DECENTRALIZED MODEL PREDICTIVE CONTROL OF URBAN DRAINAGE SYSTEMS

BY

YICHUAN LI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Advisers:

Professor Petros G. Voulgaris
Professor Dusan M. Stipanovic

ABSTRACT

This thesis applies a modeling method proposed by previous literature to a specific example and develops control techniques based on this model. The inherent nonlinear behaviors of the drainage systems were accommodated by introducing binary variables and linear inequalities to merge different modes of operation into a single expression. The objective function is constituted by 3 cost functions considering several priorities. Except the normal objective of minimizing overflows, we present two methods of reducing operation costs by harvesting rain power and energy from rainfall collecting locales. *Pressure forebay* regulates water with large potential energy and generates electricity as the water is directed through hydraulic pumps. *Surface aqueduct* collects water with high kinetic energy and pushes water through spiral case to generate electricity. Locations with these devices installed induce lower operation cost and thus have higher priorities to be utilized. Once we formulate the water management problem as an optimization problem with specified constraints, we can apply Model Predictive Control (MPC) to compensate for modeling errors and prediction inaccuracies. As we regularly update system states and disturbances information, we achieve our goal of applying real time control to drainage systems.

As the system size grows, the system is partitioned into several subsections and each one of them forms a subsystem which makes local decisions based on partial information. The performance of different partition schemes was compared against centralized MPC and open loop controllers. It was shown that even decentralized controllers may suffer performance loss, the computation time was significantly reduced compared with centralized controllers. In rain scenarios with large intensity, the performance loss of decentralized controllers is insignificant compared with the advantage gained by computation time reduced.

ACKNOWLEDGEMENTS

I would like to thank my advisors, Professor Petros Voulgaris and Professor Dusan Stipanovic for their invaluable guidance and encouragement. I enjoyed every meeting with them as each meeting was delightful and enlightening. This work would not be possible without the help of any of them. I feel most honored to have the opportunity to learn from Petros and continue my Ph.D. studies under his guidance.

I would also want to thank my parents for their unconditional love and support. They always let myself to choose what I want and encourage me to chase my own dreams. They taught me to work hard but also to enjoy life, to dream great but also to stand firm on ground.

I would also like to thank all the friends, colleagues and classmates during the past several years. My years in Champaign and Urbana were most enjoyable because of all of you.

Finally, I would like to thank the financial support from ZJUI institute and the opportunity to start this project.

To my parents and loved ones

TABLE OF CONTENTS

Chapter 1

Introduction

Various studies indicate that extreme weather conditions leading to increased flooding frequency and severity may continue in the near future [1,2]. A recent study estimated that the global cost of flooding in the world's 136 largest cities could rise to $52 billion a year by 2050, a significant increase from $6 billion in 2005 [3]. Urban drainage system collects rainfall and waste water from all parts of the city using open canals and sewer pipes. Then the water is transported through interceptors, weirs and main sewer pipes into temporary storage tanks and water treatment plant before it is released to the environment. When severe regional rain storm occurs, large volume of water can easily overload parts of the system and excess water is released to the nearest receiving environment. The excess discharge of rainfall along with untreated waste water, known as *Combined Sewage Overflow* (CSO), endangers city infrastructure and contains biological and chemical contaminant which brings significant hazard to public health. The associated social, economical and environmental costs have lead to increased interest in the management of urban sewage system. A prevalent solution to the CSO problem is to enhance existing drainage system by increasing sewer volume and water treatment plant capacity. Examples include the Tunnel and Reservoir Plan (TARP) in Chicago [15] and *Escola Industrial* reservoir in Barcelona [12]. To take the profit of these expensive infrastructures, it is necessary to apply real-time control (RTC) techniques which can efficiently utilize the total storage volume and avoid overflowing in parts of sewage system while other parts operating under capacities. *Model Predictive Control* (MPC), also refereed as *Receding*

Horizon Control, has proven to be one of the most effective and successful control schemes for large interconnected systems [4], [5]. The ability to incorporate multi-objectives and large-scale system of MPC makes it can be easily applied to urban sewage system which is hierarchical and distributed in nature.

To use MPC within the framework of sewage RTC, suitable models that can capture internal dynamics of sewage connection is needed. The open-canal flow dynamics is described by Saint-Vennant's partial differential equations which can be used to simulate the flow conditions inside sewer pipes. However, these differential equations are highly complex and difficult to solve in a timely manner. The details provided by these differential equations are unnecessary to our purpose and therefore not suitable for our formulation of the problem. For our control-oriented characterization, the model should be descriptive enough to capture mass-balance dynamics and easy to compute in a fashion so that it is scalable to large-complex systems. There exit several modeling techniques in current literature that use linear models to represent the system [6]. Such formulation preserves the convexity of the optimization problem and therefore well-established optimization techniques can be employed to design control strategies. However, there are inherent dynamics in sewage system which possess different operation mode depending on the states of the system. These nonlinear behaviors can not be neglected nor described by a linear model. A modeling approach and its application to sewage systems are presented in Chapter 2. Control formulation of the problem is introduced in Chapter 3. Objective and optimization formulation relating to MPC is also discussed in this chapter. Chapter 4 presents the formulation of MPC in decentralized scheme. Several decentralizing schemes can be designed depending on the topography and components coupling. Simulation results demonstrating both centralized and decentralized control strategies are presented in Chapter 5. Conclusions and future work are presented in Chapter 6.

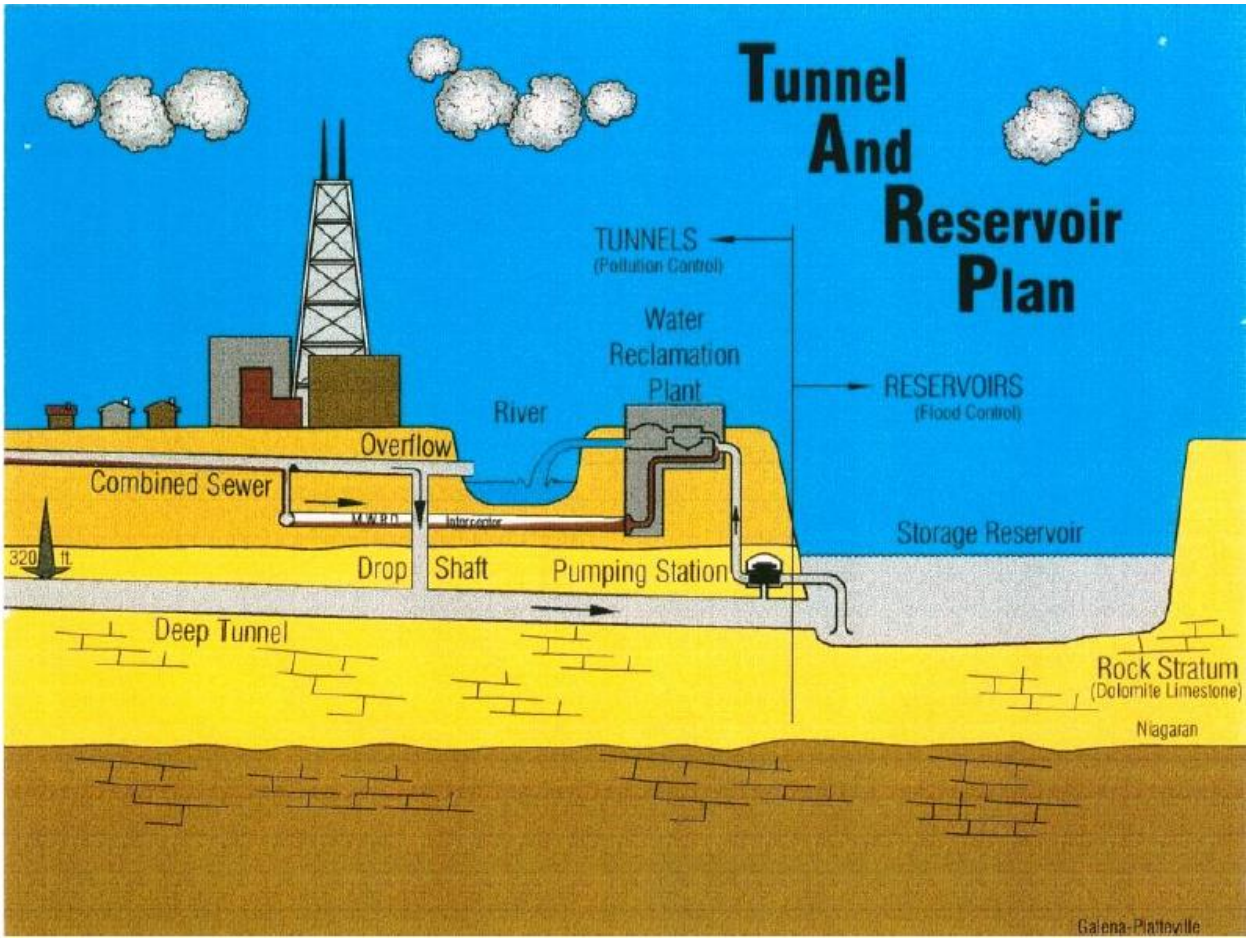

Figure 1 Chicago TARP system [15].

Chapter 2

Modeling

To capture the nonlinear behavior of the system discussed earlier, we present a modeling technique involving logical variables [7]. This technique allows us to include both continuous dynamics of the system under normal operation mode and discrete dynamics which are usually associated with the event of overflowing. For example, when large volume of flow occurs during high intensity of rainfall, excess water is released into the receiving environment and extra flow path is created. Due to the mixture of these continuous and discrete dynamics, we present *Mixed Logical Dynamical* (MLD) systems introduced in [8] and its application to our system in following sections.

2.1 Logical variables

Control models are often described as a set of differential or difference equation derived from physical laws governing the evolution of the system of interest. Therefore, most of the control models are linear or nonlinear but continuous transition functions. However, in many engineering applications, the system to be controlled involves discrete dynamics introduced by switches, valves, gates and gears etc. These elements give rises to a class of system named *hybrid systems*. Following formulation introduces logical variables that correspond to on/off events by constraining the function involved with logical variables and therefore forcing the logical variables to take certain value when certain events happens.

Consider the following system:

$$\begin{aligned} x(t+1) &= \ \ 0.8x(t)+u(t) \quad \text{if} \quad x(t)\ge 0 \\ x(t+1) &= -0.8x(t)+u(t) \quad \text{if} \quad x(t)<0 \end{aligned} \tag{1}$$

where $x(t)\in[-10,10]$, and $u(t)\in[-1,1]$. The logical event of $x(t)\ge 0$ can be described using a binary variable $\delta(t)$ as follows:

$$\delta(t)=1 \Leftrightarrow x(t)\ge 0 \tag{2}$$

And introduce following constraints:

$$\begin{aligned} &-m\delta(t)\le x(t)-m \\ &-(M+\varepsilon)\delta(t)\le -x(t)-\varepsilon \end{aligned} \tag{3}$$

where M(m) is the maximum(minimum) of the state and in this case $M=-m=10$. ε is a small positive scalar corresponds to machine accuracies. Theoretically, an overestimate(underestimate) suffice for our purpose but tighter bounds provide computational advantages. Eq. (1) can therefore be rewritten as

$$x(t+1)=1.6\delta(t)x(t)-0.8x(t)+u(t) \tag{4}$$

By introducing an auxiliary variable $z(t)=\delta(t)x(t)$ which is constrained as follows:

$$\begin{aligned} &z(t)\le M\delta(t) \\ &z(t)\ge m\delta(t) \\ &z(t)\le x(t)-m(1-\delta(t)) \\ &z(t)\ge x(t)-M(1-\delta(t)) \end{aligned} \tag{5}$$

Then system (1) can be described by the following linear equation

$$x(t+1)=1.6z(t)-0.8x(t)+u(t) \tag{6}$$

subject to above introduced constraints.

For logical variables defined in terms of other logical variables:

$$\delta_3(t)=\delta_1(t)\delta_2(t) \tag{7}$$

introduce following constraints:

$$\begin{aligned} &-\delta_1(t)+\delta_3(t)\le 0 \\ &-\delta_2(t)+\delta_3(t)\le 0 \\ &\delta_1(t)+\delta_2(t)-\delta_3(t)\le 1 \end{aligned} \tag{8}$$

This example can be generalized to the MLD systems as shown below:

$$x(t+1)=Ax(t)+B_1u(t)+B_2\delta(t)+B_3z(t)+B_4d(t) \tag{9}$$

$$y(t)=Cx(t)+D_1u(t)+D_2\delta(t)+D_3z(t) \tag{10}$$

$$E_2\delta(t)+E_3z(t)\le E_1u(t)+E_4x(t)+E_5 \tag{11}$$

where $t\in\mathbb{Z}$,and

$$x=\begin{bmatrix} x_c \\ x_l \end{bmatrix},\ x_c\in\mathbb{R}^{n_c},\ x_l\in\{0,1\}^{n_l},\ n\triangleq n_c+n_l$$

collects the continuous states x_c and binary states x_l.

Similarly,

$$y=\begin{bmatrix} y_c \\ y_l \end{bmatrix},\ y_c\in\mathbb{R}^{p_c},\ y_l\in\{0,1\}^{p_l},\ p\triangleq p_c+p_l$$

denotes the system output.

$$u=\begin{bmatrix} u_c \\ u_l \end{bmatrix},\ u_c\in\mathbb{R}^{m_c},\ u_l\in\{0,1\}^{m_l},\ m\triangleq m_c+m_l$$

denotes the system's continuous and binary inputs.

Equations (9)(8)(9) describe a linear time-invariant discrete dynamical system. Continuous dynamics can be obtained by replacing the difference equation with the governing differential equation. Time dependence can be introduced by prescribing time dependence of the system matrix.

2.2 Drainage network dynamics

There exist several modeling techniques that deal with sewage system control. The modeling methods presented here follow closely from [9], [10]. In this framework, the sewage system is divided into several catchments according to geography and their couplings with each other. Each one of them will be represented as a virtual tank which aggregates the total storage volume of a given neighborhood of sewage. The total volume can be computed by the mass balance of the inflows, outflows and stored volume of rainfall [9], [10]. Some other elements of sewage systems can be incorporated into this frame work easily such as detention tanks, diversion gates, nodes and weirs. We demonstrate above molding techniques with some typical elements of sewage system.

2.2.1 Virtual and real Tanks

The virtual tanks mentioned earlier represents the basic storing element of each neighborhood. Considering the mass balance of inflow and outflow, we express the virtual tank dynamics as follow:

$$v_n(k+1) = v_n(k) + \Delta t \varphi_n S_n P_n(k) + \Delta t(q_n^{in}(k) - q_n^{out}(k) - q_n^d(k)) \quad (12)$$

where Δt is the sample time, φ_n is the ground absorption coefficient, S_n is the surface area and $P_n(k)$ is the rain intensity at k th sample time. $q_n^{in}(k)$ represents the combined input of manipulated flows and sewer flows into the corresponding tank. Tank outflows are assumed to be proportional to the tank volumes and represented as follows:

$$q_n^{out}(k) = \beta_n v_n(k) \quad (13)$$

where β_n is defined as the volumetric flow coefficient suggested in [11].

Virtual tanks do not have physical upper limits and when excess water is redirected to other parts of sewage system or to the nearest receiving environment. When overflow occurs, extra flow path is created and is denoted as q_n^d in Eq. (12), which can be explicitly expressed as shown below:

$$q_n^d(k) = \begin{cases} \dfrac{v(k) - \bar{v}}{\Delta t} & \text{if } v(k) > \bar{v} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where $\bar{v}$ is the maximum capacity of the corresponding tank.

Therefore, the outflow in Eq. (13) can be further expressed as follows:

$$q_n^{out}(k) = \begin{cases} \beta_n \bar{v}_n & \text{if } v(k) > \bar{v} \\ \beta_n v_n(k) & \text{otherwise} \end{cases} \tag{15}$$

Next step would be to introduce logical variables to integrate the dynamics described by above equations. Using Eq. (2),22 we express the overflow at the first tank as an example:

$$\delta_1 = 1 \Leftrightarrow v_1 - \bar{v}_1 \geq 0 \tag{16}$$

constraints will be introduced using Eq. (3)

$$\begin{aligned} -m_1 \delta_1(k) &\leq v_1(k) - \bar{v}_1 - m_1 \\ -(M_1 + \varepsilon)\delta_1(k) &\leq -v_1(k) + \bar{v}_1 - \varepsilon \end{aligned} \tag{17}$$

where the lower bound and upper bound is defined as follows:

$$\begin{aligned} m_1 &= -\bar{v}_1 \\ M_1 &= v_1^{upper} - \bar{v}_1 \end{aligned} \tag{18}$$

v_1^{upper} corresponds to the maximum possible value for virtual tank 1 can take for the horizon under consideration which is a constant for each optimization iteration.

$$v_1^{upper} = 2\bar{v}_1 + S_1 \Delta t \max(P) \varphi_1 \tag{19}$$

When there is overflow occurring, excess water is released to the environment at the end of the sample time. The coefficient 2 ensures that when consecutive high intensity rain episodes occur, the optimization problem to be formulated in the end is feasible. The $\max(P)$ corresponds to the highest intensity of rainfall over the horizon and therefore the latter part of Eq. (19) represent the maximum rainfall input to the tank of interest.

Eq. (14) can therefore be expressed as

$$q_1^d(k) = \frac{v_1(k) - \bar{v}_1}{\Delta t}\delta_1(k) = \frac{1}{\Delta t}(z_1 - \bar{v}_1\delta_1(k)) \tag{20}$$

where $z_1(k) = v_1(k)\delta_1(k)$ and is similarly constrained as Eq. (5)

$$\begin{aligned} &z_1(k) \le v_1^{upper}\delta_1(k) \\ &z_1(k) \ge 0 \\ &z_1(k) \le v_1(k) \\ &z_1(k) \ge v_1(k) - v_1^{upper}(1 - \delta_1(k)) \end{aligned} \tag{21}$$

Eq. (15) is expressed as

$$q_1^{out}(k) = \beta_1\bar{v}_1\delta_1(k) + \beta_1 v_1(k)\cdot(1 - \delta_1) \tag{22}$$

which can be expanded as

$$q_1^{out}(k) = \beta_1\bar{v}_1\delta_1(k) + \beta_1 v_1(k) - \beta_1 z_1(k) \tag{23}$$

with the same constraints for logical variables as introduced in Eq. (17) and Eq. (21).

If there are no other inflows from other tanks to tank 1, the difference equation for tank 1can be expanded from Eq. (12) as follows:

$$v_1(k+1) = (1 - \Delta t\beta_1)v_1(k) + (\Delta t\beta_1 - 1)z_1(k) + (1 - \Delta t\beta_1)\bar{v}_1\delta_1 + \Delta t\varphi_1 S_1 P_1(k) \tag{24}$$

Real tanks represent the storage elements in the sewage system such as reservoirs and detention gates and do not receive rainfall input from the environment. Note that real tanks are considered to be without overflowing capabilities and therefore the upper limit capacities are physical constraints and no more water can be sent into real tanks than their maximum capacities. Therefore, inflows to real tanks should be pre-manipulated to ensure these constraints are always respected. For real tanks, Eq. (12) is simplified to be as

$$v_{real}(k+1) = v_{real}(k) + \Delta t(q_{real}^{in}(k) - q_{real}^{out}(k)) \tag{25}$$

with pre-manipulated inflows constrained as follow:

$$\Delta t(q_{real}^{in}(k) - q_{real}^{out}(k)) < \overline{v}_{real} - v_{real}(k) \tag{26}$$

since real tanks are without overflowing capabilities, the outflow can be expressed as

$$q_{real}^{out}(k) = \beta_{real} v_{real}(k) \tag{27}$$

2.2.2 Controlled gates and sewer dynamics

In sewer networks, diversion gates are used to divert flows to desired locations so that sewage is appropriately distributed in the system. Detention gates are used to temporally stop the flows at certain locations. Real tanks are often equipped with detention gates to control the outflow. When water is discharged from the tanks, it is transported through canals, sewer pipes, weirs and interceptors to other parts of the system. It is possible that the discharged water exceeds the capacity of sewer pipes so that overflows occur at these elements. From mass balance law, the sum of water in sewer pipes and manipulated flows equals the outflows from the tanks:

$$q^{out}(k) = \sum q_s + \sum q_u \tag{28}$$

where q_s denotes the water in default sewer path and q_u denotes the manipulated flows.

In following formulations, we assume there is one default sewer path and one manipulated flow path.

We further express the default sewer path as below:

$$q_i^s(k) = \begin{cases} q^{out} - q^u & \text{if } q_i^{out} - q_i^u \le \bar{q}_i^s \\ \bar{q}_i^s & \text{otherwise} \end{cases} \tag{29}$$

where $\bar{q}_s$ is the sewer pipe capacity.

We demonstrate sewer path dynamics modeling with the following example:

$$\delta_1^s = 1 \Leftrightarrow q_1^{out} - q_1^u - \bar{q}_1^s \ge 0 \tag{30}$$

Eq. (29) can therefore be applied and rewritten as

$$q_1^s = \bar{q}_1^s \delta_1^s + (q_1^{out} - q_1^u)(1 - \delta_1^s) \tag{31}$$

Introduce auxiliary logical variables:

$$\begin{aligned} z_{uS11} &= q_1^u \delta_1^s \\ z_{S11} &= v_1 \delta_1^s \\ \delta_{S11} &= \delta_1^s \delta_1 \\ z_{S111} &= v_1 \delta_{S11} \end{aligned} \tag{32}$$

where constraints are introduced using Eq. (17) and Eq. (21) as follows:

$$\begin{aligned} & z_{uS11}(k) \le \bar{q}_1^u \delta_1^s(k) \\ & z_{uS11}(k) \ge 0 \\ & z_{uS11}(k) \le q_1^u(k) \\ & z_{uS11}(k) \ge q_1^u(k) - \bar{q}_1^u + \bar{q}_1^u \delta_1^s \end{aligned} \tag{33}$$

where $\bar{q}_1^u$ is the upper limit for manipulated flows.

$$\begin{aligned} & z_{S11}(k) \le v_1^{upper} \delta_1^s(k) \\ & z_{S11}(k) \ge 0 \\ & z_{S11}(k) \le v_1(k) \\ & z_{S11}(k) \ge v_1(k) - v_1^{upper} + v_1^{upper} \delta_1^s \end{aligned} \tag{34}$$

where v_1^{upper} is same as Eq. (19).

Using Eq. (8) to constraint δ_{S11}

$$\begin{aligned}
&-\delta_1(k)+\delta_{S11}(k)\le 0\\
&-\delta_1^s(k)+\delta_{S11}(k)\le 0\\
&\delta_1(k)+\delta_1^s(k)-\delta_{S11}(k)\le 1
\end{aligned} \tag{35}$$

Therefore, the constraints for z_{S111} can be expressed by replacing δ_1^s with δ_{S11} in Eq. (34)

$$\begin{aligned}
&z_{S111}(k)\le v_1^{upper}\delta_{S11}(k)\\
&z_{S111}(k)\ge 0\\
&z_{S111}(k)\le v_1(k)\\
&z_{S111}(k)\ge v_1(k)-v_1^{upper}+v_1^{upper}\delta_{S11}
\end{aligned} \tag{36}$$

Eq. (31) can be expanded with above expression substituted and Eq. (23) as

$$q_1^s=\overline{q}_1^s\delta_1^s+\beta_1\overline{v}_1\delta_1+\beta_1 v_1-\beta_1 z_1-q_1^u-\beta_1\overline{v}_1\delta_{S11}-\beta_1 z_{S11}+\beta_1 z_{S111}+z_{uS11} \tag{37}$$

note that the purpose of introducing auxiliary logical variables is to ensure the linear dependence of the expression. With above compositional elements we collect variables into matrices with suitable dimensions and express the sewage system dynamics in *MLD* form as in Eq. (9).

$$x(t+1)=Ax(t)+B_1u(t)+B_2\delta(t)+B_3z(t) \tag{9}$$

2.3 Constraints formulation

In previous sections, we presented the constraints associated with logical variables. These constraints ensure the exact correspondence of the logical variables with the designated events and δ_i are binary variables taking values 0 and 1 while z_i take values either zero or the state values they correspond to. Now we present the physical constraints such as operation range of controlled flow, mass balance of inflows and outflows and pre-manipulated flows. We demonstrate these constraints with a specific example shown below [12]. The system is composed of three virtual tanks and one real tank. The sewer paths are composed by three controlled flows and four default sewer paths. One waste water treatment plant will be used to collect sewage.

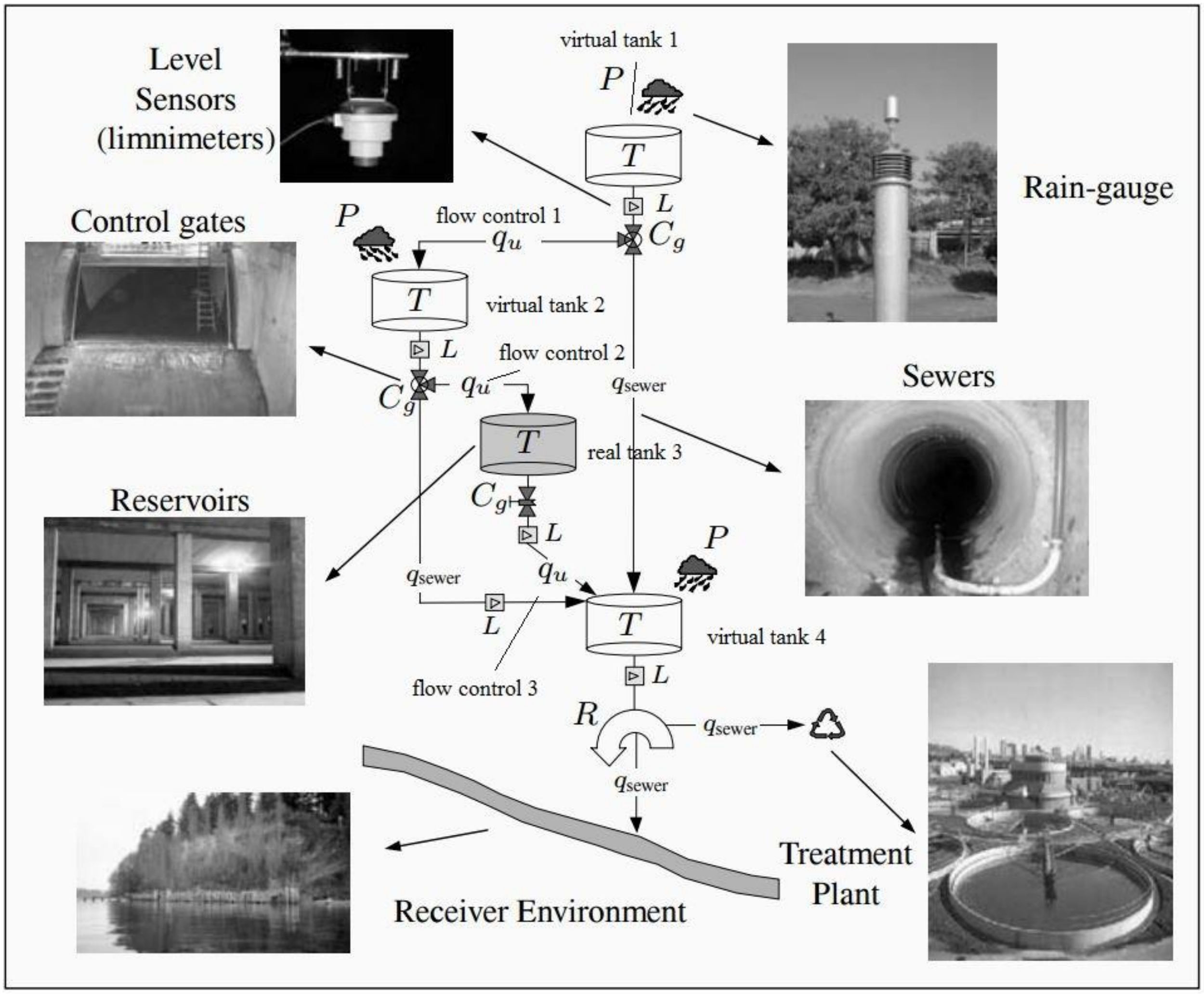

Figure 2 Sewer network.

Controlled flows are constrained by the amount of water available for operation, namely, the discharged water from tanks. We focus on the flow control 2 in the above figure.

$$q_2^u \le q_2^{out} \tag{38}$$

where the outflow can be further expressed using Eq. (23)

$$q_2^u \le \beta_2 \bar{v}_2 \delta_2(k) + \beta_2 v_2(k) - \beta_2 z_2(k) \tag{39}$$

The upper limit of controlled flows will be included later in the decision variables range in Eq. (62).

Note that since flow control 2 direct inflows to real tank 3 which is considered to be without overflowing capabilities. The inflow therefore has to be pre-manipulated to ensure the total volume do not exceed the real tank physical capacity as considered in Eq. (26).

$$\Delta t(q_2^u(k) - q_3^u(k)) \leq \overline{v}_3 - v_3(k) \tag{40}$$

when there are multiple inflows into a single real tank, the above constraint must include all the inflows and outflows and the feasibility of the optimization problem to be formulated later may be unsatisfied if the outflows are not large enough.

Note that all the constraints are expressed linearly in terms of controlled flows q_i^u, logical binary variables δ_i^s and δ_i, auxiliary logical variables z_i^s and z_i, system states variables v_i and constants. We collect these variables into vectors of appropriate dimension and express it in the form of Eq. (11)

$$E_2\delta(t) + E_3 z(t) \leq E_1 u(t) + E_4 v(t) + E_5 \tag{11}$$

For the system in Figure , we express the system dynamics as follows:

$$v(k+1) = Av(k) + B_1 u(k) + B_2\delta(k) + B_3 z(k) + B_4 d(k) \tag{41}$$

where

$$\begin{aligned}
&v(k) = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad u(k) = \begin{bmatrix} q_1^u \\ q_2^u \\ q_3^u \end{bmatrix} \quad d(k) = \begin{bmatrix} P_1 \\ P_2 \\ P_4 \end{bmatrix} \\
&\delta(k) = \begin{bmatrix} \delta_1 & \delta_2 & \delta_3 & \delta_4 & \delta_1^S & \delta_2^S & \delta_{11}^S & \delta_{22}^S \end{bmatrix}^T \\
&z(k) = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_{S11} & z_{S22} & z_{S111} & z_{S222} & z_{uS11} & z_{uS22} \end{bmatrix}^T
\end{aligned} \tag{42}$$

System matrices:

$$A=\begin{bmatrix} 1-\Delta t\beta_1 & 0 & 0 & 0 \\ 0 & 1-\Delta t\beta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \Delta t\beta_1 & \Delta t\beta_2 & 0 & 1-\Delta t\beta_4 \end{bmatrix} \tag{43}$$

$$B_1=\begin{bmatrix} 0 & 0 & 0 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & -\Delta t \\ -\Delta t & -\Delta t & \Delta t \end{bmatrix} \tag{44}$$

$$B_2=\begin{bmatrix} (1-\Delta t\beta_1)\overline{v_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & (1-\Delta t\beta_2)\overline{v_2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta t\beta_1\overline{v_1} & \Delta t\beta_2\overline{v_2} & 0 & (1-\Delta t\beta_4)\overline{v_4} & \Delta t\overline{q_1^s} & \Delta t\overline{q_2^s} & -\Delta t\beta_1\overline{v_1} & -\Delta t\beta_2\overline{v_2} \end{bmatrix} \tag{45}$$

Multiplied by

$$\begin{bmatrix} \delta_1 & \delta_2 & \delta_3 & \delta_4 & \delta_1^S & \delta_2^S & \delta_{S11} & \delta_{S22} \end{bmatrix}^T \tag{46}$$

$$B_3=\begin{bmatrix} \Delta t\beta_1-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Delta t\beta_2-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\Delta t\beta_1 & -\Delta t\beta_2 & 0 & \Delta t\beta_4-1 & -\beta_1\Delta t & -\beta_2\Delta t & \beta_1\Delta t & \beta_2\Delta t & \Delta t & \Delta t \end{bmatrix} \tag{47}$$

Multiplied by

$$\begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_{S11} & z_{S22} & z_{S111} & z_{S222} & z_{uS11} & z_{uS22} \end{bmatrix}^T \tag{48}$$

$$B_4=\begin{bmatrix} \Delta t\varphi_1 S_1 & 0 & 0 \\ 0 & \Delta t\varphi_2 S_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Delta t\varphi_4 S_4 \end{bmatrix} \tag{49}$$

Chapter 3

Control Methods and Optimization

3.1 Model Predictive Control

Model Predictive Control (MPC) is often used in process control such as chemical plants, oil refineries and power electronics. With a dynamics model that is usually linear and empirical derived from system identification, MPC optimizes the prescribed cost function over some finite horizon constituted by several sampling time steps. Once the control strategies are computed for the whole horizon, only the first set of control actions are implemented and then the controller solves the optimization problem again with shifted horizon and it is therefore also known as *Receding Horizon Control*. MPC allows updated information to be incorporated at the start of every optimization iteration and therefore is able to anticipate future events and taking actions accordingly. This approach is not necessarily optimal but its ability to allow for system modeling error and the effectiveness of considering multi-objectives have led to great results in practice.

For sewage systems, *MPC* is used to compute the decision variables, the controlled flows, ahead of time according to a set of control goals expressed as cost functions with possibly different priorities. These computed control goals are then can be achieved by local PID controllers at each part of the sewage system.

3.2 Cost function formulation

Each sewage system can have different objectives depending on the size of the system, local environment, city infrastructure level and operator goals. We discuss some common goals

[7] for urban sewage system and its formulation in the frame work of *MLD* systems. Other objectives can be easily achieved by using similar techniques.

3.2.1 Overflow in virtual tanks and sewer paths

Virtual tanks often represent open canals, sewage pipes in certain neighborhoods of urban environment. These elements can easily be overloaded during extreme weather conditions and the presence of wastewater in sewer pipes makes the minimization of virtual tanks as our first priority. For virtual tanks:

$$J_1 = \begin{cases} \sum z_i - \bar{v}_i & \text{if } \delta_i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{50}$$

which can be simplified as

$$J_1 = \sum z_i - \bar{v}_i \delta_i \tag{51}$$

For sewer pipes:

$$J_2 = \begin{cases} \sum q_i^{out} - q_i^u - \bar{q}_i^S & \text{if } \delta_i^S = 1 \\ 0 & \text{otherwise} \end{cases} \tag{52}$$

Similarly, the above expression can be simplified as:

$$J_2 = \sum (q_i^{out} - q_i^u - \bar{q}_i^S)\delta_i^S \tag{53}$$

Take default sewer path 1 in Figure () as an example, we further expand the above expression using Eq. (23) and Eq. (32) as follows:

$$\text{Sewer path 1 cost} = (-\bar{q}_1^S \delta_1^S + \beta_1 \bar{v}_1 \delta_{S11} + \beta_1 z_{S11} - \beta_1 z_{S111} - z_{uS11})\Delta t \tag{54}$$

3.2.2 Pumping costs from manipulated flows

The controlled flows in sewage system usually require hydraulic pumps to achieve desired volumetric flows at certain locations. The associated pumping cost is therefore directly related to the amount of water pumped. In this section, we mainly introduce an engineering application to reduce electricity costs by harvesting rainwater energy [13].

We present two ways of generating electricity and thus reducing pumping costs using rainwater energy. First approach is characterized as *Pressure forebay* which utilizes rainwater collected from high altitudes such as building roof top. The water collected will be transported through pipes into pressure regulating bay. When enough water is collected and reaches operating level, the valve is opened, and water pushes through hydraulic generator with constant pressure. The water will then be directed to reservoirs in lower altitudes. When collected water flow is not sufficient for generator's operating conditions, the valve will be closed. Pressure forebay thus function as both temporary storage element and as a pressure regulating device.

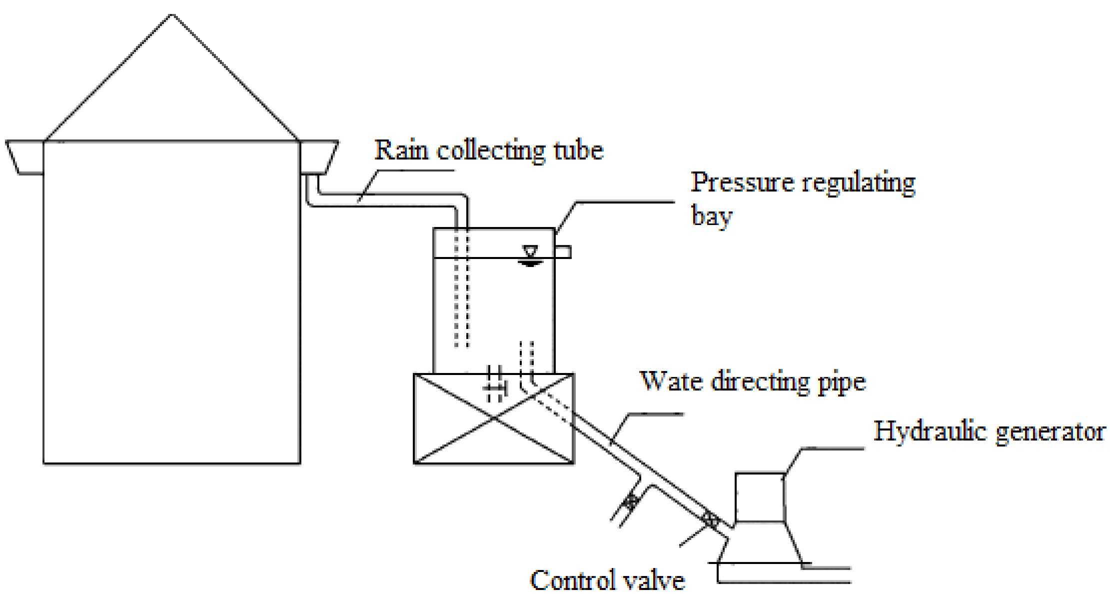

Figure 3 Pressure forebay.

Second approach is characterized as *Surface aqueduct* in which water with comparatively large kinetic energy is collected and directed through collecting tubes into surface aqueduct. Once it is regulated through valves, water enters the spiral case and pushes through turbine to generate electricity. The after water is then directed to near storage elements for further recycling use. When water exceeds the limit capacity of aqueducts and spiral cases, the excess water will be directly discharged through overflow holes into storage elements mentioned earlier. When the volumetric flow is deficient for operating, the valve will be closed, and the generator stops working.

