Nagoya University

Traffic Network System Control Based on Hybrid Dynamical System

Graduate School of Nagoya University
Department of Electrical Engineering and Computer Science
Okuma laboratory
Tatsuya KATO

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Traffic network system

The traffic network system is the collective entity of traffic network, traffic flow and traffic signals. We can find several traffic network systems and some of them include the marine traffic system, the air traffic system, the railway traffic system and so on. Among them, the road traffic system is most closely linked with modern lives.

The management of the road traffic system (hereafter traffic system) is very important especially in urban lifestyle, where number of vehicles is increased rapidly, and traffic signal is not fully efficient from place to place. At some traffic inter-sections, very big and chronic traffic congestion is generated by heavy traffic jam. Also at other traffic inter-sections, too wide roads or tracks are paved regarding the traffic flow. In order to fully utilize the existing traffic network and to reduce large waste of resources such as fuel, the well-developed traffic network control systems are needed, where some of control parameters are traffic signal cycle length, split of traffic signal, offset of traffic signal and so on.

In order to alleviate the traffic congestion, many approaches have been proposed. They are 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 car is affected by neighboring cars, and the entire traffic flow is represented as the statistical occurrences [1]. The Cellular Automaton (CA) based model[2][3][4][5] 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 described into many small cells. Each cell can be either empty or occupied by only one car. The behavior of each car in each cell is specified by the geometrical relationship with other cars together with stochastic parameters. Also, in the FL model, each car is supposed to have a tracking response to the preceding car, which is described by first order or second order differential equations. 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 because it requires enormous computational efforts to find all cars' behavior. Furthermore, the precise information on initial positions and speeds of all cars are usually not available in advance.

On the other hand, it has been 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;

$$\frac{\partial k(x,t)}{\partial t} + \frac{\partial q(x,t)}{\partial x} = 0. \tag{1.1}$$

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

$$q(x,t) = k(x,t)v(x,t), \tag{1.2}$$

is introduced together with the appropriate model of the $v(x,t)$, where $v(x,t)$ denotes the velocity of the 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 is also applicable to the urban traffic network which involves many discontinuities of the density coming from the existence of the intersection controlled by the 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 [6] [7] [8] [9][10][11][12][13][14][15][16][17][18][19]. 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 an integrated model descriptions in the Hybrid Dynamical System (HDS) framework. 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 the Mixed Logical Dynamical Systems (MLDS) form in order to introduce the optimization technique.

From these points of view, the author proposes the piece-wise affine traffic flow model[20][21], where the traffic flow is represented with the traffic densities of two consecutive districts in order to consider the behavior of shock wave. The traffic flow dynamics are optimized based on MLDS framework. The method used in [20] is the well-established optimization procedure. However, the method based on Mixed Integer Linear Programming (MILP) problem associated with piece-wise affine traffic flow dynamics is unfit for large-scale traffic network control, since it is computationally expensive. Consider the traffic light control of a pedestrian crossover on a one-way street. The previous method requires one binary variable (δ_S) to represent traffic light states, three binary variables (δ_P) to represent the traffic flow dynamics, and three binary variables (δ_M) to optimize the dynamics, transforming it to the linear form. This means in the worst case that MILP has 2^7 sub problems to solve[22][23][24][25][26][27].

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.

1.2 Hybrid dynamical system

In the literature researchers stated dealing with hybrid systems, namely hierarchical systems constituted by dynamical system components at the lower level, governed by upper level logical/discrete components[28][29]. Hybrid systems arise in a large number of applications areas , and are attracting increasing attention in both academic theory-oriented circles and in industry. Great interest is motivated by several clearly discernible trend in the process industries which point toward an extended need for new tools to design control and supervisory schemes for hybrid systems and analyze their performance.

For this class of systems, design procedures have been proposed which naturally lead to hierarchical, hybrid control schemes, with continuous controllers are the lower level calibrated for each dynamical subsystem in order to provide regulation and tracking properties, and discrete controllers supervising resolving conflicts, and planning strategies at a higher level [30] [31]. However, in several applications a precise distinction between different hierarchic levels is not possible, especially when dynamical and logical facts are dramatically interdependent. For such a class of systems, it is not clear how to design feedback controllers, and it is not known how to obtain models in a systematic way.

In this dissertation, a framework for modeling and controlling models of systems is handled by interacting physical laws, logical rules, and operating constraints.

According to the techniques described in [32][33][34], propositional logic is transformed into linear inequalities involving integer and continuous variables. This allows to arrive at Mixed Logical Dynamical Systems (MLDS) described by linear dynamic equations subject to linear mixed-integer inequalities, i.e. inequalities involving both and continuous and binary (logical or 0-1) variables. These include physical/discrete states, continuous/integer inputs, and continuous/binary auxiliary variables. The MLDS generalizes a wide set of models, among which there are linear hybrid systems, finite state machines, classes of discrete event systems, constrained linear systems, and nonlinear systems whose nonlinearities can be expressed (or, at least, suitably approximated) by piecewise linear functions.

The traffic flow control system is a typical hybrid dynamical system, where entire behavior of the system dynamics is constrained by the interaction of discrete state such as traffic signal, traffic mode and so on, and continuous state such as traffic flow, traffic density and so on.

1.3 Preparations for hybrid dynamical system

We introduce several tools for modeling and controlling of hybrid dynamical system.

1.3.1 Hybrid Petri net

Petri Net(PN) is well known as one of the most powerful modeling tools. As a graphical and mathematical tool, Petri Net has been used to provide a uniform environment for modeling, formal analysis, as well as systematic construction of discrete-event simulators and controllers. Petri Net were named after Carl A. Petri who created in 1962 a net-like mathematical tool for the study of communication with automata. Their further development was facilitated by the fact that Petri Nets(PN) can be used to model properties such as process synchronization,asynchronous events, sequential operations, concurrent operations, conflicts and resource sharing. These properties characterize Discrete-Event Systems(DES) whose examples include industrial automated systems, communication systems, and computer-based systems.

The Hybrid Petri Net has four types of node. A continuous place is represented by a double circle. A discrete place is represented by a single circle. A continuous transition is represented by a square. A discrete transition is represented by a bar. A continuous place indicates a continuous status, and it has continuous marking. A discrete place indicates a discrete status, and it has discrete marking. The 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

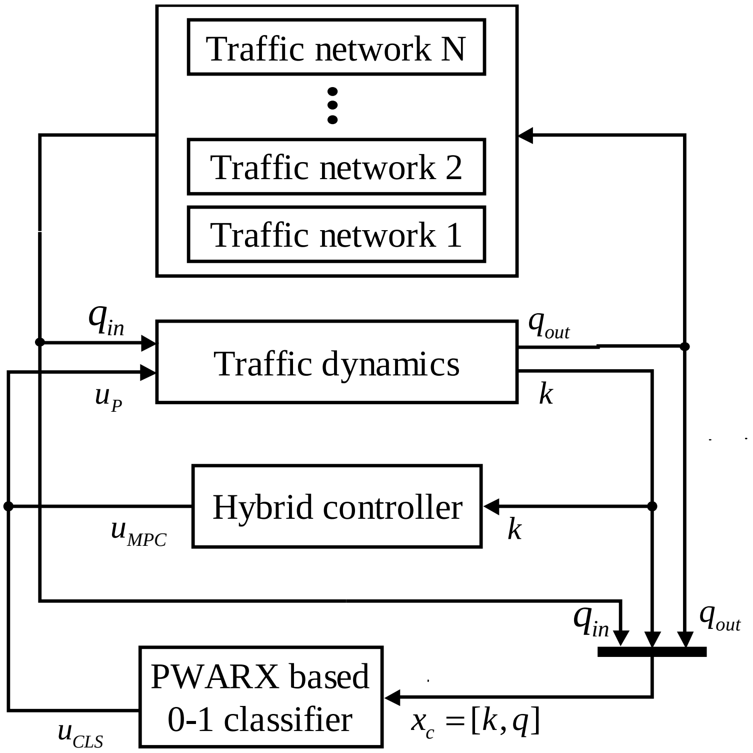

Figure 1.1: Controller outline

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).

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 state of the traffic signal. Thus, the HPN model can be regarded as one of the discretized macroscopic model of the traffic flow including 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 an human-machine interface due to its graphical understanding.

1.3.2 Mixed logical dynamical system

The Mixed Logical Dynamical System(MLDS) is powerful tool for modeling discrete-time linear hybrid systems. The MLDS form can generally be formalized as follows [35]:

$$\boldsymbol{x}(\kappa+1) = \boldsymbol{A_\kappa x}(\kappa) + \boldsymbol{B_{1\kappa} u}(\kappa) + \boldsymbol{B_{2\kappa} \delta}(\kappa) + \boldsymbol{B_{3\kappa} z}(\kappa), \tag{1.3}$$

$$\boldsymbol{y}(\kappa) = \boldsymbol{C_\kappa x}(\kappa) + \boldsymbol{D_{1\kappa} u}(\kappa) + \boldsymbol{D_{2\kappa} \delta}(\kappa) + \boldsymbol{D_{3\kappa}}(\kappa), \tag{1.4}$$

$$\boldsymbol{E_{2\kappa} \delta}(\kappa) + \boldsymbol{E_{3\kappa} z}(\kappa) \leq \boldsymbol{E_{1\kappa} u}(\kappa) + \boldsymbol{E_{4\kappa} x}(\kappa) + \boldsymbol{E_{5\kappa}}. \tag{1.5}$$

This is a extension of dynamical system. $\boldsymbol{x}$ represents a continuous status variable. $\boldsymbol{u}$ represents a continuous input variable. $\boldsymbol{y}$ represents a continuous output variable. $\boldsymbol{\delta}$ represents a discrete variables. $\boldsymbol{z}$ represents a binary and an auxiliary continuous variable. Equation (1.3) is a state equation,eq.(1.4) is an output equation, and eq.(1.5) is an inequality constraint[35].

When we look at the control problem for the traffic signal, it is natural to introduce algebraic representation of the traffic flow. Although the HPN has algebraic description, it is not suitable form to formulate optimization problems. 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 [36][37][38]. 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 description.

1.3.3 Mixed integer linear/non-linear programming problem

Mixed-Integer Linear/Non-Linear Programming (MILP/MINLP) is a very general framework for capturing problems with both discrete decisions and continuous variables. Mixed-integer optimization techniques have been investigated in [39][34], for chemical process synthesis. For feedback control purposes, we propose a predictive control scheme which is able to stabilize the MLDS on desired reference trajectories while fulfilling operating constraints, and possibly take into account previous qualitative knowledge in the form of heuristic rules.

Due to the presence of integer variables, the optimization procedure is a MILP/MINLP problem [40][41][42], for which efficient solvers exist [43]. A first attempt to use on-line mixed integer programming to control dynamic systems subject to logical conditions has appeared in [44].

1.3.4 Piecewise auto regressive exogenous system

A general method for obtaining the hybrid model is to use the PieceWise Affine (PWA) systems since the PWA approximation has universal properties and the obtained system can be directly transformed to several classes of hybrid dynamical systems. Both the state and output maps of the PWA systems are a piecewise affine form, where the PWA map $f : \chi \to R^q$ is defined as follows.

$$f(x) = \begin{cases} \theta_1^T \rho(k) & \text{if} \quad x(k) \in \chi_1 = \{x(k) | H_1 x(k) \leq W_1\} \\ \vdots & \qquad\qquad \vdots \\ \theta_s^T \rho(k) & \text{if} \quad x(k) \in \chi_s = \{x(k) | H_s x(k) \leq W_s\} \end{cases} \tag{1.6}$$

$$x = [y(k-1), \cdots, y(k-n_a), u^T(k-1), \cdots, u^T(k-n_b)]^T \tag{1.7}$$

$$\theta_i = [a_{i,1}, \cdots, a_{i,n_a}, b_{i,1}^T, \cdots, b_{i,n_b}^T, f_i]^T \tag{1.8}$$

, where $\rho(k)$ is $[x(k), 1]^T$ ($x(k)$ is the regression vector, consists of the past inputs and outputs), χ_i is the convex polyhedron which satisfies $\bigcup_{i=1}^{s} \chi_i = \chi \subseteq R^q$ and $\chi_i \bigcap \chi_j = \phi$, $\forall i \neq j$, y is the control output, and the pair (H_i, W_i) is the guard of χ_i.

The PieceWise Auto Regressive eXogenous (PWARX) model [45][46] is the discontinuous output map along the boundary of each region. The main difference compared with conventional K-means based clustering is that a confidence level is measured coupled with the covariance of the data in the θ-x space. χ_i and θ_i are obtained by iteratively applying the piecewise fitting process and the cluster updating process. Although this method is an efficient clustering procedure for the hybrid dynamical systems, the number of sub models must be a priori fixed, randomly choosing the initial bases of each clustering group. However, the performance of this iterative clustering procedure is in general very sensitive to the initialization. The hierarchical subspace clustering [47] is a challenging problem for identifying the system with the highly nonlinear and complex dynamics. This method is mainly applied in the neural networks community. However, the neural networks based hierarchical clustering schemes take long time for the network learning.

1.4 Objectives and organization of dissertation

Based on the backgrounds of the previous sections, the purpose of researches in this dissertation is to construct theory and control framework for the traffic network system. This dissertation presents

(1) Model predictive control of traffic flow based on hybrid system modeling

(2) Large-scale traffic network control based on convex programming coupled with B&B strategy

(3) Traffic network hybrid feedback controller via 0-1 classification of PWARX system with hierarchy

A brief overview of this dissertation is given in the following.

Chapter 2 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 (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 algebraic representation of the traffic network. Although the HPN has 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.

Chapter 3 presents a new method for large-scale traffic network control is proposed. First of all, we formulate, based on the Mixed Integer Non-Linear Programming (MINLP) problem, the traffic network control system that is one of typical hybrid systems with nonlinear dynamics. 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 the efficient method such as Steepest Descent Method (SDM). We use in chapter 3 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 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 yields substantial computational amounts increased. 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 (B&B) strategy. This implies we can find global optimal solution in a short time since no more auxiliary variables (such as δ_P and δ_M [20]) are introduced.

In chapter 2, we present the integrated model description for the large-scale traffic network control, where the geometrical information on the traffic network is characterized by using Hybrid Petri Net (HPN) [48, 49], and the algebraic description of the traffic flow is provided in consideration of the shock wave. And then they are integrated into MLDS formulation. In chapter 3, the developed model is recast to the canonical form of MINLP. And the proposed B&B algorithm coupled with convexity analysis is applied to the problem. Finally the usefulness of the proposed method is verified through numerical experiments.

Chapter 4 presents a new design method for the traffic network hybrid feedback controller. We propose a new design method for the traffic network hybrid feedback controller. The method reported in [50] is an elaborate contrivance to avoid redundant introduction of binary variables. Although the solution optimality was guaranteed in [50], this method requires much computational efforts. Since the output of the traffic network controller is the 0-1 binary signals, the control output obtained by applying the controller design method of [50], is reproduced in chapter 4 applying 0-1 classifications of PWARX systems. In the proposed method, the PWARX classifier describes the nonlinear feedback control law of the traffic control system. This implies we don't need the time-consuming searching process of the solver such as the B & B algorithm to solve MINLP problem, and furthermore the exactly same solutions or very similar solutions are obtained in a very short time.

The classification problem we address in chapter 4 is a special problem where the output y is a 0-1 binary variable, and very good classification performance is desirable even with large number of introduced cluster. If we plot the observational data in a same cluster in the x-y space, it shows zero inclination, since we have the binary output, i.e., all the components of θ, a and b expect for f are zeros. This implies we need consideration for the binary output. A new performance criterion is presented in chapter 4 to consider not only previously covariance of θ but also the covariance of y. The proposed method is a hierarchical classification procedure, where the cluster splitting process is introduced to the worst cluster at every iteration which includes 0-1 mixed values of y. The cluster splitting process is followed 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 numerical experiments.

Chapter 5 gives the summary of this dissertation and discusses the scope of future work.

Chapter 2

Model predictive control of traffic flow based on hybrid system modeling

2.1 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 [2][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 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;

$$\frac{\partial k(x,t)}{\partial t} + \frac{\partial q(x,t)}{\partial x} = 0. \tag{2.1}$$

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

$$q(x,t) = k(x,t)v(x,t), \tag{2.2}$$

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 impossible 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 consider 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 areas, has been introduced in [6] [7] [8] [9]. Although these approaches include judicious use of theoretical ideas of 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 System (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 number 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 that it can 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 solvers.

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.

2.2 Modeling of traffic flow control system based on Hybrid Petri net

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 Hybrid Petri Net(HPN) model is developed, which provides both graphical and algebraic descriptions for the TFCS.

2.2.1 Representation of traffic flow control system as Hybrid Petri net

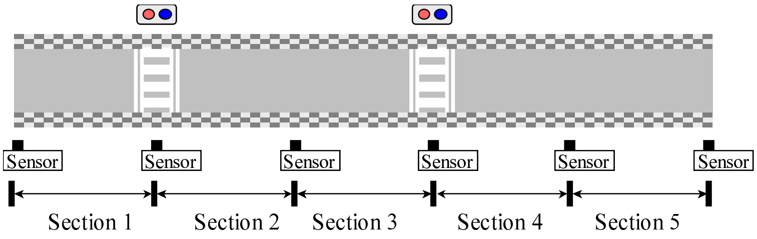

Figure 2.1: Straight road

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 of the traffic flow becomes possible. In this chapter, our targets are single straight load and simple crossroad. These load have some signal that have red and blue status. The input / output cars are assumed to same past time index data. Each section have traffic density sensor and we can use the real-time traffic density data.

The HPN is one of the useful tools to model and visualize the system behavior with both continuous and discrete variables. Figure 2.2 shows the HPN model for the road of Fig.2.1. In Fig.2.2, each section i of l_i-meters long constitutes the straight road, and two traffic lights are installed at the points of crosswalk. The 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 boundary of the section as show in Fig.2.1. Also, we do not consider any measurement error

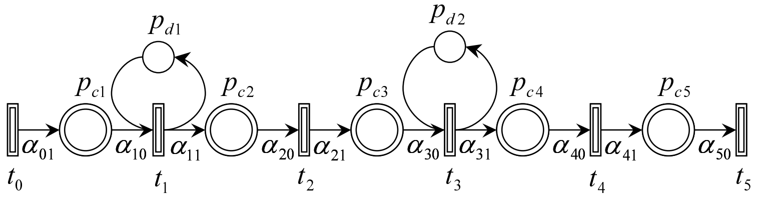

Figure 2.2: HPN model of straight road

for sensors in this chapter. The function $I_+(p,t)$ is a forward incidence relationship between transition t and place p which precedes the transition. The function $I_-(p,t)$ is a backward one which follow on the transition. The element of $I(p,t)$ is 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 the HPN is represented by a simple first order differential equation for each continuous place $p_{c_i} \in P_c$ as follows:

$$\frac{dm_{C,i}(\tau)}{dt} = \sum_{t_j \in p_{c_i}{}^\bullet \cup {}^\bullet p_{c_i}} I(p_{c_i}, t_j) \cdot q_j(\tau) \cdot m_{D,j}(\tau), \tag{2.3}$$

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 (2.3) is transformed to its discrete-time version, supposing that $q_j(\tau)$ is constant during two successive sampling instants as follows:

$$\begin{aligned} m_{C,i}((\kappa+1)T_s) &= m_{C,i}(\kappa T_s) + \\ & \sum_{t_j \in p_{c_i}{}^\bullet \cup {}^\bullet p_{c_i}} I(p_{c_i}, t_j) \cdot q_j(\kappa T_s) \cdot m_{D,j}(\kappa T_s) \cdot T_s. \end{aligned} \tag{2.4}$$

where κ is a sampling index, and T_s is a 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*.

