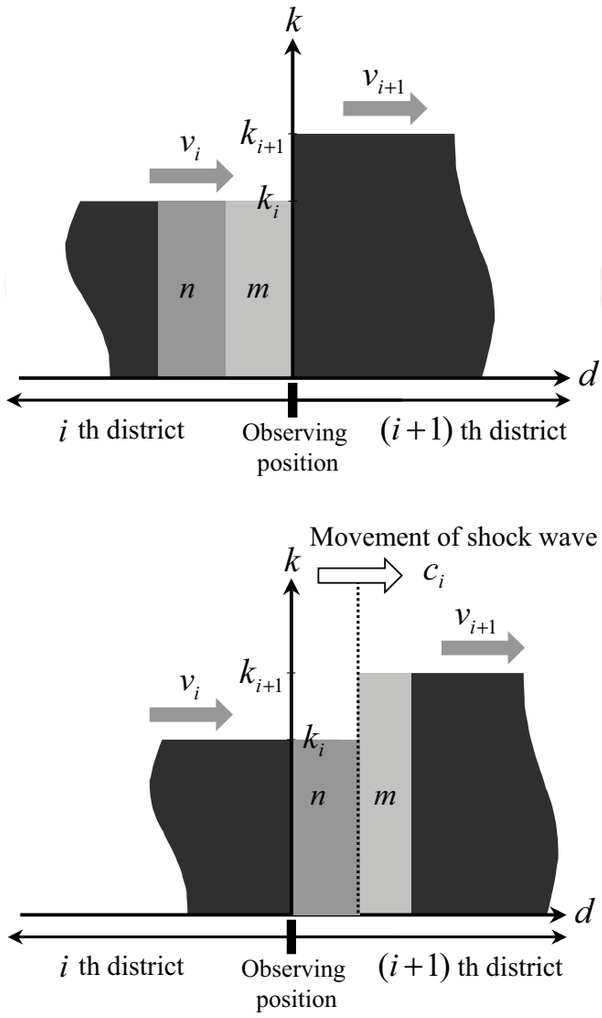


Fig. 6. Movement of shock wave in the case of $k_i(\tau) < k_{i+1}(\tau)$ and $c_i(\tau) > 0$

(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.6 ($c_i(\tau) > 0$) and Fig.7 ($c_i(\tau) \leq 0$). In Fig.6 and Fig.7, the 'measuring position' implies the position where transition $t_i$ is assigned. Since the traffic flow $q_i(\tau)$ represents the numbers of vehicles passing through the measuring position per unit time, in the case of (C1), it can be represented by $n + m$ in Fig.6, 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.7.
These considerations lead to the following models:

in the case of (C1)

$$q_i(\tau) = v_i(\tau)k_i(\tau) \tag{9}$$

$$= v_{f_i}\left(1 - \frac{k_i(\tau)}{k_{jam}}\right)k_i(\tau), \tag{10}$$

in the case of (C2)

$$q_i(\tau) = v_{i+1}(\tau)k_{i+1}(\tau) \tag{11}$$

$$= v_{f_{i+1}}\left(1 - \frac{k_{i+1}(\tau)}{k_{jam}}\right)k_{i+1}(\tau). \tag{12}$$

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),

$$q_i(\tau) = \left(\frac{k_i(\tau) + k_{i+1}(\tau)}{2}\right)v_f(\tau)\left(1 - \frac{k_i(\tau) + k_{i+1}(\tau)}{2k_{jam}}\right). \tag{13}$$

As the results, the flow model (9) $\sim$ (13) taking into account the discontinuity of the density can be summarized as follows:

$$q_i(\tau) = \begin{cases} \left(\frac{k_i(\tau)+k_{i+1}(\tau)}{2}\right)v_f\left(1 - \frac{k_i(\tau)+k_{i+1}(\tau)}{2k_{jam}}\right) \\ \quad if\ k_i(\tau) \geq k_{i+1}(\tau) \\ v_{f_i}\left(1 - \frac{k_i(\tau)}{k_{jam}}\right)k_i(\tau) \\ \quad if\ k_i(\tau) < k_{i+1}(\tau) \text{ and } c(\tau) > 0 \\ v_{f_{i+1}}\left(1 - \frac{k_{i+1}(\tau)}{k_{jam}}\right)k_{i+1}(\tau) \\ \quad if\ k_i(\tau) < k_{i+1}(\tau) \text{ and } c(\tau) \leq 0 \end{cases}. \tag{14}$$

Figure 8 shows the HPN model of the $i$th intersection, where the notation for other than southwardly entrance lane is omitted. In Fig.8, $l_{j,E}$, $l_{j,W}$, $l_{j,S}$ and $l_{j,N}$ are the length of the corresponding districts, and the numbers of the vehicles in the districts are obtained as for example $p_{c,j_{IS}}(\tau) = k_{j_{IS}}(\tau) \cdot l_{j,S}$. The vehicles in $p_{c,j_{IS}}$ are assumed to have the probability $\zeta_{j,SW}$, $\zeta_{j,SN}$, and $\zeta_{j,SE}$ to proceed into the district corresponding to $p_{c,j_{OW}}$, $p_{c,j_{ON}}$, and $p_{c,j_{OE}}$ as follows,

$$k_{j_{SW}}(\tau) = k_{j_{IS}}(\tau)\zeta_{j,SW}(\tau), \tag{15}$$

$$k_{j_{SN}}(\tau) = k_{j_{IS}}(\tau)\zeta_{j,SN}(\tau), \tag{16}$$

$$k_{j_{SE}}(\tau) = k_{j_{IS}}(\tau)\zeta_{j,SE}(\tau). \tag{17}$$

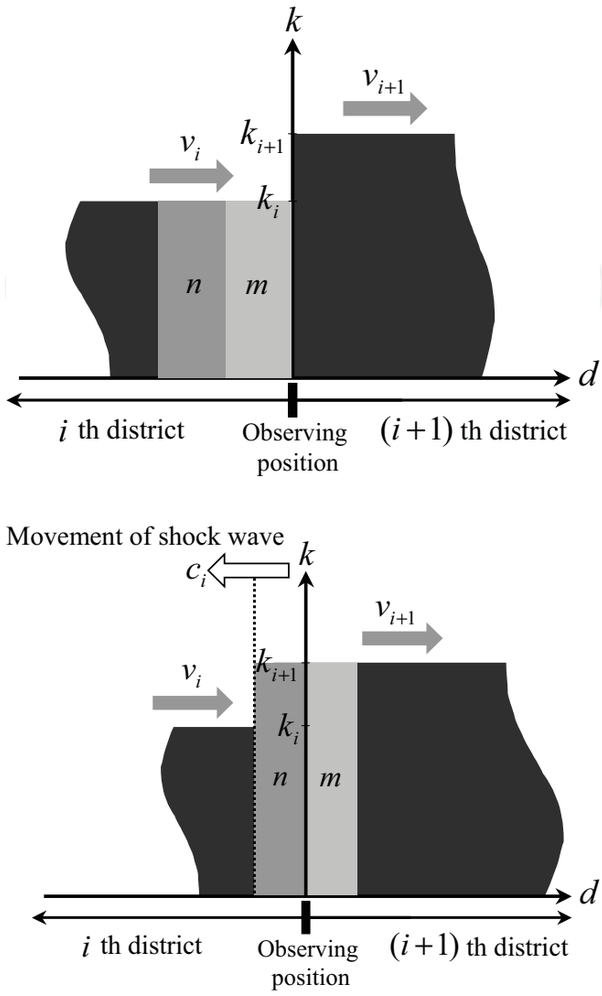


Fig. 7. Movement of shock wave in the case of $k_i(\tau) < k_{i+1}(\tau)$ and $c_i(\tau) \leq 0$

Note that these probabilities are determined by the traffic network structure, and satisfy at $\tau$,

$$0 \leq \zeta_{j,SW}(\tau) \leq 1, \tag{18}$$
$$0 \leq \zeta_{j,SN}(\tau) \leq 1, \tag{19}$$
$$0 \leq \zeta_{j,SE}(\tau) \leq 1, \tag{20}$$
$$\zeta_{j,SW}(\tau) + \zeta_{j,SN}(\tau) + \zeta_{j,SE}(\tau) = 1. \tag{21}$$

Therefore, the traffic flows of the three directions are represented with

$$q\left(k_{j_{SN}}(\tau), k_{j_{ON}}(\tau)\right), \tag{22}$$
$$q\left(k_{j_{SW}}(\tau), k_{j_{OW}}(\tau)\right), \tag{23}$$
$$q\left(k_{j_{SE}}(\tau), k_{j_{OS}}(\tau)\right). \tag{24}$$

Note that the mutual exclusion of the same traffic light with the intersecting road is represented in the Fig.8.

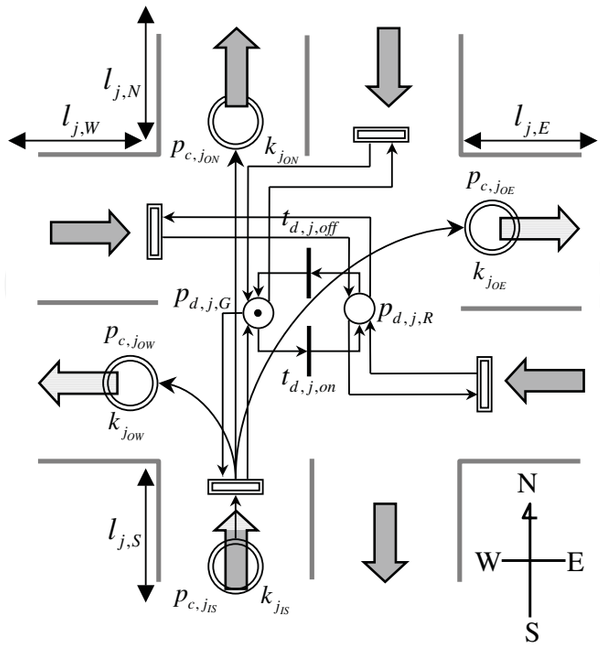


Fig. 8. Hybrid Petri Net model of intersection

### 2.3 Derived flow model

In this subsection, we confirm 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

$$n_j(\tau+1) = n_j^{in}(\tau)(1-n_j(\tau)) - n_j^{out}(\tau), \tag{25}$$

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

Step 1, 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:

$$v_{\langle j\rangle}(\tau+\Delta\tau) \equiv v_{\langle j\rangle}(\tau) + v_{unit}. \tag{26}$$

Step 2, 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:

$$v_{\langle j \rangle}(\tau + \Delta\tau) \equiv \min\{e, v_{\langle j \rangle}(\tau + \Delta\tau)\}. \tag{27}$$

Step 3, Randomization rule: With probability $p$, the velocity is reduced by one unit velocity as follows:

$$v_{\langle j \rangle}(\tau + \Delta\tau) \equiv v_{\langle j \rangle}(\tau + \Delta\tau) - p \cdot v_{unit}. \tag{28}$$

Figure 9 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] $\cdot$ 5 [cells/$\Delta\tau$] $\cdot$ 3600[sec]/1000). The left figure of Fig.9 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 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 $i$th and $(i+1)$th section. Then, $q_i(\tau)$ is proportionally increased as $k_i(\tau)$ increases, and reaches the saturation point ($k_i(\tau) = 0.9$). Next, we look at the behavior along the edge $b$ which implies that the $i$th 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.10 shows the behavior in case of using HPN where the proposed flow model given by (14) is embedded. We can see that Fig.10 shows the similar characteristics to Fig.9, 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 (13) ) for all cases. Figure 11 shows the behavior in case of using HPN where the flow model is supposed to be given by (13) 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 case of using CA, it took 140 seconds to construct the traffic flow dynamics using Athlon XP 2400 and Windows 2000, while only 0.06 seconds in case of using HPN and (14).