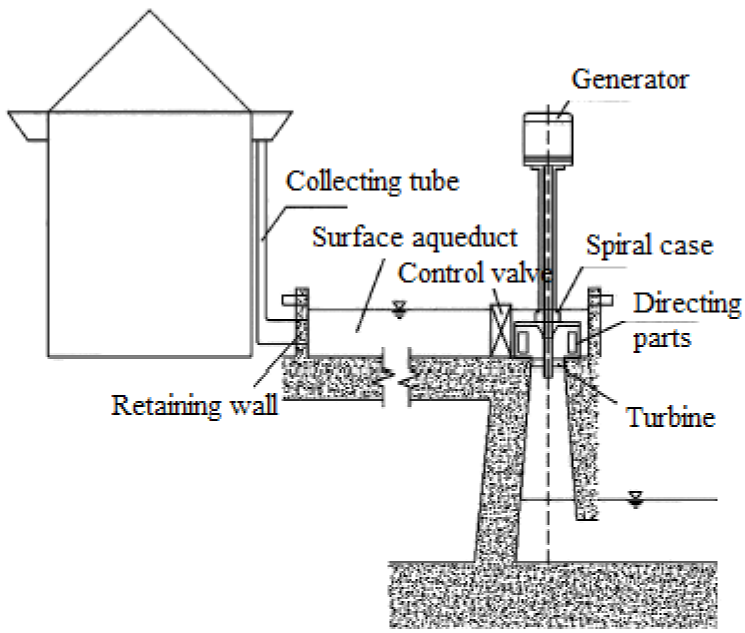

Figure 4 Surface aqueduct.

Generating electricity using rain energy shares a lot of similarities as normal hydraulic power generating scheme. In essence they all utilize radiation energy from the sum but normal hydraulic power generating requires terrain altitude difference but the approaches we presented here are more suitable for urban environment. Since the amount of electricity generated is directly related to the waterhead in reservoirs, high buildings have significant advantages and it is therefore preferable to implement these applications on high buildings. It is also advantageous to connect buildings through rain water pipes and form up networks to increase rain water collecting area and increase power output.

With the above characterization of pumping costs, we express the cost function for controlled flows as

$$J_3 = \sum C_i(P_i) q_i^u \Delta t \tag{55}$$

where $C_i(P_i)$ is a cost coefficient for pumping and varies for each neighborhood and since the electricity generated is directly related to rain water, it will be a function of local precipitation.

3.3 Optimization problem formulation

We now move on to express the sewage network management as an optimization problem with controlled flows as decision variables along with the physical constraints of operation range for controlled flows, mass balance law at junctions and constraints introduced along with logical variables. Using the state equation for discrete dynamics,

$$v(k) = A^k v_0 + \sum_{i=0}^{k-1} A^i [B_1 u(k-1-i) + B_2 \delta(k-1-i) + B_3 z(k-1-i) + B_4 d(k-1-i)] \tag{56}$$

we can express the states at arbitrary step time as a function of initial states, control actions and logical variables in previous sample time. We demonstrate it with a specific example when $k = 3$. From the constraint formulation Eq. (11),

$$E_2 \delta(3) + E_3 z(3) \le E_1 u(3) + E_4 v(3) + E_5 \tag{11}$$

Expand the states variable using Eq. (56)

$$\begin{aligned} v(3) = {} & A^3 v(0) + A^2 B_1 u(0) + A^2 B_2 \delta(0) + A^2 B_3 z(0) + A^2 B_4 d(0) + AB_1 u(1) + AB_2 \delta(1) \\ & + AB_3 z(1) + AB_4 d(1) + B_1 u(2) + B_2 \delta(2) + B_3 z(2) + B_4 d(2) \end{aligned} \tag{57}$$

Substitute in Eq. (11)

$$\begin{aligned} & -E_4 A^2 B_1 u(0) - E_4 A^2 B_2 \delta(0) - E_4 A^2 B_3 z(0) - E_4 AB_1 u(1) - E_4 AB_2 \delta(1) - E_4 AB_3 z(1) - E_4 B_1 u(2) \\ & -E_4 B_2 \delta(2) - E_4 B_3 z(2) - E_1 u(3) + E_2 \delta(3) + E_3 z(3) \le E_4 A^3 v(0) + A^2 B_4 d(0) + AB_4 d(1) \\ & + B_4 d(2) + E_5 \end{aligned} \tag{58}$$

Observe that left hand side of Eq. (58) is composed entirely of $u(i)$, $\delta(i)$ and $z(i)$ and right hand side is composed of rainfall, constant terms and the initial state vector, therefore we collect them into one vector

$$\begin{aligned} &\alpha \triangleq \left[u(0)...u(k)\right]^T \\ &\beta \triangleq \left[\delta(0)...\delta(k)\right]^T \\ &\psi \triangleq \left[\mathrm{z}(0)...\mathrm{z}(k)\right]^T \\ &\gamma = \left[\alpha, \beta, \psi\right]^T \end{aligned} \tag{59}$$

and generalize the expression for Eq. (11) as

$$F_1\gamma \leq F_2 + F_3 v_0 \tag{60}$$

With cost functions defined as Eq. (51), Eq. (53) and Eq. (55), we formulate the optimization problem as follows:

$$\begin{aligned} \min \quad & c_1 J_1 + c_2 J_2 + c_3 J_3 \\ s.t \quad & F_1\gamma \leq F_2 + F_3 v_0 \\ & \gamma_l \leq \gamma \leq \gamma_u \end{aligned} \tag{61}$$

since overflows in different areas have different extent of impacts on environment, urban infrastructure and social activity, we can associate different weight coefficients such as c_1, c_2 and c_3 depending on our priorities. Since all the cost functions depends linearly on γ, we obtain the following *Mixed Integer Linear Programming* problem:

$$\begin{aligned} \min \quad & L\gamma \\ s.t \quad & F_1\gamma \leq F_2 + F_3 v_0 \\ & \gamma_l \leq \gamma \leq \gamma_u \end{aligned} \tag{62}$$

where L collects all the constant coefficients of cost functions, γ_l and γ_u corresponds to lower and upper limit of logical variables and controlled flows. In the form of Eq. (62), we can solve the optimization problem using OPTI Toolbox in MATLAB [14].

```
xtype = 'CCCBBBBBBBBCCCCCCCCCCCCBBBBBBBBCCCCCCCCCCCCCBBBBB
Opt = opti('f',f,'ineq',a,b,'bounds',lb,ub,'xtype',xtype);
[x,fval,exitflag,info] = solve(Opt)
```

Figure 5 OPTI Toolbox solver.

where “xtype” in the first line of above figure corresponds to the data type of the decision variables, “B” stands for binary variables and “C” stands for continues variables. In the second line, “f” corresponds to the cost function we specified, “a” corresponds to F_1 and “b” corresponds to $F_2 + F_3 v_0$.

Chapter 4

Decentralized MPC

Although the mixed integer linear programming problem formulated in Chapter 3 can be solved easily using OPTI Toolbox [14], the complexity of solving such problem grows exponentially with the number of variables. For urban drainage systems, it is typical to establish system models with tens or even hundreds of virtual tanks to achieve desirable performance. Thusly, the search space become too large to have the optimization problem be solved in an efficient manner and suits our real time control purpose. We therefore propose to partition the system into several sections according to geographical proximities and coupling relationships. Each subsystem receives local rainfall prediction and considers neighboring coupling as external disturbances. Then individual controller solves for local actions in parallel and exchange information with neighboring subsystems to update actions taken and system states. (System states will not be necessary in real application since state information can be measured directly and provided to subsystems.) Similar to the centralized case, only the first set of control actions are applied to the system and each subsystem will solve the optimization problem again with updated information and shifted horizon. Since each subsystem only receives partial information and makes local decisions, the performance of decentralized MPC is usually worse than centralized one. However, the computation time can be significantly reduced by breaking the large system into several small ones as shown in latter sections. There are naturally different separation schemes depending on priorities on geographical consideration, coupling emphasis and information transmission. Different separation schemes have different performance under various rain scenarios as shown in 4.2.

4.1 Decentralized construction

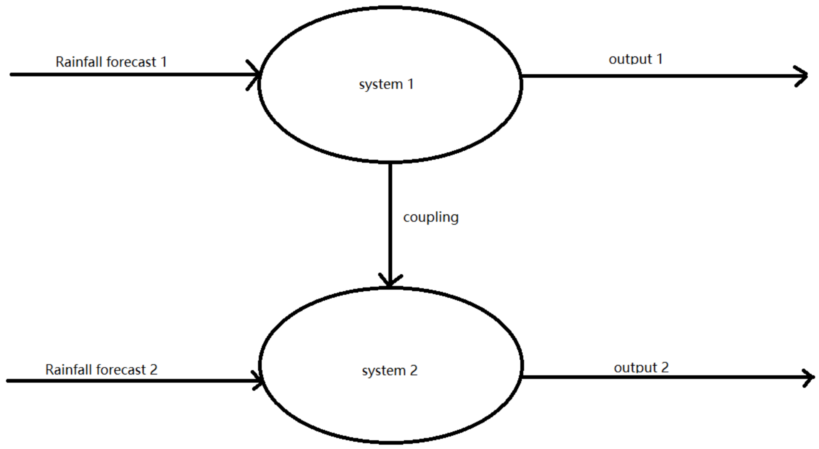

Figure 6 Subsystem example.

t =1

<u>**System 1**</u>

Known: initial condition of system 1 states and local rainfall prediction

Solve for control actions over the horizon and compute coupling information. Apply control actions corresponding to first time step.

<u>**System 2**</u>

Known: initial condition of system 2 states and local rainfall prediction

Assumption: coupling from system 1 based on historical data

Solve for control actions over the horizon and apply control actions corresponding to first time step.

t =1+

System 1

Communicate with system 2 about actual coupling information at t =1

System 2

Correct system states information with actual coupling information at t=1.

t =2

System 1

Known: Final states of previous time step and updated local rainfall prediction

Solve for control actions with shifted horizon and update coupling information. Only the first set of control actions are applied.

System 2

Known: Final states of previous time step and updated local rainfall prediction

Assumption: coupling from system 1 based on historical data

Solve for control actions with shifted horizon and apply control actions corresponding to first time step.

t =2+

System 1

Communicate with system 2 about actual coupling information at t =2.

System 2

Correct system states information with actual coupling information at t=2.

……

4.2 Partitioning methods comparison

We consider two different partitioning schemes, one with emphasis on geographical proximity and the other with emphasis on coupling relationships. In Figure [2], virtual tank 2 receives input from virtual tank 1 but two tanks could be geographically separated afar. Therefore, we propose the following partitioning schemes:

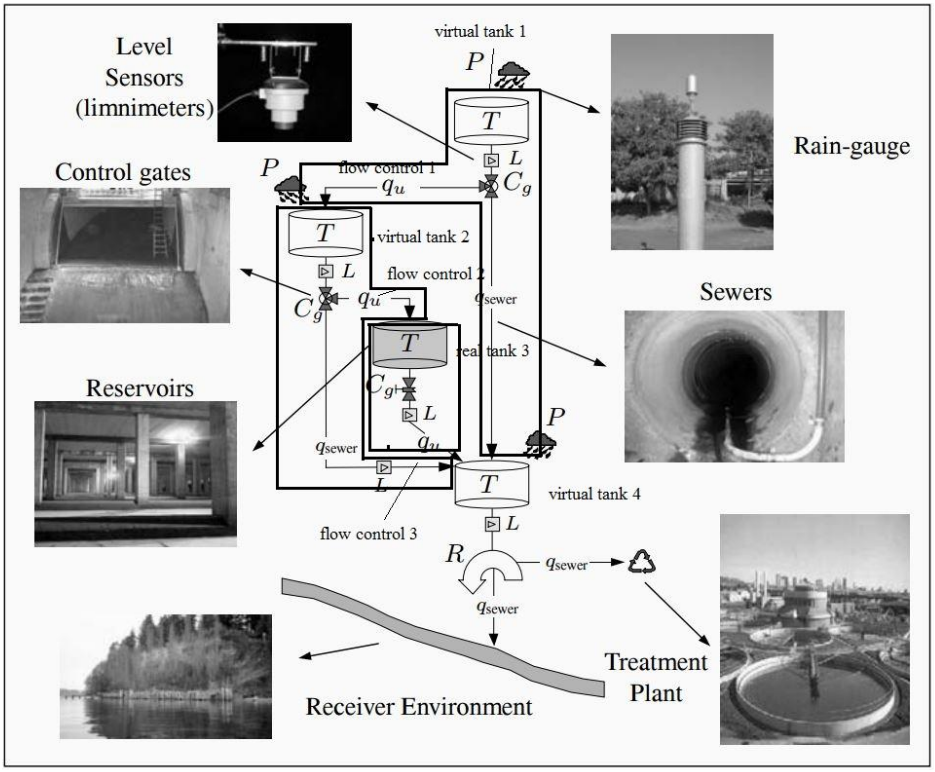

Figure 7 Geographical partition.

In this scheme, we partition the whole system into four subsystems as shown below:

Geographical Partitioning	
Subsystem 1	Virtual tank 1. Manipulated flow 1 Default sewer path 1
Subsystem 2	Virtual tank 2. Manipulated flow 2 Default sewer path 2
Subsystem 3	Real tank 3. Manipulated flow 3
Subsystem 4	Virtual tank 4

Table 1 Geographical Partition.

However, the only neighboring coupling of virtual tank 2 originates from virtual tank 1 and thusly it is logical to group them into one subsystem. In this scenario, subsystem 2 has no storage elements and previous decisions will not have any impact on current decisions.

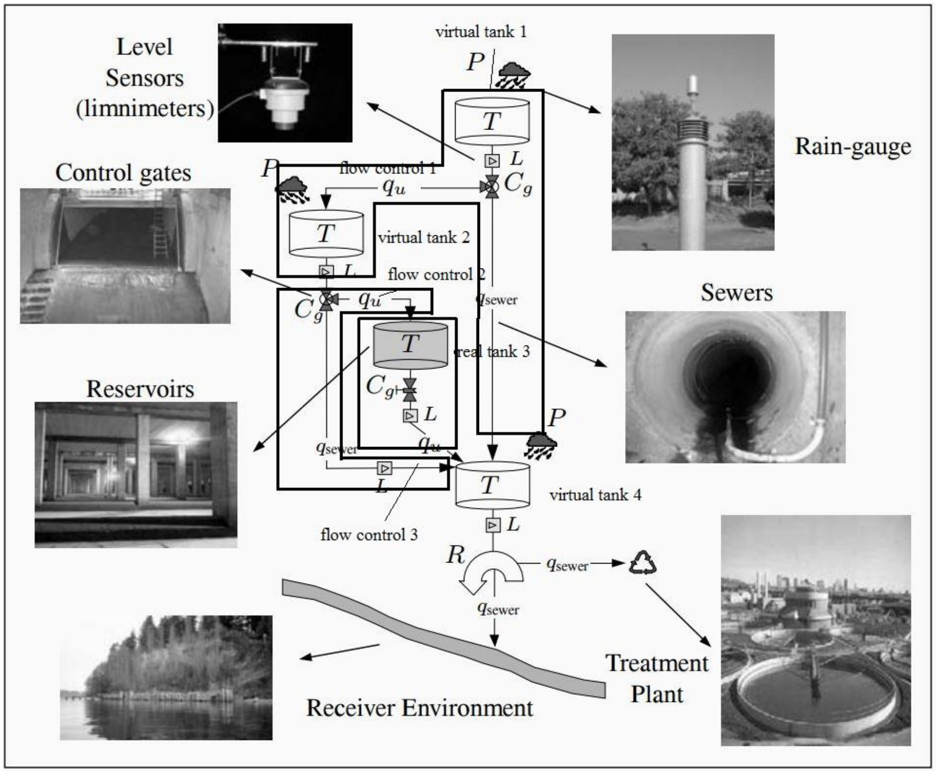

Figure 8 Coupling Partition.

Coupling Partitioning	
Subsystem 1	Virtual tank 1 and 2. Manipulated flow 1 Default sewer path 1.
Subsystem 2	Manipulated flow 2 Default sewer path 2
Subsystem 3	Real tank 3. Manipulated flow 3
Subsystem 4	Virtual tank 4

Table 2 Coupling Partition.

We briefly compare the performances of these two partitioning and the centralized optimal open loop controller. Each horizon is constituted by 5 time steps, each of length 300 seconds. The unit of the rainfall refers to the number of tipping of the bucket gauge with details explained in Chapter 5.

For constant rainfall of 50 units over the horizon

tle =

Flow1	Flow2	Flow3
0	0	0
0	1.5724	0
0	5.6797	0
0	9.0723	0
0	11.875	0

totalcost =

4.2298e+03

tle =

Flow1	Flow2	Flow3
0	0	0
0	1.5724	0
0	5.6797	0
0	9.0723	0
0	11.875	0

totalcost =

4.2298e+03

tle =

Flow1	Flow2	Flow3
0	0	0
0	1.5724	0
0	5.6797	0
0	9.0723	0
0	11.875	0

fval =

4.2298e+03

Figure 9 Performance comparison between Geographical partition, Coupling partition and Centralized open loop controller with 50 units rainfall.

For constant rainfall of 200 units over the horizon

tle =

Flow1	Flow2	Flow3
0	0	0
0	16.491	0
0	21.54	0
2.2484	21.54	0
2.8597	21.54	0

totalcost =

8.9236e+04

tle =

Flow1	Flow2	Flow3
0	0	0
0	16.49	0
0	21.54	0
0	21.54	0
0	21.54	0

totalcost =

8.9327e+04

tle =

Flow1	Flow2	Flow3
0	0	0
0	16.49	0
0	21.54	0
0	21.54	0
2.8597	21.54	0

fval =

8.8898e+04

Figure 10 Performance comparison between Geographical partition, Coupling partition and Centralized open loop controller with 200 units rainfall.

For constant rainfall of 500 units over the horizon

tle =

Flow1	Flow2	Flow3
0	0	0
2.6916	21.54	0
2.8597	21.54	0
2.8597	21.54	0
2.8597	21.54	0

totalcost =

3.8830e+05

tle =

Flow1	Flow2	Flow3
0	0	0
0	21.54	0
0	21.54	0
0	21.54	0
0	21.54	0

totalcost =

3.8746e+05

tle =

Flow1	Flow2	Flow3
0	0	0
0	24.94	0
0	24.94	0
0	24.94	0
2.8597	21.54	0

fval =

3.6643e+05

Figure 11 Performance comparison between Geographical partition, Coupling partition and Centralized open loop controller with 500 units rainfall.

As we can see from the results shown in Figure 9, when the rainfall is not large and therefore not much actions are required by systems, three controllers share exact same control strategies and final costs. The centralized open loop controller has the best performance as it takes into account the whole system and makes decisions based on all rainfall predictions. Geographical and Coupling partitioning has different performances under different scenarios and as we can see from Figure 10 and 11, Geographical partition generates better solution compared with Coupling partition under huge rainfall while coupling partition perform better under medium rainfall condition.

Chapter 5

Simulation Results

5.1 Small system

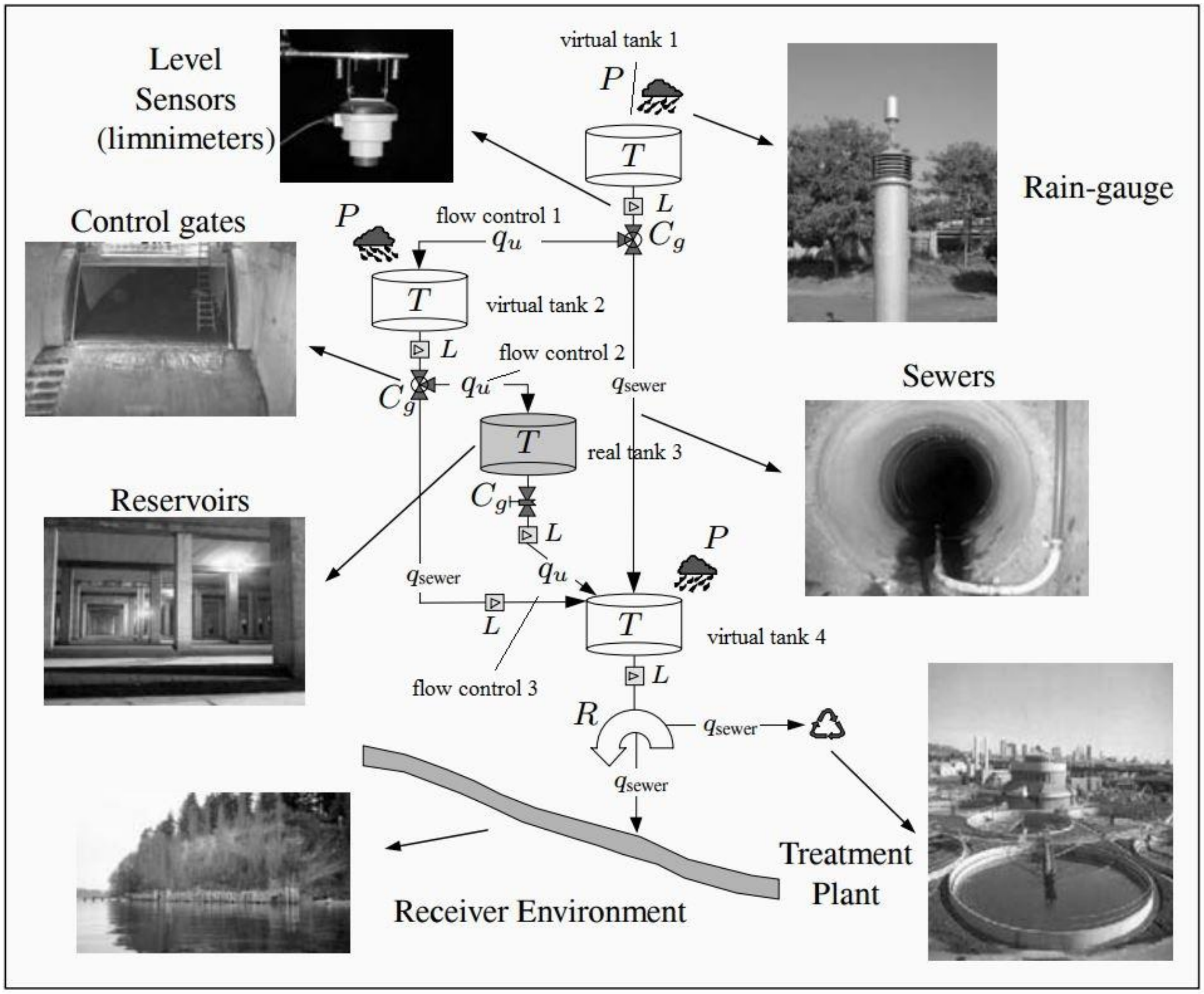

Figure 12 Sewer network.

To demonstrate the use of developed model, we present a simulation assuming perfect knowledge of rain prediction and solve the optimization problem in one open loop iteration. Parameters are based on a physical system introduced in [12] and the sewage system used is shown in Figure 2. System dynamics is introduced in Eq. (41) – Eq. (49). There are four tanks in the system and three of them are virtual tanks. We use three manipulated flow to manage the system with operation bounds shown in Table **4** and we consider a sampling time equal to 300 seconds. Using Eq. (62), we arrange the decision variables as follow:

$$\begin{aligned} \min \quad & L\gamma \\ s.t \quad & F_1\gamma \leq F_2 + F_3 v_0 \\ & \gamma_l \leq \gamma \leq \gamma_u \end{aligned} \tag{62}$$

$$\gamma = [q_1^u, q_2^u, q_3^u, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_{15}, \delta_{26}, z_1, z_2, z_3, z_4, z_{15}, z_{26}, z_{115}, z_{226}, z_{15}^u, z_{26}^u]^T \tag{63}$$

Tank parameters are presented in the following table :

Tank	S(m^2)	φ_i	$\beta_i(s^{-1})$	$\bar{v}_i(m^3)$
Tank number	Surface area	Absorption coefficient	Volumetric flow coefficient	Tank capacity
T1	323,576	1.03	7.1×10^{-4}	16,901
T2	164,869	10.4	5.8×10^{-4}	43,000
T3	5,076	-	2.0×10^{-4}	35,000
T4	754,131	0.48	1.0×10^{-4}	26,659

Table 3 Tank parameters.

Manipulated flow/ Sewer path	Capacity ($m^3\ s^{-1}$)
q_1^u	9.1
q_2^u	25
q_3^u	7
q_1^s	9.14
q_2^s	3.4

Table 4 Control variables and sewer path.

As introduced in Chapter 2, we collect all the physical constraints, such as mass balance flow and real tank physical capacities, and logical constraints associated with δ_i and z_i into the following inequalities. We demonstrate it for this system in the following section.

$$E_2\delta(t) + E_3 z(t) \le E_1 u(t) + E_4 x(t) + E_5 \quad (11)$$

5.1.1 Logical variables and constraints

There are 18 logical variables in this simulation and we elaborate their definitions as follow:

$$\delta_1 = 1 \Leftrightarrow v_1 - \overline{v}_1 \ge 0 \quad (64)$$

$$\delta_2 = 1 \Leftrightarrow v_2 - \overline{v}_2 \ge 0 \quad (65)$$

$$\delta_3 = 1 \Leftrightarrow v_3 - \overline{v}_3 \ge 0 \quad (66)$$

$$\delta_4 = 1 \Leftrightarrow v_4 - \overline{v}_4 \ge 0 \quad (67)$$

$$\delta_4 = 1 \Leftrightarrow q_1^{out} - q_1^u - \overline{q}_1^s \ge 0 \quad (68)$$

$$\delta_6 = 1 \Leftrightarrow q_2^{out} - q_2^u - \overline{q}_2^s \ge 0 \quad (69)$$

$$\delta_{15} = \delta_1 \delta_5 \quad (70)$$

$$\delta_{26} = \delta_2 \delta_6 \quad (71)$$

$$z_1 = v_1 \delta_1 \quad (72)$$

$$z_2 = v_2 \delta_2 \quad (73)$$

$$z_3 = v_3 \delta_3 \quad (74)$$

$$z_4 = v_4 \delta_4 \quad (75)$$

$$z_{15} = v_1 \delta_5 \quad (76)$$

$$z_{26} = v_2 \delta_6 \quad (77)$$

$$z_{115} = v_1 \delta_{15} \quad (78)$$

$$z_{226} = v_2 \delta_{26} \quad (79)$$

$$z_{15}^{u} = q_1^{u}\delta_5 \tag{80}$$

$$z_{26}^{u} = q_2^{u}\delta_6 \tag{81}$$

There are 47 constraints including both physical and those introduced from logical variables. We present as follow:

$$-\beta_1\overline{v}_1\delta_1 + \beta_1 z_1 \le -q_1^{u} + \beta_1 v_1 \tag{82}$$

$$-\beta_2\overline{v}_2\delta_2 + \beta_2 z_2 \le -q_2^{u} + \beta_2 v_2 \tag{83}$$

$$0 \le -\Delta t \cdot q_2^{u} + \Delta t \cdot q_3^{u} - v_3 + \overline{v}_3 \tag{84}$$

$$0 \le -q_3^{u} + \beta_3 v_3 \tag{85}$$

$$-v_1^{upper}\delta_1 + z_1 \le 0 \tag{86}$$

$$z_1 \le v_1 \tag{87}$$

$$v_1^{upper}\delta_1 - z_1 \le -v_1 + v_1^{upper} \tag{88}$$

$$-v_2^{upper}\delta_2 + z_2 \le 0 \tag{89}$$

$$z_2 \le v_2 \tag{90}$$

$$v_2^{u}\delta_2 - z_2 \le -v_2 + v_2^{u} \tag{91}$$

$$-v_4^{u}\delta_4 + z_4 \le 0 \tag{92}$$

$$z_4 \le v_4 \tag{93}$$

$$v_4^{u}\delta_4 - z_4 \le -v_4 + v_4^{u} \tag{94}$$

$$-v_1^{u}\delta_5 + z_{15} \le 0 \tag{95}$$

$$z_{15} \le v_1 \tag{96}$$

$$v_1^{u}\delta_5 - z_5 \le -v_1 + v_1^{u} \tag{97}$$

$$-v_2^u\delta_6 + z_{26} \le 0 \quad (98)$$

$$z_{26} \le v_2 \quad (99)$$

$$v_2^u\delta_6 - z_{26} \le -v_2 + v_2^u \quad (100)$$

$$-v_1^u\delta_{15} + z_{115} \le 0 \quad (101)$$

$$z_{115} \le v_1 \quad (102)$$

$$v_1^u\delta_{15} - z_{115} \le -v_1 + v_1^u \quad (103)$$

$$-v_2^u\delta_{26} + z_{226} \le 0 \quad (104)$$

$$z_{226} \le v_2 \quad (105)$$

$$v_2^u\delta_{26} - z_{226} \le v_2 + v_2^u \quad (106)$$

$$-\overline{q}_1^u\delta_5 + z_{15}^u \le 0 \quad (107)$$

$$z_{15}^u \le q_1^u \quad (108)$$

$$\overline{q}_1^u\delta_5 - z_{15}^u \le -q_1^u + \overline{q}_1^u \quad (109)$$

$$-\overline{q}_2^u\delta_6 + z_{26}^u \le 0 \quad (110)$$

$$z_{26}^u \le q_2^u \quad (111)$$

$$\overline{q}_2^u\delta_6 - z_{26}^u \le -q_2^u + \overline{q}_2^u \quad (112)$$

$$-m_1\delta_1 \le v_1 - \overline{v}_1 - m_1 \quad (113)$$

$$-(M_1 + \varepsilon)\delta_1 \le -v_1 + \overline{v}_1 - \varepsilon \quad (114)$$

$$-m_2\delta_2 \le v_2 - \overline{v}_2 - m_2 \quad (115)$$

$$-(M_2 + \varepsilon)\delta_2 \le -v_2 + \overline{v}_2 - \varepsilon \quad (116)$$

$$-m_4\delta_4 \le v_4 - \overline{v}_4 - m_4 \quad (117)$$

$$-(M_4+\varepsilon)\delta_4 \le -v_4+\overline{v}_4-\varepsilon \tag{118}$$

$$-\beta_1\overline{v}_1\delta_1+\overline{q}_1^s\delta_5+\beta_1 z_1 \le -q_1^u+\beta_1 v_1 \tag{119}$$

$$\beta_1\overline{v}_1\delta_1+(\overline{q}_1^s-\varepsilon)\delta_5-\beta_1\overline{v}_1\delta_{15}-\beta_1 z_1-\beta_1 z_{15}+\beta_1 z_{115} \le q_1^u-\beta_1 v_1+(\overline{q}_1^s-\varepsilon) \tag{120}$$

$$-\beta_2\overline{v}_2\delta_2+\overline{q}_2^s\delta_6+\beta_2 z_2 \le -q_2^u+\beta_2 v_2 \tag{121}$$

$$\beta_2\overline{v}_2\delta_2+(\overline{q}_2^s-\varepsilon)\delta_6-\beta_2\overline{v}_2\delta_{26}-\beta_2 z_2-\beta_2 z_{26}+\beta_2 z_{226} \le q_2^u-\beta_2 v_2+(\overline{q}_2^s-\varepsilon) \tag{122}$$

$$-\delta_1+\delta_{15} \le 0 \tag{123}$$

$$-\delta_5+\delta_{15} \le 0 \tag{124}$$

$$\delta_1+\delta_5-\delta_{15} \le 1 \tag{125}$$

$$-\delta_2+\delta_{26} \le 0 \tag{126}$$

$$-\delta_6+\delta_{26} \le 0 \tag{127}$$

$$\delta_2+\delta_6-\delta_{26} \le 1 \tag{128}$$

where v_i^u refers to the possible maximum value the corresponding tank can take and it is defined as follow:

$$v_i^{upper}=\overline{v}_i+S_i\cdot\Delta t\cdot\max\{d_i\}\cdot\varphi_i \tag{129}$$

and $M_i\,/\,m_i$ refers to the possible maximum/minimum of the difference between tank volumes and tank capacity:

$$M_i=v_i^{upper}-\overline{v}_i \tag{130}$$

$$m_i=-\overline{v}_i \tag{131}$$

With 3 control variables and 18 logical variables for each time step, we collect them into a column vector with a length of 21.

5.1.2 Cost functions

In this simulation, we consider following three objectives and associate them with weight coefficients as follows:

1. Minimize the overflow in virtual tanks with $c_1 = 1$.

2. Minimize the overflow in sewer paths with $c_2 = 1$.

3. Minimize the operation costs of manipulated flow $c_3 = 0.5$.

Using Eq. (63), we characterize the cost function as follows:

$$
\begin{aligned}
L\gamma &= 0.5\Delta t \cdot (q_1^u + q_2^u + q_3^u) + \beta_1 \overline{v}_1 \Delta t \cdot \delta_{15} + \beta_2 \overline{v}_2 \Delta t \cdot \delta_{26} + \beta_1 \Delta t \cdot z_{15} + \beta_2 \Delta t \cdot z_{26} \\
&- \beta_1 \Delta t \cdot z_{115} - \beta_2 \Delta t \cdot z_{226} - \Delta t \cdot z_{15}^u - \Delta t \cdot z_{26}^u - \overline{q}_1^s \Delta t \cdot \delta_5 - \overline{q}_2^s \Delta t \cdot \delta_6 \\
&+ z_1 + z_2 + z_4 - \overline{v}_1 \delta_1 - \overline{v}_2 \delta_2 - \overline{v}_4 \delta_4
\end{aligned} \tag{132}
$$

5.1.3 Results

We now present the results using the above formulation to demonstrate the performance of the controller under various initial conditions and rain scenarios. We denote manipulated flows as Flow 1, Flow 2 and Flow 3 respectively with units of $m^3 \cdot s^{-1}$ and use 1 to denote the logical events of overflowing at corresponding location. Each row corresponds to the situation at one time step and we observe that if we start with zero initial conditions of four tanks, the first row will always be zero since there is no water available to control with. Note that the status of tank 3 is not included since it is a real tank and it is considered without overflowing capabilities. The unit of rain profile corresponds to the number of tipping of bucket rain gauge [12] in one sampling time (300 s) where each tipping corresponds to an event of 1.2 mm h^{-1} of rainfall. The rain intensity is then calculated using appropriate units conversion and is assumed to be the same over all tanks at each time step.

Profile	Rain	Initial conditions	Total cost
Profile 1	[0,20,50,80,100]	[0,0,0,0]	1.9853e+03
Profile 2	[100,120,130,80,20]	[10000,20000,10000,5000]	4.1062e+04
Profile 3	[200,200,200,200,200]	[0,0,0,0]	8.8898e+04

Table 5 Rain profile and initial conditions.

Flow1	Flow2	Flow3	Tank1CSO	Tank2CSO	Tank4CSO	Sewage1CSO	Sewage2CSO
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	3.2153	0	0	0	0	0	1
0	10.02	0	0	0	0	0	1

totalcost =

1.9853e+03

Figure 13 Profile 1 actions.

Flow1	Flow2	Flow3	Tank1CSO	Tank2CSO	Tank4CSO	Sewage1CSO	Sewage2CSO
0	8.2	0	0	0	0	0	1
0	16.126	0	0	0	0	0	1
0	21.54	0	0	1	0	0	1
0	21.54	0	0	1	0	1	1
0.8099	21.54	5.6132	0	1	0	1	1

totalcost =

4.1062e+04

Figure 14 Profile 2 actions.

Flow1	Flow2	Flow3	Tank1CSO	Tank2CSO	Tank4CSO	Sewage1CSO	Sewage2CSO
0	0	0	0	0	0	0	0
0	16.49	0	0	0	0	0	1
0	21.54	0	0	1	0	0	1
0	21.54	0	0	1	0	1	1
2.8597	21.54	0	1	1	0	1	1

:otalcost =

8.8898e+04

Figure 15 Profile 3 actions.

5.1.4 Centralized MPC solutions

Now we consider the case when the rain prediction is not accurate and we use *Model Predictive Control* to compensate for modeling errors and unpredicted external disturbances. We consider a horizon of 5 time steps, each of length 300 seconds. Rainfall prediction over the whole horizon is provided to the controller at the beginning of each time step. The controller will apply the first set of control actions at the end of the iteration and new rainfall prediction is updated to the controller afterwards. The optimization problem will then be solved again with updated system information and shifted horizon. We compare the performance of MPC controller under different scenarios where in one case the rainfall prediction is almost same as the initial prediction while the other differs drastically from the initial one. The rainfall profile uses same units as in previous section and each row corresponds to the prediction provided to the controller at each time step. Therefore, the accurate rainfall is the first element of each row and is marked with red color. The initial conditions in all cases are zero.

Initial prediction	130, 100, 140, 90, 60	Initial prediction	130, 100, 140, 90, 60
1st update	90, 130, 80, 60, 50	1st update	0, 130, 80, 60, 50
2nd update	120, 80, 62, 53, 82	2nd update	20, 80, 62, 53, 82
3rd update	85, 60, 45, 70, 50	3rd update	180, 60, 45, 70, 50
4th update	45, 40, 62, 48, 65	4th update	150, 40, 62, 48, 65
Accurate rain prediction	130, 90, 120, 85, 45	Accurate rain prediction	130, 0, 20, 180, 150

Table 6 Centralized MPC rain profile.

When the discrepancy between the initial prediction and the following updates is small, as in the rain profile on left of Table **6**, the MPC controller generates same strategies as the open loop optimal controller with perfect rainfall knowledge.

Flow1	Flow2	Flow3	Tank1CSO	Tank2CSO	Tank4CSO	Sewage1CSO	Sewage2CSO
0	0	0	0	0	0	0	0
0	9.5284	0	0	0	0	0	1
0	16.229	0	0	0	0	0	1
0	21.54	0	0	1	0	0	1
0	21.54	0	0	1	0	0	1

```
totalcost =

   2.2948e+04
```

Figure 16 Same performance between open loop controller and MPC.

However, when the differences between the initial rainfall prediction and following updates become significant, as in the rain profile in the right of Table **6**, we observe that the MPC method generate different control strategies than the open loop one.

Flow1	Flow2	Flow3	Tank1CSO	Tank2CSO	Tank4CSO	Sewage1CSO	Sewage2CSO
0	0	0	0	0	0	0	0
0	9.5284	0	0	0	0	0	1
0	7.2788	0	0	0	0	0	1
0	7.4097	0	0	0	0	0	1
0	21.54	0	0	1	0	0	1

```
:otalcost =

   1.0122e+04
```

Figure 17 Open loop controller performance.

Flow1	Flow2	Flow3	Tank1CSO	Tank2CSO	Tank4CSO	Sewage1CSO	Sewage2CSO
0	0	0	0	0	0	0	0
0	9.5284	0	0	0	0	0	1
0	7.2788	0	0	0	0	0	1
0	7.4097	0	0	0	0	0	1
0	6.6514	0	0	1	0	0	1

```
:otalcost =

   1.2355e+04
```

Figure 18 MPC performance.