2.2.2 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

$$q_i(\tau) = \frac{(k_i(\tau) + k_{i+1}(\tau))}{2} \frac{v_i(k_i(\tau)) + v_{i+1}(k_{i+1}(\tau))}{2} \tag{2.5}$$

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 th and $(i+1)$th sections are almost identical. Then, by incorporating the velocity model

$$v_i(\tau) = v_{f_i} \cdot \left(1 - \frac{k_i(\tau)}{k_{jam}}\right), \tag{2.6}$$

with (2.5), 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 vehicles exist 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 consider the discontinuities of the density among neighboring sections (i.e. neighboring continuous places), the idea of 'shock wave'[7] is introduced as follows. We consider the case as shown in Fig.2.3 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 [7]:

$$c_i(\tau) \quad = \quad \frac{v_i(\tau)k_i(\tau) - v_{i+1}(\tau)k_{i+1}(\tau)}{k_i(\tau) - k_{i+1}(\tau)}. \tag{2.7}$$

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

(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.2.3 ($c_i(\tau) > 0$) and Fig.2.4

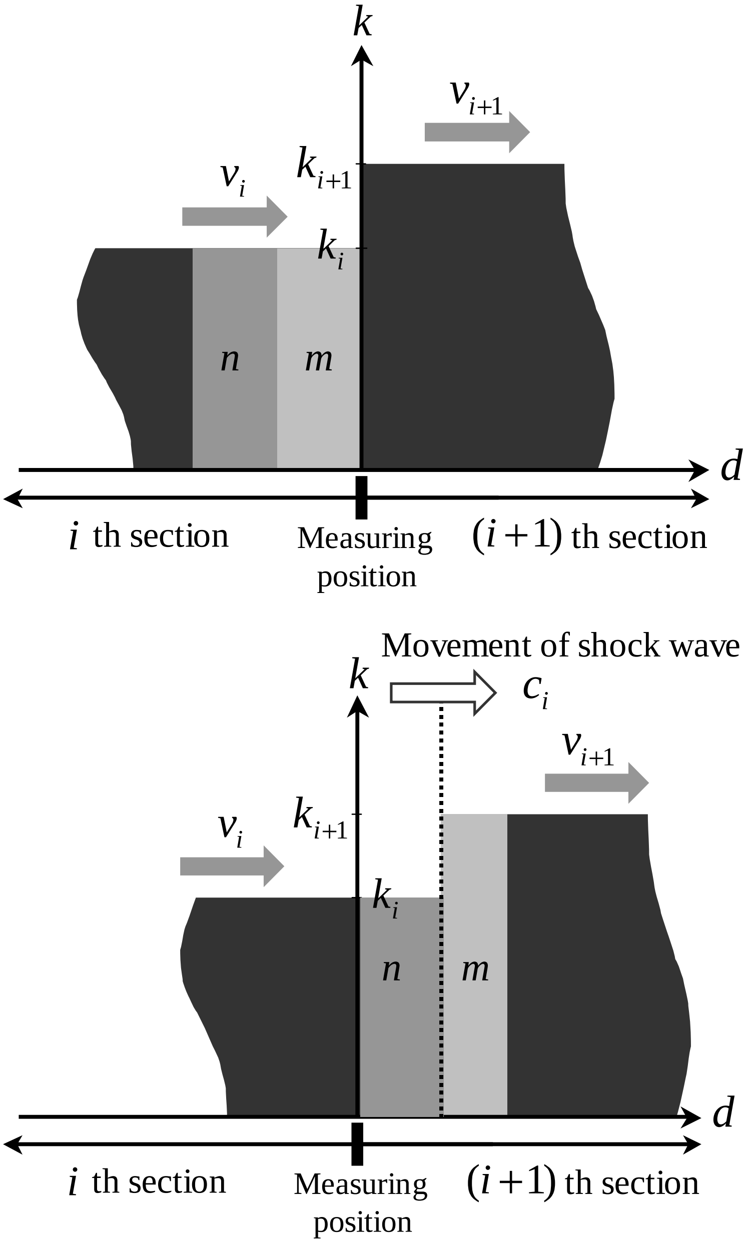

Figure 2.3: Movement of shock wave in the case of $k_i(\tau) < k_{i+1}(\tau)$ and $c_i(\tau) > 0$

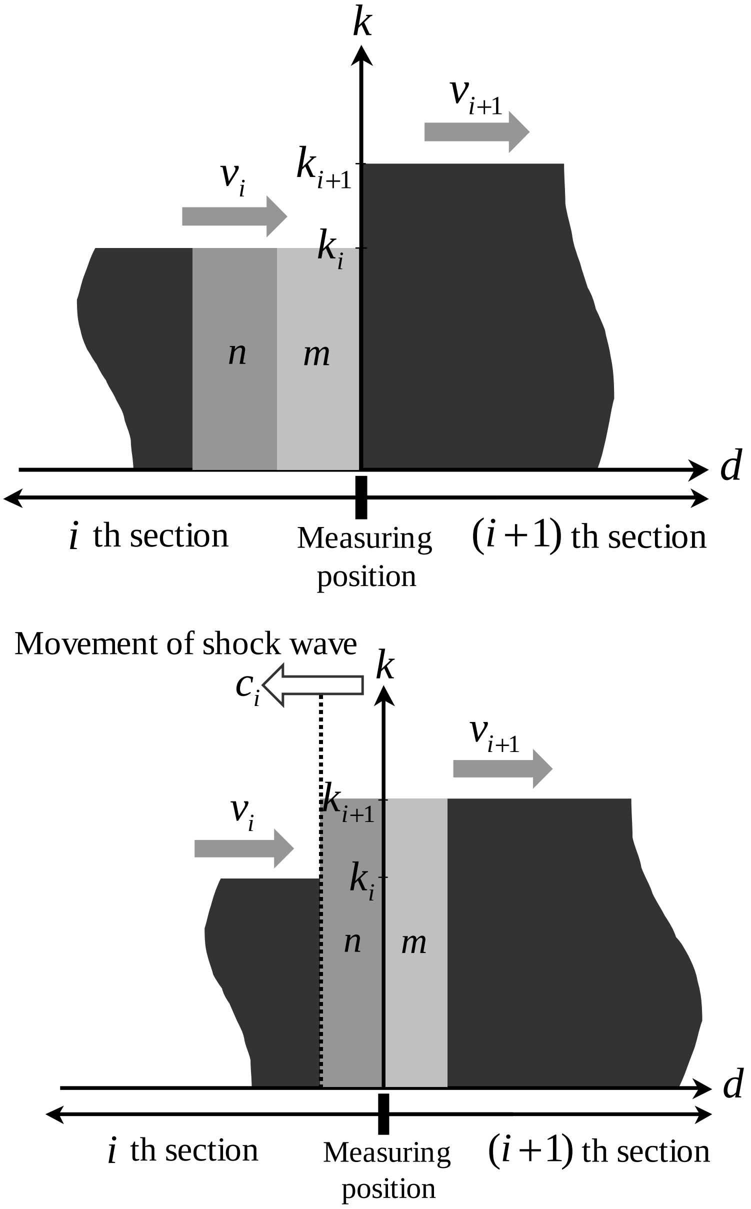

Figure 2.4: Movement of shock wave in the case of $k_i(\tau) < k_{i+1}(\tau)$ and $c_i(\tau) \leq 0$

$(c_i(\tau) \leq 0)$. In Fig.2.3 and 2.4, 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.2.3, where n and m represent the area of the corresponding rectangular, i.e. the product of the $v_i(\tau)$ and the $k_i(\tau)$. Similarly, in the case of (C2), $q_i(\tau)$ can be represented by m in Fig.2.4. These considerations lead to the following models:

in the case of (C1)

$$\begin{aligned} q_i(\tau) &= v_i(\tau)k_i(\tau) && (2.8) \\ &= v_{f_i}\left(1 - \frac{k_i(\tau)}{k_{jam}}\right)k_i(\tau), && (2.9) \end{aligned}$$

in the case of (C2)

$$\begin{aligned} q_i(\tau) &= v_{i+1}(\tau)k_{i+1}(\tau) && (2.10) \\ &= v_{f_{i+1}}\left(1 - \frac{k_{i+1}(\tau)}{k_{jam}}\right)k_{i+1}(\tau). && (2.11) \end{aligned}$$

In the cases of (C3) and (C4) where $k_i(\tau)$ is larger 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, 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),

$$q_i(\tau) = \left(\frac{k_i(\tau) + k_{i+1}(\tau)}{2}\right)v_f(\tau)\left(1 - \frac{k_i(\tau) + k_{i+1}(\tau)}{2k_{jam}}\right). \qquad (2.12)$$

As the results, the flow model (2.8) $\sim$ (2.12) taking into account the discontinuity

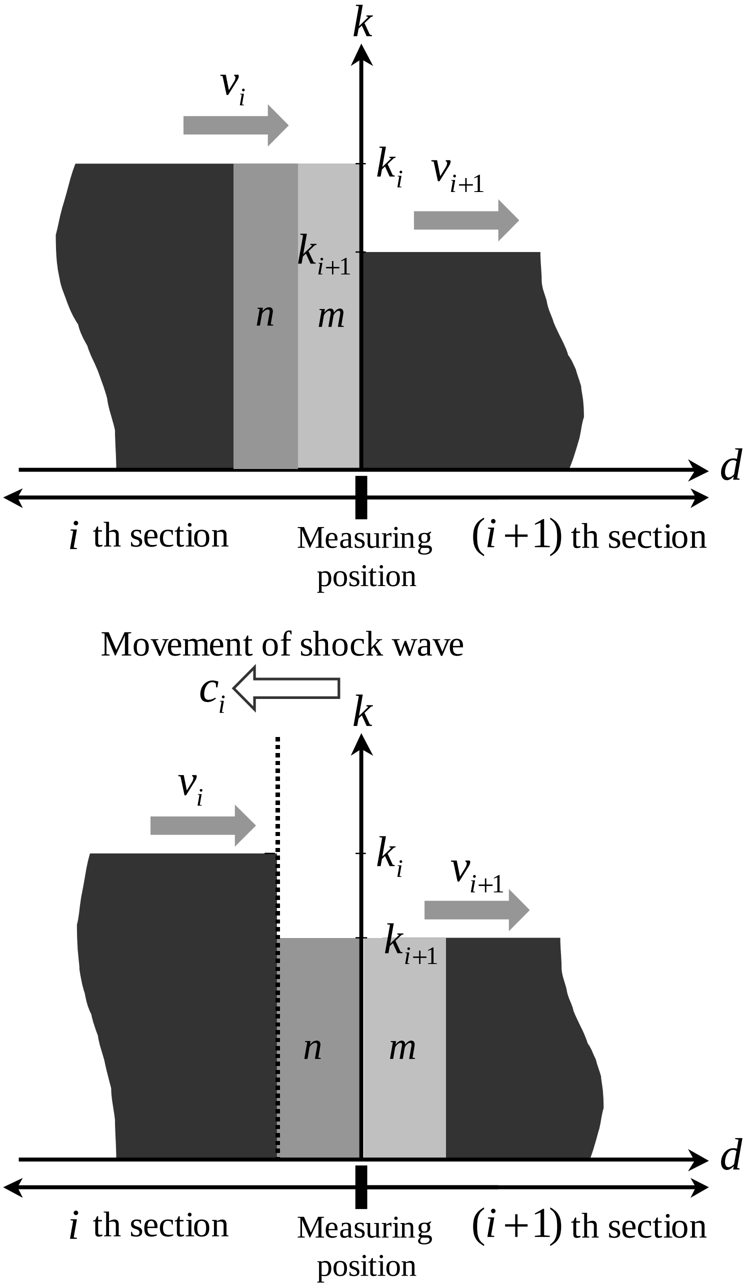

Figure 2.5: Movement of shock wave in the case of $k_i(\tau) \geq k_{i+1}(\tau)$

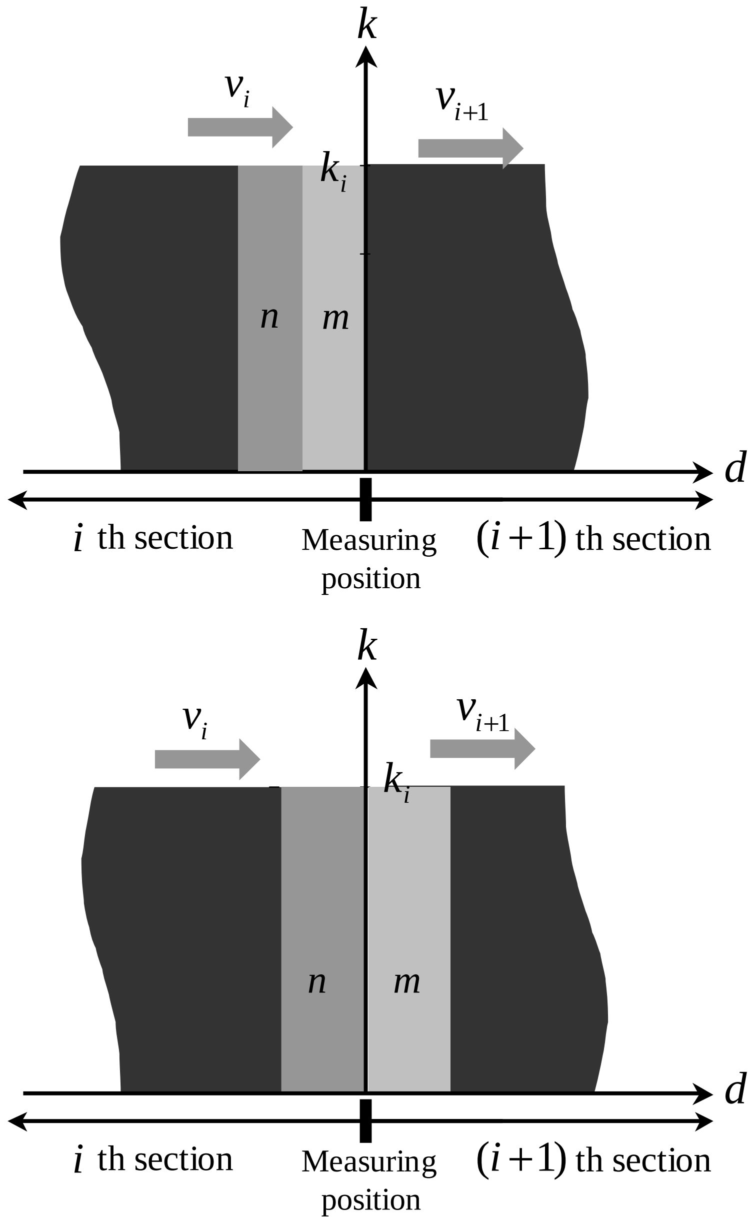

Figure 2.6: Movement of shock wave in the case of $k_i(\tau) = k_{i+1}(\tau)$

of the density can be summarized as follows:

$$q_i(\tau) = \begin{cases} \left(\frac{k_i(\tau)+k_{i+1}(\tau)}{2}\right) v_f \left(1 - \frac{k_i(\tau)+k_{i+1}(\tau)}{2k_{jam}}\right) \\ \quad if \;\; k_i(\tau) \geq k_{i+1}(\tau) \\ v_{f_i} \left(1 - \frac{k_i(\tau)}{k_{jam}}\right) k_i(\tau) \\ \quad if \;\; k_i(\tau) < k_{i+1}(\tau) \text{ and } c(\tau) > 0 \\ v_{f_{i+1}} \left(1 - \frac{k_{i+1}(\tau)}{k_{jam}}\right) k_{i+1}(\tau) \\ \quad if \;\; k_i(\tau) < k_{i+1}(\tau) \text{ and } c(\tau) \leq 0 \end{cases} . \tag{2.13}$$

2.2.3 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 [2]. Well-known traffic flow simulators such as TRANSIMS and MICROSIM are based on the CA model.

The essential property of the 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 the 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 the CA model can be expressed by

$$n_j(\tau+1) = n_j^{in}(\tau)(1 - n_j(\tau)) - n_j^{out}(\tau), \tag{2.14}$$

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 the 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 $n_j^{out}(\tau)$, three 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:

$$v_{\langle j\rangle}(\tau+\Delta\tau) \equiv v_{\langle j\rangle}(\tau) + v_{unit}. \tag{2.15}$$

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

$$v_{\langle j\rangle}(\tau+\Delta\tau) \equiv \min\{e, v_{\langle j\rangle}(\tau+\Delta\tau)\}. \tag{2.16}$$

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

$$v_{\langle j\rangle}(\tau+\Delta\tau) \equiv v_{\langle j\rangle}(\tau+\Delta\tau) - p\cdot v_{unit}. \tag{2.17}$$

Figure 2.7 shows the behavior of traffic flow obtained by applying the CA model to the two successive sections which 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.2.7 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.

The begging and ending of the CA network has an arbitrary input and output cars.

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 sections. 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 the CA model consists of quite simple procedures, it can show quite natural traffic flow behavior.

On the other hand, Fig.2.8 shows the behavior in case of using the HPN where proposed flow model given by (2.13) is embedded. We can see that Fig.2.8 shows the similar characteristics to Fig.2.7and that, especially, the saturation characteristics are 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 $k_{i+1}(\tau)$ is used to decide the flow $q_i(\tau)$ (i.e. use (2.12)) for all cases. Figure 2.9 shows the behavior in case of using the HPN where the flow model is supposed to be given by (2.12) 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 the CA,we need 140 seconds. On the other hand, in case of using the HPN and (2.13), we need only 0.06 seconds.

2.3 Transformation to mixed logical dynamical systems

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 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 be formalized generally as follows [35]:

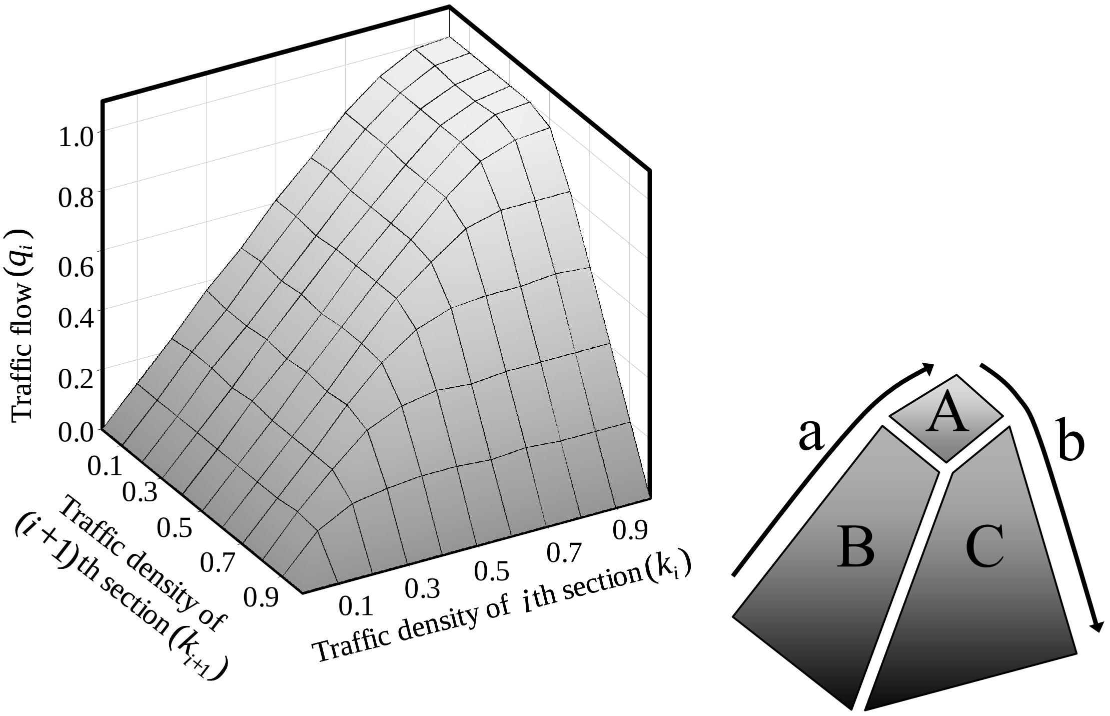

Figure 2.7: Traffic flow behavior obtained from the CA model

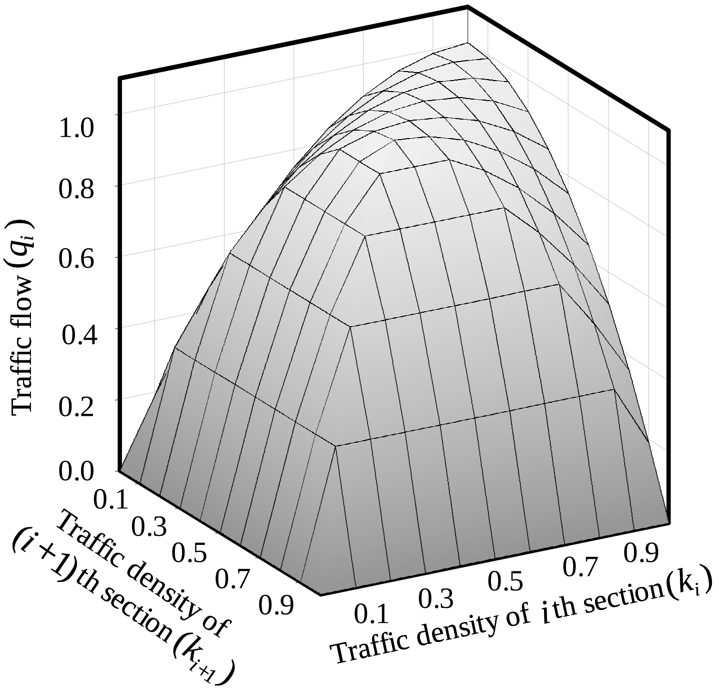

Figure 2.8: Traffic flow behavior obtained from the proposed traffic flow model

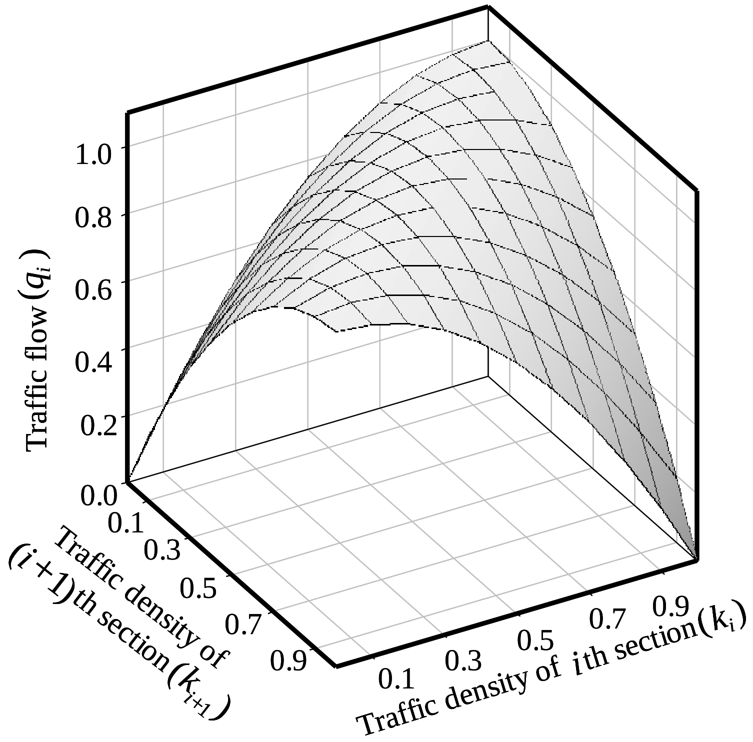

Figure 2.9: Traffic flow behavior obtained by averaging k_i and k_{i+1}

$$
\begin{aligned}
\boldsymbol{x}(\kappa+1) &= \boldsymbol{A_\kappa x}(\kappa)+\boldsymbol{B_{1\kappa}u}(\kappa)+\boldsymbol{B_{2\kappa}\delta}(\kappa)+\boldsymbol{B_{3\kappa}z}(\kappa), && (2.18)\\
\boldsymbol{y}(\kappa) &= \boldsymbol{C_\kappa x}(\kappa)+\boldsymbol{D_{1\kappa}u}(\kappa)+\boldsymbol{D_{2\kappa}\delta}(\kappa)+\boldsymbol{D_{3\kappa}}(\kappa), && (2.19)\\
\boldsymbol{E_{2\kappa}\delta}(\kappa)+\boldsymbol{E_{3\kappa}z}(\kappa) &\leq \boldsymbol{E_{1\kappa}u}(\kappa)+\boldsymbol{E_{4\kappa}x}(\kappa)+\boldsymbol{E_{5\kappa}}. && (2.20)
\end{aligned}
$$

In the MLDS form,κ represents the sampling index. Note that sampling period T_s is eliminated in the following. Equations (2.18), (2.19) and (2.20) are a state equation,an output equation and constraint inequality, respectively, where $x(\kappa)$, $y(\kappa)$ and $u(\kappa)$ are a state, an output and an input variable, whose components are represented 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 the HDS such as Piece-Wise Affine (PWA), Hybrid Automaton (HA) and so on.