## 3. Model Predictive Control of Traffic Network Control based on MLDS description

The Receding Horizon Control (RHC) or Model Predictive Control (MPC) is one of well - known paradigms for optimizing the systems with constraints and uncertainties. In RHC paradigms, the solutions are elements of finite dimensional vector spaces, and finite-horizon optimization is carried out in order to provide stability or performance analysis. However, the application of RHC has been mainly restricted to the system with sufficiently long sampling interval, since finite-horizon optimization is computationally demanding.
This chapter firstly formulate the traffic flow model developed in chapter 2 in the form of MLDS description coupled with RHC strategy, where wide range of traffic flow is considered.

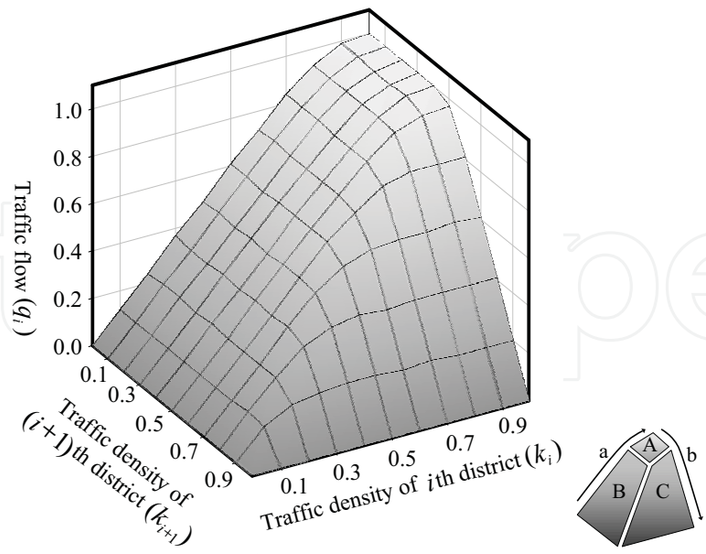


Fig. 9. Traffic flow behavior obtained from CA model

This formulation is recast to the canonical form of 0-1 Mixed Integer Linear Programming (MILP) problem to optimize its behavior and a new Branch and Bound (B&B) based algorithm is presented in order to abate computational cost of MILP problem.

### 3.1 MLDS representation of TCCS based on Piece-Wise Affine (PWA) linearization of traffic flow

Since TCCS is the hybrid dynamical system including both continuous traffic flow dynamics and discrete aspects for traffic light signal control, some algebraic formulation, which handles both continuous and discrete behaviors, must be introduced. The MLDS description has been developed to describe such class of systems considering some constraints shown in the form of inequalities and can be combined with powerful search engine such as Mixed Integer Linear Programming (MILP).
The MLDS (12) description can be formalized as following.

$$\begin{aligned} \boldsymbol{x}(\tau+1) &= \boldsymbol{A_\tau x}(\tau) + \boldsymbol{B_{1\tau}u}(\tau) \\ &\quad + \boldsymbol{B_{2\tau}}\delta(\tau) + \boldsymbol{B_{3\tau}z}(\tau) \qquad (29) \\ \boldsymbol{y}(\tau) &= \boldsymbol{C_\tau x}(\tau) + \boldsymbol{D_{1\tau}u}(\tau) \\ &\quad + \boldsymbol{D_{2\tau}}\delta(t) + \boldsymbol{D_{3\tau}}(\tau) \qquad (30) \\ \boldsymbol{E_{2\tau}}\delta(\tau) + & \boldsymbol{E_{3\tau}z}(\tau) \leq \\ & \boldsymbol{E_{1\tau}u}(\tau) + \boldsymbol{E_{4\tau}x}(\tau) + \boldsymbol{E_{5\tau}} \qquad (31) \end{aligned}$$

In MLDS formulation, (29), (30) and (31) are state equation, output equation and constraint inequality, respectively, where $x$, $y$ and $u$ are the state, output and input variable, whose components are constituted by continuous and/or 0-1 binary variables, $\delta(\tau) \in \{0, 1\}$ and $z(\tau) \in \Re$

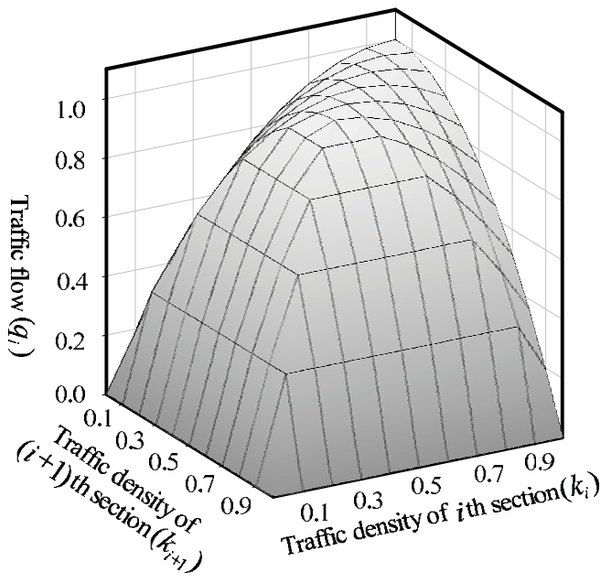


Fig. 10. Traffic flow behavior obtained from the proposed traffic flow model

represent auxiliary logical and continuous variables. By introducing the constraint inequality of (31), non-linear constraints as (14) can be transformed to the computationally tractable Piece-Wise Affine (PWA) forms.
The traffic flow of Fig. 9 can be approximated as the right figure of Fig. 9 which consists of three planes as follows,

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

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

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

where $a$ is the threshold value to describe saturation characteristic of traffic flow that if $k_i(\tau) > a$ and/or $k_{i+1}(\tau) < k_{jam} - a$, the value of $q_i(\tau)$ hovers at its maximum value $q_{max}$.

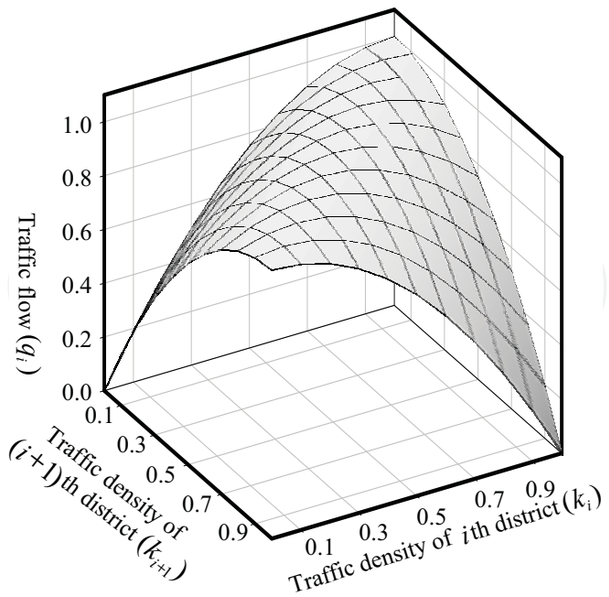


Fig. 11. Traffic flow behavior obtained by averaging $k_i$ and $k_{i+1}$

Fig.12 shows three planes partitioned by introducing three auxiliary variables $\delta_{P,i,1}(\tau)$, $\delta_{P,i,2}(\tau)$ and $\delta_{P,i,3}(\tau)$ which are defined as follows,

$$[\delta_{P,i,1}(\tau) = 1] \leftrightarrow \begin{cases} k_i(\tau) & \geq & a \\ k_{i+1}(\tau) & \leq & k_{jam} - a \end{cases} \tag{32}$$

$$[\delta_{P,i,2}(\tau) = 1] \leftrightarrow \begin{cases} k_i(\tau) & \leq & a - \varepsilon \\ k_i(\tau) + k_{i+1}(\tau) & \leq & k_{jam} \end{cases} \tag{33}$$

$$[\delta_{P,i,3}(\tau) = 1] \leftrightarrow \begin{cases} k_{i+1}(\tau) & \geq & k_{jam} - a + \varepsilon \\ k_i(\tau) + k_{i+1}(\tau) & \geq & k_{jam} + \varepsilon \end{cases} \tag{34}$$

$$\delta_{P,i,1}(\tau) + \delta_{P,i,2}(\tau) + \delta_{P,i,3}(\tau) = 1 \tag{35}$$

where $\varepsilon$ is small tolerance to consider equality sign.
Therefore, the traffic flow $q_i(\tau)$ can be rewritten in a compact form as follows

$$\begin{aligned} q_i(\tau) &= q_{max}\delta_{P,i,1}(\tau) + \frac{q_{max}k_i(\tau)}{a}\delta_{P,i,2}(\tau) \\ &\quad + \frac{q_{max}(1 - k_{i+1}(\tau))}{a}\delta_{P,i,3}(\tau) \end{aligned} \tag{36}$$