5.1.5 Decentralized MPC solution

Next, we compare the performances between the centralized controller and two partition schemes introduced in Chapter 4 under various rain scenarios. The rainfall prediction is fed to the controller at the beginning of each time step and the actual rainfall is marked with red color as before. The cost function is composed of virtual tank overflowing, sewer path overflowing and operation cost. The controller with highest costs is marked with red color in Table 7. Note that when the total rainfall that enters the system becomes large and unpredictable, decentralized controllers can perform better than the centralized one. This is possibly due to the fact that the rainfall predictions between time steps are largely different therefore the decisions made by centralized controller taking into account the whole system suffers larger costs than decentralized controllers and therefore the centralized controller induce higher costs.

```
rain=[500,200,300,400,200; %profile 3
          300, 20, 80,500,600;
              200,200,200,200,200;
                  200,200,200,200,200;
                 200,200,200,200,200];
```

```
rain=[130,100,140,90,60;  %profile 2
                  0,130,80,60,50;
                     20,80,62,53,82;
                       180,60,45,70,50;
                          150,40,62,48,65];
```

```
rain=[160,250,250,130,230; %profile 3
          250,250,130,230,0;
              250,130,230,0,0;
                  130,230,0,0,0;
                      230,0,0,0,0];
```

```
rain=[200,200,200,200,200; %profile 4
          200,200,200,200,200;
              200,200,200,200,200;
                  200,200,200,200,200;
                 200,200,200,200,200];
```

```
rain=[50,80,90,20,10; %profile 6
          60,70,50,40,30;
              53,42,26,23,32;
                  20,30,62,43,36;
                      43,39,53,15,20];
```

```
rain=[120,120,110,170,150; %profile 9
          150,100,150,140,130;
              130,160,126,123,132;
                  140,140,162,143,136;
                 110,139,153,115,120];
```

Figure 19 Rain profiles.

	Accurate prediction centralized solution	Inaccurate prediction centralized MPC	Coupling decentralized MPC	Geographical decentralized MPC
Profile 1	1.7376e+05	1.7657e+05	1.7845e+05	1.7929e+05
Profile 2	1.0122e+04	1.0122e+04	1.0122e+04	1.0122e+04
Profile 3	8.7599e+04	8.8029e+04	8.8029e+04	8.8029e+04
Profile 4	8.8898e+04	9.0456e+04	8.9327e+04	8.9236e+04
Profile 6	4.2382e+03	4.2382e+03	4.2382e+03	4.2388e+03
Profile 9	4.2071e+04	4.2097e+04	4.2097e+04	4.2071e+04

Table 7 Performance Comparison.

5.2 Large System

In the previous section, we demonstrated the use of incorporating logical variables into our modeling and investigated the validity of the model by performing numerical simulations. We compared the performance of the decentralized MPC and centralized MPC in terms of costs function with a small system shown in Figure 2. Due to the fact that the decentralized MPC only receives local information, the performance of decentralized MPC is naturally worse than the centralized one. To demonstrate the advantage of decentralized MPC in terms of reduced computation time, we construct a larger system shown in Figure 20.

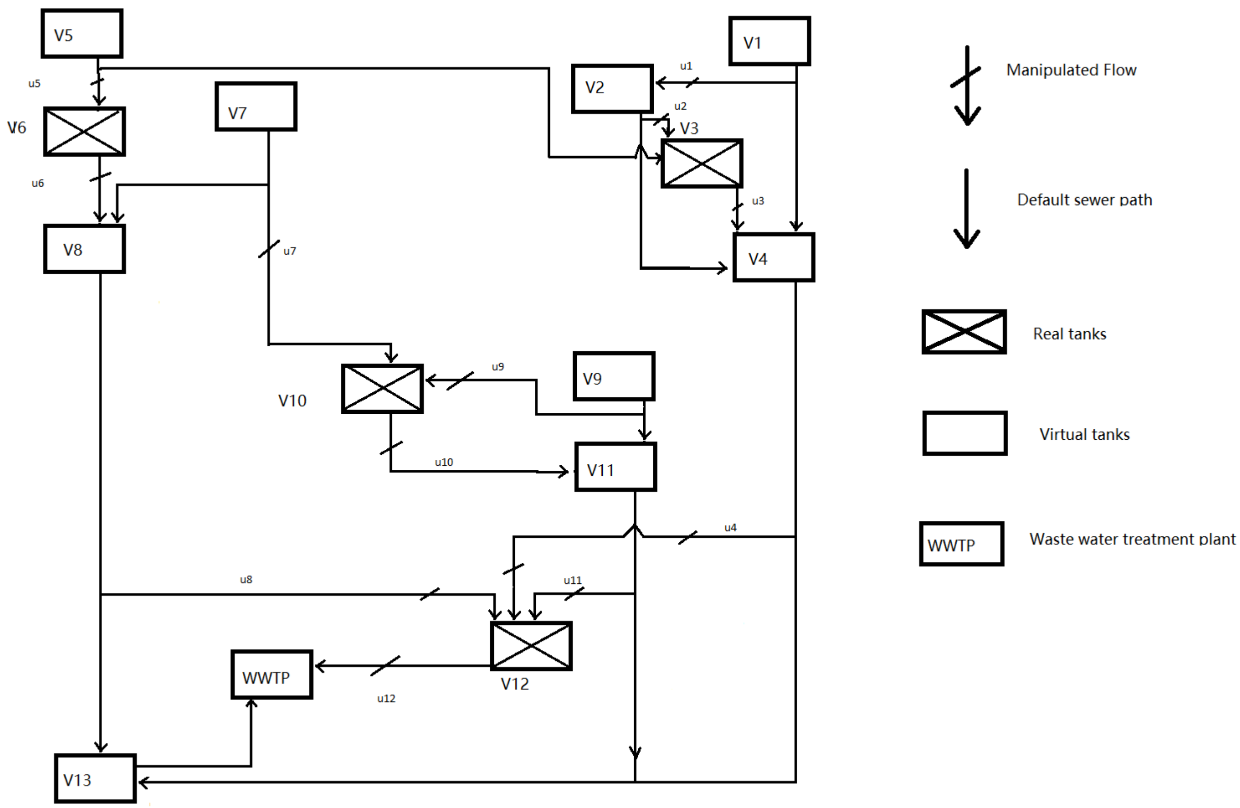

Figure 20 Large sewer network.

There are 13 thanks in this configuration and 4 of them are real tanks marked with a cross. All the tanks are connected either through controlled flows or default sewer path. We use 12 manipulated flows to manage the system with a slash marked on the connection. These virtual tanks can be considered to be an urban sewage system covering a neighborhood of a city with each virtual tank geographically separated and at the final level, all the sewage is processed through waste water treatment plant before it is released to the environment. Real tank 12 refers to the large reservoir of the urban sewage system that is usually used to storage all the untreated sewage that exceeds the capacity of the treatment plant. For example, the large underground tunnel of Chicago and Escola Industrial reservoir of Barcelona.

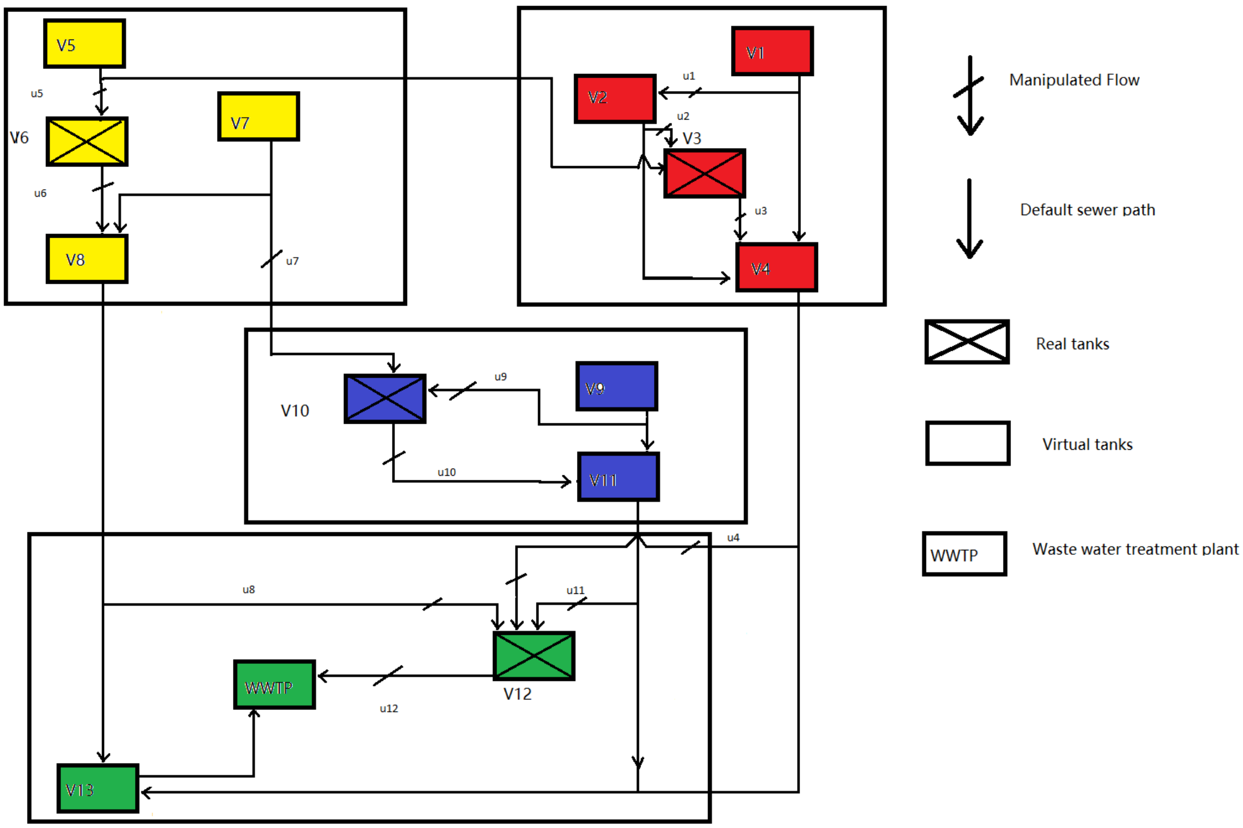

Figure 21 Decentralized large system.

We partition the system into 4 subsystems shown in different colors based mainly on geographical considerations. Since each subsystem covers different area, the local rainfall at each subsystem may differ. The structure of decentralized MPC is similar to those discussed in Chapter 4 but now each subsystem is of comparable size of a small system shown in previous section and therefore the advantage of decentralized MPC in terms of reducing computation time becomes obvious. The rainfall profile is again fed to each subsystem at the beginning of the each time step and updated afterwards when one iteration is completed. We omit the details of the modeling formulation, constraints inequalities and rainfall profile due to the size of the system. The details of the simulation can be found in the Appendix. We show the strategies generated by each controller where 5 columns correspond to 5 time step and each column contains 12 actions of manipulated flows with unit cubic meters per second.

5.2.1 Results

Rain profile 1 results:

action2 =

0	0	0	0	0
0	6.5449	4.8145	3.3852	2.2045
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0.2877
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Figure 22 Profile 1 open loop actions.

0	0	0	0	0
0	6.5449	4.8145	3.3852	2.2045
0	0	0	0	0
0	0	0	0	0
0	0	0	-0.0000	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0.2877
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Figure 23 Profile 1 centralized MPC actions.

0	0	0	0	0
0	6.5449	4.8145	3.3852	2.2045
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0.2877
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Figure 24 Profile 1 decentralized MPC actions.

Rain profile 2 results:

```
action2 =

         0         0         0         0         0
         0         0    5.5285    9.4447   11.9832
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0    0.3832
         0         0         0    0.3535    3.0850
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
```

Figure 25 Profile 2 open loop actions.

```
action =

         0         0         0         0         0
         0         0    5.5285    9.4447   11.9832
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0   12.8447    0.3832
         0         0         0    0.3535    0.3491
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
```

Figure 26 Profile 2 centralized MPC actions.

```
control =

         0         0         0         0         0
         0         0    5.5285    9.4447   11.9832
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0    1.6598   12.8447   15.3832
         0         0         0         0    0.0708
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
```

Figure 27 Profile 2 decentralized MPC actions.

Rain profile 3 results:

```
action2 =

         0         0         0         0         0
         0    5.5504   13.4407   21.2509   21.5400
         0         0         0         0         0
         0         0         0    1.2714    4.8745
         0         0         0         0    0.4340
         0         0         0         0         0
         0    3.8990    1.8407   17.2101    9.9400
         0         0         0    4.6856    6.9997
         0         0    1.2407    9.0509    9.3400
         0         0         0         0         0
         0         0         0         0    1.0822
         0         0         0         0         0
```

Figure 28 Profile 3 open loop actions.

```
action =

         0         0         0         0         0
         0    5.5504   13.4407   10.5259   21.5400
         0         0         0         0         0
         0         0         0    1.2714    4.8745
         0         0    3.9241    5.6438    7.2340
         0         0         0         0         0
         0    3.8990    5.3686   14.4337    9.9400
         0         0         0    3.9342    6.9997
         0         0    1.2407         0    9.3400
         0         0         0         0         0
         0         0         0         0   10.0945
         0         0         0         0         0
```

Figure 29 Profile 3 centralized MPC actions.

```
control =

         0         0         0         0         0
         0    5.5504   13.4407   21.2509    4.7162
         0         0    4.6081         0         0
         0         0         0    2.6538   14.8546
         0         0         0         0    0.4340
         0         0         0         0         0
         0    3.8990   16.8407    9.6509   24.9400
         0         0         0    1.4906    6.0954
         0         0    1.2407    9.0509    0.0000
         0         0         0         0         0
         0         0         0         0    1.0822
         0         0         0         0    8.9932
```

Figure 30 Profile 3 decentralized MPC actions.

Rain profile 4 results:

action2 =

0	0	0	0	2.8597
0	9.5284	21.5400	21.5400	21.5400
0	0	0	0	0
0	0	1.3906	17.8042	14.5506
0	0	0	2.7672	5.1997
0	0	0	0	0
0	12.9284	24.9400	24.9400	9.9400
0	0	1.4437	5.8825	6.9997
0	0	9.3400	12.5628	9.3400
0	0	0	0	0
0	0	0	20.8861	8.6590
0	0	0	0	0

Figure 31 Profile 4 open loop actions.

action =

0	0	0	0	0
0	9.5284	21.5400	0	21.5400
0	0	0	0	0
0	0	1.3906	17.8042	15.9375
0	3.0762	5.6230	2.7672	5.1997
0	0	0	0	0
0	12.9284	9.9400	9.9395	24.9400
0	0	1.4437	6.9997	6.9997
0	0	0	0	18.7934
0	0	0	0	0
0	0	0	20.8861	25.0000
0	0	0	1.7006	28.0942

Figure 32 Profile 4 centralized MPC actions.

control =

0	0	0	0	0
0	9.5284	21.5400	20.1128	23.0137
0	0	0	0	0
0	0	1.3906	7.8042	14.5506
0	0	0	2.7672	5.1997
0	0	0	0	0
0	12.9284	24.9400	24.9400	24.9400
0	0	1.4437	4.5672	6.9997
0	0	9.3400	1.0376	0
0	0	0	0	0
0	0	0	2.8861	8.6590
0	0	0	0	0

Figure 33 Profile 4 decentralized MPC actions.

	Open loop	Centralized MPC	Decentralized MPC
Profile 1	2.5855e+03	2.5855e+03	2.5855e+03
Profile 2	4.6167e+03	6.1330e+03	8.5327e+03
Profile 3	5.7065e+04	6.3838e+04	6.7485e+04
Profile 4	2.2906e+05	2.5244e+05	2.3970e+05

Table 8 Large system costs comparison.

	Open loop	Centralized MPC	Decentralized MPC
Profile 1	0.1331	0.4394	0.2834
Profile 2	0.0573	5.5762	1.2529
Profile 3	0.7662	77.7161	4.0638
Profile 4	2.8142	64.6046	3.7019

Table 9 Large system computation time comparison.

From Table 8 we observe similar trend for performance differences between the controllers. Open loop controller generates best control strategies in all scenarios and therefore can serve as an optimal solution standard for other controllers to compare against for sanity check. The decentralized controller usually perform worse than the centralized controller but when the rain fall becomes huge and the system reaches capacity, the performance difference between decentralized MPC and centralize MPC is insignificant as shown in profile 3 and profile 4. However, the computation time of decentralized MPC is comparatively much smaller than the centralized MPC, especially in huge rain scenarios. We also observe similar phenomena where centralized MPC induce larger costs than decentralized MPC in some cases similar to those for small system.

Chapter 6

Conclusion

6.1 Summary

The need for real time control techniques are getting increased attention around the globe due to increased extreme weather conditions and immediate needs for fully taking advantage of drainage infrastructures. Appropriate modeling of drainage systems is one of first challenge met by engineers and scientists. The physical feature of sewage flows are described by Saint-Venant equations but this level of detail is not necessary for our purpose of global management and real time control. We desire a linear model that captures all the characteristics of drainage systems and preserve the convexity of the problem in consideration of the computation. Mixed Logical Dynamical Systems [8], which associates binary variables to logical events by introducing linear inequality constraints involving binary variables, helps us to develop such linear models using virtual tank [12] concepts. Manipulated flows controlled by hydraulic pumps are used to divert flows from tank to tank to ensure appropriate utilization of the storage capacity. By considering rainfall as external disturbances, we express the tank volume as discrete dynamics equations. We then define appropriate cost functions, such as minimizing Combined Sewer Overflow in urban areas and minimizing operation costs of manipulated flows, to pose the drainage system management problem as an optimization problem respecting sewage discrete dynamics, and constraints introduced by logical variables. We also presented two engineering devices introduced in [13] to reduce operation costs by harvesting rain power and rain energy at various locations to generate electricity. Based on our priorities on minimizing overflows at different areas and minimizing operation costs, we can tailor the controllers to our needs by associating different weight coefficients with different cost functions.

Model Predictive Control is usually used in process control such as chemical plants and oil refineries. By only applying the first set of control actions and constantly re-solving the optimization problem with shifted horizon and updated information, Model Predictive Control is able to compensate for modeling errors and rainfall prediction inaccuracies and readily suits our purpose. With the cost functions formulated earlier for the purpose of minimizing combined sewer overflows, we can easily formulate our problem in the framework of Model Predictive Control. However, the presence of logical variables introduced earlier makes the problem to be a Mixed Integer Linear Programming problem and the complexity of solving such problems grows exponentially with the number of the variables. The size of urban drainage systems naturally induces models with tens or even hundreds of variables. Thusly, directly applying Model Predictive Control would take too much time and defeat our real time control purpose. We propose to partition the system into several subsections based on geographical proximity and coupling relationships. Each subsystem computes local control actions based on local rainfall prediction and neighboring couplings are considered as external disturbances. Neighboring controllers will also exchange information to increase performance. We demonstrate the advantage of decentralized Model Predictive Control with several numerical simulations under different rain scenarios and initial conditions. Several partition schemes were also compared to explore the best structure under various situations. Decentralized Model Predictive Control induces larger costs in most of scenarios but significantly reduced computation time compared with centralized Model Predictive Control justifies its application.

The general applicability of the presented modeling methods and decentralizing schemes make this framework can applied to any modern urban sewage systems provided that enough system information can be obtained. The built in robustness provided by Model Predictive Control techniques proved to be an effective method for manage drainage system to accommodate for modeling errors and rainfall prediction inaccuracies.

6.2 Future work

The partition scheme based on coupling relations in this thesis was completed by human observation logical deduction. For systems with larger sizes and finer details, this approach is not suitable and susceptible to errors. We can automates this process by inspecting the coupling coefficients between the tanks and separate them according to prescribed threshold. This method is more tractable and less prone to human errors especially in the case of large systems.

As observed in previous simulations, decentralized MPC controllers of different partition schemes have different performances in various scenarios. This feature can be explored further by performing more numerical simulations and generalize the pattern for the optimal one. The drainage system can therefore be adapted to different decentralized schemes according to incoming rain scenarios to ensure optimal performances.

Rain profiles used in the simulation are provided to controllers as a vector containing the predicted rain data over the horizon. Statistical rainfall profile based on historical data can be introduced in the formulation and therefore we can develop performance guarantees for different controllers.

References

[1] S.l. Seneviratne, N. Nicholls, D. Easterling, C. M. Goodess, S. Kanae, J. Kossin, Y. Luo, J. Marengo, K. Mclnnes, M. Rahimi, M. Reirchstein, A. Sorteberg, C. Vera and X. Zhang, "Changes in climate extremes and their impacts on the natural physical environment," in Managing the Risks of Extreme Events and Disasters to Advance Climate Change Adaptation, 2012.

[2] K.E. Kunkel, D.R. Easterling, D.A.R. Kristovich, B. Gleason, L. Stoecker and R. Smith, "Meteorological causes of the secular variations in observed extreme precipitation events for the conterminous United States," J. Hydrometeror., vol. 13, pp. 1131-1141, 2012.

[3] S. Hallegatte, C. Green, R.J. Nicholls and J. Corfee-Morlot, "Future flood losses in major coastal cities," Nature Climate Change, vol. 3, no.9, pp. 802-806,2013.

[4] J. Rawlings and D. Mayne, " Model Predictive Control: Theory and Design," Madison, WI: Nob Hill, 2009.

[5] M. Gelormino and N. Ricker, " Model predictive control of a combined sewer system," International Journal of Control 59, 793-816.

[6] S. Darsono, and J. W. Labadie, " Neural-optimal control algorithm for real-time regulation of inline storage in combined sewer systems," Environ. Modell. Softw., vol. 22, no. 9, pp. 1349 – 1361, 2007.

[7] M. Marinaki and M. Papageorgiou, "Optimal Real-time Control of Sewer Networks," Secaucus, NJ: Springer-Verlag, 2005.

[8] A. Bemporad and M. Morari, " Control of systems integrating logic, dynamics, and constraints," Automatica 36, 1249-1274.

[9] M. Gelormino and N. Ricker, “Model predictive control of a combined sewer system,” Int. J. Control, vol. 59, no. 3, pp. 793–816, 1994.

[10] C. Ocampo-Martinez, V. Puig, G. Cembrano, and J. Quevedo, “Application of Predictive Control Strategies to the Management of Complex Networks in the Urban Water Cycle,” IEEE Control Systems 33(1): 15-41 (2013).

[11] V. Singh, “Hydrologic systems,” in Rainfall-Runoff Modeling, vol. 1. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[12] C. Ocampo-Martinez, A. Bemporad, A. Ingimundarson, and V. Puig, “On hybrid model predictive control of sewer networks,” in Identification and Control: The Gap Between Theory and Practice. New York: Springer-Verlag, 2007, pp. 87–114.

[13] Z. Gu, S. Zhao, Y. Jiao, H. Pan, S. Shang, and X. Xu, “A new way to construct sponge city by co-use of rain water and rain energy,” Yangtze River, vol.47, No.10, 1001-4179(2016)10-0015-05.

[14] OPTI Toolbox .Main page: https://www.inverseproblem.co.nz/OPTI/index.php

[15] M. Garcia, “ Impact of CSOs from Chicago Area Waterways on Lake Michigan.”

[16] G, Fu, D. Butler and S.T. Khu, “Multiple objective optimal control of integrated urban wastewater system,” Environ. Model. Softw., vol. 23, no. 2, p.225-234,2008.

[17] J. Barreiro-Gomez, G. Obando, G.Riano-Briceno, N. Quijano, and C. Ocampo-Martinez, “Decentralized Control for Urban Drainage Systems Via Population Dynamics: Bogota Case Study,” European Control Conference, 2015.

[18] P. V. Overloop, “Model Predictive Control on Open Water Systems,” Delft, The Netherlands: Delft Univ. Press, 2006.

[19] Y. Ermolin, “Mathematical modelling for optimized control of Moscow’s sewer network,” Appl. Math. Modell., vol. 23, no. 7, pp. 543–556, 1999.

[20] J. Abadie and J. Carpentier, “Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints,” in Optimization. New York: Academic, 1969, pp. 37–47.

[21] J. Maciejowski, “ Predictive Control with Constraints,” Upper Saddle River, NJ: Prentice Hall, 2002.

[22] A. Chen and X. Xu, “Goal programming approach to solving network design problem with multiple objectives and demand uncertainty,” Expert Syst. Appl., vol. 39, no. 4, pp. 4160-4170, 2012.

[23] M. Schütze, A. Campisanob, H. Colas, W. Schillingd, and P. Vanrolleghem, “Real time control of urban wastewater systems: Where do we stand today?” J. Hydrol., vol. 299, nos. 3–4, pp. 335–348, 2004.

[24] Abbott, M. B., J. C. Bathurst, J. A. Cunge, and P. E. O’Connell (1986a), “ An introduction to the European Hydrological System – Systeme Hydrologique Europeen, “SHE”, 1: History and philosophy of a physically based distributed modeling system,” J. Hydrol., 87, 45–59, doi:10.1016/0022-1694(86)90114-9.

[25] Cantone, J. P. (2010), “ Illinois Urban Hydrologic Model (IUHM): A probabilistic approach for predicting hydrologic response in highly urbanized catchments,” Ph.D. dissertation, Dep. of Civil and Environ. Eng., Univ. of Ill. at Urbana - Champaign, Urbana.

[26] J. Baillieul and P. J. Antsaklis, “Control and communication challenges in networked real-time systems,” Proceedings of the IEEE, vol. 95, no. 1, pp. 9–28, 2007.

[27] M. K. Stinson, “Benefits of sewerage system real-time control,” in Impacts of Global Climate Change. ASCE, 2005, pp. 1–11.

[28] S. Amin, X. Litrico, S. S. Sastry, and A. M. Bayen, “Cyber security of water SCADA systems - Part II: Attack detection using enhanced hydrodynamic models,” IEEE Transactions on Control Systems Technology, vol. 21, no. 5, pp. 1679–1693, 2013.

[29] J. Alex, M. Schutze, M. Ogurek, and U. Jumar, “Systematic design of distributed controllers for sewer networks,” in Proc. IFAC World Congress, Seoul, 2008, pp. 557–551.

[30] G. Cembrano, G. Wells, J. Quevedo, R. Perez, R. Argelaguet, “Optimal control of a water distribution network in a supervisory control system,” Control Eng. Pract. 8 (10) (2000) 1177–1188.

[31] G. Cembrano, J. Quevedo, M. Salamero, V. Puig, J. Figueras, J. Marti, “Optimal control of urban drainage systems. A case study,” Control Engineering Practice, 12 (1) (2004), pp. 1-9.

[32] M. S. Branicky, V. S. Borkar and S. K. Mitter, "A unified framework for hybrid control: model and optimal control theory," in IEEE Transactions on Automatic Control, vol. 43, no. 1, pp. 31-45, Jan 1998.

[33] Bemporad A., Borrelli F., Morari M. “ On the Optimal Control Law for Linear Discrete Time Hybrid Systems.” In: Tomlin C.J., Greenstreet M.R. (eds) Hybrid Systems: Computation and Control. HSCC 2002. Lecture Notes in Computer Science, vol 2289. Springer, Berlin, Heidelberg.

[34] Farahani, S.S., Raman, V., & Murray, R.M. (2015). “ Robust Model Predictive Control for Signal Temporal Logic Synthesis.”