In the TFCS represented by the HPN, equation (2.4) is directly transformed to the state equation in the MLDS form by regarding the continuous marking as the state variable. Also, the TFSC has only binary input variables which denote 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 (2.20) often plays an essential role to represent nonlinearity which exists in the original system. In the TFCS, the nonlinearity appears in (2.13). 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.2.7) can be approximated by the Piece-Wise Affine (PWA) model shown in the right figure of Fig.2.7, 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}.

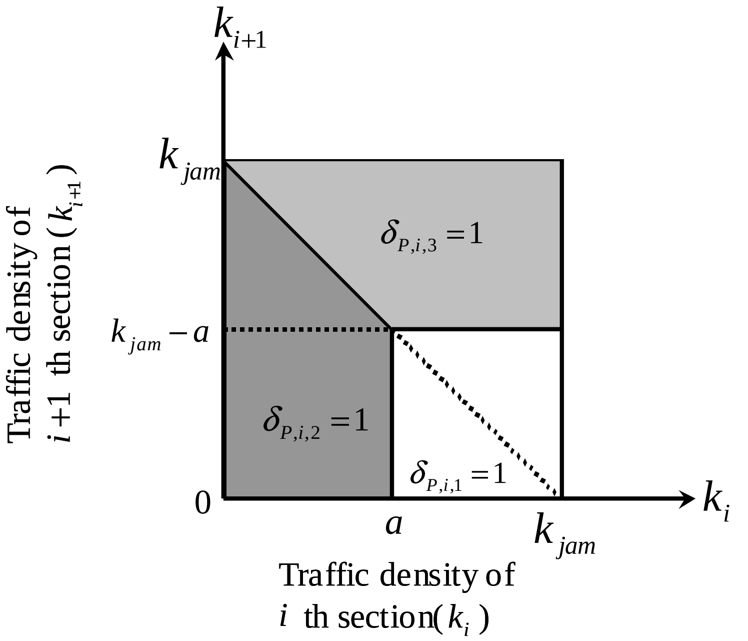

Figure 2.10: Division of flow model by introducing auxiliary variables

Figure 2.10 shows these partitions on $k_{i+1}(\kappa) - k_i(\kappa)$ plane. In order to derive the linear inequality 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:

$$[\delta_{P,i,1}(\kappa) = 1] \leftrightarrow \begin{cases} k_i(\kappa) & \geq a \\ k_{i+1}(\kappa) & \leq k_{jam} - a \end{cases}, \tag{2.21}$$

$$[\delta_{P,i,2}(\kappa) = 1] \leftrightarrow \begin{cases} k_i(\kappa) & \leq a - \varepsilon \\ k_i(\kappa) + k_{i+1}(\kappa) & \leq k_{jam} \end{cases}, \tag{2.22}$$

$$[\delta_{P,i,3}(\kappa) = 1] \leftrightarrow \begin{cases} k_{i+1}(\kappa) & \geq k_{jam} - a + \varepsilon \\ k_i(\kappa) + k_{i+1}(\kappa) & \geq k_{jam} + \varepsilon \end{cases}, \tag{2.23}$$

$$\delta_{P,i,1}(\kappa) + \delta_{P,i,2}(\kappa) + \delta_{P,i,3}(\kappa) = 1, \tag{2.24}$$

where ε is a small tolerance.

By using these binary variables, the flow model $q_i(\kappa)$ given by (2.13) can be rewritten in a compact linear form as follows:

$$\begin{aligned} q_i(\kappa) &= q_{max}\delta_{P,i,1}(\kappa) + \frac{q_{max}k_i(\kappa)}{a}\delta_{P,i,2}(\kappa) \\ &+\frac{q_{max}(1-k_{i+1}(\kappa))}{a}\delta_{P,i,3}(\kappa), \qquad (2.25) \\ \sum_{i=1}^{3} & \delta_{P,i,j}(\kappa) = 1, \end{aligned}$$

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 2.11 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 (2.21) to (2.23) can be generalized as follows:

$$[\delta_{P,i,j}(\kappa) = 1] \leftrightarrow \left[\begin{bmatrix} k_i(\kappa) \\ k_{i+1}(\kappa) \end{bmatrix} \in \Re_j\right], \qquad (2.26)$$

$$\Re_j = \left\{\begin{bmatrix} k_i(\kappa) \\ k_{i+1}(\kappa) \end{bmatrix} : \boldsymbol{S_j k_i}(\kappa) \leq \boldsymbol{T_j}\right\}, \qquad (2.27)$$

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.

$$\boldsymbol{S_j k_i}(\kappa) \quad - \quad \boldsymbol{T_j} \leq \boldsymbol{M_j}^*[1 - \delta_{P,i,j}(\kappa)] \qquad (2.28)$$

$$\boldsymbol{M_j^*} \quad \overset{\triangle}{=} \quad \max_{\boldsymbol{k_i} \in \Re_j} \boldsymbol{S_j k_i}(\kappa) - \boldsymbol{T_j} \qquad (2.29)$$

The flow $q_i(\kappa)$ of (2.25) can be represented by the vector form by using $\boldsymbol{\delta_{P,i}}(\kappa) = [\delta_{P,i,1}(\kappa)\ \delta_{P,i,2}(\kappa)\ \delta_{P,i,3}(\kappa)]$ as follows:

$$\begin{aligned} q_i(\kappa) &= f(\boldsymbol{\delta_{P,i}}(\kappa), \boldsymbol{k_{i+1}}(\kappa)) \qquad (2.30) \\ &= \sum_{j=1}^{3}(\boldsymbol{f_i^j}(\kappa)\boldsymbol{k_i}(\kappa) + h_i^j)\delta_{P,i,j}(\kappa), \qquad (2.31) \end{aligned}$$

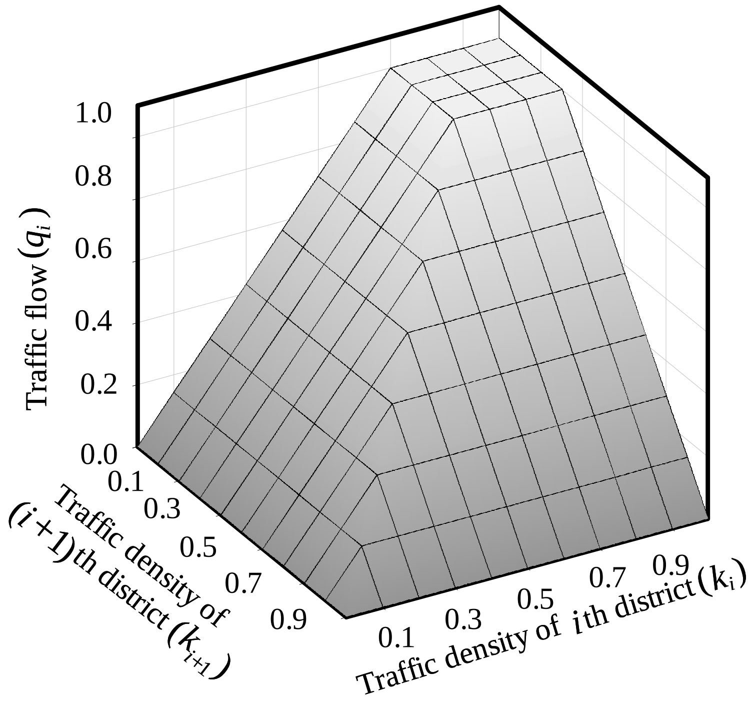

Figure 2.11: Traffic flow behavior obtained by PWA model

where $\boldsymbol{f_i^j}$ and h_i^j are given as follows (see Fig. 2.11):

$$\begin{aligned}
\boldsymbol{f_i^1} &= [\ 0 \quad 0\ ], & (2.32)\\
h_i^1 &= q_{max}, & (2.33)\\
\boldsymbol{f_i^2} &= [\ \tfrac{q_{max}}{a} \quad 0\ ], & (2.34)\\
h_i^2 &= 0, & (2.35)\\
\boldsymbol{f_i^3} &= [\ 0 \quad -\tfrac{q_{max}}{a}\ ], & (2.36)\\
h_i^3 &= \frac{q_{max}}{a}. & (2.37)
\end{aligned}$$

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

$$z_{i,j}(\kappa) = (\boldsymbol{f_i^j}(\kappa)\boldsymbol{k_i}(\kappa) + h_i^j)u_i(\kappa)\delta_{P,i,j}(\kappa). \tag{2.38}$$

That is,

$$q_i(\kappa)u_i(\kappa) = \sum_{j=1}^{3} z_{i,j}(\kappa), \tag{2.39}$$

where $u_i(\kappa) \in \{0, 1\}$ denotes the binary control input which represents the state of traffic signal. Then the equivalent inequalities to (2.38) are given as follows:

$$\begin{aligned}
z_{i,j}(\kappa) &\leq M_i u_i(\kappa)\delta_{P,i,j}(\kappa), & (2.40)\\
z_{i,j}(\kappa) &\geq m_i u_i(\kappa)\delta_{P,i,j}(\kappa), & (2.41)\\
z_{i,j}(\kappa) &\leq \boldsymbol{f_i^j}\boldsymbol{k_i}(\kappa) + h_i^j \\
&\quad -m_i(1 - u_i(\kappa)\delta_{P,i,j}(\kappa)), & (2.42)\\
z_{i,j}(\kappa) &\geq \boldsymbol{f_i^j}\boldsymbol{k_i}(\kappa) + h_i^j \\
&\quad -M_i(1 - u_i(\kappa)\delta_{P,i,j}(\kappa)), & (2.43)
\end{aligned}$$

where M_i and m_i are

$$\begin{aligned}
M_i &= \max_{\boldsymbol{k_i}(\kappa)\in\Re_j} \left\{\boldsymbol{f_i^j}\boldsymbol{k_i}(\kappa) + h_i^j\right\}, & (2.44)\\
m_i &= \min_{\boldsymbol{k_i}(\kappa)\in\Re_j} \left\{\boldsymbol{f_i^j}\boldsymbol{k_i}(\kappa) + h_i^j\right\}. & (2.45)
\end{aligned}$$

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:

$$-u_i(\kappa) + \delta_{M,i,j}(\kappa) \leq 0, \tag{2.46}$$

$$-\delta_{P,i,j}(\kappa) + \delta_{M,i,j}(\kappa) \leq 0, \tag{2.47}$$

$$u_i(\kappa) + \delta_{P,i,j}(\kappa) + \delta_{M,i,j}(\kappa) \leq 1. \tag{2.48}$$

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

$$\boldsymbol{x}(\kappa+1) = \boldsymbol{A}\boldsymbol{x}(\kappa) + \boldsymbol{B}\boldsymbol{z}(\kappa), \tag{2.49}$$

$$\boldsymbol{z}(\kappa) = diag(\boldsymbol{C}\boldsymbol{u}(\kappa))\boldsymbol{\delta}(\kappa), \tag{2.50}$$

$$\begin{aligned}\boldsymbol{E_2}\boldsymbol{\delta}(\kappa) &+ \boldsymbol{E_3}\boldsymbol{z}(\kappa)\\ &\leq \boldsymbol{E_1}\boldsymbol{u}(\kappa) + \boldsymbol{E_4}\boldsymbol{x}(\kappa) + \boldsymbol{E_5},\end{aligned} \tag{2.51}$$

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 and 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 $\boldsymbol{\delta}(\kappa)$=$[\boldsymbol{\delta_P}(\kappa),\ \boldsymbol{\delta_M}(\kappa)]^T$. 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.2.1 are described in the appendix A.

2.4 Model predictive control for traffic flow control system

The Model Predictive Control (MPC) [51][52] 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 the RHC, finite-horizon optimization is carried out based on the measured state at each sampling instant, and 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, ideas 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 constraints on these parameters (for example, constant offset), it can be embedded in our problem setup in a straightforward manner.

2.4.1 Receding horizon control for traffic flow control system

In the 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 (2.49) is modified to evaluate the state and input variables in the prediction horizon as follows:

$$\begin{aligned}\boldsymbol{x}(\kappa+\lambda|\kappa) \;=\; & \boldsymbol{A}^{\boldsymbol{\kappa}} x(\kappa) \\ & +\sum_{\eta=0}^{\lambda-1}\{\boldsymbol{A}^{\boldsymbol{\eta}}(\boldsymbol{B}(diag(\boldsymbol{C}\boldsymbol{u}(\kappa+\lambda-1-\eta|\kappa))) \\ & \qquad \cdot\boldsymbol{\delta}(\kappa+\lambda-1-\eta|\kappa))\}, \qquad (2.52)\end{aligned}$$

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 (2.49) 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.

$$\begin{aligned}
& J(\boldsymbol{u}(\kappa|\kappa), \cdots, \boldsymbol{u}(\kappa+N|\kappa) \\
& \quad , \boldsymbol{x}(\kappa|\kappa), \cdots, \boldsymbol{x}(\kappa+N|\kappa) \\
& \quad , \boldsymbol{\delta}(\kappa|\kappa), \cdots, \boldsymbol{\delta}(\kappa+N|\kappa)) \\
= & \sum_{\lambda=1}^{N} \Bigg\{ -\sum_i w_{1,i} \Bigg\{ \Bigg(\boldsymbol{\Theta_i} \left[\begin{array}{c} x_i(\kappa+\lambda|\kappa)/l_i \\ x_{i+1}(\kappa+\lambda|\kappa)/l_{i+1} \end{array} \right] \\
& \quad + \boldsymbol{\Phi_i} \Bigg)^T \boldsymbol{\delta_{M,i}}(\kappa+\lambda|\kappa) \Bigg\} \\
& \quad - \sum_i w_{2,i} \Big\{ 1 - \big| u_i(\kappa+\lambda|\kappa) - u_i(\kappa+\lambda+1|\kappa) \big| \Big\} \\
& \quad + \sum_i w_{3,i} \Bigg\{ \Bigg| \frac{x_i(\kappa+\lambda|\kappa)}{l_i} - \frac{x_{i+1}(\kappa+\lambda|\kappa)}{l_{i+1}} \Bigg| \Bigg\} \Bigg\}. \qquad (2.53)
\end{aligned}$$

where

$$\boldsymbol{\Theta_i} = \left[\begin{array}{cc} 0 & 0 \\ \frac{q_{max}}{a} & 0 \\ 0 & -\frac{q_{max}}{a} \end{array} \right], \qquad (2.54)$$

$$\boldsymbol{\Phi_i} = \left[\begin{array}{c} q_{max} \\ 0 \\ \frac{q_{max} x_i(\kappa)}{a l_i} \end{array} \right], \qquad (2.55)$$

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 $w_{1,i} + w_{2,i} + w_{3,i} = 1$, $0 \leq w_{1,i} \leq 1$, $0 \leq w_{2,i} \leq 1$ and $0 \leq w_{3,i} \leq 1$. The three terms in the left side of (2.53) correspond to the requirement (R1), (R2) and (R3), respectively.

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

$$
\begin{aligned}
&\textit{find} \\
&\quad \boldsymbol{\delta}(\kappa+\lambda|\kappa) = [\boldsymbol{\delta_P}(\kappa+\lambda|\kappa), \boldsymbol{\delta_M}(\kappa+\lambda|\kappa)]^T \\
&\quad (\lambda = 1, \cdots N) \\
&\quad \textit{which minimizes } (2.53) \\
&\textit{subject to } (2.28), (2.29), (2.32), (2.33), (2.34), (2.35), (2.36), (2.37), \\
&\qquad (2.39), (2.40), (2.41), (2.42), (2.43), (2.44), (2.45), (2.46). \\
&\qquad (2.47), (2.48), (2.49), (2.50), \textit{and } (2.51)
\end{aligned}
$$

The objective function (2.53) 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.

$$
\begin{aligned}
&J(\boldsymbol{u}(\kappa|\kappa), \cdots, \boldsymbol{u}(\kappa+N|\kappa) \\
&\quad , \boldsymbol{x}(\kappa|\kappa), \cdots, \boldsymbol{x}(\kappa+N|\kappa) \\
&\quad , \boldsymbol{\delta}(\kappa|\kappa), \cdots, \boldsymbol{\delta}(\kappa+N|\kappa)) \\
= \; & \sum_{\lambda=1}^{N} \Bigg\{ -\sum_i w_{1,i} \Bigg\{ \Bigg(\boldsymbol{\Theta_i} \begin{bmatrix} x_i(\kappa+\lambda|\kappa)/l_i \\ x_{i+1}(\kappa+\lambda|\kappa)/l_{i+1} \end{bmatrix} \\
&+ \boldsymbol{\Phi_i} \Bigg)^T \boldsymbol{\delta_{M,i}}(\kappa+\lambda|\kappa) \Bigg\} \\
&- \sum_i w_{2,i} \Big\{ 1 - \big(e^+_{u,i}(\kappa+\lambda|\kappa) + e^-_{u,i}(\kappa+\lambda|\kappa) \big) \Big\} \\
&+ \sum_i w_{3,i} \Big\{ \big(e^+_{x,i}(\kappa+\lambda|\kappa) + e^-_{x,i}(\kappa+\lambda|\kappa) \big) \Big\} \Bigg\},
\end{aligned}
\tag{2.56}
$$

where

$$
\begin{aligned}
e^+_{u,i}(\kappa+\lambda|\kappa) &- e^-_{u,i}(\kappa+\lambda|\kappa) \\
&= u_i(\kappa+\lambda|\kappa) - u_i(\kappa+\lambda+1|\kappa),
\end{aligned}
\tag{2.57}
$$

$$
\begin{aligned}
e^+_{x,i}(\kappa+\lambda|\kappa) &- e^-_{x,i}(\kappa+\lambda|\kappa) \\
&= \frac{x_i(\kappa+\lambda|\kappa)}{l_i} - \frac{x_{i+1}(\kappa+\lambda|\kappa)}{l_{i+1}},
\end{aligned}
\tag{2.58}
$$

$$e^{+}_{u,i}(\kappa+\lambda|\kappa) \geq 0 \quad , \quad e^{-}_{u,i}(\kappa+\lambda|\kappa) \geq 0, \tag{2.59}$$

$$e^{+}_{x,i}(\kappa+\lambda|\kappa) \geq 0 \quad , \quad e^{-}_{x,i}(\kappa+\lambda|\kappa) \geq 0. \tag{2.60}$$

The MLDS formulation coupled with the RHC scheme can be transformed to the canonical form of 0-1 the MILP problem with the objective function of (2.56). As a solver for the 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 the variables are fixed at zero or one. Then, Linear Programming (LP) method is applied to the sub-problem.

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 J may be specified by referring (2.26) and (2.27) automatically. This idea accelerates the B&B algorithm.

2.5 Numerical experiments

2.5.1 Signal control on straight road

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

Firstly, we consider the straight road which has two signals and is divided into five sections as shown in Fig.2.2. 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 $v_{max} = 5$ [2].

The obtained signal patterns and distributions of vehicles are listed in Table 2.1. 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 2.1, 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.

2.5.2 Signal control at intersections

In this subsection, we consider the signal control for the traffic network as shown in Figs. 2.12 and 2.13. 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 one. We have examined following five methods as follows:

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

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

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

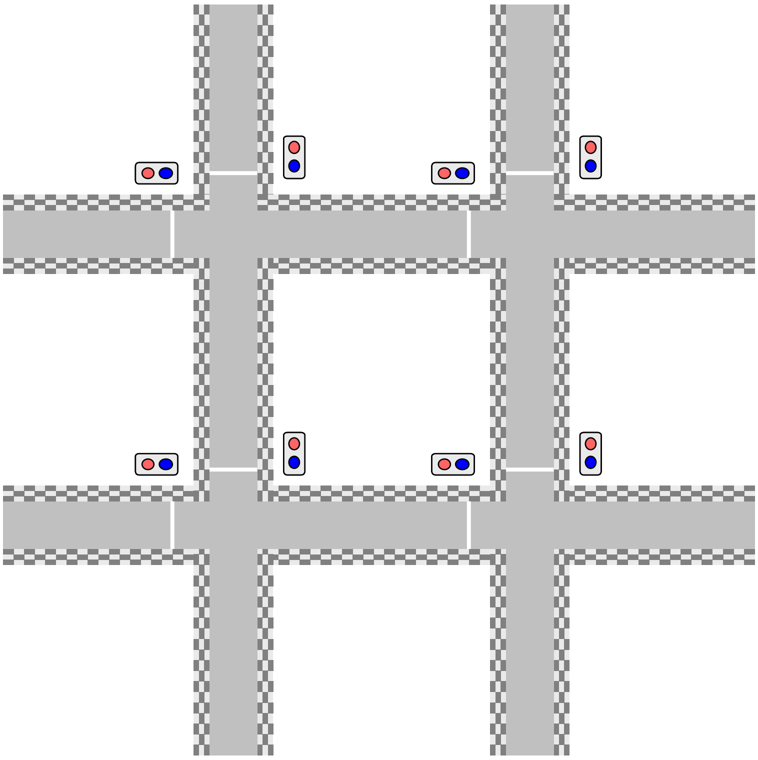

Figure 2.12: Traffic network with four single-way intersections

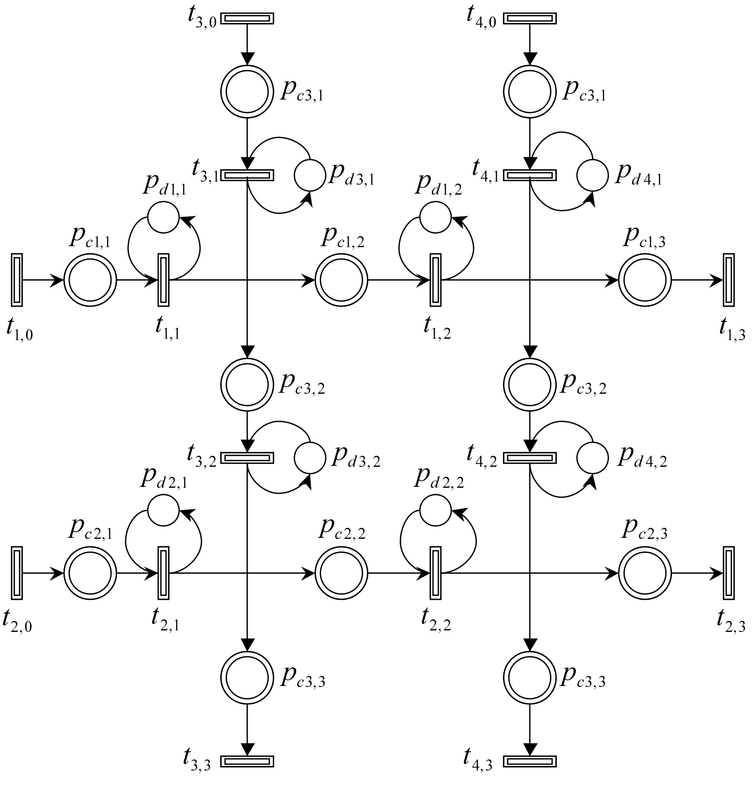

Figure 2.13: HPN of traffic network with four single-way intersections

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 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 2.2 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 2.2 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. 2.14.