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

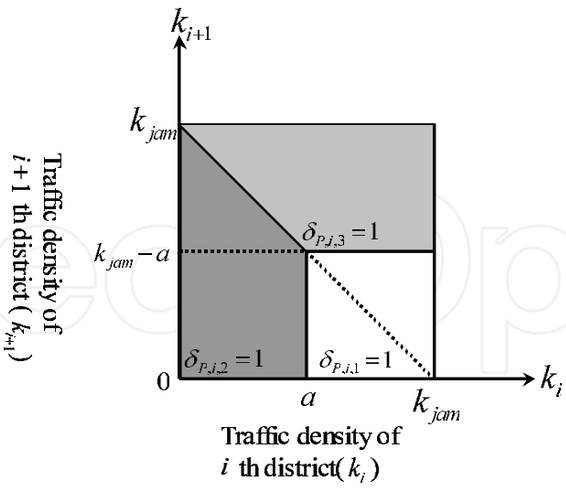


Fig. 12. Assignation of planes by introducing auxiliary variables

where $0 \leq k_i(\tau) \leq k_{jam}$, $0 \leq k_{i+1} \leq k_{jam}$ $(= 1)$, $q_{max}$ is the maximum value of traffic flow. Figure xxx shows the piece-wise affine (PWA) dynamics of the traffic flow model developed in the previous chapter where $a = 0.3$ and $q_{max} = 1$.
The equations (32) to (34) can be generalized as (37) and (38), and transformed to inequality as (39) The equations (32) and (34) can be generalized as (37) and (38), and transformed to inequality as (39)

$$[\delta_{P,i,j}(\tau) = 1] \quad \leftrightarrow \quad \left[ \begin{bmatrix} k_i \\ k_{i+1} \end{bmatrix} \in \ell_j \right] \tag{37}$$

$$\ell_j \quad = \quad \left\{ \begin{bmatrix} k_i \\ k_{i+1} \end{bmatrix} : \boldsymbol{S_j k_i}(\tau) \leq \boldsymbol{T_j} \right\} \tag{38}$$

where $\boldsymbol{k_i}(\tau) = [k_i(\tau) k_{i+1}(\tau)]^T$ and $\boldsymbol{S_j}$ and $\boldsymbol{T_j}$ are the matrices with suitable dimensions which satisfy

$$\boldsymbol{S_j k_i}(\tau) \quad - \quad \boldsymbol{T_j} \leq \boldsymbol{M_j}^* [1 - \delta_{P,i,j}(\tau)], \tag{39}$$

$$\boldsymbol{M_j^*} \quad \stackrel{\triangle}{=} \quad \max_{\boldsymbol{k_i} \in \ell_j} \boldsymbol{S_j k_i}(\boldsymbol{\tau}) - \boldsymbol{T_j}. \tag{40}$$

The traffic flow $q_i(\tau)$ of (37) is the relationship between $\boldsymbol{k_i}(\tau)$ and $\boldsymbol{\delta_{P,i}}(\tau) = [\delta_{P,i,1}(\tau)\ \delta_{P,i,2}(\tau)\ \delta_{P,i,3}(\tau)]$ which can be rewritten as follows,

$$q_i(\tau) \quad = \quad f(\boldsymbol{\delta_{P,i}}(\tau), \boldsymbol{k_{i+1}}(\tau)) \tag{41}$$

$$\quad = \quad \sum_{j=1}^{3} (\boldsymbol{F_i^j}(\tau) \boldsymbol{k_i}(\tau) + \boldsymbol{H_i^j}) \delta_{P,i,j}(\tau) \tag{42}$$

where $\boldsymbol{\delta_{P,i}} = [\delta_{P,i,1}, \delta_{P,i,2}, \delta_{P,i,3}]'$. In these equations, each pair of $\boldsymbol{F_i^j}$ and $\boldsymbol{H_i^j}$ represents the corresponding domain of Fig. 12 as follows,

$$\begin{aligned}
\boldsymbol{F_i^1} &= [\ 0 \quad 0\ ] && (43)\\
\boldsymbol{H_i^1} &= q_{max} && (44)\\
\boldsymbol{F_i^2} &= [\ \tfrac{q_{max}}{a} \quad 0\ ] && (45)\\
\boldsymbol{H_i^2} &= 0 && (46)\\
\boldsymbol{F_i^3} &= [\ 0 \quad -\tfrac{q_{max}}{a}\ ] && (47)\\
\boldsymbol{H_i^3} &= \frac{q_{max}}{a} && (48)
\end{aligned}$$

The traffic flow $\boldsymbol{z_i}(\tau) = [z_{i,1}(\tau)\ z_{i,2}(\tau)\ z_{i,3}(\tau)]$ in consideration of the binary input $u_i(\tau) \in \{0,1\}$ for traffic light control can be represented by

$$\begin{aligned}
z_{i,j}(\tau) &\leq M_i u_i(\tau)\delta_{P,i,j}(\tau), && (49)\\
z_{i,j}(\tau) &\geq m_i u_i(\tau)\delta_{P,i,j}(\tau), && (50)\\
z_{i,j}(\tau) &\leq \boldsymbol{F_i^j k_i}(\tau) + \boldsymbol{H_i^j} \\
&\quad -m_i(1 - u_i(\tau)\delta_{P,i,j}(\tau)), && (51)\\
z_{i,j}(\tau) &\geq \boldsymbol{F_i^j k_i}(\tau) + \boldsymbol{H_i^j} \\
&\quad -M_i(1 - u_i(\tau)\delta_{P,i,j}(\tau)). && (52)
\end{aligned}$$

where $M_i$ and $m_i$ are respectively

$$\begin{aligned}
M_i &= \max_{\boldsymbol{k_i}(\tau)\in\ell_j} \left\{ \boldsymbol{F_i^j k_i}(\tau) + \boldsymbol{H_j} \right\}, && (53)\\
m_i &= \min_{\boldsymbol{k_i}(\tau)\in\ell_j} \left\{ \boldsymbol{F_i^j k_i}(\tau) + \boldsymbol{H_j} \right\}. && (54)
\end{aligned}$$

The product $u_i(\tau)\ \delta_{P,i,j}(\tau)$ can be replaced by an auxiliary logical variable $\delta_{M,i,j}(\tau) = u_i(\tau)\ \delta_{P,i,j}(\tau)$ in order to make it tractable to deal with MILP problem. Then this relationship can be equivalently represented as follows,

$$\begin{aligned}
-u_i(\tau) + \delta_{M,i,j}(\tau) &\leq 0, && (55)\\
-\delta_{P,i,j}(\tau) + \delta_{M,i,j}(\tau) &\leq 0, && (56)\\
u_i(\tau) + \delta_{P,i,j}(\tau) + \delta_{M,i,j}(\tau) &\leq 1. && (57)
\end{aligned}$$

Therefore, the MLDS description for the proposed system can be formalized as follows,

$$\begin{aligned}
\boldsymbol{x}(\kappa+1) &= \boldsymbol{A}\boldsymbol{x}(\kappa) + \boldsymbol{B}\boldsymbol{z}(\kappa), && (58)\\
\boldsymbol{z}(\kappa) &= \boldsymbol{C_1} diag(\boldsymbol{u}(\kappa))\boldsymbol{C_2}\boldsymbol{\delta}(\kappa), && (59)\\
\boldsymbol{E_2}\boldsymbol{\delta}(\kappa) &+ \boldsymbol{E_3}\boldsymbol{z}(\kappa) \\
&\leq \boldsymbol{E_1}\boldsymbol{u}(\kappa) + \boldsymbol{E_4}\boldsymbol{x}(\kappa) + \boldsymbol{E_5} && (60)
\end{aligned}$$

where the element $x_i(\kappa)$ of $\boldsymbol{x}(\kappa) \in \Re^{|P|}$, is marking of the place $p_{c_i}$ at the sampling instance $\kappa$, the element $u_i(\kappa)(\in \{0,1\})$ of $\boldsymbol{u}(\kappa) \in Z^{|T|}$, is the signal of traffic light installed at $i$th district

and $\delta(\kappa)$=$[\delta_{\boldsymbol{P}}(\kappa), \delta_{\boldsymbol{M}}(\kappa)]^{'}$. Note that if there is no traffic light installed at $i$th district, $u_i(\kappa)$ is always set to 1. And $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C_1}$, $\boldsymbol{C_2}$, $\boldsymbol{E_1}$, $\boldsymbol{E_2}$, $\boldsymbol{E_3}$, $\boldsymbol{E_4}$ and $\boldsymbol{E_5}$ are the matrices with appropriate dimensions.

### 3.2 Model predictive control policy for traffic network control

The traffic system is large-scale dynamical system with uncertainty in the behavior of each car. In order to develop efficient traffic light control system, a wide range of traffic flow should be fully considered. In this subchapter, model predictive control policy for traffic light control is applied to the traffic flow model developed in the previous chapter. In RHC scheme, an input for next sampling period is decided based on the prediction for next several periods called the prediction horizon. This allows for the fact that the spatially changing dynamics of traffic flow are represented by temporal behavior over prediction horizon, since traffic flow can be considered as probabilistic time-series behavior.
The equation (58) can be modified, enumerating the state and input variables for the future periods as follows,

$$\begin{aligned} \boldsymbol{x}(\kappa+\lambda|\kappa) &= \boldsymbol{A^{\kappa}}x(\kappa) \\ &\quad + \sum_{\eta=0}^{\lambda-1}\{\boldsymbol{A^{\eta}}(\boldsymbol{BC_1}(diag(\boldsymbol{u}(\kappa+\lambda-1-\eta|\kappa))) \\ &\qquad \cdot \boldsymbol{C_2}\boldsymbol{\delta}(\kappa+\lambda-1-\eta|\kappa))\} \end{aligned} \tag{61}$$

where $x(\kappa+\lambda|\kappa)$ denotes the predicted state vector at time $\kappa+\lambda$, obtained by applying the input sequence $u(\lambda|\kappa)=u(\kappa),\cdots,u(\kappa+\lambda)$ to (58) starting from the state $x(\lambda|\kappa)=x(\kappa)$.
Now we consider following requirements that usually appear in the traffic light control problems.

(R1) Maximizes traffic flow over entire traffic network.

(R2) Avoid frequent change of traffic signal.

(R3) Avoid concentration of traffic flow in a certain district.