Appendix A

Simulation Code for the large system

Centralized MPC

```
%% system specification

syms dt eps v1 v1u v1l v1m beta1 S1 phi1 delta1 deltaS1 deltaS1a1 z1 zS1a1 zS1a1a1 zuS1 qu1
qs1m...
        v2 v2u v2l v2m beta2 S2 phi2 delta2 deltaS2 deltaS2a2 z2 zS2a2 zS2a2a2 zuS2 qu2 qs2m...
        v3 v3m qu3 beta3...
        v4 v4u v4l v4m beta4 S4 phi4 delta4 deltaS4 deltaS4a4 z4 zS4a4 zS4a4a4 zuS4 qu4 qs4m...
        v5 v5u v5l v5m beta5 S5 phi5 delta5 deltaS5 deltaS5a5 z5 zS5a5 zS5a5a5 zuS5 qu5 qs5m...
        v6 v6m beta6 qu6 ...
        v7 v7u v7l v7m beta7 S7 phi7 delta7 deltaS7 deltaS7a7 z7 zS7a7 zS7a7a7 zuS7 qu7 qs7m...
        v8 v8u v8l v8m beta8 S8 phi8 delta8 deltaS8 deltaS8a8 z8 zS8a8 zS8a8a8 zuS8 qu8 qs8m...
        v9 v9u v9l v9m beta9 S9 phi9 delta9 deltaS9 deltaS9a9 z9 zS9a9 zS9a9a9 zuS9 qu9 qs9m...
        v10 v10m beta10 qu10 ...
        v11 v11u v11l v11m beta11 S11 phi11 delta11 deltaS11 deltaS11a11 z11 zS11a11 zS11a11a11
zuS11 qu11 qs11m...
        v12 v12u  v12m beta12 qu12 ...
        v13 v13u v13l v13m beta13 S13 phi13 delta13 deltaS13 deltaS13a13 z13 zS13a13 zS13a13a13
zuS13 qu13 ...
        qu1m qu2m qu3m qu4m qu5m qu8m qu7m qu8m qu9m qu10m qu11m qu12m ...
        qs1m qs2m qs4m qs5m qs7m qs8m qs9m qs11m  ...
        m1 m2  m4 m5 m6 m7 m8 m9 m10 m11  m13 ...
        M1 M2  M4 M5 M6 M7 M8 M9 M10 M11  M13 ...
        v01 v02 v03 v04 v05 v06 v07 v08 v09 v010 v011 v012 v013 ...
        p01 p02 p03 p04 p05 p06 p07 p08 p09 p010 p011 p012 p013 ...
        p11 p12 p13 p14 p15 p16 p17 p18 p19 p110 p111 p112 p113 ...
        p21 p22 p23 p24 p25 p26 p27 p28 p29 p210 p211 p212 p213 ...
        p31 p32 p33 p34 p35 p36 p37 p38 p39 p310 p311 p312 p313 ...
        p41 p42 p43 p44 p45 p46 p47 p48 p49 p410 p411 p412 p413 ...
        z01 z02 z03 z04 z05 z06 z07 z08 z09 z010 z011 z012 z013 ...
        z11 z12 z13 z14 z15 z16 z17 z18 z19 z110 z111 z112 z113 ...
        z21 z22 z23 z24 z25 z26 z27 z28 z29 z210 z211 z212 z213 ...
        z31 z32 z33 z34 z35 z36 z37 z38 z39 z310 z311 z312 z313 ...
        z0S1a1 z0S1a1a1 z0S2a2 z0S2a2a2 z0S4a4 z0S4a4a4 z0S5a5 z0S5a5a5 z0S7a7 z0S7a7a7 z0S8a8
z0S8a8a8 z0S9a9 z0S9a9a9 z0S11a11 z0S11a11a11 z0uS1 z0uS2 z0uS4 z0uS5 z0uS7 z0uS8 z0uS9 z0uS11 ...
        z1S1a1 z1S1a1a1 z1S2a2 z1S2a2a2 z1S4a4 z1S4a4a4 z1S5a5 z1S5a5a5 z1S7a7 z1S7a7a7 z1S8a8
z1S8a8a8 z1S9a9 z1S9a9a9 z1S11a11 z1S11a11a11 z1uS1 z1uS2 z1uS4 z1uS5 z1uS7 z1uS8 z1uS9 z1uS11 ...
        z2S1a1 z2S1a1a1 z2S2a2 z2S2a2a2 z2S4a4 z2S4a4a4 z2S5a5 z2S5a5a5 z2S7a7 z2S7a7a7 z2S8a8
z2S8a8a8 z2S9a9 z2S9a9a9 z2S11a11 z2S11a11a11 z2uS1 z2uS2 z2uS4 z2uS5 z2uS7 z2uS8 z2uS9 z2uS11 ...
        z3S1a1 z3S1a1a1 z3S2a2 z3S2a2a2 z3S4a4 z3S4a4a4 z3S5a5 z3S5a5a5 z3S7a7 z3S7a7a7 z3S8a8
z3S8a8a8 z3S9a9 z3S9a9a9 z3S11a11 z3S11a11a11 z3uS1 z3uS2 z3uS4 z3uS5 z3uS7 z3uS8 z3uS9 z3uS11 ...
        delta01 delta02 delta03 delta04 delta05 delta06 delta07 delta08 delta09 delta010 delta011
delta012 delta013 delta0S1 delta0S1a1 delta0S2 delta0S2a2 delta0S4 delta0S4a4 delta0S5 delta0S5a5
delta0S7 delta0S7a7 delta0S8 delta0S8a8 delta0S9 delta0S9a9 delta0S11 delta0Sa11 ...
        delta11 delta12 delta13 delta14 delta15 delta16 delta17 delta18 delta19 delta110 delta111
delta112 delta113 delta1S1 delta1S1a1 delta1S2 delta1S2a2 delta1S4 delta1S4a4 delta1S5 delta1S5a5
delta1S7 delta1S7a7 delta1S8 delta1S8a8 delta1S9 delta1S9a9 delta1S11 delta1Sa11 ...
        delta21 delta22 delta23 delta24 delta25 delta26 delta27 delta28 delta29 delta210 delta211
delta212 delta213 delta2S1 delta2S1a1 delta2S2 delta2S2a2 delta2S4 delta2S4a4 delta2S5 delta2S5a5
delta2S7 delta2S7a7 delta2S8 delta2S8a8 delta2S9 delta2S9a9 delta2S11 delta2Sa11 ...
        delta31 delta32 delta33 delta34 delta35 delta36 delta37 delta38 delta39 delta310 delta311
delta312 delta313 delta3S1 delta3S1a1 delta3S2 delta3S2a2 delta3S4 delta3S4a4 delta3S5 delta3S5a5
delta3S7 delta3S7a7 delta3S8 delta3S8a8 delta3S9 delta3S9a9 delta3S11 delta3Sa11 ...
        u01 u02 u03 u04 u05 u06 u07 u08 u09 u010 u011 u012 ...
        u11 u12 u13 u14 u15 u16 u17 u18 u19 u110 u111 u112 ...
        u21 u22 u23 u24 u25 u26 u27 u28 u29 u210 u211 u212 ...
        u31 u32 u33 u34 u35 u36 u37 u38 u39 u310 u311 u312 ...
```

```
A = sym(zeros(13,13));   %tank
B1 = sym(zeros(13,12));  %control
B2 = sym(zeros(13,29));   %delta
B3 = sym(zeros(13,37));   %z
B4= sym(zeros(13,13));    % disturbance

d0=[p01;p02;p03;p04;p05;p06;p07;p08;p09;p010;p011;p012;p013];
d1=[p11;p12;p13;p14;p15;p16;p17;p18;p19;p110;p111;p112;p113];
d2=[p21;p22;p23;p24;p25;p26;p27;p28;p29;p210;p211;p212;p213];
d3=[p31;p32;p33;p34;p35;p36;p37;p38;p39;p310;p311;p312;p313];
d4=[p41;p42;p43;p44;p45;p46;p47;p48;p49;p410;p411;p412;p413];

z0=[z01;z02;z03;z04;z05;z06;z07;z08;z09;z010;z011;z012;z013;

z0S1a1;z0S1a1a1;z0S2a2;z0S2a2a2;z0S4a4;z0S4a4a4;z0S5a5;z0S5a5a5;z0S7a7;z0S7a7a7;z0S8a8;z0S8a8a8;z
0S9a9;z0S9a9a9;z0S11a11;z0S11a11a11;
    z0uS1;z0uS2;z0uS4;z0uS5;z0uS7;z0uS8;z0uS9;z0uS11];
z1=[z11;z12;z13;z14;z15;z16;z17;z18;z19;z110;z111;z112;z113;

z1S1a1;z1S1a1a1;z1S2a2;z1S2a2a2;z1S4a4;z1S4a4a4;z1S5a5;z1S5a5a5;z1S7a7;z1S7a7a7;z1S8a8;z1S8a8a8;z
1S9a9;z1S9a9a9;z1S11a11;z1S11a11a11;
    z1uS1;z1uS2;z1uS4;z1uS5;z1uS7;z1uS8;z1uS9;z1uS11];
z2=[z21;z22;z23;z24;z25;z26;z27;z28;z29;z210;z211;z212;z213;

z2S1a1;z2S1a1a1;z2S2a2;z2S2a2a2;z2S4a4;z2S4a4a4;z2S5a5;z2S5a5a5;z2S7a7;z2S7a7a7;z2S8a8;z2S8a8a8;z
2S9a9;z2S9a9a9;z2S11a11;z2S11a11a11;
    z2uS1;z2uS2;z2uS4;z2uS5;z2uS7;z2uS8;z2uS9;z2uS11];
z3=[z31;z32;z33;z34;z35;z36;z37;z38;z39;z310;z311;z312;z313;

z3S1a1;z3S1a1a1;z3S2a2;z3S2a2a2;z3S4a4;z3S4a4a4;z3S5a5;z3S5a5a5;z3S7a7;z3S7a7a7;z3S8a8;z3S8a8a8;z
3S9a9;z3S9a9a9;z3S11a11;z3S11a11a11;
    z3uS1;z3uS2;z3uS4;z3uS5;z3uS7;z3uS8;z3uS9;z3uS11];

delta0 =
[delta01;delta02;delta03;delta04;delta05;delta06;delta07;delta08;delta09;delta010;delta011;delta0
12;delta013;delta0S1;delta0S1a1;delta0S2;delta0S2a2;delta0S4;delta0S4a4;delta0S5;delta0S5a5;delta
0S7;delta0S7a7;delta0S8;delta0S8a8;delta0S9;delta0S9a9;delta0S11;delta0Sa11];
delta1 =
[delta11;delta12;delta13;delta14;delta15;delta16;delta17;delta18;delta19;delta110;delta111;delta1
12;delta113;delta1S1;delta1S1a1;delta1S2;delta1S2a2;delta1S4;delta1S4a4;delta1S5;delta1S5a5;delta
1S7;delta1S7a7;delta1S8;delta1S8a8;delta1S9;delta1S9a9;delta1S11;delta1Sa11];
delta2 =
[delta21;delta22;delta23;delta24;delta25;delta26;delta27;delta28;delta29;delta210;delta211;delta2
12;delta213;delta2S1;delta2S1a1;delta2S2;delta2S2a2;delta2S4;delta2S4a4;delta2S5;delta2S5a5;delta
2S7;delta2S7a7;delta2S8;delta2S8a8;delta2S9;delta2S9a9;delta2S11;delta2Sa11];
delta3 =
[delta31;delta32;delta33;delta34;delta35;delta36;delta37;delta38;delta39;delta310;delta311;delta3
12;delta313;delta3S1;delta3S1a1;delta3S2;delta3S2a2;delta3S4;delta3S4a4;delta3S5;delta3S5a5;delta
3S7;delta3S7a7;delta3S8;delta3S8a8;delta3S9;delta3S9a9;delta3S11;delta3Sa11];

u0 = [u01;u02;u03;u04;u05;u06;u07;u08;u09;u010;u011;u012];
u1 = [u11;u12;u13;u14;u15;u16;u17;u18;u19;u110;u111;u112];
u2 = [u21;u22;u23;u24;u25;u26;u27;u28;u29;u210;u211;u212];
u3 = [u31;u32;u33;u34;u35;u36;u37;u38;u39;u310;u311;u312];

v0=[v01;v02;v03;v04;v05;v06;v07;v08;v09;v010;v011;v012;v013];

A(1,1)=1-dt*beta1;
B2(1,1)=v1m-dt*beta1*v1m;
B3(1,1)=dt*beta1-1;
B4(1,1)=dt*phi1*S1;

A(2,2)=1-dt*beta2;
B1(2,1)=dt;
B2(2,2)=(1-beta2*dt)*v2m;
B3(2,2)=beta2*dt-1;
B4(2,2)=dt*phi2*S2;

A(3,3)=1;
A(3,5)=dt*beta5;
```

```
B1(3,2)= dt;
B1(3,3)=-dt;
B1(3,5)=-dt;
B2(3,5)=dt*beta5*v5m;
B2(3,20)=dt*qs5m;
B2(3,21)=-dt*beta5*v5m;
B3(3,5)=-dt*beta5;
B3(3,20)=-dt*beta5;
B3(3,21)=dt*beta5;
B3(3,33)=dt;

A(4,1)=dt*beta1;
A(4,2)=dt*beta2;
A(4,4)=1-beta4*dt;
B1(4,3)=dt;
B1(4,1)=-dt;
B1(4,2)=-dt;
B2(4,1)=dt*beta1*v1m;
B2(4,2)=dt*beta2*v2m;
B2(4,4)=(1-dt*beta4)*v4m;
B2(4,14)=dt*qs1m;
B2(4,16)=dt*qs2m;
B2(4,15)=-dt*beta1*v1m;
B2(4,17)=-dt*beta2*v2m;
B3(4,1)=-dt*beta1;
B3(4,2)=-dt*beta2;
B3(4,4)=dt*beta4-1;
B3(4,14)=-dt*beta1;
B3(4,16)=-dt*beta2;
B3(4,15)=dt*beta1;
B3(4,17)=dt*beta2;
B3(4,30)=dt;
B3(4,31)=dt;
B4(4,4)=dt*S4*phi4;


A(5,5)=1-beta5*dt;
B2(5,5)=1-dt*beta5;
B3(5,5)=dt*beta5-1;
B4(5,5)=dt*S5*phi5;

A(6,6)=1;
B1(6,5)=dt;
B1(6,6)=-dt;

A(7,7)=1-dt*beta7;
B2(7,7)=(1-dt*beta7)*v7m;
B3(7,7)=dt*beta7-1;
B4(7,7)=dt*S7*phi7;

A(8,8)=1-dt*beta8;
A(8,7)=dt*beta7;
B1(8,6)=dt;
B1(8,7)=-dt;
B2(8,22)=dt*qs7m;
B2(8,8)=(1-dt*beta8)*v8m;
B2(8,7)=dt*beta7*v7m;
B2(8,23)=-dt*beta7*v7m;
B3(8,7)=-dt*beta7;
B3(8,8)=dt*beta8-1;
B3(8,22)=-dt*beta7;
B3(8,23)=dt*beta7;
B3(8,34)=dt;
B4(8,8)=dt*S8*phi8;

A(9,9)=1-dt*beta9;
B2(9,9)=(1-dt*beta9)*v9m;
B3(9,9)=dt*beta9-1;
B4(9,9)=dt*phi9*S9;

A(10,10)=1;
```

```
B1(10,7)=dt;
B1(10,9)=dt;
B1(10,10)=-dt;

A(11,9)=dt*beta9;
A(11,11)=1-dt*beta11;
B1(11,9)=-dt;
B1(11,10)=dt;
B2(11,9)=dt*beta9*v9m;
B2(11,11)=(1-dt*beta11)*v11m;
B2(11,26)=dt*qs9m;
B2(11,27)=-dt*beta9*v9m;
B3(11,9)=-dt*beta9;
B3(11,11)=dt*beta11-1;
B3(11,26)=-dt*beta9;
B3(11,27)=dt*beta9;
B3(11,36)=dt;
B4(11,11)=dt*phi11*S11;

A(12,12)=1;
B1(12,4)=dt;
B1(12,8)=dt;
B1(12,11)=dt;
B1(12,12)=-dt;

A(13,13)=1-dt*beta13;
A(13,4)=dt*beta4;
A(13,8)=dt*beta8;
A(13,11)=dt*beta11;
B1(13,4)=-dt;
B1(13,8)=-dt;
B1(13,11)=-dt;
B2(13,4)=dt*beta4*v4m;
B2(13,8)=dt*beta8*v8m;
B2(13,11)=dt*beta11*v11m;
B2(13,13)=(1-dt*beta13)*v13m;
B2(13,18)=dt*qs4m;
B2(13,24)=dt*qs8m;
B2(13,28)=dt*qs11m;
B2(13,19)=-dt*beta4*v4m;
B2(13,29)=-dt*beta11*v11m;
B2(13,25)=-dt*beta8*v8m;
B3(13,4)=-dt*beta4;
B3(13,8)=-dt*beta8;
B3(13,11)=-dt*beta11;
B3(13,13)=dt*beta13-1;
B3(13,18)=-dt*beta4;
B3(13,24)=-dt*beta8;
B3(13,28)=-dt*beta11;
B3(13,19)=dt*beta4;
B3(13,25)=dt*beta8;
B3(13,29)=dt*beta11;
B3(13,32)=dt;
B3(13,35)=dt;
B3(13,37)=dt;
B4(13,13)=dt*phi4*S13;

v1=A*v0+B1*u0+B2*delta0+B3*z0+B4*d0;
v2=A*v1+B1*u1+B2*delta1+B3*z1+B4*d1;
v3=A*v2+B1*u2+B2*delta2+B3*z2+B4*d2;
v4=A*v3+B1*u3+B2*delta3+B3*z3+B4*d3;
%% constraints
E1=sym(zeros(206,12)); %qu
E2=sym(zeros(206,29)); %delta
E3=sym(zeros(206,37)); %z
E4=sym(zeros(206,13)); %v
E5=sym(zeros(206,1));  %constants

E2(1,1)=-beta1*v1m;
E3(1,1)=beta1;
E1(1,1)=-1;
```

```
E4(1,1)=beta1;      %qu1

E2(2,2)=-beta2*v2m;
E3(2,2)=beta2;
E1(2,2)=-1;
E4(2,2)=beta2;   %qu2

E2(3,5)=dt*beta5*v5m;
E2(3,20)=dt*qs5m;
E2(3,21)=-dt*beta5*v5m;
E3(3,5)=-dt*beta5;
E3(3,20)=-dt*beta5;
E3(3,21)=dt*beta5;
E3(3,33)=dt;
E1(3,2)=-dt;
E1(3,3)=dt;
E1(3,5)=dt;
E4(3,5)=-dt*beta5;
E4(3,3)=-1;
E5(3,1)=v3m;    %qu2

E1(4,3)=-1;
E4(4,3)=beta3;   %qu3

E2(5,4)=-beta4*v4m;
E3(5,4)=beta4;
E1(5,4)=-1;
E4(5,4)=beta4; %qu4

E1(6,12)=dt;
E1(6,4)=-dt;
E1(6,11)=-dt;
E1(6,8)=-dt;
E4(6,12)=-1;
E5(6,1)=v12m;   %qu4

E2(7,5)=-beta5*v5m;
E3(7,5)=beta5;
E1(7,5)=-1;
E4(7,5)=beta5;  %qu5

E1(8,6)=dt;
E1(8,5)=-dt;
E4(8,6)=-1;
E5(8,1)=v6m;  %qu5

E1(9,6)=-1;
E4(9,6)=beta6; %qu6

E2(10,7)=-beta7*v7m;
E3(10,7)=beta7;
E1(10,7)=-1;
E4(10,7)=beta7; %qu7

E1(11,10)=dt;
E1(11,7)=-dt;
E1(11,9)=-dt;
E4(11,10)=-1;
E5(11,1)=v10m; %qu7

E2(12,8)=-beta8*v8m;
E3(12,8)=beta8;
E1(12,8)=-1;
E4(12,8)=beta8;   %qu8

E2(13,9)=-beta9*v9m;
E3(13,9)=beta9;
E1(13,9)=-1;
E4(13,9)=beta9; %qu9

% E1(14,10)=-1;
```

```
% E4(14,10)=beta10; %qu10

E1(14,9)=dt;
E1(14,7)=dt;
E1(14,10)=-dt;
E4(14,10)=1;

E2(15,11)=-beta11*v11m;
E3(15,11)=beta11;
E1(15,11)=-1;
E4(15,11)=beta11; %qu11

E1(16,12)=-1;
E4(16,12)=beta12; %qu12

% E1(16,4)=dt;
% E1(16,11)=dt;
% E1(16,8)=dt;
% E1(16,12)=-dt;
% E4(16,1)=1;         %qu12

E2(17,1)=-v1u;
E3(17,1)=1;

E2(18,1)=v1l;
E3(18,1)=-1;

E2(19,1)=-v1l;
E3(19,1)=1;
E4(19,1)=1;
E5(19,1)=-v1l;

E2(20,1)=v1u;
E3(20,1)=-1;
E4(20,1)=-1;
E5(20,1)=v1u;   %z1

E2(21,2)=-v2u;
E3(21,2)=1;

E2(22,2)=v2l;
E3(22,2)=-1;

E2(23,2)=-v2l;
E3(23,2)=1;
E4(23,2)=1;
E5(23,1)=-v2l;

E2(24,2)=v2u;
E3(24,2)=-1;
E4(24,2)=-1;
E5(24,1)=v2u;   %z2

E2(25,4)=-v4u;
E3(25,4)=1;

E2(26,4)=v4l;
E3(26,4)=-1;

E2(27,4)=-v4l;
E3(27,4)=1;
E4(27,4)=1;
E5(27,1)=-v4l;

E2(28,4)=v4u;
E3(28,4)=-1;
E4(28,4)=-1;
E5(28,1)=v4u;  %z4

E2(29,5)=-v5u;
E3(29,5)=1;
```

```
E2(30,5)=v5l;
E3(30,5)=-1;

E2(31,5)=-v5l;
E3(31,5)=1;
E4(31,5)=1;
E5(31,1)=-v5l;

E2(32,5)=v5u;
E3(32,5)=-1;
E4(32,5)=-1;
E5(32,1)=v5u; %z5

E2(33,7)=-v7u;
E3(33,7)=1;

E2(34,7)=v7l;
E3(34,7)=-1;

E2(35,7)=-v7l;
E3(35,7)=1;
E4(35,7)=1;
E5(35,1)=-v7l;

E2(36,7)=v7u;
E3(36,7)=-1;
E4(36,7)=-1;
E5(36,1)=v7u; %z7

E2(37,8)=-v8u;
E3(37,8)=1;

E2(38,8)=v8l;
E3(38,8)=-1;

E2(39,8)=-v8l;
E3(39,8)=1;
E4(39,8)=1;
E5(39,1)=-v8l;

E2(40,8)=v8u;
E3(40,8)=-1;
E4(40,8)=-1;
E5(40,1)=v8u; %z8

E2(41,9)=-v9u;
E3(41,9)=1;

E2(42,9)=v9l;
E3(42,9)=-1;

E2(43,9)=-v9l;
E3(43,9)=1;
E4(43,9)=1;
E5(43,1)=-v9l;

E2(44,9)=v9u;
E3(44,9)=-1;
E4(44,9)=-1;
E5(44,1)=v9u; %z9

E2(45,11)=-v11u;
E3(45,11)=1;

E3(46,11)=-1;

E3(47,11)=1;
E4(47,11)=1;

E2(48,11)=v11u;
```

```
E3(48,11)=-1;
E4(48,11)=-1;
E5(48,1)=v11u; %z11

E2(49,13)=-v13u;
E3(49,13)=1;

E2(50,13)=v13l;
E3(50,13)=-1;

E2(51,13)=-v13l;
E3(51,13)=1;
E4(51,13)=1;
E5(51,1)=-v13l;

E2(52,13)=v13u;
E3(52,13)=-1;
E4(52,13)=-1;
E5(52,1)=v13u;  %z13

E2(53,14)=-v1u;
E3(53,14)=1;

E2(54,14)=v1l;
E3(54,14)=-1;

E2(55,14)=-v1l;
E3(55,14)=1;
E4(55,1)=1;
E5(55,1)=-v1l;

E2(56,14)=v1u;
E3(56,14)=-1;
E4(56,1)=-1;
E5(56,1)=v1u;  %zs11

E2(57,15)=-v1u;
E3(57,15)=1;

E2(58,15)=v1l;
E3(58,15)=-1;

E2(59,15)=-v1l;
E3(59,15)=1;
E4(59,1)=1;
E5(59,1)=-v1l;

E2(60,15)=v1u;
E3(60,15)=-1;
E4(60,1)=-1;
E5(60,1)=v1u; %zs111

E2(61,16)=-v2u;
E3(61,16)=1;

E2(62,16)=v2l;
E3(62,16)=-1;

E2(63,16)=-v2l;
E3(63,16)=1;
E4(63,2)=1;
E5(63,1)=-v2l;

E2(64,16)=v2u;
E3(64,16)=-1;
E4(64,2)=-1;
E5(64,1)=v2u; %zs22

E2(65,17)=-v2u;
E3(65,17)=1;
```

```
E2(66,17)=v2l;
E3(66,17)=-1;

E2(67,17)=-v2l;
E3(67,17)=1;
E4(67,2)=1;
E5(67,1)=-v2l;

E2(68,17)=v2u;
E3(68,17)=-1;
E4(68,2)=-1;
E5(68,1)=v2u;  %zs222

E2(69,18)=-v4u;
E3(69,18)=1;

E2(70,18)=v4l;
E3(70,18)=-1;

E2(71,18)=-v4l;
E3(71,18)=1;
E4(71,4)=1;
E5(71,1)=-v4l;

E2(72,18)=v4u;
E3(72,18)=-1;
E4(72,4)=-1;
E5(72,1)=v4u;  %zs44

E2(73,19)=-v4u;
E3(73,19)=1;

E2(74,19)=v4l;
E3(74,19)=-1;

E2(75,19)=-v4l;
E3(75,19)=1;
E4(75,4)=1;
E5(75,1)=-v4l;

E2(76,19)=v4u;
E3(76,19)=-1;
E4(76,4)=-1;
E5(76,1)=v4u;  %zs444

E2(77,20)=-v5u;
E3(77,20)=1;

E2(78,20)=v5l;
E3(78,20)=-1;

E2(79,20)=-v5l;
E3(79,20)=1;
E4(79,5)=1;
E5(79,1)=-v5l;

E2(80,20)=v5u;
E3(80,20)=-1;
E4(80,5)=-1;
E5(80,1)=v5u;  %zs55

E2(81,21)=-v5u;
E3(81,21)=1;

E2(82,21)=v5l;
E3(82,21)=-1;

E2(83,21)=-v5l;
E3(83,21)=1;
E4(83,5)=1;
E5(83,1)=-v5l;
```

```
E2(84,21)=v5u;
E3(84,21)=-1;
E4(84,5)=-1;
E5(84,1)=v5u;   %zs555

E2(85,22)=-v7u;
E3(85,22)=1;

E2(86,22)=v7l;
E3(86,22)=-1;

E2(87,22)=-v7l;
E3(87,22)=1;
E4(87,7)=1;
E5(87,1)=-v7l;

E2(88,22)=v7u;
E3(88,22)=-1;
E4(88,7)=-1;
E5(88,1)=v7u;  %zs77

E2(89,23)=-v7u;
E3(89,23)=1;

E2(90,23)=v7l;
E3(90,23)=-1;

E2(91,23)=-v7l;
E3(91,23)=1;
E4(91,7)=1;
E5(91,1)=-v7l;

E2(92,23)=v7u;
E3(92,23)=-1;
E4(92,7)=-1;
E5(92,1)=v7u;  %zs777

E2(93,24)=-v8u;
E3(93,24)=1;

E2(94,24)=v8l;
E3(94,24)=-1;

E2(95,24)=-v8l;
E3(95,24)=1;
E4(95,8)=1;
E5(95,1)=-v8l;

E2(96,24)=v8u;
E3(96,24)=-1;
E4(96,8)=-1;
E5(96,1)=v8u;  %zs88

E2(97,25)=-v8u;
E3(97,25)=1;

E2(98,25)=v8l;
E3(98,25)=-1;

E2(99,25)=-v8l;
E3(99,25)=1;
E4(99,8)=1;
E5(99,1)=-v8l;

E2(100,25)=v8u;
E3(100,25)=-1;
E4(100,8)=-1;
E5(100,1)=v8u;  %zs888

E2(101,26)=-v9u;
```

```
E3(101,26)=1;

E2(102,26)=v9l;
E3(102,26)=-1;

E2(103,26)=-v9l;
E3(103,26)=1;
E4(103,9)=1;
E5(103,1)=-v9l;

E2(104,26)=v9u;
E3(104,26)=-1;
E4(104,9)=-1;
E5(104,1)=v9u;  %zs99

E2(105,27)=-v9u;
E3(105,27)=1;

E2(106,27)=v9l;
E3(106,27)=-1;

E2(107,27)=-v9l;
E3(107,27)=1;
E4(107,9)=1;
E5(107,1)=-v9l;

E2(108,27)=v9u;
E3(108,27)=-1;
E4(108,9)=-1;
E5(108,1)=v9u;  %zs999

E2(109,28)=-v11u;
E3(109,28)=1;

E2(110,28)=v11l;
E3(110,28)=-1;

E2(111,28)=-v11l;
E3(111,28)=1;
E4(111,11)=1;
E5(111,1)=-v11l;

E2(112,28)=v11u;
E3(112,28)=-1;
E4(112,11)=-1;
E5(112,1)=v11u;  %zs11a11

E2(113,29)=-v11u;
E3(113,29)=1;

E2(114,29)=v11l;
E3(114,29)=-1;

E2(115,29)=-v11l;
E3(115,29)=1;
E4(115,11)=1;
E5(115,1)=-v11l;

E2(116,29)=v11u;
E3(116,29)=-1;
E4(116,11)=-1;
E5(116,1)=v11u;  %zs11a11a11

E2(117,14)=-qu1m;
E3(117,30)=1;

E3(118,30)=-1;

E3(119,30)=1;
E1(119,1)=1;
```

```
E2(120,14)=qu1m;
E3(120,30)=-1;
E1(120,1)=-1;
E5(120,1)=qu1m;  %zuS1

E2(121,16)=-qu2m;
E3(121,31)=1;

E3(122,31)=-1;

E3(123,31)=1;
E1(123,2)=1;

E2(124,16)=qu2m;
E3(124,31)=-1;
E1(124,2)=-1;
E5(124,1)=qu2m;   %zuS2

E2(125,18)=-qu4m;
E3(125,32)=1;

E3(126,32)=-1;

E3(127,32)=1;
E1(127,4)=1;

E2(128,18)=qu4m;
E3(128,32)=-1;
E1(128,4)=-1;
E5(128,1)=qu4m; %zuS4

E2(129,20)=-qu5m;
E3(129,33)=1;

E3(130,33)=-1;

E3(131,33)=1;
E1(131,5)=1;

E2(132,20)=qu5m;
E3(132,33)=-1;
E1(132,5)=-1;
E5(132,1)=qu5m;  %zuS5

E2(133,22)=-qu7m;
E3(133,34)=1;

E3(134,34)=-1;

E3(135,34)=1;
E1(135,7)=1;

E2(136,22)=qu7m;
E3(136,34)=-1;
E1(136,7)=-1;
E5(136,1)=qu7m;  %zuS7

E2(137,24)=-qu8m;
E3(137,35)=1;

E3(138,35)=-1;

E3(139,35)=1;
E1(139,8)=1;

E2(140,24)=qu8m;
E3(140,35)=-1;
E1(140,8)=-1;
E5(140,1)=qu8m; %zuS8

E2(141,26)=-qu9m;
```

```
E3(141,36)=1;

E3(142,36)=-1;

E3(143,36)=1;
E1(143,9)=1;

E2(144,26)=qu9m;
E3(144,36)=-1;
E1(144,9)=-1;
E5(144,1)=qu9m;  %zuS9

E2(145,28)=-qu11m;
E3(145,37)=1;

E3(146,37)=-1;

E3(147,37)=1;
E1(147,11)=1;

E2(148,28)=qu11m;
E3(148,37)=-1;
E1(148,11)=-1;
E5(148,1)=qu11m;  %zuS11

E2(149,1)=-m1;
E4(149,1)=1;
E5(149,1)=-v1m-m1;

E2(150,1)=-(M1+eps);
E4(150,1)=-1;
E5(150,1)=v1m-eps;  %delta1

E2(151,2)=-m2;
E4(151,2)=1;
E5(151,1)=-v2m-m2;

E2(152,2)=-(M2+eps);
E4(152,2)=-1;
E5(152,1)=v2m-eps; %delta2

E2(153,4)=-m4;
E4(153,4)=1;
E5(153,1)=-v4m-m4;

E2(154,4)=-(M4+eps);
E4(154,4)=-1;
E5(154,1)=v4m-eps; %delta4

E2(155,5)=-m5;
E4(155,5)=1;
E5(155,1)=-v5m-m5;

E2(156,5)=-(M5+eps);
E4(156,5)=-1;
E5(156,1)=v5m-eps;  %delta5

E2(157,7)=-m7;
E4(157,7)=1;
E5(157,1)=-v7m-m7;

E2(158,7)=-(M7+eps);
E4(158,7)=-1;
E5(158,1)=v7m-eps;  %delta7

E2(159,8)=-m8;
E4(159,8)=1;
E5(159,1)=-v8m-m8;

E2(160,8)=-(M8+eps);
E4(160,8)=-1;
```

```
E5(160,1)=v8m-eps; %delta8

E2(161,9)=-m9;
E4(161,9)=1;
E5(161,1)=-v9m-m9;

E2(162,9)=-(M9+eps);
E4(162,9)=-1;
E5(162,1)=v9m-eps;  %delta9

E2(163,11)=-m11;
E4(163,11)=1;
E5(163,1)=-v11m-m11;

E2(164,11)=-(M11+eps);
E4(164,11)=-1;
E5(164,1)=v11m-eps;  %delta 11

E2(165,13)=-m13;
E4(165,13)=1;
E5(165,1)=-v13m-m13;

E2(166,13)=-(M13+eps);
E4(166,13)=-1;
E5(166,1)=v13m-eps;   %delta 13

% E2(167,1)=-beta1*v1m;
% E2(167,14)=qs1m+qu1m;
% E3(167,1)=beta1;
% E1(167,1)=-1;
% E4(167,1)=beta1;

E2(167,14)=qs1m+qu1m;
E2(167,15)=-beta1*v1m;
E3(167,14)=-beta1;
E3(167,15)=beta1;
E1(167,1)=-1;
E5(167,1)=qu1m;

E2(168,1)=beta1*v1m;
E2(168,14)=qs1m-eps;
E2(168,15)=-beta1*v1m;
E3(168,1)=-beta1;
E3(168,14)=-beta1;
E3(168,15)=beta1;
E1(168,1)=1;
E4(168,1)=-beta1;
E5(168,1)=qs1m-eps;    %deltaS1

E2(169,1)=-1;
E2(169,15)=1;

E2(170,14)=-1;
E2(170,15)=1;

E2(171,1)=1;
E2(171,14)=1;
E2(171,15)=-1;
E5(171,1)=1;          %deltaS1a1


E2(172,16)=qs2m+qu2m;
E2(172,17)=-beta2*v2m;
E3(172,16)=-beta2;
E3(172,17)=beta2;
E1(172,2)=-1;
E5(172,1)=qu2m;

E2(173,2)=beta2*v2m;
E2(173,16)=qs2m-eps;
E2(173,17)=-beta2*v2m;
```

```
E3(173,2)=-beta2;
E3(173,16)=-beta2;
E3(173,17)=beta2;
E1(173,2)=1;
E4(173,2)=-beta2;
E5(173,1)=qs2m-eps;     %deltaS2

E2(174,2)=-1;
E2(174,17)=1;

E2(175,16)=-1;
E2(175,17)=1;

E2(176,2)=1;
E2(176,16)=1;
E2(176,17)=-1;        %deltaS2a2
E5(176,1)=1;

% E2(177,4)=-beta4*v4m;
% E2(177,18)=qs4m+qu4m;
% E3(177,4)=beta4;
% E1(177,4)=-1;
% E4(177,4)=beta4;

E2(177,18)=qs4m+qu4m;
E2(177,19)=-beta4*v4m;
E3(177,18)=-beta4;
E3(177,19)=beta4;
E1(177,4)=-1;
E5(177,1)=qu4m;

E2(178,4)=beta4*v4m;
E2(178,18)=qs4m-eps;
E2(178,19)=-beta4*v4m;
E3(178,4)=-beta4;
E3(178,18)=-beta4;
E3(178,19)=beta4;
E1(178,4)=1;
E4(178,4)=-beta4;
E5(178,1)=qs4m-eps;  %deltaS4

E2(179,4)=-1;
E2(179,19)=1;

E2(180,18)=-1;
E2(180,19)=1;

E2(181,4)=1;
E2(181,18)=1;
E2(181,19)=-1; %deltaS4a4
E5(181,1)=1;

% E2(182,5)=-beta5*v5m;
% E2(182,20)=qs5m+qu5m;
% E3(182,5)=beta5;
% E1(182,5)=-1;
% E4(182,5)=beta5;

E2(182,20)=qs5m+qu5m;
E2(182,21)=-beta5*v5m;
E3(182,20)=-beta5;
E3(182,21)=beta5;
E1(182,5)=-1;
E5(182,1)=qu5m;

E2(183,5)=beta5*v5m;
E2(183,20)=qs5m-eps;
E2(183,21)=-beta5*v5m;
E3(183,5)=-beta5;
E3(183,20)=-beta5;
E3(183,21)=beta5;
```

```
E1(183,5)=1;
E4(183,5)=-beta5;
E5(183,1)=qs5m-eps;   %deltaS5

E2(184,5)=-1;
E2(184,21)=1;

E2(185,20)=-1;
E2(185,21)=1;

E2(186,5)=1;
E2(186,20)=1;
E2(186,21)=-1; %deltaS5a5
E5(186,1)=1;

E2(187,22)=qs7m+qu7m;
E2(187,23)=-beta7*v7m;
E3(187,22)=-beta7;
E3(187,23)=beta7;
E1(187,7)=-1;
E5(187,1)=qu7m;

E2(188,7)=beta7*v7m;
E2(188,22)=qs7m-eps;
E2(188,23)=-beta7*v7m;
E3(188,7)=-beta7;
E3(188,22)=-beta7;
E3(188,23)=beta7;
E1(188,7)=1;
E4(188,7)=-beta7;
E5(188,1)=qs7m-eps; %deltaS7

E2(189,7)=-1;
E2(189,23)=1;

E2(190,22)=-1;
E2(190,23)=1;

E2(191,7)=1;
E2(191,22)=1;
E2(191,23)=-1;   %deltaS7a7
E5(191,1)=1;

% E2(192,8)=-beta8*v8m;
% E2(192,24)=qs8m+qu8m;
% E3(192,8)=beta8;
% E1(192,8)=-1;
% E4(192,8)=beta8;

E2(192,24)=qs8m+qu8m;
E2(192,25)=-beta8*v8m;
E3(192,24)=-beta8;
E3(192,25)=beta8;
E1(192,8)=-1;
E5(192,1)=qu8m;

E2(193,8)=beta8*v8m;
E2(193,24)=qs8m-eps;
E2(193,25)=-beta8*v8m;
E3(193,8)=-beta8;
E3(193,24)=-beta8;
E3(193,25)=beta8;
E1(193,8)=1;
E4(193,8)=-beta8;
E5(193,1)=qs8m-eps; %deltaS8

E2(194,8)=-1;
E2(194,25)=1;

E2(195,24)=-1;
E2(195,25)=1;
```

```
E2(196,8)=1;
E2(196,24)=1;
E2(196,25)=-1;  %deltaS8a8
E5(196,1)=1;

% E2(197,9)=-beta9*v9m;
% E2(197,26)=qs9m+qu9m;
% E3(197,9)=beta9;
% E1(197,9)=-1;
% E4(197,9)=beta9;

E2(197,26)=qs9m+qu9m;
E2(197,27)=-beta9*v9m;
E3(197,26)=-beta9;
E3(197,27)=beta9;
E1(197,9)=-1;
E5(197,1)=qu9m;

E2(198,9)=beta9*v9m;
E2(198,26)=qs9m-eps;
E2(198,27)=-beta9*v9m;
E3(198,9)=-beta9;
E3(198,26)=-beta9;
E3(198,27)=beta9;
E1(198,9)=1;
E4(198,9)=-beta9;
E5(198,1)=qs9m-eps;  %deltaS9

E2(199,9)=-1;
E2(199,27)=1;

E2(200,26)=-1;
E2(200,27)=1;

E2(201,9)=1;
E2(201,26)=1;
E2(201,27)=-1;  %deltaS9a9
E5(201,1)=1;

%E2(202,11)=-beta11*v11m;
%E2(202,28)=qs11m+qu11m;
%E3(202,11)=beta11;
%E1(202,11)=-1;
%E4(202,11)=beta11;    there is a problem with this bounding

E2(202,28)=qs11m+qu11m;
E2(202,29)=-beta11*v11m;
E3(202,28)=-beta11;
E3(202,29)=beta11;
E1(202,11)=-1;
E5(202,1)=qu11m;

E2(203,11)=beta11*v11m;
E2(203,28)=qs11m-eps;
E2(203,29)=-beta11*v11m;
E3(203,11)=-beta11;
E3(203,28)=-beta11;
E3(203,29)=beta11;
E1(203,11)=1;
E4(203,11)=-beta11;
E5(203,1)=qs11m-eps;  %deltaS11

E2(204,11)=-1;
E2(204,29)=1;

E2(205,28)=-1;
E2(205,29)=1;

E2(206,11)=1;
E2(206,28)=1;
```

```
E2(206,29)=-1;  %deltaS11a11
E5(206,1)=1;
%% constraints integration
F11=sym(zeros(206,78*5));    %initialize
F21=sym(zeros(206,1));
F31=sym(zeros(206,13));
F11=[-E1,E2,E3,zeros(206,78*4)];
F12=[-E4*B1,-E4*B2,-E4*B3,-E1,E2,E3,zeros(206,78*3)];
F13=[-E4*A*B1,-E4*A*B2,-E4*A*B3,-E4*B1,-E4*B2,-E4*B3,-E1,E2,E3,zeros(206,78*2)];
F14=[-E4*A^2*B1,-E4*A^2*B2,-E4*A^2*B3,-E4*A*B1,-E4*A*B2,-E4*A*B3,-E4*B1,-E4*B2,-E4*B3,-
E1,E2,E3,zeros(206,78)];
F15=[-E4*A^3*B1,-E4*A^3*B2,-E4*A^3*B3,-E4*A^2*B1,-E4*A^2*B2,-E4*A^2*B3,-E4*A*B1,-E4*A*B2,-
E4*A*B3,-E4*B1,-E4*B2,-E4*B3,-E1,E2,E3];

F21=E5;
F22=E4*B4*d0+E5;
F23=E4*A*B4*d0+E4*B4*d1+E5;
F24=E4*A*A*B4*d0+E4*A*B4*d1+E4*B4*d2+E5;
F25=E4*A*A*A*B4*d0+E4*A*A*B4*d1+E4*A*B4*d2+E4*B4*d3+E5;

F31=E4*v0;
F32=E4*A*v0;
F33=E4*A*A*v0;
F34=E4*A*A*A*v0;
F35=E4*A*A*A*A*v0;

F1=[F11;F12;F13;F14;F15];
F2=[F21;F22;F23;F24;F25];
F3=[F31;F32;F33;F34;F35];
%% cost function 1
L2=sym(zeros(1,78*5));

L2(1,1)=0.5*dt ;
L2(1,2)=0.5*dt ;
L2(1,3)=0.5*dt ;
L2(1,4)=0.5*dt ;
L2(1,5)=0.5*dt ;
L2(1,6)=0.5*dt ;
L2(1,7)=0.5*dt ;
L2(1,8)=0.5*dt ;
L2(1,9)=0.5*dt ;
L2(1,10)=0.5*dt ;
L2(1,11)=0.5*dt ;
L2(1,12)=0.5*dt ;   %operation cost from manipulation flows

L2(1,42)=1;
L2(1,43)=1;
L2(1,45)=1;
L2(1,46)=1;
L2(1,48)=1;
L2(1,49)=1;
L2(1,50)=1;
L2(1,52)=1;
L2(1,54)=1;

L2(1,13)=-v1m;
L2(1,14)=-v2m;
L2(1,16)=-v4m;
L2(1,17)=-v5m;
L2(1,19)=-v7m;
L2(1,20)=-v8m;
L2(1,21)=-v9m;
L2(1,23)=-v11m;
L2(1,25)=-v13m;   %tank over flow

L2(1,26)=-dt*qs1m;
L2(1,27)=dt*beta1*v1m;
L2(1,55)=dt*beta1;
L2(1,56)=-dt*beta1;
L2(1,71)=-dt;     %qs1 overflow
```

```
L2(1,28)=-dt*qs2m;
L2(1,29)=dt*beta2*v2m;
L2(1,57)=dt*beta2;
L2(1,58)=-dt*beta2;
L2(1,72)=-dt;     %qs2 overflow

L2(1,30)=-dt*qs4m;
L2(1,31)=dt*beta4*v4m;
L2(1,59)=dt*beta4;
L2(1,60)=-dt*beta4;
L2(1,73)=-dt;     %qs4 overflow

L2(1,32)=-dt*qs5m;
L2(1,33)=dt*beta5*v5m;
L2(1,61)=dt*beta5;
L2(1,62)=-dt*beta5;
L2(1,74)=-dt;     %qs5 overflow

L2(1,34)=-dt*qs7m;
L2(1,35)=dt*beta7*v7m;
L2(1,63)=dt*beta7;
L2(1,64)=-dt*beta7;
L2(1,75)=-dt;     %qs7 overflow

L2(1,36)=-dt*qs8m;
L2(1,37)=dt*beta8*v8m;
L2(1,65)=dt*beta8;
L2(1,66)=-dt*beta8;
L2(1,76)=-dt;     %qs8 overflow

L2(1,38)=-dt*qs9m;
L2(1,39)=dt*beta9*v9m;
L2(1,67)=dt*beta9;
L2(1,68)=-dt*beta9;
L2(1,77)=-dt;     %qs9 overflow

L2(1,40)=-dt*qs11m;
L2(1,41)=dt*beta11*v11m;
L2(1,69)=dt*beta11;
L2(1,70)=-dt*beta11;
L2(1,78)=-dt;     %qs11 overflow
%% cost function 2

L2(1,1+78)=0.5*dt ;
L2(1,2+78)=0.5*dt ;
L2(1,3+78)=0.5*dt ;
L2(1,4+78)=0.5*dt ;
L2(1,5+78)=0.5*dt ;
L2(1,6+78)=0.5*dt ;
L2(1,7+78)=0.5*dt ;
L2(1,8+78)=0.5*dt ;
L2(1,9+78)=0.5*dt ;
L2(1,10+78)=0.5*dt ;
L2(1,11+78)=0.5*dt ;
L2(1,12+78)=0.5*dt ;   %operation cost from manipulation flows

L2(1,42+78)=1;
L2(1,43+78)=1;
L2(1,45+78)=1;
L2(1,46+78)=1;
L2(1,48+78)=1;
L2(1,49+78)=1;
L2(1,50+78)=1;
L2(1,52+78)=1;
L2(1,54+78)=1;

L2(1,13+78)=-v1m;
L2(1,14+78)=-v2m;
L2(1,16+78)=-v4m;
L2(1,17+78)=-v5m;
L2(1,19+78)=-v7m;
```

```
L2(1,20+78)=-v8m;
L2(1,21+78)=-v9m;
L2(1,23+78)=-v11m;
L2(1,25+78)=-v13m;   %tank over flow

L2(1,26+78)=-dt*qs1m;
L2(1,27+78)=dt*beta1*v1m;
L2(1,55+78)=dt*beta1;
L2(1,56+78)=-dt*beta1;
L2(1,71+78)=-dt;     %qs1 overflow

L2(1,28+78)=-dt*qs2m;
L2(1,29+78)=dt*beta2*v2m;
L2(1,57+78)=dt*beta2;
L2(1,58+78)=-dt*beta2;
L2(1,72+78)=-dt;     %qs2 overflow

L2(1,30+78)=-dt*qs4m;
L2(1,31+78)=dt*beta4*v4m;
L2(1,59+78)=dt*beta4;
L2(1,60+78)=-dt*beta4;
L2(1,73+78)=-dt;     %qs4 overflow

L2(1,32+78)=-dt*qs5m;
L2(1,33+78)=dt*beta5*v5m;
L2(1,61+78)=dt*beta5;
L2(1,62+78)=-dt*beta5;
L2(1,74+78)=-dt;     %qs5 overflow

L2(1,34+78)=-dt*qs7m;
L2(1,35+78)=dt*beta7*v7m;
L2(1,63+78)=dt*beta7;
L2(1,64+78)=-dt*beta7;
L2(1,75+78)=-dt;     %qs7 overflow

L2(1,36+78)=-dt*qs8m;
L2(1,37+78)=dt*beta8*v8m;
L2(1,65+78)=dt*beta8;
L2(1,66+78)=-dt*beta8;
L2(1,76+78)=-dt;     %qs8 overflow

L2(1,38+78)=-dt*qs9m;
L2(1,39+78)=dt*beta9*v9m;
L2(1,67+78)=dt*beta9;
L2(1,68+78)=-dt*beta9;
L2(1,77+78)=-dt;     %qs9 overflow

L2(1,40+78)=-dt*qs11m;
L2(1,41+78)=dt*beta11*v11m;
L2(1,69+78)=dt*beta11;
L2(1,70+78)=-dt*beta11;
L2(1,78+78)=-dt;     %qs11 overflow
%% cost function 3
L2(1,1+78*2)=0.5*dt ;
L2(1,2+78*2)=0.5*dt ;
L2(1,3+78*2)=0.5*dt ;
L2(1,4+78*2)=0.5*dt ;
L2(1,5+78*2)=0.5*dt ;
L2(1,6+78*2)=0.5*dt ;
L2(1,7+78*2)=0.5*dt ;
L2(1,8+78*2)=0.5*dt ;
L2(1,9+78*2)=0.5*dt ;
L2(1,10+78*2)=0.5*dt ;
L2(1,11+78*2)=0.5*dt ;
L2(1,12+78*2)=0.5*dt ;   %operation cost from manipulation flows

L2(1,42+78*2)=1;
L2(1,43+78*2)=1;
L2(1,45+78*2)=1;
L2(1,46+78*2)=1;
L2(1,48+78*2)=1;
```

```
L2(1,49+78*2)=1;
L2(1,50+78*2)=1;
L2(1,52+78*2)=1;
L2(1,54+78*2)=1;

L2(1,13+78*2)=-v1m;
L2(1,14+78*2)=-v2m;
L2(1,16+78*2)=-v4m;
L2(1,17+78*2)=-v5m;
L2(1,19+78*2)=-v7m;
L2(1,20+78*2)=-v8m;
L2(1,21+78*2)=-v9m;
L2(1,23+78*2)=-v11m;
L2(1,25+78*2)=-v13m;   %tank over flow

L2(1,26+78*2)=-dt*qs1m;
L2(1,27+78*2)=dt*beta1*v1m;
L2(1,55+78*2)=dt*beta1;
L2(1,56+78*2)=-dt*beta1;
L2(1,71+78*2)=-dt;     %qs1 overflow

L2(1,28+78*2)=-dt*qs2m;
L2(1,29+78*2)=dt*beta2*v2m;
L2(1,57+78*2)=dt*beta2;
L2(1,58+78*2)=-dt*beta2;
L2(1,72+78*2)=-dt;     %qs2 overflow

L2(1,30+78*2)=-dt*qs4m;
L2(1,31+78*2)=dt*beta4*v4m;
L2(1,59+78*2)=dt*beta4;
L2(1,60+78*2)=-dt*beta4;
L2(1,73+78*2)=-dt;     %qs4 overflow

L2(1,32+78*2)=-dt*qs5m;
L2(1,33+78*2)=dt*beta5*v5m;
L2(1,61+78*2)=dt*beta5;
L2(1,62+78*2)=-dt*beta5;
L2(1,74+78*2)=-dt;     %qs5 overflow

L2(1,34+78*2)=-dt*qs7m;
L2(1,35+78*2)=dt*beta7*v7m;
L2(1,63+78*2)=dt*beta7;
L2(1,64+78*2)=-dt*beta7;
L2(1,75+78*2)=-dt;     %qs7 overflow

L2(1,36+78*2)=-dt*qs8m;
L2(1,37+78*2)=dt*beta8*v8m;
L2(1,65+78*2)=dt*beta8;
L2(1,66+78*2)=-dt*beta8;
L2(1,76+78*2)=-dt;     %qs8 overflow

L2(1,38+78*2)=-dt*qs9m;
L2(1,39+78*2)=dt*beta9*v9m;
L2(1,67+78*2)=dt*beta9;
L2(1,68+78*2)=-dt*beta9;
L2(1,77+78*2)=-dt;     %qs9 overflow

L2(1,40+78*2)=-dt*qs11m;
L2(1,41+78*2)=dt*beta11*v11m;
L2(1,69+78*2)=dt*beta11;
L2(1,70+78*2)=-dt*beta11;
L2(1,78+78*2)=-dt;     %qs11 overflow
%% cost function 4

L2(1,1+78*3)=0.5*dt ;
L2(1,2+78*3)=0.5*dt ;
L2(1,3+78*3)=0.5*dt ;
L2(1,4+78*3)=0.5*dt ;
L2(1,5+78*3)=0.5*dt ;
L2(1,6+78*3)=0.5*dt ;
L2(1,7+78*3)=0.5*dt ;
```

```
L2(1,8+78*3)=0.5*dt ;
L2(1,9+78*3)=0.5*dt ;
L2(1,10+78*3)=0.5*dt ;
L2(1,11+78*3)=0.5*dt ;
L2(1,12+78*3)=0.5*dt ;   %operation cost from manipulation flows

L2(1,42+78*3)=1;
L2(1,43+78*3)=1;
L2(1,45+78*3)=1;
L2(1,46+78*3)=1;
L2(1,48+78*3)=1;
L2(1,49+78*3)=1;
L2(1,50+78*3)=1;
L2(1,52+78*3)=1;
L2(1,54+78*3)=1;

L2(1,13+78*3)=-v1m;
L2(1,14+78*3)=-v2m;
L2(1,16+78*3)=-v4m;
L2(1,17+78*3)=-v5m;
L2(1,19+78*3)=-v7m;
L2(1,20+78*3)=-v8m;
L2(1,21+78*3)=-v9m;
L2(1,23+78*3)=-v11m;
L2(1,25+78*3)=-v13m;   %tank over flow

L2(1,26+78*3)=-dt*qs1m;
L2(1,27+78*3)=dt*beta1*v1m;
L2(1,55+78*3)=dt*beta1;
L2(1,56+78*3)=-dt*beta1;
L2(1,71+78*3)=-dt;     %qs1 overflow

L2(1,28+78*3)=-dt*qs2m;
L2(1,29+78*3)=dt*beta2*v2m;
L2(1,57+78*3)=dt*beta2;
L2(1,58+78*3)=-dt*beta2;
L2(1,72+78*3)=-dt;     %qs2 overflow

L2(1,30+78*3)=-dt*qs4m;
L2(1,31+78*3)=dt*beta4*v4m;
L2(1,59+78*3)=dt*beta4;
L2(1,60+78*3)=-dt*beta4;
L2(1,73+78*3)=-dt;     %qs4 overflow

L2(1,32+78*3)=-dt*qs5m;
L2(1,33+78*3)=dt*beta5*v5m;
L2(1,61+78*3)=dt*beta5;
L2(1,62+78*3)=-dt*beta5;
L2(1,74+78*3)=-dt;     %qs5 overflow

L2(1,34+78*3)=-dt*qs7m;
L2(1,35+78*3)=dt*beta7*v7m;
L2(1,63+78*3)=dt*beta7;
L2(1,64+78*3)=-dt*beta7;
L2(1,75+78*3)=-dt;     %qs7 overflow

L2(1,36+78*3)=-dt*qs8m;
L2(1,37+78*3)=dt*beta8*v8m;
L2(1,65+78*3)=dt*beta8;
L2(1,66+78*3)=-dt*beta8;
L2(1,76+78*3)=-dt;     %qs8 overflow

L2(1,38+78*3)=-dt*qs9m;
L2(1,39+78*3)=dt*beta9*v9m;
L2(1,67+78*3)=dt*beta9;
L2(1,68+78*3)=-dt*beta9;
L2(1,77+78*3)=-dt;     %qs9 overflow

L2(1,40+78*3)=-dt*qs11m;
L2(1,41+78*3)=dt*beta11*v11m;
L2(1,69+78*3)=dt*beta11;
```

```
L2(1,70+78*3)=-dt*beta11;
L2(1,78+78*3)=-dt;     %qs11 overflow
%% cost function 5
L2(1,1+78*4)=0.5*dt ;
L2(1,2+78*4)=0.5*dt ;
L2(1,3+78*4)=0.5*dt ;
L2(1,4+78*4)=0.5*dt ;
L2(1,5+78*4)=0.5*dt ;
L2(1,6+78*4)=0.5*dt ;
L2(1,7+78*4)=0.5*dt ;
L2(1,8+78*4)=0.5*dt ;
L2(1,9+78*4)=0.5*dt ;
L2(1,10+78*4)=0.5*dt ;
L2(1,11+78*4)=0.5*dt ;
L2(1,12+78*4)=0.5*dt ;   %operation cost from manipulation flows

L2(1,42+78*4)=1;
L2(1,43+78*4)=1;
L2(1,45+78*4)=1;
L2(1,46+78*4)=1;
L2(1,48+78*4)=1;
L2(1,49+78*4)=1;
L2(1,50+78*4)=1;
L2(1,52+78*4)=1;
L2(1,54+78*4)=1;

L2(1,13+78*4)=-v1m;
L2(1,14+78*4)=-v2m;
L2(1,16+78*4)=-v4m;
L2(1,17+78*4)=-v5m;
L2(1,19+78*4)=-v7m;
L2(1,20+78*4)=-v8m;
L2(1,21+78*4)=-v9m;
L2(1,23+78*4)=-v11m;
L2(1,25+78*4)=-v13m;   %tank over flow

L2(1,26+78*4)=-dt*qs1m;
L2(1,27+78*4)=dt*beta1*v1m;
L2(1,55+78*4)=dt*beta1;
L2(1,56+78*4)=-dt*beta1;
L2(1,71+78*4)=-dt;     %qs1 overflow

L2(1,28+78*4)=-dt*qs2m;
L2(1,29+78*4)=dt*beta2*v2m;
L2(1,57+78*4)=dt*beta2;
L2(1,58+78*4)=-dt*beta2;
L2(1,72+78*4)=-dt;     %qs2 overflow

L2(1,30+78*4)=-dt*qs4m;
L2(1,31+78*4)=dt*beta4*v4m;
L2(1,59+78*4)=dt*beta4;
L2(1,60+78*4)=-dt*beta4;
L2(1,73+78*4)=-dt;     %qs4 overflow

L2(1,32+78*4)=-dt*qs5m;
L2(1,33+78*4)=dt*beta5*v5m;
L2(1,61+78*4)=dt*beta5;
L2(1,62+78*4)=-dt*beta5;
L2(1,74+78*4)=-dt;     %qs5 overflow

L2(1,34+78*4)=-dt*qs7m;
L2(1,35+78*4)=dt*beta7*v7m;
L2(1,63+78*4)=dt*beta7;
L2(1,64+78*4)=-dt*beta7;
L2(1,75+78*4)=-dt;     %qs7 overflow

L2(1,36+78*4)=-dt*qs8m;
L2(1,37+78*4)=dt*beta8*v8m;
L2(1,65+78*4)=dt*beta8;
L2(1,66+78*4)=-dt*beta8;
L2(1,76+78*4)=-dt;     %qs8 overflow
```

```
L2(1,38+78*4)=-dt*qs9m;
L2(1,39+78*4)=dt*beta9*v9m;
L2(1,67+78*4)=dt*beta9;
L2(1,68+78*4)=-dt*beta9;
L2(1,77+78*4)=-dt;     %qs9 overflow

L2(1,40+78*4)=-dt*qs11m;
L2(1,41+78*4)=dt*beta11*v11m;
L2(1,69+78*4)=dt*beta11;
L2(1,70+78*4)=-dt*beta11;
L2(1,78+78*4)=-dt;     %qs11 overflow
%% constants specification
eps =1e-20;
S=[323576,164869,5076,754131];%surface area
phi=[1.03,10.4,0,0.48];%absorption coefficient
beta=[7.1e-4,5.8e-4,2e-3,1e-3];%volumetric flow coefficient
vm=[16901,43000,35000,26659];%maximum volume
dt=300;%sample time

% x0=[v1m,v2m/2,v3m/4,v4m/2.2,0,0,0,0,0,0,0,0,0];
% x0=[v1m,v2m,v3m,v4m,v5m,v6m,v7m,v8m,v9m,v10m,v11m,v12m,v13m];
% x0=[8020,43008,8370,11427,8020,0,43008,14868,43008,1716,14696,404,16551];
x0 = zeros(1,13);
% x0 = [ 4332.7;
%         22290;
%             0;
%        4705.8;
%        4332.7;
%             0;
%         22290;
%        4332.7;
%         22290;
%             0;
%        4705.8;
%             0;
%        4705.8];
% x0=[9075.6;
%         47561;
%        2872.8;
%         12299;
%        9075.6;
%             0;
%         47561;
%         12954;
%         47561;
%             0;
%         13326;
%             0;
%         13194];
% x0 = [  13475;
%         68096;
%         11022;
%         18686;
%         13475;
%             0;
%         68096;
%         16901;
%         68096;
%         14590;
%         16206;
%             0;
%         26560];
%  x0 = [ 18270;
%         74955;
%         21374;
%         24276;
%         18270;
%             0;
%         74955;
%         20967;
```

```
%          74955;
%          29554;
%          19670;
%          13198;
%          27788];

% x0=[6956,36453,2673,8970,6956,0,36456,7042,36453,2897,10223,0,10121];
% x0 = [16634,17146,0,6362,3333,0,17146,3333,17146,0,3620,0,3620];



% rain=[200,200,200,200,200];
% rain=[100,110,120,130,140];
% rain = [200,200,200,200,200];
% rain=[130,170,190,230,170];
% rain = [190,250,190,0,0];
% rain = [130,160,210,250,180];
% rain = [170,200,270,170,0];
% rain = [190,250,190,0,0];
% rain = [230,170,0,0,0];
% rain = [170,0,0,0,0];
% rain = [100,0,0,0,0];
rain = [30,65,55,48,49];
% rain = [100,0,0,0,0];

p = rain.*1.2./1000./3600;% precipitation in unit of m/s
v1m=vm(1);
v2m=vm(2);
v3m=vm(3);
v4m=vm(4);
v5m=vm(1);
v6m=vm(3);
v7m=vm(2);
v8m=vm(1);
v9m=vm(2);
v10m=vm(3);
v11m=vm(4);
v12m=vm(3);
v13m=vm(4);


% x0=[v1m/2.2-eps,v2m/3.3-eps,v3m/4.4-eps,v4m/2.5-eps,v5m/3.1-eps,v6m/2.9-eps,v7m/1.9-
eps,v8m/1.8-eps,v9m/2.1-eps,v10m/1.7-eps,v11m/1.6-eps,v12m/3.2-eps,v13m/2.1-eps];

v01=x0(1);
v02=x0(2);
v03=x0(3);
v04=x0(4);
v05=x0(5);
v06=x0(6);
v07=x0(7);
v08=x0(8);
v09=x0(9);
v010=x0(10);
v011=x0(11);
v012=x0(12);
v013=x0(13);

S1=S(1);
S2=S(2);
S4=S(4);
S5=S(1);
S7=S(2);
S8=S(1);
S9=S(2);
S11=S(4);
S13=S(4);

phi1=phi(1);
phi2=phi(2);
phi4=phi(4);
```

```
phi5=phi(1);
phi7=phi(2);
phi8=phi(1);
phi9=phi(2);
phi11=phi(4);

phi13=phi(4);

beta1=beta(1);
beta2=beta(2);
beta3=beta(3);
beta4=beta(4);
beta5=beta(1);
beta6=beta(3);
beta7=beta(2);
beta8=beta(1);
beta9=beta(2);
beta10=beta(3);
beta11=beta(4);
beta12=beta(3);
beta13=beta(4);



p01=p(1);
p02=p(1);
p03=0;%real
p04=p(1);
p05=p(1);
p06=0;%real
p07=p(1);
p08=p(1);
p09=p(1);
p010=0;%real
p011=p(1);
p012=0;%real
p013=p(1);


% p01=88*0.84;
% p02=92*0.84;
% p03=0;
% p04=77*0.84;
% p05=65*0.84;
% p06=0;
% p07=79*0.84;
% p08=63*0.84;
% p09=78*0.84;
% p010=0;
% p011=69*0.84;
% p012=0;
% p013=46*0.84;

p11=p(2);
p12=p(2);
p13=0;%real
p14=p(2);
p15=p(2);
p16=0;%real
p17=p(2);
p18=p(2);
p19=p(2);
p110=0;%real
p111=p(2);
p112=0;%real
p113=p(2);

% p11=102*0.84;
% p12=98*0.84;
% p13=0;%real
% p14=96*0.84;
```

```
% p15=73*0.84;
% p16=0;%real
% p17=72*0.84;
% p18=69*0.84;
% p19=36*0.84;
% p110=0;%real
% p111=90*0.84;
% p112=0;%real
% p113=59*0.84;

p21=p(3);
p22=p(3);
p23=0;%real
p24=p(3);
p25=p(3);
p26=0;%real
p27=p(3);
p28=p(3);
p29=p(3);
p210=0;%real
p211=p(3);
p212=0;%real
p213=p(3);


% p21=102*0.84;
% p22=105*0.84;
% p23=0;
% p24=104*0.84;
% p25=112*0.84;
% p26=0;
% p27=103*0.84;
% p28=106*0.84;
% p29=116*0.84;
% p210=0;
% p211=107*0.84;
% p212=0;
% p213=99*0.84;

p31=p(4);
p32=p(4);
p33=0;
p34=p(4);
p35=p(4);
p36=0;
p37=p(4);
p38=p(4);
p39=p(4);
p310=0;
p311=p(4);
p312=0;
p313=p(4);

%
% p31=89*0.84;
% p32=79*0.84;
% p33=0;
% p34=90*0.84;
% p35=96*0.84;
% p36=0;
% p37=76*0.84;
% p38=77*0.84;
% p39=69*0.84;
% p310=0;
% p311=75*0.84;
% p312=0;
% p313=49*0.84;

p41=p(5);
p42=p(5);
p43=0;
```

```
p44=p(5);
p45=p(5);
p46=0;
p47=p(5);
p48=p(5);
p49=p(5);
p410=0;
p411=p(5);
p412=0;
p413=p(5);

% for i=1:13
%     d0(i)=p(1);
% end    % d0
% d0=eval(d0);
% for i=1:13
%     d1(i)=p(2);
% end    %d1
% d1=eval(d1);
% for i=1:13
%     d2(i)=p(3);
% end    %d2
% d2=eval(d2);
% for i=1:13
%     d3(i)=p(4);
% end    %d3
% d3=eval(d3);
% for i=1:13
%     d4(i)=p(5);
% end    %d4
% d4=eval(d4);

% qs1m=9.14;
% qs2m=3.4;
% qs4m=10;
% qs5m=6.8;
% qs7m=15;
% qs8m=5;
% qs9m=15.6;
% qs11m=18;

qs1m=1000;
qs2m=1000;
qs4m=1000;
qs5m=1000;
qs7m=1000;
qs8m=1000;
qs9m=1000;
qs11m=1000;

v1u=v1m*2+S1*dt*max(p)*phi1;
v2u=v2m*2+S2*dt*max(p)*phi2;
v4u=v4m*2+S4*dt*max(p)*phi4;
v5u=v5m*2+S5*dt*max(p)*phi5;
v7u=v7m*2+S7*dt*max(p)*phi7;
v8u=v8m*2+S8*dt*max(p)*phi8;
v9u=v9m*2+S9*dt*max(p)*phi9;
v11u=v11m*2+S11*dt*max(p)*phi11;
v13u=v13m*2+S13*dt*max(p)*phi13;

v3u=35000;
v6u=35000;
v10u=35000;
v12u=35000;

% v1u=58432;
% v2u=58432;
% v4u=58432;
% v5u=58432;
% v7u=58432;
% v8u=58432;
```

```
% v9u=58432;
% v11u=58432;
% v13u=58432;
% v3u=58432;
% v6u=58432;
% v10u=58432;
% v12u=58432;
%
vu=[v1u,v2u,v3u,v4u,v5u,v6u,v7u,v8u,v9u,v10u,v11u,v12u,v13u];


v1l=0;
v2l=0;
v4l=0;
v5l=0;
v7l=0;
v8l=0;
v9l=0;
v11l=0;
v13l=0;



qu1m=9.1;
qu2m=25;
qu3m=12;
qu4m=25;
qu5m=9.1;
qu6m=7;
qu7m=25;
qu8m=9.1;
qu9m=25;
qu10m=30;
qu11m=25;
qu12m=40;

M1=v1u-v1m;
m1=-v1m;

M2=v2u-v2m;
m2=-v2m;

M4=v4u-v4m;
m4=-v4m;

M5=v5u-v5m;
m5=-v5m;

M7=v7u-v7m;
m7=-v7m;

M8=v8u-v8m;
m8=-v8m;

M9=v9u-v9m;
m9=-v9m;

% M10=v10u-v10m;
% m10=-v10m;

M11=v11u-v11m;
m11=-v11m;

M13=v13u-v13m;
m13=-v13m;

lb = zeros(78*5,1);                  %Bounds on x (lb <= x)
uub = [qu1m;qu2m;qu3m;qu4m;qu5m;qu6m;qu7m;qu8m;qu9m;qu10m;qu11m;qu12m];         %u upper bound


dub = ones(29,1);                 %delta upperbound
```

```
zub = zeros(37,1);            %z upper bound
for i=1:37
    zub(i,1)=inf;
end

ub = [uub;dub;zub;uub;dub;zub;uub;dub;zub;uub;dub;zub;uub;dub;zub];
% ub = zeros(78*5,1);




f=eval(L2');
% lyc = zeros(1030,390);
% for i=1:1030
%     for j=1:390
%         lyc(i,j)=eval(F1(i,j));
%         disp(i)
%         disp(j)
%     end
% end

a=eval(F1);
b=eval(F2+F3);
%%

xtype =
'CCCCCCCCCCCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCC';

Opt = opti('f',f,'ineq',a,b,'bounds',lb,ub,'xtype',xtype);
[x,fval,exitflag,info] = solve(Opt);


u01=x(1);
u02=x(2);
u03=x(3);
u04=x(4);
u05=x(5);
u06=x(6);
u07=x(7);
u08=x(8);
u09=x(9);
u010=x(10);
u011=x(11);
u012=x(12);

u11=x(1+78);
u12=x(2+78);
u13=x(3+78);
u14=x(4+78);
u15=x(5+78);
u16=x(6+78);
u17=x(7+78);
u18=x(8+78);
u19=x(9+78);
u110=x(10+78);
u111=x(11+78);
u112=x(12+78);

u21=x(1+78*2);
u22=x(2+78*2);
u23=x(3+78*2);
u24=x(4+78*2);
u25=x(5+78*2);
u26=x(6+78*2);
u27=x(7+78*2);
```

```
u28=x(8+78*2);
u29=x(9+78*2);
u210=x(10+78*2);
u211=x(11+78*2);
u212=x(12+78*2);

u31=x(1+78*3);
u32=x(2+78*3);
u33=x(3+78*3);
u34=x(4+78*3);
u35=x(5+78*3);
u36=x(6+78*3);
u37=x(7+78*3);
u38=x(8+78*3);
u39=x(9+78*3);
u310=x(10+78*3);
u311=x(11+78*3);
u312=x(12+78*3);

u41=x(1+78*4);
u42=x(2+78*4);
u43=x(3+78*4);
u44=x(4+78*4);
u45=x(5+78*4);
u46=x(6+78*4);
u47=x(7+78*4);
u48=x(8+78*4);
u49=x(9+78*4);
u410=x(10+78*4);
u411=x(11+78*4);
u412=x(12+78*4);

tle=table;
tle.Control1=[x(1);x(1+78);x(1+78*2);x(1+78*3);x(1+78*4)];
tle.Control2=[x(2);x(2+78);x(2+78*2);x(2+78*3);x(2+78*4)];
tle.Control3=[x(3);x(3+78);x(3+78*2);x(3+78*3);x(3+78*4)];
tle.Control4=[x(4);x(4+78);x(4+78*2);x(4+78*3);x(4+78*4)];
tle.Control5=[x(5);x(5+78);x(5+78*2);x(5+78*3);x(5+78*4)];
tle.Control6=[x(6);x(6+78);x(6+78*2);x(6+78*3);x(6+78*4)];
tle.Control7=[x(7);x(7+78);x(7+78*2);x(7+78*3);x(7+78*4)];
tle.Control8=[x(8);x(8+78);x(8+78*2);x(8+78*3);x(8+78*4)];
tle.Control9=[x(9);x(9+78);x(9+78*2);x(9+78*3);x(9+78*4)];
tle.Control10=[x(10);x(10+78);x(10+78*2);x(10+78*3);x(10+78*4)];
tle.Control11=[x(11);x(11+78);x(11+78*2);x(11+78*3);x(11+78*4)];
tle.Control12=[x(12);x(12+78);x(12+78*2);x(12+78*3);x(12+78*4)];

delta01=x(13);
delta02=x(14);
delta03=x(15);
delta04=x(16);
delta05=x(17);
delta06=x(18);
delta07=x(19);
delta08=x(20);
delta09=x(21);
delta010=x(22);
delta011=x(23);
delta012=x(24);
delta013=x(25);
delta0S1=x(26);
delta0S1a1=x(27);
delta0S2=x(28);
delta0S2a2=x(29);
delta0S4=x(30);
delta0S4a4=x(31);
delta0S5=x(32);
delta0S5a5=x(33);
delta0S7=x(34);
delta0S7a7=x(35);
delta0S8=x(36);
delta0S8a8=x(37);
```

```
delta0S9=x(38);
delta0S9a9=x(39);
delta0S11=x(40);
delta0Sa11=x(41);

delta11=x(13+78);
delta12=x(14+78);
delta13=x(15+78);
delta14=x(16+78);
delta15=x(17+78);
delta16=x(18+78);
delta17=x(19+78);
delta18=x(20+78);
delta19=x(21+78);
delta110=x(22+78);
delta111=x(23+78);
delta112=x(24+78);
delta113=x(25+78);
delta1S1=x(26+78);
delta1S1a1=x(27+78);
delta1S2=x(28+78);
delta1S2a2=x(29+78);
delta1S4=x(30+78);
delta1S4a4=x(31+78);
delta1S5=x(32+78);
delta1S5a5=x(33+78);
delta1S7=x(34+78);
delta1S7a7=x(35+78);
delta1S8=x(36+78);
delta1S8a8=x(37+78);
delta1S9=x(38+78);
delta1S9a9=x(39+78);
delta1S11=x(40+78);
delta1Sa11=x(41+78);

delta21=x(13+78*2);
delta22=x(14+78*2);
delta23=x(15+78*2);
delta24=x(16+78*2);
delta25=x(17+78*2);
delta26=x(18+78*2);
delta27=x(19+78*2);
delta28=x(20+78*2);
delta29=x(21+78*2);
delta210=x(22+78*2);
delta211=x(23+78*2);
delta212=x(24+78*2);
delta213=x(25+78*2);
delta2S1=x(26+78*2);
delta2S1a1=x(27+78*2);
delta2S2=x(28+78*2);
delta2S2a2=x(29+78*2);
delta2S4=x(30+78*2);
delta2S4a4=x(31+78*2);
delta2S5=x(32+78*2);
delta2S5a5=x(33+78*2);
delta2S7=x(34+78*2);
delta2S7a7=x(35+78*2);
delta2S8=x(36+78*2);
delta2S8a8=x(37+78*2);
delta2S9=x(38+78*2);
delta2S9a9=x(39+78*2);
delta2S11=x(40+78*2);
delta2Sa11=x(41+78*2);

delta31=x(13+78*3);
delta32=x(14+78*3);
delta33=x(15+78*3);
delta34=x(16+78*3);
delta35=x(17+78*3);
delta36=x(18+78*3);
```

```
delta37=x(19+78*3);
delta38=x(20+78*3);
delta39=x(21+78*3);
delta310=x(22+78*3);
delta311=x(23+78*3);
delta312=x(24+78*3);
delta313=x(25+78*3);
delta3S1=x(26+78*3);
delta3S1a1=x(27+78*3);
delta3S2=x(28+78*3);
delta3S2a2=x(29+78*3);
delta3S4=x(30+78*3);
delta3S4a4=x(31+78*3);
delta3S5=x(32+78*3);
delta3S5a5=x(33+78*3);
delta3S7=x(34+78*3);
delta3S7a7=x(35+78*3);
delta3S8=x(36+78*3);
delta3S8a8=x(37+78*3);
delta3S9=x(38+78*3);
delta3S9a9=x(39+78*3);
delta3S11=x(40+78*3);
delta3Sa11=x(41+78*3);

delta41=x(13+78*4);
delta42=x(14+78*4);
delta43=x(15+78*4);
delta44=x(16+78*4);
delta45=x(17+78*4);
delta46=x(18+78*4);
delta47=x(19+78*4);
delta48=x(20+78*4);
delta49=x(21+78*4);
delta410=x(22+78*4);
delta411=x(23+78*4);
delta412=x(24+78*4);
delta413=x(25+78*4);
delta4S1=x(26+78*4);
delta4S1a1=x(27+78*4);
delta4S2=x(28+78*4);
delta4S2a2=x(29+78*4);
delta4S4=x(30+78*4);
delta4S4a4=x(31+78*4);
delta4S5=x(32+78*4);
delta4S5a5=x(33+78*4);
delta4S7=x(34+78*4);
delta4S7a7=x(35+78*4);
delta4S8=x(36+78*4);
delta4S8a8=x(37+78*4);
delta4S9=x(38+78*4);
delta4S9a9=x(39+78*4);
delta4S11=x(40+78*4);
delta4Sa11=x(41+78*4);

z01=x(42);
z02=x(43);
z03=x(44);
z04=x(45);
z05=x(46);
z06=x(47);
z07=x(48);
z08=x(49);
z09=x(50);
z010=x(51);
z011=x(52);
z012=x(53);
z013=x(54);
z0S1a1=x(55);
z0S1a1a1=x(56);
z0S2a2=x(57);
z0S2a2a2=x(58);
```

```
z0S4a4=x(59);
z0S4a4a4=x(60);
z0S5a5=x(61);
z0S5a5a5=x(62);
z0S7a7=x(63);
z0S7a7a7=x(64);
z0S8a8=x(65);
z0S8a8a8=x(66);
z0S9a9=x(67);
z0S9a9a9=x(68);
z0S11a11=x(69);
z0S11a11a11=x(70);
z0uS1=x(71);
z0uS2=x(72);
z0uS4=x(73);
z0uS5=x(74);
z0uS7=x(75);
z0uS8=x(76);
z0uS9=x(77);
z0uS11=x(78);


z11=x(42+78);
z12=x(43+78);
z13=x(44+78);
z14=x(45+78);
z15=x(46+78);
z16=x(47+78);
z17=x(48+78);
z18=x(49+78);
z19=x(50+78);
z110=x(51+78);
z111=x(52+78);
z112=x(53+78);
z113=x(54+78);
z1S1a1=x(55+78);
z1S1a1a1=x(56+78);
z1S2a2=x(57+78);
z1S2a2a2=x(58+78);
z1S4a4=x(59+78);
z1S4a4a4=x(60+78);
z1S5a5=x(61+78);
z1S5a5a5=x(62+78);
z1S7a7=x(63+78);
z1S7a7a7=x(64+78);
z1S8a8=x(65+78);
z1S8a8a8=x(66+78);
z1S9a9=x(67+78);
z1S9a9a9=x(68+78);
z1S11a11=x(69+78);
z1S11a11a11=x(70+78);
z1uS1=x(71+78);
z1uS2=x(72+78);
z1uS4=x(73+78);
z1uS5=x(74+78);
z1uS7=x(75+78);
z1uS8=x(76+78);
z1uS9=x(77+78);
z1uS11=x(78+78);

z21=x(42+78*2);
z22=x(43+78*2);
z23=x(44+78*2);
z24=x(45+78*2);
z25=x(46+78*2);
z26=x(47+78*2);
z27=x(48+78*2);
z28=x(49+78*2);
z29=x(50+78*2);
z210=x(51+78*2);
z211=x(52+78*2);
```

```
z212=x(53+78*2);
z213=x(54+78*2);
z2S1a1=x(55+78*2);
z2S1a1a1=x(56+78*2);
z2S2a2=x(57+78*2);
z2S2a2a2=x(58+78*2);
z2S4a4=x(59+78*2);
z2S4a4a4=x(60+78*2);
z2S5a5=x(61+78*2);
z2S5a5a5=x(62+78*2);
z2S7a7=x(63+78*2);
z2S7a7a7=x(64+78*2);
z2S8a8=x(65+78*2);
z2S8a8a8=x(66+78*2);
z2S9a9=x(67+78*2);
z2S9a9a9=x(68+78*2);
z2S11a11=x(69+78*2);
z2S11a11a11=x(70+78*2);
z2uS1=x(71+78*2);
z2uS2=x(72+78*2);
z2uS4=x(73+78*2);
z2uS5=x(74+78*2);
z2uS7=x(75+78*2);
z2uS8=x(76+78*2);
z2uS9=x(77+78*2);
z2uS11=x(78+78*2);

z31=x(42+78*3);
z32=x(43+78*3);
z33=x(44+78*3);
z34=x(45+78*3);
z35=x(46+78*3);
z36=x(47+78*3);
z37=x(48+78*3);
z38=x(49+78*3);
z39=x(50+78*3);
z310=x(51+78*3);
z311=x(52+78*3);
z312=x(53+78*3);
z313=x(54+78*3);
z3S1a1=x(55+78*3);
z3S1a1a1=x(56+78*3);
z3S2a2=x(57+78*3);
z3S2a2a2=x(58+78*3);
z3S4a4=x(59+78*3);
z3S4a4a4=x(60+78*3);
z3S5a5=x(61+78*3);
z3S5a5a5=x(62+78*3);
z3S7a7=x(63+78*3);
z3S7a7a7=x(64+78*3);
z3S8a8=x(65+78*3);
z3S8a8a8=x(66+78*3);
z3S9a9=x(67+78*3);
z3S9a9a9=x(68+78*3);
z3S11a11=x(69+78*3);
z3S11a11a11=x(70+78*3);
z3uS1=x(71+78*3);
z3uS2=x(72+78*3);
z3uS4=x(73+78*3);
z3uS5=x(74+78*3);
z3uS7=x(75+78*3);
z3uS8=x(76+78*3);
z3uS9=x(77+78*3);
z3uS11=x(78+78*3);

z41=x(42+78*4);
z42=x(43+78*4);
z43=x(44+78*4);
z44=x(45+78*4);
z45=x(46+78*4);
z46=x(47+78*4);
```

```
z47=x(48+78*4);
z48=x(49+78*4);
z49=x(50+78*4);
z410=x(51+78*4);
z411=x(52+78*4);
z412=x(53+78*4);
z413=x(54+78*4);
z4S1a1=x(55+78*4);
z4S1a1a1=x(56+78*4);
z4S2a2=x(57+78*4);
z4S2a2a2=x(58+78*4);
z4S4a4=x(59+78*4);
z4S4a4a4=x(60+78*4);
z4S5a5=x(61+78*4);
z4S5a5a5=x(62+78*4);
z4S7a7=x(63+78*4);
z4S7a7a7=x(64+78*4);
z4S8a8=x(65+78*4);
z4S8a8a8=x(66+78*4);
z4S9a9=x(67+78*4);
z4S9a9a9=x(68+78*4);
z4S11a11=x(69+78*4);
z4S11a11a11=x(70+78*4);
z4uS1=x(71+78*4);
z4uS2=x(72+78*4);
z4uS4=x(73+78*4);
z4uS5=x(74+78*4);
z4uS7=x(75+78*4);
z4uS8=x(76+78*4);
z4uS9=x(77+78*4);
z4uS11=x(78+78*4);





beta = [beta1,beta2,beta3,beta4,beta5,beta6,beta7,beta8,beta9,beta10,beta11,beta12,beta13];
betamatrix = eye(13);
for i=1:13
    betamatrix(i,i)=beta(i);
end

qout=eval(dt*betamatrix*v1);
eval(v1-qout);
action2 = [x(1:12),x(78+1:12+78),x(78*2+1:12+78*2),x(78*3+1:12+78*3),x(78*4+1:12+78*4)];
stepcost =eval([L2(1:78)*x(1:78);
           L2(78*1+1:78*2)*x(78*1+1:78*2);
           L2(78*2+1:78*3)*x(78*2+1:78*3);
           L2(78*3+1:78*4)*x(78*3+1:78*4);
           L2(78*4+1:78*5)*x(78*4+1:78*5)]);

rain0 = eval(B4*d0);
rain1 = eval(B4*d1);
rain2 = eval(B4*d2);
rain3 = eval(B4*d3);
rain4 = eval(B4*d4);

total_storage = v1m+v2m+v3m+v4m+v5m+v6m+v7m+v8m+v9m+v10m+v11m+v12m+v13m;
total_rainfall = ones(1,13)*rain0+ones(1,13)*rain1+ones(1,13)*rain2+ones(1,13)*rain3;
operation_cost2 =
eval(L2(1:12)*x(1:12)+L2(1+78:12+78)*x(1+78:12+78)+L2(1+78*2:12+78*2)*x(1+78*2:12+78*2)+L2(1+78*3
:12+78*3)*x(1+78*3:12+78*3)+L2(1+78*4:12+78*4)*x(1+78*4:12+78*4));

Logical_state0 = [x(13:25);x(26);x(28);x(30);x(32);x(34);x(36);x(38);x(40)];%
Logical_state1 =
[x(13+78*1:25+78*1);x(26+78*1);x(28+78*1);x(30+78*1);x(32+78*1);x(34+78*1);x(36+78*1);x(38+78*1);
x(40+78*1)];%
Logical_state2 =
[x(13+78*2:25+78*2);x(26+78*2);x(28+78*2);x(30+78*2);x(32+78*2);x(34+78*2);x(36+78*2);x(38+78*2);
x(40+78*2)];%
```

```
Logical_state3 =
[x(13+78*3:25+78*3);x(26+78*3);x(28+78*3);x(30+78*3);x(32+78*3);x(34+78*3);x(36+78*3);x(38+78*3);
x(40+78*3)];%
Logical_state4 =
[x(13+78*4:25+78*4);x(26+78*4);x(28+78*4);x(30+78*4);x(32+78*4);x(34+78*4);x(36+78*4);x(38+78*4);
x(40+78*4)];%

Logic_states_transpose =
[Logical_state0,Logical_state1,Logical_state2,Logical_state3,Logical_state4]';
logic = table;
logic.tank1 = Logic_states_transpose(:,1);
logic.tank2 = Logic_states_transpose(:,2);
logic.tank3 = Logic_states_transpose(:,3);
logic.tank4 = Logic_states_transpose(:,4);
logic.tank5 = Logic_states_transpose(:,5);
logic.tank6 = Logic_states_transpose(:,6);
logic.tank7 = Logic_states_transpose(:,7);
logic.tank8 = Logic_states_transpose(:,8);
logic.tank9 = Logic_states_transpose(:,9);
logic.tank10 = Logic_states_transpose(:,10);
logic.tank11 = Logic_states_transpose(:,11);
logic.tank12 = Logic_states_transpose(:,12);
logic.tank13 = Logic_states_transpose(:,13);
logic.sewage1 = Logic_states_transpose(:,14);
logic.sewage2 = Logic_states_transpose(:,15);
logic.sewage4 = Logic_states_transpose(:,16);
logic.sewage5 = Logic_states_transpose(:,17);
logic.sewage7 = Logic_states_transpose(:,18);
logic.sewage8 = Logic_states_transpose(:,19);
logic.sewage9 = Logic_states_transpose(:,20);
logic.sewage11 = Logic_states_transpose(:,21);

Final_states = [eval(v1),eval(v2),eval(v3),eval(v4)];
step_costs = [eval(L2(1:78*1)*x(1:78*1));
              eval(L2(1+78*1:78*2)*x(1+78*1:78*2));
              eval(L2(1+78*2:78*3)*x(1+78*2:78*3));
              eval(L2(1+78*3:78*4)*x(1+78*3:78*4));
              eval(L2(1+78*4:78*5)*x(1+78*4:78*5))];

% info.Time
controlsum =
[sum(action2(:,1)),sum(action2(:,2)),sum(action2(:,3)),sum(action2(:,4)),sum(action2(:,5)),];
```

Decentralized MPC main function

```
%%   Large system decentralized MPC
% sub1 initial
clear
sub1_x0 = [0;0;0;0];

% rain = [30,70,60,40,30;
%            65,56,55,32,42;
%               55,53,28,43,39;
%                  48,50,39,37,40;
%                     49,39,34,36,40];

rain = [90,110,120,130,90;
            95,110,126,86,76;
               108,123,77,69,53;
                   118,75,67,50,49;
                       68,61,48,39,20];

% rain = [100,100,100,100,100;
%         100,100,100,100,  0;
%         100,100,100,  0,  0;
%         100,100,  0,  0,  0;
%         100,  0,  0,  0,  0];
```

```
% rain = [200,200,200,200,200;
%         200,200,200,200,200;
%         200,200,200,200,200;
%         200,200,200,200,200;
%         200,200,200,200,200];

% rain = [130,160,210,250,180;
%             170,200,270,170,0;
%                 190,250,190,0,0;
%                     230,170,0,0,0;
%                         170,0,0,0,0];
% rain = [100,100,100,100,100;
% 100,100,100,100,100;
% 100,100,100,100,100;
% 100,100,100,100,100;
% 100,100,100,100,100];

% sub1_rain = [200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200];
% sub1_rain = [100,100,100,100,100;
%              100,100,100,100,  0;
%              100,100,100,  0,  0;
%              100,100,  0,  0,  0;
%              100,  0,  0,  0,  0];
% sub1_rain = [100,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0];
% sub1_rain =[130,160,210,250,180;
%                 170,200,270,170,0;
%                     190,250,190,0,0;
%                         230,170,0,0,0;
%                             170,0,0,0,0];
sub1_rain = rain;
sub1_disturbance = [0,0,0,0,0]; %qs5m = 6.8  could use rain profile to
% increase accuracy, if sub2 was calculated b4 sub1, no need to initialize
sub1_finalstates = zeros(4,5);
sub1_cost=zeros(1,5);
sub1_action=zeros(3,5);
sub1_cptime=zeros(1,5);

sub1_op_cost = zeros(1,5);
sub1_of_cost = zeros(1,5);

% sub2 initial
sub2_x0 = [0;0;0;0];
% sub2_rain = [200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200];
% sub2_rain = [100,100,100,100,100;
%              100,100,100,100,  0;
%              100,100,  0,  0,  0;
%              100,100,  0,  0,  0;
%              100,  0,  0,  0,  0];
% sub2_rain = [100,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0];
% sub2_rain =[130,160,210,250,180;
%                 170,200,270,170,0;
%                     190,250,190,0,0;
%                         230,170,0,0,0;
%                             170,0,0,0,0];
sub2_rain = rain;
```

```
sub2_finalstates = zeros(4,5);
sub2_cost=zeros(1,5);
sub2_action=zeros(4,5);
sub2_cptime=zeros(1,5);

sub1_op_cost = zeros(1,5);
sub1_of_cost = zeros(1,5);

% sub3 initial
sub3_x0 = [0;0;0];
% sub3_rain = [200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200];
% sub3_rain = [100,100,100,100,100;
%              100,100,100,100,  0;
%              100,100,  0,  0,  0;
%              100,100,  0,  0,  0;
%              100,  0,  0,  0,  0];
% sub3_rain = [100,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0];
% sub3_rain =[130,160,210,250,180;
%                 170,200,270,170,0;
%                     190,250,190,0,0;
%                         230,170,0,0,0;
%                             170,0,0,0,0];
sub3_rain = rain ;
sub3_disturbance = [0,0,0,0,0];
sub3_finalstates = zeros(3,5);
sub3_cost=zeros(1,5);
sub3_action=zeros(3,5);
sub3_cptime=zeros(1,5);

sub1_op_cost = zeros(1,5);
sub1_of_cost = zeros(1,5);


% sub4 initial
sub4_x0 = [0,0];
% sub4_rain = [200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200;
%              200,200,200,200,200];
% sub4_rain = [100,100,100,100,100;
%              100,100,100,100,  0;
%              100,100,  0,  0,  0;
%              100,100,  0,  0,  0;
%              100,  0,  0,  0,  0];
% sub4_rain = [100,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0;
%              0,0,0,0,0];
% sub4_rain =[130,160,210,250,180;
%                 170,200,270,170,0;
%                     190,250,190,0,0;
%                         230,170,0,0,0;
%                             170,0,0,0,0];
sub4_rain = rain ;
sub4_disturbance = [0,0,0,0,0;   %qout4
                    0,0,0,0,0;   %qu8
                    0,0,0,0,0;   %qs8
                    0,0,0,0,0;   %qu11
                    0,0,0,0,0];  %qs11
sub4_finalstates = zeros(2,5);
sub4_cost=zeros(1,5);
```

```
sub4_action=zeros(2,5);
sub4_cptime=zeros(1,5);

sub1_op_cost = zeros(1,5);
sub1_of_cost = zeros(1,5);

for i = 1:5
%subsystem 1

[sub1_endstates1,step_cost1,action1,cp_time1,qout4,op_cost1,of_cost1,suc1]=sub1function(sub1_x0,s
ub1_rain(i,:),sub1_disturbance);
    if suc1 == -1
        disp('Optimization Failure at subsystem 1')
    end
%     if i == 2
%         of_cost1
%         sub1_x0
%         action1
%     end
%     sub1_finalstates(:,i)=sub1_endstates1;
%     sub1_x0=sub1_endstates1;
%     sub1_action(:,i)=action1;
%     sub1_cost(i)=step_cost1;
%     sub1_cptime(i)=cp_time1;
sub1_op_cost(i) = op_cost1;
sub1_of_cost(i) = of_cost1;

%subsystem 2

[sub2_endstates2,step_cost2,action2,cp_time2,qu7,qu8,qs5,qs8,op_cost2,of_cost2,suc2]=sub2function
(sub2_x0,sub2_rain(i,:));
    if suc2 == -1
        disp('Optimization Failure at subsystem 2')
    end
sub2_op_cost(i) = op_cost2;
sub2_of_cost(i) = of_cost2;
    %beta5 = 7.1e-04

%     sub2_finalstates(:,i)=sub2_endstates2;
%     sub2_x0=sub2_endstates2;
%     sub2_action(:,i)=action2;
%     sub2_cost(i)=step_cost2;
%     sub2_cptime(i)=cp_time2;

%     disp(['t=',num2str(i)])
%     disp(qu7)

%subsystem 3
%     disp(['t=',num2str(i)])
%     disp(num2str(sub3_disturbance))
    [sub3_endstates3,step_cost3,action3,cp_time3,qu11,qs11,op_cost3,of_cost3,suc3]=
sub3function(sub3_x0,sub3_rain(i,:),sub3_disturbance);
     if suc3 == -1
        disp(['Optimization Failure at subsystem 3 at t= ',num2str(i)])
        break
     end

sub3_op_cost(i) = op_cost3;
sub3_of_cost(i) = of_cost3;
%      action3.*300
%      sub3_disturbance.*300
%      disp(num2str(sub3_endstates3(2)))
%     sub3_finalstates(:,i)=sub3_endstates3;
%     sub3_x0=sub3_endstates3;
%     sub3_action(:,i)=action3;
%     sub3_cost(i)=step_cost3;
%     sub3_cptime(i)=cp_time3;


%subsystem 4
```

```
[sub4_endstates4,step_cost4,action4,cp_time4,qout13,op_cost4,of_cost4,suc4]=sub4function(sub4_x0,
sub4_rain(i,:),sub4_disturbance);
    if suc4 == -1
       disp(['Optimization Failure at subsystem 4 at t= ',num2str(i)])
    end




sub4_op_cost(i) = op_cost4;
sub4_of_cost(i) = of_cost4;

%     sub4_finalstates(:,i)=sub4_endstates4;
%     sub4_x0=sub4_endstates4;
%     sub4_action(:,i)=action4;
%     sub4_cost(i)=step_cost4;
%     sub4_cptime(i)=cp_time4;

%disturbance correction
disp(i)
sub1_endstates1(3)=sub1_endstates1(3)+(qs5(6)-sub1_disturbance(1))*300;
(qs5(6)-sub1_disturbance(1))*300

sub3_endstates3(2)=sub3_endstates3(2)+(qu7(6)-sub3_disturbance(1))*300;
(qu7(6)-sub3_disturbance(1))*300

sub4_endstates4(1)=sub4_endstates4(1)+(qu8(6)+qu11(6)-sub4_disturbance(2,1)-
sub4_disturbance(4,1))*300;
(qu8(6)+qu11(6)-sub4_disturbance(2,1)-sub4_disturbance(4,1))*300

sub4_endstates4(2)=sub4_endstates4(2)+(qs8(6)+qs11(6)-sub4_disturbance(3,1)-
sub4_disturbance(5,1)+qout4(6)-sub4_disturbance(1,1))*300;
(qs8(6)+qs11(6)-sub4_disturbance(3,1)-sub4_disturbance(5,1)+qout4(6)-sub4_disturbance(1,1))*300

%state update
    sub1_finalstates(:,i)=sub1_endstates1;
    sub1_x0=sub1_endstates1;
    sub1_action(:,i)=action1;
    sub1_cost(i)=step_cost1;
    sub1_cptime(i)=cp_time1;

    sub2_finalstates(:,i)=sub2_endstates2;
    sub2_x0=sub2_endstates2;
    sub2_action(:,i)=action2;
    sub2_cost(i)=step_cost2;
    sub2_cptime(i)=cp_time2;

    sub3_finalstates(:,i)=sub3_endstates3;
    sub3_x0=sub3_endstates3;
    sub3_action(:,i)=action3;
    sub3_cost(i)=step_cost3;
    sub3_cptime(i)=cp_time3;

    sub4_finalstates(:,i)=sub4_endstates4;
    sub4_x0=sub4_endstates4;
    sub4_action(:,i)=action4;
    sub4_cost(i)=step_cost4;
    sub4_cptime(i)=cp_time4;


% disturbances update
sub4_disturbance(1,:)=qout4(1:5);
sub4_disturbance(2,:)=qu8(1:5);
sub4_disturbance(3,:)=qs8(1:5);
sub4_disturbance(4,:)=qu11(1:5);
sub4_disturbance(5,:)=qs11(1:5);
sub3_disturbance = qu7(1:5);
sub1_disturbance = qs5(1:5);

end
```

```
sub1_totalcost=sum(sub1_cost);
sub2_totalcost=sum(sub2_cost);
sub3_totalcost=sum(sub3_cost);
sub4_totalcost=sum(sub4_cost);
total_cost = sub1_totalcost+sub2_totalcost+sub3_totalcost+sub4_totalcost;

sub1_total_time = sum(sub1_cptime);
sub2_total_time = sum(sub2_cptime);
sub3_total_time = sum(sub3_cptime);
sub4_total_time = sum(sub4_cptime);

total_time = sub1_total_time+sub2_total_time+sub3_total_time+sub4_total_time;

control = [sub1_action;
           sub4_action(1,:);
           sub2_action;
           sub3_action;
           sub4_action(2,:)];
total_control = control.*300;

step_cost = [sub1_cost(1)+sub2_cost(1)+sub3_cost(1)+sub4_cost(1);
             sub1_cost(2)+sub2_cost(2)+sub3_cost(2)+sub4_cost(2);
             sub1_cost(3)+sub2_cost(3)+sub3_cost(3)+sub4_cost(3);
             sub1_cost(4)+sub2_cost(4)+sub3_cost(4)+sub4_cost(4);
             sub1_cost(5)+sub2_cost(5)+sub3_cost(5)+sub4_cost(5);];

 Final_states = [sub1_finalstates;
                sub2_finalstates;
                sub3_finalstates;
                sub4_finalstates];
%% cost function checking
vm=[16901,43000,35000,26659];
v1m=vm(1);
v2m=vm(2);
v3m=vm(3);
v4m=vm(4);
v5m=vm(1);
v6m=vm(3);
v7m=vm(2);
v8m=vm(1);
v9m=vm(2);
v10m=vm(3);
v11m=vm(4);
v12m=vm(3);
v13m=vm(4);
vmc=[v1m;v2m;v3m;v4m;v5m;v6m;v7m;v8m;v9m;v10m;v11m;v12m;v13m];

fsd1 = Final_states(:,1)-vmc;
sum1 = 0;
for i=1:13
    if fsd1(i)>0
        sum1 = sum1+fsd1(i);
    end
end

fsd2 = Final_states(:,2)-vmc;
sum2 = 0;
for i=1:13
    if fsd2(i)>0
        sum2 = sum2+fsd2(i);
    end
end

fsd3 = Final_states(:,3)-vmc;
sum3 = 0;
for i=1:13
    if fsd3(i)>0
        sum3 = sum3+fsd3(i);
    end
end
```

```
fsd4 = Final_states(:,4)-vmc;
sum4 = 0;
for i=1:13
    if fsd4(i)>0
        sum4 = sum4+fsd4(i);
    end
end


total_op_cost = sub1_op_cost+sub2_op_cost+sub3_op_cost+sub4_op_cost;
total_of_cost = sub1_of_cost+sub2_of_cost+sub3_of_cost+sub4_of_cost;
controlsum =
[sum(control(:,1)),sum(control(:,2)),sum(control(:,3)),sum(control(:,4)),sum(control(:,5))];
```

Decentralized MPC subsystem 1

```
%% subsystem 1 function
function
[end_states,step_cost,action,cp_time,qout4,op_cost,of_cost,suc]=sub1function(initial_states,rain_
profile,disturbance)
%% Large system DMPC subsystem 1

syms dt epsilon v1 v1u v1l v1m beta1 S1 phi1 delta1 deltaS1 deltaS1a1 z1 zS1a1 zS1a1a1 zuS1 qu1
qs1m...
        v2 v2u v2l v2m beta2 S2 phi2 delta2 deltaS2 deltaS2a2 z2 zS2a2 zS2a2a2 zuS2 qu2 qs2m...
        v3 v3m qu3 beta3...
        v4 v4u v4l v4m beta4 S4 phi4 delta4 deltaS4 deltaS4a4 z4 zS4a4 zS4a4a4 zuS4 qu4 qs4m...
        qu1m qu2m qu3m ...
        qs1m qs2m ...
        m1 m2  m4 ...
        M1 M2  M4 ...
        v01 v02 v03 v04 ...
        p01 p02 p03 p04 ...
        p11 p12 p13 p14 ...
        p21 p22 p23 p24 ...
        p31 p32 p33 p34 ...
        p41 p42 p43 p44 ...
        z01 z02 z03 z04 ...
        z11 z12 z13 z14 ...
        z21 z22 z23 z24 ...
        z31 z32 z33 z34 ...
        z0S1a1 z0S1a1a1 z0S2a2 z0S2a2a2 z0S4a4 z0S4a4a4 ...
        z0uS1 z0uS2 z0uS4 ...
        z1S1a1 z1S1a1a1 z1S2a2 z1S2a2a2 ...
        z1uS1 z1uS2 z1uS4 ...
        z2S1a1 z2S1a1a1 z2S2a2 z2S2a2a2 z2S4a4 z2S4a4a4 ...
        z2uS1 z2uS2 z2uS4 ...
        z3S1a1 z3S1a1a1 z3S2a2 z3S2a2a2 ...
        z3uS1 z3uS2 z3uS4 ...
        delta01 delta02 delta03 delta04 ...
        delta11 delta12 delta13 delta14 ...
        delta21 delta22 delta23 delta24 ...
        delta31 delta32 delta33 delta34 ...
        delta0S1 delta0S1a1 delta0S2 delta0S2a2 delta0S4 delta0S4a4 ...
        delta1S1 delta1S1a1 delta1S2 delta1S2a2 delta1S4 delta1S4a4 ...
        delta2S1 delta2S1a1 delta2S2 delta2S2a2 delta2S4 delta2S4a4 ...
        delta3S1 delta3S1a1 delta3S2 delta3S2a2 delta3S4 delta3S4a4 ...
        u01 u02 u03 ...
        u11 u12 u13 ...
        u21 u22 u23 ...
        u31 u32 u33 ...
        qs5 ... % external disturbance to real tank


qs5 = disturbance; % disturbance
```

```
A = sym(zeros(4,4));   %tank
B1 = sym(zeros(4,3));  %control
B2 = sym(zeros(4,8));   %delta
B3 = sym(zeros(4,10));   %z
B4= sym(zeros(4,4));    % disturbance


A(1,1)=1-dt*beta1;
B2(1,1)=v1m-dt*beta1*v1m;
B3(1,1)=dt*beta1-1;
B4(1,1)=dt*phi1*S1;

A(2,2)=1-dt*beta2;
B1(2,1)=dt;
B2(2,2)=(1-beta2*dt)*v2m;
B3(2,2)=beta2*dt-1;
B4(2,2)=dt*phi2*S2;

A(3,3)=1;
B1(3,2)= dt;
B1(3,3)=-dt;
B4(3,3)= dt; %consider qs5 as disturbances


A(4,1)=dt*beta1;
A(4,2)=dt*beta2;
A(4,4)=1-beta4*dt;
B1(4,3)=dt;
B1(4,1)=-dt;
B1(4,2)=-dt;
B2(4,1)=dt*beta1*v1m;
B2(4,2)=dt*beta2*v2m;
B2(4,4)=(1-dt*beta4)*v4m;
B2(4,5)=dt*qs1m;   % deltaS1
B2(4,7)=dt*qs2m;   % deltaS2
B2(4,6)=-dt*beta1*v1m; %deltaS11
B2(4,8)=-dt*beta2*v2m; %deltaS22
B3(4,1)=-dt*beta1;
B3(4,2)=-dt*beta2;
B3(4,4)=dt*beta4-1;
B3(4,5)=-dt*beta1;   %zs11
B3(4,7)=-dt*beta2;   %zs22
B3(4,6)=dt*beta1;    %zs111
B3(4,8)=dt*beta2;    %zs222
B3(4,9)=dt;          %zus1
B3(4,10)=dt;          %zus2
B4(4,4)=dt*S4*phi4;

d0=[p01;p02;p03;p04];
d1=[p11;p12;p13;p14];
d2=[p21;p22;p23;p24];
d3=[p31;p32;p33;p34];
d4=[p41;p42;p43;p44];

z0=[z01;z02;z03;z04;z0S1a1;z0S1a1a1;z0S2a2;z0S2a2a2;z0uS1;z0uS2];
z1=[z11;z12;z13;z14;z1S1a1;z1S1a1a1;z1S2a2;z1S2a2a2;z1uS1;z1uS2];
z2=[z21;z22;z23;z24;z2S1a1;z2S1a1a1;z2S2a2;z2S2a2a2;z2uS1;z2uS2];
z3=[z31;z32;z33;z34;z3S1a1;z3S1a1a1;z3S2a2;z3S2a2a2;z3uS1;z3uS2];


delta0 = [delta01;delta02;delta03;delta04;delta0S1;delta0S1a1;delta0S2;delta0S2a2];
delta1 = [delta11;delta12;delta13;delta14;delta1S1;delta1S1a1;delta1S2;delta1S2a2];
delta2 = [delta21;delta22;delta23;delta24;delta2S1;delta2S1a1;delta2S2;delta2S2a2];
delta3 = [delta31;delta32;delta33;delta34;delta3S1;delta3S1a1;delta3S2;delta3S2a2];

u0 = [u01;u02;u03];
u1 = [u11;u12;u13];
u2 = [u21;u22;u23];
u3 = [u31;u32;u33];
```

```
v0=[v01;v02;v03;v04];

v1=A*v0+B1*u0+B2*delta0+B3*z0+B4*d0;
v2=A*v1+B1*u1+B2*delta1+B3*z1+B4*d1;
v3=A*v2+B1*u2+B2*delta2+B3*z2+B4*d2;
v4=A*v3+B1*u3+B2*delta3+B3*z3+B4*d3;

%% constraints E
E1=sym(zeros(56,3)); %qu
E2=sym(zeros(56,8)); %delta
E3=sym(zeros(56,10)); %z
E4=sym(zeros(56,4)); %v
E5=sym(zeros(56,1));  %constants


E2(1,1)=-beta1*v1m;
E3(1,1)=beta1;
E1(1,1)=-1;
E4(1,1)=beta1;     %qu1

E2(2,2)=-beta2*v2m;
E3(2,2)=beta2;
E1(2,2)=-1;
E4(2,2)=beta2;   %qu2

E1(3,3)=dt;
E1(3,2)=-dt;
E4(3,3)=-1;
E5(3,1)=v3m-dt*qs5(1); %qu2

E1(4,3)=-1;
E4(4,3)=beta3;   %qu3

E2(5,1)=-v1u;
E3(5,1)=1;

E3(6,1)=-1;

E3(7,1)=1;
E4(7,1)=1;

E2(8,1)=v1u;
E3(8,1)=-1;
E4(8,1)=-1;
E5(8,1)=v1u;   %z1

E2(9,2)=-v2u;
E3(9,2)=1;


E3(10,2)=-1;

E3(11,2)=1;
E4(11,2)=1;

E2(12,2)=v2u;
E3(12,2)=-1;
E4(12,2)=-1;
E5(12,1)=v2u;   %z2

E2(13,4)=-v4u;
E3(13,4)=1;

E3(14,4)=-1;

E3(15,4)=1;
E4(15,4)=1;

E2(16,4)=v4u;
E3(16,4)=-1;
E4(16,4)=-1;
```

```
E5(16,1)=v4u; %z4

E2(17,5)=-v1u;
E3(17,5)=1;

E3(18,5)=-1;

E3(19,5)=1;
E4(19,1)=1;

E2(20,5)=v1u;
E3(20,5)=-1;
E4(20,1)=-1;
E5(20,1)=v1u;  %zs11

E2(21,6)=-v1u;
E3(21,6)=1;

E3(22,6)=-1;

E3(23,6)=1;
E4(23,1)=1;

E2(24,6)=v1u;
E3(24,6)=-1;
E4(24,1)=-1;
E5(24,1)=v1u; %zs111

E2(25,7)=-v2u;
E3(25,7)=1;

E3(26,7)=-1;

E3(27,7)=1;
E4(27,2)=1;

E2(28,7)=v2u;
E3(28,7)=-1;
E4(28,2)=-1;
E5(28,1)=v2u; %zs22

E2(29,8)=-v2u;
E3(29,8)=1;

E3(30,8)=-1;

E3(31,8)=1;
E4(31,2)=1;

E2(32,8)=v2u;
E3(32,8)=-1;
E4(32,2)=-1;
E5(32,1)=v2u;  %zs222

E2(33,5)=-qu1m;
E3(33,9)=1;

E3(34,9)=-1;

E3(35,9)=1;
E1(35,1)=1;

E2(36,5)=qu1m;
E3(36,9)=-1;
E1(36,1)=-1;
E5(36,1)=qu1m;  %zuS1

E2(37,7)=-qu2m;
E3(37,10)=1;

E3(38,10)=-1;
```

```
E3(39,10)=1;
E1(39,2)=1;

E2(40,7)=qu2m;
E3(40,10)=-1;
E1(40,2)=-1;
E5(40,1)=qu2m;   %zuS2

E2(41,1)=-m1;
E4(41,1)=1;
E5(41,1)=-v1m-m1;

E2(42,1)=-(M1+epsilon);
E4(42,1)=-1;
E5(42,1)=v1m-epsilon;  %delta1

E2(43,2)=-m2;
E4(43,2)=1;
E5(43,1)=-v2m-m2;

E2(44,2)=-(M2+epsilon);
E4(44,2)=-1;
E5(44,1)=v2m-epsilon; %delta2

E2(45,4)=-m4;
E4(45,4)=1;
E5(45,1)=-v4m-m4;

E2(46,4)=-(M4+epsilon);
E4(46,4)=-1;
E5(46,1)=v4m-epsilon; %delta4

E2(47,5)=qs1m+qu1m;
E2(47,6)=-beta1*v1m;
E3(47,5)=-beta1;
E3(47,6)=beta1;
E1(47,1)=-1;
E5(47,1)=qu1m;

E2(48,1)=beta1*v1m;
E2(48,5)=qs1m-epsilon;
E2(48,6)=-beta1*v1m;
E3(48,1)=-beta1;
E3(48,5)=-beta1;
E3(48,6)=beta1;
E1(48,1)=1;
E4(48,1)=-beta1;
E5(48,1)=qs1m-epsilon;    %deltaS1

E2(49,1)=-1;
E2(49,6)=1;

E2(50,5)=-1;
E2(50,6)=1;

E2(51,1)=1;
E2(51,5)=1;
E2(51,6)=-1;
E5(51,1)=1;          %deltaS1a1

E2(52,7)=qs2m+qu2m;
E2(52,8)=-beta2*v2m;
E3(52,7)=-beta2;
E3(52,8)=beta2;
E1(52,2)=-1;
E5(52,1)=qu2m;

E2(53,2)=beta2*v2m;
E2(53,7)=qs2m-epsilon;
E2(53,8)=-beta2*v2m;
```

```
E3(53,2)=-beta2;
E3(53,7)=-beta2;
E3(53,8)=beta2;
E1(53,2)=1;
E4(53,2)=-beta2;
E5(53,1)=qs2m-epsilon;     %deltaS2

E2(54,2)=-1;
E2(54,8)=1;

E2(55,7)=-1;
E2(55,8)=1;

E2(56,2)=1;
E2(56,7)=1;
E2(56,8)=-1;        %deltaS2a2
E5(56,1)=1;
%% constraints E1
E11 = E1;
E21 = E2;
E31 = E3;
E41 = E4;
E51 = E5;
E51(3,1)=v3m-dt*qs5(2); %qu2
%% constraints E2
E12 = E1;
E22 = E2;
E32 = E3;
E42 = E4;
E52 = E5;
E52(3,1)=v3m-dt*qs5(3); %qu2
%% constraints E3
E13 = E1;
E23 = E2;
E33 = E3;
E43 = E4;
E53 = E5;
E53(3,1)=v3m-dt*qs5(4); %qu2
%% constraints E4
E14 = E1;
E24 = E2;
E34 = E3;
E44 = E4;
E54 = E5;
E54(3,1)=v3m-dt*qs5(5); %qu2


%% constraints integration
F11=sym(zeros(56,21*5));    %initialize
F21=sym(zeros(56,1));
F31=sym(zeros(56,4));
F11=[-E1,E2,E3,zeros(56,21*4)];
F12=[-E41*B1,-E41*B2,-E41*B3,-E11,E21,E31,zeros(56,21*3)];
F13=[-E42*A*B1,-E42*A*B2,-E42*A*B3,-E42*B1,-E42*B2,-E42*B3,-E12,E22,E32,zeros(56,21*2)];
F14=[-E43*A^2*B1,-E43*A^2*B2,-E43*A^2*B3,-E43*A*B1,-E43*A*B2,-E43*A*B3,-E43*B1,-E43*B2,-E43*B3,-
E13,E23,E33,zeros(56,21)];
F15=[-E44*A^3*B1,-E44*A^3*B2,-E44*A^3*B3,-E44*A^2*B1,-E44*A^2*B2,-E44*A^2*B3,-E44*A*B1,-
E44*A*B2,-E44*A*B3,-E44*B1,-E44*B2,-E44*B3,-E14,E24,E34];

F21=E5;
F22=E41*B4*d0+E51;
F23=E42*A*B4*d0+E42*B4*d1+E52;
F24=E43*A*A*B4*d0+E43*A*B4*d1+E43*B4*d2+E53;
F25=E44*A*A*A*B4*d0+E44*A*A*B4*d1+E44*A*B4*d2+E44*B4*d3+E54;

F31=E4*v0;
F32=E41*A*v0;
F33=E42*A*A*v0;
F34=E43*A*A*A*v0;
F35=E44*A*A*A*A*v0;
```

```
F1=[F11;F12;F13;F14;F15];
F2=[F21;F22;F23;F24;F25];
F3=[F31;F32;F33;F34;F35];

%% cost function 1
% costfactor1=1/(rain_profile(1)/300);
costfactor1 =1;
L2=sym(zeros(1,21*5));

L2(1,1)=costfactor1*0.5*dt ;
L2(1,2)=costfactor1*0.5*dt ;
L2(1,3)=costfactor1*0.5*dt ;  %operation cost from manipulation flows

L2(1,12)=1;
L2(1,13)=1;
L2(1,15)=1;

L2(1,4)=-v1m;
L2(1,5)=-v2m;
L2(1,7)=-v4m;    %tank over flow

L2(1,8)=-dt*qs1m;   % deltaS1
L2(1,9)=dt*beta1*v1m;  %deltaS11
L2(1,16)=dt*beta1;     %zs1a1
L2(1,17)=-dt*beta1;     %zs1a1a1
L2(1,20)=-dt;  %zus1   %qs1 overflow

L2(1,10)=-dt*qs2m;       %deltaS2
L2(1,11)=dt*beta2*v2m;   %deltaS22
L2(1,18)=dt*beta2;       %zs2a2
L2(1,19)=-dt*beta2;      %zs2a2a2
L2(1,21)=-dt;  %zus2   %qs2 overflow
%% cost function 2
% costfactor2=1/(rain_profile(2)/300);
costfactor2=1;
L2(1,1+21)=costfactor2*0.5*dt ;
L2(1,2+21)=costfactor2*0.5*dt ;
L2(1,3+21)=costfactor2*0.5*dt ;  %operation cost from manipulation flows

L2(1,12+21)=1;
L2(1,13+21)=1;
L2(1,15+21)=1;

L2(1,4+21)=-v1m;
L2(1,5+21)=-v2m;
L2(1,7+21)=-v4m;    %tank over flow

L2(1,8+21)=-dt*qs1m;   % deltaS1
L2(1,9+21)=dt*beta1*v1m;  %deltaS11
L2(1,16+21)=dt*beta1;     %zs1a1
L2(1,17+21)=-dt*beta1;     %zs1a1a1
L2(1,20+21)=-dt;  %zus1   %qs1 overflow

L2(1,10+21)=-dt*qs2m;       %deltaS2
L2(1,11+21)=dt*beta2*v2m;   %deltaS22
L2(1,18+21)=dt*beta2;       %zs2a2
L2(1,19+21)=-dt*beta2;      %zs2a2a2
L2(1,21+21)=-dt;  %zus2   %qs2 overflow
%% cost function 3
% costfactor3=1/(rain_profile(3)/300);
costfactor3=1;
L2(1,1+21*2)=costfactor3*0.5*dt ;
L2(1,2+21*2)=costfactor3*0.5*dt ;
L2(1,3+21*2)=costfactor3*0.5*dt ;  %operation cost from manipulation flows

L2(1,12+21*2)=1;
L2(1,13+21*2)=1;
L2(1,15+21*2)=1;

L2(1,4+21*2)=-v1m;
L2(1,5+21*2)=-v2m;
```

```
L2(1,7+21*2)=-v4m;    %tank over flow

L2(1,8+21*2)=-dt*qs1m;   % deltaS1
L2(1,9+21*2)=dt*beta1*v1m;  %deltaS11
L2(1,16+21*2)=dt*beta1;     %zs1a1
L2(1,17+21*2)=-dt*beta1;     %zs1a1a1
L2(1,20+21*2)=-dt;  %zus1   %qs1 overflow

L2(1,10+21*2)=-dt*qs2m;       %deltaS2
L2(1,11+21*2)=dt*beta2*v2m;   %deltaS22
L2(1,18+21*2)=dt*beta2;       %zs2a2
L2(1,19+21*2)=-dt*beta2;      %zs2a2a2
L2(1,21+21*2)=-dt;  %zus2   %qs2 overflow
%% cost function 4
% costfactor4=1/(rain_profile(4)/300);
costfactor4=1;
L2(1,1+21*3)=costfactor4*0.5*dt ;
L2(1,2+21*3)=costfactor4*0.5*dt ;
L2(1,3+21*3)=costfactor4*0.5*dt ;  %operation cost from manipulation flows

L2(1,12+21*3)=1;
L2(1,13+21*3)=1;
L2(1,15+21*3)=1;

L2(1,4+21*3)=-v1m;
L2(1,5+21*3)=-v2m;
L2(1,7+21*3)=-v4m;    %tank over flow

L2(1,8+21*3)=-dt*qs1m;   % deltaS1
L2(1,9+21*3)=dt*beta1*v1m;  %deltaS11
L2(1,16+21*3)=dt*beta1;     %zs1a1
L2(1,17+21*3)=-dt*beta1;     %zs1a1a1
L2(1,20+21*3)=-dt;  %zus1   %qs1 overflow

L2(1,10+21*3)=-dt*qs2m;       %deltaS2
L2(1,11+21*3)=dt*beta2*v2m;   %deltaS22
L2(1,18+21*3)=dt*beta2;       %zs2a2
L2(1,19+21*3)=-dt*beta2;      %zs2a2a2
L2(1,21+21*3)=-dt;  %zus2   %qs2 overflow
%% cost function 5
% costfactor5=1/(rain_profile(5)/250);
costfactor5=1;
L2(1,1+21*4)=costfactor5*0.5*dt ;
L2(1,2+21*4)=costfactor5*0.5*dt ;
L2(1,3+21*4)=costfactor5*0.5*dt ;  %operation cost from manipulation flows

L2(1,12+21*4)=1;
L2(1,13+21*4)=1;
L2(1,15+21*4)=1;

L2(1,4+21*4)=-v1m;
L2(1,5+21*4)=-v2m;
L2(1,7+21*4)=-v4m;    %tank over flow

L2(1,8+21*4)=-dt*qs1m;   % deltaS1
L2(1,9+21*4)=dt*beta1*v1m;  %deltaS11
L2(1,16+21*4)=dt*beta1;     %zs1a1
L2(1,17+21*4)=-dt*beta1;     %zs1a1a1
L2(1,20+21*4)=-dt;  %zus1   %qs1 overflow

L2(1,10+21*4)=-dt*qs2m;       %deltaS2
L2(1,11+21*4)=dt*beta2*v2m;   %deltaS22
L2(1,18+21*4)=dt*beta2;       %zs2a2
L2(1,19+21*4)=-dt*beta2;      %zs2a2a2
L2(1,21+21*4)=-dt;  %zus2   %qs2 overflow
%% constants specification

epsilon =1e-13;
S=[323576,164869,5076,754131];%surface area
phi=[1.03,10.4,0,0.48];%absorption coefficient
beta=[7.1e-4,5.8e-4,2e-3,1e-3];%volumetric flow coefficient
```

```
vm=[16901,43000,35000,26659];%maximum volume
dt=300;%sample time


% x0=[0,0,0,0];
x0 = initial_states;



% rain = [100,130,140,150,90];
rain = rain_profile;


p = rain.*1.2./1000./3600;% precipitation in unit of m/s

v1m=vm(1);
v3m=vm(3);
v2m=vm(2);
v4m=vm(4);


v01=x0(1);
v02=x0(2);
v03=x0(3);
v04=x0(4);


S1=S(1);
S2=S(2);
S4=S(4);


phi1=phi(1);
phi2=phi(2);
phi4=phi(4);


beta1=beta(1);
beta2=beta(2);
beta3=beta(3);
beta4=beta(4);

p01=p(1);
p02=p(1);
p03=qs5(1);%real
p04=p(1);

p11=p(2);
p12=p(2);
p13=qs5(2);%real
p14=p(2);

p21=p(3);
p22=p(3);
p23=qs5(3);%real
p24=p(3);

p31=p(4);
p32=p(4);
p33=qs5(4);
p34=p(4);



p41=p(5);
p42=p(5);
p43=qs5(5);
p44=p(5);
```

```
% qs1m=9.14;
% qs2m=3.4;

qs1m=1000;
qs2m=1000;



v1u=v1m*2+S1*dt*max(p)*phi1;
v2u=v2m*2+S2*dt*max(p)*phi2;
v4u=v4m*2+S4*dt*max(p)*phi4;

v3u=35000;


vu=[v1u,v2u,v3u,v4u];


qu1m=9.1;
qu2m=25;
qu3m=12;

M1=v1u-v1m;
m1=-v1m;

M2=v2u-v2m;
m2=-v2m;

M4=v4u-v4m;
m4=-v4m;

lb = zeros(21*5,1);                  %Bounds on x (lb <= x)
uub = [qu1m;qu2m;qu3m];          %u upper bound

dub = zeros(8,1);                %delta upperbound
for i = 1:8
   dub(i,1)=1;
end
zub = zeros(10,1);            %z upper bound
for i=1:10
    zub(i,1)=inf;
end

ub = [uub;dub;zub;uub;dub;zub;uub;dub;zub;uub;dub;zub;uub;dub;zub];
f=eval(L2');


a=eval(F1);
b=eval(F2+F3);
%%
xtype =
'CCCBBBBBBBBCCCCCCCCCCCCCBBBBBBBBCCCCCCCCCCCCCBBBBBBBBCCCCCCCCCCCCCBBBBBBBBCCCCCCCCCCCCCBBBBBBBBC
CCCCCCCCC';

Opt = opti('f',f,'ineq',a,b,'bounds',lb,ub,'xtype',xtype);
[x,fval,exitflag,info] = solve(Opt);

u01=x(1);
u02=x(2);
u03=x(3);
delta01=x(4);
delta02=x(5);
delta03=x(6);
delta04=x(7);
delta0S1=x(8);
delta0S1a1=x(9);
delta0S2=x(10);
delta0S2a2=x(11);
z01=x(12);
z02=x(13);
z03=x(14);
```

```
z04=x(15);
z0S1a1=x(16);
z0S1a1a1=x(17);
z0S2a2=x(18);
z0S2a2a2=x(19);
z0uS1=x(20);
z0uS2=x(21);

u11=x(1+21);
u12=x(2+21);
u13=x(3+21);
delta11=x(4+21);
delta12=x(5+21);
delta13=x(6+21);
delta14=x(7+21);
delta1S1=x(8+21);
delta1S1a1=x(9+21);
delta1S2=x(10+21);
delta1S2a2=x(11+21);
z11=x(12+21);
z12=x(13+21);
z13=x(14+21);
z14=x(15+21);
z1S1a1=x(16+21);
z1S1a1a1=x(17+21);
z1S2a2=x(18+21);
z1S2a2a2=x(19+21);
z1uS1=x(20+21);
z1uS2=x(21+21);

u21=x(1+21*2);
u22=x(2+21*2);
u23=x(3+21*2);
delta21=x(4+21*2);
delta22=x(5+21*2);
delta23=x(6+21*2);
delta24=x(7+21*2);
delta2S1=x(8+21*2);
delta2S1a1=x(9+21*2);
delta2S2=x(10+21*2);
delta2S2a2=x(11+21*2);
z21=x(12+21*2);
z22=x(13+21*2);
z23=x(14+21*2);
z24=x(15+21*2);
z2S1a1=x(16+21*2);
z2S1a1a1=x(17+21*2);
z2S2a2=x(18+21*2);
z2S2a2a2=x(19+21*2);
z2uS1=x(20+21*2);
z2uS2=x(21+21*2);

u31=x(1+21*3);
u32=x(2+21*3);
u33=x(3+21*3);
delta31=x(4+21*3);
delta32=x(5+21*3);
delta33=x(6+21*3);
delta34=x(7+21*3);
delta3S1=x(8+21*3);
delta3S1a1=x(9+21*3);
delta3S2=x(10+21*3);
delta3S2a2=x(11+21*3);
z31=x(12+21*3);
z32=x(13+21*3);
z33=x(14+21*3);
z34=x(15+21*3);
z3S1a1=x(16+21*3);
z3S1a1a1=x(17+21*3);
z3S2a2=x(18+21*3);
z3S2a2a2=x(19+21*3);
```

```
z3uS1=x(20+21*3);
z3uS2=x(21+21*3);

u41=x(1+21*4);
u42=x(2+21*4);
u43=x(3+21*4);
delta41=x(4+21*4);
delta42=x(5+21*4);
delta43=x(6+21*4);
delta44=x(7+21*4);
delta4S1=x(8+21*4);
delta4S1a1=x(9+21*4);
delta4S2=x(10+21*4);
delta4S2a2=x(11+21*4);
z41=x(12+21*4);
z42=x(13+21*4);
z43=x(14+21*4);
z44=x(15+21*4);
z4S1a1=x(16+21*4);
z4S1a1a1=x(17+21*4);
z4S2a2=x(18+21*4);
z4S2a2a2=x(19+21*4);
z4uS1=x(20+21*4);
z4uS2=x(21+21*4);

end_states= eval(v1);
step_cost=eval(L2(1:21)*x(1:21));
action= [x(1);x(2);x(3)];
cp_time = info.Time;
% qout4=[beta4*v4m*x(7)+beta4*x0(4)-beta4*x(15),
%        beta4*v4m*x(7+21)+eval(beta4*v1(4))-beta4*x(15+21),
%        beta4*v4m*x(7+21*2)+eval(beta4*v2(4))-beta4*x(15+21*2),
%        beta4*v4m*x(7+21*3)+eval(beta4*v3(4))-beta4*x(15+21*3),
%        beta4*v4m*x(7+21*4)+eval(beta4*v4(4))-beta4*x(15+21*4)];

qout4=[
       beta4*v4m*x(7+21)+eval(beta4*v1(4))-beta4*x(15+21),
       beta4*v4m*x(7+21*2)+eval(beta4*v2(4))-beta4*x(15+21*2),
       beta4*v4m*x(7+21*3)+eval(beta4*v3(4))-beta4*x(15+21*3),
       beta4*v4m*x(7+21*4)+eval(beta4*v4(4))-beta4*x(15+21*4),
       beta4*v4m*x(7+21*4)+eval(beta4*v4(4))-beta4*x(15+21*4),
       beta4*v4m*x(7)+beta4*x0(4)-beta4*x(15)];  %Last entry used for disturbance correction

op_cost = eval(L2(1:3)*x(1:3));   %control cost
of_cost = eval(L2(4:21)*x(4:21));  %overflow cost

suc = exitflag;
end
```

Decentralized MPC subsystem 2

```
%% susbsytem 2 function
function
[end_states,step_cost,action,cp_time,qu7,qu8,qs5,qs8,op_cost,of_cost,suc]=sub2function(initial_st
ates,rain_profile)
% system dynamics
syms    dt epsilon ...
        v5 v5u v5l v5m beta5 S5 phi5 delta5 deltaS5 deltaS5a5 z5 zS5a5 zS5a5a5 zuS5 qu5 qs5m...
        v6 v6m beta6 qu6 ...
        v7 v7u v7l v7m beta7 S7 phi7 delta7 deltaS7 deltaS7a7 z7 zS7a7 zS7a7a7 zuS7 qu7 qs7m...
        v8 v8u v8l v8m beta8 S8 phi8 delta8 deltaS8 deltaS8a8 z8 zS8a8 zS8a8a8 zuS8 qu8 qs8m...
        qu5m qu7m qu8m ...
        qs5m qs7m qs8m...
        m5 m6 m7 m8 ...
        M5 M6 M7 M8 ...
        v05 v06 v07 v08...
        p05 p06 p07 p08...
        p15 p16 p17 p18...
        p25 p26 p27 p28...
        p35 p36 p37 p38...
        p45 p46 p47 p48...
        z05 z06 z07 z08...
        z15 z16 z17 z18...
        z25 z26 z27 z28...
        z35 z36 z37 z38...
        z0S5a5 z0S5a5a5 z0S7a7 z0S7a7a7 z0S8a8 z0S8a8a8 z0uS5 z0uS7 z0uS8...
        z1S5a5 z1S5a5a5 z1S7a7 z1S7a7a7 z1S8a8 z1S8a8a8 z1uS5 z1uS7 z1uS8...
        z2S5a5 z2S5a5a5 z2S7a7 z2S7a7a7 z2S8a8 z2S8a8a8 z2uS5 z2uS7 z2uS8...
        z3S5a5 z3S5a5a5 z3S7a7 z3S7a7a7 z3S8a8 z3S8a8a8 z3uS5 z3uS7 z3uS8...
        delta05 delta06 delta07 delta08 delta0S5 delta0S5a5 delta0S7 delta0S7a7 delta0S8
delta0S8a8...
        delta15 delta16 delta17 delta18 delta1S5 delta1S5a5 delta1S7 delta1S7a7 delta1S8
delta1S8a8...
        delta25 delta26 delta27 delta28 delta2S5 delta2S5a5 delta2S7 delta2S7a7 delta2S8
delta2S8a8...
        delta35 delta36 delta37 delta38 delta3S5 delta3S5a5 delta3S7 delta3S7a7 delta3S8
delta3S8a8...
        u05 u06 u07 u08...
        u15 u16 u17 u18...
        u25 u26 u27 u28...
        u35 u36 u37 u38...




A = sym(zeros(4,4));   %tank
B1 = sym(zeros(4,4));  %control
B2 = sym(zeros(4,10));   %delta
B3 = sym(zeros(4,13));   %z
B4= sym(zeros(4,4));    % disturbance

d0=[p05;p06;p07;p08];
d1=[p15;p16;p17;p18];
d2=[p25;p26;p27;p28];
d3=[p35;p36;p37;p38];
d4=[p45;p46;p47;p48];

z0=[z05;z06;z07;z08;z0S5a5;z0S5a5a5;z0S7a7;z0S7a7a7;z0S8a8;z0S8a8a8;z0uS5;z0uS7;z0uS8];
z1=[z15;z16;z17;z18;z1S5a5;z1S5a5a5;z1S7a7;z1S7a7a7;z1S8a8;z1S8a8a8;z1uS5;z1uS7;z1uS8];
z2=[z25;z26;z27;z28;z2S5a5;z2S5a5a5;z2S7a7;z2S7a7a7;z2S8a8;z2S8a8a8;z2uS5;z2uS7;z2uS8];
z3=[z35;z36;z37;z38;z3S5a5;z3S5a5a5;z3S7a7;z3S7a7a7;z3S8a8;z3S8a8a8;z3uS5;z3uS7;z3uS8];

delta0 =
[delta05;delta06;delta07;delta08;delta0S5;delta0S5a5;delta0S7;delta0S7a7;delta0S8;delta0S8a8];
delta1 =
[delta15;delta16;delta17;delta18;delta1S5;delta1S5a5;delta1S7;delta1S7a7;delta1S8;delta1S8a8];
```

```
delta2 =
[delta25;delta26;delta27;delta28;delta2S5;delta2S5a5;delta2S7;delta2S7a7;delta2S8;delta2S8a8];
delta3 =
[delta35;delta36;delta37;delta38;delta3S5;delta3S5a5;delta3S7;delta3S7a7;delta3S8;delta3S8a8];

u0 = [u05;u06;u07;u08];
u1 = [u15;u16;u17;u18];
u2 = [u25;u26;u27;u28];
u3 = [u35;u36;u37;u38];


v0=[v05;v06;v07;v08];

A(1,1)=1-beta5*dt;      %+4
B2(1,1)=1-dt*beta5;
B3(1,1)=dt*beta5-1;
B4(1,1)=dt*S5*phi5;

A(2,2)=1;
B1(2,1)=dt;
B1(2,2)=-dt;

A(3,3)=1-dt*beta7;
B2(3,3)=(1-dt*beta7)*v7m;
B3(3,3)=dt*beta7-1;
B4(3,3)=dt*S7*phi7;

A(4,4)=1-dt*beta8;
A(4,3)=dt*beta7;
B1(4,2)=dt;
B1(4,3)=-dt;
B2(4,7)=dt*qs7m;
B2(4,4)=(1-dt*beta8)*v8m;
B2(4,3)=dt*beta7*v7m;
B2(4,8)=-dt*beta7*v7m;
B3(4,3)=-dt*beta7;
B3(4,4)=dt*beta8-1;
B3(4,7)=-dt*beta7;
B3(4,8)=dt*beta7;
B3(4,12)=dt;
B4(4,4)=dt*S8*phi8;

v1=A*v0+B1*u0+B2*delta0+B3*z0+B4*d0;
v2=A*v1+B1*u1+B2*delta1+B3*z1+B4*d1;
v3=A*v2+B1*u2+B2*delta2+B3*z2+B4*d2;
v4=A*v3+B1*u3+B2*delta3+B3*z3+B4*d3;
%% constraints
E1=sym(zeros(74,4)); %qu
E2=sym(zeros(74,10)); %delta
E3=sym(zeros(74,13)); %z
E4=sym(zeros(74,4)); %v
E5=sym(zeros(74,1));  %constants

E2(1,1)=-beta5*v5m;
E3(1,1)=beta5;
E1(1,1)=-1;
E4(1,1)=beta5;  %qu5

E1(2,2)=dt;
E1(2,1)=-dt;
E4(2,2)=-1;
E5(2,1)=v6m;  %qu5

E1(3,2)=-1;
E4(3,2)=beta6; %qu6

E2(4,3)=-beta7*v7m;
E3(4,3)=beta7;
E1(4,3)=-1;
E4(4,3)=beta7; %qu7
```

```
% E1(5,10)=dt;
% E1(5,7)=-dt;
% E1(5,9)=-dt;
% E4(5,10)=-1;
% E5(5,1)=v10m; %qu7 this is ignored as it will be considered in subsystem3

E2(5,4)=-beta8*v8m;
E3(5,4)=beta8;
E1(5,4)=-1;
E4(5,4)=beta8;   %qu8

E2(6,1)=-v5u;
E3(6,1)=1;

E3(7,1)=-1;

E3(8,1)=1;
E4(8,1)=1;

E2(9,1)=v5u;
E3(9,1)=-1;
E4(9,1)=-1;
E5(9,1)=v5u; %z5

E2(10,3)=-v7u;
E3(10,3)=1;

E3(11,3)=-1;

E3(12,3)=1;
E4(12,3)=1;

E2(13,3)=v7u;
E3(13,3)=-1;
E4(13,3)=-1;
E5(13,1)=v7u; %z7

E2(14,4)=-v8u;
E3(14,4)=1;

E3(15,4)=-1;

E3(16,4)=1;
E4(16,4)=1;

E2(17,4)=v8u;
E3(17,4)=-1;
E4(17,4)=-1;
E5(17,1)=v8u; %z8

E2(18,5)=-v5u;
E3(18,5)=1;

E3(19,5)=-1;

E3(20,5)=1;
E4(20,1)=1;

E2(21,5)=v5u;
E3(21,5)=-1;
E4(21,1)=-1;
E5(21,1)=v5u;  %zs55

E2(22,6)=-v5u;
E3(22,6)=1;

E3(23,6)=-1;

E3(24,6)=1;
E4(24,1)=1;
```

```
E2(25,6)=v5u;
E3(25,6)=-1;
E4(25,1)=-1;
E5(25,1)=v5u;   %zs555

E2(26,7)=-v7u;
E3(26,7)=1;

E3(27,7)=-1;

E3(28,7)=1;
E4(28,3)=1;

E2(29,7)=v7u;
E3(29,7)=-1;
E4(29,3)=-1;
E5(29,1)=v7u;  %zs77

E2(30,8)=-v7u;
E3(30,8)=1;

E3(31,8)=-1;

E3(32,8)=1;
E4(32,3)=1;

E2(33,8)=v7u;
E3(33,8)=-1;
E4(33,3)=-1;
E5(33,1)=v7u;  %zs777

E2(34,9)=-v8u;
E3(34,9)=1;

E3(35,9)=-1;

E3(36,9)=1;
E4(36,4)=1;

E2(37,9)=v8u;
E3(37,9)=-1;
E4(37,4)=-1;
E5(37,1)=v8u;  %zs88

E2(38,10)=-v8u;
E3(38,10)=1;

E3(39,10)=-1;

E3(40,10)=1;
E4(40,4)=1;

E2(41,10)=v8u;
E3(41,10)=-1;
E4(41,4)=-1;
E5(41,1)=v8u;  %zs888

E2(42,5)=-qu5m;
E3(42,11)=1;

E3(43,11)=-1;

E3(44,11)=1;
E1(44,1)=1;

E2(45,5)=qu5m;
E3(45,11)=-1;
E1(45,1)=-1;
E5(45,1)=qu5m;  %zuS5

E2(46,7)=-qu7m;
```

```
E3(46,12)=1;

E3(47,12)=-1;

E3(48,12)=1;
E1(48,3)=1;

E2(49,7)=qu7m;
E3(49,12)=-1;
E1(49,3)=-1;
E5(49,1)=qu7m;  %zuS7

E2(50,9)=-qu8m;
E3(50,13)=1;

E3(51,13)=-1;

E3(52,13)=1;
E1(52,4)=1;

E2(53,9)=qu8m;
E3(53,13)=-1;
E1(53,4)=-1;
E5(53,1)=qu8m; %zuS8

E2(54,1)=-m5;
E4(54,1)=1;
E5(54,1)=-v5m-m5;

E2(55,1)=-(M5+epsilon);
E4(55,1)=-1;
E5(55,1)=v5m-epsilon;  %delta5

E2(56,3)=-m7;
E4(56,3)=1;
E5(56,1)=-v7m-m7;

E2(57,3)=-(M7+epsilon);
E4(57,3)=-1;
E5(57,1)=v7m-epsilon;  %delta7

E2(58,4)=-m8;
E4(58,4)=1;
E5(58,1)=-v8m-m8;

E2(59,4)=-(M8+epsilon);
E4(59,4)=-1;
E5(59,1)=v8m-epsilon; %delta8

E2(60,5)=qs5m+qu5m;
E2(60,6)=-beta5*v5m;
E3(60,5)=-beta5;
E3(60,6)=beta5;
E1(60,1)=-1;
E5(60,1)=qu5m;

E2(61,1)=beta5*v5m;
E2(61,5)=qs5m-epsilon;
E2(61,6)=-beta5*v5m;
E3(61,1)=-beta5;
E3(61,5)=-beta5;
E3(61,6)=beta5;
E1(61,1)=1;
E4(61,1)=-beta5;
E5(61,1)=qs5m-epsilon;   %deltaS5

E2(62,1)=-1;
E2(62,6)=1;

E2(63,5)=-1;
E2(63,6)=1;
```

```
E2(64,1)=1;
E2(64,5)=1;
E2(64,6)=-1; %deltaS5a5
E5(64,1)=1;

E2(65,7)=qs7m+qu7m;
E2(65,8)=-beta7*v7m;
E3(65,7)=-beta7;
E3(65,8)=beta7;
E1(65,3)=-1;
E5(65,1)=qu7m;

E2(66,3)=beta7*v7m;
E2(66,7)=qs7m-epsilon;
E2(66,8)=-beta7*v7m;
E3(66,3)=-beta7;
E3(66,7)=-beta7;
E3(66,8)=beta7;
E1(66,3)=1;
E4(66,3)=-beta7;
E5(66,1)=qs7m-epsilon; %deltaS7

E2(67,3)=-1;
E2(67,8)=1;

E2(68,7)=-1;
E2(68,8)=1;

E2(69,3)=1;
E2(69,7)=1;
E2(69,8)=-1;   %deltaS7a7
E5(69,1)=1;

E2(70,9)=qs8m+qu8m;
E2(70,10)=-beta8*v8m;
E3(70,9)=-beta8;
E3(70,10)=beta8;
E1(70,4)=-1;
E5(70,1)=qu8m;

E2(71,4)=beta8*v8m;
E2(71,9)=qs8m-epsilon;
E2(71,10)=-beta8*v8m;
E3(71,4)=-beta8;
E3(71,9)=-beta8;
E3(71,10)=beta8;
E1(71,4)=1;
E4(71,4)=-beta8;
E5(71,1)=qs8m-epsilon; %deltaS8

E2(72,4)=-1;
E2(72,10)=1;

E2(73,9)=-1;
E2(73,10)=1;

E2(74,4)=1;
E2(74,9)=1;
E2(74,10)=-1;  %deltaS8a8
E5(74,1)=1;
%% constraints E1

%% constraints integration
F11=sym(zeros(74,27*5));    %initialize
F21=sym(zeros(74,1));
F31=sym(zeros(74,4));
F11=[-E1,E2,E3,zeros(74,27*4)];
F12=[-E4*B1,-E4*B2,-E4*B3,-E1,E2,E3,zeros(74,27*3)];
F13=[-E4*A*B1,-E4*A*B2,-E4*A*B3,-E4*B1,-E4*B2,-E4*B3,-E1,E2,E3,zeros(74,27*2)];
```

```
F14=[-E4*A^2*B1,-E4*A^2*B2,-E4*A^2*B3,-E4*A*B1,-E4*A*B2,-E4*A*B3,-E4*B1,-E4*B2,-E4*B3,-
E1,E2,E3,zeros(74,27)];
F15=[-E4*A^3*B1,-E4*A^3*B2,-E4*A^3*B3,-E4*A^2*B1,-E4*A^2*B2,-E4*A^2*B3,-E4*A*B1,-E4*A*B2,-
E4*A*B3,-E4*B1,-E4*B2,-E4*B3,-E1,E2,E3];

F21=E5;
F22=E4*B4*d0+E5;
F23=E4*A*B4*d0+E4*B4*d1+E5;
F24=E4*A*A*B4*d0+E4*A*B4*d1+E4*B4*d2+E5;
F25=E4*A*A*A*B4*d0+E4*A*A*B4*d1+E4*A*B4*d2+E4*B4*d3+E5;

F31=E4*v0;
F32=E4*A*v0;
F33=E4*A*A*v0;
F34=E4*A*A*A*v0;
F35=E4*A*A*A*A*v0;

F1=[F11;F12;F13;F14;F15];
F2=[F21;F22;F23;F24;F25];
F3=[F31;F32;F33;F34;F35];

%% cost function 1
L2=sym(zeros(1,27*5));

L2(1,1)=0.5*dt ;
L2(1,2)=0.5*dt ;
L2(1,3)=0.5*dt ;
L2(1,4)=0.5*dt ;  %operation cost from manipulation flows

L2(1,15)=1;
L2(1,17)=1;
L2(1,18)=1;

L2(1,5)=-v5m;
L2(1,7)=-v7m;
L2(1,8)=-v8m;   %tank over flow

L2(1,9)=-dt*qs5m;    %deltaS5
L2(1,10)=dt*beta5*v5m;  %deltaS55
L2(1,19)=dt*beta5;      %zs55
L2(1,20)=-dt*beta5;     %zs555
L2(1,25)=-dt;   %zus5  %qs5 overflow

L2(1,11)=-dt*qs7m;    %delaS7
L2(1,12)=dt*beta7*v7m;  %deltaS77
L2(1,21)=dt*beta7;      %zs77
L2(1,22)=-dt*beta7;     %zs777
L2(1,26)=-dt;   %zus7  %qs7 overflow

L2(1,13)=-dt*qs8m;    %deltaS8
L2(1,14)=dt*beta8*v8m;  %deltaS88
L2(1,23)=dt*beta8;      %zs88
L2(1,24)=-dt*beta8;     %zs888
L2(1,27)=-dt;  %zus8   %qs8 overflow
%% cost function 2
L2(1,1+27)=0.5*dt ;
L2(1,2+27)=0.5*dt ;
L2(1,3+27)=0.5*dt ;
L2(1,4+27)=0.5*dt ;  %operation cost from manipulation flows

L2(1,15+27)=1;
L2(1,17+27)=1;
L2(1,18+27)=1;

L2(1,5+27)=-v5m;
L2(1,7+27)=-v7m;
L2(1,8+27)=-v8m;   %tank over flow

L2(1,9+27)=-dt*qs5m;    %deltaS5
L2(1,10+27)=dt*beta5*v5m;  %deltaS55
L2(1,19+27)=dt*beta5;      %zs55
```

```
L2(1,20+27)=-dt*beta5;     %zs555
L2(1,25+27)=-dt;   %zus5  %qs5 overflow

L2(1,11+27)=-dt*qs7m;    %delaS7
L2(1,12+27)=dt*beta7*v7m;  %deltaS77
L2(1,21+27)=dt*beta7;      %zs77
L2(1,22+27)=-dt*beta7;     %zs777
L2(1,26+27)=-dt;   %zus7  %qs7 overflow

L2(1,13+27)=-dt*qs8m;    %deltaS8
L2(1,14+27)=dt*beta8*v8m;  %deltaS88
L2(1,23+27)=dt*beta8;      %zs88
L2(1,24+27)=-dt*beta8;     %zs888
L2(1,27+27)=-dt;  %zus8   %qs8 overflow
%% cost function 3
L2(1,1+27*2)=0.5*dt ;
L2(1,2+27*2)=0.5*dt ;
L2(1,3+27*2)=0.5*dt ;
L2(1,4+27*2)=0.5*dt ;  %operation cost from manipulation flows

L2(1,15+27*2)=1;
L2(1,17+27*2)=1;
L2(1,18+27*2)=1;

L2(1,5+27*2)=-v5m;
L2(1,7+27*2)=-v7m;
L2(1,8+27*2)=-v8m;   %tank over flow

L2(1,9+27*2)=-dt*qs5m;    %deltaS5
L2(1,10+27*2)=dt*beta5*v5m;  %deltaS55
L2(1,19+27*2)=dt*beta5;      %zs55
L2(1,20+27*2)=-dt*beta5;     %zs555
L2(1,25+27*2)=-dt;   %zus5  %qs5 overflow

L2(1,11+27*2)=-dt*qs7m;    %delaS7
L2(1,12+27*2)=dt*beta7*v7m;  %deltaS77
L2(1,21+27*2)=dt*beta7;      %zs77
L2(1,22+27*2)=-dt*beta7;     %zs777
L2(1,26+27*2)=-dt;   %zus7  %qs7 overflow

L2(1,13+27*2)=-dt*qs8m;    %deltaS8
L2(1,14+27*2)=dt*beta8*v8m;  %deltaS88
L2(1,23+27*2)=dt*beta8;      %zs88
L2(1,24+27*2)=-dt*beta8;     %zs888
L2(1,27+27*2)=-dt;  %zus8   %qs8 overflow
%% cost function 4
L2(1,1+27*3)=0.5*dt ;
L2(1,2+27*3)=0.5*dt ;
L2(1,3+27*3)=0.5*dt ;
L2(1,4+27*3)=0.5*dt ;  %operation cost from manipulation flows

L2(1,15+27*3)=1;
L2(1,17+27*3)=1;
L2(1,18+27*3)=1;

L2(1,5+27*3)=-v5m;
L2(1,7+27*3)=-v7m;
L2(1,8+27*3)=-v8m;   %tank over flow

L2(1,9+27*3)=-dt*qs5m;    %deltaS5
L2(1,10+27*3)=dt*beta5*v5m;  %deltaS55
L2(1,19+27*3)=dt*beta5;      %zs55
L2(1,20+27*3)=-dt*beta5;     %zs555
L2(1,25+27*3)=-dt;   %zus5  %qs5 overflow

L2(1,11+27*3)=-dt*qs7m;    %delaS7
L2(1,12+27*3)=dt*beta7*v7m;  %deltaS77
L2(1,21+27*3)=dt*beta7;      %zs77
L2(1,22+27*3)=-dt*beta7;     %zs777
L2(1,26+27*3)=-dt;   %zus7  %qs7 overflow
```

```
L2(1,13+27*3)=-dt*qs8m;    %deltaS8
L2(1,14+27*3)=dt*beta8*v8m;  %deltaS88
L2(1,23+27*3)=dt*beta8;      %zs88
L2(1,24+27*3)=-dt*beta8;     %zs888
L2(1,27+27*3)=-dt;  %zus8   %qs8 overflow
%% cost function 5
L2(1,1+27*4)=0.5*dt ;
L2(1,2+27*4)=0.5*dt ;
L2(1,3+27*4)=0.5*dt ;
L2(1,4+27*4)=0.5*dt ;  %operation cost from manipulation flows

L2(1,15+27*4)=1;
L2(1,17+27*4)=1;
L2(1,18+27*4)=1;

L2(1,5+27*4)=-v5m;
L2(1,7+27*4)=-v7m;
L2(1,8+27*4)=-v8m;   %tank over flow

L2(1,9+27*4)=-dt*qs5m;    %deltaS5
L2(1,10+27*4)=dt*beta5*v5m;  %deltaS55
L2(1,19+27*4)=dt*beta5;      %zs55
L2(1,20+27*4)=-dt*beta5;     %zs555
L2(1,25+27*4)=-dt;   %zus5  %qs5 overflow

L2(1,11+27*4)=-dt*qs7m;    %delaS7
L2(1,12+27*4)=dt*beta7*v7m;  %deltaS77
L2(1,21+27*4)=dt*beta7;      %zs77
L2(1,22+27*4)=-dt*beta7;     %zs777
L2(1,26+27*4)=-dt;   %zus7  %qs7 overflow

L2(1,13+27*4)=-dt*qs8m;    %deltaS8
L2(1,14+27*4)=dt*beta8*v8m;  %deltaS88
L2(1,23+27*4)=dt*beta8;      %zs88
L2(1,24+27*4)=-dt*beta8;     %zs888
L2(1,27+27*4)=-dt;  %zus8   %qs8 overflow

%% constants specification
epsilon =1e-13;
S=[323576,164869,5076,754131];%surface area
phi=[1.03,10.4,0,0.48];%absorption coefficient
beta=[7.1e-4,5.8e-4,2e-3,1e-3];%volumetric flow coefficient
vm=[16901,43000,35000,26659];%maximum volume
dt=300;%sample time


x0=initial_states;
rain = rain_profile;

p = rain.*1.2./1000./3600;% precipitation in unit of m/s


v5m=vm(1);
v6m=vm(3);
v7m=vm(2);
v8m=vm(1);

v05=x0(1);
v06=x0(2);
v07=x0(3);
v08=x0(4);

S5=S(1);
S7=S(2);
S8=S(1);

phi5=phi(1);
phi7=phi(2);
phi8=phi(1);

beta5=beta(1);
```

```
beta6=beta(3);
beta7=beta(2);
beta8=beta(1);

p05=p(1);
p06=0;%real
p07=p(1);
p08=p(1);

p15=p(2);
p16=0;%real
p17=p(2);
p18=p(2);

p25=p(3);
p26=0;%real
p27=p(3);
p28=p(3);

p35=p(4);
p36=0;
p37=p(4);
p38=p(4);


p45=p(5);
p46=0;
p47=p(5);
p48=p(5);

% qs5m=6.8;
% qs7m=15;
% qs8m=5;

qs5m=1000;
qs7m=1000;
qs8m=1000;


v5u=v5m*2+S5*dt*max(p)*phi5;
v7u=v7m*2+S7*dt*max(p)*phi7;
v8u=v8m*2+S8*dt*max(p)*phi8;

v6u=35000;


vu=[v5u,v6u,v7u,v8u];


qu5m=9.1;
qu6m=7;
qu7m=25;
qu8m=9.1;

M5=v5u-v5m;
m5=-v5m;

M7=v7u-v7m;
m7=-v7m;

M8=v8u-v8m;
m8=-v8m;

lb = zeros(27*5,1);                  %Bounds on x (lb <= x)
uub = [qu5m;qu6m;qu7m;qu8m];          %u upper bound

dub = zeros(10,1);                  %delta upperbound
for i = 1:10
   dub(i,1)=1;
end
zub = zeros(13,1);            %z upper bound
```

```
for i=1:13
    zub(i,1)=inf;
end

ub = [uub;dub;zub;uub;dub;zub;uub;dub;zub;uub;dub;zub;uub;dub;zub];
f=eval(L2');


a=eval(F1);
b=eval(F2+F3);
%%
xtype =
'CCCCBBBBBBBBBBCCCCCCCCCCCCCCCCCBBBBBBBBBBCCCCCCCCCCCCCCCCCBBBBBBBBBBCCCCCCCCCCCCCCCCCBBBBBBBBBBC
CCCCCCCCCCCCCCCCBBBBBBBBBBCCCCCCCCCCCCC';

Opt = opti('f',f,'ineq',a,b,'bounds',lb,ub,'xtype',xtype);
[x,fval,exitflag,info] = solve(Opt);

u05 = x(1);
u06 = x(2);
u07 = x(3);
u08 = x(4);
delta05=x(5);
delta06=x(6);
delta07=x(7);
delta08=x(8);
delta0S5=x(9);
delta0S5a5=x(10);
delta0S7=x(11);
delta0S7a7=x(12);
delta0S8=x(13);
delta0S8a8=x(14);
z05=x(15);
z06=x(16);
z07=x(17);
z08=x(18);
z0S5a5=x(19);
z0S5a5a5=x(20);
z0S7a7=x(21);
z0S7a7a7=x(22);
z0S8a8=x(23);
z0S8a8a8=x(24);
z0uS5=x(25);
z0uS7=x(26);
z0uS8=x(27);

u15 = x(1+27);
u16 = x(2+27);
u17 = x(3+27);
u18 = x(4+27);
delta15=x(5+27);
delta16=x(6+27);
delta17=x(7+27);
delta18=x(8+27);
delta1S5=x(9+27);
delta1S5a5=x(10+27);
delta1S7=x(11+27);
delta1S7a7=x(12+27);
delta1S8=x(13+27);
delta1S8a8=x(14+27);
z15=x(15+27);
z16=x(16+27);
z17=x(17+27);
z18=x(18+27);
z1S5a5=x(19+27);
z1S5a5a5=x(20+27);
z1S7a7=x(21+27);
z1S7a7a7=x(22+27);
z1S8a8=x(23+27);
z1S8a8a8=x(24+27);
z1uS5=x(25+27);
```

```
z1uS7=x(26+27);
z1uS8=x(27+27);

u25 = x(1+27*2);
u26 = x(2+27*2);
u27 = x(3+27*2);
u28 = x(4+27*2);
delta25=x(5+27*2);
delta26=x(6+27*2);
delta27=x(7+27*2);
delta28=x(8+27*2);
delta2S5=x(9+27*2);
delta2S5a5=x(10+27*2);
delta2S7=x(11+27*2);
delta2S7a7=x(12+27*2);
delta2S8=x(13+27*2);
delta2S8a8=x(14+27*2);
z25=x(15+27*2);
z26=x(16+27*2);
z27=x(17+27*2);
z28=x(18+27*2);
z2S5a5=x(19+27*2);
z2S5a5a5=x(20+27*2);
z2S7a7=x(21+27*2);
z2S7a7a7=x(22+27*2);
z2S8a8=x(23+27*2);
z2S8a8a8=x(24+27*2);
z2uS5=x(25+27*2);
z2uS7=x(26+27*2);
z2uS8=x(27+27*2);

u35 = x(1+27*3);
u36 = x(2+27*3);
u37 = x(3+27*3);
u38 = x(4+27*3);
delta35=x(5+27*3);
delta36=x(6+27*3);
delta37=x(7+27*3);
delta38=x(8+27*3);
delta3S5=x(9+27*3);
delta3S5a5=x(10+27*3);
delta3S7=x(11+27*3);
delta3S7a7=x(12+27*3);
delta3S8=x(13+27*3);
delta3S8a8=x(14+27*3);
z35=x(15+27*3);
z36=x(16+27*3);
z37=x(17+27*3);
z38=x(18+27*3);
z3S5a5=x(19+27*3);
z3S5a5a5=x(20+27*3);
z3S7a7=x(21+27*3);
z3S7a7a7=x(22+27*3);
z3S8a8=x(23+27*3);
z3S8a8a8=x(24+27*3);
z3uS5=x(25+27*3);
z3uS7=x(26+27*3);
z3uS8=x(27+27*3);

u45 = x(1+27*4);
u46 = x(2+27*4);
u47 = x(3+27*4);
u48 = x(4+27*4);
delta45=x(5+27*4);
delta46=x(6+27*4);
delta47=x(7+27*4);
delta48=x(8+27*4);
delta4S5=x(9+27*4);
delta4S5a5=x(10+27*4);
delta4S7=x(11+27*4);
delta4S7a7=x(12+27*4);
```

```
delta4S8=x(13+27*4);
delta4S8a8=x(14+27*4);
z45=x(15+27*4);
z46=x(16+27*4);
z47=x(17+27*4);
z48=x(18+27*4);
z4S5a5=x(19+27*4);
z4S5a5a5=x(20+27*4);
z4S7a7=x(21+27*4);
z4S7a7a7=x(22+27*4);
z4S8a8=x(23+27*4);
z4S8a8a8=x(24+27*4);
z4uS5=x(25+27*4);
z4uS7=x(26+27*4);
z4uS8=x(27+27*4);

end_states =eval(v1);
step_cost = eval(L2(1:27)*x(1:27));
action = [x(1);x(2);x(3);x(4)];
cp_time = info.Time;
% qu7= [x(3);x(3+27);x(3+27*2);x(3+27*3);x(3+27*4)];
% qu8= [x(3);x(4+27);x(4+27*2);x(4+27*3);x(4+27*4)];
% qs5= [(beta5*v5m*x(5)+beta5*x0(1)-beta5*x(15)-x(1))*(1-x(9))+qs5m*x(9),
%        (beta5*v5m*x(5+27)+eval(beta5*v1(1))-beta5*x(15+27)-x(1+27))*(1-x(9+27))+qs5m*x(9+27),
%        (beta5*v5m*x(5+27*2)+eval(beta5*v2(1))-beta5*x(15+27*2)-x(1+27*2))*(1-
x(9+27*2))+qs5m*x(9+27*2),
%        (beta5*v5m*x(5+27*3)+eval(beta5*v3(1))-beta5*x(15+27*3)-x(1+27*3))*(1-
x(9+27*3))+qs5m*x(9+27*3),
%        (beta5*v5m*x(5+27*4)+eval(beta5*v4(1))-beta5*x(15+27*4)-x(1+27*4))*(1-
x(9+27*3))+qs5m*x(9+27*4)];
% qs8 = [(beta8*v8m*x(8)+beta8*x0(4)-beta8*x(18)-x(4))*(1-x(13))+qs8m*x(13),
%         (beta8*v8m*x(8+27)+eval(beta8*v1(4))-beta8*x(18+27)-x(4+27))*(1-x(13+27))+qs8m*x(13+27),
%         (beta8*v8m*x(8+27*2)+eval(beta8*v2(4))-beta8*x(18+27*2)-x(4+27*2))*(1-
x(13+27*2))+qs8m*x(13+27*2),
%         (beta8*v8m*x(8+27*3)+eval(beta8*v3(4))-beta8*x(18+27*3)-x(4+27*3))*(1-
x(13+27*3))+qs8m*x(13+27*3),
%         (beta8*v8m*x(8+27*4)+eval(beta8*v4(4))-beta8*x(18+27*4)-x(4+27*4))*(1-
x(13+27*4))+qs8m*x(13+27*4)];

qu7= [x(3+27);x(3+27*2);x(3+27*3);x(3+27*4);x(3+27*4);x(3)];
qu8= [x(4+27);x(4+27*2);x(4+27*3);x(4+27*4);x(4+27*4);x(4)];
qs5= [(beta5*v5m*x(5+27)+eval(beta5*v1(1))-beta5*x(15+27)-x(1+27))*(1-x(9+27))+qs5m*x(9+27),
      (beta5*v5m*x(5+27*2)+eval(beta5*v2(1))-beta5*x(15+27*2)-x(1+27*2))*(1-
x(9+27*2))+qs5m*x(9+27*2),
      (beta5*v5m*x(5+27*3)+eval(beta5*v3(1))-beta5*x(15+27*3)-x(1+27*3))*(1-
x(9+27*3))+qs5m*x(9+27*3),
      (beta5*v5m*x(5+27*4)+eval(beta5*v4(1))-beta5*x(15+27*4)-x(1+27*4))*(1-
x(9+27*3))+qs5m*x(9+27*4),
      (beta5*v5m*x(5+27*4)+eval(beta5*v4(1))-beta5*x(15+27*4)-x(1+27*4))*(1-
x(9+27*3))+qs5m*x(9+27*4),
      (beta5*v5m*x(5)+beta5*x0(1)-beta5*x(15)-x(1))*(1-x(9))+qs5m*x(9)];
qs8 = [(beta8*v8m*x(8+27)+eval(beta8*v1(4))-beta8*x(18+27)-x(4+27))*(1-x(13+27))+qs8m*x(13+27),
       (beta8*v8m*x(8+27*2)+eval(beta8*v2(4))-beta8*x(18+27*2)-x(4+27*2))*(1-
x(13+27*2))+qs8m*x(13+27*2),
       (beta8*v8m*x(8+27*3)+eval(beta8*v3(4))-beta8*x(18+27*3)-x(4+27*3))*(1-
x(13+27*3))+qs8m*x(13+27*3),
       (beta8*v8m*x(8+27*4)+eval(beta8*v4(4))-beta8*x(18+27*4)-x(4+27*4))*(1-
x(13+27*4))+qs8m*x(13+27*4),
       (beta8*v8m*x(8+27*4)+eval(beta8*v4(4))-beta8*x(18+27*4)-x(4+27*4))*(1-
x(13+27*4))+qs8m*x(13+27*4),
       (beta8*v8m*x(8)+beta8*x0(4)-beta8*x(18)-x(4))*(1-x(13))+qs8m*x(13)];

op_cost = eval(L2(1:4)*x(1:4));   %control cost
of_cost = eval(L2(5:27)*x(5:27));  %overflow cost

suc = exitflag;



end
```

Decentralized MPC subsystem 3

```
%% susbsytem 3 function
function
[end_states,step_cost,action,cp_time,qu11,qs11,op_cost,of_cost,suc]=sub3function(initial_states,r
ain_profile,disturbance)

% system dynamics
syms dt epsilon ...
        v9 v9u v9l v9m beta9 S9 phi9 delta9 deltaS9 deltaS9a9 z9 zS9a9 zS9a9a9 zuS9 qu9 qs9m...
        v10 v10m beta10 qu10 ...
        v11 v11u v11l v11m beta11 S11 phi11 delta11 deltaS11 deltaS11a11 z11 zS11a11 zS11a11a11
zuS11 qu11 qs11m...
        qu9m qu10m qu11m...
        qs9m qs11m  ...
        m9 m10 m11...
        M9 M10 M11...
        v09 v010 v011...
        p09 p010 p011...
        p19 p110 p111...
        p29 p210 p211...
        p39 p310 p311...
        p49 p410 p411...
        z09 z010 z011...
        z19 z110 z111...
        z29 z210 z211...
        z39 z310 z311...
        z0S9a9 z0S9a9a9 z0S11a11 z0S11a11a11 z0uS9 z0uS11 ...
        z1S9a9 z1S9a9a9 z1S11a11 z1S11a11a11 z1uS9 z1uS11 ...
        z2S9a9 z2S9a9a9 z2S11a11 z2S11a11a11 z2uS9 z2uS11 ...
        z3S9a9 z3S9a9a9 z3S11a11 z3S11a11a11 z3uS9 z3uS11 ...
        delta09 delta010 delta011 delta0S9 delta0S9a9 delta0S11 delta0S11a11 ...
        delta19 delta110 delta111 delta1S9 delta1S9a9 delta1S11 delta1S11a11 ...
        delta29 delta210 delta211 delta2S9 delta2S9a9 delta2S11 delta2S11a11 ...
        delta39 delta310 delta311 delta3S9 delta3S9a9 delta3S11 delta3S11a11 ...
        u09 u010 u011...
        u19 u110 u111...
        u29 u210 u211...
        u39 u310 u311...

% qu7=[0,0,0,0,0];
qu7 = disturbance;

A = sym(zeros(3,3));   %tank
B1 = sym(zeros(3,3));  %control
B2 = sym(zeros(3,7));   %delta
B3 = sym(zeros(3,9));   %z
B4= sym(zeros(3,3));    % disturbance

d0=[p09;p010;p011];
d1=[p19;p110;p111];
d2=[p29;p210;p211];
d3=[p39;p310;p311];
d4=[p49;p410;p411];

z0=[z09;z010;z011;z0S9a9;z0S9a9a9;z0S11a11;z0S11a11a11;z0uS9;z0uS11];
z1=[z19;z110;z111;z1S9a9;z1S9a9a9;z1S11a11;z1S11a11a11;z1uS9;z1uS11];
z2=[z29;z210;z211;z2S9a9;z2S9a9a9;z2S11a11;z2S11a11a11;z2uS9;z2uS11];
z3=[z39;z310;z311;z3S9a9;z3S9a9a9;z3S11a11;z3S11a11a11;z3uS9;z3uS11];

delta0 = [delta09;delta010;delta011;delta0S9;delta0S9a9;delta0S11;delta0S11a11];
delta1 = [delta19;delta110;delta111;delta1S9;delta1S9a9;delta1S11;delta1S11a11];
delta2 = [delta29;delta210;delta211;delta2S9;delta2S9a9;delta2S11;delta2S11a11];
delta3 = [delta39;delta310;delta311;delta3S9;delta3S9a9;delta3S11;delta3S11a11];

u0 = [u09;u010;u011];
u1 = [u19;u110;u111];
u2 = [u29;u210;u211];
u3 = [u39;u310;u311];
```

```
v0=[v09;v010;v011];

A(1,1)=1-dt*beta9;    %+8
B2(1,1)=(1-dt*beta9)*v9m;
B3(1,1)=dt*beta9-1;
B4(1,1)=dt*phi9*S9;

A(2,2)=1;
B1(2,1)=dt;
B1(2,2)=-dt;
B4(2,2)=dt;  % u7 considered as disturbance

A(3,1)=dt*beta9;
A(3,3)=1-dt*beta11;
B1(3,1)=-dt;
B1(3,2)=dt;
B2(3,1)=dt*beta9*v9m;
B2(3,3)=(1-dt*beta11)*v11m;
B2(3,4)=dt*qs9m;
B2(3,5)=-dt*beta9*v9m;
B3(3,1)=-dt*beta9;
B3(3,3)=dt*beta11-1;
B3(3,4)=-dt*beta9;
B3(3,5)=dt*beta9;
B3(3,8)=dt;
B4(3,3)=dt*phi11*S11;

v1=A*v0+B1*u0+B2*delta0+B3*z0+B4*d0;
v2=A*v1+B1*u1+B2*delta1+B3*z1+B4*d1;
v3=A*v2+B1*u2+B2*delta2+B3*z2+B4*d2;
v4=A*v3+B1*u3+B2*delta3+B3*z3+B4*d3;
%% constraints E
E1=sym(zeros(50,3)); %qu
E2=sym(zeros(50,7)); %delta
E3=sym(zeros(50,9)); %z
E4=sym(zeros(50,3)); %v
E5=sym(zeros(50,1));  %constants

E2(1,1)=-beta9*v9m;
E3(1,1)=beta9;
E1(1,1)=-1;
E4(1,1)=beta9; %qu9

% E1(2,2)=-1;
% E4(2,2)=beta10; %qu10

E1(2,1)=dt;
E1(2,2)=-dt;
E4(2,2)=1;
E5(2,1)=qu7(1)*dt;

E1(3,1)=-dt;
E1(3,2)=dt;
E4(3,2)=-1;
E5(3,1)=v10m-dt*qu7(1); %qu9

E2(4,3)=-beta11*v11m;
E3(4,3)=beta11;
E1(4,3)=-1;
E4(4,3)=beta11; %qu11

E2(5,1)=-v9u;
E3(5,1)=1;

E3(6,1)=-1;

E3(7,1)=1;
E4(7,1)=1;
```

```
E2(8,1)=v9u;
E3(8,1)=-1;
E4(8,1)=-1;
E5(8,1)=v9u; %z9

E2(9,3)=-v11u;
E3(9,3)=1;

E3(10,3)=-1;

E3(11,3)=1;
E4(11,3)=1;

E2(12,3)=v11u;
E3(12,3)=-1;
E4(12,3)=-1;
E5(12,1)=v11u; %z11

E2(13,4)=-v9u;
E3(13,4)=1;

E3(14,4)=-1;

E3(15,4)=1;
E4(15,1)=1;

E2(16,4)=v9u;
E3(16,4)=-1;
E4(16,1)=-1;
E5(16,1)=v9u;  %zs99

E2(17,5)=-v9u;
E3(17,5)=1;

E3(18,5)=-1;

E3(19,5)=1;
E4(19,1)=1;

E2(20,5)=v9u;
E3(20,5)=-1;
E4(20,1)=-1;
E5(20,1)=v9u;  %zs999

E2(21,6)=-v11u;
E3(21,6)=1;

E3(22,6)=-1;

E3(23,6)=1;
E4(23,3)=1;

E2(24,6)=v11u;
E3(24,6)=-1;
E4(24,3)=-1;
E5(24,1)=v11u;  %zs11a11

E2(25,7)=-v11u;
E3(25,7)=1;

E3(26,7)=-1;

E3(27,7)=1;
E4(27,3)=1;

E2(28,7)=v11u;
E3(28,7)=-1;
E4(28,3)=-1;
E5(28,1)=v11u;  %zs11a11a11

E2(29,4)=-qu9m;
```

```
E3(29,8)=1;

E3(30,8)=-1;

E3(31,8)=1;
E1(31,1)=1;

E2(32,4)=qu9m;
E3(32,8)=-1;
E1(32,1)=-1;
E5(32,1)=qu9m;  %zuS9

E2(33,6)=-qu11m;
E3(33,9)=1;

E3(34,9)=-1;

E3(35,9)=1;
E1(35,3)=1;

E2(36,6)=qu11m;
E3(36,9)=-1;
E1(36,3)=-1;
E5(36,1)=qu11m;  %zuS11

E2(37,1)=-m9;
E4(37,1)=1;
E5(37,1)=-v9m-m9;

E2(38,1)=-(M9+epsilon);
E4(38,1)=-1;
E5(38,1)=v9m-epsilon;  %delta9

E2(39,3)=-m11;
E4(39,3)=1;
E5(39,1)=-v11m-m11;

E2(40,3)=-(M11+epsilon);
E4(40,3)=-1;
E5(40,1)=v11m-epsilon;  %delta 11

E2(41,4)=qs9m+qu9m;
E2(41,5)=-beta9*v9m;
E3(41,4)=-beta9;
E3(41,5)=beta9;
E1(41,1)=-1;
E5(41,1)=qu9m;

E2(42,1)=beta9*v9m;
E2(42,4)=qs9m-epsilon;
E2(42,5)=-beta9*v9m;
E3(42,1)=-beta9;
E3(42,4)=-beta9;
E3(42,5)=beta9;
E1(42,1)=1;
E4(42,1)=-beta9;
E5(42,1)=qs9m-epsilon;  %deltaS9

E2(43,1)=-1;
E2(43,5)=1;

E2(44,4)=-1;
E2(44,5)=1;

E2(45,1)=1;
E2(45,4)=1;
E2(45,5)=-1;  %deltaS9a9
E5(45,1)=1;

E2(46,6)=qs11m+qu11m;
E2(46,7)=-beta11*v11m;
```

```
E3(46,6)=-beta11;
E3(46,7)=beta11;
E1(46,3)=-1;
E5(46,1)=qu11m;

E2(47,3)=beta11*v11m;
E2(47,6)=qs11m-epsilon;
E2(47,7)=-beta11*v11m;
E3(47,3)=-beta11;
E3(47,6)=-beta11;
E3(47,7)=beta11;
E1(47,3)=1;
E4(47,3)=-beta11;
E5(47,1)=qs11m-epsilon;  %deltaS11

E2(48,3)=-1;
E2(48,7)=1;

E2(49,6)=-1;
E2(49,7)=1;

E2(50,3)=1;
E2(50,6)=1;
E2(50,7)=-1;  %deltaS11a11
E5(50,1)=1;
%% constraints E1
E11=E1;
E21=E2;
E31=E3;
E41=E4;
E51=E5;
E51(2,1)=qu7(2)*dt;
E51(3,1)=v10m-dt*qu7(2); %qu9

%% constraints E2
E12=E1;
E22=E2;
E32=E3;
E42=E4;
E52=E5;
E52(2,1)=qu7(3)*dt;
E52(3,1)=v10m-dt*qu7(3); %qu9
%% constraints E3
E13=E1;
E23=E2;
E33=E3;
E43=E4;
E53=E5;
E53(2,1)=qu7(4)*dt;
E53(3,1)=v10m-dt*qu7(4); %qu9
%% constraints E4
E14=E1;
E24=E2;
E34=E3;
E44=E4;
E54=E5;
E54(2,1)=qu7(5)*dt;
E54(3,1)=v10m-dt*qu7(5); %qu9
%% constraints integration
F11=sym(zeros(49,19*5));    %initialize
F21=sym(zeros(49,1));
F31=sym(zeros(49,3));
F11=[-E1,E2,E3,zeros(49,19*4)];
F12=[-E41*B1,-E41*B2,-E41*B3,-E11,E21,E31,zeros(49,19*3)];
F13=[-E42*A*B1,-E42*A*B2,-E42*A*B3,-E42*B1,-E42*B2,-E42*B3,-E12,E22,E32,zeros(49,19*2)];
F14=[-E43*A^2*B1,-E43*A^2*B2,-E43*A^2*B3,-E43*A*B1,-E43*A*B2,-E43*A*B3,-E43*B1,-E43*B2,-E43*B3,-
E13,E23,E33,zeros(49,19)];
F15=[-E44*A^3*B1,-E44*A^3*B2,-E44*A^3*B3,-E44*A^2*B1,-E44*A^2*B2,-E44*A^2*B3,-E44*A*B1,-
E44*A*B2,-E44*A*B3,-E44*B1,-E44*B2,-E44*B3,-E14,E24,E34];

F21=E5;
```

```
F22=E41*B4*d0+E51;
F23=E42*A*B4*d0+E42*B4*d1+E52;
F24=E43*A*A*B4*d0+E43*A*B4*d1+E43*B4*d2+E53;
F25=E44*A*A*A*B4*d0+E44*A*A*B4*d1+E44*A*B4*d2+E44*B4*d3+E54;

F31=E4*v0;
F32=E41*A*v0;
F33=E42*A*A*v0;
F34=E43*A*A*A*v0;
F35=E44*A*A*A*A*v0;

F1=[F11;F12;F13;F14;F15];
F2=[F21;F22;F23;F24;F25];
F3=[F31;F32;F33;F34;F35];
%% cost function 1

L2=sym(zeros(1,19*5));

L2(1,1)=0.5*dt ;
L2(1,2)=0.5*dt ;
L2(1,3)=0.5*dt ;   %operation cost from manipulation flows

L2(1,11)=1;
L2(1,13)=1;

L2(1,4)=-v9m;
L2(1,6)=-v11m;   %tank over flow


L2(1,7)=-dt*qs9m;   %deltaS9
L2(1,8)=dt*beta9*v9m;  %deltaS99
L2(1,14)=dt*beta9;     %zs99
L2(1,15)=-dt*beta9;    %zs999
L2(1,18)=-dt;  %zus9   %qs9 overflow

L2(1,9)=-dt*qs11m;   %deltaS11
L2(1,10)=dt*beta11*v11m;  %deltaS111
L2(1,16)=dt*beta11;     %zs11a11
L2(1,17)=-dt*beta11;    %zs11a11a11
L2(1,19)=-dt;  %zus11   %qs11 overflow
%% cost function 2
L2(1,1+19)=0.5*dt ;
L2(1,2+19)=0.5*dt ;
L2(1,3+19)=0.5*dt ;   %operation cost from manipulation flows

L2(1,11+19)=1;
L2(1,13+19)=1;

L2(1,4+19)=-v9m;
L2(1,6+19)=-v11m;   %tank over flow


L2(1,7+19)=-dt*qs9m;   %deltaS9
L2(1,8+19)=dt*beta9*v9m;  %deltaS99
L2(1,14+19)=dt*beta9;     %zs99
L2(1,15+19)=-dt*beta9;    %zs999
L2(1,18+19)=-dt;  %zus9   %qs9 overflow

L2(1,9+19)=-dt*qs11m;   %deltaS11
L2(1,10+19)=dt*beta11*v11m;  %deltaS111
L2(1,16+19)=dt*beta11;     %zs11a11
L2(1,17+19)=-dt*beta11;    %zs11a11a11
L2(1,19+19)=-dt;  %zus11   %qs11 overflow

%% cost function 3

L2(1,1+19*2)=0.5*dt ;
L2(1,2+19*2)=0.5*dt ;
L2(1,3+19*2)=0.5*dt ;   %operation cost from manipulation flows

L2(1,11+19*2)=1;
```

```
L2(1,13+19*2)=1;

L2(1,4+19*2)=-v9m;
L2(1,6+19*2)=-v11m;   %tank over flow


L2(1,7+19*2)=-dt*qs9m;   %deltaS9
L2(1,8+19*2)=dt*beta9*v9m;  %deltaS99
L2(1,14+19*2)=dt*beta9;     %zs99
L2(1,15+19*2)=-dt*beta9;    %zs999
L2(1,18+19*2)=-dt;  %zus9   %qs9 overflow

L2(1,9+19*2)=-dt*qs11m;   %deltaS11
L2(1,10+19*2)=dt*beta11*v11m;  %deltaS111
L2(1,16+19*2)=dt*beta11;     %zs11a11
L2(1,17+19*2)=-dt*beta11;    %zs11a11a11
L2(1,19+19*2)=-dt;  %zus11   %qs11 overflow
%% cost function 4
L2(1,1+19*3)=0.5*dt ;
L2(1,2+19*3)=0.5*dt ;
L2(1,3+19*3)=0.5*dt ;   %operation cost from manipulation flows

L2(1,11+19*3)=1;
L2(1,13+19*3)=1;

L2(1,4+19*3)=-v9m;
L2(1,6+19*3)=-v11m;   %tank over flow


L2(1,7+19*3)=-dt*qs9m;   %deltaS9
L2(1,8+19*3)=dt*beta9*v9m;  %deltaS99
L2(1,14+19*3)=dt*beta9;     %zs99
L2(1,15+19*3)=-dt*beta9;    %zs999
L2(1,18+19*3)=-dt;  %zus9   %qs9 overflow

L2(1,9+19*3)=-dt*qs11m;   %deltaS11
L2(1,10+19*3)=dt*beta11*v11m;  %deltaS111
L2(1,16+19*3)=dt*beta11;     %zs11a11
L2(1,17+19*3)=-dt*beta11;    %zs11a11a11
L2(1,19+19*3)=-dt;  %zus11   %qs11 overflow

%% cost function 5
L2(1,1+19*4)=0.5*dt ;
L2(1,2+19*4)=0.5*dt ;
L2(1,3+19*4)=0.5*dt ;   %operation cost from manipulation flows

L2(1,11+19*4)=1;
L2(1,13+19*4)=1;

L2(1,4+19*4)=-v9m;
L2(1,6+19*4)=-v11m;   %tank over flow


L2(1,7+19*4)=-dt*qs9m;   %deltaS9
L2(1,8+19*4)=dt*beta9*v9m;  %deltaS99
L2(1,14+19*4)=dt*beta9;     %zs99
L2(1,15+19*4)=-dt*beta9;    %zs999
L2(1,18+19*4)=-dt;  %zus9   %qs9 overflow

L2(1,9+19*4)=-dt*qs11m;   %deltaS11
L2(1,10+19*4)=dt*beta11*v11m;  %deltaS111
L2(1,16+19*4)=dt*beta11;     %zs11a11
L2(1,17+19*4)=-dt*beta11;    %zs11a11a11
L2(1,19+19*4)=-dt;  %zus11   %qs11 overflow

%%  constants specification
epsilon =1e-13;
S=[323576,164869,5076,754131];%surface area
phi=[1.03,10.4,0,0.48];%absorption coefficient
beta=[7.1e-4,5.8e-4,2e-3,1e-3];%volumetric flow coefficient
vm=[16901,43000,35000,26659];%maximum volume
```

```
dt=300;%sample time

% x0=[0,0,0];
x0 = initial_states;

% rain = [100,130,140,150,90];
rain = rain_profile;

p = rain.*1.2./1000./3600;% precipitation in unit of m/s

v9m=vm(2);
v10m=vm(3);
v11m=vm(4);


v09=x0(1);
v010=x0(2);
v011=x0(3);

S9=S(2);
S11=S(4);


phi9=phi(2);
phi11=phi(4);


beta9=beta(2);
beta10=beta(3);
beta11=beta(4);

p09=p(1);
p010=qu7(1);%real
p011=p(1);

p19=p(2);
p110=qu7(2);%real
p111=p(2);

p29=p(3);
p210=qu7(3);%real
p211=p(3);

p39=p(4);
p310=qu7(4);
p311=p(4);

p49=p(5);
p410=qu7(5);%;
p411=p(5);

% qs9m=15.6;
% qs11m=18;

qs9m=1000;
qs11m=1000;

v9u=v9m*2+S9*dt*max(p)*phi9;
v11u=v11m*2+S11*dt*max(p)*phi11;

v10u=35000;
```

```
vu=[v9u,v10u,v11u];


qu9m=25;
qu10m=30;
qu11m=25;



M9=v9u-v9m;
m9=-v9m;

M11=v11u-v11m;
m11=-v11m;



lb = zeros(19*5,1);                 %Bounds on x (lb <= x)
uub = [qu9m;qu10m;qu11m];         %u upper bound

dub = ones(7,1);                %delta upperbound

zub = zeros(9,1);            %z upper bound
for i=1:9
    zub(i,1)=inf;
end

ub = [uub;dub;zub;uub;dub;zub;uub;dub;zub;uub;dub;zub;uub;dub;zub];
f=eval(L2');

a=eval(F1);
b=eval(F2+F3);

%%
xtype =
'CCCBBBBBBBCCCCCCCCCCCCBBBBBBBCCCCCCCCCCCCBBBBBBBCCCCCCCCCCCCBBBBBBBCCCCCCCCCCCCBBBBBBBCCCCCCCCC';
Opt = opti('f',f,'ineq',a,b,'bounds',lb,ub,'xtype',xtype);
[x,fval,exitflag,info] = solve(Opt);

u09=x(1);
u010=x(2);
u011=x(3);
delta09=x(4);
delta010=x(5);
delta011=x(6);
delta0S9=x(7);
delta0S9a9=x(8);
delta0S11=x(9);
delta0S11a11=x(10);
z09=x(11);
z010=x(12);
z011=x(13);
z0S9a9=x(14);
z0S9a9a9=x(15);
z0S11a11=x(16);
z0S11a11a11=x(17);
z0uS9=x(18);
z0uS11=x(19);

u19=x(1+19);
u110=x(2+19);
u111=x(3+19);
delta19=x(4+19);
delta110=x(5+19);
delta111=x(6+19);
delta1S9=x(7+19);
delta1S9a9=x(8+19);
delta1S11=x(9+19);
delta1S11a11=x(10+19);
z19=x(11+19);
```

```
z110=x(12+19);
z111=x(13+19);
z1S9a9=x(14+19);
z1S9a9a9=x(15+19);
z1S11a11=x(16+19);
z1S11a11a11=x(17+19);
z1uS9=x(18+19);
z1uS11=x(19+19);

u29=x(1+19*2);
u210=x(2+19*2);
u211=x(3+19*2);
delta29=x(4+19*2);
delta210=x(5+19*2);
delta211=x(6+19*2);
delta2S9=x(7+19*2);
delta2S9a9=x(8+19*2);
delta2S11=x(9+19*2);
delta2S11a11=x(10+19*2);
z29=x(11+19*2);
z210=x(12+19*2);
z211=x(13+19*2);
z2S9a9=x(14+19*2);
z2S9a9a9=x(15+19*2);
z2S11a11=x(16+19*2);
z2S11a11a11=x(17+19*2);
z2uS9=x(18+19*2);
z2uS11=x(19+19*2);

u39=x(1+19*3);
u310=x(2+19*3);
u311=x(3+19*3);
delta39=x(4+19*3);
delta310=x(5+19*3);
delta311=x(6+19*3);
delta3S9=x(7+19*3);
delta3S9a9=x(8+19*3);
delta3S11=x(9+19*3);
delta3S11a11=x(10+19*3);
z39=x(11+19*3);
z310=x(12+19*3);
z311=x(13+19*3);
z3S9a9=x(14+19*3);
z3S9a9a9=x(15+19*3);
z3S11a11=x(16+19*3);
z3S11a11a11=x(17+19*3);
z3uS9=x(18+19*3);
z3uS11=x(19+19*3);

u49=x(1+19*4);
u410=x(2+19*4);
u411=x(3+19*4);
delta49=x(4+19*4);
delta410=x(5+19*4);
delta411=x(6+19*4);
delta4S9=x(7+19*4);
delta4S9a9=x(8+19*4);
delta4S11=x(9+19*4);
delta4S11a11=x(10+19*4);
z49=x(11+19*4);
z410=x(12+19*4);
z411=x(13+19*4);
z4S9a9=x(14+19*4);
z4S9a9a9=x(15+19*4);
z4S11a11=x(16+19*4);
z4S11a11a11=x(17+19*4);
z4uS9=x(18+19*4);
z4uS11=x(19+19*4);
```

```
end_states = eval(v1);
step_cost = eval(L2(1:19)*x(1:19));
action = [x(1);x(2);x(3)];
cp_time = info.Time;
% qu11 = [x(3);x(3+19);x(3+19*2);x(3+19*3);x(3+19*4)];
% qs11 = [(beta11*v11m*x(6+19*0)+beta11*x0(3)-x(3)-beta11*x(13))*(1-x(9))+qs11m*x(9),
%          (beta11*v11m*x(6+19*1)+eval(beta11*v1(3)-x(3+19*1))-beta11*x(13+19*1))*(1-
x(9+19*1))+qs11m*x(9+19*1),
%          (beta11*v11m*x(6+19*2)+eval(beta11*v2(3)-x(3+19*2))-beta11*x(13+19*2))*(1-
x(9+19*2))+qs11m*x(9+19*2),
%          (beta11*v11m*x(6+19*3)+eval(beta11*v3(3)-x(3+19*3))-beta11*x(13+19*3))*(1-
x(9+19*3))+qs11m*x(9+19*3),
%          (beta11*v11m*x(6+19*4)+eval(beta11*v4(3)-x(3+19*4))-beta11*x(13+19*4))*(1-
x(9+19*4))+qs11m*x(9+19*4)];

qu11 = [x(3+19);x(3+19*2);x(3+19*3);x(3+19*4);x(3+19*4);x(3)];
qs11 = [(beta11*v11m*x(6+19*1)+eval(beta11*v1(3)-x(3+19*1))-beta11*x(13+19*1))*(1-
x(9+19*1))+qs11m*x(9+19*1),
        (beta11*v11m*x(6+19*2)+eval(beta11*v2(3)-x(3+19*2))-beta11*x(13+19*2))*(1-
x(9+19*2))+qs11m*x(9+19*2),
        (beta11*v11m*x(6+19*3)+eval(beta11*v3(3)-x(3+19*3))-beta11*x(13+19*3))*(1-
x(9+19*3))+qs11m*x(9+19*3),
        (beta11*v11m*x(6+19*4)+eval(beta11*v4(3)-x(3+19*4))-beta11*x(13+19*4))*(1-
x(9+19*4))+qs11m*x(9+19*4),
        (beta11*v11m*x(6+19*4)+eval(beta11*v4(3)-x(3+19*4))-beta11*x(13+19*4))*(1-
x(9+19*4))+qs11m*x(9+19*4),
        (beta11*v11m*x(6+19*0)+beta11*x0(3)-x(3)-beta11*x(13))*(1-x(9))+qs11m*x(9)];

op_cost = eval(L2(1:3)*x(1:3));   %control cost
of_cost = eval(L2(4:19)*x(4:19));  %overflow cost

suc = exitflag;

end
```

Decentralized MPC subsystem 4

```
%% Large system DMPC subsystem 4
function
[end_states,step_cost,action,cp_time,qout13,op_cost,of_cost,suc]=sub4function(initial_states,rain
_profile,disturbance)
syms dt epsilon ...
        v12 v12u  v12m beta12 qu12 ...
        v13 v13u v13l v13m beta13 S13 phi13 delta13 deltaS13 deltaS13a13 z13 zS13a13 zS13a13a13
zuS13 qu13 ...
        qu12m ...
        m13 ...
        Ms4 ms4 M13 ...
        v012 v013 ...
        p012 p013 ...
        p112 p113 ...
        p212 p213 ...
        p312 p313 ...
        p412 p413 ...
        z012 z013 ...
        z112 z113 ...
        z212 z213 ...
        z312 z313 ...
        zuS4 ...
        delta0S4 delta1S4 delta2S4 delta3S4...
        delta012 delta013 ...
        delta112 delta113 ...
        delta212 delta213 ...
        delta312 delta313 ...
        z0S4a4 z1S4a4 z2S4a4 z3S4a4...
        u04 u012 ...
        u14 u112 ...
        u24 u212 ...
        u34 u312 ...
        qu4m qs4m...
        qu11 qu8 qout4 ...  % external disturbances

qout4 = disturbance(1,:);
qu8 = disturbance(2,:);
qs8 = disturbance(3,:);
qu11 =disturbance(4,:);
qs11 = disturbance(5,:);



A = sym(zeros(2,2));   %tank
B1 = sym(zeros(2,2));  %control
B2 = sym(zeros(2,3));   %delta
B3 = sym(zeros(2,3));   %z
B4= sym(zeros(2,2));    % disturbance

d0=[p012;p013];
d1=[p112;p113];
d2=[p212;p213];
d3=[p312;p313];
d4=[p412;p413];

z0=[z012;z013;z0S4a4];
z1=[z112;z113;z1S4a4];
z2=[z212;z213;z2S4a4];
z3=[z312;z313;z3S4a4];

delta0 = [delta012;delta013;delta0S4];
delta1 = [delta112;delta113;delta1S4];
delta2 = [delta212;delta213;delta2S4];
delta3 = [delta312;delta313;delta3S4];

u0 = [u04;u012];
u1 = [u14;u112];
```

```
u2 = [u24;u212];
u3 = [u34;u312];

v0=[v012;v013];

A(1,1)=1;
B1(1,1)=dt;
B1(1,2)=-dt;
B4(1,1)=dt;

A(2,2)=1-dt*beta13;
B1(2,1)=-dt;
B2(2,2)=(1-dt*beta13)*v13m;
B3(2,2)=dt*beta13-1;
B4(2,2)=dt;% phi4*S13; need to put this coefficient into rainprofile

v1=A*v0+B1*u0+B2*delta0+B3*z0+B4*d0;
v2=A*v1+B1*u1+B2*delta1+B3*z1+B4*d1;
v3=A*v2+B1*u2+B2*delta2+B3*z3+B4*d2;
v4=A*v3+B1*u3+B2*delta3+B3*z3+B4*d3;

%% constraints  E1
E1=sym(zeros(15,2)); %qu
E2=sym(zeros(15,2)); %delta
E3=sym(zeros(15,2)); %z
E4=sym(zeros(15,2)); %v
E5=sym(zeros(15,1));  %constants


E1(1,1)=-1;
E5(1,1)=qout4(1); %qu4

E1(2,1)=-dt;
E1(2,2)=dt;
E4(2,1)=-1;
E5(2,1)=v12m-dt*(qu11(1)+qu8(1));   %qu4

% E1(3,2)=-1;
% E4(3,1)=beta12; %qu12

E1(3,1)=dt;
E1(3,2)=-dt;
E4(3,1)=1;
E5(3,1)=(qu11(1)+qu8(1))*dt;   %qu12

E2(4,2)=-v13u;
E3(4,2)=1;

E3(5,2)=-1;

E3(6,2)=1;
E4(6,2)=1;

E2(7,2)=v13u;
E3(7,2)=-1;
E4(7,2)=-1;
E5(7,1)=v13u;  %z13

E2(8,3)=-qu4m;
E3(8,3)=1;

E3(9,3)=-1;

E3(10,3)=1;
E1(10,1)=1;

E2(11,3)=qu4m;
E3(11,3)=-1;
E1(11,1)=-1;
E5(11,1)=qu4m; %zuS4
```

```
E2(12,2)=-m13;
E4(12,2)=1;
E5(12,1)=-v13m-m13;

E2(13,2)=-(M13+epsilon);
E4(13,2)=-1;
E5(13,1)=v13m-epsilon;   %delta 13

E1(14,1)=-1;
E2(14,3)=-ms4;
E5(14,1)=qout4(1);

E2(15,3)=-(Ms4+epsilon);
E1(15,1)=1;
E5(15,1)=-epsilon-qout4(1)+qs4m;  %deltaS4

%% constraints  E11
E11=E1;
E21=E2;
E31=E3;
E41=E4;
E51=E5;

E51(1,1)=qout4(2); %qu4

E51(2,1)=v12m-dt*(qu11(2)+qu8(2));   %qu4

E51(3,1)=(qu11(2)+qu8(2))*dt;

E51(14,1)=qout4(2);

E51(15,1)=-eps-qout4(2)+qs4m;  %deltaS4
%% constraints  E12
E12=E1;
E22=E2;
E32=E3;
E42=E4;
E52=E5;

E52(1,1)=qout4(3); %qu4

E52(2,1)=v12m-dt*(qu11(3)+qu8(3));   %qu4

E52(3,1)=(qu11(3)+qu8(3))*dt;

E52(14,1)=qout4(3);

E52(15,1)=-eps-qout4(3)+qs4m;  %deltaS4

%% constraints  E13
E13=E1;
E23=E2;
E33=E3;
E43=E4;
E53=E5;

E53(1,1)=qout4(4); %qu4

E53(2,1)=v12m-dt*(qu11(4)+qu8(4));   %qu4

E53(3,1)=(qu11(4)+qu8(4))*dt;

E53(14,1)=qout4(4);

E53(15,1)=-eps-qout4(4)+qs4m;  %deltaS4
%% constraints  E14
E14=E1;
E24=E2;
E34=E3;
E44=E4;
E54=E5;
```

```
E54(1,1)=qout4(5); %qu4

E54(2,1)=v12m-dt*(qu11(5)+qu8(5));   %qu4

E54(3,1)=(qu11(5)+qu8(5))*dt;

E54(14,1)=qout4(5);

E54(15,1)=-eps-qout4(5)+qs4m;  %deltaS4
%% constraints integration
F11=sym(zeros(15,8*5));    %initialize
F21=sym(zeros(15,1));
F31=sym(zeros(15,2));
F11=[-E1,E2,E3,zeros(15,8*4)];
F12=[-E41*B1,-E41*B2,-E41*B3,-E11,E21,E31,zeros(15,8*3)];
F13=[-E42*A*B1,-E42*A*B2,-E42*A*B3,-E42*B1,-E42*B2,-E42*B3,-E12,E22,E32,zeros(15,8*2)];
F14=[-E43*A^2*B1,-E43*A^2*B2,-E43*A^2*B3,-E43*A*B1,-E43*A*B2,-E43*A*B3,-E43*B1,-E43*B2,-E43*B3,-
E13,E23,E33,zeros(15,8)];
F15=[-E44*A^3*B1,-E44*A^3*B2,-E44*A^3*B3,-E44*A^2*B1,-E44*A^2*B2,-E44*A^2*B3,-E44*A*B1,-
E44*A*B2,-E44*A*B3,-E44*B1,-E44*B2,-E44*B3,-E14,E24,E34];

F21=E5;
F22=E41*B4*d0+E51;
F23=E42*A*B4*d0+E42*B4*d1+E52;
F24=E43*A*A*B4*d0+E43*A*B4*d1+E43*B4*d2+E53;
F25=E44*A*A*A*B4*d0+E44*A*A*B4*d1+E44*A*B4*d2+E44*B4*d3+E54;

F31=E4*v0;
F32=E41*A*v0;
F33=E42*A*A*v0;
F34=E43*A*A*A*v0;
F35=E44*A*A*A*A*v0;

F1=[F11;F12;F13;F14;F15];
F2=[F21;F22;F23;F24;F25];
F3=[F31;F32;F33;F34;F35];




%% cost function 1
L2=sym(zeros(1,8*5));
operation_cost = 0.5;
L2(1,1)=operation_cost*dt ;
L2(1,2)=operation_cost*dt ;  %operation cost from manipulation flows

L2(1,7)=1;
L2(1,4)=-v13m;   %tank over flow

L2(1,5)=dt*(qout4(1)-qs4m);
L2(1,8)=-dt;         %qs4 over flow
%% cost function 2
L2(1,1+8)=operation_cost*dt ;
L2(1,2+8)=operation_cost*dt ;  %operation cost from manipulation flows

L2(1,7+8)=1;
L2(1,4+8)=-v13m;   %tank over flow

L2(1,5+8)=dt*(qout4(2)-qs4m);
L2(1,8+8)=-dt;         %qs4 over flow
%% cost function 3
L2(1,1+8*2)=operation_cost*dt ;
L2(1,2+8*2)=operation_cost*dt ;  %operation cost from manipulation flows

L2(1,7+8*2)=1;
L2(1,4+8*2)=-v13m;   %tank over flow

L2(1,5+8*2)=dt*(qout4(3)-qs4m);
L2(1,8+8*2)=-dt;         %qs4 over flow
%% cost function 4
```

```
L2(1,1+8*3)=operation_cost*dt ;
L2(1,2+8*3)=operation_cost*dt ;  %operation cost from manipulation flows

L2(1,7+8*3)=1;
L2(1,4+8*3)=-v13m;   %tank over flow

L2(1,5+8*3)=dt*(qout4(4)-qs4m);
L2(1,8+8*3)=-dt;         %qs4 over flow
%% cost function 5
L2(1,1+8*4)=operation_cost*dt ;
L2(1,2+8*4)=operation_cost*dt ;  %operation cost from manipulation flows

L2(1,7+8*4)=1;
L2(1,4+8*4)=-v13m;   %tank over flow

L2(1,5+8*4)=dt*(qout4(5)-qs4m);
L2(1,8+8*4)=-dt;         %qs4 over flow
%% constants specification
epsilon =1e-13;
S=[323576,164869,5076,754131];%surface area
phi=[1.03,10.4,0,0.48];%absorption coefficient
beta=[7.1e-4,5.8e-4,2e-3,1e-3];%volumetric flow coefficient
vm=[16901,43000,35000,26659];%maximum volume
dt=300;%sample time


% x0=[0,0];
x0 = initial_states ;


% rain = [0,0,0,0,0];
rain = rain_profile;

p = rain.*1.2./1000./3600;% precipitation in unit of m/s

v12m=vm(3);
v13m=vm(4);




v012=x0(1);
v013=x0(2);

S13=S(4);

phi4=phi(4);
phi13=phi(4);

beta12=beta(3);
beta13=beta(4);


p012=qu11(1)+qu8(1);%real
p013=p(1)*phi4*S13+qs8(1)+qs11(1)+qout4(1);

p112=qu11(2)+qu8(2);%real
p113=p(2)*phi4*S13+qs8(2)+qs11(2)+qout4(2);

p212=qu11(3)+qu8(3);%real
p213=p(3)*phi4*S13+qs8(3)+qs11(3)+qout4(3);

p312=qu11(4)+qu8(4);
p313=p(4)*phi4*S13+qs8(4)+qs11(4)+qout4(4);

p412=qu11(5)+qu8(5);
p413=p(5)*phi4*S13+qs8(5)+qs11(5)+qout4(5);

% qs4m=10;

qs4m=1000;
```

```
v13u=v13m*2+S13*dt*max(p)*phi13;
v12u=35000;

vu=[v12u,v13u];

qu4m=25;
qu12m=40;

M13=v13u-v13m;
m13=-v13m;

Ms4 = 16.659;
ms4 = -qs4m;

lb = zeros(8*5,1);                  %Bounds on x (lb <= x)
uub = [qu4m;qu12m];         %u upper bound

dub = ones(3,1);                 %delta upperbound
zub = zeros(3,1);            %z upper bound
for i=1:3
    zub(i,1)=inf;
end

ub = [uub;dub;zub;uub;dub;zub;uub;dub;zub;uub;dub;zub;uub;dub;zub];

f=eval(L2');


a=eval(F1);
b=eval(F2+F3);
%%
xtype = 'CCBBBCCCCCBBBCCCCCBBBCCCCCBBBCCCCCBBBCCC';

Opt = opti('f',f,'ineq',a,b,'bounds',lb,ub,'xtype',xtype);
[x,fval,exitflag,info] = solve(Opt);

u04=x(1);
u012=x(2);
delta012=x(3);
delta013=x(4);
z012=x(5);
z013=x(6);

u14=x(1+6);
u112=x(2+6);
delta112=x(3+6);
delta113=x(4+6);
z112=x(5+6);
z113=x(6+6);

u24=x(1+6*2);
u212=x(2+6*2);
delta212=x(3+6*2);
delta213=x(4+6*2);
z212=x(5+6*2);
z213=x(6+6*2);

u34=x(1+6*3);
u312=x(2+6*3);
delta312=x(3+6*3);
delta313=x(4+6*3);
z312=x(5+6*3);
z313=x(6+6*3);

u44=x(1+6*4);
u412=x(2+6*4);
delta412=x(3+6*4);
delta413=x(4+6*4);
z412=x(5+6*4);
z413=x(6+6*4);
```

```
end_states = eval(v1);
step_cost = eval(L2(1:8)*x(1:8));
action=[x(1);x(2)];
cp_time = info.Time;
% qout13 = [beta13*v13m*x(4)+beta13*x0(2)-beta13*x(6),
%            beta13*v13m*x(4+6*1)+eval(beta13*v1(2))-beta13*x(6+6*1),
%            beta13*v13m*x(4+6*2)+eval(beta13*v2(2))-beta13*x(6+6*2),
%            beta13*v13m*x(4+6*3)+eval(beta13*v3(2))-beta13*x(6+6*3),
%            beta13*v13m*x(4+6*4)+eval(beta13*v4(2))-beta13*x(6+6*4)];

qout13 = [beta13*v13m*x(4+6*1)+eval(beta13*v1(2))-beta13*x(6+6*1),
          beta13*v13m*x(4+6*2)+eval(beta13*v2(2))-beta13*x(6+6*2),
          beta13*v13m*x(4+6*3)+eval(beta13*v3(2))-beta13*x(6+6*3),
          beta13*v13m*x(4+6*4)+eval(beta13*v4(2))-beta13*x(6+6*4),
          beta13*v13m*x(4+6*4)+eval(beta13*v4(2))-beta13*x(6+6*4),
          beta13*v13m*x(4)+beta13*x0(2)-beta13*x(6)];

op_cost =eval(L2(1:2)*x(1:2));   %control cost
of_cost =eval(L2(3:8)*x(3:8));  %overflow cost

suc = exitflag;

end
```