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 (2.4). From Fig.2.14, we can see that the difference of the computational efforts between two schemes becomes larger with the increase of the horizon.

Table 2.1: 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

Table 2.2: 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

2.6 Conclusions

In this chapter, we have proposed a new method for traffic signal control based on the hybrid dynamical system theory. First of all, the synthetic modeling method for the Traffic Flow Control System (TFCS) has been proposed where the information on geometrical traffic network was modeled by using the Hybrid Petri Net (HPN), whereas the information on the behavior of traffic flow was modeled by means of the Mixed Logical Dynamical Systems (MLDS) form. The former allows us to easily apply our method to complicated 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 consider 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. 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.

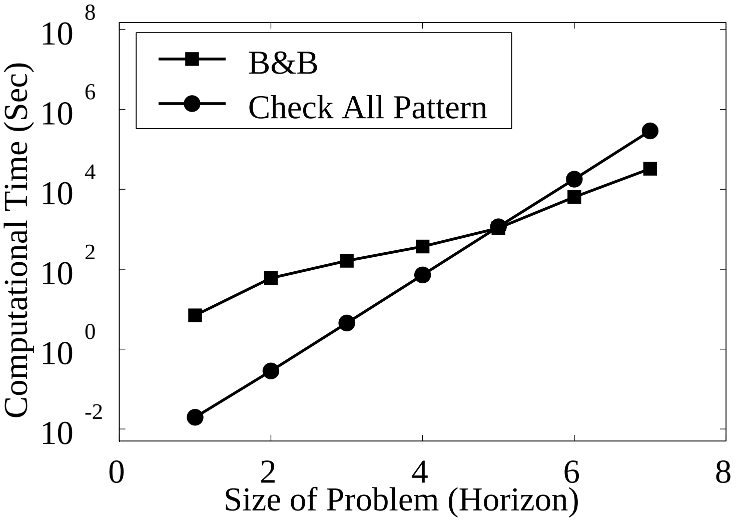

Figure 2.14: Computational efforts

Chapter 3

Traffic network control based on convex programming coupled with branch-and-bound strategy

3.1 Introduction

The author proposed the piece-wise affine traffic flow model in chapter 2, where the traffic flow was represented with the traffic densities of two consecutive districts in order to consider the behavior of shock wave. The traffic flow dynamics were optimized based on the Mixed Logical Dynamical System (MLDS) framework. The method used in chapter 2 is the well-established optimization procedure. However, the method based on the Mixed Integer Linear Programming (MILP) problem associated with piece-wise affine traffic flow dynamics is unfit for large-scale traffic network control, since it is computationally expensive. Consider the traffic light control of a pedestrian crossover on a one-way street. The previous method requires one binary variable (δ_S) to represent traffic light states, three binary variables (δ_P) to represent the traffic flow dynamics, and three binary variables (δ_M) to optimize the dynamics, transforming it to the linear form. This means in the worst case that MILP has 2^7 sub problems to solve.

In this chapter, a new method for large-scale traffic network control is proposed. First of all, we formulate, based on the Mixed Integer Non-Linear Programming (MINLP) problem, the traffic network control system that is one of typical hybrid systems with nonlinear dynamics. 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 the efficient method such as the Steepest Descent Method (SDM). We use in this chapter 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 the Model Predictive Control (MPC) policy. Note that the MLDS formulation often encounters multiplication of two decision variables, and that without modification, it cannot be directly applied to the 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 yields substantial computational amounts increased. A new method for this type of control problem is proposed. Although the system representation is nonlinear, the MPC policy is successfully applied by means of the proposed Branch and Bound (B&B) strategy. This implies we can find global optimal solution in a short time since no more auxiliary variables (such as δ_P and δ_M in chapter 2) are introduced.

This chapter is organized as follows. In section 3.2, we present the integrated model description for the large-scale traffic network control, where the geometrical information on the traffic network is characterized by using the Hybrid Petri Net (HPN) [48, 49], and the algebraic description of the traffic flow is provided in consideration of the shock wave. And then they are integrated into the the MLDS formulation. In section 3.3, the developed model is recast to the canonical form of the MINLP. And the proposed B&B algorithm coupled with convexity analysis is applied to the problem. Finally the usefulness of the proposed method is verified through numerical experiments.

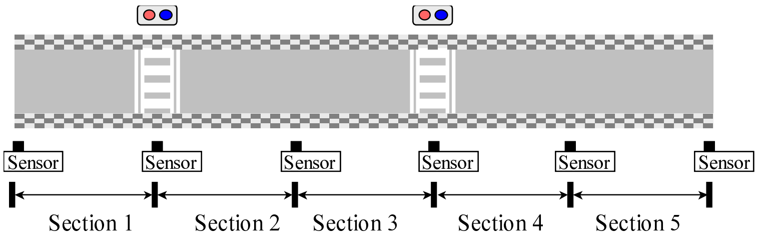

Figure 3.1: Straight road

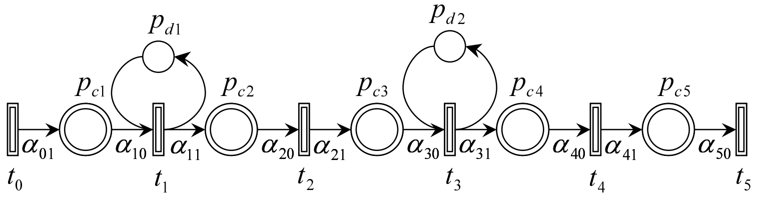

Figure 3.2: HPN model of straight road

3.2 Modeling of traffic flow control system based on the Hybrid Petri net

The Traffic Flow Control System (TFCS) is the collective entity of the traffic network, traffic flow and traffic lights. 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 presented, which provides both graphical and algebraic descriptions for the TFCS.

3.2.1 The Hybrid Petri net model of traffic network

The HPN is one of the useful tools to model and visualize the system behavior with both continuous and discrete variables. Figure 3.2 shows the HPN model for the road of Fig.3.1. In Fig.3.2, each section i of l_i-meter long constitutes the straight road, and two traffic lights are installed at the points of crosswalks. The HPN has a structure of $N = (P, T, q, I_+, I_-, M^0)$. The set of places P is partitioned into both 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 cars). Also, P_d represents the traffic light where green 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 light (i.e. binary value). Note that each light 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 cars passing through the boundary of two successive sections (measuring position) at time τ. The functions $I_{\pm}(p,t)$ are forward and backward incidence relationships between transition t and place p which connects the transition. The element of $I(p,t)$ is always 0 or 1. Finally, M^0 is specified as the initial marking of the place $p \in P$.

The net dynamics of the HPN is represented by a simple first order differential equation for each continuous place $p_{c_i} \in P_c$ as follows :

$$\frac{dm_{C,i}(\tau)}{dt} = \sum_{t_j \in p_{c_i}{}^\bullet \cup {}^\bullet p_{c_i}} I(p_{c_i}, t_j) \cdot q_j(\tau) \cdot m_{D,j}(\tau) \tag{3.1}$$

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 (3.1) is transformed to its discrete-time expression supposing that $q_j(\tau)$ is constant during two successive sampling instants as follows :

$$\begin{aligned} m_{C,i}((\kappa+1)T_s) &= m_{C,i}(\kappa T_s) \\ &+ \sum_{t_j \in p_{c_i}{}^\bullet \cup {}^\bullet p_{c_i}} I(p_{c_i}, t_j) \cdot q_j(\kappa T_s) \cdot m_{D,j}(\kappa T_s) \cdot T_s \end{aligned} \tag{3.2}$$

where κ and T_s are a sampling index and a period, respectively.

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.*

In the proposed modeling, processes of parallel processing, formal analysis, mutual exchange of shared resources, and synchronization are expressed with the Petri Net. If the Petri Net model is not introduced, the modeling is still possible but it is very difficult to express the processes. In our previous work in chapter 2, the Petri Net based model was proposed and it was shown to be very useful.

3.2.2 Traffic flow dynamics

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

$$q_i(\tau) = -\frac{(k_i(\tau) + k_j(\tau))}{2} \frac{v_i(k_i(\tau)) + v_j(k_j(\tau))}{2} \tag{3.3}$$

supposing that the density k_* and average velocity v_* of the flow in ith and $i + 1$th sections are almost identical. Then, by incorporating the velocity model

$$v_i(\tau) = \quad v_{f_i} \cdot \left(1 - \frac{k_i(\tau)}{k_{jam}}\right) \tag{3.4}$$

with eq.(3.3), 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 free velocity, that is, the velocity of the vehicle when no other car 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 lights. In order to treat the discontinuities of the density among neighboring sections (i.e. neighboring continuous places), the idea of 'shock wave'[53] is introduced as follows. Here, the movement of this boundary is called shock wave and the moving velocity of the shock wave c_i depends on the densities and average velocities of ith and jth sections as follows [53]:

$$c_i(\tau) \quad = \quad \frac{v_i(\tau)k_i(\tau) - v_j(\tau)k_j(\tau)}{k_i(\tau) - k_j(\tau)} \tag{3.5}$$

And these considerations lead to the following models:

$$q(k_i(\tau), k_j(\tau)) = \begin{cases} -\left(\frac{k_i(\tau)+k_j(\tau)}{2}\right) v_f \left(1 - \frac{k_i(\tau)+k_j(\tau)}{2k_{jam}}\right) \\ \quad if \;\; k_i(\tau) \geq k_j(\tau) \\ -v_{f_i}\left(1 - \frac{k_i(\tau)}{k_{jam}}\right) k_i(\tau) \\ \quad if \;\; k_i(\tau) < k_j(\tau) \text{ and } c_i(\tau) > 0 \\ -v_{f_j}\left(1 - \frac{k_j(\tau)}{k_{jam}}\right) k_j(\tau) \\ \quad if \;\; k_i(\tau) < k_j(\tau) \text{ and } c_i(\tau) \leq 0 \end{cases} \tag{3.6}$$

, where $0 \leq k_i(\tau) \leq k_{jam}, 0 \leq k_j(\tau) \leq k_{jam}$.

3.2.3 Traffic network model at an intersection

In this subsection, we develop traffic network model at an intersection. Figure 3.3 shows the HPN model of the ith intersection, where the notation for other than southward entrance lane is omitted. In Fig.3.3, $l_{i,IS}$ and $l_{i,ON}$ are the length of the districts $p_{C,i,IS}$ and $p_{C,i,ON}$, and the numbers of the vehicles at $p_{C,i,IS}$ and $p_{C,i,ON}$ are $k_{i,IS} \cdot l_{i,IS}$ and $k_{i,ON} \cdot l_{i,ON}$, respectively. The vehicles in $p_{C,i,IS}$ are assumed to have the probability $\zeta_{i,SW}$, $\zeta_{i,SN}$, and $\zeta_{i,SE}$ to proceed into the district corresponding to $p_{C,i,OW}$, $p_{C,i,ON}$, and $p_{C,i,OE}$ as follows,

$$k_{i,SW}(\tau) = k_{i,IS}(\tau)\zeta_{i,SW}, \tag{3.7}$$

$$k_{i,SN}(\tau) = k_{i,IS}(\tau)\zeta_{i,SN}, \tag{3.8}$$

$$k_{i,SE}(\tau) = k_{i,IS}(\tau)\zeta_{i,SE}. \tag{3.9}$$

Note that these probabilities are determined by the traffic network structure, and satisfy $0 \leq \zeta_{i,SW}(\tau) \leq 1$, $0 \leq \zeta_{i,SN}(\tau) \leq 1$, $0 \leq \zeta_{i,SE}(\tau) \leq 1$, and $\zeta_{i,SW}(\tau)+\zeta_{i,SN}(\tau)+\zeta_{i,SE}(\tau) = 1$. Therefore, the traffic flows of the three directions are represented by

$$q\left(k_{i,SN}(\tau), k_{i,ON}(\tau)\right), \tag{3.10}$$

$$q\left(k_{i,SW}(\tau), k_{i,OW}(\tau)\right), \tag{3.11}$$

$$q\left(k_{i,SE}(\tau), k_{i,OE}(\tau)\right). \tag{3.12}$$

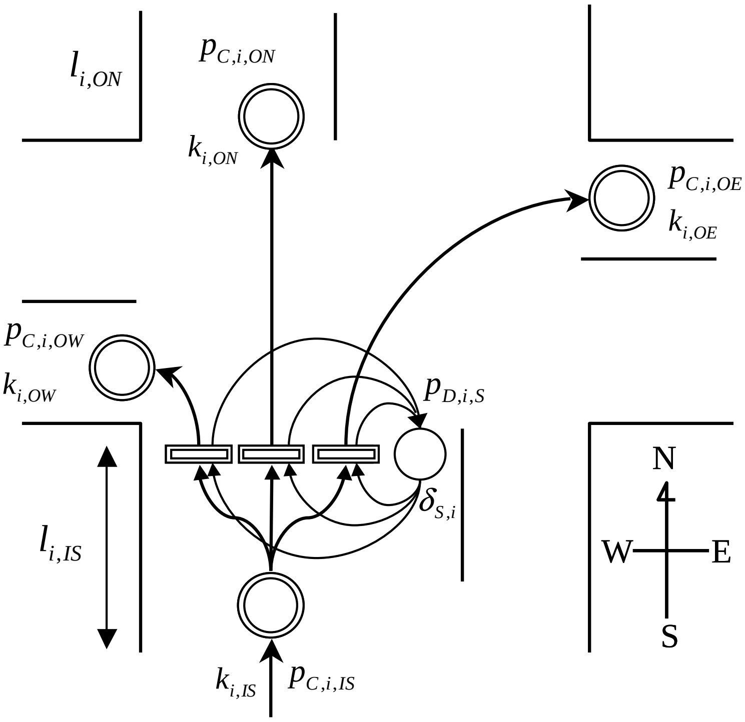

Figure 3.3: HPN model of the intersection

Since the probability ζ includes the affection of yellow light, yellow light is not explicitly represented in Fig. 3.3.

3.3 Model predictive control of traffic network control system

In this section, the MLDS form is introduced to formulate the Model Predictive Control (MPC) as shown in the next subsection. The MLDS form can generally be formalized as follows [35]:

$$\begin{aligned}
\boldsymbol{x}(\kappa+1) &= \boldsymbol{A_\kappa x}(\kappa) + \boldsymbol{B_{1\kappa} u}(\kappa) + \boldsymbol{B_{2\kappa} \delta}(\kappa) + \boldsymbol{B_{3\kappa} z}(\kappa), && (3.13)\\
\boldsymbol{y}(\kappa) &= \boldsymbol{C_\kappa x}(\kappa) + \boldsymbol{D_{1\kappa} u}(\kappa) + \boldsymbol{D_{2\kappa} \delta}(\kappa) + \boldsymbol{D_{3\kappa}}(\kappa), && (3.14)\\
\boldsymbol{E_{2\kappa} \delta}(\kappa) + \boldsymbol{E_{3\kappa} z}(\kappa) &\leq \boldsymbol{E_{1\kappa} u}(\kappa) + \boldsymbol{E_{4\kappa} x}(\kappa) + \boldsymbol{E_{5\kappa}}. && (3.15)
\end{aligned}$$

, where κ represents the sampling index. [1] The equations (3.13), (3.14) and (3.15) are a state equation, an output equation and a constraint inequality, respectively, where $x(\kappa)$, $y(\kappa)$ and $u(\kappa)$ are a state, a output and an 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, respectively

Although the MLDS description allows us to represent systematically nonlinear behavior of the system dynamics by introducing auxiliary variables to approximate it to the (piece-wise) linear dynamics, methods for finding optimal solution were computationally expensive. This is because of a number of auxiliary variables introduced. A new method for optimizing the system with nonlinear dynamics is proposed, where nonlinear dynamics is not linearized in a piece-wise manner. The developed Mixed Integer Non-Linear Programming (MINLP) problem is solved by proposed Branch & Bound strategy.

[1] Note that, for simplicity, the sampling period T_s is eliminated from the description of the MLDS form.

3.3.1 Mixed logical dynamical system formulation

The traffic flow q is the function of k_i, k_{i+1} which are the traffic densities of the two consecutive district i and $i+1$, and contains nonlinearity because of the multiplications of the two variables as in (3.6). Since q is also the function of traffic light, traffic flow adjoining the intersection i can be represented by introducing (continuous) auxiliary variable z as follows

$$z_{i,\tilde{i}}(\kappa) \;=\; q\Big(\frac{x_i(\kappa)}{l_i}, \frac{x_{\tilde{i}}(\kappa)}{l_{\tilde{i}}}\Big)u_{i+1}(\kappa) \tag{3.16}$$

, where the traffic light at intersection i is $u_i(\kappa)$. Here, $BLUE(RED)$ light of the north-south orientation and $RED(BLUE)$ light of the east-west orientation are represented by 0 (1). With z of (3.16), the state equation and the constraint inequality are formulated as follows

$$\boldsymbol{x}\ (\kappa+1) \;=\; \boldsymbol{Ax}\ (\kappa) + \boldsymbol{Bz}(\kappa) \tag{3.17}$$

$$\boldsymbol{E_{2\kappa}z}(\kappa) \;\leq\; \boldsymbol{E_{1\kappa}u}(\kappa) + \boldsymbol{E_{4\kappa}x}(\kappa) + \boldsymbol{E_{5\kappa}} \tag{3.18}$$

where $\boldsymbol{x} = [x_1, x_2, ..., x_n]^T$,$u_j \in \{0,1\}$,$\tilde{i} \in t_j\bullet$ and $\boldsymbol{A}$ is the matrix with the suitable dimension. Equation (3.18) represents the logical relationship (proposition) of eq.(3.6) and the maximum capacity constraints of each place, and the firing speed of each transition respectively.

Although the Mixed Logical Dynamical System is represented in a compact form as (3.17) and (3.18), this cannot be directly applied to the model predictive control scheme, since z has the multiplication of three decision variables (z amounts to α $u \cdot x^2$).

However, if we know the values of k_i and k_j, q is uniquely determined. We propose in this chapter a new B&B algorithm which consists of refining process as well as conventional B&B processes. The combination of the three processes makes it easy to handle the special type of the nonlinear programming problem.

3.3.2 Model predictive control coupled with branch and bound strategy

The Model Predictive Control (MPC) [51, 52] is one of the well-known paradigms for optimizing the systems with constraints and uncertainties. In the MPC policy, the control input at each sampling instant is decided based on the prediction of the behavior for the next several sampling periods called the prediction horizon. In order to formulate the optimization procedure, firstly, equation (3.17) is modified to evaluate the state and input variables in the prediction horizon as follows:

$$\begin{aligned}\boldsymbol{x}(\kappa+\lambda|\kappa) &= \boldsymbol{A^{\lambda}}x(\kappa)\\ &+\sum_{\eta=0}^{\lambda-1}\{\boldsymbol{A^{\eta}}(\boldsymbol{Bz}(\kappa+\lambda-1-\eta|\kappa)\cdot\boldsymbol{u}(\kappa+\lambda-1-\eta|\kappa))\},\end{aligned} \tag{3.19}$$

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 (3.19) starting from the state $x(\kappa|\kappa)=x(\kappa)$.

The MPC scheme with the MLDS formulation can be transformed to the canonical form of 0-1 MINLP problem. We propose a new B&B algorithm to solve this class of programming problem, since the conventional method is not applicable for MINLP problem. The proposed method guarantees for solution optimality based on convexity analysis coupled with the proposed B&B strategy. The proposed algorithm is the combination of branching process, bounding process, and refining process as follows.

Branching Process: If the solution to the given sub-problem is proven to satisfy 0-1 constraints, the algorithm constructs two new sub-problems, in which variables are fixed at zero or one. Then, conventional nonlinear programming (NLP) method is applied to the problem.

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 (which was proven to satisfy all 0-1 constraints).

Refining Process: If the constraint equations and/or inequalities have the nonlinear terms of known variables, these terms are reformed to have a linear form of the branching variables. This process is carried out by assigning the values of the known variables to the nonlinear terms.

By introducing the refining process, the nonlinear function $\boldsymbol{z}(\kappa+i)$ in (3.19) comes to have a linear form. However, if we apply the MPC scheme with the MINLP problem, the selection of the branching procedure should be carefully carried out. For example, $x(\kappa+1)$ can be easily obtained since q is a known variable at the time of κ. However, enumeration of $x(\kappa+2)$ contains the multiplication of variables as follows,

$$\begin{aligned}\boldsymbol{x}(\kappa+2|\kappa) &= \boldsymbol{A}(\boldsymbol{A}\boldsymbol{x}(\kappa|\kappa)+\boldsymbol{B}\boldsymbol{z}(\kappa|\kappa)\boldsymbol{u}(\kappa|\kappa))+\boldsymbol{B}\boldsymbol{z}(\kappa+1|\kappa)\boldsymbol{u}(\kappa+1|\kappa) \quad (3.20)\\ &= \boldsymbol{A}^2\boldsymbol{x}(\kappa|\kappa)+\boldsymbol{A}\boldsymbol{B}\boldsymbol{z}(\kappa|\kappa)\boldsymbol{u}(\kappa|\kappa)+\boldsymbol{B}\boldsymbol{z}(\kappa+1|\kappa)\boldsymbol{u}(\kappa+1|\kappa). \quad (3.21)\end{aligned}$$

Therefore the refining process (RP) and the branching process (BP) should be in the following order ; RP of $q(\kappa)$ → BP of $u(\kappa)$ → RP of $q(\kappa+1)$ → BP of $u(\kappa+1)$ and so on.

The proposed algorithm is formulated as follows.

Modified Branch-and-Bound algorithm

Step 1(Initialization) Set List $L \equiv \{P_0\}$, $\xi^* \equiv \infty$, and $l=0$. Here, P_0 is the problem in which all 0-1 constraints are relaxed.

Step 2(Optimality Assessment) If $L=\phi$, terminate the algorithm. Here, if $\xi^* < \infty$, the solution corresponding to ξ^* is the optimal solution. Otherwise, there is no feasible solution.

Step 3(Selection of Sub-Problem) Select sub-problem P_k from the list L and substitute L with $L - \{P_k\}$.

Step 4(Bounding Process) Solve P_k. If P_k has no feasible solution, go to Step 2. If P_k has feasible solution with $\xi^k \geq \xi^*$, go to Step 2. If P_k has feasible solution with $\xi^k < \xi^*$, go to Step 5.

Step 5(Renewal of Incumbent Solution) If the solution to P_k satisfies all 0-1 constraints, substitute $\boldsymbol{x}^*$ with $\boldsymbol{x}^k$, $\xi^* \equiv \xi^k$, and go to Step 2.

Step 6(Selection of Branching Variable) If the solution to P_k violates at least one of 0-1 variables, set N whose elements are 0-1 variables, but they do not satisfy 0-1 constraints yet. Select the branching variable x_s^k whose predicted sampling index is closest to the present sampling index among N.

Step 7(Branching Process) Generate two new sub-problems P_{l+1} and P_{l+2}. Impose the constraints, $x_x^k = 0$ on P_{l+1} and $x_x^k = 1$ on P_{l+2}, respectively. Substitute L with $L \cup \{P_{l+1}, P_{l+2}\}$ and l with $l + 2$.

Step 8(Refining Process) If there is any decision variable which is dependent on x_s^k, substitute the value of x_s^k to the variable, and reformulate P_{l+1} and P_{l+2} in a linear form. And go to Step 2.

By selecting the branching variable x_s^k whose predicted sampling index is closest to the present sampling index, all the dependent variables of x_s^k can be represented as the numerical form with the lower order. In our problem setting, the refining process of Step 8 can be translated as follows.

Step 8-1 Obtain $k(\Gamma)$ from the value of $x(\Gamma)$.

Step 8-2 Obtain $q(\Gamma)$ by substituting $k(\Gamma)$ to the equation (3.6).

Step 8-3 Obtain $z(\Gamma)$ by substituting $q(\Gamma)$ and $u(\Gamma)$ to the equation (3.16).

Step 8-4 Reformulate the sub-problem in a linear form.

Here, the decision variable $u(\Gamma)$ is selected as the branching variable and nonlinear term $z(\Gamma)$ is obtained based on the known variables $q(\Gamma)$ and $u(\Gamma)$. If all the multivariate nonlinear terms can be transformed to have first order or zero order form of unknown variables by applying refining process, you can conceal nonlinear constraints from the problem setup. Consider Step 8-1 to Step 8-4, where by substituting the value of the branching variable $u(\Gamma)$, $z(\Gamma)$ comes to have zero order form of the unknown variables. This implies that we do not need to introduce the auxiliary (binary) variables in chapter 2 in order to represent the logical relations between the three modes of (3.6) and corresponding dynamics, respectively, and to associate with optimization scheme.

Note that the bounding process is very important to reduce the problem size to a computationally manageable one. However this process is effective only when the performance criterion is a convex function. In the next section, the convexity analysis is applied to our problem setup.

3.4 Convexity analysis

The problem we formulated in the previous section is recast to the convex programming problem in this section. The convex programming problem, where the constraint and objective functions are convex, has become quite popular recently for a number of reasons. The 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.

3.4.1 Performance criteria

In this subsection we firstly 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 section: (1) maximization

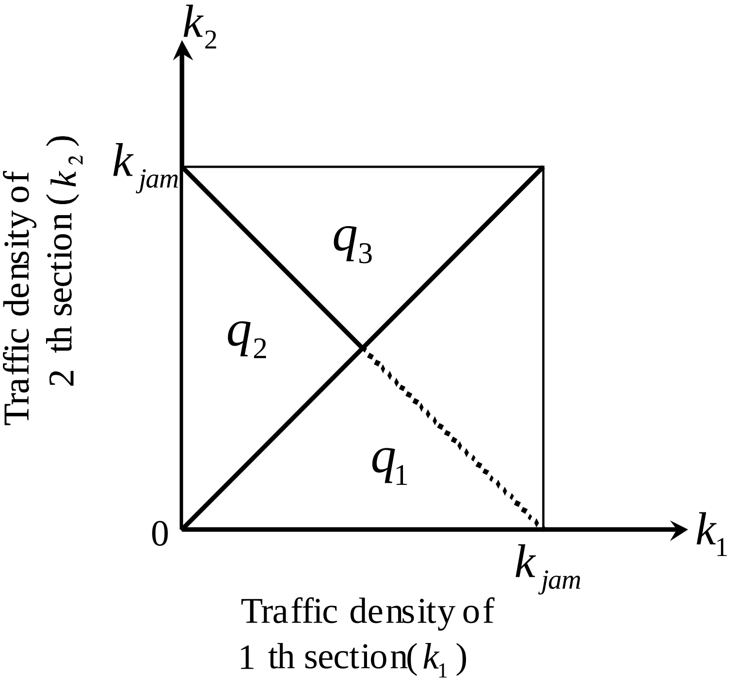
k_2
k_{jam}
Traffic density of 2 th section (k_2)
q_3
q_2
q_1
0
k_1
k_{jam}
Traffic density of 1 th section(k_1)

Figure 3.4: Assignation of traffic flow mode

of traffic flow and (2) minimization of traffic density difference between neighboring districts. These criteria are numerically represented as follows,

$$f = \sum_{i=n+1}^{n+m} z_i, \tag{3.22}$$

and

$$f = \sum_{i=0}^{n-1} |x_i - x_{i+1}|. \tag{3.23}$$

In order to verify the convexity of (3.22), we firstly 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.

Consider Fig.(3.4), 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,

$$\begin{aligned} \nabla^2 q_1(\boldsymbol{x}) &= \left[\frac{\partial^2 q_1(\boldsymbol{x})}{\partial x_1 \partial x_2}\right] \\ &= \begin{bmatrix} \frac{v_f}{2x_{jam}} & \frac{v_f}{2x_{jam}} \\ \frac{v_f}{2x_{jam}} & \frac{v_f}{2x_{jam}} \end{bmatrix} \geq 0, \end{aligned} \tag{3.24}$$

$$\nabla^2 q_2(\boldsymbol{x}) = \left[\frac{\partial^2 q_2(\boldsymbol{x})}{\partial x_1 \partial x_2}\right] = \begin{bmatrix} \frac{v_f}{2x_{jam}} & 0 \\ 0 & 0 \end{bmatrix} \geq 0, \tag{3.25}$$

and

$$\begin{aligned} \nabla^2 q_3(\boldsymbol{x}) &= \left[\frac{\partial^2 q_3(\boldsymbol{x})}{\partial x_1 \partial x_2}\right] \\ &= \begin{bmatrix} 0 & 0 \\ 0 & \frac{v_f}{2x_{jam}} \end{bmatrix} \geq 0, \end{aligned} \tag{3.26}$$

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 $\forall\ i$, γ and μ

$$\nabla_\gamma D_1(\Psi)\Big|_{\psi_i=\hat{\psi_i}} \leq (\geq) \nabla_\gamma D_2(\Psi)\Big|_{\psi_i=\hat{\psi_i}}, \tag{3.27}$$

then

$$\nabla^2_{\mu,\mu} D_1(\Psi)\Big|_{\psi_i=\hat{\psi_i}} \leq (\geq) \nabla^2_{\mu,\mu} D_2(\Psi)\Big|_{\psi_i=\hat{\psi_i}}, \tag{3.28}$$

where over-line denote the closure of the set, $1 \leq i, \gamma, \mu \leq n$, $\nabla_\gamma D$ is the γth element of ∇D, and $\nabla^2_{\mu,\mu} D$ is the (μ, μ)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,

$$\begin{aligned} q_1(k_1(\tau), k_2(\tau)) &= q_2(k_1(\tau), k_2(\tau)) && (3.29) \\ &= q_3(k_1(\tau), k_2(\tau)) && (3.30) \\ &= k(\tau) v_f \Big(1 - \frac{k(\tau)}{k_{jam}}\Big). && (3.31) \end{aligned}$$

Lastly, with following (3.32) to (3.35),

$$\begin{aligned} \nabla q_1(\boldsymbol{x})\Big|_{x_1=\hat{x_1}} &= \Big[\frac{v_f}{k_j} k - \frac{v_f}{2}, \frac{v_f}{k_j} k - \frac{v_f}{2}\Big] && (3.32) \\ \nabla^2 q_1(\boldsymbol{x})\Big|_{x_1=\hat{x_1}} &= \begin{bmatrix} \frac{v_f}{2x_{jam}} & \frac{v_f}{2x_{jam}} \\ \frac{v_f}{2x_{jam}} & \frac{v_f}{2x_{jam}} \end{bmatrix} && (3.33) \\ \nabla q_2(\boldsymbol{x})\Big|_{x_1=\hat{x_1}} &= \Big[2v_f \frac{k}{k_{jam}} - v_f, 0\Big] && (3.34) \\ \nabla^2 q_2(\boldsymbol{x})\Big|_{x_1=\hat{x_1}} &= \begin{bmatrix} \frac{v_f}{2x_{jam}} & 0 \\ 0 & 0 \end{bmatrix}, && (3.35) \end{aligned}$$

the convexity condition of lemma 1 was satisfied, since

$$\nabla_1 q_1(x) \leq \nabla_1 q_2(x) \qquad (3.36)$$

in pair with

$$\nabla_1^2 q_1(x) \leq \nabla_1^2 q_2(x). \qquad (3.37)$$

In the same way,

$$\nabla_1 q_2(x) \leq \nabla_1 q_3(x) \quad , \quad \nabla_1 q_1(x) \leq \nabla_1 q_3(x) \qquad (3.38)$$

are satisfied , paired together with

$$\nabla_1^2 q_2(x) \leq \nabla_1^2 q_3(x) \quad , \quad \nabla_1^2 q_1(x) \leq \nabla_1^2 q_3(x). \qquad (3.39)$$

Therefore, the convexity of overall dynamics are confirmed.

Note that although z is the multiplication of q and u, the performance criteria (3.22) 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 (3.22) 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

$$\begin{aligned} e_x^+ &\geq 0 && (3.40)\\ e_x^- &\geq 0 && (3.41)\\ e_x^+ - e_x^- &= x_1 - x_2, && (3.42) \end{aligned}$$

where e_x^+ and e_x^- are equivalently

$$\begin{aligned} e_x^+ &= \frac{(x_1 - x_2) + |x_1 - x_2|}{2} && (3.43)\\ e_x^- &= \frac{-(x_1 - x_2) + |x_1 - x_2|}{2}. && (3.44) \end{aligned}$$

Since all the constraints are described in the form of (3.18), the problems (3.22) and (3.23) are included in the class of the convex programming problem.

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

$$\begin{aligned} \text{minimize} \quad & F(\boldsymbol{x}, r) = f(\boldsymbol{x}) + r\boldsymbol{P}(\boldsymbol{x}) && (3.45)\\ \boldsymbol{P}(\boldsymbol{x}) \quad & \begin{cases} = 0, & \boldsymbol{x} \in X \\ > 0, & \boldsymbol{x} \notin X \end{cases}, && (3.46) \end{aligned}$$

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 (3.46).

If we can select the feasible initial solution, the optimal solution would be found in a short time. In this chapter, 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 :

$$\begin{aligned} & l_i k_i(\kappa+1) - l_i k_i(\tau) \\ = \; & -\boldsymbol{q}(k_{i-1}(\kappa), k_i(\kappa))T_s + \boldsymbol{q}(k_i(\kappa), k_{i+1}(\kappa))T_s. \end{aligned} \tag{3.47}$$

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

$$l_i k_i(\tau) \;\geq\; -\boldsymbol{q}(k_i(\tau), k_{i+1}(\tau))T_s \tag{3.48}$$

$$l_i k_{jam} - l_i k_i(\tau) \;\leq\; \boldsymbol{q}(k_{i-1}(\tau), k_i(\tau))T_s. \tag{3.49}$$

By substituting q of (3.48) to (3.6), following inequality is obtained from the both $k_i(\kappa) \geq k_{i+1}(\kappa)$ and $k_i(\kappa) < k_{i+1}(\kappa)$.

$$1 \;\geq\; \frac{v_f}{l_i}\left(1 - \frac{k_i}{k_{jam}}\right)T_s. \tag{3.50}$$

Since $\frac{v_f T_s}{l_i} \ll 1$, (3.48) can be easily confirmed. In the similar way, the condition (3.49) can be easily confirmed.

3.5 Numerical experiments

3.5.1 Numerical environments

In order to show the usefulness of our proposed method, we show, in this section, the numerical experiments. We considered the traffic network of Fig. 3.5, where the square network with 1000×1000 $[m^2]$ 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. It is known that the cellular automaton model (CA model) can simulate the real traffic flow with great granularity [2], although it takes too much computation time. Therefore, the CA model was constructed based on [2] to show feasibility of the proposed model. In our previous work in chapter 2, the model proposed in the paper was compared with the CA model to confirm the feasibility of the model. Each controller is assigned to the network with 500×500 $[m^2]$. 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 chapter are as follows; $\boldsymbol{x} \in \Re^{56}$, $\boldsymbol{q} \in \Re^{80}$, $\boldsymbol{\delta} \in \{0, 1\}^4$.

3.5.2 Traffic flow control system for traffic network

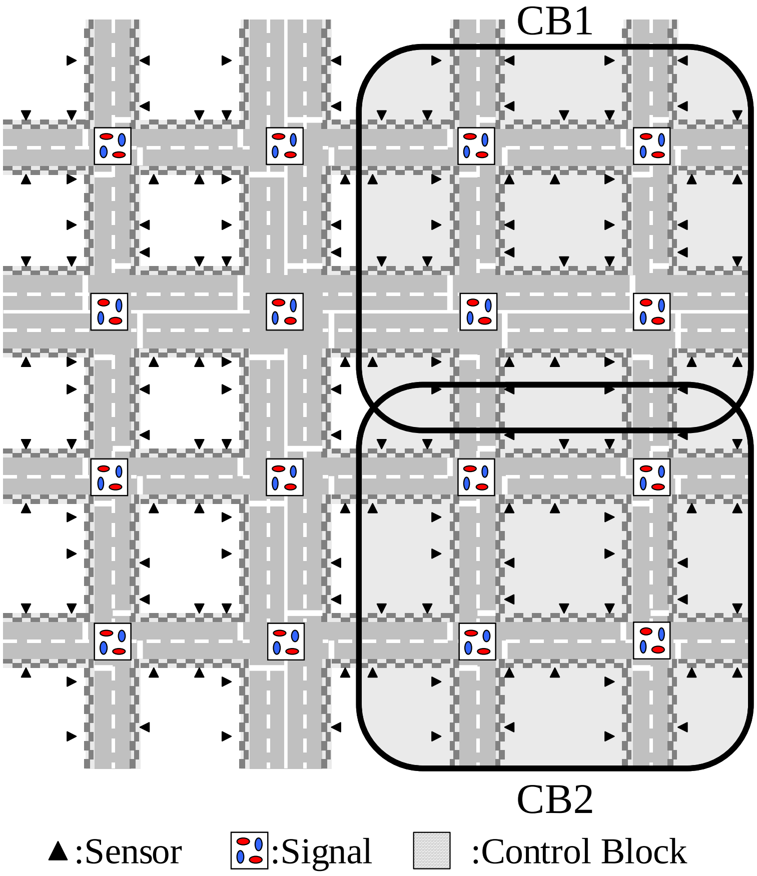

Figure 3.5: Traffic network

We show the results obtained by applying our proposed methods in Table 3.1, where H denotes the length of the prediction horizon, 'No Control' implies that the traffic light is changed at every 30 seconds, and

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

B: Average computation time,

C: Average number of the sub-problem generated.

From the results in Table 3.1, 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.

3.5.3 Comparison of computational amount

In order to evaluate the computational amount of the proposed method, we compare in Table 3.2 computational times obtained by applying our proposed method and conventional method in chapter 2. We used Athlon XP 2400+ Windows 2000 for this experiments. In Table 3.2, A implies the introduced number of δ, B implies the computation time, and C implies the number of cars passed during the corresponding sampling interval. Note that our proposed method finds better solution with a shorter time. This is because the proposed method does not approximate nonlinear dynamics in chapter 2 and solves non-linear programming problem, reformulating it to the convex programming problem. Furthermore, the proposed refining process enables to eliminate the introduction of auxiliary variables that without them, the logical relation between k_i, k_j, c_i, and q in (3.6) is well formulated and optimized.

3.5.4 Traffic flow control system for large-scale traffic network

In this subsection, the effectiveness of our proposed 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

Table 3.1: Numerical experimental result WRT H

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

Table 3.2: Comparison of the computational efforts

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

Table 3.3: Experimental result in case of no arterial road

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

amount would be fairly enormous. The proposed method, as in Fig. 3.5, designates the control block which groups some traffic lights in order that the feasible solutions may be obtained during the sampling interval. Fig. 3.5 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 3.3 and Table 3.4 shows the obtained solutions by applying the proposed 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. The results in both cases show that the proposed method has good solution in both cases. Note that our method has always better solutions than the cases of No Control.

3.5.5 Traffic flow control system with traffic accident

In this subsection, we investigate the adaptability of the proposed method in case of abrupt traffic congestion. Consider the Fig.3.6,and Fig.3.7, where arterial road is curved from south side to left side. The maximum speed limitations are 80 km/h on the arterial road and 60 km/h on the other road respectively. The considered traffic situation is as follows: (1) heavy traffic goes along the arterial road, (2) traffic accident happens on the arterial road marked with a " X ". The table 3.10 shows the traffic densities of corresponding traffic sections, and table 3.8 shows the traffic signal of corresponding traffic sections. These results are compared with the case that there is no traffic accident that table 3.9 and 3.7 show the traffic densities and traffic signal of the corresponding traffic section. All results are the average values of 90 minute simulation.

From this result, we see that the roundabout ways are made from the nearer side of the arterial road and as a result, there are only a small difference between the case of traffic accident and the case of no accident.

Table 3.4: Experimental result in case of 2 arterial roads

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

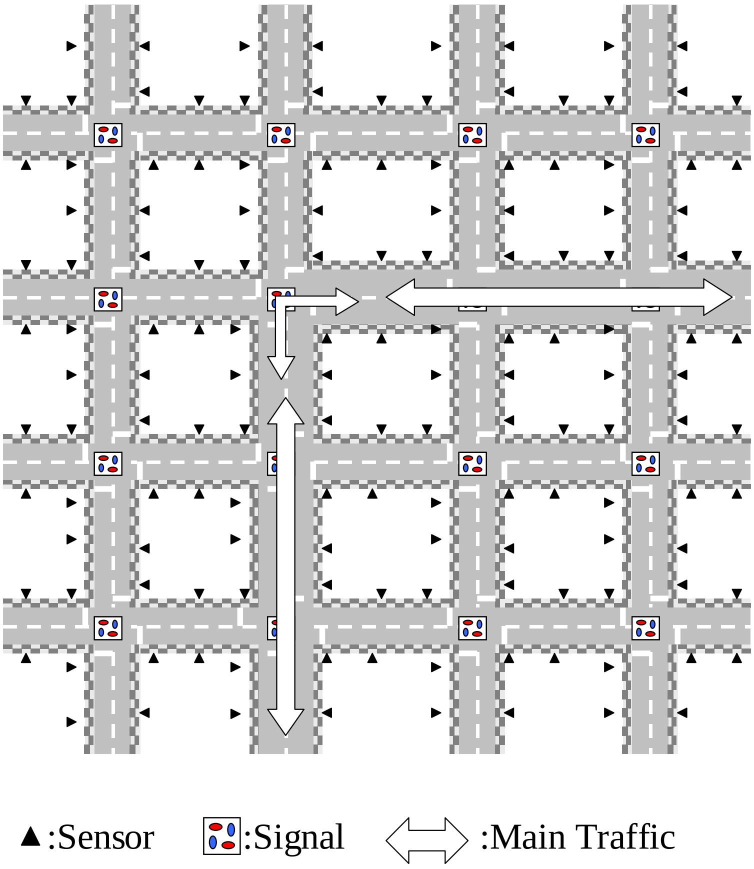

Figure 3.6: Traffic network road

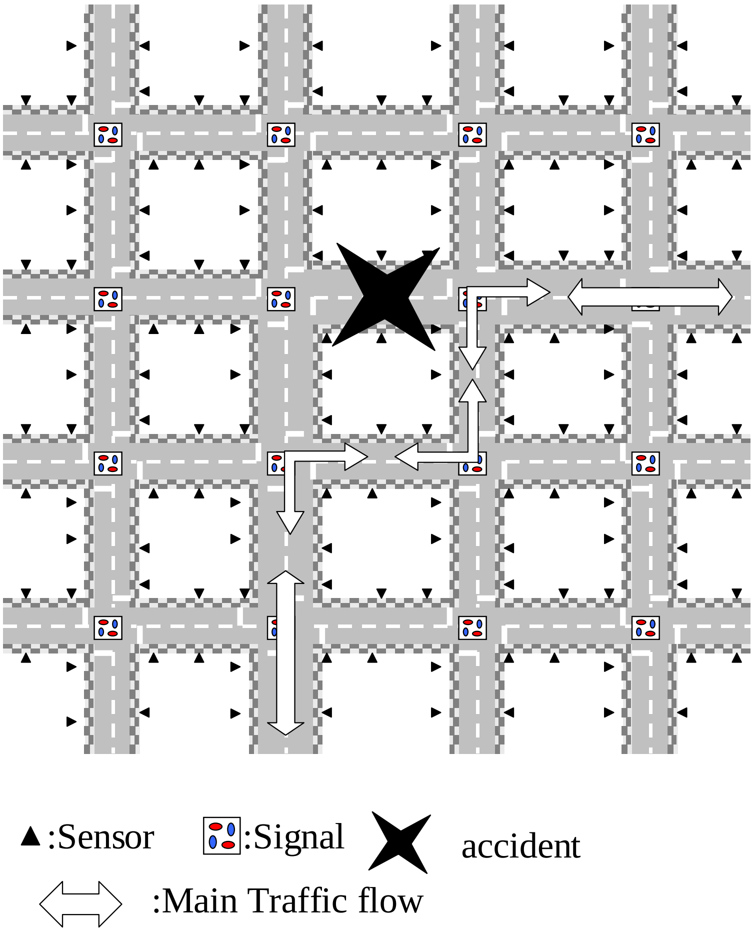

Figure 3.7: Traffic accident

control horizon	passed car
No Control	39567
MPC(H=1)	43083
MPC(H=2)	44173
MPC(H=3)	45291
MPC(H=4)	45508
MPC(H=5)	43662

Table 3.5: Passed car in normal road

control horizon	passed car
No Control	18261
MPC(H=1)	24799
MPC(H=2)	26305
MPC(H=3)	29131
MPC(H=4)	29929
MPC(H=5)	28668

Table 3.6: Passed car in accidented road

horizon	signal rates							
n/c	50.0	50.0	50.0	50.0	50.0	40.0	70.0	70.0
	50.0	30.0	50.0	50.0	50.0	30.0	50.0	50.0
h=1	9.0	9.0	89.8	80.8	91.0	89.0	82.4	2.8
	1.6	5.8	13.0	83.4	4.4	5.6	81.2	4.2
h=2	9.0	9.4	89.6	81.0	91.0	89.0	82.0	3.2
	1.6	1.4	8.4	88.8	4.8	4.8	83.2	84.4
h=3	9.0	9.0	8.4	5.6	91.0	89.0	81.8	4.0
	1.6	1.6	8.2	81.0	4.2	5.2	81.0	83.0
h=4	1.2	76.4	8.0	4.8	91.0	89.0	81.6	6.8
	1.6	1.6	8.4	87.0	5.2	89.6	83.8	4.8
h=5	9.2	9.4	7 .6	4.2	91.0	89.8	82.0	3.4
	3.0	1.6	8.8	81.2	5.6	5.0	82.0	5.2

Table 3.7: Signal rate in normal road

3.6 Conclusions

In this chapter, a new method for traffic light control based on the hybrid dynamical system theory has been proposed. First of all, the synthetic modeling method for the traffic flow control system has been proposed where the information on geometrical traffic network was modeled by using the Hybrid Petri Net, whereas the information on the behavior of traffic flow was modeled by means of the Mixed Logical Dynamical Systems (MLDS) form. The former allows us to easily apply our method to complicated traffic network due to its graphical understanding. The latter enables us to optimize the control policy for the traffic light by means of its algebraic manipulability. Secondly, the shock wave model has been introduced in order to treat the discontinuity of the traffic flow. The developed non-linear dynamics was formulated based on the Mixed Integer Non-Linear Programming problem, and yields global optimal solution coupled with convexity analysis. Lastly, the proposed Branch and Bound algorithm, which introduced the refining process, enables to minimize the introduced number of the auxiliary variables, whereas the conventional MINLP problems are known as computationally expensive. Numerical experiments have been carried out, and have shown the usefulness of the proposed design framework. Our future works include the analytical consideration of stochastically changing traffic network dynamics.

horizon	signal rates							
n/c	50.0	50.0	50.0	50.0	50.0	40.0	70.0	70.0
	50.0	30.0	50.0	50.0	50.0	30.0	50.0	50.0
h=1	81.6	82.6	87.8	85.2	91.0	87.8	89.6	86.4
	82.4	88.2	88.2	85.2	82.6	85.8	86.4	84.0
h=2	81.6	80.4	86.4	84.0	89.0	86.6	89.2	87.2
	80.2	86.8	88.0	86.6	81.6	83.6	86.4	84.4
h=3	81.6	88.8	85.8	83.8	88.0	85.4	88.2	86.0
	81.8	86.2	84.8	86.0	88.6	86.4	83.2	83.6
h=4	81.6	80.0	86.6	10.0	88.0	85.8	87.6	8.8
	88.6	7.2	83.0	81.8	80.2	86.4	81.0	81.6
h=5	81.4	89.8	84.2	83.8	88.0	85.8	89.2	86.0
	84.0	6.6	87.2	84.6	80.6	86.0	87.4	84.0

Table 3.8: Signal rate in accidented road

n/c						h=1					
44.7	21.2	35.8	13.7	33.7	15.8	43.5	12.3	33.0	17.0	59.2	53.0
89.5	42.5	15.4	12.5	58.0	10.0	90.0	10.3	10.8	69.8	82.9	10.5
35.1	31.4	38.5	17.6	19.4	29.5	13.1	10.9	29.4	13.3	86.4	28.4
10.5	11.7	61.4	12.4	14.2	13.4	11.8	71.9	40.5	14.4	15.5	69.4
49.3	89.1	10.0	89.0	10.0	10.0	87.1	89.8	10.0	63.8	25.5	19.2
68.2	89.8	10.0	87.8	53.4	64.7	85.1	90.0	10.0	89.1	89.8	10.2
82.9	11.5	10.7	10.6	10.8	42.4	19.2	69.4	21.6	12.3	11.4	34.7
81.5	13.8	12.7	17.6	14.2	12.4	41.0	11.3	21.6	19.5	20.1	11.1
h=2						h=3					
17.5	11.7	65.5	17.2	20.4	61.0	38.0	18.7	48.1	18.2	38.0	83.8
90.0	10.3	18.3	77.6	83.4	10.5	90.0	10.3	12.3	86.1	82.9	10.5
16.1	10.7	21.9	40.1	80.4	11.5	11.1	10.7	16.4	27.2	82.6	15.8
12.1	84.3	25.3	11.4	13.3	46.8	12.2	60.6	43.3	13.3	31.3	53.4
86.9	89.8	10.0	62.5	25.5	19.1	86.7	89.8	10.0	64.2	25.5	14.5
85.9	90.0	10.0	88.9	89.8	10.6	85.9	90.0	10.0	87.8	89.8	10.1
17.6	41.3	13.6	15.9	11.2	16.7	17.3	27.3	16.7	14.4	12.4	41.7
18.7	16.7	48.1	15.3	16.1	10.5	55.5	11.0	11.7	13.7	42.8	11.3
h=4						h=5					
14.2	24.4	49.1	17.5	44.6	83.2	83.3	19.9	17.2	21.1	86.0	67.4
90.0	10.3	18.7	85.9	82.9	10.5	90.0	10.3	13.3	79.3	82.7	10.5
11.1	10.8	27.2	59.9	82.4	10.7	29.4	11.0	31.1	46.8	82.4	10.3
11.7	81.7	29.9	12.3	14.1	50.5	36.2	40.9	42.6	12.5	52.0	42.5
84.6	89.8	10.0	73.5	25.5	11.9	83.1	89.8	10.0	66.6	25.5	16.1
85.1	90.0	10.0	88.5	89.8	10.1	86.5	90.0	10.0	88.1	89.8	10.1
18.4	47.2	12.2	17.2	11.1	29.8	16.6	47.3	16.7	17.5	12.6	28.7
38.9	11.4	10.9	16.1	45.7	18.9	63.0	19.5	10.6	18.9	35.0	17.5

Table 3.9: Average density in normal road

n/c						h=1					
90.0	10.0	10.4	76.4	88.8	15.2	89.2	10.4	18.0	10.9	12.2	80.3
90.0	11.8	16.2	69.5	87.0	11.5	89.8	10.5	10.9	89.9	88.1	11.0
23.3	11.4	16.0	17.2	83.1	10.2	86.6	10.6	14.8	12.0	85.3	10.1
78.0	86.5	12.7	85.4	13.8	15.1	17.3	45.6	35.1	19.7	25.7	30.9
82.0	89.8	10.5	59.5	10.0	12.1	80.3	82.8	15.9	23.8	30.7	12.4
45.0	90.0	10.0	89.2	89.7	11.9	86.3	90.0	10.5	89.7	89.7	10.3
12.3	11.6	28.0	11.0	11.4	16.4	10.8	10.6	10.6	87.3	20.1	81.0
47.7	11.3	86.2	11.5	20.3	12.9	33.0	11.6	74.7	13.3	18.0	18.8
h=2						h=3					
85.7	11.2	53.5	11.3	18.6	12.9	86.6	12.9	28.2	11.5	11.1	56.4
89.8	10.3	10.9	89.9	88.3	11.0	89.8	10.5	10.9	81.4	88.2	11.0
86.6	10.6	18.8	15.3	85.5	10.1	74.3	10.6	17.9	11.0	82.2	10.1
13.8	80.8	13.5	64.8	11.5	70.8	66.3	72.3	15.8	50.6	11.4	31.9
74.4	81.1	10.5	53.6	33.0	13.8	69.0	88.9	10.1	43.9	25.2	12.4
85.7	90.0	10.5	89.6	89.7	10.3	86.6	90.0	10.0	89.6	89.7	10.3
10.8	15.8	10.7	89.5	13.1	57.5	10.5	13.6	10.8	80.2	11.7	63.0
76.6	12.2	70.7	13.5	38.0	11.4	48.2	11.2	49.3	13.9	15.3	17.9
h=4						h=5					
44.0	12.9	51.8	13.1	54.2	11.6	88.2	10.7	11.0	12.4	68.2	15.3
90.0	10.3	11.7	82.9	88.2	11.5	89.8	10.4	10.9	81.1	88.3	11.0
81.0	10.4	14.5	57.3	85.9	10.1	81.4	10.4	15.5	16.7	85.3	10.1
25.2	70.3	15.8	36.9	53.4	17.6	16.6	68.4	16.4	50.5	13.7	27.9
82.4	89.7	10.1	74.0	10.0	15.8	78.2	89.5	11.5	43.5	31.8	13.0
86.6	90.0	10.0	89.1	89.7	10.3	86.4	90.0	10.5	89.7	89.7	10.3
12.2	10.8	11.1	86.6	12.2	70.0	11.0	17.3	10.5	88.2	15.8	79.7
15.4	24.5	65.6	14.8	14.0	33.1	89.1	10.6	39.9	15.7	31.3	10.8

Table 3.10: Average density in accidented road

Chapter 4

Traffic network hybrid feedback controller via 0-1 classification of piecewise autoregressive exogenous system with hierarchy

4.1 Introduction

In chapter 2, a piece-wise affine traffic flow model is proposed, where the traffic flow was represented with the traffic densities of two consecutive districts in order to consider the behavior of shock wave. The traffic flow dynamics were optimized based on the Mixed Logical Dynamical System (MLDS) framework [35]. Although the method proposed in chapter 2 is the well-established optimization procedure, it is unfit for large-scale traffic network control, since the method based on the Mixed Integer Linear Programming (MILP) problem is computationally expensive.

The traffic network control system is typical hybrid systems with nonlinear dynamics. The model reported in chapter 3 is formulated based on the Mixed Integer Non-Linear Programming (MINLP) problem, where the mixed integer nonlinear traffic control problem is recast to the convex programming problem, guaranteeing the solution optimality. Even though the control problem in chapter 3 contained the non-convex constraint function as a whole, the generated sub-problems are always included in the class of the convex programming problem, using a general performance criterion.

A general method for obtaining the hybrid model is to use the Piece-Wise Affine systems since the PWA approximation has universal properties and the obtained system can be directly transformed to several classes of the hybrid dynamical systems. The state and output maps of the PWA systems are both piecewise affine form, where the PWA map $f : \chi \rightarrow R^q$ is defined as follows.

$$f(\boldsymbol{x}) = \begin{cases} \boldsymbol{\theta}_1^T \boldsymbol{\rho}(\kappa) & \text{if } \boldsymbol{x}(\kappa) \in \chi_1 = \{\boldsymbol{x}(\kappa) | \boldsymbol{H}_1 \boldsymbol{x}(\kappa) \leq \boldsymbol{W}_1\} \\ \vdots & \vdots \\ \boldsymbol{\theta}_s^T \boldsymbol{\rho}(\kappa) & \text{if } \boldsymbol{x}(\kappa) \in \chi_s = \{\boldsymbol{x}(\kappa) | \boldsymbol{H}_s \boldsymbol{x}(\kappa) \leq \boldsymbol{W}_s\} \end{cases} \tag{4.1}$$

$$\boldsymbol{x} = [y(\kappa-1), \cdots, y(\kappa-n_a), u^T(\kappa-1), \cdots, u^T(\kappa-n_b)]^T \tag{4.2}$$

$$\boldsymbol{\theta}_i = [a_{i,1}, \cdots, a_{i,n_a}, b_{i,1}^T, \cdots, b_{i,n_b}^T, f_i]^T, \tag{4.3}$$

where $\boldsymbol{\rho}(\kappa)$ is $[\boldsymbol{x}(\kappa), 1]^T$ ($\boldsymbol{x}(\kappa)$ is the regression vector, consists of the past inputs and outputs), χ_i is the convex polyhedron which satisfies $\bigcup_{i=1}^{s} \chi_i = \chi \subseteq R^q$ and $\chi_i \bigcap \chi_j = \phi$, $\forall i \neq j$,, $y(\kappa)$ is the control output, and the pair $(\boldsymbol{H}_i, \boldsymbol{W}_i)$ is the guard of χ_i.

The Piece-Wise Auto Regressive eXogenous (PWARX) model [45] is the discontinuous output map along the boundary of each region. The main difference compared with conventional K-means based classification is that a confidence level is measured, coupled with the covariance of the data in the θ-x space. χ_i and θ_i are obtained by iteratively applying the piecewise fitting process and the cluster updating process. Although this method is an efficient clustering procedure for the hybrid dynamical systems, the number of sub models must be a priori fixed, randomly choosing the initial bases of each clustering group. However, the performance of this iterative clustering procedure is in general very sensitive to the initialization. The hierarchical subspace clustering [47] is a challenging problem for identifying the system with highly nonlinear and complex dynamics. This method is mainly applied in the neural networks community. However, the neural networks based hierarchical clustering schemes in general take long time for the network learning.

In this paper, we propose a new design method for the traffic network hybrid feedback controller. The method reported in chapter 3 is an elaborate contrivance to avoid redundant introduction of binary variables. Although the solution optimality is guaranteed in chapter 3, this method requires much computational efforts. Since the output of the traffic network controller is the 0-1 binary signals, the control output obtained by applying the controller design method in chapter 3, is reproduced in this chapter applying 0-1 classifications of the PWARX systems. In the proposed method, the PWARX classifier describes the nonlinear feedback control law of the traffic control system. This implies we don't need the time-consuming searching process of the solver such as the Branch-and-Bound algorithm to solve the mixed integer nonlinear programming (MINLP) problem, and furthermore the exactly same solutions or very similar solutions are obtained in a very short time.

The classification problem we address in this chapter is a special 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 the binary output, i.e., all the components of θ, a and b except for f will be zeros. This implies we need consideration for the binary output. A new performance criterion is presented in this paper to consider not only previously covariance of θ, but also the covariance of y. The proposed method is a hierarchical classification procedure, where the cluster splitting process is introduced to the cluster with worst classification performance at every iteration 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.

4.2 Traffic flow modeling

The Traffic Flow Control System (TFCS) is the collective entity of traffic network, traffic flow and traffic signals. In this section, the HPN (Hybrid Petri Net) model is developed, which provides an algebraic descriptions for the TFCS.

4.2.1 Traffic flow dynamics

In order to obtain the traffic flow dynamics, the relationship among $q_i(\tau)$, $k_i(\tau)$ and $v_i(\tau)$ must be specified, where $q_i(\tau)$ is the traffic flow i.e., the number of vehicles passing through the boundary per unit time of two successive traffic sections at time τ, $k_i(\tau)$ is the traffic density i.e., the number of vehicles on the ith l_i meters long section, and $v_i(\tau)$ is the traffic flow speed i.e., the average speed of the traffic flow $q_i(\tau)$. One of the simple ideas is to use the well-known model

$$q_i(\tau) = -\frac{(k_i(\tau) + k_j(\tau))}{2} \frac{v_i(k_i(\tau)) + v_j(k_j(\tau))}{2} \tag{4.4}$$

supposing that the density k_* and average velocity v_* of the flow in i and jth sections are almost identical. Then, by incorporating the velocity model

$$v_i(\tau) = \quad v_{f_i} \cdot \left(1 - \frac{k_i(\tau)}{k_{jam}}\right) \tag{4.5}$$

with eq.(4.4), the flow dynamics can be uniquely defined. Here, k_{jam} is the density in which the vehicles on the roadway are spaced at the minimum intervals (traffic-jammed), and v_{f_i} is the free velocity, that is, the velocity of the vehicle when no other car 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 lights. In order to treat the discontinuities of the density among neighboring sections (i.e. neighboring continuous places), the idea of 'shock wave'[53] is introduced as follows. Here, the movement of this boundary is called shock wave and the moving velocity of the shock wave c_i depends on the densities and average velocities of ith and jth sections as follows [53]:

$$c_i(\tau) \ = \ \frac{v_i(\tau)k_i(\tau) - v_j(\tau)k_j(\tau)}{k_i(\tau) - k_j(\tau)} \tag{4.6}$$

The traffic situation can be categorized into the following four types, and these considerations lead to the following models:

$$q(k_i(\tau), k_j(\tau)) \ = \ \begin{cases} -\left(\frac{k_i(\tau)+k_j(\tau)}{2}\right) v_f \left(1 - \frac{k_i(\tau)+k_j(\tau)}{2k_{jam}}\right) \\ \quad if \ \ k_i(\tau) \geq k_j(\tau) \\ \\ -v_{f_i}\left(1 - \frac{k_i(\tau)}{k_{jam}}\right) k_i(\tau) \\ \quad if \ \ k_i(\tau) < k_j(\tau) \text{ and } c_i(\tau) > 0 \\ \\ -v_{f_j}\left(1 - \frac{k_j(\tau)}{k_{jam}}\right) k_j(\tau) \\ \quad if \ \ k_i(\tau) < k_j(\tau) \text{ and } c_i(\tau) \leq 0 \end{cases} \tag{4.7}$$

, where $0 \leq k_i(\tau) \leq k_{jam}, 0 \leq k_j(\tau) \leq k_{jam}$.

4.2.2 Traffic network model at an intersection

In this subsection, we develop the traffic network model at an intersection. Figure 4.1 shows the HPN model of the jth intersection, where the notation for other than southward entrance lane is omitted. In Fig.4.1, $l_{j,S}$ and $l_{j,N}$ are the length of the districts $p_{c,j_{IS}}$ and $p_{c,j_{ON}}$, and the numbers of the vehicles at $p_{c,j_{IS}}$ and $p_{c,j_{ON}}$ are $k_{j_{IS}} \cdot l_{j,IS}$ and $k_{j_{ON}} \cdot l_{j,ON}$, respectively. 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,

$$k_{j_{SW}}(\tau) \ = \ k_{j_{IS}}(\tau)\zeta_{j,SW}, \tag{4.8}$$

$$k_{j_{SN}}(\tau) \ = \ k_{j_{IS}}(\tau)\zeta_{j,SN}, \tag{4.9}$$

$$k_{j_{SE}}(\tau) \ = \ k_{j_{IS}}(\tau)\zeta_{j,SE}. \tag{4.10}$$

Note that these probabilities are determined by the traffic network structure, and satisfy $0 \leq \zeta_{j,SW}(\tau) \leq 1$, $0 \leq \zeta_{j,SN}(\tau) \leq 1$, $0 \leq \zeta_{j,SE}(\tau) \leq 1$, and $\zeta_{j,SW}(\tau) +$

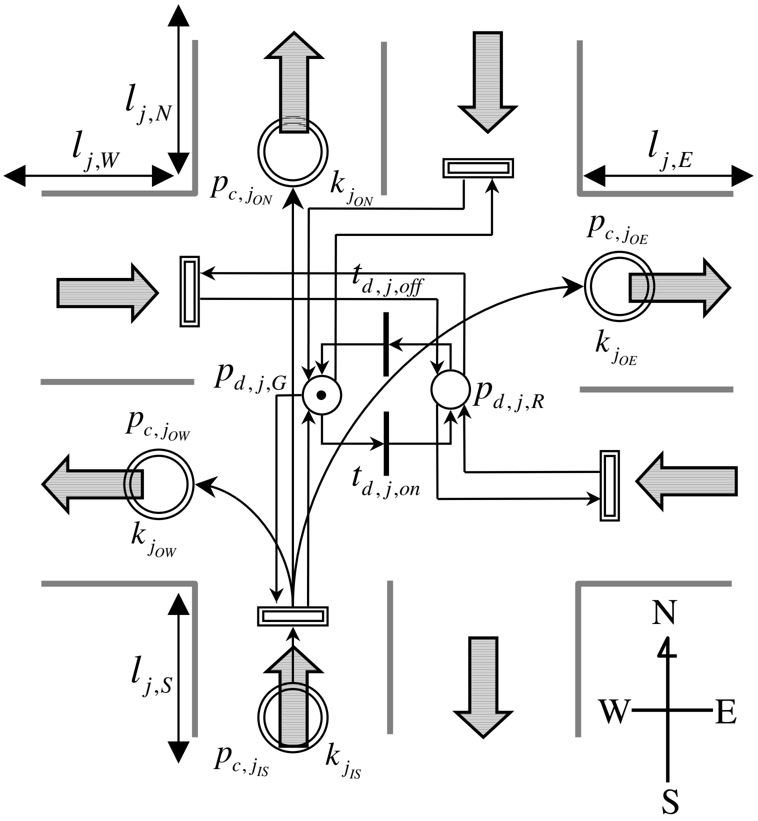

Figure 4.1: HPN model of the intersection

$\zeta_{j,SN}(\tau) + \zeta_{j,SE}(\tau) = 1$. Therefore, the traffic flows of the three directions are represented by

$$q\left(k_{j_{SN}}(\tau), k_{j_{ON}}(\tau)\right), \tag{4.11}$$

$$q\left(k_{j_{SW}}(\tau), k_{j_{OW}}(\tau)\right), \tag{4.12}$$

$$q\left(k_{j_{SE}}(\tau), k_{j_{OE}}(\tau)\right). \tag{4.13}$$

Since the probability ζ includes the affection of yellow light, yellow light is not explicitly represented in Fig. 4.1.

4.3 Traffic network control system

4.3.1 Mixed logical dynamical system-like representation

In this section, the Mixed Logical Dynamical System (MLDS) -like form is introduced to formulate the Model Predictive Control (MPC) [35].

The MLDS description allows us to represent systematically the behavior of system dynamics, where the nonlinear dynamics may be approximated by introducing the auxiliary variables. However the methods for finding optimal solution are in general computationally very expensive. This is because of the large number of the introduced auxiliary variables. In chapter 3, the nonlinear dynamics is not linearized in a piece-wise manner with the intention of improving the modeling accuracy, where the Mixed Integer Non-Linear Programming (MINLP) problem is solved by the proposed Branch & Bound strategy.

The traffic flow q is the function of k_i and k_j which are the traffic densities of the two consecutive district i and j, and contain nonlinearity because of the multiplications of the two variables as in Eq.(4.7). Since q is also the function of traffic light, the traffic flow adjoining the intersection i can be represented by introducing the (continuous) auxiliary variable z as follows

$$z_{i,\tilde{i}}(\kappa) \quad = \quad q\Big(\frac{\boldsymbol{x}_i(\kappa)}{l_i}, \frac{\boldsymbol{x}_{\tilde{i}}(\kappa)}{l_{\tilde{i}}}\Big) u_j(\kappa) \tag{4.14}$$

, where the traffic light at the intersection i is $u_i(\kappa)$. Here, $GREEN(RED)$ light of the north-south orientation and $RED(GREEN)$ light of the east-west orientation are represented by 0 (1). With z of Eq.(4.14), the state equation and the constraint inequality are formulated as follows

$$\begin{aligned}\boldsymbol{x}\ (\kappa+1) &= \boldsymbol{Ax}\ (\kappa)+\boldsymbol{Bz}(\kappa) && (4.15)\\ \boldsymbol{E_{2\kappa}z}(\kappa) &\leq \boldsymbol{E_{1\kappa}u}(\kappa)+\boldsymbol{E_{4\kappa}x}(\kappa)+\boldsymbol{E_{5\kappa}} && (4.16)\end{aligned}$$

, where $\boldsymbol{x}=[x_1,x_2,...,x_{n_x}]^T$,$u_j \in \{0,1\}$,$\tilde{i} \in t_j^{\bullet}$ and $\boldsymbol{A}$ is the matrix with the suitable dimension. The equation (4.16) represents the logical relationship (proposition) of Eq.(4.7), the maximum capacity constraints of each place, the firing speed of each transition respectively and so on.

Although the mixed logic dynamical system is represented in a compact form as Eq.(4.15) and Eq.(4.16), this cannot be directly applied to the model predictive control scheme, since z has the multiplication of the three decision variables (z amounts to $\alpha\ u \cdot x^2$).

4.3.2 Mixed integer non-linear programming problem

The Model Predictive Control (MPC) [51][52] is one of the well-known paradigms for optimizing the systems with constraints and uncertainties. In the MPC policy, the control input at each sampling instant is decided based on the prediction of the behavior for the next several sampling periods called the prediction horizon. In order to formulate the optimization procedure, first, Eq.(4.15) is modified to evaluate the state and input variables in the prediction horizon as follows:

$$\begin{aligned}\boldsymbol{x}(\kappa+\lambda|\kappa) = {} & \boldsymbol{A^{\lambda}}x(\kappa)\\ & +\sum_{\eta=0}^{\lambda-1}\{\boldsymbol{A^{\eta}}(\boldsymbol{Bz}(\kappa+\lambda-1-\eta|\kappa)\\ & \quad \cdot\boldsymbol{u}(\kappa+\lambda-1-\eta|\kappa))\}, \qquad (4.17)\end{aligned}$$

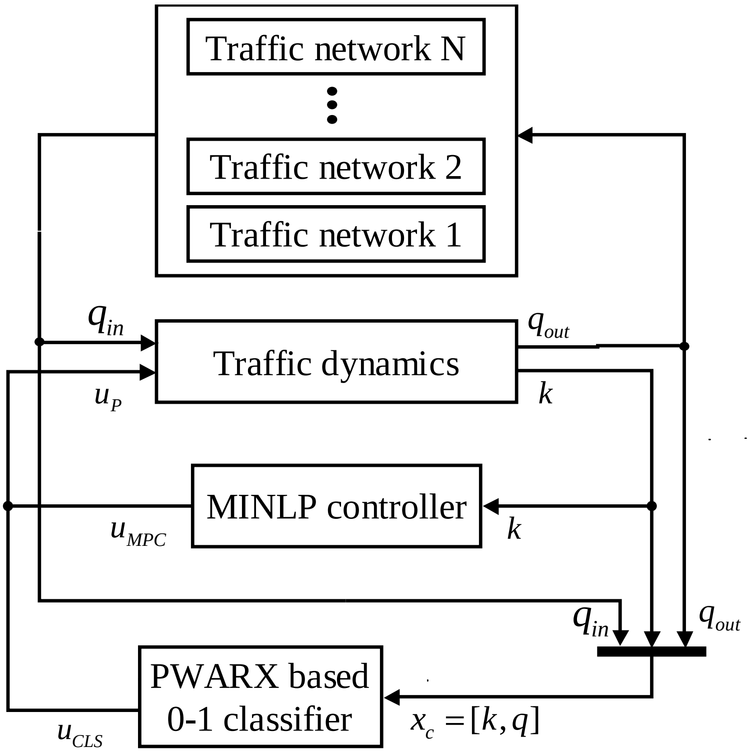

Figure 4.2: Outline of the proposed controller

where $\boldsymbol{x}(\kappa+\lambda|\kappa)$ 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 Eq.(4.15) starting from the state $\boldsymbol{x}(\kappa|\kappa) = \boldsymbol{x}(\kappa)$.

The following performance criterion is introduced to maximize the traffic flow.

$$J(\kappa) \quad = \quad \sum_{\eta=0}^{H-1} \sum_{i=0}^{n_z-1} z_i(\kappa+\eta), \tag{4.18}$$

Then the MINLP problem for the traffic network control can be formulated as follows.

$$\begin{aligned}
&\mathit{minimize} \quad -J(\kappa) \\
&s.t. \quad \boldsymbol{x} \ \in \Re^{\ r}, \\
&\qquad \boldsymbol{u} \ \in \{0,1\}^l, \\
&\qquad \forall\ \eta \in [1,H],\ \ \boldsymbol{x}\ (\kappa+\eta+1) = \boldsymbol{A}\boldsymbol{x}\ (\kappa+\eta) + \boldsymbol{B}\boldsymbol{z}(\kappa+\eta) \\
&\qquad \forall\ \eta \in [1,H],\ \boldsymbol{E_{2\kappa+\eta}}\boldsymbol{z}(\kappa+\eta) \leq \boldsymbol{E_{1\kappa+\eta}}\boldsymbol{u}(\kappa+\eta) \\
&\qquad\qquad\qquad + \boldsymbol{E_{4\kappa+\eta}}\boldsymbol{x}(\kappa+\eta) + \boldsymbol{E_{5\kappa+\eta}}
\end{aligned}$$

,where $r = Hn_x$,$l = Hn_z$.

4.4 0-1 classification based on piecewise auto regressive exogenous system

The MINLP based traffic network controller introduced in the previous chapter is generally known to require large computational efforts. In this chapter, we propose a new controller design method for the hybrid systems with the binary inputs. The proposed method develops the classification map using the modified PWARX systems, which relates the binary input variables and all the observational variables including the past input and output variables. The output variables $y(\kappa)$ (which corresponds to the plant input $u_p(\kappa)$) are obtained by only finding the corresponding cluster among the classification map, while in the conventional methods, the MINLP problems are solved at every sampling instant.

The Fig.4.2 describes the 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 inflow from the outside and outflow to the outside are closely affected by the traffic flow in the adjoining traffic intersections. In order to construct the classification map, we need the data of the input and output variables of the MINLP controller obtained by applying to various traffic situation of the network. For this purpose, we adopted a Cellular Automaton based simulator in this paper.

4.4.1 Classification problem of hybrid dynamics

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

$$\boldsymbol{x}(\kappa) = [y^T(\kappa-1), y^T(\kappa-2), \cdots, y^T(\kappa-n_a), \tag{4.19}$$

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

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

$$\chi_i = \{\boldsymbol{x} | \boldsymbol{V}_i \boldsymbol{x}(\kappa) \leq \boldsymbol{W}_i\}. \tag{4.20}$$

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

$$f_i(\boldsymbol{x}(\kappa)) = \boldsymbol{\theta}_i \boldsymbol{\rho}(\kappa) \tag{4.21}$$

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

$$\boldsymbol{\theta}_i = [a_{i,1}, \cdots, a_{i,n_a}, b_{i,1}^T, \cdots, b_{i,n_b}^T, f_i]^T \tag{4.22}$$

The classification problem we address in this paper is a special 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 the binary output, i.e., all the components of θ, a and b expect for f will be zeros. Therefore the value of $f(\boldsymbol{x}(\kappa))$ must be only 0 or 1.

For this type of clustering problem, the conventional PWARX system cannot well reproduce the restricted 0-1 output variable. Since they simultaneously obtain the clusters and the (linear) dynamics of the clusters applying the least squared method to each of the fixed number of clustering region, the overall accuracy of their reproduced model is not so high. Furthermore they are very sensitive to the initialization concerning the number of the clusters, the position of the initial cluster, and so on.

4.4.2 Classification based on piecewise auto regressive exogenous system

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

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

Step 2 Obtain the cluster D_i of ξ points which minimize the following performance criterion

$$J = \sum_{i=1}^{s} \sum_{\xi_j \in D_i} ||\xi_j - \mu_i||^2_{R_j^{-1}} \tag{4.23}$$

Step 3 Update the centers μ according to the following formula.

$$\tilde{\mu}_i = \frac{\sum_{j:\xi_j \in D_i} \xi_j w_j}{\sum_{j:\xi_j \in D_i} w_j} \tag{4.24}$$

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

$$\mu = \tilde{\mu}_i \tag{4.25}$$

and go to Step 2.

In Step 2, R_j is defined as

$$R_j = \begin{bmatrix} V_j & 0 \\ 0 & Q_j \end{bmatrix} \tag{4.26}$$

, where

$$\begin{aligned}
V_j &= \frac{S_j}{c-(n_a+n_b)+1}(\Phi_j^T\Phi_j)^{-1} && (4.27)\\
Q_j &= \sum_{(x,y)\in C_j}(x-m_j)(x-m_j)^T && (4.28)\\
\Phi_j &= \begin{bmatrix} x_1 & x_2 & \cdots & x_c \\ 1 & 1 & \cdots & 1 \end{bmatrix} && (4.29)\\
S_j &= y_{c_j}^T(I-\Phi_j(\Phi_j^T\Phi_j)^{-1}\Phi_j^T)y_{c_j} && (4.30)\\
m_j &= \frac{1}{c}\sum_{(x,y)\in C_j}\boldsymbol{x}, j=1,\cdots,N && (4.31)\\
\xi_j &= [(\theta_j)^T, m_j^T] && (4.32)\\
w_j &= \frac{1}{\sqrt{(2\pi)^{(2n_a+2n_b+1)}\det(R_i)}}. && (4.33)
\end{aligned}$$

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

The main difference of this method compared with K-means is that based on the confidence level w_j, this method assigns the vectors ξ to the cluster D_i in the parameter vector θ-$\boldsymbol{x}$ space, while K-means assigns the data to the cluster C_i in the state vector $\boldsymbol{x}$ space. This property serves for the identification of y that the mixed cluster is abated addressing the dynamics of y.

4.4.3 0-1 classification based on modified PWARX system

The desirable outputs in a pure cluster of the problem addressed in this paper are continued by the same values, 0 or 1 in the $\boldsymbol{x}$-y space. All the values except for the offset variable f among the parameters of θ will be zeros, i.e., the dynamics in θ-$\boldsymbol{x}$ space will be almost same. Therefore in the conventional PWARX system, the regions with same dynamics are often considered to be included in the same cluster.

The proposed method described below is a hierarchical PWARX system for 0-1 classification as follows.

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

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

$$J_\chi = \sum_{i=1}^{s} \sum_{\xi_j \in D_i} ||\xi_j - \mu_i||^2_{R_j^{-1}} \tag{4.34}$$

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

$$\begin{aligned} \text{find} \quad & \boldsymbol{V}_{i,i^T} \text{ and } \boldsymbol{W}_{i,i^T} & (4.35)\\ \text{minimize} \quad & \boldsymbol{V}_{i,i^T}\boldsymbol{V}_{i,i^T}^T & (4.36)\\ \text{subject to} \quad & \zeta_l(\boldsymbol{V}_{i,i^T}^T \boldsymbol{x_l} + \boldsymbol{W}_{i,i^T}) \geq 1 & (4.37) \end{aligned}$$

, where l is the data number and ζ is defined as follows.

$$\zeta_l = \begin{cases} 1 & if \quad \xi(\boldsymbol{x_l}) \in D_i \\ -1 & if \quad \xi(\boldsymbol{x_l}) \in D_{i^T} \end{cases} \tag{4.38}$$

Here $\xi(\boldsymbol{x})$ is the function which obtains the corresponding value of ξ from $\boldsymbol{x}$, i.e., ξ is a translation of $\boldsymbol{x}$ in the θ-$\boldsymbol{x}$ 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 μ according to the following formula.

$$\tilde{\mu}_i = \frac{\sum_{j:\xi_j \in D_i} \xi_j w_j}{\sum_{j:\xi_j \in D_i} w_j} \tag{4.39}$$

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

$$\mu = \tilde{\mu}_i \tag{4.40}$$

and go to Step 2.

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

$$J_i = \sigma^2(y(\kappa)). \tag{4.41}$$

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

$$\chi = \chi - \chi_i \tag{4.42}$$

$$\chi_i = \{\boldsymbol{x} | \boldsymbol{V}_i \boldsymbol{x} \leq \boldsymbol{W}_i\} \tag{4.43}$$

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

$$s = s + s_r \tag{4.44}$$

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

Step 4-2 Set i_m as follows.

$$i_m = \arg \min_{i \in [1,s]} \sigma^2(y(\kappa)) \tag{4.45}$$

Step 4-3 If $J_{i_m} \leq \gamma$, terminate with success, otherwise, obtain N^T 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 hyperplane $\boldsymbol{V}_{i,i^T}\boldsymbol{x} + \boldsymbol{W}_{i,i^T} \leq 0$ is proportional to $(\boldsymbol{V}_{i,i^T}^T \boldsymbol{V}_{i,i^T})^{-1}$ since letting the hyperplane which maximize the margin α from the data points $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ as follows

$$\boldsymbol{V}\boldsymbol{x}^+ + \boldsymbol{W} = \alpha \tag{4.46}$$

and

$$\boldsymbol{V}\boldsymbol{x}^- + \boldsymbol{W} = -\alpha, \tag{4.47}$$

the maximal margin α_{MAX} is defined as follows

$$\begin{aligned} \alpha_{MAX} &= \frac{1}{2}\Big(\frac{\boldsymbol{V}}{||\boldsymbol{V}||_2}\boldsymbol{x}^+ - \frac{\boldsymbol{V}}{||\boldsymbol{V}||_2}\boldsymbol{x}^-\Big) && (4.48)\\ &= \frac{1}{2||\boldsymbol{V}||_2}(\boldsymbol{V}\boldsymbol{x}^+ - \boldsymbol{V}\boldsymbol{x}^-) && (4.49)\\ &= \frac{\alpha}{||\boldsymbol{V}||_2} && (4.50) \end{aligned}$$

Therefore by minimizing $(\boldsymbol{V}_{i,i^T}^T \boldsymbol{V}_{i,i^T})$, the margin can be maximized.

4.5 Generation of traffic flow data

4.5.1 Cellular automaton based traffic network simulator

When we construct the input-output PWA map of the traffic network controller, the traffic flow data with sufficient amounts of a variety traffic situation must be provided. However it is generally impossible to take the experimental data, directly applying the developed traffic controller to the real traffic system. Therefore in this paper, we used Cellular Automaton model, which is well known to reproduce real traffic flow dynamics [2].

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

$$n_j(\tau+1) = n_j^{in}(\tau)(1-n_j(\tau)) - n_j^{out}(\tau), \tag{4.51}$$

where $n_j(\tau)$ is the state of cell j which represents the occupation by the vehicle ($n_j(\tau) = 0$ implies that the 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 a 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:

Step 1 (Acceleration rule) All vehicles, that have not reached at the speed of the maximum speed v_f, accelerate their speed $v_{\langle j\rangle}(\tau)$ by one unit velocity v_{unit} as follows:

$$v_{\langle j\rangle}(\tau+\Delta\tau) \equiv v_{\langle j\rangle}(\tau) + v_{unit}. \tag{4.52}$$

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:

$$v_{\langle j\rangle}(\tau+\Delta\tau) \equiv \min\{e, v_{\langle j\rangle}(\tau+\Delta\tau)\}. \tag{4.53}$$

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

$$v_{\langle j\rangle}(\tau+\Delta\tau) \equiv v_{\langle j\rangle}(\tau+\Delta\tau) - p\cdot v_{unit}. \tag{4.54}$$

The parameters used in the simulation are as follows: the computational interval $\Delta\tau$ is 1 [sec], each cell in the CA is assigned to 4.5 [m]-long interval on the road, the 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).

4.5.2 Simulation environment

For this simulation, we used following traffic network: we try to develop the traffic controller for intersection and entire traffic network consists of 4 intersections connected with each other as Fig.4.4. The length of each block is 1000 $[m]$ long and the length of each section is 500 $[m]$ long, all with 2 lanes bi-directionally. We empirically took the traffic flow data using the CA based simulator, where the sampling interval of CA is 1 sec and the sampling interval in the traffic network data saving is 10 sec. This is for the traffic network controller construction.

In order to consider a variety of traffic situation, we used two types of the traffic flow dynamics as follows: a sinusoidal wave is for considering the steady state traffic flow with a variety size, and a square wave is for considering the non-steady state traffic flow such as the effect from the stalled traffic at the adjoining section(s) or change of the adjoining traffic signal. Furthermore, in order to consider the combination of the traffic flow from the adjoining 4 sections, the periods of the waves are set to be different as 2000, 4000, 8000, and 16000 sec for the sinusoidal wave and 100, 200, 400, and 800 sec for the square wave, respectively. The figure 4.3 are the traffic flow data from 4 directions in this paper, where k_E, k_W, k_S and k_N imply the traffic inflows from east, west south and north sides of the intersection and 5000 patterns of traffic situations during 50000 seconds are simulated.

4.6 Classification results

4.6.1 Mixed integer non-linear programming controller coupled with model predictive control

The 5000 data sets obtained in the previous chapter are classified based on the proposed 0-1 classification method. For this simulation, we set the initial number of cluster, s, to be 100 and whenever 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 4.1 - 4.4. In TABLE 4.1 and 4.3,

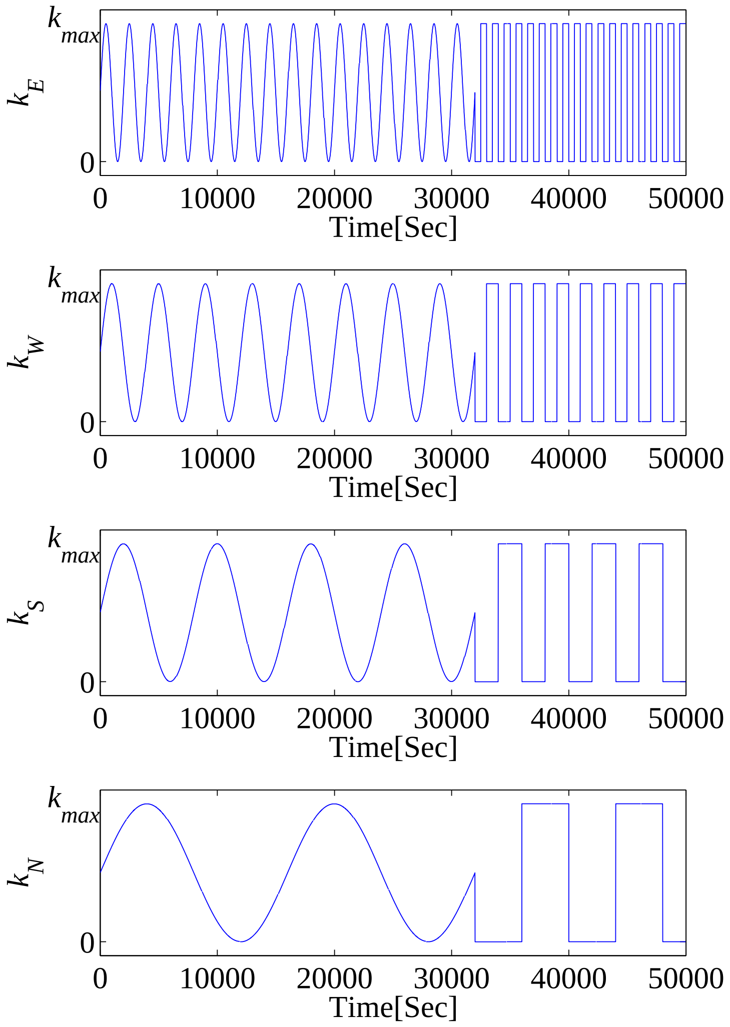

Figure 4.3: Density of traffic flow

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 4.1: Stepwise cluster number (H=1)

"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 4.2 and 4.4.

While the data shown in TABLE 4.1 and 4.2 are obtained by applying MPC horizon $H = 1$, the data shown in TABLE 4.3 and 4.4 are obtained by applying MPC horizon $H = 3$, respectively.

4.6.2 Comparison with conventional PWARX system

In this subsection, the conventional PWARX system is compared with our proposed method. TABLE 4.5 and 4.6 compare the cluster number and data number in the clusters of the results obtained by applying the proposed method and conventional method. In the TABLE 4.5 and 4.6, the conventional method is applied with the initial cluster number of 100, 200, 300, 400 and 500 respectively. Although most of data are well classified using the conventional PWARX system with introduction of large number of clusters, 2.8 and 1.6 percents of the total data were not correctly classified. In contrast the proposed method was perfectly classified

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 4.2: 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 4.3: 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 4.4: Stepwise data number in the cluster(H=3)

introducing relatively small cluster number.

4.7 Traffic network control simulation results

4.7.1 Case study example for traffic network control

In this subsection, the effectiveness of our proposed method for the large-scale traffic network control with the arterial roads shown in Fig.6. The proposed 0-1 PWARX classification based controller is adopted to each intersection. In Fig.4.4, the center intersection is surrounded by black line called control block (CB), where the traffic flow information such as the traffic density is measured for the 0-1 classification of the center intersection traffic network controller. Fig.4.4 illustrates that

	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 4.5: 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 4.6: 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 4.7: 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 4.8: Comparison of data number in the cluster (H=3)

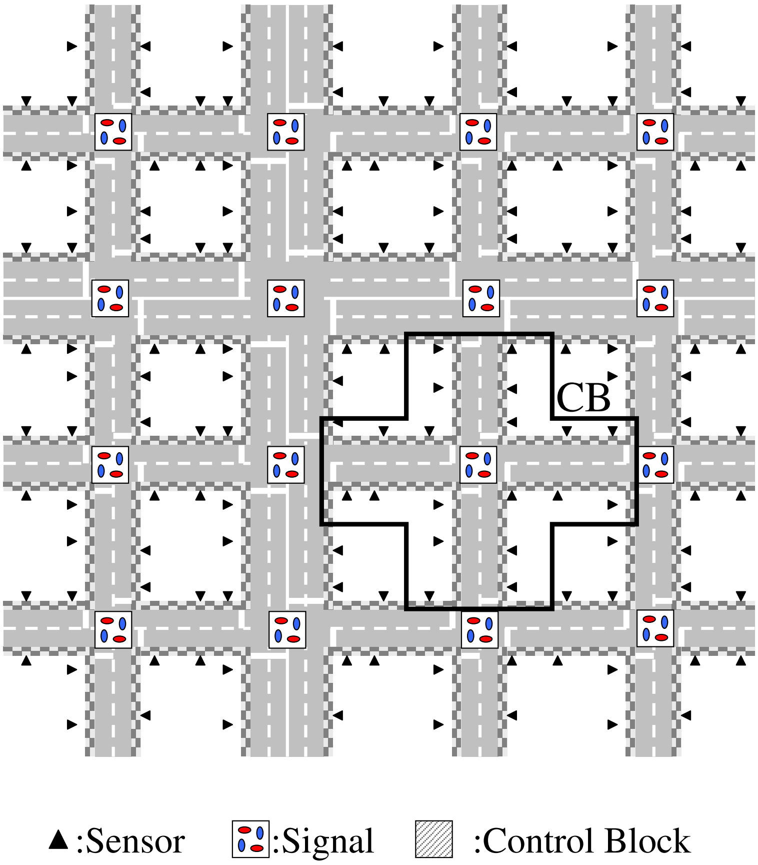

Figure 4.4: Traffic network

sixteen control blocks constitute the entire traffic network where the sensory information at each boundary of CBs is shared with the control of the adjoining blocks. Note that two arterial roads are running north-south (second road from the left) and east-west (second road from the top), respectively.

We assume that from the outside of the network, the 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 the maximum velocity (no congestion arises and affects the traffic flow inside the network). We used (4.18) as a performance criterion. All results are obtained from simulations over 30 minutes, where the sampling interval T_s is 10 [sec].

TABLE 4.9 shows the obtained solutions by applying the proposed method in the case that there are 2 arterial roads. The simulation result shows that the proposed method has good performance. Note that our method has always better or equal solutions, compared with the cases of 'No Control'. Here 'No Control' implies the traffic signals are changed every 30 minutes. The reason why the controller with H is 4 or 5 is worse than the case with $H = 3$ is that the traffic network is dynamically changing system as shown in Fig. 4. we used the random probabilities of left and right turning at every intersection. Therefore too long prediction horizon may deteriorate the control performance. If traffic jam occurs in a specific section, the controllers in the adjoining section or intersection will take action to alleviate the congestion that a new action will be taken during long horizon H. In other words the controller with H=3 is good enough for this problem.

4.7.2 Comparison of computational efforts

Lastly the computational amounts are investigated in this subsection. TABLE 4.10 compares the computational efforts of the results obtained by applying the proposed 0-1 classification based method and the MINLP controller reported in chapter 3. It is known that the solution method of the hybrid dynamical system is extremely burdensome, requiring exponential time of the binary variable number.

	Passed Vehicles
No Control	30089
MINLP controller with H=1	39363
MINLP controller with H=2	39246
MINLP controller with H=3	51059
MINLP controller with H=4	50697
MINLP controller with H=5	40939

Table 4.9: Comparison of control performance

In contrast, the proposed method requires only 0.07 seconds regardless of the binary variable number. In this simulation, the plant input u_P (controller output $u(\kappa)$) is obtained by applying the MINLP controller with MPC horizon H=3 as saw in the result of TABLE 4.9. The same solutions are obtained using only 5.7×10^{-2} percent time the MINLP controller requires.

4.8 Conclusions

In this paper we have proposed a new design method for the traffic network hybrid feedback controller. Since the output of the traffic network controller is the 0-1 binary signals, the output of the developed controller has been reproduced applying the 0-1 classifications of the PWARX systems. The developed PWARX classifier describes the nonlinear feedback control law of the traffic control system. As we checked in the chapter VII, very good solutions are obtained in a very short

	Computation Time[sec]
Proposed	0.0746
MINLP controller with H=1	0.8095
MINLP controller with H=2	20.5678
MINLP controller with H=3	123.2846
MINLP controller with H=4	286.0026
MINLP controller with H=5	912.6755

Table 4.10: Comparison of computational efforts

time, compared with the one obtained with the conventional MINLP controller.

In the 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. Otherwise, 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 the traffic network control system, successfully reproducing the output of the conventional MINLP controller.

Chapter 5

Conclusions

5.1 Remark

In this paper, we have proposed a new method for the traffic network control system based on hybrid dynamical system theory.

First of all, the synthetic modeling method for the Traffic Flow Control System (TFCS) 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.

The developed non-linear dynamics was formulated based on the Mixed Integer Non-Linear Programming problem, and yields global optimal solution coupled with convexity analysis. Lastly, the proposed Branch and Bound algorithm, which introduced the refining process, enables to minimize the introduced number of the auxiliary variables, whereas the conventional MINLP problems are known as computationally expensive. Some numerical experiments have been carried out, and have shown the usefulness of the proposed design framework.

Thirdly, 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.

Fourthly, the output of the developed controller has been reproduced applying 0-1 classifications of PWARX systems, where the output of the traffic network controller is the 0-1 binary signals. The developed PWARX classifier describes the nonlinear feedback control law of the traffic control system. As we checked in the chapter 4, very good solutions are obtained in a very short time, compared with the one obtained with the conventional MINLP controller. In the classification problem in this paper, very good classification performance is required even with very large number of introduced cluster. 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. Otherwise, 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 in fines. The developed classification method has been applied to the traffic network control system, successfully reproducing the output of the conventional MINLP controller.

5.2 Scope of future research

Our future work is listed as follows

(1) The observer's development for estimating highly-detailed traffic state from limited sensors. The traffic flow observer is considered to underlie the development of fault tolerant traffic system as well as infrastructure cost reduction.

(2) The development of hierarchical modeling and planning schemes, and analytical consideration of stochastically changing traffic network dynamics.

Appendix A

Matrices in MLDS form for Fig.2.1

The matrices in MLDS form for the straight road illustrated in Fig.2.1 are given as follows,

$$
\begin{aligned}
\boldsymbol{A} &= \boldsymbol{I}, && \text{(A.1)} \\
\boldsymbol{B} &= \begin{bmatrix}
0 & 1 & 0 & \cdots & \cdots & 0 \\
0 & 1 & 0 & \ddots & \ddots & \vdots \\
0 & -1 & 1 & \ddots & \ddots & \vdots \\
0 & -1 & 1 & \ddots & \ddots & \vdots \\
0 & -1 & 1 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & -1 & 0
\end{bmatrix}^{T}, && \text{(A.2)} \\
\boldsymbol{C} &= \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
1 & 0 & 0 & \cdots & \vdots \\
1 & 0 & 0 & \cdots & \vdots \\
0 & 1 & 0 & \cdots & \vdots \\
0 & 1 & 0 & \cdots & \vdots \\
0 & 1 & 0 & \cdots & \vdots \\
0 & 0 & 1 & \cdots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & \cdots & 1
\end{bmatrix}, && \text{(A.3)} \\
& && \text{(A.4)}
\end{aligned}
$$

$$\boldsymbol{E1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{C} & \mathbf{0} & \mathbf{0} \end{bmatrix}^T, \tag{A.5}$$

$$\boldsymbol{E2} = \begin{bmatrix} -\boldsymbol{\Gamma} & \mathbf{0} \\ \boldsymbol{\Lambda} & \mathbf{0} \\ -\boldsymbol{\Lambda} & \mathbf{0} \\ \boldsymbol{\Gamma} & \mathbf{0} \\ \boldsymbol{I} & \mathbf{0} \\ \boldsymbol{I} & -\boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{I} \end{bmatrix}, \tag{A.6}$$

$$\boldsymbol{E3} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} & \boldsymbol{I} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^T, \tag{A.7}$$

$$\boldsymbol{E4} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{F} & -\boldsymbol{F} & \mathbf{0} & \mathbf{0} & -\boldsymbol{C} \end{bmatrix}^T, \tag{A.8}$$

$$\boldsymbol{E5} = \begin{bmatrix} 0 \\ 0 \\ h_0 - m \\ h_0 - m \\ h_0 - m \\ h_1 - m \\ \vdots \\ h_5 - m \\ M - h_0 \\ M - h_0 \\ M - h_0 \\ M - h_1 \\ \vdots \\ M - h_5 \\ 0 \\ 0 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{A.9}$$

,where

$$\mathbf{\Gamma} = \begin{bmatrix} M_0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & M_0 & 0 & \ddots & \ddots & \vdots \\ \vdots & 0 & M_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & M_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & M_6 \end{bmatrix}, \tag{A.10}$$

$$\mathbf{\Lambda} = \begin{bmatrix} m_0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & m_0 & 0 & \ddots & \ddots & \vdots \\ \vdots & 0 & m_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & m_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & m_6 \end{bmatrix}, \tag{A.11}$$

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{f}_0^1 & 0 & \cdots & \cdots & \cdots & 0 \\ \boldsymbol{f}_0^2 & 0 & \ddots & \ddots & \ddots & 0 \\ \boldsymbol{f}_0^3 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \boldsymbol{f}_1^1 & 0 & \ddots & \ddots & 0 \\ 0 & \boldsymbol{f}_1^2 & 0 & \ddots & \ddots & 0 \\ 0 & \boldsymbol{f}_1^3 & 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \boldsymbol{f}_5^3 \end{bmatrix}, \tag{A.12}$$

and

$$\boldsymbol{x} = [\ 0 \quad x_1 \quad x_2 \quad \cdots \quad x_5 \quad 0\ ]^T. \tag{A.13}$$

Appendix B

Matrices in MLDS for MINLP

$$\boldsymbol{A} = \boldsymbol{I} \tag{B.1}$$

$$\boldsymbol{B} = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & -1 & 1 & 0 & \ddots & \ddots & \vdots \\ 0 & -1 & 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \tag{B.2}$$

$$\boldsymbol{E_1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{I} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Gamma_1} \end{bmatrix}^T \tag{B.3}$$

$$\boldsymbol{E_2} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{I} & \boldsymbol{I} & \mathbf{0} \end{bmatrix}^T \tag{B.4}$$

$$\boldsymbol{E_4} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^T \tag{B.5}$$

$$\boldsymbol{E_5} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda_x} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda_y} & \boldsymbol{\Gamma_5} \end{bmatrix}^T \tag{B.6}$$

where

$$\boldsymbol{\Lambda_x} = \begin{bmatrix} -x_{max} & -x_{max} & \cdots & -x_{max} \end{bmatrix} \tag{B.7}$$

$$\boldsymbol{\Lambda_z} = \begin{bmatrix} z_{max} & z_{max} & \cdots & z_{max} \end{bmatrix} \tag{B.8}$$

$$\boldsymbol{\Gamma_1} = \begin{bmatrix} \boldsymbol{\gamma_1} & 0 & \cdots & \cdots & 0 \\ 0 & \boldsymbol{\gamma_1} & 0 & \ddots & 0 \\ \vdots & 0 & \boldsymbol{\gamma_1} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \boldsymbol{\gamma_1} \end{bmatrix} \tag{B.9}$$

$$\boldsymbol{\Gamma_5} = \begin{bmatrix} \boldsymbol{\gamma_5} & \boldsymbol{\gamma_5} & \cdots & \boldsymbol{\gamma_5} \end{bmatrix}^T \tag{B.10}$$

$$\boldsymbol{\gamma_1} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \tag{B.11}$$

$$\boldsymbol{\gamma_5} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{B.12}$$

$$\boldsymbol{u} = \begin{bmatrix} u_{E,1} \\ u_{W,1} \\ u_{N,1} \\ u_{S,1} \\ u_{E,2} \\ u_{W,2} \\ u_{N,2} \\ u_{S,2} \\ \vdots \\ u_{S,m} \end{bmatrix} \tag{B.13}$$

Bibliography

[1] H. Hisao, “Considering the traffic flow as the pulverulent body (in japanese),” *Parity*, vol. 13, no. 5, pp. 13–22, 1998.

[2] Y. Kato, “Traffic flow simulation by cellular automaton method,” *JSIE*, vol. 15, no. 2, pp. 242–250, 2000.

[3] T. Tokihiro and J. Satuma, “On the physics hidden in cellular automata - an approach through ultradiscretization-,” *The Physical Society of Japan magazine*, vol. 58, no. 12, pp. 895–902, 2003.

[4] K. Nagel and M. S. A, “Cellular automaton model for freeway traffic,” *Journal de Physique I France*, pp. 2–2221, 1992.

[5] C. P.P *et al.*, “Additive cellular automata -theory and applications,” *IEEE Computer Society Press*, 1997.

[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, p. 281, 1955.

[7] H. J. Payne, “Models of freeway traffic and control in mathematical models of public systems,” *Simulation Council Proceedings Series*, vol. 1, no. 1, pp. 51–61, 1971.

[8] I. Prigogine and R. Herman, *Kinetic Theory of Vehicular Traffic.* Elsevier, New York, 1971.

[9] 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, pp. 297–303, 1986.

[10] R. Kitamura, T. Morikawa, H. Sasaki, S. Fujii, and T. Yamamoto, *Modeling Travel Behavior (in Japanese)*. Gihoudou-shuppan, 2002.

[11] W. Blunden, *THE LAND-USE/TRANSPORT SYSTEM Analysis and Synthesis*. Pergamon Press, Oxford, 1972.

[12] A. Kasahara, *Transportation System Engineering (in Japanese)*. Kyouritsu-shuppan, 1993.

[13] T. Chishaki and N. Inoue, *Traffic Planning (in Japanese)*. Kyouritsu-shuppan, 1993.

[14] S.Darabha and K.R.Rahagopal, “Intelligent cruise control systems and traffic flow stability,” *Transportation Research Part C*, vol. 7, pp. 329–352, 1999.

[15] M. Sugi, H. Yuasa, and T. Arai, “Autonomous distributed control of traffic signal network by reaction-diffusion equation on a graph,” *14th SICE Symposium on Decentralized Autonomous Systems*, pp. 235–238, 2002.

[16] J.-H. Lee and H. Lee-Kwang, “Distributed and cooperative fuzzy controllers for traffic intersections group,” *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 29, no. 2, pp. 263–271, 1999.

[17] A. J and AL-KHALILI, “Urban traffic control - a general approach,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. smc-15, no. 2, pp. 260–271, 1985.

[18] D. E. and O. U., “Decentralized control of traffic networks,” *IEEE Transactions on Automatic Control*, vol. 28, no. 6, pp. 677–688, 1983.

[19] M. C. Choy, S. D., and C. R.L., “Cooperative, hybrid agent architecture for real-time traffic signal control,” *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 33, no. 5, pp. 597–607, 2003.

[20] T. Kato, Y. Kim, and S. Okuma, “Model predictive control of traffic flow based on hybrid system modeling,” *ICCAS*, pp. 368–373, 2005.

[21] T. Kato, Y. Kim, T. Suzuki, and S. Okuma, “Model predictive control of traffic flow based on hybrid system modeling,” *IEICE*, vol. E88-A, no. 2, pp. 549–560, 2005.

[22] G. Ferrari-Trecate, F. A. Cuzzola, and M. Morari, “Analysis of discrete-time pwa systems with logic states,” in *Proceedings of the 5th International Workshop on Hybrid Systems: Computation and Control*, vol. 2289 of *Lecture Notes in Computer Science*, pp. 194–208, Springer, 2002.

[23] A. Bemporad, F. D. Torrisi, and M. Morari, “Optimization-based verification and stability characterization of piecewise affine and hybrid systems,” in *HSCC*, pp. 45–58, 2000.

[24] A. Bemporad, F. Borrelli, and M. Morari, “On the optimal control law for linear discrete time hybrid systems,” in *Proceedings of the 5th International Workshop on Hybrid Systems: Computation and Control*, vol. 2289 of *Lecture Notes in Computer Science*, pp. 105–119, Springer, 2002.

[25] A. Bemporad, G. Ferrari-Trecate, and M. Morari, “Observability and controllability of piecewise affine and hybrid systems,” *In Proceedings of the 38th IEEE Conference on Decision and Control*, pp. 3966–3971, 1999.

[26] Y. Yin and S. Hosoe, “Tracking control of mixed logic dynamical systems,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E87-A, pp. 2929–2936, 2004.

[27] Y. Kim, I. Akio, S. Tatsuya, and O. Shigeru, "Hierarchical scheduling for large-scale production system based on continuous and timed petri net model," *SICE 2002. Proceedings of the 41st SICE Annual Conference*, vol. 1, pp. 268– 271, 2002.

[28] R. L. Grossman, A. Nerode, and A. P. Ravn, *Hybrid Systems*, vol. 736 of *Lecture Notes in Computer Science.* Springer, 1993.

[29] M. Buss, "Hybrid control of mechatronic systems," *IEICE*, vol. 46, no. 3, pp. 129–137, 2002.

[30] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, 1989.

[31] J. Lygeros, D. Godbole, and S. S. Sastry, "A game theoretic approach to hybrid system design," in *Hybrid Systems III. Verification and Control*, vol. 1066 of *Lecture Notes in Computer Science*, pp. 1–12, Springer, 1996.

[32] H. P. Williams, *Model Building in Mathematical Programming.* John Wiley and Sons Ltd, 1993.

[33] T. M. Cavalier, P. M. Pardalos, and A. L. Soyster, "Modeling and integer programming techniques applied to propositional calculus," *Computers and Operations Research archive*, vol. 17, no. 6, pp. 561–570, 1990.

[34] R. R. and I. Grossmann, "Integration of logic and heuristic knowledge in minlp optimization for process synthesis," *Computers & chemical engineering*, vol. 16, no. 3, pp. 155–171, 1992.

[35] 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.

[36] J. H. Lee and B. Cooley, “Recent advances in model predictive control and other related areas,” *AIChE Symposium Series*, vol. 91, no. 316, pp. 201–216, 1997. 5th International Conference on Chemical Process Control.

[37] D. Mayne, “Nonlinear model predictive control: An assessment.,” *Fifth International Conference on Chemical Process Control(CPC V)*, pp. 217–231, 1996.

[38] S. Qin and T. Badgewell, “An overview of industrial model predictive control technology,” *Chemical Process Control - V*, vol. 93, no. 316, pp. 232–256, 1997.

[39] R. R. and I. Grossmann, “Relation between milp modeling and logical inference for chemical process synthesis,” *Computers and Chemical Engineering*, vol. 15, no. 2, pp. 73–84, 1991.

[40] R. Fletcher and S. Leyffer, “Numerical experience with lower bounds for MIQP branch-and-bound,” *SIAM Journal on Optimization*, vol. 8, pp. 604–616, 1998.

[41] R. Lazimy, “Improved algorithm for mixed-integer quadratic programs and a computational study,” *Mathematical Programming*, vol. 32, no. 1, pp. 100–113, 1985.

[42] O. V. Volkovich, V. A. Roshchin, and I. V. Sergienko, “Models and methods of solution of quadratic integer programming problems,” *Cybernetics and Systems Analysis*, vol. 23, no. 3, pp. 289–305, 1987.

[43] R. Fletcher and S. Leyffer, “A mixed integer quadratic programming package,” tech. rep., University of Dundee, 1994.

[44] Y. M. L. and M. M., “Propositional logic in control and monitoring problems,” *Automatica (Automatica)*, vol. 35, no. 4, pp. 565–582, 1999.

[45] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, “A clustering technique for the identification of piecewise affine systems,” in *Hybrid Systems: Computaion and Control*, vol. 2034 of *Lecture Notes in Computer Science*, pp. 218–231, Springer, 2001.

[46] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods.* Cambridge Univ Pr, 2000.

[47] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: A review," *SIGKDD Explorations*, vol. 6, no. 1, pp. 90–105, 2004.

[48] B. F., G. A., and M. 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.

[49] M. Aoyama, N. Uchihira, and K. Hiraishi, *Petri Net theory and its applications (in Japanese).* ISCIE, 1995.

[50] T. Kato, Y. Kim, S. Okuma, , and T. Narikiyo, "Large-scale traffic network control based on convex programming coupled with b&b strategy," *IEEJ Trans.EIS*, vol. 126, no. 6, pp. 752–760, 2006.

[51] M. Morari and E. Zafirou., *Robust Process Control.* Prentice Hall, 1989.

[52] E. Camacho and C. Bordons, *Model predictive control in the process industry.* Springer-Verlag, 1995.

[53] R. Haberman, *Mathematical Models.* Prentice-Hall, 1977.

Acknowledgements

The author considers himself extremely fortunate to be under the guidance of his direct supervisor, Dr. Shigeru Okuma, Professor of graduate school of engineering, Nagoya University. The author wishes to express deepest gratitude for Professor Shigeru Okuma's untiring guidance and unfailing support both in his research and non-academic areas.

The author has been greatly indebted both in his both master's and doctor's programs to Dr. YoungWoo Kim, researcher of Toyota Technological Institute who was the supervisor in both master's and doctor's degrees. Without his assistance, this dissertation would be incomplete

The author also wants to express special thanks to Dr. Tatsuya Suzuki and Dr. Masayuki Tanimoto, Professor of graduate school of engineering, Nagoya University, without whose powerful instruction, this work would not be possible. They generously gave the author many valuable advises.

The author also wishes to express his profound gratitude to Dr. Morio Takahama, Professor of graduate school of information science, Nagoya University, who supported the author in every possible way.

The author would like to express heartfelt thanks to Dr. Shinji Doki, Associate Professor of graduate school of engineering, Nagoya University.

The author wishes to acknowledge support and help of all members of Okuma laboratory for their contribution in various ways to this dissertation.The author renew his appreciation to all the members hybrid group which include Dr.Eiji Konaka and Dr.Hayato Komatsu, Dr.Kazuaki Hirana etc. for their advices and discussions.

Finally, expressions of gratitude are directed to author's beloved family. Without their support and encouragement, this dissertation would never be bought to fruition.

Tatsuya KATO, The Author