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

$$\begin{aligned} & J(\boldsymbol{u}(\lambda|\kappa),\cdots,\boldsymbol{u}(\lambda+N_I|\kappa) \\ & \quad ,\boldsymbol{x}(\lambda|\kappa),\cdots,\boldsymbol{x}(\lambda+N_I|\kappa) \\ & \quad ,\boldsymbol{\delta}(\lambda|\kappa),\cdots,\boldsymbol{\delta}(\lambda+N_I|\kappa)) \\ &= \sum_{\lambda=1}^{N}\Bigg\{-\sum_i w_{1,i}\Bigg\{\Bigg(\boldsymbol{\Theta_i}\begin{bmatrix} x_i(\lambda|\kappa)/l_i \\ x_{i+1}(\lambda|\kappa)/l_{i+1}\end{bmatrix} \\ & \quad +\boldsymbol{\Phi}\Bigg)^{'}\boldsymbol{\delta_{M,i}}(\lambda|\kappa)\Bigg\} \\ & \quad -\sum_i w_{2,i}\Big\{1-\big|u_i(\lambda|\kappa)-u_i(\lambda+1|\kappa)\big|\Big\} \\ & \quad +\sum_i w_{3,i}\Big\{\Big|\frac{x_i(\lambda|\kappa)}{l_i}-\frac{x_{i+1}(\lambda|\kappa)}{l_{i+1}}\Big|\Big\}\Bigg\} \end{aligned} \tag{62}$$

where

$$\boldsymbol{\Theta_i} = \begin{bmatrix} 0 & 0 \\ \frac{q_{max}}{a} & 0 \\ 0 & -\frac{q_{max}}{a} \end{bmatrix} \tag{63}$$

$$\boldsymbol{\Phi_i} = \begin{bmatrix} q_{max} \\ 0 \\ \frac{q_{max} k_i(\tau)}{a} \end{bmatrix} \tag{64}$$

and $w_{1,i}$, $w_{2,i}$ and $w_{3,i}$ are positive weight values for $i$th district which satisfy $w_{1,i} + w_{2,i} + w_{3,i} = 1$, and $0 \leq w_{1,i} \leq 1$, $0 \leq w_{2,i} \leq 1$ and $0 \leq w_{3,i} \leq 1$. In (62), the three terms correspond to the requirement (R1), (R2) and (R3) in order.
Therefore, the optimization problem can be formulated as follows:

$$\begin{gathered} find\ \boldsymbol{\delta}(\boldsymbol{\lambda}|\boldsymbol{\kappa}) = [\ \boldsymbol{\delta_P}(\boldsymbol{\lambda}|\boldsymbol{\kappa}), \boldsymbol{\delta_M}(\boldsymbol{\lambda}|\boldsymbol{\kappa})\ ]' \\ which\ minimizes\ (62) \\ subject\ to\ (32)\ \text{to}\ (61) \end{gathered}$$

The objective function (62) contains absolute functions, which are not directly trac table for MILP formulation. Therefore, these absolute functions are equivalently represented as follows:

$$\begin{aligned} & J(\boldsymbol{u}(\lambda|\kappa), \cdots, \boldsymbol{u}(\lambda + N_I|\kappa) \\ & \quad , \boldsymbol{x}(\lambda|\kappa), \cdots, \boldsymbol{x}(\lambda + N_I|\kappa) \\ & \quad , \boldsymbol{\delta}(\lambda|\kappa), \cdots, \boldsymbol{\delta}(\lambda + N_I|\kappa)) \\ = & \sum_{\lambda=1}^{N} \Bigg\{ - \sum_i w_{1,i} \Bigg\{ \Bigg( \boldsymbol{\Theta_i} \begin{bmatrix} x_i(\lambda|\kappa)/l_i \\ x_{i+1}(\lambda|\kappa)/l_{i+1} \end{bmatrix} \\ & \quad + \boldsymbol{\Phi} \Bigg)' \boldsymbol{\delta_{M,i}}(\lambda|\kappa) \Bigg\} \\ & \quad - \sum_i w_{2,i} \Big\{ 1 - \left( e^+_{u,i}(\lambda|\kappa) + e^-_{u,i}(\lambda|\kappa) \right) \Big\} \\ & \quad + \sum_i w_{3,i} \Big\{ \left( e^+_{x,i}(\lambda|\kappa) + e^-_{x,i}(\lambda|\kappa) \right) \Big\} \Bigg\} \end{aligned} \tag{65}$$

where

$$u_i(\lambda|\kappa) - u_i(\lambda + 1|\kappa) = e^+_{u,i}(\lambda|\kappa) - e^-_{u,i}(\lambda|\kappa), \tag{66}$$

$$\frac{x_i(\lambda|\kappa)}{l_i} - \frac{x_{i+1}(\lambda|\kappa)}{l_{i+1}} = e^+_{x,i}(\lambda|\kappa) - e^-_{x,i}(\lambda|\kappa), \tag{67}$$

$$e^+_{u,i}(\lambda|\kappa) \geq 0 \quad , \quad e^-_{u,i}(\lambda|\kappa) \geq 0, \tag{68}$$

$$e^+_{x,i}(\lambda|\kappa) \geq 0 \quad , \quad e^-_{x,i}(\lambda|\kappa) \geq 0. \tag{69}$$

The MLDS formulation coupled with RHC scheme can be transformed to the canonical form of 0-1 Mixed Integer Linear Programming (MILP) problem to find optimal solution for the objective function (65).
Note that the requirements (R1), (R2) and (R3) also can be realized by solving Mixed Integer Quadratic Programming (MIQP) problem instead of solving Mixed Integer Linear Programming (MILP) problem as in this paper. However, since RHC scheme is by nature computationally demanding as is witnessed by many applications, the computational effort is one of the key performance criteria. In this regard, this paper firstly handles MILP problem with the objective function of (65) that has faster procedure in solution method than conventional MIQP problems have. And next subchapter, this paper presents a new algorithm designed to reduce computational amount in 0-1 MILP problems.

## 4. Convexity Analysis

The problem we formulated in the previous section is recast to the convex programming problem in this subsection. The convex programming problem, where the constraint and objective functions are convex, has become quite popular recently for a number of reasons. Some of them are summarized as follows : (1) The global optimality is guaranteed for the obtained solution, (2) The attractive algorithm is easily applied, obtaining the solution with high speed due to the simple structure of the problem, and (3) The bounding process can be efficiently applied for the MINLP problem.

### 4.1 Convexity Analysis

In this subsection we first introduce the well-known performance criteria of traffic network control system and show they can be realized with convex functions. The following performance criteria are introduced in this paper: (1) maximization of traffic flow and (2) minimization of traffic density difference between neighboring districts. These criteria are numerically represented as follows,

$$f = \sum_{\eta=0}^{H-1} \sum_{i=0}^{N-1} z_i(\kappa+\eta), \tag{70}$$

and

$$f = \sum_{\eta=1}^{H} \sum_{i=0}^{M-1} |x_i(\kappa+\eta|\kappa) - x_{i+1}(\kappa+\eta|\kappa)|, \tag{71}$$

where $H$ is the predictive horizon and $N$ and $M$ are the dimension of $\boldsymbol{x}(\kappa)$ and $\boldsymbol{z}(\kappa)$, respectively.
In order to verify the convexity of (70), we first show the traffic flow dynamics with three modes are convex functions at each mode, and show that these dynamics at each mode are continuous to the neighboring ones. By using this continuity, the overall dynamics of the traffic flow is proven to be convex.

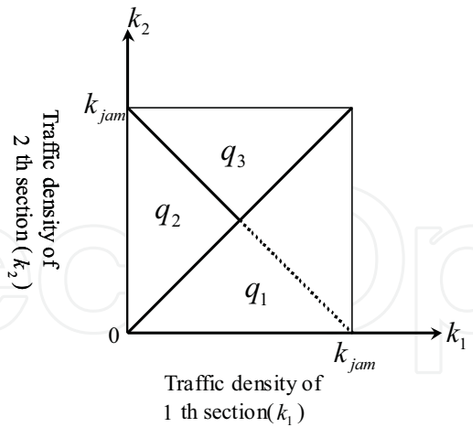


Fig. 13. Assignation of traffic flow mode

Consider Fig.(13), where each mode of traffic flow is assigned. Since the Hessian matrices of $q_1(\frac{x_1}{l_1}, \frac{x_2}{l_2})$, $q_2(\frac{x_1}{l_1}, \frac{x_2}{l_2})$, and $q_3(\frac{x_1}{l_1}, \frac{x_2}{l_2})$ are nonsingular as follows,

$$\begin{aligned} \nabla^2 q_1(\boldsymbol{x}) &= \left[\frac{\partial^2 q_1(\boldsymbol{x})}{\partial x_1 \partial x_2}\right] \\ &= \begin{bmatrix} \frac{v_f}{2x_{jam}} & \frac{v_f}{2x_{jam}} \\ \frac{v_f}{2x_{jam}} & \frac{v_f}{2x_{jam}} \end{bmatrix} \geq 0, \end{aligned} \tag{72}$$

$$\nabla^2 q_2(\boldsymbol{x}) = \left[\frac{\partial^2 q_2(\boldsymbol{x})}{\partial x_1 \partial x_2}\right] = \begin{bmatrix} \frac{v_f}{2x_{jam}} & 0 \\ 0 & 0 \end{bmatrix} \geq 0, \tag{73}$$

and

$$\begin{aligned} \nabla^2 q_3(\boldsymbol{x}) &= \left[\frac{\partial^2 q_3(\boldsymbol{x})}{\partial x_1 \partial x_2}\right] \\ &= \begin{bmatrix} 0 & 0 \\ 0 & \frac{v_f}{2x_{jam}} \end{bmatrix} \geq 0, \end{aligned} \tag{74}$$

they are convex at each mode.
In order to show the convexity of the overall dynamics of the traffic flow, we use following lemma :

**Lemma 1** The neighboring two closed convex dynamics $D_1(\Psi=(\psi_1, \psi_2, \cdots, \psi_n))$ and $D_2(\Psi)$ are convex if they are continuous at the boundary point $(\hat{\psi}_1, \hat{\psi}_2, \cdots, \hat{\psi}_n) \in \Theta$ $(\Theta = \overline{D_1(\Psi)} \bigcap \overline{D_2(\Psi) \backslash D_1(\Psi)})$ and satisfy that

if for $\forall\, i$, $\gamma$ and $\mu$

$$\nabla_\gamma D_1(\Psi)\Big|_{\psi_i=\hat{\psi}_i} \leq (\geq) \nabla_\gamma D_2(\Psi)\Big|_{\psi_i=\hat{\psi}_i} \tag{75}$$

then

$$\nabla^2_{\mu,\mu} D_1(\Psi)\Big|_{\psi_i=\hat{\psi}_i} \leq (\geq) \nabla^2_{\mu,\mu} D_2(\Psi)\Big|_{\psi_i=\hat{\psi}_i} \tag{76}$$

where overline denote the closure of the set, $1 \leq i, \gamma, \mu \leq n$, $\nabla_\gamma D$ is the $\gamma$th element of $\nabla D$, and $\nabla^2_{\mu,\mu} D$ is the $(\mu, \mu)$th element of the matrix $\nabla^2 D$.

The continuity at the boundary is easily confirmed by letting $k_1(\tau) = k_2(\tau) = k(\tau)$ as follows,

$$\begin{aligned} q_1(k_1(\tau), k_2(\tau)) &= q_2(k_1(\tau), k_2(\tau)) && (77)\\ &= q_3(k_1(\tau), k_2(\tau)) && (78)\\ &= k(\tau) v_f \Big(1 - \frac{k(\tau)}{k_{jam}}\Big). && (79) \end{aligned}$$

Lastly, with following eqs. (80) to (83),

$$\begin{aligned} \nabla q_1(\boldsymbol{x})\Big|_{x_1=\hat{x_1}} &= \left[\frac{v_f}{k_j}k - \frac{v_f}{2}, \frac{v_f}{k_j}k - \frac{v_f}{2}\right] && (80)\\ \nabla^2 q_1(\boldsymbol{x})\Big|_{x_1=\hat{x_1}} &= \begin{bmatrix} \frac{v_f}{2x_{jam}} & \frac{v_f}{2x_{jam}} \\ \frac{v_f}{2x_{jam}} & \frac{v_f}{2x_{jam}} \end{bmatrix} && (81)\\ \nabla q_2(\boldsymbol{x})\Big|_{x_1=\hat{x_1}} &= \left[2v_f \frac{k}{k_{jam}} - v_f, 0\right] && (82)\\ \nabla^2 q_2(\boldsymbol{x})\Big|_{x_1=\hat{x_1}} &= \begin{bmatrix} \frac{v_f}{2x_{jam}} & 0 \\ 0 & 0 \end{bmatrix}, && (83) \end{aligned}$$

the convexity condition of lemma 1 was satisfied, since

$$\nabla_1 q_1(x) \leq \nabla_1 q_2(x) \tag{84}$$

in pair with

$$\nabla^2_1 q_1(x) \leq \nabla^2_1 q_2(x). \tag{85}$$

In the same way,

$$\nabla_1 q_2(x) \leq \nabla_1 q_3(x) \quad , \quad \nabla_1 q_1(x) \leq \nabla_1 q_3(x) \tag{86}$$

are satisfied , paired together with

$$\nabla^2_1 q_2(x) \leq \nabla^2_1 q_3(x) \quad , \quad \nabla^2_1 q_1(x) \leq \nabla^2_1 q_3(x). \tag{87}$$

Therefore, the convexity of overall dynamics are confirmed.
Note that although $z$ is the multiplication of $q$ and $u$, the performance criteria (70) is a convex function. This is because $u$ is the vector whose elements $u_i \in \{0, 1\}$ are binary variables, if $u_i = 1$, $z_i$ remains as it stands now, otherwise the term $z_i$ is dropped off from the performance

criterion. And (71) is also a convex function, since $|x_1 - x_2|$ can be transformed to $(e_x^+ + e_x^-)$, minimizing $e_x^+ + e_x^-$ with the conditions of

$$e_x^+ \geq 0 \tag{88}$$

$$e_x^- \geq 0 \tag{89}$$

$$e_x^+ - e_x^- = x_1 - x_2 \tag{90}$$

where $e_x^+$ and $e_x^-$ are equivalently

$$e_x^+ = \frac{(x_1 - x_2) + |x_1 - x_2|}{2} \tag{91}$$

$$e_x^- = \frac{-(x_1 - x_2) + |x_1 - x_2|}{2}. \tag{92}$$

Since all the constraints are described in the form of Eq.(30), the problems (70) and (71) are included in the class of the convex programming problem.

### 4.2 Convex Programming

The efficient method such as Penalty Method (PM) can be easily applied to the convex programming problem with performance scheme as follows,

$$\text{minimize} \qquad F(\boldsymbol{x}, r) = f(\boldsymbol{x}) + r\boldsymbol{P}(\boldsymbol{x}) \tag{93}$$

$$\boldsymbol{P}(\boldsymbol{x}) \qquad \begin{cases} = 0, & \boldsymbol{x} \in X \\ > 0, & \boldsymbol{x} \notin X \end{cases} \tag{94}$$

where $f(\boldsymbol{x})$ is the convex performance criterion of the original problem, $r(> 0)$ is the cost coefficient which increases as iteration $l$ increases, $X$ is the convex set, and $P$ is the continuous penalty function satisfying Eq.(94).
This function can be constructed as follows.

**Step 1** Describe the solution space in the following form: $\boldsymbol{Gx} \leq \boldsymbol{W}$.

**Step 2** Define active constraints as the set of constraints which fulfill $\boldsymbol{G_i x} = \boldsymbol{W_i}$, and inactive constraints as the set which fulfills $\boldsymbol{G_i x} < \boldsymbol{W_i}$. Here, $\boldsymbol{G_i}$ and $\boldsymbol{W_i}$ are the $i$th raw of the matrix $\boldsymbol{G}$ and $\boldsymbol{W}$, respectively. The active set $\Sigma(\boldsymbol{x})$ is the set of indices of the active constraints, that is, $\Sigma(\boldsymbol{x}) = \{i \in \{1, \cdots, q\} | \boldsymbol{G_i x} = \boldsymbol{W_i}\}$.

**Step 3** Define $p_i$ as follows :

$$p_i(\boldsymbol{x}) = \boldsymbol{N_i x} + \boldsymbol{e} \tag{95}$$

where $\boldsymbol{x} \in \Re^n$, $\boldsymbol{N} \in \Re^n$, $|\boldsymbol{N}| = 1$ is the unit normal vector to the line $\boldsymbol{G_i x} - \boldsymbol{W_i} = 0$, and $\boldsymbol{e} \in \Re^n$ is the vector which describes parallel translation from the origin. Note that $\boldsymbol{N}$ takes outward direction from the convex sets defined by the active constraints, that is $\boldsymbol{Nx} + \boldsymbol{e} \leq 0$ for the feasible solution $\boldsymbol{x}$.

Step 4 Obtain the distance $d_i$ as follows,

$$\textit{if} \quad \boldsymbol{G_i x} - \boldsymbol{W_i} \leq 0, \textit{then} \quad d_i(\boldsymbol{x}) = 0$$

$$\textit{otherwise} \quad d_i(\boldsymbol{x}) = |p_i(\boldsymbol{x})|$$

Therefore the condition (94) can be translated as follows,

$$\boldsymbol{P}(\boldsymbol{x}) \quad = \quad \begin{cases} if\ (\mathrm{Y} = \phi) & \boldsymbol{P} = 0 \\ otherwise & \boldsymbol{P} = min[\mathrm{Y}] \end{cases} \tag{96}$$

where Y is the set of $d_i(x)$, $(i \in \Sigma)$ which is not equal to 0.
The penalty algorithm is implemented as follows,

**Step 1** Select initial point $\boldsymbol{x}^{l_O}$ $(= \boldsymbol{x}^{l_I})$ and $r^1$, and set $l_I \equiv 1$ and $l_O \equiv 1$.

**Step 2** If $r^{l_O} P(\boldsymbol{x}^{l_O}) < \epsilon$, terminate the algorithm. Otherwise, set $r^{l_O+1} \equiv cr^{l_O}$, $l_O \equiv l_O + 1$ and $\boldsymbol{x}^{l_I} \equiv \boldsymbol{x}^{l_O}$.

**Step 3** If $||\nabla f(\boldsymbol{x})|| < \epsilon$, set $x^{l_O} \equiv x^{l_I}$ and go to Step 2. Otherwise, go to Step 4.

**Step 4** Find the steepest descent direction, $\boldsymbol{d}$ $(= -\nabla^T f(\boldsymbol{x}^{l_I}))$.

**Step 5** Find the step width $\alpha^{l_I}$, do $\boldsymbol{x}^{l_I+1} = \boldsymbol{x}^{l_I} + \alpha^{l_I}\boldsymbol{d}^{l_I}$, and set $l_I \equiv l_I + 1$. And go to Step 3.

Here, $\epsilon$ is small tolerance, and $c$ and $\alpha^{l_I}$ are heuristically obtained.
If we can select the feasible initial solution, the optimal solution would be found in a short time. In this paper, the existence of solution is verified as follows.

**Lemma 2** The range of $x_i(\kappa)$ where $1 \leq i \leq m$ is $0 \leq x_i(\kappa) \leq l_i k_{jam}$. If $x_i(\kappa+1)$ always exists within the range for all $i$ in the case of $0 \leq x_i(\kappa) \leq l_i k_{jam}$ for all $i$, the feasible solution $\boldsymbol{x}(\kappa+1)$ can be found.

**Proof :** Consider the following equation :

$$\begin{aligned} & l_i k_i(\kappa+1) - l_i k_i(\tau) \\ = \quad & -\boldsymbol{q}(k_{i-1}(\kappa), k_i(\kappa))T_s \\ & +\boldsymbol{q}(k_i(\kappa), k_{i+1}(\kappa))T_s. \end{aligned} \tag{97}$$

It is obvious that $x_i$ is within the range if and only if

$$l_i k_i(\tau) \quad \geq \quad -\boldsymbol{q}(k_i(\tau), k_{i+1}(\tau))T_s \tag{98}$$

$$l_i k_{jam} - l_i k_i(\tau) \quad \leq \quad \boldsymbol{q}(k_{i-1}(\tau), k_i(\tau))T_s. \tag{99}$$

By substituting $q$ of (98) to (14), following inequality is obtained from the both $k_i(\kappa) \geq k_{i+1}(\kappa)$ and $k_i(\kappa) < k_{i+1}(\kappa)$.

$$1 \quad \geq \quad \frac{v_f}{l_i}\left(1 - \frac{k_i}{k_{jam}}\right)T_s. \tag{100}$$

Since $\frac{v_f T_s}{l_i} \ll 1$, (98) can be easily confirmed. In the similar way, the condition (99) can be easily confirmed.

## 5. 0-1 Classification based on PWARX System

The MINLP based traffic network controller introduced in the previous chapter is generally known to require large computational effort. In this chapter, we propose a new controller design method for hybrid systems with a binary output. The proposed method develops a classification map of the modified PWARX system, which relates a binary output and all observational variables including past inputs and outputs. The output $y(\kappa)$ (which corresponds to the plant input $u_p(\kappa)$) is obtained by finding the corresponding cluster among the classification map, while in the conventional methods, the MINLP problems were solved at every sampling instant.
Figure describes a block diagram of the proposed controller design method, where the MINLP controller is constructed to control the traffic flow in each traffic intersection in a decentralized manner. The traffic inflows from the outside and outflow to the outside are closely affected by the traffic flows at the adjoining traffic intersections. In order to construct the classification map, we need history of inputs and output of the MINLP controller obtained by applying it to various situations of the network.

### 5.1 Classification problem of hybrid dynamics

The PWARX (Piece-Wise Auto Regressive eXogeneous) system is a well-formulated classification technique for a hybrid and nonlinear dynamics. The PWARX system contains the state vector $\boldsymbol{x}$ which consists of past inputs and past outputs of the system as follows

$$\boldsymbol{x}(\kappa) = [y'(\kappa-1), y'(\kappa-2), \cdots, y'(\kappa-n_a), \tag{101}$$

$$u'(\kappa-1), u'(\kappa-2), \cdots, u'(\kappa-n_b)]$$

and this vector is involved in one of the polyhedral convex regions defined by

$$\chi_i = \{\boldsymbol{x} | \boldsymbol{V}_i \boldsymbol{x}(\kappa) \leq \boldsymbol{W}_i\}. \tag{102}$$

The entire behavior of the state vector is represented in a piece-wise manner. The dynamics of each region is defined as follows

$$f_i(\boldsymbol{x}(\kappa)) = \boldsymbol{\theta}_i \boldsymbol{\rho}(\kappa) \tag{103}$$

where $\boldsymbol{\rho}(\kappa)$ is $[\boldsymbol{x}(\kappa), 1]'$, and $\boldsymbol{\theta}$ is the coefficient vector as follows.

$$\boldsymbol{\theta}_i = [a_{i,1}, \cdots, a_{i,n_a}, b'_{i,1}, \cdots, b'_{i,n_b}, f_i]' \tag{104}$$

The problem we address in this paper is a special classification problem where the output $y$ is a 0-1 binary variable, and very good classification performance is desirable even with very large number of the introduced clusters. If we plot the observational data in a pure (not mixed) cluster in the $\boldsymbol{x}$-$y(k)$ space, it will show always zero inclination, since we have a binary output, i.e., all the components of $\theta$, $a$ and $b$ expect for $f$ will be zeros.
For this type of clustering problem, the conventional PWARX system does not well reproduce the 0-1 output. Since it simultaneously obtains clusters and its (linear) dynamics applying the least squared method to each of the fixed number of clusters, the overall accuracy of their reproduced model is not so high. Furthermore they are very sensitive to the initialization concerning the number of clusters, center of initial clusters, and so on.

### 5.2 Classification based on PWARX system

The identification procedure of the hybrid dynamics using the PWARX system is described as follows.

Step 1 Set the number of clusters, $s$, centers of $s$ clusters, $\mu$, and the threshold value $\epsilon > 0$.

Step 2 Obtain the cluster $D_i$ of $\xi$ points which minimize the following performance criterion

$$J = \sum_{i=1}^{s} \sum_{\xi_j \in D_i} ||\xi_j - \mu_i||^2_{R_j^{-1}} \tag{105}$$

Step 3 Update the centers $\mu$ according to the following formula.

$$\tilde{\mu}_i = \frac{\sum_{j:\xi_j \in D_i} \xi_j w_j}{\sum_{j:\xi_j \in D_i} w_j} \tag{106}$$

If $max(||\tilde{\mu}_i - \mu_i||) < \epsilon$, exit, else set

$$\mu = \tilde{\mu}_i \tag{107}$$

and go to Step 2.

In Step 2, $R_j$ is defined as

$$R_j = \begin{bmatrix} V_j & 0 \\ 0 & Q_j \end{bmatrix} \tag{108}$$

where

$$V_j = \frac{S_j}{c - (n_a + n_b) + 1} (\Phi_j' \Phi_j)^{-1} \tag{109}$$

$$Q_j = \sum_{(x,y) \in C_j} (x - m_j)(x - m_j)' \tag{110}$$

$$\Phi_j = \begin{bmatrix} x_1 & x_2 & \cdots & x_c \\ 1 & 1 & \cdots & 1 \end{bmatrix} \tag{111}$$

$$S_j = y_{c_j}'(I - \Phi_j(\Phi_j'\Phi_j)^{-1}\Phi_j')y_{c_j} \tag{112}$$

$$m_j = \frac{1}{c} \sum_{(x,y) \in C_j} \boldsymbol{x}, j = 1, \cdots, N \tag{113}$$

$$\xi_j = [(\theta_j)', m_j'] \tag{114}$$

$$w_j = \frac{1}{\sqrt{(2\pi)^{(2n_a+2n_b+1)} \det(R_i)}}. \tag{115}$$

$V_j$ is the empirical covariance matrix which measures the relevance criterion, $Q_j$ is the scatter matrix which measures the sparsity of data in the cluster $j$, $S_j$ is the sum of squared residuals, $C_j$ is the cluster in the $\boldsymbol{x}$ space, $x_j$ is the regressor vector belonging to $C_j$, $y_{c_j}$ is the output vector included in $C_j$.

The main difference of this method from the conventional K-means method is that based on the confidence level $w_j$, the proposed method assigns the vectors $\xi$ to the cluster $D_i$ in the parameter vector $\theta$-$\boldsymbol{x}$ space, while K-means assigns the data to the cluster $C_i$ in the state vector $\boldsymbol{x}$ space. This property serves for identification of $y$ that mixed clusters are suppressed being referred to the dynamics of $y$.

### 5.3 0-1 classification based on modified PWARX system

The desired outputs are continued by the same values, 0 or 1 in the $\boldsymbol{x}$-$y$ space. All values except for the offset variable $f$ among parameters of $\theta$ will be zeros, i.e., the dynamics in $\theta$-$\boldsymbol{x}$ space will be almost same. Therefore in the conventional PWARX system, the regions with same dynamics were often considered to be included in the same cluster.
The proposed method described below is a hierarchical PWARX system for a 0-1 classification as follows.

Step 1 (*Initialization Process*) Set the number of clusters, $s$, the number of the splitting clusters, $s_r$, the cluster centers, $\mu_i$ ($i \in [1, s]$), the initial data group number $N$, the renew data group number $N'$ and the threshold values $\epsilon > 0$ and $\gamma > 0$. Using $K$-means, obtain small $N$ data groups so that neighboring data may be belonged to the same groups.

Step 2 (*Piecewise Fitting Process*) Obtain the cluster $D_i$ of $\xi$ points which minimizes the following performance criterion.

$$J_\chi = \sum_{i=1}^{s} \sum_{\xi_j \in D_i} ||\xi_j - \mu_i||^2_{R_j^{-1}} \tag{116}$$

Obtain the guard $\boldsymbol{V}_i$ and $\boldsymbol{W}_i$ by solving a quadratic problem for all $i$ and $i'$ which satisfy $1 \leq i \leq s$ and $1 \leq i' \leq s$ ($i \neq i'$) as follows.

$$\text{find} \qquad \boldsymbol{V}_{i,i'} \quad \text{and} \quad \boldsymbol{W}_{i,i'} \tag{117}$$

$$\text{minimize} \qquad \boldsymbol{V}_{i,i'} \boldsymbol{V}_{i,i'}^T \tag{118}$$

$$\text{subject to} \qquad \zeta_l (\boldsymbol{V}_{i,i'}^T \boldsymbol{x_l} + \boldsymbol{W}_{i,i'}) \geq 1 \tag{119}$$

where $l$ is the data number and $\zeta$ is defined as follows.

$$\zeta_l \quad = \quad \begin{cases} 1 & if \quad \xi(\boldsymbol{x_l}) \in D_i \\ -1 & if \quad \xi(\boldsymbol{x_l}) \in D_{i'} \end{cases} \tag{120}$$

Here $\xi(\boldsymbol{x})$ is the function which obtains the corresponding value of $\xi$ from $\boldsymbol{x}$, i.e., $\xi$ is a translation of $\boldsymbol{x}$ in the $\theta$-$\boldsymbol{x}$ space. Then $\boldsymbol{V}_i$ and $\boldsymbol{W}_i$ are obtained as follows.
$\boldsymbol{V}_i = [\boldsymbol{V}_{i,1}^T, \cdots, \boldsymbol{V}_{i,i-1}^T, \boldsymbol{V}_{i,i+1}^T, \cdots, \boldsymbol{V}_{i,s}^T]^T$ and $\boldsymbol{W}_i = [\boldsymbol{W}_{i,1}, \cdots, \boldsymbol{W}_{i,i-1}, \boldsymbol{W}_{i,i+1}, \cdots, \boldsymbol{W}_{i,s}]^T$.

Step 3 (*Cluster Updating Process*) Update the centers $\mu$ according to the following formula.

$$\tilde{\mu}_i = \frac{\sum_{j:\xi_j \in D_i} \xi_j w_j}{\sum_{j:\xi_j \in D_i} w_j} \tag{121}$$

If $max||\tilde{\mu}_i - \mu_i||) < \epsilon$, go to Step 4, otherwise set

$$\mu = \tilde{\mu}_i \tag{122}$$

and go to Step 2.

Step 4 (*Cluster Splitting Process*) Obtain $J_i$ for all $i \in [1, s]$ which is defined by

$$J_i = \sigma^2(y(\kappa)). \tag{123}$$

Step 4-1 For all $i \in [1, s]$, do the following. If $J_i \leq \gamma$, do the following

$$\chi = \chi - \chi_i \tag{124}$$

$$\chi_i = \{\boldsymbol{x} | \boldsymbol{V}_i \boldsymbol{x} \leq \boldsymbol{W}_i\} \tag{125}$$

otherwise set new centers of the $s_r$ clusters, $\mu_r$ in $D_i$ randomly, and do the following.

$$s = s + s_r \tag{126}$$

Here, $\sigma^2(y(\kappa))$ is the covariance of $y(\kappa)$ in the cluster $D_i$.

Step 4-2 Set $i_m$ as follows.

$$i_m = \arg \min_{i \in [1,s]} \sigma^2(y(\kappa)) \tag{127}$$

Step 4-3 If $J_{i_m} \leq \gamma$, terminate with success, otherwise, obtain $N'$ data group of the corresponding region of $D_{i_m}$ and go to Step 2.

Note that in Step 2, the maximum margin of the data point $\boldsymbol{x}$ from the hyper-plane $\boldsymbol{V}_{i,i'}\boldsymbol{x} + \boldsymbol{W}_{i,i'} \leq 0$ is proportional to $(\boldsymbol{V}_{i,i'}^T \boldsymbol{V}_{i,i'})^{-1}$ since letting the hyperplane which maximize the margin $\alpha$ from the data points $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ as follows

$$\boldsymbol{V}\boldsymbol{x}^+ + \boldsymbol{W} = \alpha \tag{128}$$

and

$$\boldsymbol{V}\boldsymbol{x}^- + \boldsymbol{W} = -\alpha, \tag{129}$$

the maximal margin $\alpha_{MAX}$ is defined as follows

$$\alpha_{MAX} = \frac{1}{2}\Big(\frac{\boldsymbol{V}}{||\boldsymbol{V}||_2}\boldsymbol{x}^+ - \frac{\boldsymbol{V}}{||\boldsymbol{V}||_2}\boldsymbol{x}^-\Big) \tag{130}$$

$$= \frac{1}{2||\boldsymbol{V}||_2}(\boldsymbol{V}\boldsymbol{x}^+ - \boldsymbol{V}\boldsymbol{x}^-) \tag{131}$$

$$= \frac{\alpha}{||\boldsymbol{V}||_2} \tag{132}$$

Therefore by minimizing $(\boldsymbol{V}_{i,i'}^T \boldsymbol{V}_{i,i'})$, the margin can be maximized.

## 6. Numerical Experiments

### 6.1 Numerical Environments

In order to show the usefulness of our proposed method, we show, in this section, some results of the numerical experiments. We considered the traffic network of Fig. 14, where the square network with $1000 \times 1000$ $[m^2]$ consists of 16 intersections and 112 districts, all with 2 lanes bi-directionally. Four controllers are applied to find optimal traffic light for the overall network. Each controller is assigned to the network with $500 \times 500$ $[m^2]$. We assume that from the outside of the network traffic flows of vehicles move into the network with random speeds, whereas the traffic flows inside the network, move from the network with the speed of infinity (no congestion arises and affects the traffic flow inside the network). The variables used in this paper are as follows; $\boldsymbol{x} \in \Re^{56}$, $\boldsymbol{q} \in \Re^{80}$, $\boldsymbol{\delta} \in \{0, 1\}^4$. We used (70) as a performance criterion. All results are obtained from simulations over 30 minutes, where the sampling interval $T_s$ is 10 [sec].

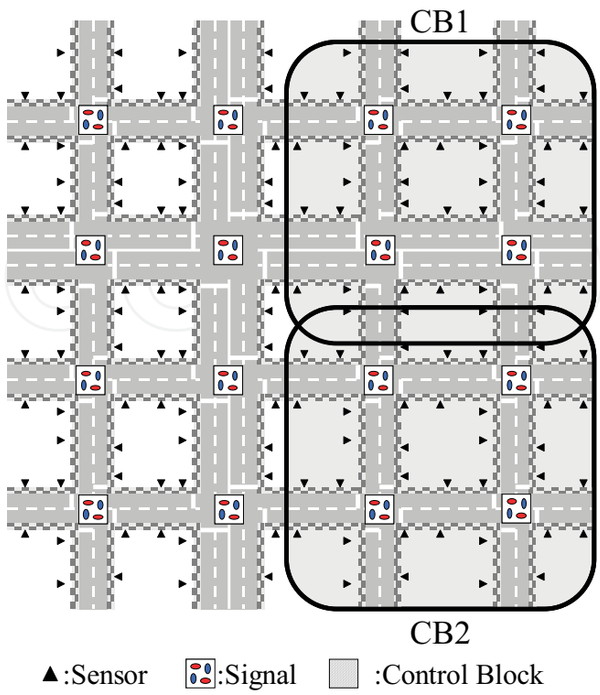


Fig. 14. Traffic network

| | No Control | $H = 1$ | $H = 2$ |
|---|---|---|---|
| A | 2724 | 2884 | 2913 |
| B | - | 3.1 | 370.4 |
| C | - | 1.2 | 14.6 |

Table 1. Numerical experimental result WRT $H$

### 6.2 Traffic Flow Control System for Traffic Network

We show the results obtained by applying our proposed methods in Table 1, where $H$ denotes the length of the prediction horizon, 'No Control' implies that the traffic light is changed at every 30 second, and

A: Number of cars passing through the boundary of every two consecutive districts,

B: Average computation time,

C: Average number of the generated sub-problems.

From the results in Table 1, we find that although the MPC with longer prediction horizon enables more vehicles to pass through the traffic network, the difference between the cases of $H = 1$ and $H = 2$ is not so remarkable. This implies that the proposed method can be applied to find semi-optimal solution for the real traffic control system with a proper selection of prediction horizon length.

| Length of $H$ | Proposed Method | | | Method of (1) | | |
|---|---|---|---|---|---|---|
| | A | B | C | A | B | C |
| 1 | 616 | 0.02 | 4 | 616 | 14.98 | 244 |
| 2 | 724 | 1.34 | 8 | 718 | 265.18 | 488 |
| 3 | 869 | 129.20 | 12 | 870 | 2688.6 | 732 |

Table 2. Comparison of the computational efforts

| | No Control | $H = 1$ | $H = 2$ |
|---|---|---|---|
| A | 5249 | 5660 | 5717 |
| B | - | 3.1 | 370.4 |
| C | - | 1.2 | 14.6 |

Table 3. Experimental result in case of no arterial road

### 6.3 Comparison of computational amount

In order to evaluate the computational amount of the presented method, we compare in Table 2 computational times obtained by applying our method and conventional method (1). We used Athlon XP 2400 and Windows 2000 for this experiments. Note that our method finds better solution with a shorter time. This is because the presented method does not approximate nonlinear dynamics(1) and solves non-linear programming problem, reformulating it to the convex programming problem. Furthermore, the presented refining process enables to avoid introduction of an enormous number of auxiliary variables.

### 6.4 TFCS for large-scale traffic network

In this subsection, the effectiveness of our method for large-scale traffic network control with the arterial roads is shown. If the traffic light controller is applied to the large-scale traffic network in a centralized manner, the computational amount would be fairly enormous. The presented method, as in Fig.14, designates the control block which groups some traffic lights in order that the feasible solutions may be obtained during the sampling interval. Fig. 14 illustrates that four control blocks (CB) constitute the entire traffic network where the sensory information at each boundary of CBs is shared for the control of both blocks. Note that two arterial roads are running north-south (second road from the left) and east-west (second road from the top), respectively. Table 3 and Table 4 show the obtained solutions by applying the presented method both in the case that there is no arterial roads and in the case that there are 2 arterial roads. In both numerical experiments, traffic densities at each road were set to exactly same value. The results in both cases show that the presented method has good solution in both cases. Note that our method has always better or equal solutions, compared with the cases of 'No Control'.

## 7. Classification Results

### 7.1 MINLP controller coupled with Model Predictive Control

The 5000 data sets obtained in the previous chapter are classified based on the 0-1 classification method. For this simulation, we set the number of initial clusters, $s$, to be 100 and whenever

| | No Control | $H = 1$ | $H = 2$ |
|---|---|---|---|
| A | 6060 | 6980 | 7185 |
| B | - | 3.6 | 250.4 |
| C | - | 1.3 | 10.4 |

Table 4. Experimental result in case of 2 arterial roads

| Step | Red | Blue | Mixed | Total |
|---|---|---|---|---|
| 1 | 7 | 5 | 38 | 50 |
| 2 | 27 | 29 | 32 | 88 |
| 3 | 48 | 46 | 26 | 120 |
| 4 | 64 | 63 | 19 | 146 |
| 5 | 72 | 75 | 18 | 165 |
| 6 | 87 | 81 | 15 | 183 |
| 7 | 95 | 92 | 11 | 198 |
| 8 | 102 | 98 | 9 | 209 |
| 9 | 110 | 103 | 5 | 218 |
| 10 | 114 | 107 | 2 | 223 |
| 11 | 116 | 109 | 0 | 225 |

Table 5. Stepwise Cluster Number (H=1)

we split the polyhedron defined by the guard $\boldsymbol{V}$ and $\boldsymbol{W}$ in the cluster splitting process, we split into two ($s_r$ =2).
We show the classification results in TABLE 5 - 8. In TABLE 5 and 7, "Red" and "Blue" imply the traffic signals of the clusters that if a data set is included in this cluster, the control input $u$ will represent this colors, while "Mixed" implies the clusters are not fully classified that Red and Blue signals are mixed in the cluster. The numbers of data in "Red", "Blue" and "Mixed" are shown in TABLE 6 and 8.
While the data shown in TABLE 5 and 6 are obtained by applying the MPC horizon $H$ = 1, the data shown in TABLE 7 and 8 are obtained by applying the MPC horizon $H$ = 3, respectively.

### 7.2 Comparison with conventional PWARX system

The conventional PWARX system is compared with our presented method. TABLE 9 and 10 compare the cluster number and the data number in the clusters. In TABLE 9 and 10, the conventional method is applied with the initial cluster number of 100, 200, 300, 400 and 500 respectively. Although most of data were well classified introducing a large number of clusters, 2.8 and 1.6 percents of the total data were not correctly classified. In contrast the presented method perfectly classified introducing relatively a small number of clusters.

## 8. Concluding remarks

In this paper we have presented a new design method for a traffic network hybrid feedback controller. Since the output of the traffic network controller is 0-1 binary signals, the output of the developed controller has been reproduced through 0-1 classifications of the PWARX systems. The developed PWARX classifier describes nonlinear feedback control laws of traffic

| Step | Red | Blue | Mixed |
|---|---|---|---|
| 1 | 188 | 238 | 2072 |
| 2 | 447 | 646 | 1405 |
| 3 | 670 | 801 | 1027 |
| 4 | 862 | 1013 | 623 |
| 5 | 952 | 1066 | 480 |
| 6 | 1063 | 1132 | 303 |
| 7 | 1110 | 1238 | 150 |
| 8 | 1131 | 1273 | 94 |
| 9 | 1149 | 1304 | 45 |
| 10 | 1169 | 1317 | 12 |
| 11 | 1171 | 1327 | 0 |

Table 6. Stepwise Data Number in the cluster(H=1)

| Step | Red | Blue | Mixed | Total |
|---|---|---|---|---|
| 1 | 5 | 8 | 37 | 50 |
| 2 | 23 | 27 | 37 | 87 |
| 3 | 45 | 46 | 33 | 124 |
| 4 | 66 | 66 | 25 | 157 |
| 5 | 81 | 82 | 19 | 182 |
| 6 | 92 | 93 | 16 | 201 |
| 7 | 105 | 105 | 7 | 217 |
| 8 | 112 | 108 | 4 | 224 |
| 9 | 116 | 112 | 0 | 228 |

Table 7. Stepwise Cluster Number (H=3)

| Step | Red | Blue | Mixed |
|---|---|---|---|
| 1 | 90 | 266 | 2142 |
| 2 | 356 | 570 | 1572 |
| 3 | 614 | 803 | 1081 |
| 4 | 794 | 999 | 705 |
| 5 | 953 | 1105 | 440 |
| 6 | 1010 | 1212 | 276 |
| 7 | 1058 | 1276 | 164 |
| 8 | 1138 | 1299 | 61 |
| 9 | 1175 | 1323 | 0 |

Table 8. Stepwise Data Number in the cluster(H=3)

| | Total | Red | Blue | Mixed |
|---|---|---|---|---|
| Proposed | 225 | 116 | 109 | 0 |
| Conventional | 100 | 30 | 32 | 38 |
| | 200 | 67 | 96 | 37 |
| | 300 | 127 | 135 | 38 |
| | 400 | 185 | 183 | 32 |
| | 500 | 228 | 255 | 17 |

Table 9. Comparison of cluster number (H=1)

| | Cluster Number | Data Number of Red Clusters | Blue | Mixed |
|---|---|---|---|---|
| Proposed | 225 | 1171 | 1327 | 0 |
| Conventional | 100 | 614 | 853 | 1031 |
| | 200 | 926 | 1113 | 459 |
| | 300 | 984 | 1180 | 334 |
| | 400 | 1014 | 1250 | 234 |
| | 500 | 1095 | 1263 | 140 |

Table 10. Comparison of data number in the cluster (H=1)

| | Total | Red | Blue | Mixed |
|---|---|---|---|---|
| Proposed | 228 | 116 | 112 | 0 |
| Conventional | 100 | 32 | 29 | 39 |
| | 200 | 78 | 83 | 39 |
| | 300 | 136 | 136 | 28 |
| | 400 | 186 | 184 | 30 |
| | 500 | 240 | 246 | 14 |

Table 11. Comparison of cluster number (H=3)

| | Cluster Number | Data Number of Red Clusters | Blue | Mixed |
|---|---|---|---|---|
| Proposed | 228 | 1175 | 1323 | 0 |
| Conventional | 100 | 609 | 715 | 1174 |
| | 200 | 830 | 1089 | 579 |
| | 300 | 1040 | 1245 | 213 |
| | 400 | 1047 | 1227 | 224 |
| | 500 | 1148 | 1266 | 84 |

Table 12. Comparison of data number in the cluster (H=3)

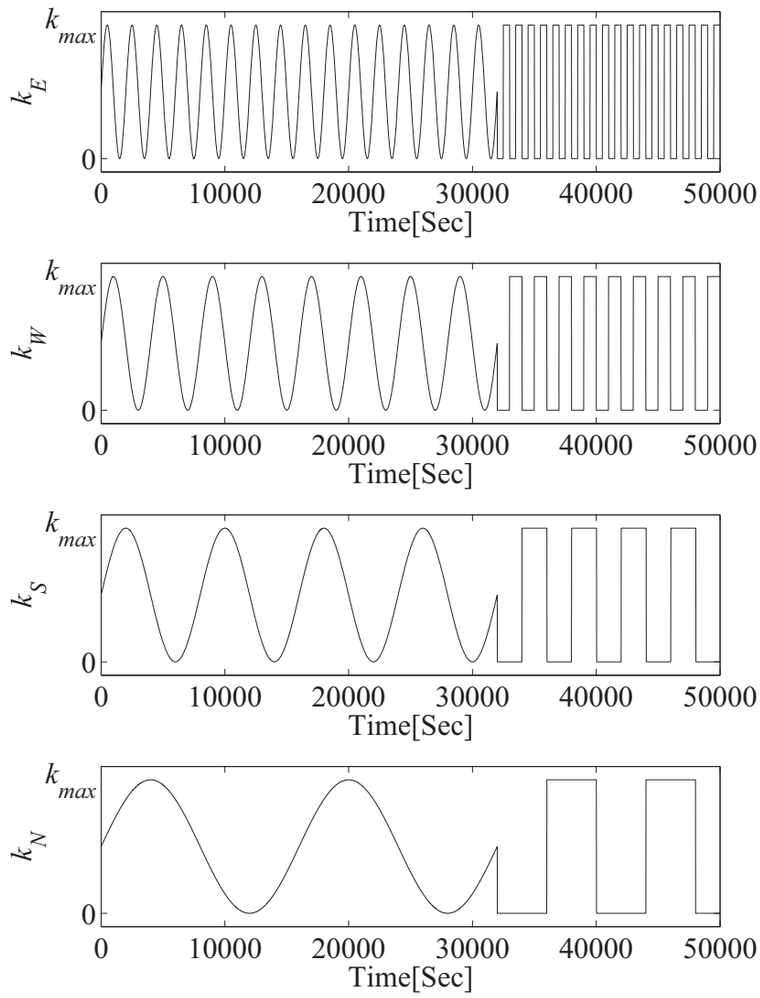


Fig. 15. Density of traffic flow

control systems. As we checked in chapter VII, very good solutions are obtained in a very short time, compared with the one obtained with the conventional MINLP controller.
In a classification problem considered in this paper, very good classification performance is required even with very large number of the introduced clusters. In our PWARX system formulation, we have adopted a new performance criterion related with the covariance of the control output. If a well-classified cluster is found, the cluster is separated from the classification map. If a bad-classified mixed cluster is found, the cluster is split into smaller $s_r$ pieces, and at the next iteration, this cluster is reclassified. The developed classification method has been applied to a traffic network control system, successfully reproducing the output of the conventional MINLP controller.

[Matrices in MLDS]

$$\boldsymbol{A} = \boldsymbol{I} \tag{133}$$

$$\boldsymbol{B} = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & -1 & 1 & 0 & \ddots & \ddots & \vdots \\ 0 & -1 & 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \tag{134}$$

$$\boldsymbol{E_1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{I} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Gamma_1} \end{bmatrix}^T \tag{135}$$

$$\boldsymbol{E_2} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{I} & \boldsymbol{I} & \mathbf{0} \end{bmatrix}^T \tag{136}$$

$$\boldsymbol{E_4} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^T \tag{137}$$

$$\boldsymbol{E_5} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda_x} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda_y} & \boldsymbol{\Gamma_5} \end{bmatrix}^T \tag{138}$$

where

$$\boldsymbol{\Lambda_x} = \begin{bmatrix} -x_{max} & -x_{max} & \cdots & -x_{max} \end{bmatrix} \tag{139}$$

$$\boldsymbol{\Lambda_z} = \begin{bmatrix} z_{max} & z_{max} & \cdots & z_{max} \end{bmatrix} \tag{140}$$

$$\boldsymbol{\Gamma_1} = \begin{bmatrix} \boldsymbol{\gamma_1} & 0 & \cdots & \cdots & 0 \\ 0 & \boldsymbol{\gamma_1} & 0 & \ddots & 0 \\ \vdots & 0 & \boldsymbol{\gamma_1} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \boldsymbol{\gamma_1} \end{bmatrix} \tag{141}$$

$$\boldsymbol{\Gamma_5} = \begin{bmatrix} \boldsymbol{\gamma_5} & \boldsymbol{\gamma_5} & \cdots & \boldsymbol{\gamma_5} \end{bmatrix}^T \tag{142}$$

$$\boldsymbol{\gamma_1} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \tag{143}$$

$$\boldsymbol{\gamma_5} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{144}$$

$$\boldsymbol{u} = \begin{bmatrix} u_{E,1} \\ u_{W,1} \\ u_{N,1} \\ u_{S,1} \\ u_{E,2} \\ u_{W,2} \\ u_{N,2} \\ u_{S,2} \\ \vdots \\ u_{S,m} \end{bmatrix} \tag{145}$$

## 9. References

[1] Tatsuya Kato and YoungWoo Kim and Shigeru Okuma: Model Predictive Control of Traffic Flow Based on Hybrid System Modeling, ICCAS, pp.368-373 (2005)
[2] Hayakawa Hisao: Considering the traffic flow as the pulverulent body (in Japanese), Parity, Vol.13, No.5, pp.13–22 (1998)
[3] Yasuyoshi Kato: Traffic Flow Simulation by Cellular Automaton Method, JSIE, Vol.15, No.2, pp.242–250 (2000)
[4] K. Nagel and M. Schreckenberg. A: Cellular Automaton Model for Freeway Traffic, Journal de Physique I France, pp.2–2221 (1992)
[5] H. J. Payne: Models of freeway traffic and control in Mathematical Models of Public Systems, Simulation Council Proceedings Series, LaJolla, California,Vol.1, No.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. London Ser. A, Vol.229, pp.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, Mathematics and Computers in Simulation, Vol.28 (1986, pp.297–303)
[9] Balduzzi, F.and Giua, A. and Menga, G: First-order hybrid Petri nets: a model for optimization and control., IEEE Trans. on Robotics and Automation, Vol.16, No.4, pp.382-399 (2000)
[10] Richard Haberman: Mathematical Models, Prentice-Hall (1977)
[11] Chaudahuri, P.P and others: Additive Cellular Automata -Theory and Applications, IEEE Computer Society Press (1997)
[12] 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)
[13] Camacho, E.F. and Bordons, C.: Model predictive control in the process industry, Springer-Verlag (1995)

IntechOpen


**Petri Nets Applications**

Edited by Pawel Pawlewski

ISBN 978-953-307-047-6

Hard cover, 752 pages

**Publisher** InTech

**Published online** 01, February, 2010

**Published in print edition** February, 2010

Petri Nets are graphical and mathematical tool used in many different science domains. Their characteristic features are the intuitive graphical modeling language and advanced formal analysis method. The concurrence of performed actions is the natural phenomenon due to which Petri Nets are perceived as mathematical tool for modeling concurrent systems. The nets whose model was extended with the time model can be applied in modeling real-time systems. Petri Nets were introduced in the doctoral dissertation by K.A. Petri, titled "„Kommunikation mit Automaten" and published in 1962 by University of Bonn. During more than 40 years of development of this theory, many different classes were formed and the scope of applications was extended. Depending on particular needs, the net definition was changed and adjusted to the considered problem. The unusual "flexibility" of this theory makes it possible to introduce all these modifications. Owing to varied currently known net classes, it is relatively easy to find a proper class for the specific application. The present monograph shows the whole spectrum of Petri Nets applications, from classic applications (to which the theory is specially dedicated) like computer science and control systems, through fault diagnosis, manufacturing, power systems, traffic systems, transport and down to Web applications. At the same time, the publication describes the diversity of investigations performed with use of Petri Nets in science centers all over the world.



**InTech Europe**

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen