OPTIMIZING MODEL PREDICTIVE CONTROL OF PROCESSES FOR WIDE RANGES OF OPERATING CONDITIONS

by

Vu Nam Tran

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

Department of Electronic, Electrical and
Computer Engineering
School of Engineering
The University of Birmingham

May 2011

Synopsis

This thesis develops robustly feasible model predictive controllers (RFMPC) for nonlinear network systems and soft switching mechanism between RFMPCs is proposed to achieve softly switched RFMPC (SSRFMPC).

The safety zones based technique is utilized to design RFMPC by two different mechanisms i.e. iterated safety zones or explicit safety zones. Although the former one is calculated on-line by the relaxation algorithm and its RFMPC achieve robust feasibility, the recursive robust feasibility is not guaranteed. In contrast to the former, the latter one is calculated off-line and its RFMPC achieves recursive robust feasibility. In addition to this, the robustly feasible invariant sets in the state space are calculated off-line and the initial states need to stay inside those invariant sets in order to achieve feasible control operation.

The computation of RFMPC is very demanding and computing time is reduced by several methods. First, the more efficient optimization solver which is gradient type solver is used to solve the optimization task. The method to provide suitable gradients of objective function and derivatives of constraints to the optimization solver is presented. The robust output prediction is approximated and its horizon is also shortened. The optimization task is formulated in the reduced space of decision variables which is used in the implementation.

The proposed methodology is verified by applying to a simulated drinking water distribution systems example. Comparative simulation results are presented and discussed.

To my parents

Acknowledgments

Most of all, I would like to thank my supervisor Prof. Tech. Sci. Mietek A. Brdys for his advice, assistance, and encouragement. Without his continuous support especially for his patient guidance and help in putting my scattered thoughts in the right direction, this work would never have been presented here. During my three years, apart from the many unforgettable academic discussion, he has also shared with me his valuable knowledge and experiences. For all of his generosity and help, I am utmost grateful.

I would also like to express my thanks to all of my friends and the members of the research group for their support and help throughout the journey and making my time in Birmingham enjoyable.

My special thanks go to my grand parents, my sister, and all other relative members. Their unconditional support is always the motivating strength in my life. Finally I am deeply grateful to my parent for their constant love, always being by my side, and their emphasis on the importance of education. To them, I dedicate this work.

Contents

List of figures

List of tables

Notations

Lower case symbols

b	Total number of branches in the network
c_{ij}	Constant based on an estimate of the level of leakage
d	Nodal demand vector
d_j	Water demand allocated to the j th junction
$d_{r,i}$	Demand flow allocated to the i th reservoir node
$d(\cdot \mid k)$	The vector of predicted disturbances over the prediction horizon
$d^{\min}$	Lower limit of disturbance
$d^{\max}$	Upper limit of disturbance
h_i	Hydraulic head of node i
Δh_{ij}	Difference between hydraulic head of node i and j
h_{si}	Suction head of the i th pump
h_{di}	Delivery head of the i th pump
$h_{ri}(t)$	Reservoir head at time instant t
k_s	Switching time instants which denotes the time when the switching starts
lq_i	Rate of leakage at the ith node
n	Total number of nodes in the network
n_r	Total number of reservoir nodes in the network
n_c	Total number of junction nodes in the network

q_{ij}	flow from node i to node j
$g^{f}(q_{ij})$	Fixed-speed pump hydraulic characteristic curve
$g_{ij}^{s}(q_{ij}, n_{ij}, s_{ij})$	Variable-speed pump hydraulic characteristic curve
p_i	The water pressure at node *i*.
$q_{s,i}$	Water delivery from treatment works
$q_{out,i}$	Disturbance flow different from $q_{s,i}$
$q_{r,i}(t)$	Total flow into the reservoir from the network
s	Vector of all states, inputs, and outputs
u	Input
$u^{\min}$	Lower limit of input
$u^{\max}$	Upper limit of input
$u_{\cdot\mid k}$	Predicted input sequence at time k
$u(\cdot \mid k)$	Vectors of predicted control input over the horizon
$u^{opt}(\cdot \mid k)$	Optimal input sequence
$w_i(t)$	Mass of water stored in the reservoir
x	State
$x^{\min}$	Lower limit of state
$x^{\max}$	Upper limit of state
$x_i(t)$	ith reservoir level at time instant t
$x(\cdot \mid k)$	Vector of predicted states over the prediction horizon
$x_{md,m}^{\min}$, $x_{md,m}^{\max}$	The modified of m th component of state constraint limits

y	Output
$y^{\min}$	Lower limit of output
$y^{\max}$	Upper limit of output
$y_{\cdot\mid k}$	Predicted output sequence at time k
$y(\cdot \mid k)$	Vector of predicted outputs over the prediction horizon
$y_p^l(k+i \mid k)$	Lower limits that robustly bound plant output at prediction time step i
$y_p^u(k+i \mid k)$	Upper limits that robustly bound plant output at prediction time step i
$y_{md}^{\min}$, $y_{md}^{\max}$	Vector of modified output constraint limits

Upper case symbols

$\overline{A}_{ij}$, $\overline{B}_{ij}$, $\overline{C}_{ij}$	Resistance coefficients
C_{ij}	Hazen-Williams roughness coefficients
C_E	Leakage coefficient
D_{ij}	Pipe diameter
E	The operator relates the state, control input, and corresponding output
E_i	Reservoir elevation
∂E_r	Head difference between the starting and final nodes of the *r*-th path
F	The operator represents the static part of the network model
H_p	Prediction horizon
H_c	Control horizon
I	Number of pump stations
J_j	Set of nodes linked with the node j

J_p	The cost of pumping energy
J_T	The cost of treatment
J_{MDC}	the cost of maximum demand charge
$J(\cdot)$	Objective function in the reduced space of decision variables
$\widetilde{J}(\cdot)$	Objective function in the full space of decision variables
K	Number of time steps
L_{ij}	Pipe length
$L_{E_m,d}$	Lipschitz constant of m th component $E_m(\cdot)$ with respect to $d(k)$
$L_{f_m,d}$	Lipschitz constant of m th component $f_m(\cdot)$ with respect to $d(k)$
M	Sets of non-leaky nodes
M_l	Sets of leaky nodes
M_i^f	Number of fixed speed pumps contained in a pump station
M_i^s	Number of variable speed pumps contained in a pump station
M_i^t	Number of variable throttle pumps contained in a pump station
Q_{ij}^l	Rate of leakage between nodes i and j
$S_i(x_i(t))$	Reservoir cross-sectional area
T_s	Switching time duration
$\mathbf{U}$	Constraint set of input
$\mathbf{U}_{combined}$	Convex combined constraint set of input
$\mathbf{U}_{old}$, $\mathbf{U}_{new}$	Constraint set for input of old and new controller respectively
V_{ij}	Controlled value of valve

$\mathbf{X}$	Constraint set of state
$\mathbf{X}_{combined}$	Convex combined constraint set of state
$\mathbf{X}_f(k)$	Composed of all feasible initial states
$\mathbf{X}_{fa}(k)$	Approximation of $\mathbf{X}_f(k)$
$\mathbf{X}_{Rf}(k)$	Composed of the one-step robustly feasible states
$\mathbf{X}_{rRf}(k)$	Robustly feasible invariant set of states
$\mathbf{X}_{old}$, $\mathbf{X}_{new}$	Constraint set of state for old and new controller respectively
$\mathbf{X}_{rRf}^{combinedRFMPC(i)}$	Robustly feasible invariant set of i th combined RFMPC
$\mathbf{Y}$	Constraint set of output
$\mathbf{Y}_{combined}$	Convex combined constraint set of output
$\mathbf{Y}_{old}$, $\mathbf{Y}_{new}$	Constraint set of output for old and new controller respectively

Greek and mathematical symbols

ρ	Density of water
σ^l, σ^u	Safety zones for lower and upper bounds
γ	Leakage exponent
λ	Tuning knob parameter
Λ	Incidence matrix
Λ_c	Junction nodes incidence matrix
Λ_r	Incidence matrix defined for the reservoir nodes
$\Psi(q)$	Vector of function defining flow-head relationship for branches
ξ_n , ξ_d , ξ_w	Chosen weights

Υ	Control law
$\gamma_1^*, \gamma_2^*, ..., \gamma_{NM}^*$	Lagrange multipliers
ε_m^u , ε_m^l	Lower and upper safety zones of the state constraints respectively
ε_{ym}^u , ε_{ym}^l	Lower and upper safety zones of the output constraints respectively
$\varnothing$	Empty set
$\overline{0:N-1}$	Integer variable changing from 0 to N-1

Abbreviations

DWDS	Drinking water distribution systems
MPC	Model predictive control
FuCL	Follow up control level
GA	Genetic Algorithm
KKT	Karush Kuhn Tucker
LCROP	Least conservative robust output prediction
OCL	Optimizing control level
RFMPC	Robustly feasible model predictive control
RHC	Receding horizon control
ROP	Robust output prediction
SSRFMPC	Softly switched robustly feasible model predictive control
SuCL	Supervisory control level
SQP	Sequential Quadratic Programming
SWROP	Stepwise robust output prediction

Chapter 1

Introduction

1.1 Motivation

The objective of automatic control in general is to influence the behaviour of a given physical plant in some ways such that the response conforms to some desired specification. The plants in reality are classified by different categories e.g. nonlinear or linear systems, constrained system or unconstrained systems, small scale or large scale systems. The treatment and analysis to design linear systems has been well developed and can be found in (Ogata, 2005). Unfortunately, almost of the plants in reality are nonlinear, multivariables, and subject to physical constraints. It therefore requires different techniques and treatments to properly design and handle the nonlinear constrained systems. Model Predictive Control has been widely known in the control community due to its capability in dealing with multivariable constrained and nonlinear problems, since constraints can be directly incorporated in the optimization problems. The MPC in principle is implemented by

repetitively solving on-line over the plant output prediction horizon the corresponding MPC optimization task and applying the first part of the produced control sequence into the plant. The initial state of the MPC is updated at every time step by taking measured output/state as the information feedback.

With the increasing demand of using MPC, it is required that the MPC needs to be designed for highly nonlinear systems under input and state/output constraints. Moreover, there are always disturbances that need to be considered. The disturbances consist of the mismatch between mathematical model of MPC and the physical plants due to the complexity of the physical world and the difference between predicted disturbance inputs used in the model and real disturbance inputs. The model-reality mismatch needs to be incorporated in designing MPC. The MPC which can take disturbances into consideration is then said to be robustly feasible. It is worth remarking that the Robustly Feasible MPC (RFMPC) needs to be designed to achieve the robust feasibility not only for one or two time step but also for as long as the control duration time is.

A single RFMPC represents for a single control strategy and is determined by a performance and certain optimization constraints which reflect the systems dynamics and operational conditions. The constrained plant under full range of operational conditions usually requires meeting several sets of objectives. It is impossible to achieve all control objectives by a single RFMPC controller. It thus calls for application of multiple RFMPC controllers each of them being best fit into specific operational conditions. It is then inevitable to switch between controllers during the plant operation. The overall RFMPC is then called Softly Switched RFMPC. For example the considered drinking water

distribution system in the thesis, three typical control strategies are pumping cost minimization control strategy, excessive pressure minimization strategy, and leakage minimization control strategy. Depending on the operational condition, the most suitable control strategy is selected to apply to the plant. Under the full range of operational conditions, the SSRFMPC is applied to the water system.

1.2 Aims of the study

The main aims of the thesis are explained as follows:

1. The RFMPC is designed by the iterated safety zone based technique. As the output/states of the plant is subject to the constraints. Model-reality mismatch is the reason to cause the violation of those constraints. The output/state constraints of the model are then tightened by so-called safety zones. By introducing the safety zones, the output/state constraint satisfaction is guaranteed.

2. The RFMPC requires tremendous computing efforts as the MPC optimization task is repetitively solved several times online during one control time step to yield the control input. Therefore reducing the computing burden of the RFMPC needs to be investigated.

3. The feasibility of the RFMPC is determined by the existence of safety zones due to the fact that the RFMPC is constructed by safety zones based technique. Every RFMPC by nature has its own feasible region. If the initial state lies outside the feasible region, then

the safety zones are never found and the RFMPC becomes infeasible. It is important to investigate that under what condition the safety zones exist.

4. The RFMPC is needed to be re-designed such that the robust feasibility is achieved not only one step but multiple time steps. In other words, the RFMPC needs to achieve recursive robust feasibly in order to have proper operational control online. The safety zones based technique is still applicable but needs to be treated differently.

5. One RFMPC represents for only one control objectives. However, the operational control of the plant often requires meeting a set of control objectives. Therefore multiple RFMPC controllers are needed for the optimal operation. It is then inevitable to switch between controllers. Although the hard switching technique is very simple, it does not usually achieve satisfactory switching outcome due to the sudden switching manner. It is necessary to investigate the soft switching technique in order to smooth and soften the switching transients.

6. In operational control of drinking water distribution systems (DWDS), a good system model that can represent the hydraulic behaviour of the real water network is very crucial for model-based controller design. However, the highly nonlinear models of water network components, e.g. pipes, valves, and pumps, together with hydraulic laws constitute a very complex nonlinear model of DWDS. The optimizing control system is needed for the DWDS and the RFMPC is the most suitable the control technology to design such systems.

1.3 Achievements

The main idea of this thesis was to analyze related issues of RFMPC controller and develop a soft switching mechanism between RFMPC controllers and eventually achieved the Softly Switched RFMPC (SSRFMPC) that can be applied to the daily operational control of the DWDS. The main contributions of this thesis are summarized below:

1. The RFMPC has been designed by the iterated safety zone based technique. Two types of robust output prediction (ROP) i.e. least conservative ROP stepwise and stepwise ROP has been proposed. The robust prediction horizon was shortened to reduce computing time.

2. The computing issue of the RFMPC has been investigated. Two different optimization solvers i.e. Genetic Algorithm (GA) and Sequential Quadratic Programming (SQP) have been used in the design of RFMPC in order to compare the computing efficiency. Using SQP requires users to supply the gradients of objective functions and derivatives of constraints. The Hamiltonian based technique has been proposed to calculate those gradients and derivatives.

3. The feasibility of the RFMPC has been achieved. The Karush-Kuhn-Tucker (KKT) optimality condition has been utilized in order to achieve the approximated region of initial states that guarantees the existence of safety zones.

4. RFMPC is then designed such that it achieves recursive robust feasibility. The explicit safety zones based technique has been proposed design such RFMPC. Comparison

between the explicit safety zones and iterated safety zones based techniques also has been made.

5. The architecture of SSRFMPC has been considered and the soft switching mechanism has been proposed. The fast soft switching algorithm has been proposed for the situation of the hard switching is infeasible.

6. The SSRFMPC has been applied to the optimal operation of DWDS under wide range of operational conditions. The switching directions between the three typical control strategies of DWDS have been proposed. Comparative simulation results have been made.

The papers that have been published or will soon be published are listed below:

- Tran, V.N and Brdys, M.A (2009). "Optimizing Control by Robustly Feasible Model Predictive Control and Application to Drinking Water Distribution Systems". In *Proceedings of 19th International Conference on Artificial Neural Networks (ICANN)*, Limassol, Cyprus, September 2009.

- Tran, V.N and Brdys, M.A (2011). "Optimizing Control by Robustly Feasible Model Predictive Control and Application to Drinking Water Distribution Systems" Journal of Artificial Intelligence and Soft Computing Research, 2011, Vol. 1, No. 1, pp. 43-57.

- Tran, V.N and Brdys, M.A (2010). "Robustly Feasible Optimizing Control of Network Systems under Uncertainty and Application to Drinking Water Distribution Systems". In *Proceedings of 12th IFAC symposium on Large Scale Complex System: Theory and Application*, Lille, France, July 2010.

- Brdys, M.A. and Tran, V.N (2011) Safety zones based robustly feasible model predictive control for nonlinear network systems. In *Proceedings of 18th IFAC World Congress,* Milan, Italy, August 2011 (accepted).

1.4 Organization of this thesis

The thesis is organized as follows:

Chapter 2: Presentation of Drinking Water Distribution Systems (DWDS) operating under wide range of operational conditions

This chapter briefly presents the presentation of the Drinking Water Distribution Systems (DWDS) and its operational conditions. The physical modelling of the DWDS components is also presented. The mathematical modelling of water leakage is introduced. Three main operational states have been distinguished and the corresponding control strategies are formulated. Mathematical formulations of control strategies, nodal model, and part of the hydraulic modelling equations have been presented in (Tran and Brdys, 2009), (Tran and Brdys 2010).

Chapter 3: Iterated Safety Zones Based Robustly Feasible Model Predictive Control and application to DWDS

This chapter considers iterated safety zones robustly feasible MPC. The robustly feasible MPC architecture and its components are presented. The computing effort of RFMPC is reduced by applying the stepwise robust output prediction (SWROP) technique and shortening robust feasible horizon. In order to use SQP as the optimization solver of the robustly feasible MPC, the Hamiltonian based technique has been considered to calculate the gradients of objective functions and derivative of constraints. RFMPC is applied to the DWDS example and illustrating simulation examples have been made. The main content is this chapter including SWROP, Hamilton based technique, and part of simulations have been presented in (Tran and Brdys, 2009), (Tran and Brdys, 2010),(Tran and Brdys, 2011).

Chapter 4 Explicit Safety Zones Based Robustly Feasible Model Predictive Control and application to DWDS

This chapter investigates feasibility, robust feasibility, and recursive robust feasibility issues of the safety zones based RFMPC. Utilizing Karush-Kuhn-Tucker optimality condition the safety zones mechanism is further developed to achieve robust feasibility. Moreover, in order to attain both recursive and robust feasibility a dedicated iterative algorithm has been proposed. The control algorithm is applied to control hydraulics in a DWDS example and simulation results are presented. The idea of using KKT optimality conditions to approximate feasible region, the algorithm to achieve recursive robust feasibility and simulation results have been presented in (Brdys and Tran, 2011)

Chapter 5 Softly Switched Robustly Feasible Model Predictive Control (SSRFMPC) and application to DWDS under wide range of operational conditions

This chapter presents Softly Switched RFMPC and its components. The soft switching mechanism is presented and the feasibility of the hard switching is discussed. The functionalities of supervisory control level and its importance with respect to switching are briefly discussed.

Chapter 6 Synthesis, analysis, and soft switching mechanisms of SSRFMPC and application to DWDS under wide range of operational conditions

This chapter presents the analysis of Softly Switched RFMPC. The switching mechanism of SSRFMPC is mathematically formulated. The algorithm for the fast soft switching between RFMPC is proposed in order to minimize the switching time duration. The SSRFMPC has been applied to DWDS example. Comparative simulation results have been made.

Chapter 7 Conclusion and future works

This chapter contain summarising remarks and conclusions of the thesis. Some limitations and drawbacks are also discussed in this chapter in order to suggest the further research.

The thesis also contains four appendices which are organised in sequence of relation with the main presentation.

The relationship between chapters of this thesis is illustrated in Figure 1.1

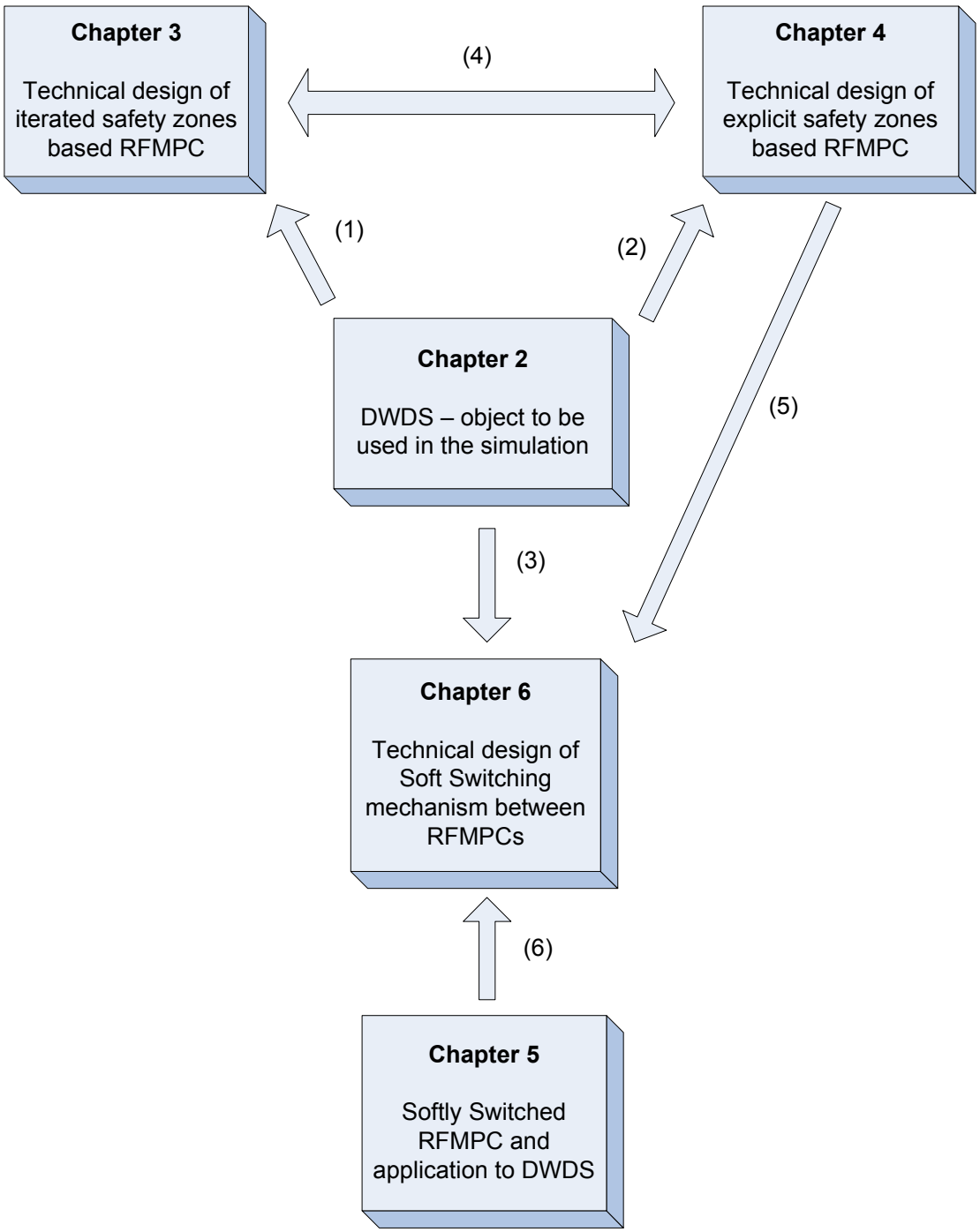


Figure 1.1 Relationships between chapters

The chapter 2 considers the DWDS presentation which is the object to be used in the simulation in chapter 3, chapter 4 and chapter 6. These relationships are presented by the arrow (1), (2), and (3) in the Figure 1.1

The chapter 3 and chapter 4 both consider safety zones based technique to design Robustly Feasible MPC. This relationship is presented by the arrow (4) in the Figure 1.1.

The common part of the chapter 5 and chapter 6 is about the Softly Switched RFMPC and is presented by the arrow (6) in the Figure 1.1. While the SSRFMPC structure in general is considered in the chapter 5, the technical design in detail is presented in chapter 6.

The SSRFMPC in the chapter 6 composes of several single RFMPCs. Each of these RFMPC has the same designed structure as described in the chapter 4. This relationship between the chapter 4 and chapter 6 is presented by arrow (5) in the Figure 1.1.

Chapter 2

Presentation of Drinking Water Distribution Systems Operating Under Wide Range of Operational Conditions

This chapter briefly presents the presentation of the Drinking Water Distribution Systems (DWDS) and its operational conditions. The physical modelling of the DWDS components is also presented. The mathematical modelling of water leakage is introduced. Three main operational states have been distinguished and the corresponding control strategies are formulated.

2.1 Introduction

The main interest of this chapter is on the drinking water distribution systems, which consists of a collection of hydraulic elements i.e. pipes, pumps, valves and local reservoirs/tanks to form a large scale network. All those hydraulic components in the DWDS are connected together to transport the water to the consumer taps (demand nodes). In order to provide water to consumers meeting the quality and quantity requirements, operational control of DWDS is performed. In the thesis, only the quantity control of the DWDS is considered. For the operational control purposes of DWDS, the mathematical modelling of the necessary components of DWDS is shown. Furthermore, nodal model of DWDS and models of optimization for operational control of DWDS are further proposed in order to achieve further analysis.

The DWDS presentation and its operational condition are briefly introduced in the Section 2.2. For the control purposes, the relations between hydraulic components of DWDS and the physical laws governing the hydraulics are presented in the Section 2.3. The mathematical modelling of water leakage will be presented in the Section 2.4. In order to reduce the computational burden of MPC, the nodal model presented in the Section 2.5 and is utilized in the Hamiltonian based technique as demonstrated in Chapter 3. In order to have the comparative simulation results in next chapters, the optimization model for operational control of DWDS is presented in the section 2.6. The Section 2.7 summarizes this chapter.

2.2 DWDS and its operational conditions

Drinking Water Distribution Systems are vitally inevitable in human lives as it supplies clean water to industrial and domestic users. It has been described in (Brdys and Ulanicki, 1994) that the DWDS typically consists of three main parts which shown in Figure 2.1:

- Treatment works
- Supply network of trunk mains and main reservoirs
- Distribution network of small diameter pipes and local reservoirs

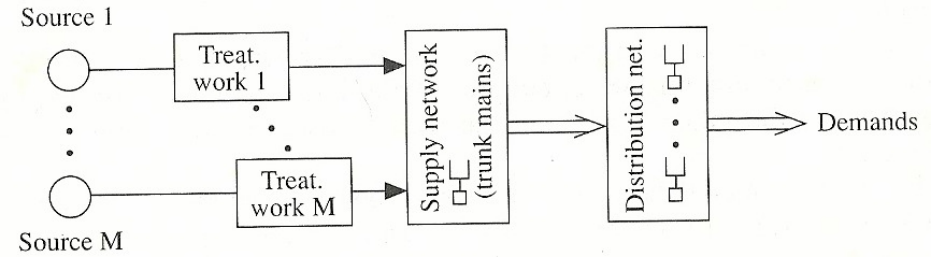


Figure 2.1 Physical structure of water supply/distribution system

The sources are points of interaction between the water supply and water retention systems. The water retention systems consist of a number of reservoirs, together with the rivers on which they are built in purpose of ensuring the continuity of water supplies. Drinking water is usually taken from ground sources such as rivers and lakes, or underground sources such as wells and springs. Source water is treated in the water plant to filter out unwanted substances using physical and chemical methods thereby making it safe, clean and healthy for people to consume. The treated drinkable water from the water plant is then transported to the supply network. The interaction between the supply part and distribution part of the system occur where water is pumped from the supply network

(trunk mains) to distribution network. The drinking water is then delivered to consumer taps by drinking network. DWDS as such is considered as large-scale network systems composed of sources, treatment works, trunk mains, pumps, valves, storage tanks and user taps connected together by pipes of various diameters made from variety of materials.

In order to provide high quality water to consumers to fulfil the water quantity and pressure demand, operational control of DWDS is performed. The purpose of the operational control of DWDS is to satisfy the time-varying water demands, maintain a stable prescribed water pressure throughout the network and minimize the operating costs. This is achieved by operating the controllable components in the network subjecting to the operational constraints. The controllable components are pumps, valves and chemical injections such as chlorine. Constraints mainly include physical laws governing the hydraulics such as relations between flows and heads, operating limitation of the controlled components and control objectives. Many of the relations are described by non-linear equations. From control viewpoint, operational control of DWDS is a constrained non-linear optimal control problem including both the real and integer variables. In the quality control of DWDS, the chlorine concentration is controlled as above the allowed level cause serious dangers to users. This topic was addressed in (Brdys and Chang 2002), (Jonkergouw P.M.R *et al*., 2004), (Grzegorz Ewald *et al*., 2008). As stated in the Chapter 1, the thesis only focuses on the quantity operational control of the DWDS.

There are two issues that need to be considered in the operational control of the DWDS. One is, of course, to keep the plant running. In other words, the goal is to satisfy the physical requirement i.e. limitations or constraints of the plant outputs, measured states,

and controlled inputs during operation. Specifically, the plant outputs (water levels of the tanks or/and nodal pressures) needs to be controlled within the permitted ranges to avoid overflow or/and prevent pipe burst. The states and inputs are also subject to constraints due to similar reasons. In operational control of DWDS terminologies, the legal and compulsory requirements are also called core control objective.

The second one is concerned with the objectives that are desired by the plant operator. While the former, which consists of all legal and compulsory requirements of the DWDS plant, is referred as a core objective, the latter is also called as secondary objectives. While there is only one core objective, there could be many secondary objectives due to its idea of giving the plant operator the chance to improve the plant operation. In the operational control of the DWDS, achieving operational objectives is understood as achieving both core objective and secondary objectives.

It is essential in operational control of DWDS to secure the quality and quantity of delivered water in achieving the operational objectives under wide range of operational conditions. The phrase ‘operational conditions’ in the context of the DWDS can be understood as the disturbance scenarios i.e. customer demands which vary over time and create the overflow or pipe burst, or accidental seriously unfavourable events such as sensor and/or actuator faults, failures of communication links or anomalies occurring in technological operation.

Although only quantity operational control is considered here, it is still a huge challenge to take all operational conditions into consideration. There has not been any known generic

and mature solution to the best knowledge of the author. It shall be assumed that the disturbance scenario, i.e. customer demands is considered as the main impact on the operational conditions in the thesis.

Hence, in order to best adopt the control actions to the actual and predicted conditions, different operational states of a process were distinguished in (Brdys et al., 2002); (Grochowski et al., 2004). The operational states are determined in predictive manner by assessing possibility of fulfilling the core control objective.. Typically, there are three main operational states of the plant: *normal, disturbed,* and *emergency*.

- If there is a guarantee of achieving all the core control objectives by running the plant inside preferred operating region, then the process is said to be in a *normal operational state.* The preferred operating region is understood as the region where the operational state is most welcome. When the operational state is not inside the preferred region means that some of the state variables have values that can lead to undesirable consequences for the plant in the near future.

- If there is no possibility of achieving all core control objectives over considered time horizon even if the plant operates in not preferred region, then the process is said to be in an *emergency operational state*.

- Finally, if there is no guarantee of achieving core control objectives over considered time horizon without entering the not preferred operating region, then the process is said to be in a *disturbed operational state*.

With the introduced operational states, there are corresponding control strategies. In the application of the DWDS, the typical operational states are pressures or flows. When the operational states is being *normal* means the pressure is within the preferred region, and the pumping cost minimization control strategy is applied. However, when the operational is being *disturbed* means that the pressure is not in the preferred region and keeping the pressure for long time easily causes pipe burst. Hence, the pumping cost minimization control strategy is no longer welcome and the excessive pressure minimization control strategy is applied. In the event of the pipe burst, the leakage minimization control strategy is applied to achieve the least amount of water loss. The mathematical formulation of all three control strategies are presented in the Section 2.6.

2.3 Physical Modelling of DWDS

In order to derive specialized models for the optimization and simulation, models of the network elements are presented in this section. In general, overall mathematical model of the network can be put into either nodal or loop form depending on its independent variables. Although the mixed nodal – loop form can be derived, only the nodal model of the network is utilized and used by the optimization solvers in the thesis. Systems approach to modelling and operational control of water distribution systems has been presented in the book (Brdys and Ulanicki, 1994). The hydraulic laws in DWDS described here are helpful in analysing the uncertainty sources in the quantity control. The details can be found in the above book and in e.g. (Coulbeck, 1988; Chen, 1997; Haestad Methods, 1997;

Brdys, Arnold, Puta and *et al.*, 1999ab; Rossman, 2000). In spite of the fact the drinking water is the main substance to be considered here, it still contains a mixture of isotopes and possibly of several polymers. However, in regard to the quantity control problem of DWDS, their influence is negligible. Hence, the following assumptions are made in the following modelling, which are commonly applied in operational control of DWDS.

1. The inertia effect of the water in the pipe is neglected.
2. Water is treated as incompressible fluid.
3. Constant temperature and air pressure within the DWDS.
4. Constant density and viscosity.

2.3.1 Pipes

A pipe transports water from the higher head node to the lower head node. The water flow direction may change due to the head change in the pipe nodes. Pipe head-flow relationship can be expressed as (Brdys and Ulanicki, 1994):

$$q_{ij} = \phi_{ij}(\Delta h_{ij}) = \phi_{ij}(h_i - h_j) \tag{2.1}$$

where q_{ij} denotes the flow from node i to node j. Note that a positive flow direction is from a node of higher head to a node of lower head.

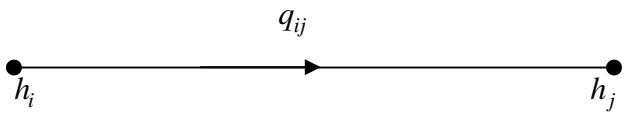


Figure 2.2 Model of a pipe

Popular methods to model hydraulic head loss by water flowing in a pipe due to friction with the pipe walls include the following three formulas (Rossman 2000):

- Hazen-Williams formula
- Darcy-Weisbach formula
- Chezy-Manning formula

The Hazen-Williams formula is the most commonly used pipe headloss formula in modelling water distribution systems. The Darcy-Weisbach formula is the most theoretically correct. It applies over all flow regimes and to all kinds of liquids. The Chezy-Manning formula is more commonly used for open channel flow. Because of Hazen-Williams formula's simplicity and calculating accuracy that can satisfy operational control purposes, it is used in the thesis.

According to the Hazen-Williams formula, pipe head drop - flow equations are written as:

$$\begin{cases} \Delta h_{ij} = h_i - h_j = R_{ij} q_{ij} \mid q_{ij} \mid^{0.852} \\ R_{ij} = \dfrac{1.21216 \times 10^{10} \times L_{ij}}{C_{ij}^{1.852} \times D_{ij}^{4.87}} \end{cases} \tag{2.2}$$

where L_{ij}, D_{ij} and C_{ij} denote the pipe length, pipe diameter and Hazen-Williams roughness coefficients, respectively. If the pipe length and diameter are in m and mm, respectively, and the heads are in m, then the flow in (2.2) are in $litre/\sec$. It needs to be pointed out that the roughness coefficient C_{ij} changes considerably with pipe age,

manufacturer, and some other factors. This parameter has to be calibrated regularly in order to establish an accurate hydraulic model.

2.3.2 Valves

There are many kinds of valves in drinking water distribution system performing varieties of functions (Chen, 1997):

- Check Valves: Control the flow in one direction only
- Flow Control Valves: Limit the flow rate at a specified value
- Pressure Reducing Valves: Reduce water pressure
- Pressure Sustaining Valves: Maintain the pressure to some value
- Pressure Breaker Valves: Create specified head loss across the valve
- Throttle Control Valves: Head loss characteristics change with time

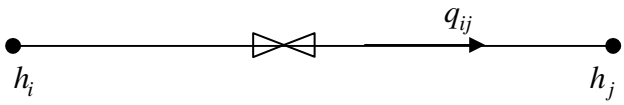


Figure 2.3 Model of Flow Control Valve

As a modelling example, the flow control valve is illustrated in Figure 2.3 (Brdys and Ulanicki, 1994). The flow control valve is a kind of variable valves that can be modelled as a pipe with controlled conductivity, that is:

$$q_{ij} = V_{ij} G_{ij} (h_i - h_j) \left| h_i - h_j \right|^{-0.46} \tag{2.3}$$

where V_{ij} denotes the controlled value and the valve is closed if $V_{ij} = 0$, and fully opened if $V_{ij} = 1$.

2.3.3 Pumps

Pumps are important control components in water distribution systems that add energy into the system by increasing the hydraulic head of the water. They provide water supply from the sources to the pipe network and act as a pressure booster device to lift up pressures at some points within the system. Pumps are typically electrically driven. They constitute the main energy consumption part in a water distribution network. The following types of pumps are considered: *fixed speed pumps* (FSP), *variable speed pumps* (VSP) and *variable throttle pumps* (VTP) (Brdys and Ulanicki, 1994; Chen, 1997), all of which are described by highly nonlinear functions. In the thesis, for operational control purposes of DWDS only the variable speed pumps is considered and its time-varying speeds are the control inputs of the DWDS.

2.3.3.1 Fixed Speed Pumps

The head-flow relationship for a fixed speed pump with the suction node *i* and delivery node *j* is a nonlinear function. It is also called *pump hydraulic characteristic curve*, which can be expressed as:

$$\Delta h_{ji} = g^f(q_{ij}) \tag{2.4}$$

where $\Delta h_{ji} = h_j - h_i$ and $h_j \geq h_i$. h_i and h_j are the *suction head* and *delivery head*, respectively. The superscript f in $g^f(q_{ij})$ stands for "fixed'.

The nonlinear function $g^f(q_{ij})$ can typically be described by a quadratic function (Coulbeck, 1988; De Moyer and Horowitz, 1975).

$$g^f(q_{ij}) = \overline{A}_{ij} q_{ij}^2 + \overline{B}_{ij} q_{ij} + \overline{C}_{ij} \tag{2.5}$$

where $\overline{A}_{ij}$ is the resistance coefficient and $\overline{A}_{ij} < 0$, and $\overline{B}_{ij}$ is a coefficient often taking the value of $\overline{B}_{ij} \leq 0$ in order to ensure a single stable operating flow point for a headloss, and $\overline{C}_{ij}$ is the *cut-off* head.

2.3.3.2 Variable Speed Pumps

For variable speed pumps, the pump speed *s* can be continuously controlled over a certain speed range. The headloss function of a group of U_{ij} variable speed pumps in parallel having the same hydraulic curves can be written as:

$$\Delta h_{ji} = g_{ij}^s(q_{ij}, n_{ij}, s_{ij}) \tag{2.6}$$

where s_{ij} is a speed factor defined as

$$s_{ij} = \frac{operating\ speed}{nominal\ speed} \tag{2.7}$$

The nonlinear function $g_{ij}^{s}(\cdot)$ can typically be described by the following quadratic function (De Moyer and Horowitz, 1975; Coulbeck, 1988).

$$g_{ij}^{s}(q_{ij}, n_{ij}, s_{ij}) = \begin{cases} \overline{A}_{ij}(\frac{q_{ij}}{n_{ij}})^2 + \overline{B}_{ij}(\frac{q_{ij}}{n_{ij}})s_{ij} + \overline{C}_{ij}s_{ij}^2, & if\ n_{ij} \neq 0\ and\ s_{ij} \neq 0 \\ 0 & otherwise \end{cases} \tag{2.8}$$

2.3.3.3 Variable Throttle Pumps

The characteristic function of a group of throttle pumps in parallel is written as:

$$\Delta h_{ji} = g_{ij}^{f}(q_{ij}, n_{ij}) - \Delta h^{t}(q_{ij}, V_{ij}) \tag{2.9}$$

The first term in the above equation represents head increase of a fixed speed pump group. The second term stands for head drop across the throttle. The variable V_{ij} denotes the control factor for the throttle conductivity that can be modelled by a variable control valve, as shown in Section 2.3.2.

2.3.3.4 Pump Station

A general pump station, which is composed of all types of pumps above, is illustrated in Figure 2.4 (Brdys and Ulanicki, 1994) where there are M_i^s variable speed pumps, M_i^t

variable throttle pumps and M_i^f fixed speed pumps, q_i or q_{pi} are the overall pump flow, and h_{si}, h_{di} are the suction head and delivery head, respectively. Since the pumps are connected in parallel, the head is the same and the flows are added.

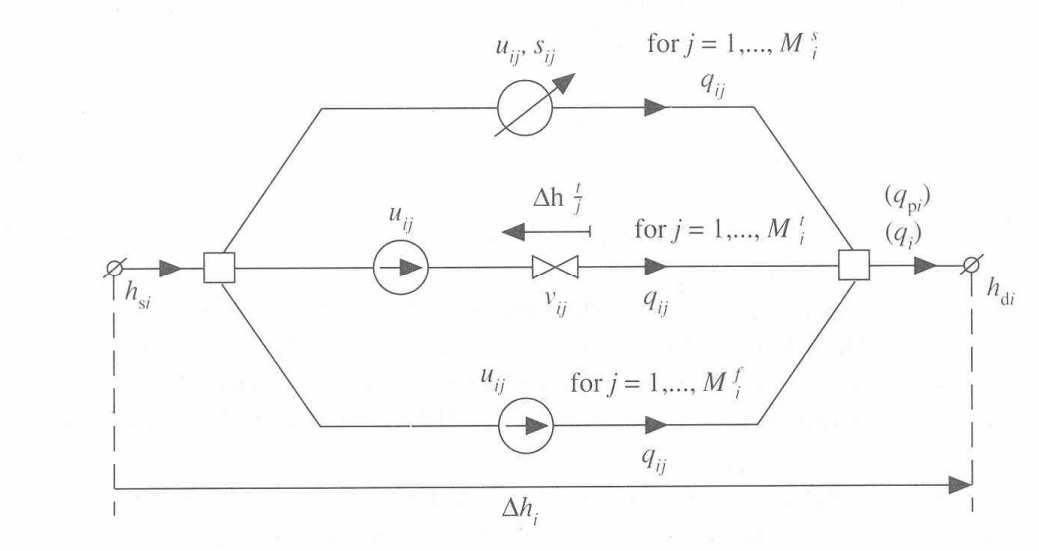


Figure 2.4 Generalized *i*th pump station configuration

2.3.4 Reservoirs and Tanks

Reservoirs and tanks are the dynamic components of water distribution systems. Due to their energy storage properties, they are different from other components in the water network, such as pipes, valves and pump stations. A typical reservoir model is illustrated in

Figure 2.5. For the *i*th reservoir node, the total water load is $q_{lr,i} = q_{s,i} - d_{r,i} + q_{out,i}$, where $q_{s,i}$ denotes water delivery from treatment works, $d_{r,i}$ is the demand flow allocated to the ith reservoir and $q_{out,i}$ is the disturbance flow different from $q_{s,i}$.

The water mass balance in the reservoir is described as:

$$\frac{dw_i(t)}{dt} = \rho[q_{r,i}(t) + q_{lr,i}(t)], \text{ for } i = 1,...,n_r \tag{2.10}$$

where ρ is the density of water, $w_i(t)$ and $q_{r,i}(t)$ denote the mass of water stored in the reservoir and the total flow into the reservoir from the network, respectively, at time instant t.

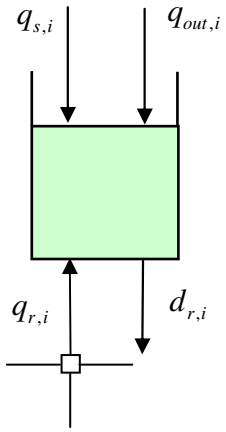


Figure 2.5 Model of a reservoir/tank

As more of our interests lie in the *i*th reservoir head rather than in the stored water, after substituting the water head and cross-sectional area of the reservoir into the above equation, the following relationship is obtained:

$$\frac{dx_i(t)}{dt} = \frac{1}{S_i(x_i(t))}[q_{r,i}(t) + q_{lr,i}(t)] \text{ for } i = 1,...,n_r \tag{2.11}$$

where $x_i(t)$ is the *ith* reservoir level and $S_i(x_i(t))$ denotes the reservoir cross-sectional area at this level.

Since it always holds that

$$h_{ri}(t) = x_i(t) + E_i \text{ for } i = 1,...,n_r \tag{2.12}$$

where $h_{ri}(t)$ is the reservoir head and E_i is the reservoir elevation, it can be obtained that:

$$\frac{dh_{ri}(t)}{dt} = \frac{1}{S_i(h_{ri}(t))}[q_{r,i}(t) + q_{lr,i}(t)] \text{ for } i = 1,...,n_r \tag{2.13}$$

The above equation can be simplified if the cross-sectional area does not depend upon the reservoir head. However, the equations still remain highly nonlinear because the flow $q_{r,i}(t)$ returned to the reservoir by the network depends on the reservoir head.

2.3.5 Physical Hydraulic Laws

2.3.5.1 Flow Continuity Law

For every junction node j, the following holds (see Figure 2.6):

$$\sum_{i \in J_j} q_{ij} = d_j \tag{2.14}$$

where J_j denotes the set of nodes linked with the node j, and d_j is the demand that is allocated to the jth junction node. For reservoir nodes, $d_j = 0$ if no demand is allocated to the jth node. If j is a reservoir node in a steady state, that is, when the reservoir level is constant, then the above equation is also satisfied at this node. The flow continuity law states that the sum of inflows and outflows is equal to zero for every non-reservoir node. It also holds for the reservoir nodes that are in a steady state.

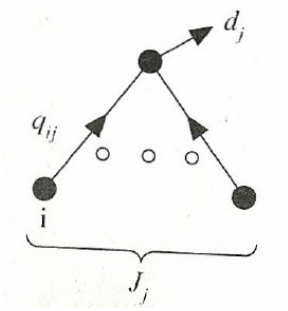


Figure 2.6 Connection Node

2.3.5.2 Energy Conservation Law

The energy conservation law is usually expressed in terms of the head change (increases or drops) along a loop or energy path. For a link (i, j) connecting nodes *i* and *j*, the difference $h_{ij} = h_i - h_j$ denotes the head drop/increase across the link. The following holds:

$$\sum h_{ij} = \partial E_r \tag{2.15}$$

where ∂E_r denotes the head difference between the starting and final nodes of the *r*-th path.

If the starting node and the final node are the same node, then the path constitutes a loop. Clearly, the following holds for a loop:

$$\sum h_{ij} = 0 \tag{2.16}$$

The energy conservation law applied to all network paths produces a number of equations of the type given in (2.15). Combining these equations with flow-head relationships describing branches, a loop network model can be obtained.

2.4 Mathematical Modelling of Water Leakage

Leakage from water distribution systems is a significant loss in water resources. For example, water leakage from distribution systems in the United Kingdom in 1989 was estimated to be 3,027 ML per day, which is more than the total supply each day by Thames Water (Pearce, 1991). Hence it is necessary for regulators to take appropriate control strategies in order to reduce levels of leakage. Several mathematical models of pipe leakage have been proposed by researchers based on experiment statistic results, among which Germanopoulos' equation (Germanopoulos, 1985) and May's equation (May, 1994) are most commonly used.

2.4.1 Germanopoulos' Equation

The mathematical model of water leakage proposed by Germanopoulos (1985) was based on the results of field experiments performed by the Water Authorities Association (1985). This model allows one to explicitly incorporate the leakage terms into the formulation of optimal operational control problem (Vairavamoorthy and Lumbers, 1998).

The rate of leakage Q_{ij}^{l} between nodes *i* and *j* can be estimated by the following expression (Germanopoulos, 1985):

$$Q_{ij}^{l} = c_{ij} L_{ij} \left\{ \frac{1}{2} \left[\left(h_i - E_i \right) + \left(h_j - E_j \right) \right] \right\}^{1.18} \tag{2.17}$$

where c_{ij} is a constant based on an estimate of the level of leakage and the corresponding average zonal pressures in the distribution network; L_{ij} is the length of pipe between nodes *i* and *j*; and h_i and E_i are the nodal heads and elevation of node *i*, respectively.

In order to explicitly incorporate the leakage terms (2.17) into the optimization model, nodal flow continuity equations (2.14) becomes

$$\sum_{i \in J_j^{'}} q_{ij} + \sum_{i \in J_j^{''}} \left\{ q_{ij} - \frac{1}{2} c_{ij} L_{ij} \left\{ \frac{1}{2} \left[\left(H_i - h_i \right) + \left(H_j - h_j \right) \right] \right\}^{1.18} \right\} = d_j \tag{2.18}$$

where $J_j^{'}$ includes all the nodes connected to node *j* without leakage and $J_j^{''}$ are those nodes connected to node *j* with a leakage link.

Note that the leakage part in (2.18) apportions the leakage term in a pipe equally between the two nodes connecting the pipe, which may be an oversimplification in some cases as noticed by Vairavamoorthy and Lumbers (1998).

2.4.2 May's Equation

May (1994) proposed a leakage-pressure model by the power law (2.19) and suggested that leakage from pipe bursts depends on pressure with a power of 0.5 and background leakage flow depends on pressure with a power of 1.5. For plastic pipes, the pressure exponent in this leakage model can sometimes be bigger up to 2.5 (Lambert, *et al*., 1998).

The rate of leakage lq_i at the *i*th node can be modelled by the following expression:

$$lq_i = C_E p_i^{\gamma} \tag{2.19}$$

where p_i is the water pressure at node *i*. C_E and γ are the leakage coefficient and leakage exponent, which are equivalent to the emitter coefficient and emitter exponent, respectively, in EPANET (Rossman, 2000). In (Rossman, 2000), the nonlinear leakage-pressure model (2.19) is implemented as *emitters*, which are devices associated with junctions that model the flow through a nozzle or orifice that discharges to the atmosphere. Emitters can also be used to model flow through sprinkler systems, irrigation networks and to simulate leakage in a pipe connected to the junction if a leakage coefficient and leakage exponent for the leaking crack or joint can be estimated (Rossman, 2000). Due to the compatibility of May's leakage-pressure model with EPANET simulator, the nonlinear leakage-pressure relationship (2.19) is used in the thesis.

In order to explicitly incorporate the leakage term, i.e. lq_i in the orifice function (2.19), into the optimization model, nodes of the network are categorized into leaky nodes and non-leaky nodes. Then the nodal flow continuity equation (2.14) becomes

$$\begin{cases} \sum_{i \in J_j} q_{ij} = d_j & \text{if } j \in M \\ \sum_{i \in J_j} q_{ij} = d_j + l q_j & \text{if } j \in M_l \end{cases} \tag{2.20}$$

where set J_j includes all the nodes connected to node j. M and M_l denote the sets of non-leaky and leaky nodes.

The above presented physical models of water distribution systems and the empirical models of water leakage are all nonlinear.

2.5 Nodal Model

The problem of the network static simulation consists of determining the instantaneous values of unknown heads and flows in the network under given values of demand to junction nodes and under given values of the control inputs. As described in (Brdys and Ulanicki, 1994), the nodal model is derived in terms of unknown heads and unknown branch flows. During simulation applications, heads at the non-reservoir nodes are unknown heads, while branch flows represent unknown flows. In case of estimation, a situation may be more general and depend upon available measurements. The nodal model is also needed to introduce here in order to have the mathematical formulation in the Chapter 3. The detail of how to obtain the nodal model can be found in (Brdys and Ulanicki, 1994).

It is important to distinguish between non-reservoir nodes and reservoir nodes. The first type of nodes will be called junction nodes, while the second type of nodes will be called fixed head nodes.

The total number of nodes, reservoir nodes, and junction nodes are respectively denoted n, n_r, and n_c. Hence,

$$n = n_r + n_c \tag{2.21}$$

In order to write the Eq. (2.20) in a matrix form, an incidence matrix for junction nodes needs to be defined. Let denote b the total number of branches. The branches are identified not by their origin and destination nodes but by subsequent integer numbers starting from 1 (see Figure 2.7). Therefore, there are b branch flows q_i, where $i = 1,...,b$

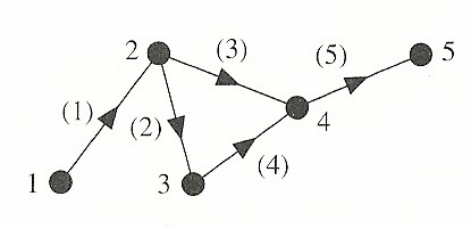


Figure 2.7 Network graph

An element in the ith column and jth row of the *junction nodes incidence matrix* Λ_c is defined as follows:

$$a_{ji} = \begin{cases} 1 & \text{if flow of branch i enters node j} \\ 0 & \text{if branch i and node j are not connected} \\ -1 & \text{if flow of branch i leaves node j} \end{cases}$$

Notice that the incidence matrix rows correspond to the non-reservoir nodes while its columns are related to the network branches. Thus there are n_c rows and b columns.

A matrix form of Eq. (2.20) is as follows:

$$\Lambda_c q = d + lq \tag{2.22}$$

where $q = (q_1,...,q_b)^T$ is the vector of branch flows, d and lq respectively denote an augmented demand vector and leakage vector by zero components corresponding to non-loaded and non-leaky nodes.

In Eq. (2.22), typically the number of junction nodes n_c is smaller than the number of branches b, i.e. $n_c < b$. Hence, given demand vector d, in order to determine the flow q in the static simulation, we need other $b - n_c$ equations relating pipes and flows. The network element flow-head relationships (see Eqs (2.2)-(2.9)).

By adding the equations to the vector flow, Eq. (2.22), we obtain $n_c + b$ equations overall, that is, more than the number of unknown flows. However, the resulted set of equations will also introduce new variables which are unknown nodal heads. There are n nodes overall in the network. It is assumed that all of the nodal heads are unknown. Therefore, there will be $n + b$ unknown quantities involved in $n_c + b$ equations covering all pipe, valve and pump station flows and nodal heads.

Let write some of representative head-flow equations.

For the i th pipe connecting the l th reservoir node as a destination node and the j th non-reservoir as an origin node, the following holds:

$$\begin{aligned} h_j - h_{r,l} &= g_{jl}(q_{jl}) = R_{jl} q_{jl} \left| q_{jl} \right|^{0.852} \\ &= R_i q_i \left| q_i \right|^{0.852} = g_i(q_i) \end{aligned} \tag{2.23}$$

where, clearly, $q_{jl} = q_i$, $R_{jl} = R_i$, and $g_{jl}(\cdot) = g_i(\cdot)$

For the i th pump station composed of only variable speed pumps and pumping water from the l th reservoir node into the j th reservoir node, the following holds:

$$\begin{aligned} h_{r,j} - h_{r,l} &= g_i^s(q_{lj}, s_i, u_i) \\ &= A_i\left(\frac{q_{lj}}{u_i}\right)^2 + B_i\left(\frac{q_{lj}}{u_i}\right)s_i + C_i s_i^2 \\ &= g_i^s(q_i, s_i, u_i) \end{aligned} \tag{2.24}$$

For the i th valve connecting the j th non-reservoir node as an origin node and the l th non-reservoir node as a destination node, the following holds:

$$\begin{aligned} h_j - h_l = g_{jl}(q_{jl}, \nu_{jl}) &= R_{jl}(\nu_{jl}) q_{jl} \left|q_{jl}\right|^{0.852} \\ &= R_i(\nu_i) q_i \left|q_i\right|^{0.852} = g_i(q_i, \nu_i) \end{aligned} \tag{2.25}$$

where, clearly $q_{jl} = q_i$, $\nu_{jl} = \nu_i$, $R_{jl} = R_i$, and $g_{jl}(\cdot) = g_i(\cdot)$

In order to write the flow-head equations in vector form, a suitable incidence matrix Λ needs to be defined. The matrix Λ clearly has n rows and b columns. The element b_{ij} of the matrix Λ^T is defined as follows:

$$\mathrm{b_{ij}} = \begin{cases} -1 & \text{if } i \text{ th branch leaves node } j \\ 0 & \text{if } i \text{ th branch and node } j \text{ are not connected} \\ 1 & \text{if } i \text{ th branch enters node } j \end{cases} \tag{2.26}$$

For the sake of convenience, the rows corresponding reservoir nodes are placed in the first n_r position. The other rows correspond to the junction nodes. The flow-head relationship now can be written in vector form:

$$\Lambda^T \begin{bmatrix} h_r \\ h \end{bmatrix} + \Psi(q) = 0 \tag{2.27}$$

where $h_r = (h_{r1},...,h_{r,n_r})^T$ - vector of heads of reservoir nodes (2.28)

$h = (h_1,...,h_{n_c})^T$ - vector of heads of junction nodes (2.29)

$q = (q_1,...,q_b)^T$ - vector of branch flows (2.30)

$\Psi(q) = (g_1(q_1),...,-g_i^f(q_i,u_i),...,-g_j^s(q_j,s_j,u_j),...,g_1(q_1,v_1),...)^T$ -

vector of function defining flow-head relationship for branches (2.31)

Combining Eqs. (2.27) and (2.22) constitutes a nodal model of the network. Assuming that the non-reservoir nodal demand vector d is known, the nodal model consists of $n_c + b$ equations and $n_r + n_c + b$ unknown quantities. There are n_r more unknown quantities than the total number of equations. The reason is that the continuity law is indeed not applicable to the reservoir nodes which are not in a steady state. In order to solve the equations, n_r flows and heads must be assumed known. There could be different theoretical and practical situations. For example, considering dynamical network simulation the reservoir heads at a given time stage are always known as a simulation product up to now. In this case, the vector of reservoir heads h_r is known; hence the number of unknown variables in the nodal model is equal to the number of equations.

The matrix Λ can be partitioned as:

$$\Lambda = \begin{bmatrix} \Lambda_r \\ \Lambda_c \end{bmatrix} \tag{2.32}$$

where Λ_c has been already used in the Eq. (2.22) and matrix Λ_r is the incidence matrix defined for the reservoir nodes.

Substituting Λ from Eq. (2.32) into Eq. (2.27):

$$\begin{bmatrix} \Lambda_r^T & \Lambda_c^T \end{bmatrix} \begin{bmatrix} h_r \\ h \end{bmatrix} + \Psi(q) = 0 \tag{2.33}$$

Combining Eq. (2.22) and Eq. (2.33) yields the nodal model:

$$\begin{cases} \Lambda_c q = d + lq \\ \Lambda_c^T h + \Psi(q) = -\Lambda_r^T h_r \end{cases} \tag{2.34}$$

2.6 Control strategies

In the Section 2.2, three typical operational states of the DWDS have been distinguished. Their corresponding control strategies need to be mathematically formulated. The control objectives (performance index), all the constraints on decision vector and state of the plant, and control technology in a sequel are understood in the thesis as a *control strategy.* Since MPC is the control technology to be considered, mathematically formulating the control strategy means to specify the performance index and the system constraints of the corresponding MPC optimization task.

Simulation of water distribution networks typically includes static simulation and dynamical simulation. The problem of static simulation consists of determining the instantaneous values of unknown nodal heads and link flows in the network under given values of water demand loads assigned to the nodes and under the prescribed values of the manipulated variables, e.g. relative pump speeds, valve settings. In operational control of water distribution networks, dynamical simulation needs to be applied since a network is considered not just at a particular time instant t but over a time horizon $[t_0, t_0 + T_c]$.

In network dynamical simulation, an operational control problem is solved for a certain time period, e.g. 24 hours, for given demand prediction initial reservoir heads. The pump schedules are applied to the real water system for a short period and then an optimal scheduling problem is solved again with updated predictions and actual values of the state variables which are measured in the real system.

2.6.1 Control strategy of normal operational states

When the DWDS is in *normal operational state*, a common control objective is to operate the system within the prescribed operational limits at minimum operational cost. The operational cost J_{op} consists of three components (Brdys and Ulanicki, 1994):

- the cost of pumping energy J_P
- the cost of treatment J_T
- the cost of maximum demand charge J_{MDC}

The pumping energy cost J_P is mainly considered in the thesis. Discussions on the cost of J_T and J_{MDC} can be found in (Brdys and Ulanicki, 1994).

The pumping cost $J_i(k)$ for the *i*th pump station composed of a number of pumps connected in parallel at the *k*th time step is written as

$$J_i(k) = J_i(q_i, \Delta h_i, u_i, s_i, k) = \gamma_i(k) \sum_{j=1}^{N_i} \frac{\xi q_{ij} \Delta h_i}{\eta_{ij}(q_{ij}, n_{ij}, s_{ij})} \tag{2.35}$$

where $\gamma_i(k)$ is a power unit charge at the $(k+1)$ time stage, and ξ is the unit conversion coefficient relating hydraulic quantities to electrical energy. n_{ij} denotes the ON-OFF state for the *j*th pump, η_{ij} is pump efficiency and s_{ij} is the pump speed, $j = 1, ..., N_i$.

The total pumping cost is the summation over all time steps for all pump stations:

$$J_p = \sum_{k=0}^{K-1} \sum_{i=1}^{I} J_i(q_i, \Delta h_i, n_i, s_i, k) \tag{2.36}$$

where *K* is the number of time steps, and *I* is the number of pump stations.

Hydraulic constraints are often split into two major categories, which are equality constraints and inequality constraints. Models of a water distribution system generate equality constraints, which include:

- head-flow relationships of network links (pipes, pumps and valves)
- mass balance equations of reservoirs/tanks
- flow continuity law

These equality constraints have been discussed in Section 2.3 and will not be repeated here. Operational requirements are in the form of inequalities imposed on reservoir levels, system pressures and flows. The bounds in these inequalities can change depending on the current objectives of the operational control of DWDS. The inequality constraints are presented as follows:

(1) Reservoir/tank levels

These are upper and lower limits of reservoir/tank levels during the whole period of operational control of DWDS in order to avoid overflow or emptying the reservoirs/tanks. The general form of this group of constraints is as follows:

$$\mathbf{x}(k) \in \mathbb{X}_k, \quad k = 1, \ldots, K \tag{2.37}$$

(2) Non-reservoir nodal heads

For some non-reservoir nodes within the network, it is often required that the heads are within certain ranges. Particularly, water pressure at consumer nodes should be maintained at certain levels. The general form of this group of constraints is

$$\mathbf{h}(k) \in \mathbb{H}_k, \quad k = 1, \ldots, K \tag{2.38}$$

(3) System flows

For certain links within the network, it is sometimes required that the flow rate of the links is within certain ranges:

$$\mathbf{q}(k) \in \mathbb{Q}_k, \quad k = 1, \ldots, K \tag{2.39}$$

Discussions of other operational constraints, such as reservoir flows and treatment work set points, are not included in the thesis, but can be found in (Brdys and Ulanicki, 1994).

2.6.2 Control strategy of disturbed operational states

During the operational control of DWDS, the soft pressure limit may start violating over certain period time due to demand changes if the DWDS is controlled by pumping cost minimization control strategy. It is because optimizing the energy cost requires varying pump speeds over the whole range. The DWDS is then seen as operating in the *disturbed operational state*. Continuing operation in this operational state over long period may cause pipe burst and water leakage. Hence, in order to prevent the leakage, the normal control strategy needs to be changed to the control strategy which enables us to reduce the pressure profile over the DWDS. The new RFMPC controller will minimize the selected pressures violating the soft constraints and the energy cost. These two components are weighted with priority of the pressure reduction and sum up to produce the performance function suitable for DWDS operation in the *disturbed operational state*. . If the pressure profile is back to normal, then this control strategy will be softly switched to the normal one which will be presented in Chapter 6. The objective function can thus be written as:

$$J_e = \xi_p J_p + \xi_d \sum_{k=0}^{K-1} \sum_{j=1}^{NPN} h_j(k) \tag{2.40}$$

where h_j is the pressure of the jth node, NPN is the number of nodes, K is the number of time steps, ξ_n and ξ_d are suitably chosen weights.

This control strategy has the form of the constraints as those of the pumping cost minimization control strategy, since in both cases no leakage is involved.

2.6.3 Control strategy of emergency operational states

When the leakage occurred, the DWDS is seen in *emergency operational state*. The control objective is usually to obtain the minimum water leakage volume within the prescribed operational limits. The corresponding objective function is expressed as

$$J_w = \xi_n J_n + \xi_w \sum_{k=0}^{K-1} \sum_{l=1}^{L} w_l(k) \tag{2.41}$$

where w_l is the leakage rate of the lth link, L is the number of links having water leakage, K is the number of time steps, ξ_n and ξ_w are suitably chosen weights.

Due to the existence of the leaky nodes, the set M_l is not empty and the nodal flow continuity Eq (2.20) is used instead of Eq (2.14). Therefore, replacing Eq (2.14) by Eq (2.20) in the constraints of the pumping cost minimization control strategy yields the constraints of the leakage minimization control strategy.

2.7 Summary

As the DWDS is the application of the thesis, the DWDS and its operational conditions have been presented. Three typical operational states of the DWDS and the corresponding control strategies have been distinguished. The physical modelling of the DWDS

components has been also presented and is used to construct the optimization model for the operational control of the DWDS in the Section 2.6.

For predictive control strategy switching purposes, models of water leakage have also been briefly presented. The advantage of the presented leakage models are that they can both be incorporated into the optimization constraints explicitly, which is more reasonable than the traditional approach of minimising water leakage proposed by (Sterling and Bargiela, 1984) which does not take the leakage term explicitly into account. Researches related to leakage-pressure modelling and control of water leakage can be found in (Jowitt and Xu, 1990; Pudar and Liggett, 1992; May, 1994; WRc-UK, 1994; Lambert, *et al*., 1998; Vairavamoorthy and Lumbers, 1998; Rossman, 2000).

In the operational control of DWDS, three common control strategies have been presented in order to either to minimize the electrical pumping cost or to minimizer excessive pressure for leakage prevention purpose or to minimize the leakage in case leakage occurs. Those control objectives are repeatedly used throughout the thesis to achieve simulation results. The nodal model has been presented and will be utilized to compute the gradients via the Hamiltonian based technique in the Chapter 4 in order to reduce the computation burden of MPC technology.

Chapter 3

Iterated Safety Zones Based Robustly Feasible MPC and Application to DWDS

This chapter considers iterated safety zones robustly feasible MPC. The robustly feasible MPC architecture and its components are presented. The computing effort of RFMPC is reduced by applying the stepwise robust output prediction technique and shortening robust feasible horizon. Furthermore, the Sequential Quadratic Programming is chosen as the optimization solver of the robustly feasible MPC to effectively handle nonlinear constraints and improve the computing time. RFMPC is applied to the DWDS example and simulation results are presented.

3.1 Introduction

MPC has been intensively used in process industry due to its ability to the handle multivariable linear and nonlinear constrained control problems. The MPC itself nevertheless heavily depends on the model of the plant and does not efficiently incorporate the model-reality mismatch. The model-reality mismatch is often caused by the differences between predicted disturbances in the model and the disturbances in reality. Several approaches have been proposed by researcher to overcome this issue such as min-max approach by (Kothare et al., 1996), (Lee and Yu, 1997); reference governor approach by (Bemporad and Mosca, 1998), constraint restriction approach by (Chisci et al., 2001), safety zone approach by (Brdys and Chang, 2002), (Kerrigan and Maciejowski, 2001), (Kerrigan, 2000), (Grieder et al., 2003). In this chapter, the iterated safety zones based RFMPC approach is considered. The key idea of the approach is to tighten the constraints to the modified constraints by the so-called safety zones. When the optimal control input is applied to the plant, the modified constraints might be violated. However, the real constraints are satisfied. The structure of the RFMPC requires significant computing effort because it iteratively uses MPC optimizer to find the safety zones during each control step. The process is repeated for all control steps over the horizon leads to the needs for improving computing time for online operation. Several techniques to improve computing time including stepwise robust output prediction, shortening robust feasibility horizon, setting up the more effective optimization solvers are proposed.

This chapter is organized as follows: a brief survey of MPC and the formulation of the MPC optimization task are presented in Section 3.2; the iterative control structure of

RFMPC is described in Section 3.3; the robust output prediction, technique of SWROP and shortening robust feasibility horizon are derived in Section 3.4; the algorithm to calculate safety zone is presented in Section 3.5; the MPC optimizer and the Hamiltonian based technique to calculate suitable gradients and derivates are presented in Section 3.6; the simulation environment is explained in Section 3.7; the DWDS case study and simulation results are illustrated in Section 3.8; Section 3.9 summarizes the chapter.

3.2 Model Predictive Control

Predictive control, or model predictive control (MPC) as it is widely known, has become an accepted standard and an advanced control technique in industry where it is used as an effective control means to handle multivariable constrained control problems. The underlying ideas for model predictive control originated in the sixties of the last century as a natural application of the optimal control theory. Firstly appeared in (Propoi, 1963), a controller with close connections to MPC was developed, and later a more general optimal control based feedback controller was discussed in (Lee and Markus, 1968). The true birth of MPC originated from industrial applications in the mid-seventies to mid-eighties advocated by the work on model predictive heuristic control (MPHC) (Richalet *et al*., 1978) and dynamic matrix control (DMC) (Cutler and Ramaker, 1980). During this period, there was a flood of new variants of MPC which differed typically in the process models but not much between their fundamental algorithms (Camacho and Bordons, 1999). During the nineties, stability and online implementation of MPC have been extensively

investigated by researchers and industrial practitioners, see (Mayne *et al*. 2000) for a summary of theoretical development and (Qin and Badgwell, 2003) for a survey of industrial applications. Other notable past paper reviews and books include those of (Garcia *et al*., 1989; Muske and Rawlings, 1993; Mosca, 1995; Morari and Lee, 1999; Camacho and Bordons, 1999; Bemporad and Morari, 1999a; Mayne *et al*. 2000; Rawlings, 2000; Kouvaritakis and Cannon, 2001; Maciejowski, 2002; Rossiter, 2003; Qin and Badgwell, 2003; Allgower *et al*., 2004; Kwon and Han, 2005).

3.2.1 MPC Architecture

MPC actually belongs to a class of model based controller design concepts. No matter what kind of plant models are considered, the basic idea of MPC algorithm remains unchanged. It determines the optimal control actions by minimizing a user-defined objective function, or performance index, which penalizes the difference between the predicted output trajectories from their reference trajectories. The current control action is calculated by solving on-line, at each control step, a finite-horizon open-loop optimization problem, using the current state of the plant process as the initial state of the optimization problem. Only the first part of the optimized control sequence is applied to the plant. At next control step the prediction horizon moves forward and the same procedure repeats. Due to its operating on a receding horizon, MPC is also referred to as receding horizon control (RHC).

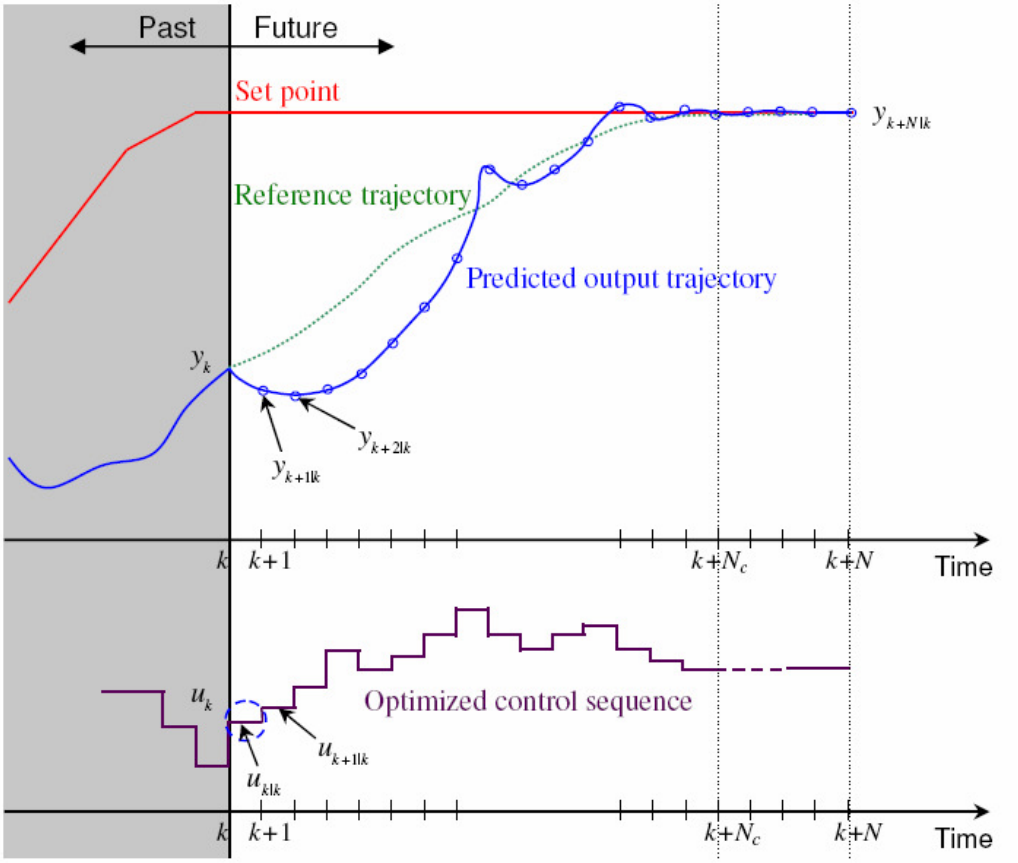


Figure 3.1 Basic idea of MPC operation on a receding horizon

The idea of MPC is illustrated in Figure 3.1 (Maciejowski, 2002). The current time instant is k and the present output is y_k. $y_{\cdot|k}$ and $u_{\cdot|k}$ denote the predicted output and control input, respectively. H_p and H_c are the finite time prediction horizon and control horizon . The set- point trajectory may be varying depending on the operation of the plant process. A reference trajectory considering an ideal or desired tracking trajectory from the current output to the set-point trajectory can be defined over the prediction horizon H_p before running MPC. In a receding horizon operation only the first control action is applied to the plant process over the control step. Next, the process variables are measured and the optimization problem is solved again over the prediction horizon with the initial conditions

updated from the measurements. The basic structure of a MPC control loop is depicted in Figure 3.2.

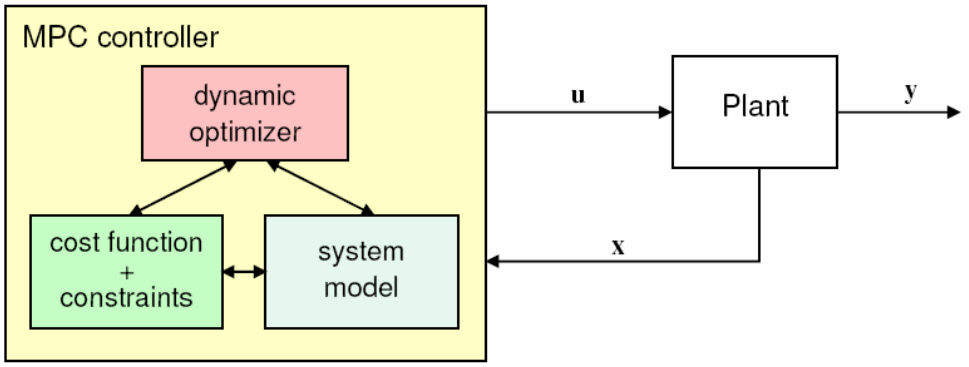


Figure 3.2 Basic MPC control loop

Putting the conceptual idea of receding horizon control into an optimization algorithm yields the following operation of a basic MPC controller.

Algorithm 3.1 (*Basic MPC controller*)

1. At time k, obtaining the current state x_k of the plant;
2. Obtaining $u_{\cdot|k}$ by solving a finite horizon optimal control problem
3. Applying the first element in the control sequence, that is $u_k = u_{k|k}$;
4. $k \leftarrow k+1$. Go to step 1.

3.2.2 Formulation of Nominal MPC optimization task in a full space of decision variables

In non-linear network, it shall be distinguished between vector of inputs u, outputs y, and states x. As all three variables need to be used either directly or indirectly, it shall be introduced a composed vector s of all network variables:

$$s = [u^T, y^T, x^T]^T \tag{3.1}$$

It is typical that in reality the value of states, output, and control input are constrained within certain lower-upper limits. For simplicity of the derivation, assume now that the inputs, outputs, and states are subject to constraints:

$$u \in \mathbf{U} = [u^{\min}, u^{\max}] \subset \mathbb{R}^m \tag{3.2a}$$

$$y \in \mathbf{Y} = [y^{\min}, y^{\max}] \subset \mathbb{R}^l \tag{3.2b}$$

$$x \in \mathbf{X} = [x^{\min}, x^{\max}] \subset \mathbb{R}^n \tag{3.2c}$$

In short, one could write:

$$s \in \mathbf{S} = [s^{\min}, s^{\max}] \subset \mathbb{R}^s \tag{3.3}$$

where $\mathbf{S} = \mathbf{U} \times \mathbf{Y} \times \mathbf{X}$

The network mathematical model is composed of two parts: static and dynamic. The static part is typically available in an implicit form represented by number of equalities describing the network individual elements and connections between the elements. The equalities are described by linear and nonlinear functions composing an operator F. The operator F represents the static part of the network model to produce the set of equalities:

$$F(s(k), d(k)) = 0 \tag{3.4}$$

where $d(k)$ denotes the disturbance at time k

In general, the network dynamics is described by:

$$x(k+1) = \tilde{f}(x(k), s(k), d(k)) \tag{3.5}$$

The network dynamics is due to the network storage capabilities. For example $x(k)$ is a vector of reservoir volumes in case of water supply and distribution network.

If the control input u is known and the state x is given, the vector s can be found by solving Eq. (3.4). The operator E relating the state x, control input u, and corresponding output y is introduced based on (3.4):

$$y(k) = E(u(k), x(k), d(k)) \tag{3.6}$$

Suppose that the control horizon and prediction horizon are equal i.e. $H_c = H_p = N$.

The vector of predicted states over the prediction horizon is defined as:

$$x(\cdot \mid k) = [x(k \mid k) \ldots x(k+N-1 \mid k)]^T \tag{3.7}$$

where $x(k+i \mid k)$, $i = \overline{0:N}$ stands for model state at $k+i$, and such model prediction is performed at time instant k.

Similarly, the vectors of predicted control input, output, and disturbance over the prediction horizon are respectively defined as

$$u(\cdot \mid k) = [u(k \mid k) \ldots u(k+N-1 \mid k)]^T$$

$$y(\cdot \mid k) = [y(k \mid k) \ldots y(k+N-1 \mid k)]^T$$

$$d(\cdot \mid k) = [d(k \mid k) \ldots d(k+N-1 \mid k)]^T$$

A nominal MPC (NMPC) is defined as a deterministic MPC with selected prediction of the disturbance inputs. The MPC optimization task in this section is understood as NMPC where the predicted disturbance is already known.

Define $s(\cdot \mid k) \triangleq [u(k \mid k)...u(k+N-1 \mid k)\ \ y(k \mid k)...y(k+N-1 \mid k)\ \ x(k \mid k)...x(k+N-1)]^T$

The MPC optimization task in the full space of decision variables at time k is formulated as follows:

$$\begin{aligned} &\min_{s(\cdot \mid k),\, x(k+N \mid k)} \tilde{J}(s(\cdot \mid k), x(k+N \mid k)) \\ &\text{subject to:} \\ &x(k \mid k) = x(k) \\ &F(s(k+i \mid k), d(k+i \mid k)) = 0 \\ &x(k+i+1 \mid k) = \tilde{f}(x(k+i \mid k), s(k+i \mid k), d(k+i \mid k)) \\ &s^{\min} \le s(k+i \mid k) \le s^{\max} \\ &x^{\min} \le x(k+N \mid k) \le x^{\max} \\ &i = \overline{0:N-1} \end{aligned} \tag{3.8}$$

where $\tilde{J}(s(\cdot \mid k), x(k+N \mid k)$ is the cost function and $x(k)$ denotes network state at k which is known from measurement.

Solving the optimization task (3.8) with respect to $s(\cdot \mid k)$ and $x(k+N \mid k)$ over the horizon N produces optimal control input sequences $u^{opt}(k+i \mid k)$, $i = \overline{0:N-1}$. Only the first control action is applied to the network. Hence, $u^{opt}(k) = u^{opt}(k \mid k)$. At $k+1$, the network state $x(k+i \mid k)$ is measured and the above procedure is repeated to produce $u^{opt}(k+1) = u^{opt}(k+1 \mid k+1)$. The MPC optimization task (3.8) is solved again over

$i = \overline{0:N-1}$. The formulation (3.8) is said to be in the full space of decision variables since all network variables i.e. states, inputs, and outputs are explicitly embedded and treated as decision variables of the optimization.

3.2.3 Formulation of Nominal MPC optimization task in a reduced space of decision variables

The formulation of MPC optimization in (3.8) has an advantage due to the simplicity to understand and implement. However, since all network variables are embedded and treated as decision variables, the optimization task (3.8) certainly becomes very computationally demanding. There are different ways to handle this issue. For example, one could find better optimization solvers which have better computation efficiency. Alternatively, the formulation (3.8) could be re-formulated in order to reduce the number of decision variables, hence reducing the computational demands. In this section, the formulation of NMPC optimization task in reduced space of decision variables is presented and will be used to obtain the simulation results of the thesis.

Given the control inputs $u(\cdot|k)$, predicted disturbance $d(\cdot|k)$ and state $x(k|k)$, the prediction $s(k+i|k)$ of $s(k+i)$ and the predicted state $x(k+N|k)$ can be calculated as follows:

- Using Eq. (3.6) with $u(k) = u(k \mid k)$ and $d(k) = d(k \mid k)$, the output prediction $y(k \mid k)$ is calculated and then substituted into (3.1) to obtain $s(k \mid k)$
- The predicted states $x(k+i \mid k)$ is calculated from Eq. (3.5) with $s(k) = [u(k \mid k), y(k \mid k), d(k \mid k)]$ and $d(k) = d(k \mid k)$.
- The procedure can be carried out for $i = \overline{0 : N-1}$

Let define the process above by an operator G, that is:

$$G(u(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) \triangleq [s(\cdot \mid k), x(k+N \mid k)] \tag{3.9}$$

It is important to notice that there is no analytical expression of $G(\cdot)$. However, the values of $G(\cdot)$ are obtained under given control input $u(\cdot \mid k)$, predicted disturbance $d(\cdot \mid k)$ and state $x(k \mid k)$.

For simplicity, $G(\cdot \mid k)$ means $G(u(\cdot \mid k), d(\cdot \mid k), x(k \mid k))$ unless otherwise specified.

Substituting $G(\cdot \mid k)$ from (3.9) into $\tilde{J}(s(\cdot \mid k), x(k+N \mid k))$ results:

$$\begin{aligned} \tilde{J}(s(\cdot \mid k), x(k+N \mid k) &= \tilde{\tilde{J}}(G(\cdot \mid k)) \\ &\triangleq J(u(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) \end{aligned} \tag{3.10}$$

Denote

$$\begin{aligned} \tilde{c}(s(\cdot \mid k), x(k+N \mid k)) = [&s(k \mid k) - s^{\min}; \ldots; s(k+N-1 \mid k) - s^{\min}; \\ &s^{\max} - s(k \mid k); \ldots; s^{\max} - s(k+N-1 \mid k);\ x(k+N \mid k) - x^{\min}; x^{\max} - x(k+N \mid k)] \end{aligned}$$

The constraints $\begin{cases} s^{\min} \le s(k+i \mid k) \le s^{\max} \\ x^{\min} \le x(k+N \mid k) \le x^{\max} \\ i = \overline{0 : N-1} \end{cases}$ are equivalent to:

$$\tilde{c}(s(\cdot \mid k), x(k+N \mid k)) \in \mathbf{C} \tag{3.11}$$

Using (3.10) and (3.11), the formulation of MPC optimization task in the so-called reduced space form is written as follows:

$$\begin{aligned} &\min_{u(\cdot|k)} J(u(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) \\ &\text{subject to:} \\ &c(u(\cdot \mid k), d(\cdot \mid k), x(k)) \in \mathbf{C} \end{aligned} \tag{3.12}$$

The function $J(\cdot)$ and $c(\cdot)$ are not known explicitly. However, the decision variables are now only the control inputs. This formulation is very useful in the practical simulation. However, in order to apply a suitable optimization solver to (3.12), a special Hamiltonian based technique will be used to find the gradients of these functions.

3.3 Iterative Control Structure

As MPC belongs to the model based controller, the accuracy of the plant model is one of the key factors to determine how effective MPC is. The model-reality mismatch is often caused by the difference between predicted disturbance and actual disturbance. When the input from the nominal model based MPC controller is applied to the plant, due to uncertainties in the system, the output constraint cannot be fulfilled and their violations

may be inevitable at certain time instants. Hence, it is important to have MPC controllers that also take disturbance into consideration when determining optimal control inputs. Such MPC controllers are called robustly feasible model predictive controllers. Several approaches have been used by researchers to design robust MPC. Specifically, min-max approach has been presented in (Kothare *et al*., 1996; Lee and Yu 1997; Scokaert and Mayne 1998; Lee and Cooley 2000; Casavola *et al*., 2000), reference governor approach has been introduced in (Bemporad and Mosca, 1998), and constraint restriction approach has been presented in (Bemporad and Garulli, 2000) and (Chisci et al, 2001). In (Brdys and Chang, 2002), safety zone approach has been proposed to design robustly feasible model predictive controller, which have been used in a quality control problem of the DWDS. The structure of the RFMPC consists of several units as illustrated in Figure 3.3. The MPC optimizer solves the nominal MPC optimization task as described in Section 3.6 to produce control inputs.

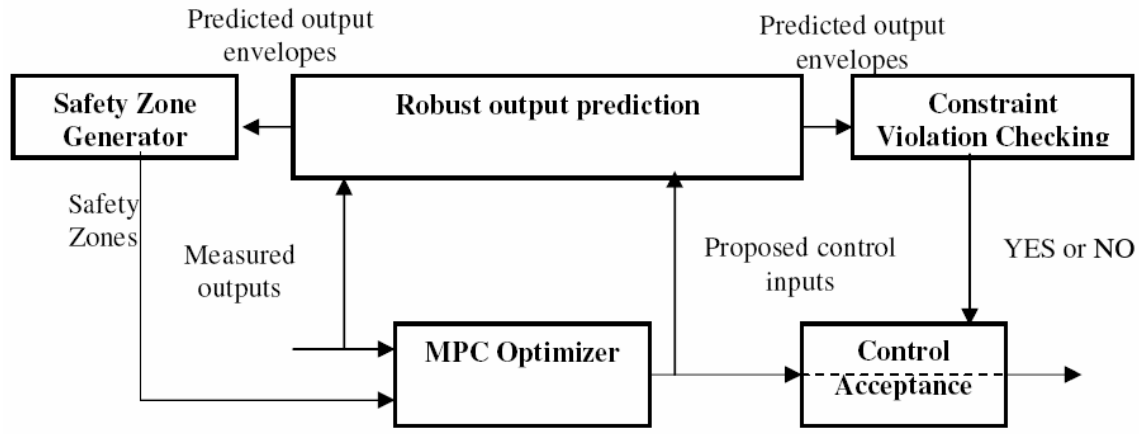


Figure 3.3 Iterative Control Structure

In the nominal model the disturbance inputs are represented by their predictions, while the internal model uncertainties are represented by a selected scenario. Before the control input is

applied to the plant its robust feasibility is assessed by the "Constraint Violation Checking" unit. The feasibility assessment is based on the robust output prediction that is generated by the "Robust Output Prediction" unit. Given the control input the corresponding robust output predictions over the prediction horizon is a region in the output space in which all the plant outputs generated by the control input and all possible scenarios of the disturbance inputs are contained. The input robust feasibility is checked by confronting the output constraints with the robust output prediction. If the control feasibility passed its assessment, then the proposed control input is applied to the plant. Otherwise, robust output prediction is fed into the "Safety Zone Generator" unit. The safety zones as such are used to tighten the output constraints. The control actions produced by the MPC optimizer under modified (tighten) output constraints are expected to produce the real plant outputs that satisfy the plant constraints although they still may violate the modified constraints. Such control actions and the corresponding safety zones are called robustly feasible.

3.4 Robust Output Prediction

Apply (3.6) into (3.1) to express $s(k)$ in term of $x(k)$ and $u(k)$, and apply the result to $\tilde{f}(x(k), s(k), d(k))$ in (3.5) to obtain:

$$x(k+i+1|k) = f(x(k+i|k), u(k+i|k), d(k+i|k)) \tag{3.13}$$

Suppose that the prediction horizon and control horizon are equal i.e. $H_p = H_c = N$

Given inputs $u(k+i\,|\,k)$ for $i=\overline{0:N-1}$, where N is the prediction horizon, the plant output over N can be predicted by using the plant model as:

$$\begin{cases} x(k+i+1\,|\,k) = f(x(k+i\,|\,k), u(k+i\,|\,k), d(k+i\,|\,k)) \\ y(k+i\,|\,k) = E(x(k+i\,|\,k), u(k+i\,|\,k), d(k+i\,|\,k)) \end{cases} \tag{3.14}$$
$$\text{with the initial conditions}: x(k\,|\,k) = x(k)$$

Vector of the predicted control inputs $u(\cdot\,|\,k)$ and output prediction $y(\cdot\,|\,k)$ over prediction horizon N has been defined in Eq. (3.7).

The robust output prediction of $y(\cdot\,|\,k)$ is composed of two envelopes (Chang, 2003):

$$y_p^l(\cdot\,|\,k) = [y_p^l(k\,|\,k) \ldots y_p^l(k+N-1\,|\,k)]^T \tag{3.15a}$$

$$y_p^u(\cdot\,|\,k) = [y_p^u(k\,|\,k) \ldots y_p^u(k+N-1\,|\,k)]^T \tag{3.15b}$$

where $y_p^l(k+i\,|\,k)$ and $y_p^u(k+i\,|\,k)$ are the upper and lower limits that robustly bound the plant output at prediction time step i:

$$y_p^l(k+i\,|\,k) \le y(k)\big|_{k=k+i} \le y_p^l(k+i\,|\,k) \tag{3.16}$$

The least conservative bounding envelopes $y_p^l(k+i\,|\,k)$ and $y_p^u(k+i\,|\,k)$ can be determined as:

$$\begin{aligned} y_p^l(k+i\,|\,k) &= \min_{d(k|k), d(k+1|k), \ldots, d(k+i|k)} y(k+i\,|\,k) \\ &= \min_{d(k|k), d(k+1|k), \ldots, d(k+i|k)} E(x(k+i\,|\,k), u(k+i\,|\,k), d(k+i\,|\,k)) \end{aligned} \tag{3.17a}$$

and

$$\begin{aligned} y_p^u(k+i\,|\,k) &= \max_{d(k|k), d(k+1|k), \ldots, d(k+i|k)} y(k+i\,|\,k) \\ &= \max_{d(k|k), d(k+1|k), \ldots, d(k+i|k)} E(x(k+i\,|\,k), u(k+i\,|\,k), d(k+i\,|\,k)) \end{aligned} \tag{3.17b}$$

where the disturbance at time $k+i$: $d(k+i\,|\,k)\in[d^{\min},d^{\max}]$, $i=\overline{0:N-1}$; the states $x(k+i\,|\,k)$ are obtained from the state space equation (3.14) with known initial condition $x(k\,|\,k)$.

Generating $y_p^l(k+i\,|\,k)$ and $y_p^u(k+i\,|\,k)$ also produces the plant state bounding envelopes $x_p^l(k+i\,|\,k)$ and $x_p^u(k+i\,|\,k)$, for $i=\overline{0:N-1}$.

Since the robust output prediction is calculated over the horizon N, there are N optimization problems to be solve to find N values of $y_p^l(k+i\,|\,k)$ and $y_p^u(k+i\,|\,k)$. As i increase from 0 to $N-1$, the optimization also increase the number of variables from 1 to N.

Indeed, when $i=\overline{0:N-1}$, (3.17a) and (3.17b) have N variables $d(k\,|\,k)$, $d(k+1\,|\,k),\ldots,d(k+N-1\,|\,k)$. The more variables the optimization has, the more computing time the optimization solver requires. As these computations are carried out online, it is desired to reduce the computing time as much as possible.

3.4.1 Stepwise Robust Output Prediction (SWROP)

In previous section, solving optimization problems (3.17a) and (3.17b) gave a least conservative solution of robust output prediction (LCROP). This approach is so called exact optimization method. In contrast to the exact optimization method, it will be proposed in this section an approximated optimization method where its advantage is to reduce the optimization process computing time.

Instead of solving the optimization task with respect to $i+1$ variables $d(k\,|\,k)$, $d(k+1\,|\,k),\ldots,d(k+i\,|\,k)$, one could approximate least conservative robust output prediction (LCROP) by solving the optimization tasks (3.17a) and (3.17b) with respect to only one variable $d(k+i\,|\,k)$ while $d(k\,|\,k)$, $d(k+1\,|\,k),\ldots,d(k+i-1\,|\,k)$ are obtained from the optimization in the previous time steps. In other words, instead of simultaneously solving the optimization with respect to all disturbance inputs, a step by step optimization is applied with respect to one disturbance input at the time starting with $x_p^l(t+k\,|\,t)$ and $x_p^u(t+k\,|\,t)$.

The resulting bounding envelopes are more conservative but the computing is vastly reduced. Unfortunately, the expression (3.18a) and (3.18b) generate the ROP only for some class of systems. The question of which class of systems that is applicable to use the expression (3.19a) and (3.19b) has not been answered yet in general. The approach of using SWROP to access robust feasibility is valid only if it is guaranteed that the LCROP entirely remains inside the SWROP as described in Figure 3.4. Otherwise, the real output

may possibly violate the upper or lower constraint even though the SWROP does not as described in Figure 3.5.

In practice, there are some classes of systems that have the characteristic as depicted in Figure 3.4 while some have the characteristic of Figure 3.5. Hence, in order to avoid the situation of having robustly infeasible control input, designers in practice should take that into consideration of choosing the appropriate method to calculate the ROP.

$$y_p^l(k+i\,|\,k) = \min_{d(k+i|k)} y(k+i\,|\,k)\Big|_{d(k|k)=d^{\min}(k|k),\ldots,d(k+i-1|k)=d^{\min}(k+i-1|k)} \tag{3.18a}$$

$$y_p^u(k+i\,|\,k) = \max_{d(k+i|k)} y(k+i\,|\,k)\Big|_{d(k|k)=d^{\max}(k|k),\ldots,d(k+i-1|k)=d^{\max}(k+i-1|k)} \tag{3.18b}$$

where $d^{\min}(k+i\,|\,k)$ and $d^{\max}(k+i\,|\,k)$ can be obtained by solving:

$$d^{\min}(k+l\,|\,k) = \arg\min_{d(k+l|k)} y(k+l\,|\,k)\Big|_{d(k|k)=d^{\min}(k|k),\ldots,d(k+i-1|k)=d^{\min}(k+l-1|k)} \tag{3.19a}$$
$$\forall l \in \overline{0:i-1}$$

and

$$d^{\max}(k+l\,|\,k) = \arg\max_{d(k+l|k)} y(k+l\,|\,k)\Big|_{d(k|k)=d^{\max}(k|k),\ldots,d(k+i-1|k)=d^{\max}(k+l-1|k)} \tag{3.19b}$$
$$\forall l \in \overline{0:i-1}$$

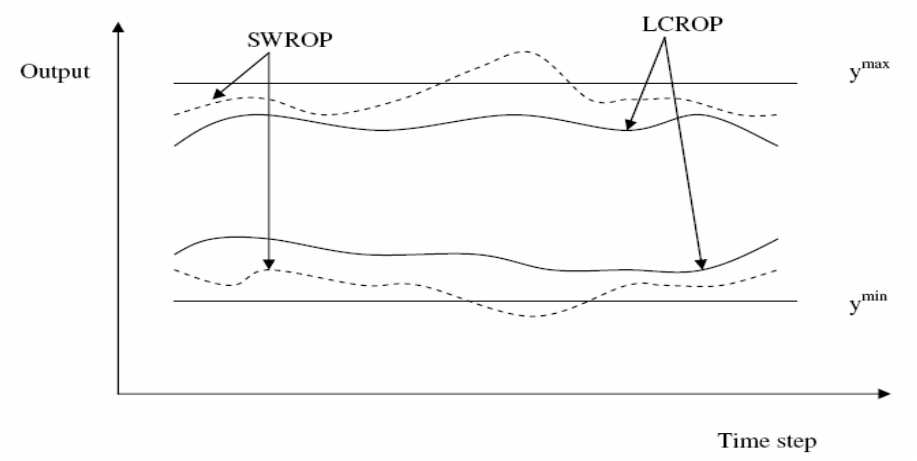


Figure 3.4 Stepwise ROP stays outside Least conservative ROP

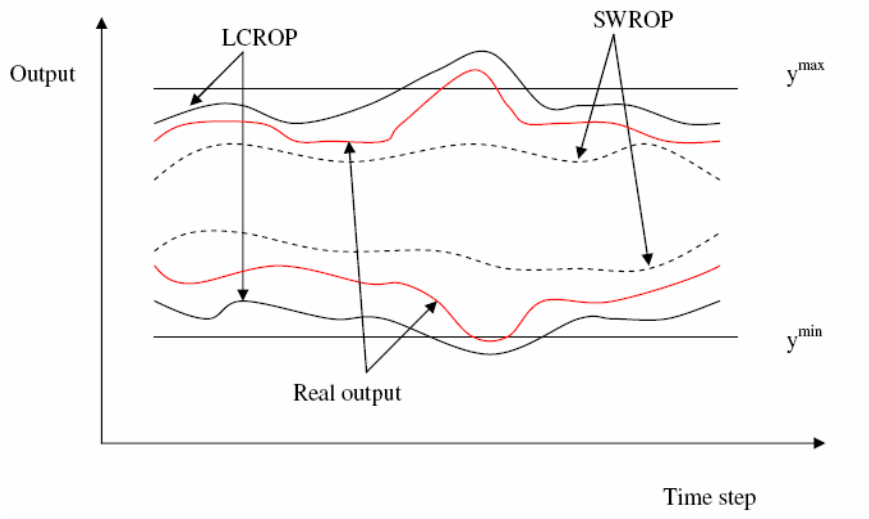


Figure 3.5 Stepwise ROP lies entirely inside Least conservative ROP

3.4.2 Reduced Robust Feasibility Horizon

So far the ROP has been considered over the whole output prediction horizon N set up for the RFMPC. This has been done in order to secure existence of the robustly feasible safety zones at any control time step. However as computing of ROP over N is computationally

very demanding this may not meet the time constraints set up by online computing requirements.

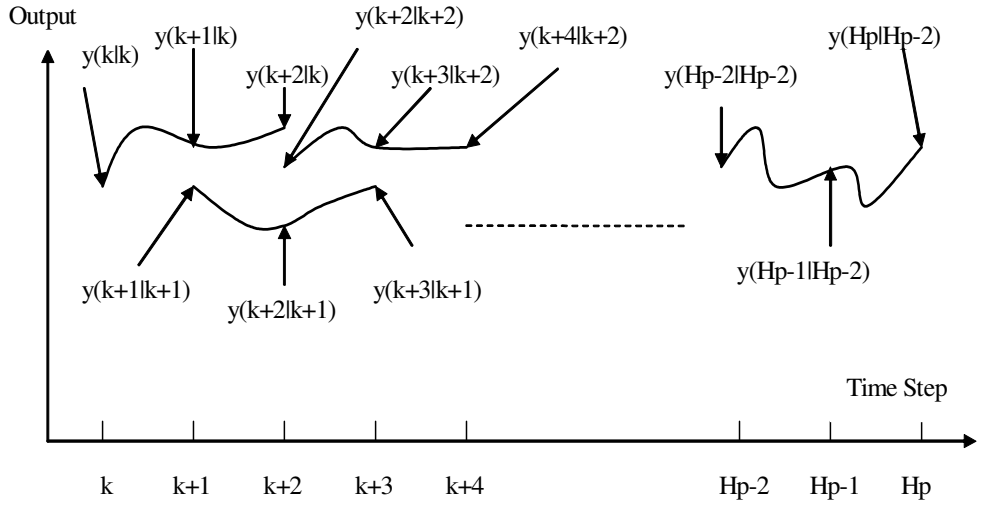


Figure 3.6 Example of reduced robust feasibility horizon to two time step - $H_r = 2$

It should be considered reducing this demand by shortening the ROP horizon. Clearly the cost to be paid is an increased risk of non existence of robustly feasible safety zones at certain control time steps. As only the first control action out of a whole sequence determined by the RFMPC is applied to the plant, the robust feasibility must be secured over the first time step. This is how far one can go with reduction of the ROP horizon from N to H_r. An attractive outcome of the ROP horizon reduction is that the very attractive computing SWROP method may become applicable over the reduced horizon while may not be applicable over the entire horizon. (see Figure 3.6).

3.5 Safety Zone Generator

Using safety zones is not a new idea to meet system constraint under unknown factors. It is widely used in engineering area, such as conservative design in many electrical devices. When the input from the nominal model base MPC controller is applied to the plant, due to the uncertainties of the system, the output constraints may not be fulfilled and their violations may be unacceptable at certain time instants. If the violation occurs, it is important to correct or modify the constraints that apply to the nominal MPC. Safety zones generator is the unit that modify the output constraints via iterative scheme. The basic idea of introducing safety zones into MPC is depicted in Figure 3.7

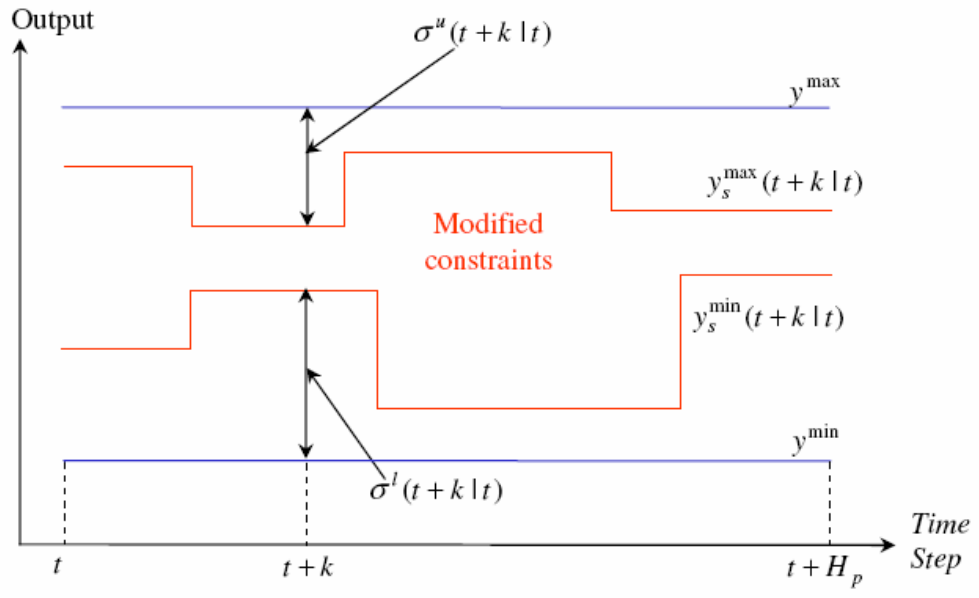


Figure 3.7 The output constraints modified by safety zones

Consider over the prediction horizon, the vectors of the lower and upper limits on the plant output:

$$Y^{\min} = [\, y^{\min} \quad \cdots \quad y^{\min} \,]^T \tag{3.20a}$$

$$Y^{\max} = [\, y^{\max} \quad \cdots \quad y^{\max} \,]^T \tag{3.20b}$$

and the vectors of the safety zones for the lower and upper output constraints, respectively where σ_i^l and σ_i^u are non negative real numbers

$$\sigma^l = [\sigma_1^l \quad \cdots \quad \sigma_N^l]^T \tag{3.21a}$$

$$\sigma^u = [\sigma_1^u \quad \cdots \quad \sigma_N^u]^T \tag{3.21b}$$

The vectors $Y_s^{\min} = Y^{\min} + \sigma^l$ and $Y_s^{\max} = Y^{\max} - \sigma^u$ are composed of the lower and upper bounds of the modified output constraints over N, respectively.

The "Safety Zones Generator" produces iteratively robustly feasible safety zones by using the following relaxation algorithm (Brdys and Chang, 2002):

Algorithm 3.2:

(i) Set $\sigma = [\sigma^l \quad \sigma^u] = 0$;

(ii) Solve MPC optimization task with modified output constraints H_p

(iii) A vector V composed of the output constraint violation over the prediction horizon is calculated as:

$$V = [V_1 \cdots V_{2H_p}]^T \triangleq \left[(Y^{\min} - y_p^l(\cdot \mid k))^T \quad (y_p^u(\cdot \mid k) - Y^{\max})^T \right]^T$$

Define $f(V_i) \triangleq \max\{0, V_i\}$ and $C(\sigma^l, \sigma^u) \triangleq [f(V_1) \cdots \quad f(V_{2H_p})]^T$

If $C(\sigma^l, \sigma^u) = 0$

is satisfied then go to step (vi),

Else go to step (iv);

(iv) Calculate the safety zone corrections by using $\delta^{(k)} = -\nu C(\sigma^{(k)})$

where $\nu = \max([diag[\nabla C(0)]]^{-1})$ is called the relaxation gain

(v) $\sigma^{(k+1)} = \sigma^{(k)} + \delta^{(k)}$, go to step (ii)

(vi) The robustly feasible safety zones have now been found and the control input $u(k \mid k)$ is applied to the plant.

Notice that at each control step the relaxation gain can be updated, by calculating the diagonal component of the gradient $\nabla C(x)$, to adapt to the online operation status. The detail of how to calculate $\nabla C(x)$ as well as the proof of convergence of the algorithm can be found in (Chang, 2002). In the obtained simulation of this thesis, pre-selected value of ν is used for simplicity reason.

3.6 MPC Optimizer

Regardless the ability to handle multivariable systems and incorporate the constraints, MPC still has some drawbacks. One of its main drawbacks is the large number of manipulated variables which requires tremendous computing efforts of the optimization solvers. Although the modern computing hardware has been growing dramatically, the computing issues still remain a challenge. Since optimization solver is required to solve MPC optimization task at every time step to produce optimal control input, effective optimization solvers certainly can improve the speed of MPC. There are several

optimization solvers that are available as commercial packages e.g. MINOS, CONOPT, SNOPT, LSGRG2. Quite often, the optimization solvers are classified into two categories based on its best effective functionality: linear and nonlinear optimization solver. While the former is used in linear systems, i.e. linear objective function and linear constraints, the latter is more efficient to handle the nonlinearities of the nonlinear systems. In spite of influence of the linear optimization solvers, our interest falls into nonlinear ones as almost every system in reality is nonlinear. In among of many available nonlinear optimization solvers, Genetic Algorithm (GA) and Sequential Quadratic Programming (SQP) are very widely used in the academia. Whereas GA is popularly used due to its simplicity to implement and ability to incorporate both continuous and discrete variables, SQP is well known due to its effective ability to handle nonlinear constraint. Another major difference between those two solvers is while GA does not require any gradient of the objective function and derivatives of the constraints, SQP is the gradient type solver i.e. user could supply suitable gradients to the solvers to achieve significant improvement on computation. Nevertheless, supplying the gradients of a function is not always a straightforward task if the function does not have the analytical expression but only its value can be obtained. Unfortunately, it is most likely the case when the MPC optimization task is formulated in the reduced space of decision variables. In order to overcome this issue, the Hamiltonian based technique to calculate the suitable gradients for the SQP is presented in this section.

3.6.1 Gradients of Objective Functions

In order to improve the computing efficiency, the formulation of the MPC optimization task in the reduced spaced of decision variables as described in the Section 3.2.2 will be used. The SQP however requires users to supply the values of gradients of the objective/performance function and constraints with respect to the decision variables i.e. control input $u(\cdot)$. As the objective function is explicitly expressed in term of $u(\cdot)$ their gradients can be easily calculated. However, as it has been pointed out in the Section 3.2 the analytical expressions of the function $J(\cdot)$ and $c(\cdot)$ are not available and their values are calculated numerically for specific values of the state initial condition $x(k \mid k)$, $u(\cdot)$, and the disturbance scenario $d(\cdot)$. In order to supply their necessary gradient values, the Hamiltonian technique will be used.

It shall be assume that the objective function $\widetilde{J}(\cdot)$ in (3.10) has the additive structure as follows:

$$\begin{aligned} J(u(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) &= \widetilde{\widetilde{J}}(G(\cdot \mid k)) \\ &= \sum_{i=0}^{N-1} Q(G(k+i \mid k)) + Q_N(x(k+N \mid k)) \end{aligned} \tag{3.22}$$

The term $Q_N(x(k+N \mid k))$ forces state to reach desired value at the end of the control horizon.

From (3.5), the following holds:

$$x(k+i+1 \mid k) = \widetilde{f}(x(k+i \mid k), s(k+i \mid k), d(k+i \mid k)), \text{ for } i = \overline{0:N-1} \tag{3.23}$$

Substituting $G(\cdot \mid k)$ from (3.9) into (3.23) yields:

$$x(k+i+1) = \widetilde{\widetilde{f}}(G(k+i \mid k)) \tag{3.23-a}$$

Recall the definition of $s = [u^T, y^T, x^T]^T$ from equation (3.1) and

$$s(\cdot \mid k) \triangleq [u(k \mid k)...u(k+N-1 \mid k)\ \ y(k \mid k)...y(k+N-1 \mid k)\ \ x(k \mid k)...x(k+N-1)]^T$$

Substituting $s(\cdot \mid k)$ into (3.23) yields:

$$x(k+i+1) = f(u(k+i \mid k), d(k+i \mid k), x(k+i \mid k)) \tag{3.23-b}$$

Combining (3.23-a) and (3.23-b) results:

$$\begin{aligned} x(k+i+1) &= \tilde{\tilde{f}}(G(k+i \mid k)) \\ &= f(u(k+i \mid k), d(k+i \mid k), x(k+i \mid k)) \end{aligned} \tag{3.24}$$

Utilizing equation (3.22) and (3.24), the mapping from $u(\cdot)$ to $J(\cdot)$ can be expressed as:

$$\begin{cases} J(u(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) = \sum_{i=\bar{t}}^{\bar{t}+N-1} Q(G(k+i \mid k)) + Q_N(x(k+N \mid k)) \\ x(k+i+1 \mid k) = \tilde{\tilde{f}}(G(k+i \mid k)) \\ i = \overline{0:N-1} \\ x(k \mid k) \text{ and } d(\cdot \mid k) \text{ are given} \end{cases} \tag{3.25}$$

An efficient algorithm for calculating $\frac{\partial J(\cdot)}{\partial u(\cdot)}$ when treating the state as the calculated variables is available and presented in (Sage, 1968), (Lewis and Syrmos, 1995), (Teo and Goh, 1991), (Fisher and Jennings, 1995), (Brdys and Tatjewski, 2005).

Let define Hamiltonian function at time instant $i = \overline{0:N-1}$

$$H(x(k+i \mid k), u(k+i \mid k)) \triangleq -Q(G(k+i \mid k)) + \mu(i)^T \tilde{\tilde{f}}(G(k+i \mid k)) \tag{3.26}$$

where $\mu(i) \in \mathbb{R}^n$

The multipliers $\mu(i)$ are determined by the conjugate equations as follows:

$$\mu(i-1)=\frac{\partial H(x(k+i\,|\,k),u(k+i\,|\,k))}{\partial x(k+i\,|\,k)},\ i=\overline{0:N-1} \tag{3.27a}$$

$$\mu(\bar{t}+N-1)=\frac{\partial Q_N(x(k+N\,|\,k))}{\partial x(k+N\,|\,k)} \tag{3.27b}$$

Finally, the derivative of the performance function $J(\cdot)$ with respect to $u(\cdot)$ equals to:

$$\frac{\partial J(u(\cdot\,|\,k),d(\cdot\,|\,k),x(k))}{\partial u(k+i\,|\,k)}=-\frac{\partial H(x(k+i\,|\,k),u(k+i\,|\,k))}{\partial u(k+i\,|\,k)},\ i=\overline{0:N-1} \tag{3.28a}$$

$$\frac{\partial J(\cdot)}{\partial u(\cdot)}=\left[\frac{\partial J(\cdot)}{\partial u(k\,|\,k)},...,\frac{\partial J(\cdot)}{\partial u(k+i\,|\,k)},...,\frac{\partial J(\cdot)}{\partial u(k+N-1\,|\,k)}\right] \tag{3.28b}$$

Notice that no iterations are required in order to calculate the derivatives. Overall, the computing process above is summarized in the form of the following algorithm:

Algorithm 3.3: Gradient Algorithm

Data: Given control input sequence $u(k+i\,|\,k),i=\overline{0:N-1}$, predicted disturbance $d(k+i\,|\,k),i=\overline{0:N-1}$, and the corresponding network state response $x(k\,|\,k)$

- Step 1: Solve the system equation (3.24) forward from $i=0,1,...,N-1$
- Step 2: Solve the system of the conjugate equations (3.27a) backward from $i=N-1,...,1,0$ with the initial point defined by the equation (3.27b).
- Step 3: Compute the gradients by (3.28)

It is worth remarking that the calculation of equation (3.27) and (3.28) requires calculating $\frac{\partial G(\cdot)}{\partial x(\cdot)}$ and $\frac{\partial G(\cdot)}{\partial u(\cdot)}$. Consequently, it means that the derivatives $\frac{\partial s(\cdot)}{\partial u(\cdot)}$ and $\frac{\partial s(\cdot)}{\partial x(\cdot)}$ also need to be calculated. By definition of s in (3.1), it turns out that in order to have $\frac{\partial s(\cdot)}{\partial u(\cdot)}$ and $\frac{\partial s(\cdot)}{\partial x(\cdot)}$, the derivatives $\frac{\partial y(\cdot)}{\partial u(\cdot)}$ and $\frac{\partial y(\cdot)}{\partial x(\cdot)}$ needs to be computed. Using the static network equations and the implicit differentiation theorem, the derivatives $\frac{\partial y(\cdot)}{\partial u(\cdot)}$ and $\frac{\partial y(\cdot)}{\partial x(\cdot)}$ are calculated. As the static network equations are determined by the nature of the network, the general formulation of calculating these derivatives is not formulated. However, for a specific case i.e. DWDS network, the calculation is explained and illustrated in the detail in Section 3.8.

3.6.2 Derivatives of Constraints

The constraints are classified into two categories: equality and inequality constraints. The equality constraints consist of all static network equations that are automatically handled by the network simulator. However the inequality constraints as described in (3.2) need to be embedded into the optimization problem. Their derivatives need to be provided to the SQP solver.

Suppose we are interested in calculating derivative of one output $y_{\bar{j}}$ with respect to $u(\cdot)$, where $\bar{j} \in \overline{1:n_y}$ and n_y is the number of outputs in the network.

As $i = \overline{0:N-1}$, for each output $y_{\bar{j}}$, there are N derivatives i.e. $\frac{\partial y_{\bar{j}}(k \mid k)}{\partial u(\cdot)}$, $\frac{\partial y_{\bar{j}}(k+i \mid k)}{\partial u(\cdot)}, \ldots, \frac{\partial y_{\bar{j}}(k+N-1 \mid k)}{\partial u(\cdot)}$ that need to be computed.

We need to find $\frac{\partial y_{\bar{j}}(k+i \mid k)}{\partial u(\cdot)}$ for $i = \overline{0:N-1}$.

The procedure to calculate $\left.\frac{\partial y_{\bar{j}}(k+i \mid k)}{\partial u(\cdot)}\right|_{i=\bar{t}}$ with $\bar{t} \in [0, N-1]$ is shown below:

For $i = \overline{0:N-1}$, let define:

$$\tilde{g}(k+i) \triangleq \begin{cases} y_{\bar{j}}(k+\bar{t} \mid k) & \text{for} \quad i = \bar{t} \\ 0 & \text{otherwise} \end{cases} \tag{3.29}$$

Instead of finding $\frac{\partial y_{\bar{j}}(k+\bar{t} \mid k)}{\partial u(\cdot)}$, we will find $\frac{\partial \tilde{g}(k+i)}{\partial u(\cdot)}$.

The *artificial objective* function $\hat{g}$ is built as follows:

$$\hat{g}(k) = \sum_{i=0}^{N-1} \tilde{g}(k+i) \tag{3.30}$$

The phrase *artificial objective* function in this context means that $\hat{g}$ has the same mathematical format as objective function $J(\cdot)$ as described in (3.25). In order to utilize the gradient algorithm, the function needs to be formulated in the appropriate format. Here, the second term in the expression of $\hat{g}$, which is the terminal state, is 0.

Utilizing (3.29) and (3.30), the following holds

$$\frac{\partial y_{\bar{j}}(k+\bar{t}\,|\,k)}{\partial u(\cdot)} = \frac{\partial \tilde{g}(k+i)}{\partial u(\cdot)}\bigg|_{i=\bar{t}} = \frac{\hat{g}(k)}{\partial u(\cdot)} \tag{3.31}$$

Combining the state equation in (3.24) and (3.30) yields the state space model:

$$\begin{cases} \hat{g}(k) = \sum_{i=0}^{N-1} \tilde{g}(k+i) \\ x(k+i+1\,|\,k) = \tilde{\tilde{f}}(G(k+i\,|\,k)) \\ i = \overline{0:N-1} \\ x(k\,|\,k) \text{ and } d(\cdot\,|\,k) \text{ are given} \end{cases} \tag{3.32}$$

Applying the gradient algorithm (Algorithm 3.3), the derivative $\frac{\hat{g}(k)}{\partial u(\cdot)}$ is computed as follows:

Define the Hamiltonian function at time instant $i = \overline{0:N-1}$:

$$\tilde{H}(x(k+i\,|\,k), u(k+i\,|\,k)) \triangleq -\tilde{g}(k) + \tilde{\mu}(i)^T \tilde{\tilde{f}}(G(k+i\,|\,k)) \tag{3.33}$$

where the multiplier $\tilde{\mu}(i)$ is calculated by the conjugate equations:

$$\tilde{\mu}(i-1) = \frac{\partial \tilde{H}(x(k+i\,|\,k), u(k+i\,|\,k))}{\partial x(k+i\,|\,k)}, \qquad i = \overline{0:N-1} \tag{3.34a}$$

$$\tilde{\mu}(N-1) = 0 \tag{3.34b}$$

Finally, the derivative of the performance function $\hat{g}(k)$ with respect to $u(\cdot)$ equals to:

$$\frac{\partial \hat{g}(k)}{\partial u(k+i\,|\,k)} = -\frac{\partial H(x(k+i\,|\,k), u(k+i\,|\,k))}{\partial u(k+i\,|\,k)}, \quad i = \overline{0:N-1} \tag{3.35a}$$

$$\frac{\partial \hat{g}(k)}{\partial u(\cdot)} = \left[\frac{\partial \hat{g}(k)}{\partial u(k \mid k)}, ..., \frac{\partial \hat{g}(k)}{\partial u(k+i \mid k)}, ..., \frac{\partial \hat{g}(k)}{\partial u(k+N-1 \mid k)} \right] \tag{3.35b}$$

Substituting $\frac{\partial \hat{g}(k)}{\partial u(\cdot)}$ from (3.31) into (3.35b) yields:

$$\frac{\partial y_{\bar{j}}(k+\bar{t} \mid k)}{\partial u(\cdot)} = \left[\frac{\partial \hat{g}(k)}{\partial u(k \mid k)}, ..., \frac{\partial \hat{g}(k)}{\partial u(k+i \mid k)}, ..., \frac{\partial \hat{g}(k)}{\partial u(k+N-1 \mid k)} \right] \tag{3.36}$$

The procedure above is applied for only one output $\frac{\partial y_{\bar{j}}(k+\bar{t} \mid k)}{\partial u(\cdot)}$. The same procedure is repeated for all remaining outputs for $i = \overline{0:N-1}$. In other words, the Hamiltonian functions are iteratively constructed as many times as number of output constraints is. Similar to the gradient of the objective function, the calculation in (3.34) and (3.35) also requires computing the derivatives $\frac{\partial y(\cdot)}{\partial u(\cdot)}$ and $\frac{\partial y(\cdot)}{\partial x(\cdot)}$.

3.7 Simulation Environment Implementation

The RFMPC structure is applied to the DWDS application and verified the simulation results. This is implemented by the computer based simulation integrating the EPANET water network simulator and GA or SQP optimization solver under MATLAB environment.

3.7.1 Optimization solver

There are two optimization solvers that are used to obtain the simulation results: Genetic Algorithm (GA) and Sequential Quadratic Programming (SQP). Although both of them are widely used in the academia, they still have differences in the way of handling constraints. A brief introduction between those two solvers is described belows.

GA is based on the principles of genetics and natural selection and was original proposed by (Holland, 1975) and further developed by (Goldberg, 1989) and (Deb, 2000). GA represents one of the most commonly employed natural optimization techniques for design of water distribution networks as evidenced by use of GA for sizing of pipes e.g. (Savic and Walters, 1997), (Wu and Simpson, 2001), evaluation of system reliability (Tolson et al, 2004) and placement of early warning detection sensors (Ostfeld and Salomons, 2004).. Genetic algorithms are applicable to a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear (Haestad 2003). In addition, since GA does not require a gradient, it can be linked with the hydraulic/water-quality mode.

The algorithm begins with random population of individuals or users-selected population in which each individual is represented by a binary string (i.e., chromosome) for one possible solution. For each population generation, a measure of the fitness in regards to the objective is calculated. Based on the fitness value, individuals are selected to create the next generation through the use of techniques such as inheritance, mutation, natural

selection, and recombination (crossover). Individuals with higher fitness values will have a greater probability of being selected to produce the next generation, thus on average the new generation will have a higher fitness value than the older population. The algorithm continues until one or more of the pre-established criteria (e.g., number of generations, time limit, fitness limit, stall generations, stall time limit, and fitness tolerance) are met.

In spite of the simplicity to implement and easy to understand, GA is especially struggle in handling the nonlinear constraints and network with high number of variables, for example over 100 variables. However, in quantity control, the DWDS network is highly nonlinear due to its nonlinear head-flow relationships for pumps and pipes. SQP turns out to be the effective solver to handle the nonlinear constraints due to its ability to take gradients of objective function and derivative of constraints.

SQP methods represent the state of the art in nonlinear programming methods In (Schittkowski, 1985) it has implemented and tested a version that outperforms every other tested method in terms of efficiency, accuracy, and percentage of successful solutions, over a large number of test problems. Based on the work of (Biggs, 1975), (Han, 1977), and (Powell, 1978), the method allows users to closely mimic Newton's method for constrained optimization just as is done for unconstrained optimization. At each iteration, an approximation is made of the Hessian of the Lagrangian function using a quasi-Newton updating method. This is then used to generate a quadratic programming sub-problem whose solution is used to form a search direction for a line search procedure. Mode detail of SQP can be found in many texts for example: (Fletcher, 1987), (Gill et al., 1981),

(Powell, 1983), and (Hock and Schittkowski, 1983), (Antoniou and Lu, 2007), (Griva et al., 2009).

GA as well as SQP is available in many commercial packages. The most commonly used package probably is MATLAB which is also used in the thesis. GA and SQP are embedded in MATLAB as toolboxes and can be called easily by the syntax *ga* and *fmincon*, respectively.

3.7.2 Water system simulator: EPANET

EPANET is a well-known water network simulation software package published in 2000 by the National Risk Management Research Laboratory of United State Environment Protection Agency (Rossman, 2000). The product is an open source software and is widely accepted and used in simulation and design of hydraulic behaviour within pressurised pipe networks. Its graphic interface with WINDOWS operating systems makes it convenient to construct the distribution network, calibrate and tune the coefficients of the network, run the simulation, and obtain the result data.

Water leakage can be modelled in EPANET simulator by orifice functions. Leaks at specific nodes are represented using emitters that are governed by the orifice relationship, which is equivalent to May's pressure-leakage equation previously presented in Section 2.4.2 with the pressure exponent γ equals 0.5 (Rossman, 2000).

However, EPANET simulator cannot be used alone to meet the objectives of different control strategies, since its hydraulic calculation is not based on optimisation. In this thesis, EPANET simulator is used in two events. Firstly, it is used to generate the water network data, including tank level, nodal head and pipe flow, etc. The generated hydraulics output is stored in a flat report file, which can be read conveniently in MATLAB environment to update the initial state of the model-based predictive controllers. Secondly, in order to reduce numbers of decision variables, it is used to handle equality constraints of the MPC optimization task. More explanation can be found in Figure 3.8 of Section 3.7.3

3.7.3 Simulation Environment Implementation

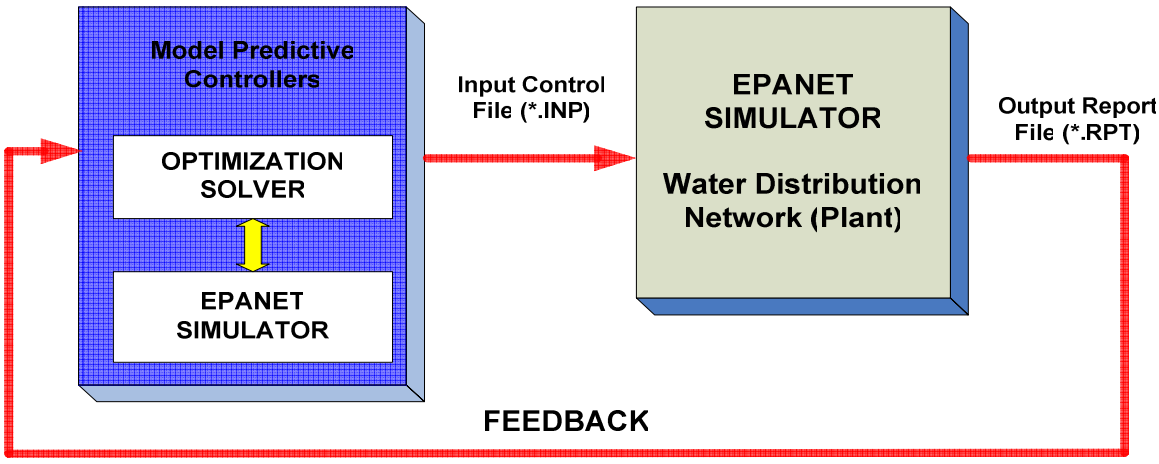


Figure 3.8: Simulation environment implementation

Leaky and non-leaky network configurations for the example water distribution network are stored in separate network files, which are given in Appendix D. For every hour time step the EPANET simulator generates massive output data in which only the tank level is fed back to the model predictive controller block for next time step use. The model

predictive controller block uses the selected optimization solvers, i.e. GA or SQP, through MATLAB to solve the MPC optimization task in the reduced space of decision variables. The EPANET simulator is embedded into the optimization solver to achieve the reduced space form. The generated optimal pump scheduling actions are transferred to the EPANET simulator in simple flat files for control purposes.

3.7.4 Main Specifications of the software and hardware in the simulation

The main features of the simulation platform are listed below:

EPANET:	Version 2.0
MATLAB:	Version 7.6.0.324 (R2008a)
OPTIMIZATION TOOLBOX:	Version 3.0.1
GA TOOLBOX:	Version 2.3 (2008a)
CPU/Memory:	2.8 GHz/2024 Mbytes
Operating System:	Windows XP Professional

3.8 DWDS Case Study

This section presents an optimal operational control of a drinking water distribution system. The case study is used to illustrate theory described in all previous sections of this

chapter. The network topological structure of the example water distribution system is illustrated in Figure 3.9. The system includes 1 source reservoir, 6 pipes, and 1 storage tank. Water is pumped from the reservoir source (node 1) by the pump station and can also be supplied by the storage tank (node 7). The assumed positive flow direction is expressed by the arrow of the flow shows, e.g. flow q_2 denotes the positive flow direction is from Node 2 to Node 3. The negative value of the flow indicates the flow direction is opposite to the assumed positive flow direction, e.g. $q_2 < 0$ means the flow direction is from Node 3 to Node 2.

The example network is depicted in Fig. 2, where q_i and $h_i, i = 1,...,7$ are the pipe flows and the nodal heads respectively. d_2, d_3, d_4, d_5, d_6 are the water demands at the consumption nodes 2, 3, 4, 5, and 6 respectively, and ps is denoted the relative pump speed. The network elements are described by the head-flow qualities. Due to its storage capabilities the tank head-flow relationship is dynamic as opposed to the other elements which are static, hence described by the nonlinear algebraic equation (3.4). The control, state, output, and disturbance variables are: $u = ps$, $x = h_7$, $y = [q, h]$, where $q = [q_1,...,q_7]$ and $h = [h_2,...,h_6]$, $d = [d_2,...,d_6]$. The head of the source is known $h_1 = 15$ and assumed to be constant over times. Given $h_7(k)$ and $ps(k)$, the forced output $y(k)$ can be obtained by solving the set of hydraulic equalities (3.4). The operational tank limits are: $x^{\max} \le 8.5[m]$ and $x^{\min} \ge 2[m]$. The relative pump speed is constrained as $0.2 = u^{\min} \le u(k) \le u^{\max} = 1.1$. Pressure of node 3 is constrained as $14 \le h_3(k) \le 19$.

For operational control purposes of DWDS, the configuration data of the above water network are given in Table 3.1 – Table 3.5, which include the nodal elevation, nodal base demand, the operating constraints for nodal pressure, daily nodal demand profile, tank initial level and level range, pipe and pump installation data, and time dependant electricity tariff.

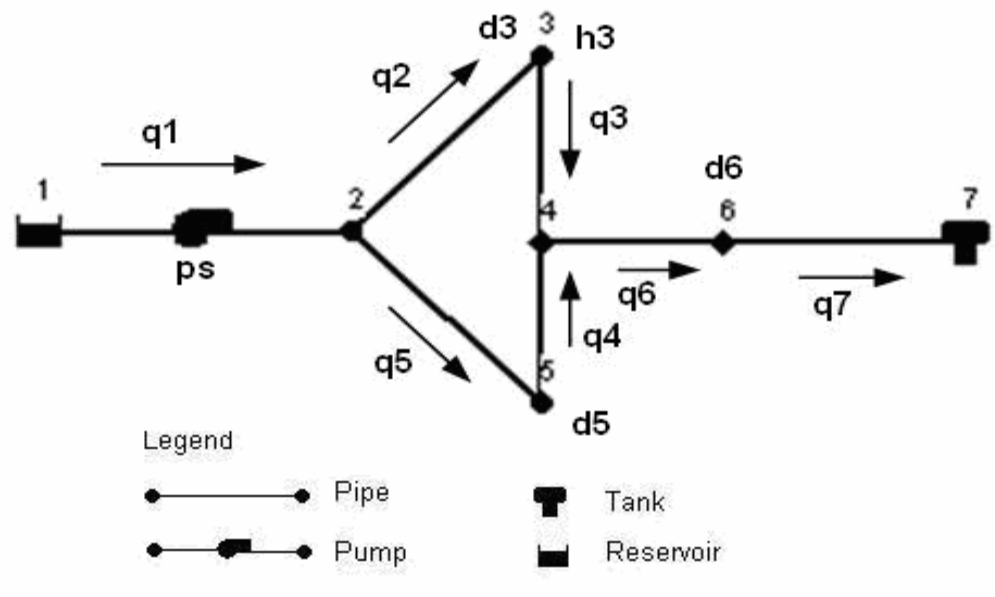


Figure 3.9: Case study – diagram of an example water distribution system

The nodal demand pattern as shown in Figure 3.10 is repeated for the extended time simulation exceeding 24 hours, which is commonly applied in simulation study of operational control of water supply/distribution systems (Vairavamoorthy and Lumbers, 1998) and is also consistent to the default configuration of the EPANET simulator. Techniques to improve the accuracy of demand prediction have been discussed in (Brdys and Ulanicki, 1994)

Table 3.1: Nodal data for the pipe network

Node ID	Elevation (m)	Minimum head (m)	Maximum head (m)	Base demand (l/s)
2	15.0	18.0	32.0	5.0
3	14.0	16.0	30.0	5.0
4	12.0	12.0	28.0	5.0
5	14.0	14.0	30.0	5.0
6	8.0	10.0	28.0	30.0
1	5.0			
7	—— Tank/Reservoir nodes ——			

Table 3.2: Tank (Reservoir) data of the example DWDS

Node ID	Elevation (m)	Initial level above bottom (m)	Min level above bottom (m)	Max level above bottom (m)	Tank diameter (m)
7	10.0	5.0	2.0	10	15.0
1	15.0	—— Source —			

Table 3.3: Pipe data of the example DWDS

Pipe ID	Start node	End node	Length (m)	Diameter (mm)	C Value
203	2	3	1000	400.0	100
205	2	5	1000	400.0	100
304	3	4	1000	400.0	100
504	5	4	1000	300.0	100
406	4	6	1000	500.0	100
607	6	7	1000	500.0	100

Table3.4: Pump data of the example DWDS

Pump ID	Head node	Tail node	Coefficient $\bar{A}$ (10^{-2})	Coefficient $\bar{B}$	Cut-off head $\bar{C}$ (m)
112	1	2	-0.5419	0	200

Table 3.5 Pump efficiency and electricity tariff

Pump ID	Efficiency η	Energy price in £/kWh	
		6:00 a.m. – 10:00 p.m. (high tariff)	10:00 p.m. – 6:00 a.m. (low tariff)
112	0.8	9.72p	4.51p

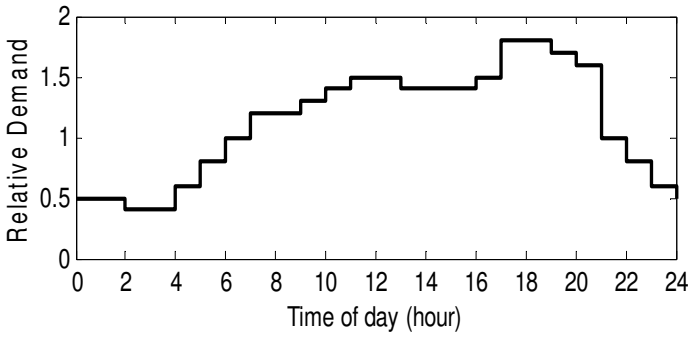


Figure 3.10: Daily demand profile

3.8.1 Formulation of the predictive control problem

In daily operation of water distribution systems, a period of water demand prediction ahead of current time is usually needed to be the basis for generating optimal pump actions so as to achieve certain control objective, e.g. least pumping cost. Since too long or infinitive time horizon demand prediction is not accurate or unavailable, a relatively short prediction horizon is more realistic, and this is applied in a receding horizon manner, which forms the key idea of MPC technique.

The main goal of DWDS is to supply water to customers and satisfy their quantity and quality demand. There are two major aspects in the control of DWDS: quantity and quality. The quality control deals with water quality parameters. Having disallowed concentration

of chemical parameter, for instance chlorine, cause serious heath dangers. Maintaining concentrations of the water quality parameters within the prescribed limits throughout the network is a major objective. When the quantity control is considered, the objective is to minimize the electrical energy cost of pumping, while satisfying consumer water demand and physical constraints such as pressure at nodes or reservoir levels, by producing optimized control input such as optimized pump speeds and valve control schedules (Brdys and Ulanicki, 1994). The uncertainty is in the demand and structure and parameters of DWDS model. In the case study, only the quantity control aspect is considered by applying RFMPC technique. The quality issues are addressed in (Brdys and *et al.*, 2002), (Chang and Brdys, 2002) for example. In the formulation of the predictive control, the corresponding optimization problems are formulated with a moving horizon $H_p = 24$ hours and the sampling period is fixed to 1 hour.

It is a very common control objective to achieve the least pumping cost while satisfying constraints. Moreover, in order to achieve a sustainable operation day after day, it is expected that tank levels can come back to their original states after a certain period. For the DWDS example, it is desired that after 24 hours, the tank level could return to the similar level, which makes long-term periodical operation possible. Hence, the overall objective function at $k = k_0$:

$$J_c = \sum_{k=k_0}^{k_0+N-1} \frac{\gamma(k)}{\eta(k)} q_1(k)(h_2(k) - h_s(k)) + \rho \left| h_7(k_0+N) - h_7(k_0) \right| \tag{3.37}$$

where $\gamma(k)$ is the power unit charge in £/kWh, $\eta(k)$ is the pump efficiency and is set $\eta(k) = 0.8$ for all k, and $h_s(k)$ is the head of the source and it set $h_s(k) = 5$ for all k.

The optimization constraints are composed of:

- Nodal flow continuity equations
- Water elements head-flow equations
- Volume/mass balance equations of tanks
- Pressure limits at certain nodes
- Physical tank limit
- Relative speed limit

3.8.2 Design and Simulations (Part 1)

3.8.2.1 Design

In this part, GA is selected to be the optimization solver to solve MPC optimization task in the reduce space form. The iterative control structure RFMPC is applied to the operational control of the DWDS example. In order to reduce computing time, SWROP and shortening robust feasibility horizon technique described in Section 3.4 are applied.

A method for generating robust output prediction (ROP) is chosen by observing the simulation results shown in Figure 3.11. The stepwise ROP (SWROP) and the least conservative ROP (LCROP) are applied at $k_0 = 0$ over 7 time steps. It can be seen in Figure 3.11 that the SWROP method generates envelopes that are outside the region determined by the LCROP method. Hence, the SWROP is applicable to our example DWDS. Moreover, the envelopes calculated by the two methods are very close over the

first 6 steps. The ROP horizon therefore is further reduced to 2 steps and the SWROP method is to be applied.

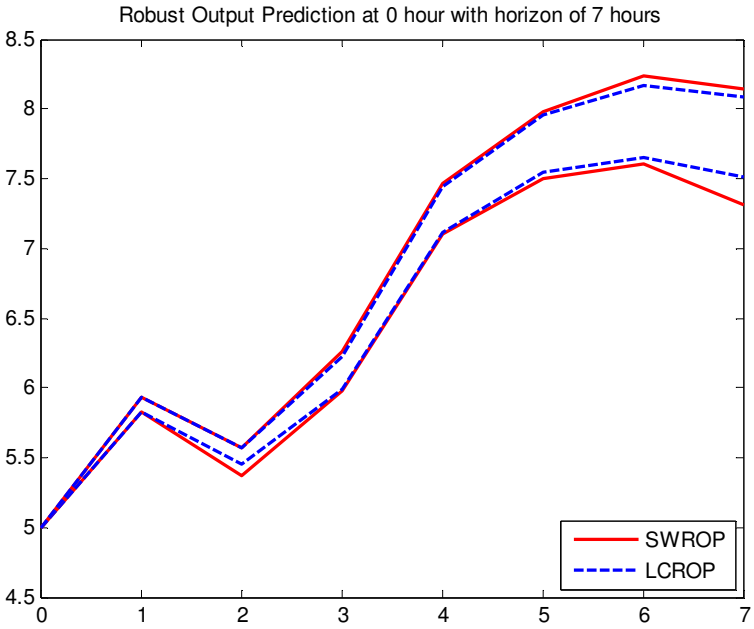


Figure 3.11: Robust output prediction at $k_0 = 0$, $H_p = 7$

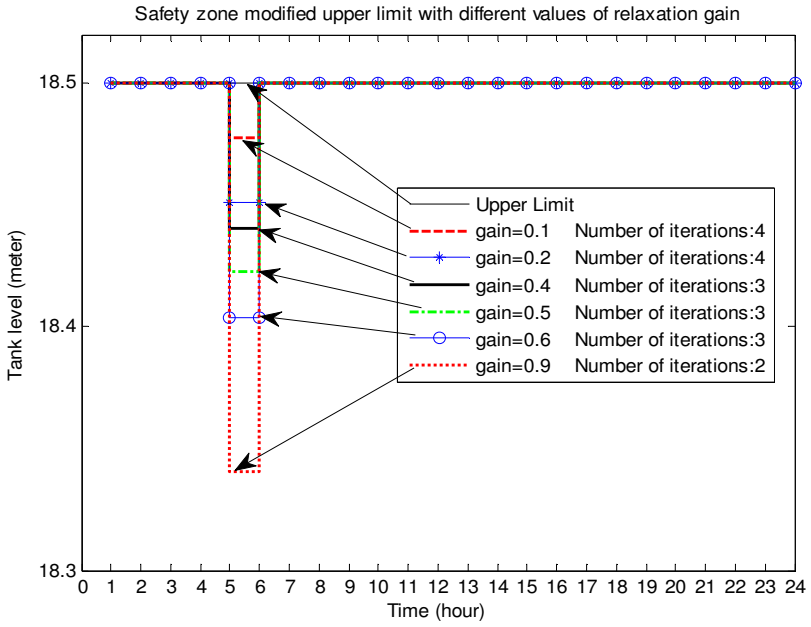


Figure 3.12: Robustly feasible safety zones and the corresponding modified tank upper limit for different relaxation gain values.

Furthermore, the relaxation gain in the algorithm for determining the robustly feasible safety zones (RFSZ) is selected by simulation where several gain values are tried and the results are illustrated in Figure 3.12 The equality (8) in the step (iii) of the RFSZ relaxation algorithm has more than one solution. Clearly, the smaller safety zones are, the less conservative control actions are, and consequently better controller performance is achieved. From Figure 3.12 this is obtained for small gain values. On the other hand, the computing time is essential; hence the number of iterations needed to reach the RFSZ should be minimized. This is obtained for high gain values as described in Figure 3.12. Therefore, in author's opinion, gain $\nu = 0.6$ is chosen in order to trade between the two aspects.

3.8.2.2 Simulation results

First the RFMPC is applied to the example DWDS at $k_0 = 0$. Robust feasibility at the obtained control sequence is checked over the horizon $H_r = 2$ and the first two control inputs are assessed as robustly feasible. Hence, there is no need to activate the "Safety Zone Generator". In Figure 3.13 two tank trajectories are illustrated: one in dash line is obtained by applying the control sequence to the model with the demand prediction while the second one in solid is the tank trajectory seen in the real system where the demand may differ from the predicted one up to 10%. It also can be seen in Figure 3.13 that the upper limit tank constraint is violated during 5 hour to 7 hour time period. Clearly, we are not aware about this violation at $k_0 = 0$. However, a lesson to be learnt is that applying a

whole control sequence obtained at $k_0 = 0$ to the network is not recommended not only in this case but in general. Therefore the RFMPC is kept applying to produce the control actions on-line by employing feedback and all its mechanism described in this chapter.

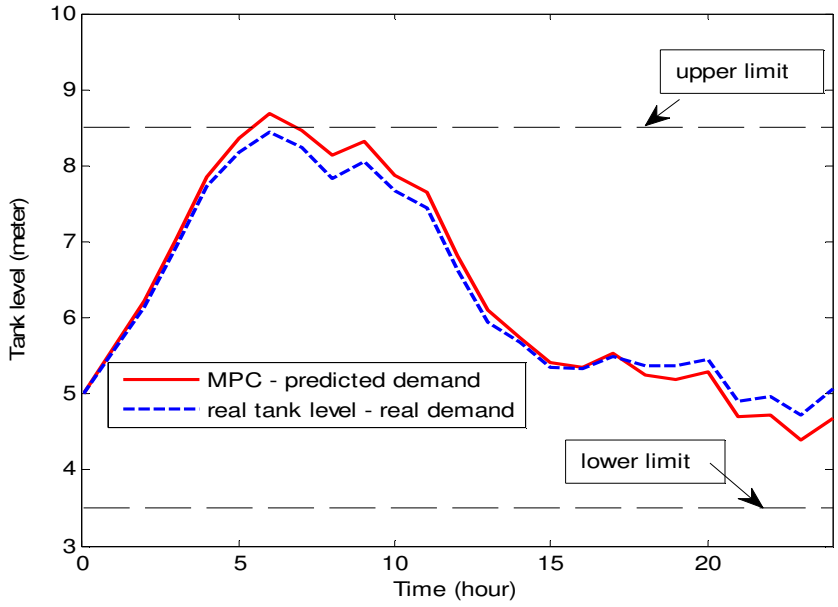


Figure 3.13: Predicted tank level trajectory by RFMPC over the horizon at time instant $k_0 = 0$ and $H_r = 2$

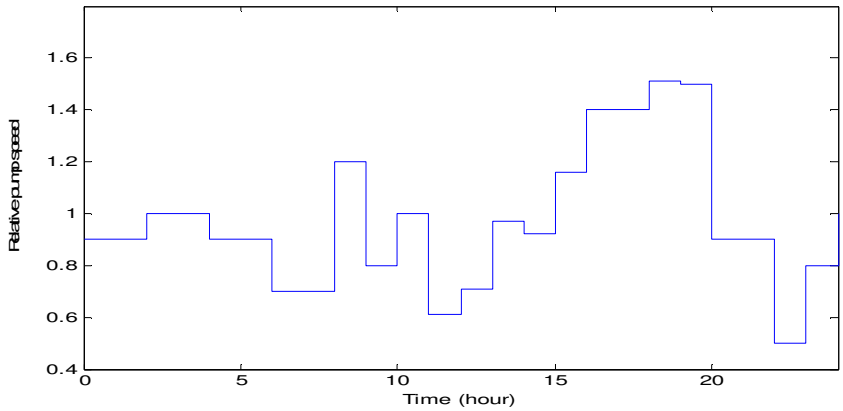


Figure 3.14: Control actions - relative pump speed

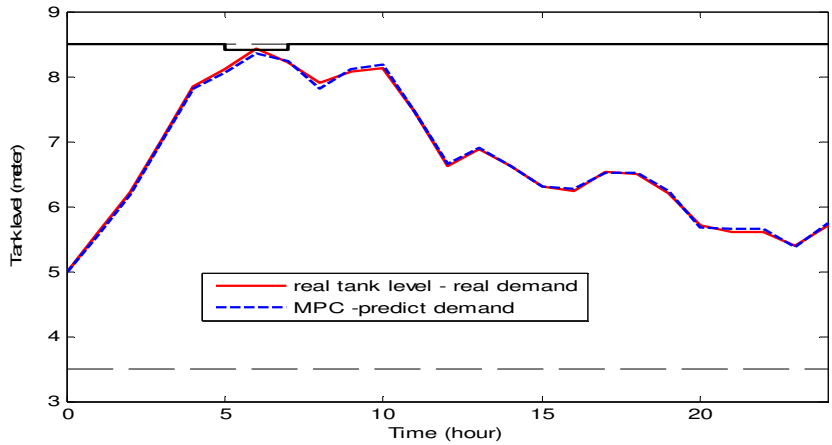


Figure 3.15: Tank trajectory over the 24 hours

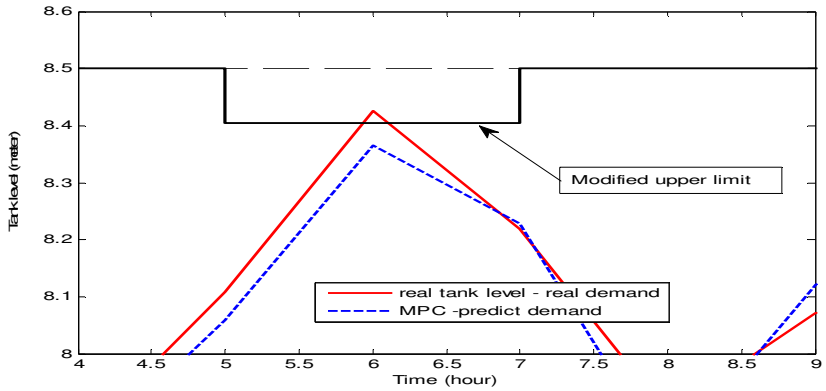


Figure 3.16: Zoom-in of Fig.10 during 4-9 hours

The results are illustrated in Figure 3.14-3.16. It can be seen in Figure 3.15 that the upper tank level constraint had to be modified by robustly feedback safety zones over 5, 6, and 7 time steps in order to achieve robust feedback of the control action over these time steps. Although the modification does not tighten the constraints excessively its conservatism would be improved by extending the robust prediction horizon. The details of the situation over 5, 6, and 7 time steps are illustrated in Figure 3.16.

In order to assess the RFMPC feedback strength, the control actions generated on-line are also applied to the DWDS model. The resulting tank trajectory and the control input are

shown in Figure 3.15 and Figure 3.14, respectively. The two trajectories are much closer in Fig.15 than in Fig.13. Hence, possible impact of the feedback in compensating the demand error impact is noticeable.

Lastly, as shown in Figure 3.16, the modified constraints are satisfied in the model but not in reality. However, the actual constraint is met in reality showing the effectiveness of the RFSZ mechanism.

3.8.3 Design and Simulations (Part 2)

In this section, further improvement on the computing time is considered. In stead of using GA, Sequential Quadratic Programming (SQP) is chosen to be the optimization solver due to its effective ability to handle the nonlinear constraints. The gradient algorithm (Algorithm 3.2) is utilized to provide the suitable gradients to the SQP solver. The comparison between GA and SQP is verified by the simulation.

3.8.3.1 Design

The gradient of the objective function can be found by using the procedure described in Section 3.6.1. The mapping from $u(\cdot)$ to $J(\cdot)$ can be expressed by equation (3.25), where the control input $u(\cdot) = ps(\cdot)$ is the relative pump speed, state $x = h_7$, prediction horizon $N = 24$. The operator G has been defined in the equation (3.9). In this situation, G is the water network simulator i.e. EPANET. It is consistent with the definition in the equation

(3.9) as the relative pump speed schedule, predicted demand, and initial tank level are given the EPANET simulator results all the values of nodal heads, flows.

Following the steps (3.26), (3.27), (3.28) eventually requires the calculation of $\frac{\partial y(\cdot)}{\partial u(\cdot)}$ and $\frac{\partial y(\cdot)}{\partial x(\cdot)}$. Recall that $y=[q,h]$, $q=[q_1,...,q_7]$, $h=[h_2,...,h_6]$

Therefore, the task is now to calculate $\frac{\partial q(\cdot)}{\partial u(\cdot)}$, $\frac{\partial q(\cdot)}{\partial h_7(\cdot)}$, $\frac{\partial h(\cdot)}{\partial u(\cdot)}$, $\frac{\partial h(\cdot)}{\partial h_7(\cdot)}$

Since the DWDS is assumed to be non-leaky, the lq in the equation (2.34) is zero and we have:

$$\begin{cases} \Lambda_c q = d \\ \Lambda_c^T h + \Psi(q) = -\Lambda_r^T h_r \end{cases} \tag{3.38}$$

There are total 7 branches, 5 junction nodes, and 1 reservoir node. Therefore, $b=7$, $n_c=5$, $n_r=1$. The incidence matrix Λ_c has $n_c=5$ rows and $b=7$ columns and the matrix Λ_r has $n_r=1$ rows and $b=7$ columns.

Rearrange (3.38) yields:

$$\begin{cases} \Lambda_c q - d = 0 \\ \Lambda_r^T h_r + \Lambda_c^T h + \Psi(q) = 0 \end{cases} \tag{3.39}$$

The set of equation (3.39) in short can be put into vector function $\Omega(q,h,h_r,u)=0$ where vector Ω has 1 column and 12 rows. In details, $\Omega(q,h,h_r,u)$ is resulted as below:

$$\begin{aligned}\Omega(q,h,h_r,u) = [&q_1 - q_2 - q_5 + d_2,\ \ q_2 - q_3 - d_3,\ \ q_3 + q_4 - q_6 + d_4,\\ &q_5 - q_4 + d_5,\ q_6 - q_7 + d_6,\ h_2 - h_1 - (Aq_1^2 + Bq_1u + Cu^2),\\ &h_2 - h_3 - R_2 q_2 |q_2|^{0.0852},\ \ h_3 - h_4 - R_3 q_3 |q_3|^{0.0852},\ \ h_5 - h_4 - R_4 q_4 |q_4|^{0.0852}\\ &h_2 - h_5 - R_5 q_5 |q_5|^{0.0852},\ \ h_4 - h_6 - R_6 q_6 |q_6|^{0.0852},\ h_6 - h_7 - R_7 q_7 |q_7|^{0.0852}]^T = 0\end{aligned} \tag{3.40}$$

Notice that in (3.40), the h_1 is the head of source and is always known over times.

Applying the implicit differentiation theorem, the following holds:

$$\frac{\partial \Omega(q,h,h_r,u)}{\partial u} + \frac{\partial \Omega(q,h,h_r,u)}{\partial q}\frac{\partial q}{\partial u} = 0$$

$$\frac{\partial q}{\partial u} = -\left(\frac{\partial \Omega(q,h,h_r,u)}{\partial q}\right)^{-1} \frac{\partial \Omega(q,h,h_r,u)}{\partial u} \tag{3.41a}$$

Similarly, we have:

$$\frac{\partial q}{\partial h_r} = -\left(\frac{\partial \Omega(q,h,h_r,u)}{\partial q}\right)^{-1} \frac{\partial \Omega(q,h,h_r,u)}{\partial h_r} \tag{3.41b}$$

$$\frac{\partial h}{\partial u} = -\left(\frac{\partial \Omega(q,h,h_r,u)}{\partial h}\right)^{-1} \frac{\partial \Omega(q,h,h_r,u)}{\partial u} \tag{3.41c}$$

$$\frac{\partial h}{\partial h_r} = -\left(\frac{\partial \Omega(q,h,h_r,u)}{\partial h}\right)^{-1} \frac{\partial \Omega(q,h,h_r,u)}{\partial h_r} \tag{3.41d}$$

Having $\frac{\partial q(\cdot)}{\partial u(\cdot)}$, $\frac{\partial q(\cdot)}{\partial h_7(\cdot)}$, $\frac{\partial h(\cdot)}{\partial u(\cdot)}$, $\frac{\partial h(\cdot)}{\partial h_7(\cdot)}$ calculated, the gradients of objective function is provided to the SQP solver. Applying the same procedure in Section 3.6.2 and results of (3.41), the derivatives of the constraints are computed.

3.8.3.2 Simulation results

The control objective is to minimizing the pumping cost. RFMPC is applied to DWDS by GA and SQP separately. Comparative simulation results between GA and SQP are shown by Figure 3.17 – 3.24 in sequence for relative pump speed, tank level, several pipe flows, and nodal pressures. In Figure 3.18, the tank level comes back to around its initial state after 24-hour operation. However, with SQP solver the tank level comes back to its initial state closer than with GA. As shown in the Figure 3.17, the pump speed schedules produced by two different solvers might not look very similar, but they both utilize the capacitance of the tank during low tariff period. This fact is also confirmed by Figure 3.24 where the flow of tank takes negative values during low tariff. In this simulation, the electrical pumping cost is computed by Eq (3.37) where following assumption is made: the power unit charge $\gamma(k)=1$ for all k, the pump efficiency $\eta(k)=0.8$ for all k, the head of the source $h_s(k)=5$ for all k, and weight $\rho=500$. The simulation shows that electricity cost achieved by SQP is smaller than by GA. Specifically, the electrical pumping cost achieved by SQP is in one day period is 1.625e+04 whereas it is 1.831e+04 by GA. In comparison, the electrical cost is saved by 11.3% by SQP.

Although SQP outperforms GA in computational time, it also has disadvantages. Specifically, SQP easily gets into local optimal solution and it requires user to supply the initial search point. The initial search point has a significant impact on the solution of SQP. By having an initial search point around the optimal region, SQP efficiently converges to the optimal solution much faster than GA (Behrang *et al.*, 2008). Regardless the fact that GA converges to the solution very slowly especially in final generations in which the

objective function is very close to the optimal solution, one of the main advantages of GA is to be able to operate without supplying the initial search point. Moreover, GA searches a wide range of population and it is better than SQP in finding global optimal solution. In order to combine the advantage of both optimization solvers, in this section GA is used to generate the simulation results first. Then, the control inputs (pump speed) generated by GA is used as the initial search point in SQP solver. By doing so, the optimality is improved as well as the computing efficiency.

In term of computation aspect, the average computing time of solving a single MPC optimization task in this case study by GA is 650 seconds, whereas it takes SQP only 320 seconds with the same hardware and software setup in Section 3.7.4. The overall computational time to achieve the simulation in this section is about 3.5 hours by SQP and 7.5 hours by GA which is confirming the convergence speed of SQP over GA. This is due to the fact that SQP utilises the gradient information while GA does not.

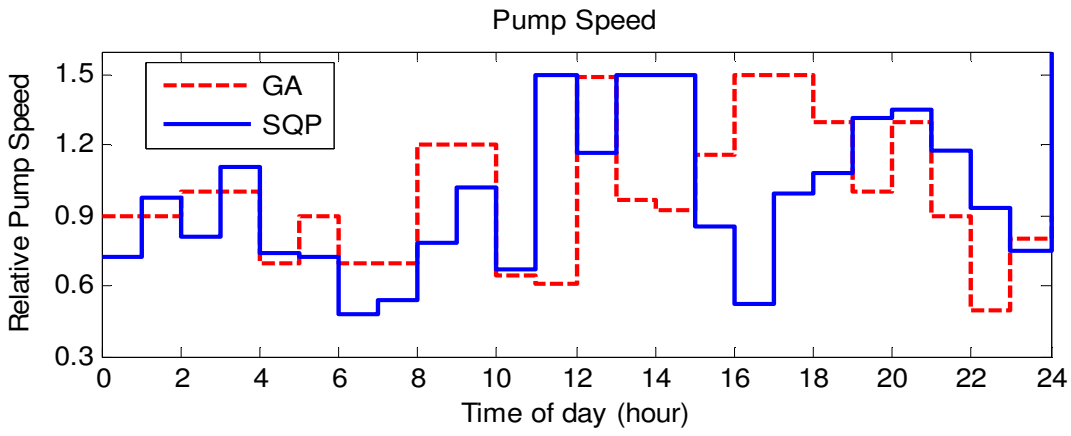


Figure 3.17 Control input – Relative pump speed by using GA and SQP to minimize pumping cost

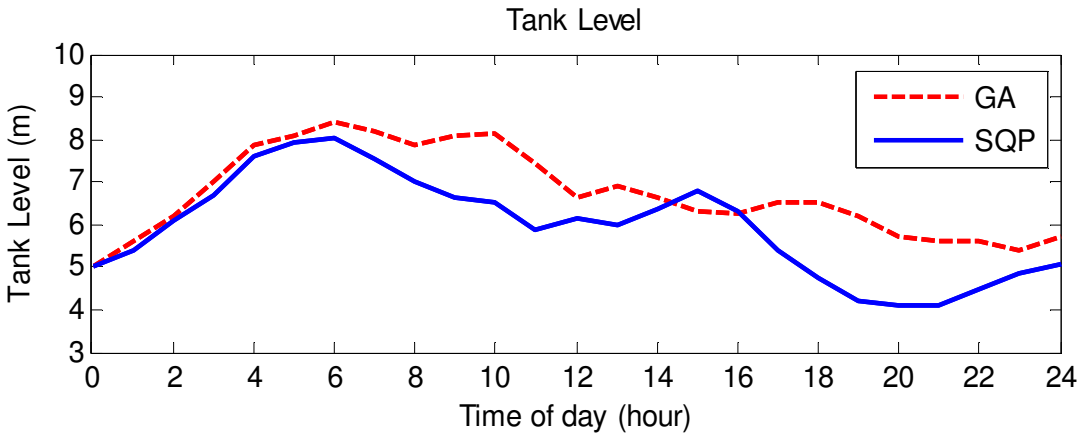


Figure 3.18 Tank level by GA and SQP

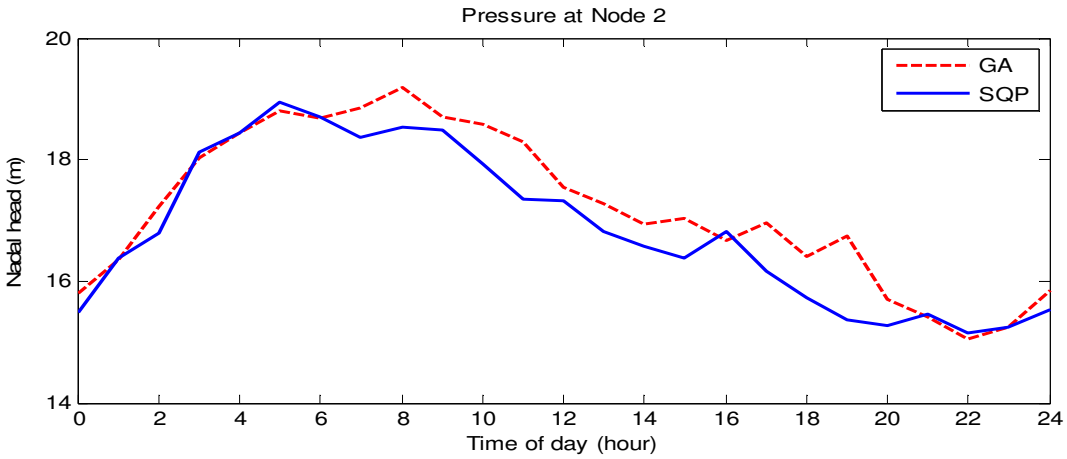


Figure 3.19 Pressure at node 2

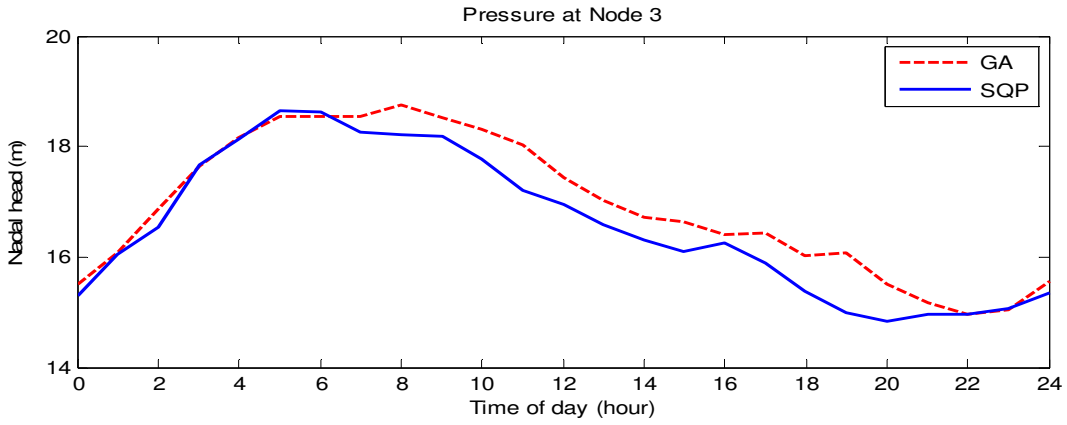


Figure 3.20 Pressure at node 3

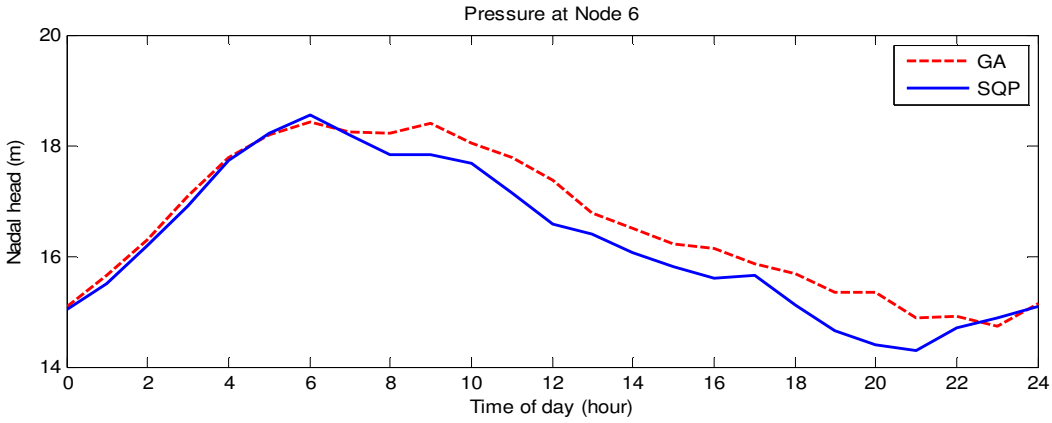


Figure 3.21 Pressure at node 6

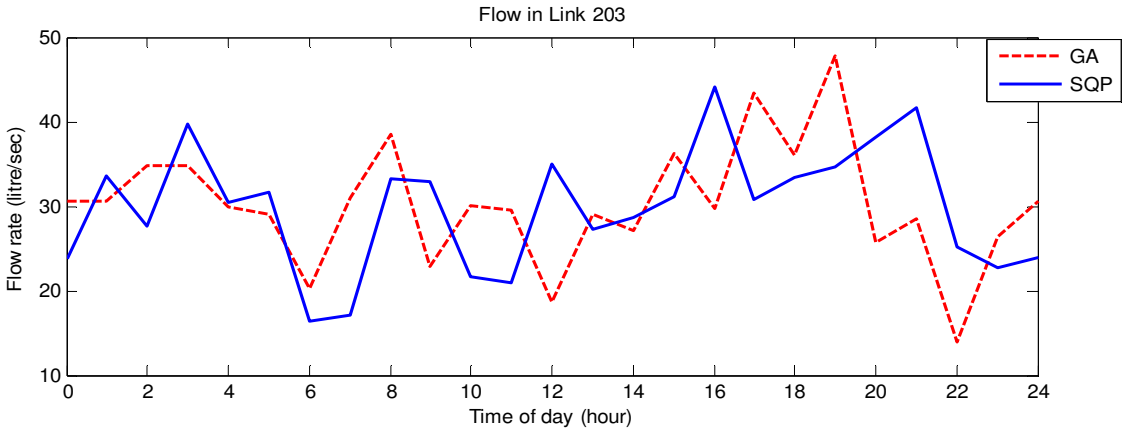


Figure 3.22 Flow in link 203

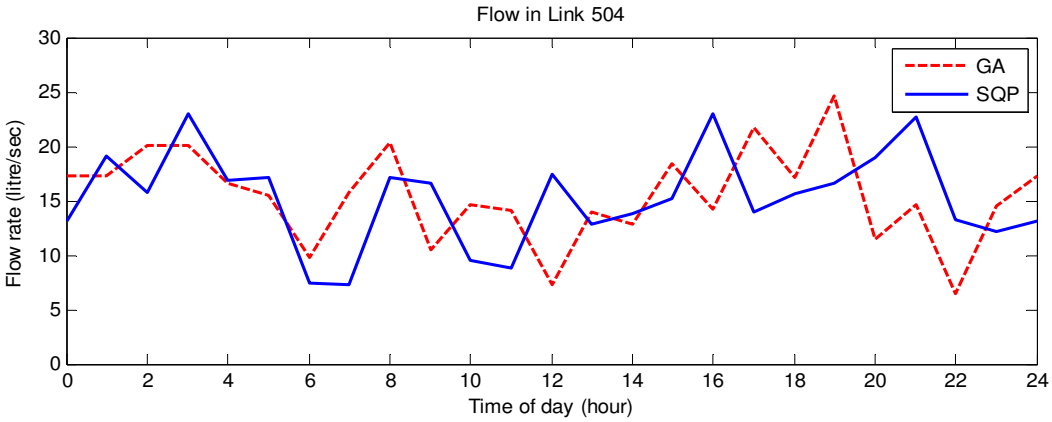


Figure 3.23 Flow in link 504

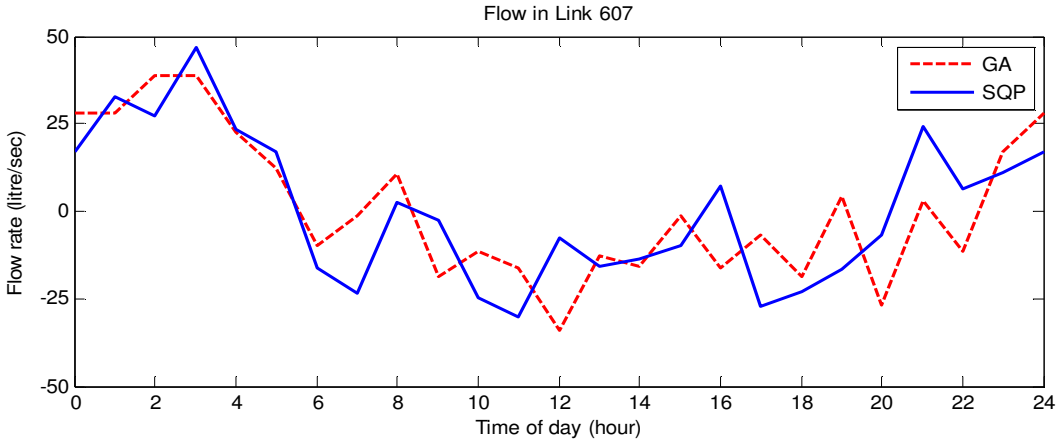


Figure 3.24 Flow in link 607

3.9 Summary

The basic idea of the MPC has been presented and the MPC optimization task has been formulated in both full space and reduced space of decision variables. The iterative control structure of RFMPC has been presented. Employing the iterated safety zones based RFMPC, techniques to improve computing time has been proposed i.e. stepwise robust output prediction and shortening robust feasibility horizon. The computational aspect has been further improved by using SQP as the MPC optimizer. Utilizing the Hamiltonian based technique, the gradients of objective functions and derivatives of constraints have been calculated to supply to the SQP solvers.

The DWDS example has been illustrated and the comparative simulation results have been obtained confirming the effectiveness of SQP over GA. Employing the RFMPC

technology, the DWDS over 24 hour period has been controlled in such a way the minimum electrical cost is achieved while satisfying the customer demands and system constraints. It has been also shown that the electrical pumping cost achieved by SQP is smaller than by GA.

Chapter 4

Explicit Safety Zones Based Robustly Feasible MPC and Application to DWDS

This chapter considers feasibility, robust feasibility, and recursive robust feasibility issues of the safety zones based RFMPC. Utilizing Karush-Kuhn-Tucker optimality condition the safety zones mechanism is further developed to achieve robust feasibility. Moreover, in order to attain both recursive and robust feasibility a dedicated iterative algorithm is proposed. The control algorithm is applied to control hydraulics in a DWDS example and simulation results are presented.

4.1 Introduction

In the previous chapter in order to maintain the output and state constraints, safety zones have been used to tighten the constraints of Nominal MPC (NMPC). The safety zones are re-designed at each time step to reduce conservatism. The calculation of safety zone is performed iteratively. However the safety zone approach still has several issues that need to be tackled. The first one is concerned with the feasibility issue i.e. under what condition of initial states there exist safety zones. For example in the operational of DWDS, if the initial tank level is at the very high level, there might be situation that there is no such feasible control input that stop the tank level from exceeding the physical limit which is not acceptable during online control. Hence, the set of feasible initial tank levels need to be found before applying control. In addition to the feasibility achievement, the robustness also needs to be considered due to the model-reality mismatch. Since the network system is considered in the operational control of DWDS for not just a particular time instant but over the time period, robust feasibility of control actions need to be guaranteed not only in one time step but as many time steps as needed. The second issue turns out to be about recursive feasibility i.e. how to guarantee that there exists safety zones in multi-steps given that safety zones exist at the current time step. The safety zone approach as described in Chapter 3 is further developed to achieve the recursive robust feasibility

In the context of MPC feasibility study, it has been known that the soft constraint approach (Scokaert and Rawlings, 1999; Maciejowski, 2002) can effectively handle mixed constraints for deterministic systems and invariant set approach (Gilbert and Tan, 1991; Blanchini, 1994, 1999; Dam and Nieuwenhuis, 1995; Santis, 1998; Kolmanovsky and

Gilbert 1998; Dorea and Hennet, 1999) can address the problem of state constraint satisfaction for uncertain systems. Especially, the latter approach has been successful in providing sufficient nominal and robust feasibility conditions in model predictive control (Kerrigan, 2000; Primbs and Nevistic, 2000; Kerrigan and Maciejowski, 2001; Cannon *et al*., 2002; Grieder *et al*., 2003). However, these approaches have been applied only to the linear systems or/with linear disturbances. Although the safety zone approach in (Brdys and Chang, 2002) is valid for nonlinear system, the recursive robust feasibility have not been demonstrated until now. In this chapter, the safety zones mechanism is further developed to achieve recursive robust feasibility for nonlinear network systems and illustrated by application to the DWDS.

The chapter is organized as follows: In section 2, the mathematical analysis of the feasibility of Nominal MPC optimization task is presented and the set $\mathbf{X}_f(k)$ composed of all feasible initial states is defined in the state space for which feasible control action exist over the prediction horizon for NMPC; an algorithm utilizing Karush-Kuhn-Tucker (KKT) optimality conditions is derived to determine a box type approximation $\mathbf{X}_{fa}(k)$ of this set. In section 3, the state and output constraints are modified by introducing one-step safety zones so that the control actions generated become one-step robustly feasible. The expressions to calculate the one-step robustly feasible safety zones are derived and starting the NMPC with modified constraints from the corresponding set of feasible state is vital in these derivations. This set is composed of the one-step robustly feasible states and it is denoted $\mathbf{X}_{Rf}(k)$. The MPC algorithm designed in this section is suitable for control over one step period only. In section 4 it is further developed to achieve the robust feasibility over a multiple-step period of controlling a system. The one-step robustly feasible safety

zones are enlarged and consequently the set $\mathbf{X}_{Rf}(k)$ is tightened to become invariant. It means that for the initial states in this set the MPC generates a control action over the first step forcing the system states at the end of this step to be also in this set. Hence, the robust feasibility is extended over a multiple-step period. The invariant set is further denoted $\mathbf{X}_{rRf}(k)$ with f, R, r respectively standing for feasible, robust, and recursive. The invariant set $\mathbf{X}_{rRf}(k)$ is composed of all recursively and robustly feasible states and the algorithm is proposed to determine this set. In section 5, the control algorithm is applied to control hydraulics in a DWDS example. Section 6 summarizes the chapter.

4.2 Feasibility of Nominal MPC optimization task

Recall the formulation of the Nominal MPC optimization task in the reduced space of variables i.e. equation (3.12):

$$\min_{u(\cdot|k)} J(u(\cdot|k), d(\cdot|k), x(k|k))$$
$$\textit{subject to}$$
$$c(u(\cdot|k), d(\cdot|k), x(k|k)) \geq 0$$

Notice that the solution of (3.12) depends on the initial state $x(k|k)$. Clearly, for some values of $x(k|k)$, there will be no sequence of control actions meeting the constraints. Hence, the MPC at k with such initial state will not be able to generate $u^{opt}(k|k)$ and MPC controller will crash. The control action $u^{opt}(k|k)$ clearly satisfies the model

constraints, which differ from the real plant constraints as $d(k \mid k) \neq d(k)$ in general. Although the control input constraints $u \in \mathbf{U}$ are satisfied but the state and output constraints may not be satisfied when $u^{opt}(k \mid k)$ is applied to the controlled plant. Hence, $u^{opt}(k \mid k)$ must be further modified in order to ensure meeting the state and output constraints in the real plant. In other words, the NMPC optimization task (3.12) must be modified so that the control actions $u^{opt}(k \mid k)$ exist and they are robustly feasible. Finally, robust feasibility is required not only for one or several steps but for all of them. The problem (3.12) thus describing the NMPC optimization task needs to be further developed to achieve robust and recursive feasibility.

Since $u(\cdot \mid k) = u^{opt}(\cdot \mid k)$ is at least the local minimizer of the problem (3.12), then $u^{opt}(\cdot \mid k)$ needs to satisfy the KKT (Antoniou and Lu, 2007) conditions:

- $c(u^{opt}(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) \geq 0$ (4.1a)
- there exist Lagrange multipliers γ_j^* such that

$$\nabla J(u^{opt}(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) - \sum_{j=1}^{NM} \gamma_j^* \nabla c_j = 0 \quad (4.1b)$$

- $\gamma_j^* c_j = 0 \;\; for \; 1 \leq j \leq M$ (4.1c)
- $\gamma_j^* \geq 0 \;\; for \;\; 1 \leq j \leq M$ (4.1d)

The feasible set $\mathbf{X}_f(k)$ is defined as the set of all states $x(k \mid k)$ such that there exists $u^{opt}(\cdot \mid k)$ satisfying KKT conditions i.e.

$$\mathbf{X}_f(k) \triangleq \{x(k \mid k) \in \mathbf{X} : \exists u^{opt}(\cdot \mid k) \in \mathbf{U}$$
$$\text{and } (u^{opt}(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) \text{ satisfies } (4.1)\}$$

If $x(k \mid k) \in \mathbf{X}_f(k)$, then a solution exists to the optimization problem (3.12) and hence the NMPC control input is defined for the given initial state. For all $x(k \mid k) \notin \mathbf{X}_f(k)$, a control input cannot be computed.

Finding $\mathbf{X}_f(k)$ is a very challenging problem due to possible complexity of $\mathbf{X}_f(k)$ and difficulties in recognizing this complexity. Therefore it shall be assumed that $\mathbf{X}_f(k)$ can be approximated by the box $\mathbf{X}_{fa}(k) = [x_f^{\min}(k), x_f^{\max}(k)] \subset \mathbf{X}_f(k)$ with sufficient accuracy. The problem becomes to find the minimum $x_f^{\min}(k)$ and maximum $x_f^{\max}(k)$. The set of KKT conditions (4.1) introduces certain new variables i.e. the Lagrange multipliers $\gamma_1, ..., \gamma_P$.

Let denote the vector Θ as the vector of all variables in (4.1). Hence,

$$\Theta \triangleq [(u^{opt}(\cdot \mid k))^T, x(k \mid k), \gamma_1^*, \gamma_2^*, ..., \gamma_{NM}^*]^T$$

The vector of states $x(k)$ is composed of n states, and can be decomposed as:

$$x(k) = [x_1(k), ..., x_n(k)]^T$$

To find $x_f^{\min}(k) = [x_{f,1}^{\min}(k), ..., x_{f,n}^{\min}(k)]^T$, we solve the following optimization problem (4.2) for $m = \overline{1:n}$.

$$
\begin{aligned}
&\text{Find} \quad x_{f,m}^{\min}(k) = Arg \min_{\Theta} x_m(k) \\
&\text{subject to:} \\
&x(k \mid k) \in \mathbf{X} \\
&c(u^{opt}(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) \geq 0 \\
&\nabla J(u^{opt}(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) - \sum_{j=1}^{NM} \gamma_j^* \nabla c_j = 0 \\
&\gamma_j^* c_j = 0 \ \ for \ 1 \leq j \leq M \\
&\gamma_j^* \geq 0 \ \ for \ \ 1 \leq j \leq M
\end{aligned}
\tag{4.2}
$$

Similarly, solving optimization problem (4.3) for $m = \overline{1:n}$ yields $x_f^{\max}(k) = [x_{f,1}^{\max}(k), ..., x_{f,l}^{\max}(k)]^T$. It should be noticed that the feasible set $\mathbf{X}_f(k)$ is non stationary due to the time varying disturbance prediction.

$$
\begin{aligned}
&\text{Find} \quad x_{f,m}^{\max}(k) = Arg \max_{\Theta} x_m(k) \\
&\text{subject to:} \\
&x(k \mid k) \in \mathbf{X} \\
&c(u^{opt}(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) \geq 0 \\
&\nabla J(u^{opt}(\cdot \mid k), d(\cdot \mid k), x(k \mid k)) - \sum_{j=1}^{NM} \gamma_j^* \nabla c_j = 0 \\
&\gamma_j^* c_j = 0 \ \ for \ 1 \leq j \leq M \\
&\gamma_j^* \geq 0 \ \ for \ \ 1 \leq j \leq M
\end{aligned}
\tag{4.3}
$$

Solving (4.2) and (4.3) is very computationally demanding as the numbers of variables grows very quickly due to the numbers of Lagrange multipliers $\gamma_1^*, \gamma_2^*, ..., \gamma_{NM}^*$. For example, a simple net work with one control input and one state. The prediction horizon is assumed to be 10. If constraints are applied to state and input over the entire horizon, then

there are 20 constraints all together. Consequently, there are also 20 Lagrange multipliers associated with those constraints. In the formulation of (4.2) and (4.3), the decision variables which are control input over the horizon, initial state, and all of the Lagrange multipliers yield a total of 31 variables. It can be clearly seen that for the medium to large network, the numbers of variables grows even much faster. It is then not even possible to carry such heavy calculations by the standard solvers within the reasonable times (hours). Fortunately, by utilizing the Hamiltonian based technique described in Chapter 3, the NMPC optimization task in the reduced space of decision variables can be solved within affordable time.

4.3 One step robust feasibility

The control action $u^{opt}(k \mid k)$ obtained by NMPC starting with $x(k \mid k) = x(k) \in \mathbf{X}_f(k)$ may not guarantee that the resulting state $x(k+1)$ in the system is feasible. The safety zones were introduced in (Brdys and Chang, 2002), (Chang, 2002) to modify model based state/output constraints in the MPC optimization task over the whole prediction disturbance in order to achieve robust feasibility of the generated control actions over the prediction horizon. The robustly feasible safety zones values were produced by a dedicated algorithm operating on-line. The algorithm is very computationally demanding. In this section, it shall be introduced the safety zones to modify the state constraints only over one step to achieve the one step robust feasibility. Moreover, the analytical expression for the

robustly feasible safety zone values are derived so that they can be calculated off-line saving on computing time.

The modified of m th component of state constraint limits are expressed as follows:

$$x_{md,m}^{\min} = x_m^{\min} + \varepsilon_m^l \tag{4.4a}$$

$$x_{md,m}^{\max} = x_m^{\max} - \varepsilon_m^u \tag{4.4b}$$

where ε_m^u and ε_m^l are non negative number, $m = \overline{1:n}$, and they are the lower and upper safety zones of the state constraints respectively.

The vector of outputs $y(k)$ is composed of n_c components, and can be decomposed as:

$$y(k) = [y_1(k), ..., y_{n_c}(k)]^T$$

The modified m th component of output constraint limits are expressed as follows:

$$y_{md,m}^{\min} = y_m^{\min} + \varepsilon_{ym}^l \tag{4.5a}$$

$$y_{md,m}^{\max} = y_m^{\max} - \varepsilon_{ym}^u \tag{4.5b}$$

where ε_{ym}^u and ε_{ym}^l are non negative numbers, $m = \overline{1:n_y}$, and they are the lower and upper safety zones of the output constraints, respectively.

A vector of lower and upper safety zones for the output constraints is defined as follows:

$$\varepsilon_y^l \triangleq [\varepsilon_{y1}^l, ..., \varepsilon_{yn_c}^l]^T$$

$$\varepsilon_y^u \triangleq [\varepsilon_{y1}^u, ..., \varepsilon_{yn_c}^u]^T$$

A vector of modified output constraint limits are expressed as follows:

$$y_{md}^{\min} = y^{\min} + \varepsilon_y^l \tag{4.6a}$$

$$y_{md}^{\max} = y^{\max} - \varepsilon_y^u \tag{4.6b}$$

The additional constraints (4.4) over the first step, i.e. $x_{md,m}^{\min} \le x_m(k+1|k) \le x_{md,m}^{\max}$ and (4.5), i.e. $y_{md}^{\min} \le y(k|k) \le y_{md}^{\max}$, are added to the optimization problems (4.2) and (4.3) to yield optimization problem (4.7) and (4.8). Clearly, the one step modified state and output constraints are tighter than the original ones and this is the price to be paid due to uncertainty in the disturbance prediction.

A state $x(k|k)$ for which solution of the NMPC optimization task (4.7) and (4.8) with the modified constraints exists is called one step robustly feasible state as the resulting state $x(k+1)$ and resulting output $y(k)$ meet the real system constraints. Denote the set of all one step robustly feasible states $[x_{Rf,1}(k),...,x_{Rf,n}(k)] \in \mathbf{X}_{Rf}(k)$. Finding $\mathbf{X}_{Rf}(k)$ is a very challenging problem due to possible complexity of $\mathbf{X}_f(k)$ and difficulties in recognizing this complexity. It therefore shall be assumed that it can be accurately enough approximated as a box $\mathbf{X}_{Rfa}(k) = [x_{Rf}^{\min}(k), x_{Rf}^{\max}(k)] \subset \mathbf{X}_{Rf}(k)$. The problem becomes to find the minimum $x_{Rf}^{\min}(k)$ and maximum $x_{Rf}^{\max}(k)$.

To find $x_{Rf}^{\min}(k) = [x_{Rf,1}^{\min}(k),...,x_{Rf,n}^{\min}(k)]^T$ and $x_{Rf}^{\max}(k) = [x_{Rf,1}^{\max}(k),...,x_{Rf,n}^{\max}(k)]^T$, we solve the following optimization problem (4.7) and (4.8) respectively for $m = \overline{1:n}$.

$$
\begin{aligned}
\text{Find}\quad & x_{f,m}^{\min}(k) = Arg \min_{\Theta} x_m(k) \\
& \text{subject to:} \\
& x_m^{\min} + \varepsilon_m^l \le x_m(k+1\,|\,k) \le x_m^{\max} - \varepsilon_m^u \\
& y^{\min} + \varepsilon_y^l \le y\ (k\,|\,k) \le y^{\max} - \varepsilon_y^u \\
& x(k\,|\,k) \in \mathbf{X} \\
& c(u^{opt}(\cdot\,|\,k), d(\cdot\,|\,k), x(k\,|\,k)) \ge 0 \\
& \nabla J(u^{opt}(\cdot\,|\,k), d(\cdot\,|\,k), x(k\,|\,k)) - \sum_{j=1}^{NM} \gamma_j^* \nabla c_j = 0 \\
& \gamma_j^* c_j = 0 \ \ for\ 1 \le j \le M \\
& \gamma_j^* \ge 0 \ \ for\ \ 1 \le j \le M
\end{aligned}
\tag{4.7}
$$

$$
\begin{aligned}
\text{Find}\quad & x_{f,m}^{\max}(k) = Arg \max_{\Theta} x_m(k) \\
& \text{subject to:} \\
& x(k\,|\,k) \in \mathbf{X} \\
& x_m^{\min} + \varepsilon_m^l \le x_m(k+1\,|\,k) \le x_m^{\max} - \varepsilon_m^u \\
& y^{\min} + \varepsilon_y^l \le y\ (k\,|\,k) \le y^{\max} - \varepsilon_y^u \\
& c(u^{opt}(\cdot\,|\,k), d(\cdot\,|\,k), x(k\,|\,k)) \ge 0 \\
& \nabla J(u^{opt}(\cdot\,|\,k), d(\cdot\,|\,k), x(k\,|\,k)) - \sum_{j=1}^{NM} \gamma_j^* \nabla c_j = 0 \\
& \gamma_j^* c_j = 0 \ \ for\ 1 \le j \le M \\
& \gamma_j^* \ge 0 \ \ for\ \ 1 \le j \le M
\end{aligned}
\tag{4.8}
$$

Notice that in order to solve optimization (4.7) and (4.8) the safety zones ε_m^u, ε_m^l, ε_{ym}^u, and ε_{ym}^l need to be calculated beforehand. The method of how to compute these safety zones is done via the Lipschitz constant.

First, let find the safety zones for state constraints, i.e. ε_m^u and ε_m^l

Recall the state equation (3.24):

$$x(k+i+1|k) = f(x(k+i|k), u(k+i|k), d(k+i|k))$$

At time k, i.e $i=0$, the initial state $x(k|k)$ is given and the optimal control input $u^{opt}(k|k)$ is calculated and applied to the plant. From equation (3.24), the predicted state which is based on the predicted $d(k|k)$ is:

$$x(k+1|k) = f(x(k|k), u^{opt}(k|k), d(k|k)) \tag{4.9}$$

In reality the real disturbance is $d(k)$, hence the state which is obtained by taking the measurement from the plant is:

$$x(k+1) = f(x(k|k), u^{opt}(k|k), d(k)) \tag{4.10}$$

It follows from (4.9) and (4.10) that:

$$x(k+1) = x(k+1|k) + f(x(k|k), u^{opt}(k|k), d(k)) - f(x(k|k), u^{opt}(k|k), d(k|k))$$

Hence, for m th component $x_m(k)$ of $x(k)$:

$$\begin{aligned} x_m(k+1) &= x_m(k+1|k) + f_m(x(k|k), u^{opt}(k|k), d(k)) - f_m(x(k|k), u^{opt}(k|k), d(k|k)) \\ &\le x_m(k+1|k) + L_{f_m,d} \left\| d(k) - d(k|k) \right\| \\ &\le x^{\max} - \varepsilon_m^u + L_{f_m,d} \left\| d(k) - d(k|k) \right\| \end{aligned} \tag{4.11}$$

where $L_{f_m,d}$ is the Lipschitz constant of m th component $f_m(\cdot)$ of the function $f(\cdot)$ with respect to $d(k)$.

Thus, if

$$\varepsilon_m^u \geq L_{f_m,d} \left\| d(k) - d(k \mid k) \right\|$$

for any $m \in [1, n]$, then $x_m^{\min} \leq x_m(k+1 \mid k) \leq x_m^{\max}$.

In order to achieve the least conservatism: $\varepsilon_m^u \geq L_{f_m,d} \left\| d(k) - d(k \mid k) \right\|$ (4.12a)

Similarly, the same argument can be applied to find the safety zones for the lower limit of state constraints i.e.

$$\varepsilon_m^l = L_{f_m,d} \left\| d(k) - d(k \mid k) \right\| \tag{4.12b}$$

Applying the same procedure as for the state constraints, the safety zones for output constraints, i.e. ε_{ym}^u and ε_{ym}^l, are calculated as follows:

Recall the output equation (3.6):

$$y(k) = E(u(k), x(k), d(k))$$

At time k, i.e $i = 0$, the initial state $x(k \mid k)$ is given and the optimal control input $u^{opt}(k \mid k)$ is calculated and applied to the plant. From equation (3.6), the predicted output which is based on the predicted $d(k \mid k)$ is:

$$y(k \mid k) = E(u(k \mid k), x(k \mid k), d(k \mid k)) \tag{4.13}$$

In reality the real disturbance is $d(k)$, hence the output which is obtained by taking the measurement from the plant is:

$$y(k) = E(u(k \mid k), x(k \mid k), d(k)) \tag{4.14}$$

Hence, utilizing (4.13) and (4.14), for m th component $y_m(k)$ of $y(k)$:

$$\begin{aligned} y_m(k) &= y_m(k \mid k) + E_m(u(k \mid k), x(k \mid k), d(k)) - E_m(u(k \mid k), x(k \mid k), d(k \mid k)) \\ &\le y_m(k \mid k) + L_{E_m,d} \left\| d(k) - d(k \mid k) \right\| \\ &\le y^{\max} - \varepsilon_{ym}^{u} + L_{E_m,d} \left\| d(k) - d(k \mid k) \right\| \end{aligned} \tag{4.15}$$

where $L_{E_m,d}$ is the Lipschitz constant of mth component $E_m(\cdot)$ of $E(\cdot)$ with respect to $d(k)$.

Thus, if $\varepsilon_{ym}^{u} \ge L_{E_m,d} \left\| d(k) - d(k \mid k) \right\|$

In order to achieve the least conservatism: $\varepsilon_{ym}^{u} = L_{E_m,d} \left\| d(k) - d(k \mid k) \right\|$ (4.16a)

for any $m \in [1, n_y]$, then $y_m^{\min} \le y_m(k \mid k) \le y_m^{\max}$.

Similarly, the same argument can be applied to find the safety zones for the lower limit of output constraints i.e.

$$\varepsilon_{ym}^{l} = L_{E_m,d} \left\| d(k) - d(k \mid k) \right\| \tag{4.16b}$$

It is now clear to see that the safety zone technique is used explicitly. In other words, the safety zones are explicitly calculated before applying control and the all computation is made off-line. In contradictory to this, the safety zones in Chapter 3 are calculated online to achieve RFMPC, hence all computation is carried out online. With the novel technique to utilize safety zones explicitly, the RFMPC is still achieved and the burden of computation goes off-line. This essence is the major difference between how safety zones based technique is used to achieve RFMPC in Chapter 3 and Chapter 4.

4.4 Recursive robust feasibility

4.4.1 Invariant sets

Due to the optimization nature of MPC algorithm, in safety-critical applications it is desirable that infeasibility of MPC optimization problems is avoided at all costs in order to guarantee constraint satisfaction. Set invariance plays a fundamental role in the design of control systems for constrained systems since the constraints can be satisfied for all time if and only if the initial state is contained inside an invariant set. In particular, invariant set theory (Blanchini, 1999) has been known to be essential in understanding the behaviour of constrained systems and in the design of MPC controllers. In (Kerrigan, 2001), the invariance set theory has been used in robust analysis MPC controllers. The iterative algorithm to calculate the maximum robust control invariant set for systems with linear disturbances has been proposed in (Grieder et al., 2003). The invariance set computation was then utilized to achieve the robustly feasible MPC strategy switching in (Wang, 2006). Following Blanchini (1999) and Gilbert and Tan (1991), the basic concepts and definition of invariant sets are introduced in this section, which will be useful in understanding the content of the next sub-section.

Definition 2.1 (*Positively invariant set*). The non-empty set $\Omega \in \mathbb{R}^n$ is *positively invariant* for the autonomous system $x_{k+1} = f(x_k)$ if and only if $\forall x_0 \in \Omega$ the system state evolution satisfies $x_k \in \Omega$, $\forall k \in \overline{1:\infty}$. The set Ω is *invariant* if and only if $x_0 \in \Omega$ implies $x_k \in \Omega$, $\forall k \in \overline{0:\infty}$.

In general, a given set Ω is not positively invariant. It is more often for one to determine the largest positively invariant set contained in Ω.

Definition 2.2 (*Maximal positively invariant set*). The non-empty set $O_\infty(\Omega)$ is the *maximal positively invariant set* contained in Ω for the autonomous system $x_{k+1} = f(x_k)$ if and only if $O_\infty(\Omega)$ is positively invariant and contains all positively invariant sets contained in Ω, i.e. Υ is positively invariant only if $\Upsilon \subseteq O_\infty(\Omega) \subseteq \Omega$.

Definition 2.3 (*Control invariant set*). The non-empty set $\Theta \in \mathbb{R}^n$ is a *control invariant set* for the system $x_{k+1} = f(x_k, u_k)$ if and only if there exists a feedback control law $u_k = g(x_k)$ such that Θ is a positively invariant set for the closed-loop system $x_{k+1} = f(x_k, g(x_k))$ and u_k is an admissible control input for $\forall x_k \in \Theta$.

In general, a given set Θ is not control invariant. It is more often for one to determine the largest control invariant set contained in Θ.

Definition 2.4 (*Maximal control invariant set*). The non-empty set $C_\infty(\Theta)$ is the *maximal control invariant set* contained in Θ for the system $x_{k+1} = f(x_k, u_k)$ if and only if $C_\infty(\Theta)$ is control invariant and contains all control invariant sets contained in Θ, i.e. Υ is control invariant only if $\Upsilon \subseteq C_\infty(\Theta) \subseteq \Theta$.

A thorough exploration of invariant set concepts falls outside the scope of the thesis. One can refer to (Bitsoris, 1988; Gilbert and Tan 1991; Rachid, 1991; Castelan and Hennet,

1993; Blanchini, 1994; De Santis, 1994; Blanchini, 1999; Dorea and Hennet, 1999; Kerrigan, 2000; Vidal *et al*., 2000) for more extensive discussions on various invariant sets.

The invariance set theory actually has been an effective tool to ensure recursive feasibility of the MPC controller due to its special property. Next, we focus on achieving the recursive robust feasibility by further develop the analysis of one step robust feasibility as described in the previous section.

4.4.2 Recursive robust feasibility - RFMPC

In the previous section, it has been concluded that in order to have at least one-step robustly feasible control action the initial state must lie in $\mathbf{X}_{Rf}(k)$. However, at the end of one step period, the current NMPC initial state is updated from the measurement and can be anywhere within $\mathbf{X}$. However, the set $\mathbf{X}_{Rf}(k)$ is the subset of $\mathbf{X}$ and if $x(k+1) \notin \mathbf{X}_{Rf}(k+1)$ then there will be no feasible solution of the NMPC optimization task. The control algorithm will then crash. To achieve robustly feasible control action over multiple-step period, the current state needs to lie inside the one-step robustly feasible sets. In other words, $x(k) \in \mathbf{X}_{Rf}(k)$ for any k, which means that $\mathbf{X}_{Rf}(k)$ is invariant set with respect to the MPC one step state to state mapping.

Let us assume first that $\mathbf{X}_{Rf}(k) = \mathbf{X}_{Rf}$, which is the case if the disturbance prediction over $k+i$, $i = \overline{1:N-1}$ is the same at any k. Let now $x(k \mid k) \in \mathbf{X}_{Rf}$ and it is required that $x(k+1) \in \mathbf{X}_{Rf}$ for recursive robust feasibility. This can be forced by using $x_{md,m}^{\min} = x_{Rf,m}^{\min} + \varepsilon_m^l$ and $x_{md,m}^{\max} = x_{Rf,m}^{\max} - \varepsilon_m^u$ as additional modified constraints which is described in (4.6) and (4.7). It means that the set $\mathbf{X}_{Rf}$ will be minimally tightened to a smaller robustly feasible $\widetilde{\mathbf{X}}_{Rf}$. When the statement $x(k+1) \in \widetilde{\mathbf{X}}_{Rf}$ if $x(k \mid k) \in \widetilde{\mathbf{X}}_{Rf}$ holds, then the tightening $\mathbf{X}_{Rf}$ into $\widetilde{\mathbf{X}}_{Rf}$ has produced the invariant set, which achieves recursive robust feasibility. If not, then the next tightening operation is performed with $x_{md,m}^{\min} = \tilde{x}_{Rf,m}^{\min} + \varepsilon_m^l$ and $x_{md,m}^{\max} = \tilde{x}_{Rf,m}^{\max} - \varepsilon_m^u$. The above can be formulated as follows:

Algorithm 4.1:

For $m = \overline{1:n}$

- Step 1: Find $x_{Rf,m}^{\min}$, $x_{Rf,m}^{\max}$ by solving (4.2) and (4.3) with an additional constraints

  $x_{md,m}^{\min} \le x_m(k+1 \mid k) \le x_{md,m}^{\max}$ and $y_{md}^{\min} \le y(k \mid k) \le y_{md}^{\max}$

  where the modified constraints $x_{md,m}^{\min}$, $x_{md,m}^{\max}$, $y_{md}^{\min}$, and $y_{md}^{\max}$ are calculated in (4.4) and (4.6)

- Step 2: Find $\tilde{x}_{Rf,m}^{\min}$, $\tilde{x}_{Rf,m}^{\max}$, by solving (4.2) and (4.3) with the additional constraints:

  $x_{Rf,m}^{\min} + \varepsilon_m^l \le x_m(k+1 \mid k) \le x_{Rf,m}^{\max} - \varepsilon_m^u$ and $y_{md}^{\min} \le y(k \mid k) \le y_{md}^{\max}$

  where $x_{Rf,m}^{\min}$ and $x_{Rf,m}^{\max}$ are obtained from Step 1

- Step 3: If $x_{Rf,m}^{\min} \neq \tilde{x}_{Rf,m}^{\min}$ and/or $x_{Rf,m}^{\max} \neq \tilde{x}_{Rf,m}^{\max}$,

then set $x_{Rf,m}^{\min} = \tilde{x}_{Rf,m}^{\min}$ and/or $x_{Rf,m}^{\max} = \tilde{x}_{Rf,m}^{\max}$ and go to Step 2

else go to step 4

- Step 4: Obtain $x_{rRf,m}^{\min} = \tilde{x}_{Rf,m}^{\min}$ and $x_{rRf,m}^{\max} = \tilde{x}_{Rf,m}^{\max}$

As $x_{rRf}^{\max} - \varepsilon^u = x^{\max} - (x^{\max} - x_{rRf}^{\max} + \varepsilon^u)$ and similarly $x_{rRf}^{\min} + \varepsilon^l = x^{\min} + (x_{rRf}^{\min} - x^{\min} + \varepsilon^l)$, it is clear that in general the safety zones ensuring one-step robust feasibility must be further increased to ensure recursive robust feasibility. This has been achieved iteratively but with only one-step robust feasibility, however resulting in the *maximal control invariant set* $\mathbf{X}_{rRf}$. The convergence of the algorithm 4.1 has not been proven and only been verified by the simulation. In general, algorithm 4.1 cannot be ensured to terminate in finite time, but confining the iteration times and defining precision degree are usual ways to obtain an approximate solution.

The final formulation of the RFMPC optimisation task is:

$$\begin{aligned} &\min_{u(\cdot|k)} J(u(\cdot\,|\,k), d(\cdot\,|\,k), x(k\,|\,k)) \\ &\text{subject to}: \quad x_{rRf}^{\min} + \varepsilon^l \le x(k+1\,|\,k) \le x_{rRf}^{\max} - \varepsilon^u \\ &\qquad\qquad y^{\min} + \varepsilon_y^l \le y(k\,|\,k) \le y^{\max} - \varepsilon_y^u \\ &\qquad\qquad c(u(\cdot\,|\,k), d(\cdot\,|\,k), x(k\,|\,k)) \ge 0 \\ &\text{where } x(k\,|\,k) \in \mathbf{X}_{rRf} \end{aligned} \tag{4.17}$$

In case when the set of robustly feasible states is truly time-varying we would pursue the recursive robust feasibility of the MPC as follows: First, the robustly feasible invariant sets

$\mathbf{X}_{rRf}(k)$, $k = k_0,...,N_c$ are designed as above over the control period. Then the stationary $\mathbf{X}_{rRf}$ which is suitable for the whole control period is produced as:

$$\mathbf{X}_{rRf} = \bigcap_{k=k_0}^{N_c} \mathbf{X}_{rRf}(k) \tag{4.18}$$

assuming that the intersection is non empty. If it is empty then the period $[k_0, N_c]$ can be partitioned into P smaller periods such that the intersection in (4.18) is non empty over each of these periods to produce $\mathbf{X}_{rRf}^{p}$, where p is the period number. These P different $RFMPC^{p}$, $p \in P$, each of them has different stationary $\mathbf{X}_{rRf}^{p}$. Robustly feasible operation of MPC over the overall control period would then involve switching from $RFMPC^{p-1}$ to $RFMPC^{p}$, $p \in P$. The soft switching technique between different robust MPC strategies was address in (Brdys, 2010) and (Brdys and Wang, 2005). Further progress in this case is reported in the Chapter 6.

4.5 DWDS Case Study

The DWDS example has the same structure as the one described in the Chapter 3. However there is a minor difference of the profile data has been applied. For the sake of coherent illustration, not all but some selected profile data is mentioned again. The detail

data of this network is provided in the Appendix D. The example network is depicted in the Figure 4.1, where q_i and $h_i, i = 1,...,7$ are the pipe flows and the nodal heads respectively. d_3, d_5, d_6 are the water demands at the consumption nodes 3, 5, and 6 respectively, and ps is denoted the relative pump speed. The network elements are described by the head-flow qualities. Due to its storage capabilities the tank head-flow relationship is dynamic as opposed to the other elements which are static, hence described by the nonlinear algebraic equation (3.4). The control, state, output, and disturbance variables are: $u = ps$, $x = h_7$, $y = [q, h]$, where $q = [q_1,...,q_7]$ and $h = [h_1,...,h_6]$, $d = [d_3, d_5, d_6]$. Given $h_7(k)$ and $ps(k)$, the forced output $y(k)$ can be obtained by solving (3.6).

The operational tank limits are: $x^{\max} \le 8[m]$ and $x^{\min} \ge 2[m]$. The relative pump speed is constrained by: $0.2 = u^{\min} \le u(k) \le u^{\max} = 1.1$. The demand patterns of node 5 and 6 are displayed respectively in Figure 4.2, and node 3 has the fixed demand of 2 [litres/second] over the entire horizon is composed of pipes, valves, pumps, and tanks, and the operational goal is to deliver water from the sources to the water users.

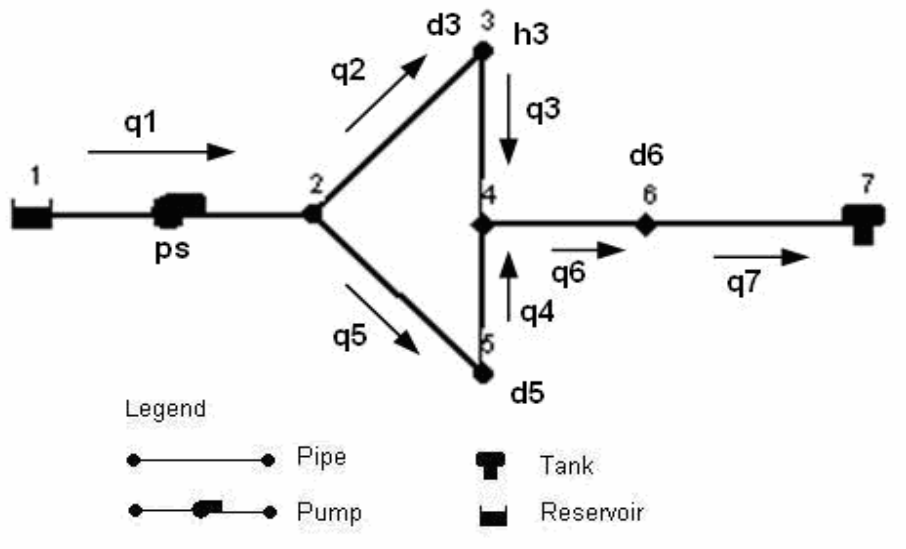


Figure 4.1 Diagram of DWDS example

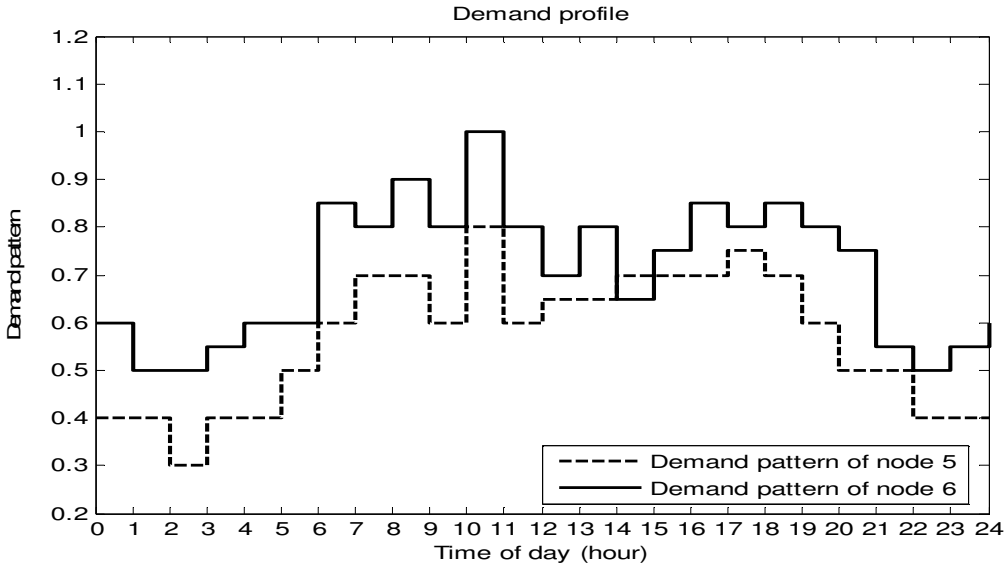


Figure 4.2 Demand pattern profile of the DWDS example

4.5.1 RFMPC design

As described in equation (3.37), the cost function J_c represents the electrical energy consumed over 24 hours. The time continuous tank equation is discretized with the

discretization step equal to 1 hour. Hence, the control inputs in the RFMPC optimization task at k are: $u(\cdot \mid k) = [u(k \mid k), ..., u(k+23 \mid k)] = [ps(0), ..., ps(23)]$

Discretizing the time continuous tank difference equation (2.11) yields the state equation:

$$\begin{aligned} x(k+1) &= x(k) + (1/A)\, y_7(k) = x(k) + (1/A) q_7(k) \\ &= x(k) + (1/A) E_7(u(k), x(k), d(k)) \end{aligned}$$

where A is the tank cross sectional area

The constraints have been described already in the introduction of the Section 4.5.

Hence, from (3.24):

$$f(x(k), s(k), d(k)) = x(k) + (1/A) E_7(u(k), x(k), d(k))$$

and $L_{f_m,d} = (1/A) L_{E_7,d}$ and $L_{E_7,d}$ is the Lipschitz constant of E_7 with respect to d is calculated based on its derivative with respect to d.

As $d = [d_3, d_5, d_6]$, the Lipschitz constant of E_7 with respect to d is the summation of Lipschitz constant of E_7 with respect to each component of d:

$$L_{E_7,d} = L_{E_7,d_3} + L_{E_7,d_5} + L_{E_7,d_6}$$

Calculating L_{E_7,d_3}, L_{E_7,d_5}, and L_{E_7,d_6} requires solving the optimization problems with the objective functions respectively are $\frac{\partial q_7}{\partial d_3}$, $\frac{\partial q_7}{\partial d_5}$, and $\frac{\partial q_7}{\partial d_6}$. Those derivatives are evaluated based on implicit differentiation theorem as described in the Section 3.8.3.1. This results in $L_{f_m,d} = 0.18$. The Algorithm 4.1 was applied to calculate $\mathbf{X}_{rRf} = [3.35, 7.21]$ and its iterations are illustrated in Table 4.1. The set $\mathbf{X}_{rRf}$ is shown in Figure 4.3 where its limits are marked red still leaving large operational capacity of the tank.

Number of iterations	$\sim^{\min}_{x_{Rf}}$	$\sim^{\max}_{x_{Rf}}$
0	2.00	8.00
1	2.41	7.70
2	2.54	7.63
3	2.85	7.34
4	2.93	7.21
5	3.33	7.21
6	3.35	7.21

Table 4.1 Iterations of Algorithm 4.1

In the Table 4.1, the Algorithm 4.1 has been terminated after six iterations in 10 hours. The reason for its termination is due to one of the following stopping criteria:

- Maximum change in function value is set to be 0.05. It means that when the change in value of the optimized function is less than 0.05, the algorithm stops.
- Stall time limit is set to be 7200 seconds. It means that if there is no improvement in the value of the optimized function for an interval of time in seconds specified by stall time limit, the algorithm stops.
- Generations is set to be 10. It means that the maximum number of generations the algorithm performs is 10.

4.5.2 Simulation results

The demand profile in DWDS and its prediction are illustrated in Figure 4.2. The simulations of RFMPC operation were carried out for several initial tank levels within the set $\mathbf{X}_{rRf} = [3.35, 7.21]$ and the results are illustrated in Figure 4.3 and Figure 4.4. It can be seen that the tank constraints are satisfied over a whole control period of 24 hours. Moreover, the tank trajectory lies inside the set $\mathbf{X}_{rRf}$ confirming the invariance of this set.

Next the initial tank level was selected outside $\mathbf{X}_{rRf}$ at 7.6[m] and the RFMPC crashed at 5 hours because there was no feasible control input trajectory at this initial state. This confirms that the invariant set $\mathbf{X}_{rRf}$ is maximal. It also can be seen in Figure 4.4 that the optimised relative pump speeds corresponding to the different initial tank levels are indeed different, which emphasizes an impact of the initial state on the resulting control input trajectory. Also the tank capacity is restored at the end of 24 hrs period; hence the system operation is sustainable over a long term.

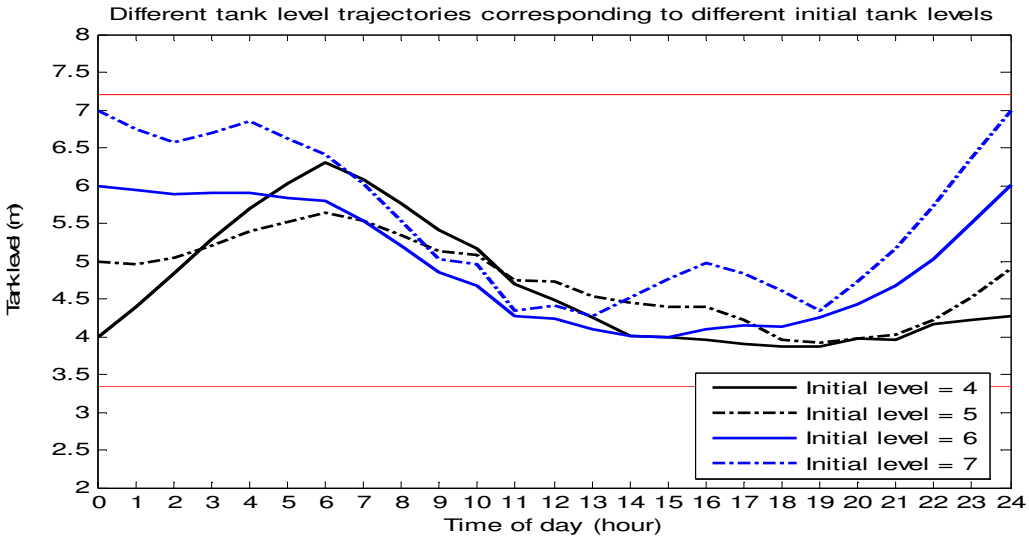


Figure 4.3 RFMPC for different initial tank levels - tank trajectories

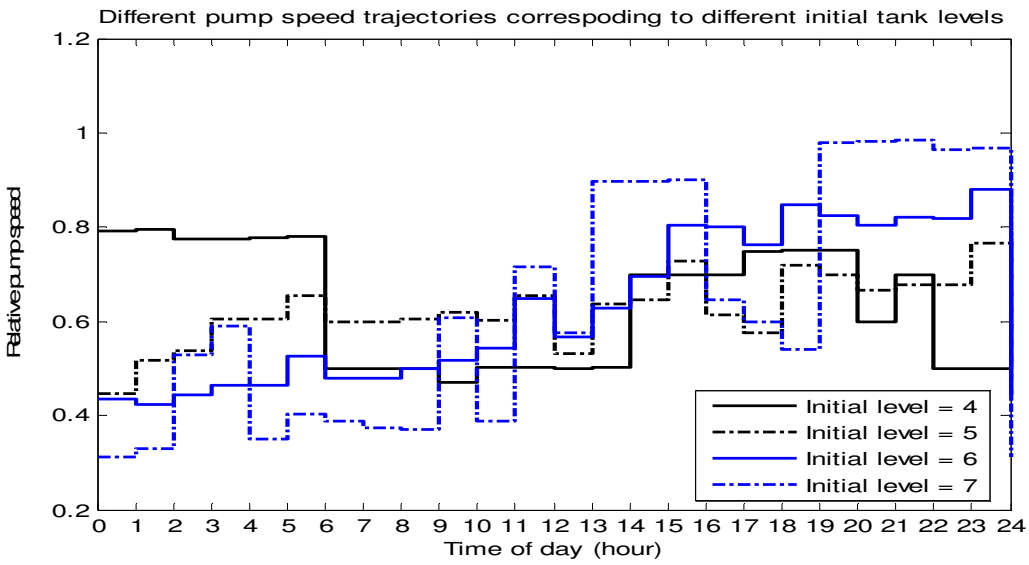


Figure 4.4 RFMPC for different initial tank levels – pump speed schedule

4.6 Summary

The analysis on the feasibility of NMPC optimization task has been presented. Utilizing KKT optimality condition, the procedure to calculate approximation of $\mathbf{X}_f(k)$ composed of all feasible initial states is defined in the state space for which feasible control action exist over the prediction horizon for NMPC has been derived. Further develop on the safety zone approach has been made in order to achieve one step robust feasibility. The safety zones based technique has been derived in order to achieve one step robustly feasible states. Moreover, the iterative algorithm has been proposed to determine the approximation of the invariant set under the assumption that the disturbance prediction is not time varying.

The RFMPC controller has been successfully applied to the DWDS example to meet water consumer demands at minimal operational cost in sustainable manner. Simulation results have been presented to confirm the property of the invariant sets.

Chapter 5

Softly Switched RFMPC (SSFMPC) and Application to DWDS under wide range of operational conditions

This chapter considers Softly Switched RFMPC and its components. The soft switching mechanism is presented and the feasibility of the hard switching is discussed. The functionalities of supervisory control level and its importance with respect to switching are briefly discussed.

5.1 Introduction

In the operational control of DWDS, it is often that there are several objectives that need to be met by the plant. In chapter 3 and chapter 4, different techniques to design RFMPC have been proposed in order to meet single control objective. In the application of DWDS under wide ranges of operational conditions, it requires more than one RFMPC controller that needs to be designed. When the current control strategy needs to be replaced by the new one, it requires a switching process between the current and new controllers. Consequently, the overall system is further developed to so-called Softly Switched RFMPC in order to accommodate, coordinate, and integrate all RFMPC controllers.

The chapter is organized as follows: The SSRFMPC and its components are presented in Section 5.2. In this section, three main operational states of the DWDS are distinguished. The switching mechanism is considered in Section 5.3. To have the better understanding of the structure of hierarchical predictive control, the Supervisory Control Level is discussed in Section 5.4. Section 5.5 summarizes the chapter.

5.2 SSRFMPC presentation and its components

Given a plant to be controlled and set of objectives to be achieved by using RFMPC controller. A single control objective is quantitatively formulated as a performance index

of MPC optimization task. As described in Chapter 3, the MPC optimization task is defined by the performance index (cost function) and constraints on decision and state vector of the plant. In has been pointed out in (Brdys *et al*., 2007) that it is impossible to achieve all objectives under wide range of operational conditions by using only one RFMPC controller. Therefore in the overall control system there must be as many RFMPC controllers as number of control strategies in order for the plant to operate properly. Appearance of different control strategies enforces finding a switching mechanism between them. The overall control system needs to be further developed to so-called Softly Switched RFMPC (SSRFMPC) in order to accommodate multiple RFMPC controllers.

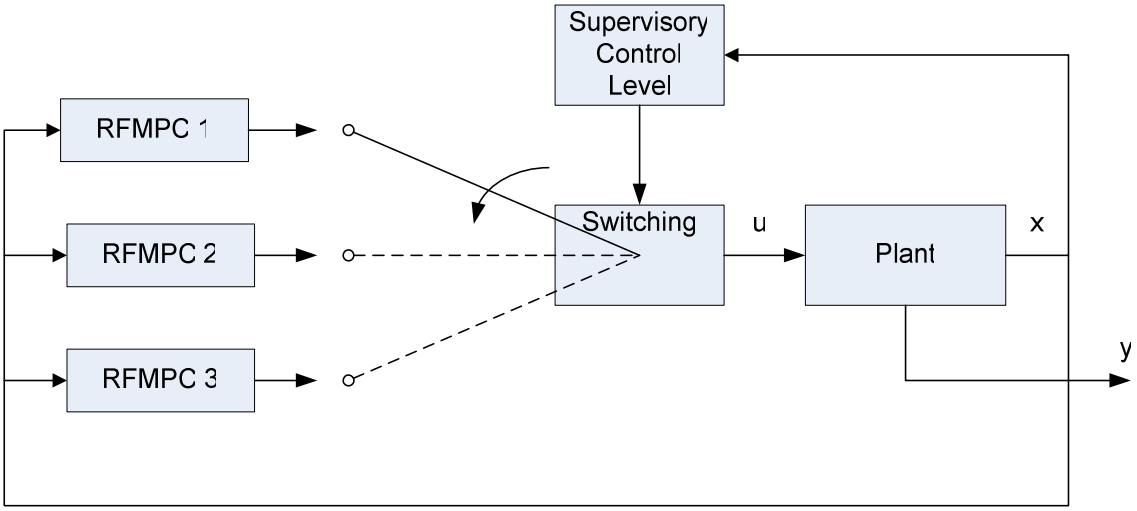


Figure 5.1 Structure of Softly Switched RFMPC

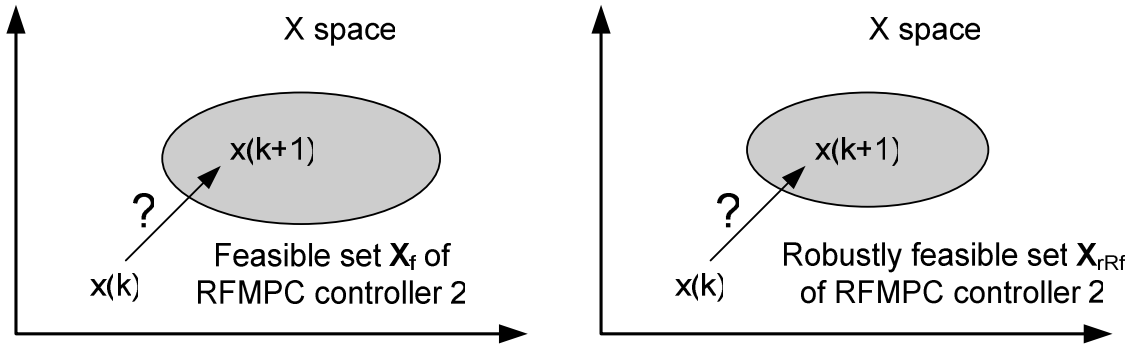
The architecture of the SSRFMPC is depicted in Figure 5.1. Of course in reality there could be possibly more than three different control strategies that need to be used to control the plant. However, for the illustration purposes, the RFMPC 1, RFMPC 2, and RFMPC 3 represent for control strategies of *normal*, *disturbed*, and *emergency* operational states that have described in Section 2.6.6. Each RFMPC can be constructed by the

techniques in Chapter 3 or Chapter 4. However, as discussed before, the technique of Chapter 4 can guarantee the recursive robust feasibility. Moreover, in term of computing aspect, the computational burden of the MPC is done off-line whereas it has to be on-line by the technique of Chapter 3. Therefore, in thesis SSRFMPC will be composed of several RFMPC controllers in which each of them has the standard structure as described in Chapter 4. The operational state is assessed by the Supervisory Control Level (SuCL). The role of the SuCL is to decide and select the proper control strategy based on the assessment of operational state. The switching signal is then transmitted from SuCL to Switching box. The function of the Switching box is to switch between RFMPC controllers. Then the above procedure is repeated over the control horizon.

5.3 Switching Mechanism

In the previous section, switching box has the function that switch from one controller to another. In order to design the switching box, a switching mechanism is required. There are two different ways of switching: hard switching and soft switching. Hard switching means the new control strategy is applied immediately without any intermediate switching process when needed. Although the hard switching is very simple, it does not usually achieve satisfactory switching outcome since the switching is made in such a sudden way that it can probably cause some unexpected and unwelcome impulsive phenomena, such as big overshoot and sudden change of system state/output, huge instant demand and abrupt

change of control action, and even actuator failure if the control objectives of the two control strategies greatly differs.



(a) (b)

Figure 5.2 Feasibility of hard switching MPC controllers: (a) feasible hard switching with the feasible set $\mathbf{X}_f$ of new RFMPC as a target set; (b) robustly feasible hard switching with the robustly feasible set $\mathbf{X}_{rRf}$ of new RFMPC as a target set.

Besides the aggressive switching transients of controller hard switching between different RFMPC controllers can not be always guaranteed. From the set inclusion point of view, each RFMPC controller has its own feasible region. This feasible region has been identified, calculated, and denoted $\mathbf{X}_f$ in Chapter 4. If at the switching point the current initial state provided by the first MPC does not belong to the feasible set $\mathbf{X}_f$ of the second RFMPC, the switching action is then infeasible. Moreover, since uncertainty has to be considered, if at the switching point the initial state provided by the first RFMPC does not belong to the robustly feasible set $\mathbf{X}_{rRf}$ of the second RFMPC, then a robustly feasible operation of RFMPC controller 2 can not be guaranteed. This issued is illustrated in Figure 5.2. Therefore, when the first control strategy and its corresponding RFMPC controller

needs to be replaced by the second one, the command of SuCL to switch can be implemented in two ways: hard switching and soft switching. Hard switching always means that the second RFMPC is put into action immediately. This, as previously explained, can be done only if the current state has been included in at least the feasible set $\mathbf{X}_f$ of RFMPC controller 2. If not, the SuCL may decide to wait till this happens while sacrificing on the operational performance. However, if the feasible set $\mathbf{X}_f$ of MPC controller 2 has no shared intersection with that of RFMPC controller 1, it then becomes hopeless to make such a hard switching no matter how long the SuCL waits, since RFMPC controller 1 can never provide such a suitable initial state for RFMPC controller 2. In this situation, only the soft switching method can be resorted to overcome.

The idea of soft switching is illustrated in Figure 5.3 where the control strategy change is commanded by the SuCL at time instant $\bar{t}$ (Grochowski et al., 2004). It can be seen that a sudden (hard) switch from the current control strategy produces state trajectory $x^{cr}(t \mid \bar{t})$ to the new one forcing the plant state moving along $x^{new}(t \mid \bar{t})$ would imply significant variations of the plant state. In order to avoid it the immediate combined control strategies are designed to produce the intermediate state trajectory $x^{comb}(t \mid \bar{t})$. Hence, over $t \in [\bar{t}, \bar{t}+T_s]$, where T_s denotes duration of the soft switching process, the plant is under the combined control strategies. The new control strategy takes over at $t = \bar{t}+T_s$, to produce the soft state trajectory $x^{soft}(t \mid \bar{t})$.

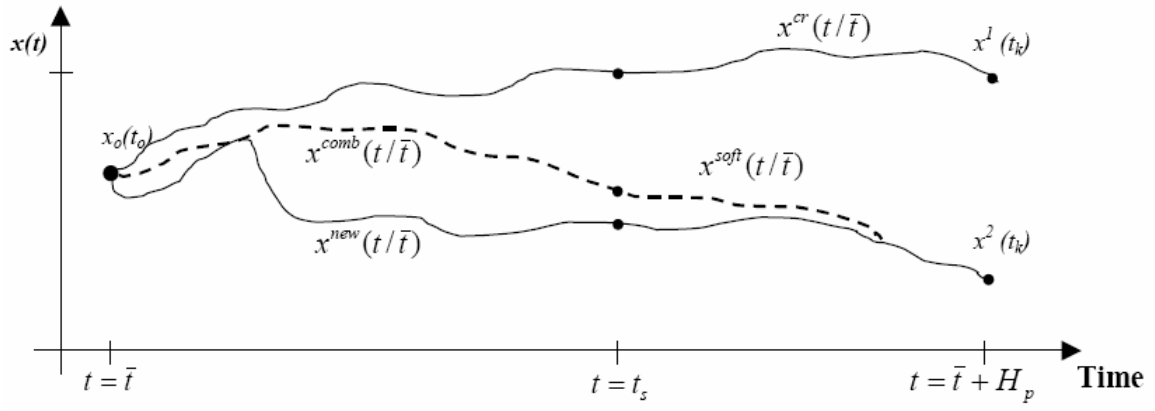


Figure 5.3 The examples of state trajectories resulting from hard and soft switching of the control strategies. The dashed line represents the state trajectory during soft switching.

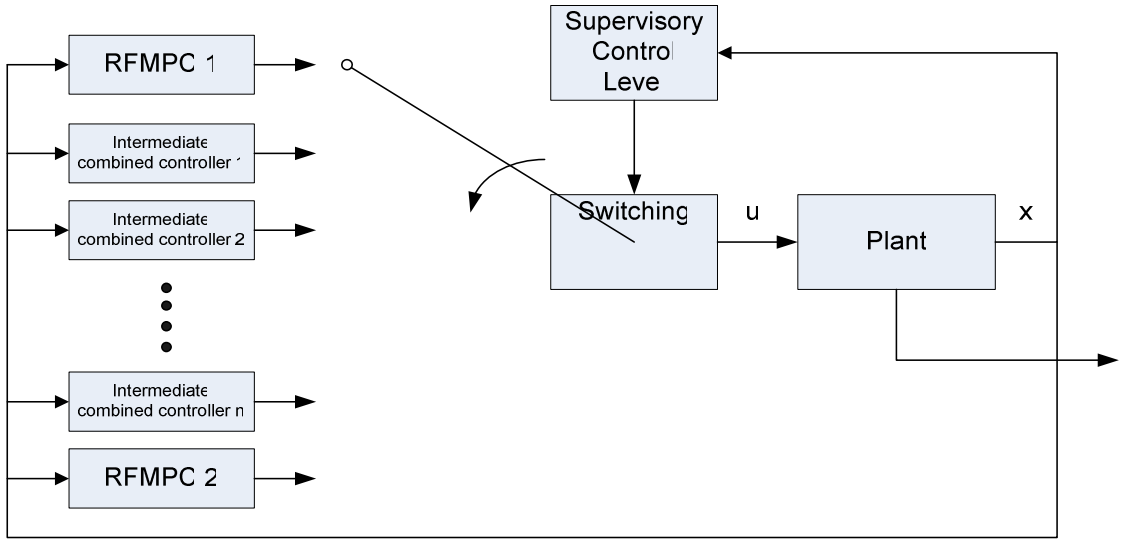


Figure 5.4 Soft switching mechanism – design structure of the SSRFMPC

Motivated by the idea of switching mechanism, the design structure of SSRFMPC is depicted in Figure 5.4 where the soft switching process composes of many intermediate combined RFMPC controllers. By gradually replacing RFMPC 1 by the sequences of intermediate RFMPC controllers, the RFMPC 2 controller will eventually be engaged without causing unwanted transients.

Based on the idea of soft switching mechanism, the Softly Switched MPC (SS-MPC) has been proposed in (Brdys and Wang, 2005) for a linear process dynamics under the additive and polytopic uncertainty. In (Wang, 2006), the SS-MPC has been further investigated and been applied to the DWDS. In this thesis, further improvement on SS-MPC has been made. Firstly, instead of the MPC the RFMPC as described in the Chapter 4 is used in the thesis in order to ensure recursive robust feasibility. Secondly, the SSRFMPC is designed not for linear but nonlinear systems. The technical designs of the soft switching mechanism are presented in the next chapter.

5.4 Supervisory Control Level

In operational control of water distribution systems under varying operating scenarios, the proposed methodology of softly switched model predictive control can be integrated into an intelligent decision-support multilayer hierarchical control structure (Brdys and Tatjewski, 2005), (Brdys and Malinowski, 1994), (Findeisen et al., 1980), (Brdys et al,. 2002), (Grochowski, Brdys & Gminski, 2004), (Brdys, *et al.*, 2007). The multilayer hierarchical control structure has been applied to Integrated Wastewater System and is depicted in Figure 5.5. This control structure allows for proper and through utilization of all available quantitative and qualitative information about the plant structure and dynamics, its interactions with the environment and up to date operational experiences.

According to different functionalities played, the hierarchical control structure can be split into three main control levels: Supervisory Control Level (SuCL), Optimizing Control Level (OCL) and Follow up Control Level (FuCL).

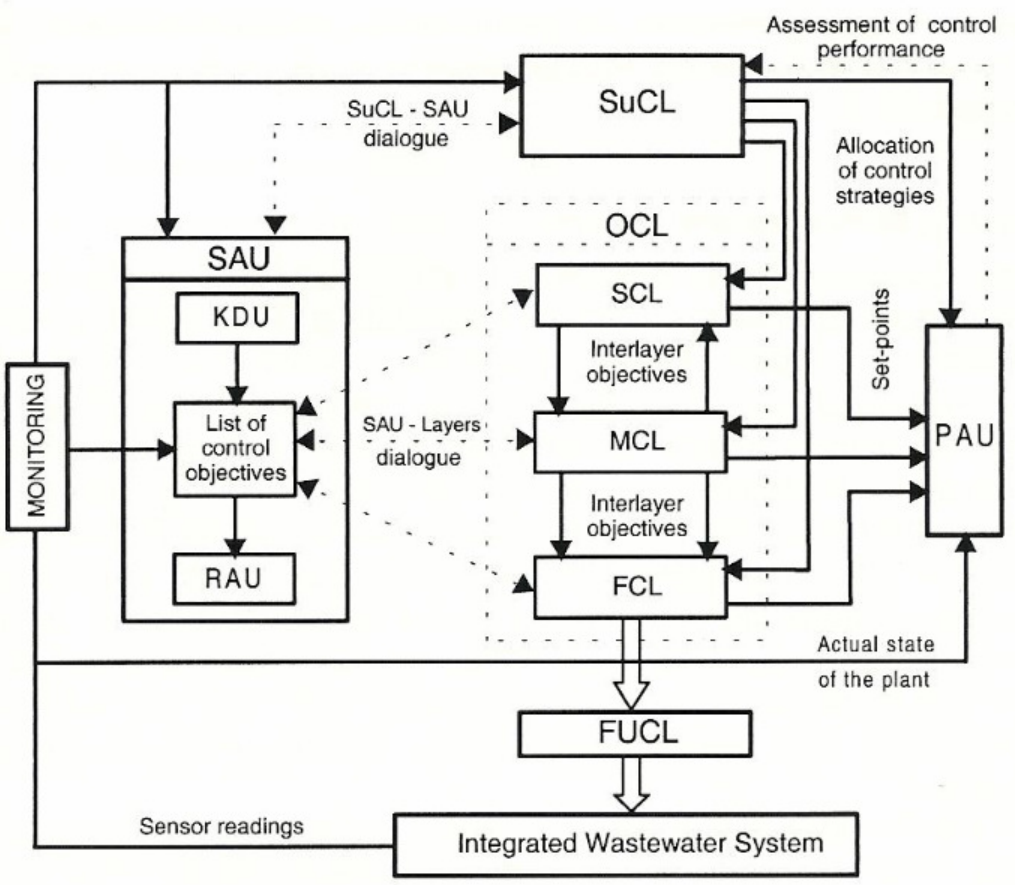


Figure 5.5: Hierarchical structure for optimizing control of integrated wastewater system

Supervisory Control Level is located at the top of the control system hierarchy and has global knowledge about current activity of the entire system. Its functionalities are presented in (Grochowski, et al., 2004) The SuCL is responsible for coordination of operation of all layers and assess the operational states to select the suitable control strategy i.e. objective function and constraints for MPCs at OCL. Information from all control structure units is available at SuCL at every time step with a time resolution that is

adequate to a time unit of a time scale being considered. Based on information delivered by monitoring system, OCL and dedicated agents of SuCL select a control strategy to be currently applied to the system. The OCL is responsible for generating the optimized and robustly feasible trajectories of manipulated variables. The control objectives of OCL can be split into Long, Medium, and Fast control sub-layer. The result of control generation activities at OCL are the manipulated variables trajectories that constitute the set-point trajectories for the FuCL. The FuCL layer is responsible for forcing the plant by direct hardware maneuverings to follow these trajectories.

It can clearly be seen that the SuCL determines whether to favor the transient switching performance or to favor engaging new control strategy as quickly as possible. Therefore, in order to make the right switching decision the SuCL must be involved and designed in the hierarchical decision support control structure. A systematic design of such SuCL requires a comprehensive knowledge of plant operational states e.g. leakage detection, weather, water demand forecast, coordination between other control levels, etc., which is beyond the scope of the thesis.

5.5 Summary

The SSRFMPC and its components have been presented. The two ways switching: hard switching and soft switching has been discussed to point out that the hard switching is not

always feasible. The idea of the soft switching mechanism has been discussed and the design structure of SSRFMPC has been showed and explained for the clear picture of the technical design in the next chapter. In order to have better understanding of operational control of DWDS, the SuCL and its functionalities have been briefly summarized.

Chapter 6

Synthesis, analysis and soft switching mechanisms for Supervisory Controller of SSRFMPC and Application to DWDS under wide range of operational conditions

This chapter presents the analysis of Softly Switched RFMPC. The switching mechanism of SSRFMPC is mathematically formulated. The algorithm for the fast soft switching between RFMPC is proposed in order to minimize the switching time duration. The theory is illustrated by the application to hydraulic optimizing control in DWDS example.

6.1 Introduction

In the operational control of DWDS, it is often that there is a set of objectives that need to be met by the plant under full range of disturbance inputs. In chapter 3 and chapter 4, two different techniques to design RFMPCs have been proposed by utilizing safety zones. However, a single RFMPC represents only one control objective that being best fit into specific operational conditions and therefore is only used for one suitable control strategy. In order for the plants to achieve a set of objectives during operational control of DWDS, it requires that several RFMPC needs to be designed. It is inevitable then to switch between the RFMPCs. A simple hard switching may introduce unwanted transients and more importantly it may not achieve recursive and robust feasibility. A similar soft switching mechanism to (Brdys and Wang 2005) and (Wang and Brdys, 2006), but for nonlinear systems, is presented in the chapter in order to softly switch from one RFMPC to a different one. The resulting controller that is able to softly switch between RFMPC is then called Softly Switched RFMPC (SSRFMPC). The major difference between these two methods originates from the different ways to incorporate and handle the uncertainties. In (Brdys and Wang 2005), (Wang and Brdys, 2006), and (Wang 2006), the linear system is considered and the corresponding robust closed-loop feasible set is calculated by the dedicated algorithm. The algorithm has been proposed in (Grieder, et al. 2003) and utilized the Pontryagin set difference. The work has been further developed and analyzed in (Brdys and Tran, 2010) and (Tran and Brdys, 2010) in which the nonlinear system is considered and the corresponding robustly feasible invariant sets are calculated by utilizing the KKT optimality conditions.

The chapter is organized as follows: The synthesis and analysis of SSRFMPC is presented in Section 6.2. An algorithm is derived for fast soft switching in case the hard switching is not feasible is presented in Section 6.3. The Section 6.4 consists of a DWDS example with sets of comparative simulation results. Section 6.5 summarizes the chapter.

6.2 Softly Switched RFMPC

As described in the Chapter 3, a general nonlinear network system has the following the formulation:

$$\begin{cases} x(k+1) = \widetilde{f}(x(k), s(k), d(k)) \\ y(k) = E(u(k), x(k), d(k)) \end{cases} \tag{6.1}$$

where the state and output equation are described in equations (3.5) and (3.6).

The system (1) is subject to the constraints (3.2a), (3.2b), and (3.2c):

$$u \in \mathbf{U},\ y \in \mathbf{Y},\ x \in \mathbf{X} \tag{6.2}$$

Consider the finite horizon MPC optimization problem at k:

$$\begin{aligned} &\min_{s(\cdot|k),x(k+N|k)} \widetilde{J}(s(\cdot\,|\,k), x(k+N\,|\,k)) \\ &\text{subject to:} \\ &x(k\,|\,k) = x(k) \\ &F(s(k+i\,|\,k), d(k+i\,|\,k)) = 0 \\ &x(k+1+i\,|\,k) = \widetilde{f}(x(k+i\,|\,k), s(k+i\,|\,k), d(k+i\,|\,k)) \\ &x(k+i\,|\,k) \in \mathbf{X},\ i = \overline{0:N} \\ &u(k+i\,|\,k) \in \mathbf{U},\ i = \overline{0:N-1} \\ &y(k+i\,|\,k) \in \mathbf{Y},\ i = \overline{0:N-1} \end{aligned} \tag{6.3}$$

It is important to mention that the formulation (6.1) above considers the general nonlinear system and the impact of disturbances on states/outputs is nonlinear and implicit whereas in (Brdys and Wang, 2005) and (Wang 2006) the linear system is considered and the disturbance state mapping is explicit.

It is clear that the MPC control law Υ above is determined by the function $\tilde{J}(\cdot)$ and the constraint sets $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{U}$. Notice that the MPC with the optimization task by (6.3) is not robustly feasible and the safety zones based technique will be used to modify state constraints of this MPC to make it one step robustly feasible. Moreover, further modification will be needed in order to achieve the recursively robustly feasible MPC. This has been done in the Chapter 4 to produce the MPC recursively achieving robust feasibility. The final formulation of RFMPC optimization task in the reduced space of variables is presented by (4.16). Similarly, the final formulation of RFMPC optimization task in the full space of variables is formulated as followed:

$$
\begin{aligned}
&\min_{s(\cdot|k),x(k+N|k)} \tilde{J}(s(\cdot\,|\,k), x(k+N\,|\,k)) \\
&\text{subject to:} \\
&x_{rRf}^{\min} + \varepsilon^{l} \le x(k+1\,|\,k) \le x_{rRf}^{\max} - \varepsilon^{u} \\
&y^{\min} + \varepsilon_{y}^{l} \le y(k\,|\,k) \le y^{\max} - \varepsilon_{y}^{u} \\
&F(s(k+i\,|\,k), d(k+i\,|\,k)) = 0 \\
&x(k+1+i\,|\,k) = \tilde{f}(x(k+i\,|\,k), s(k+i\,|\,k), d(k+i\,|\,k)) \\
&x(k+i\,|\,k) \in \mathbf{X},\ i = \overline{0:N} \\
&y(k+i\,|\,k) \in \mathbf{Y},\ i = \overline{0:N-1} \\
&u(k+i\,|\,k) \in \mathbf{U},\ i = \overline{0:N-1}
\end{aligned}
\tag{6.4}
$$

The problem considered here is how to guarantee recursive robust feasibility under uncertainty when softly switching from an old RFMPC controller with $\Upsilon(\widetilde{J}_1(\cdot), \mathbf{X}_{old}, \mathbf{U}_{old}, \mathbf{Y}_{old})$ to the new one with $\Upsilon(\widetilde{J}_2(\cdot), \mathbf{X}_{new}, \mathbf{U}_{new}, \mathbf{Y}_{new})$. It should be noticed that the new RFMPC controller differs from the old one not only in the control objective but also in the constraints. Based on robustly feasible invariant sets, further development on soft switching for nonlinear network systems will be presented.

A sequence of intermediate combined RFMPC controllers linking the old and the new RFMPC controllers $\Upsilon(\widetilde{J}_1(\cdot), \mathbf{X}_{old}, \mathbf{U}_{old}, \mathbf{Y}_{old})$ and $\Upsilon(\widetilde{J}_2(\cdot), \mathbf{X}_{new}, \mathbf{U}_{new}, \mathbf{Y}_{new})$ is applied to the plant during the soft switching process.

Performance index of the combined RFMPC controller is designed as follows:

$$\begin{aligned} J_{combined}(s(\cdot \mid k), x(k+N \mid k)) = \; & (1-\alpha(k))\widetilde{J}_1(s(\cdot \mid k), x(k+N \mid k)) \\ + \; & \alpha(k)\widetilde{J}_2(s(\cdot \mid k), x(k+N \mid k)) \end{aligned} \tag{6.5}$$

where $\alpha(k)$ is a weighting scalar at time k. It changes increasingly from 0 to 1 along the time axis during the switching process. A specific value of $\alpha(k)$ determines a specific combined RFMPC. Ideally $\alpha(k)$ should also be included as optimization variables besides $u(\cdot \mid k)$. For simplicity, the optimization problems are only handled with respect to $u(\cdot \mid k)$ but not $\alpha(k)$. If the switching process needs to completed in N time steps, then in practice a simple way to set values of $\alpha(k)$ is by evenly distributing their values between 0 and 1 as: $\alpha(k) = \overline{1:N} / N$.

If we assume the typical additive structures of the performance indices $\widetilde{J}_1(s(\cdot\,|\,k), x(k+N\,|\,k))$ and $\widetilde{J}_2(s(\cdot\,|\,k), x(k+N\,|\,k))$ as follows:

$$\begin{aligned}\widetilde{J}_1(s(\cdot\,|\,k), x(k+N\,|\,k)) &= \sum_{i=0}^{N-1} \overline{J_1}(s(k+i\,|\,k), x(k+N\,|\,k)) \\ \widetilde{J}_2(s(\cdot\,|\,k), x(k+N\,|\,k)) &= \sum_{i=0}^{N-1} \overline{J_2}(s(k+i\,|\,k), x(k+N\,|\,k))\end{aligned} \tag{6.6}$$

then a dedicated soft switching method which accommodate to a more flexible construction of the combined MPC is proposed. The performance index of this general combined MPC is defined as follows:

$$J_{gencom}(s(\cdot\,|\,k), x(k+N\,|\,k)) = \sum_{i=0}^{N-1} \left\{ \begin{array}{l} w_1(i,k)\overline{J_1}(s(k+i\,|\,k), x(k+N\,|\,k)) \\ +\ w_2(i,k)\overline{J_2}(s(k+i\,|\,k), x(k+N\,|\,k)) \end{array} \right\} \tag{6.7}$$

where $w_1(i,k)$ and $w_2(i,k)$ are the weighting vectors. Let the switching time starts at $k = k_s$, the values of $w_1(i,k)$ and $w_2(i,k)$ can be determined by the following algorithm:

Algorithm 6.1: Weighting vectors

- If $k+i<T_s+k_s$, $w_1(i,k) = \lambda^{k-k_s+i}$, $\forall i \in \overline{0:N-1}$,
- If $k+i \geq T_s+k_s$, $w_1(i,k) = 0$, $\forall i \in \overline{0:N-1}$,
- $w_1(N,k) = w_1(N-1,k)$,
- $w_2(i,k) = 1 - w_2(i,k)$, $\forall i \in \overline{0:N}$

where k_s, T_s, and λ denote the switching time instant and the time duration of the soft switching process, and tuning knob respectively, and $0 \leq \lambda \leq 1$.

The benefit of the algorithm is to allow users to have full control of the soft switching process. For example, decreasing T_s certainly speeds up the switching process or varying value of λ affects the transient switching performance. For $\lambda = 0$, the above algorithm can represent hard switching. If $\lambda = 1$, it refers to no switching at all.

It has been pointed out in (Wang and Brdys, 2006) and (Wang, 2006) that tuning of the soft switching parameter λ is problem dependent and the best choice of λ can be found by comparative simulations. For the sake of simplicity, it is assumed in the thesis that $\lambda = 0.5$.

The most direct way to design constraints for the combined MPC is to combine the old constraints and new constraints in a convex manner:

$$\begin{aligned} \mathbf{X}_{combined} &\triangleq (1-\alpha(k))\mathbf{X}_{old} + \alpha(k)\mathbf{X}_{new} \\ \mathbf{U}_{combined} &\triangleq (1-\alpha(k))\mathbf{U}_{old} + \alpha(k)\mathbf{U}_{new} \\ \mathbf{Y}_{combined} &\triangleq (1-\alpha(k))\mathbf{Y}_{old} + \alpha(k)\mathbf{Y}_{new} \end{aligned} \tag{6.8}$$

where $0 \le \alpha(k) \le 1$

By ensuring recursive robust feasibility of a whole switching process, it is meant that optimization problems of all the combined controllers have feasible solutions and the combined state constraints in the real system are satisfied. It has been shown in section 3 that this can be achieved by maintaining for each combined RFMPC controller the initial state in the corresponding invariant set $\mathbf{X}_{rRf}^{combined}$. The control invariant set based design is illustrated in Figure 6.1.

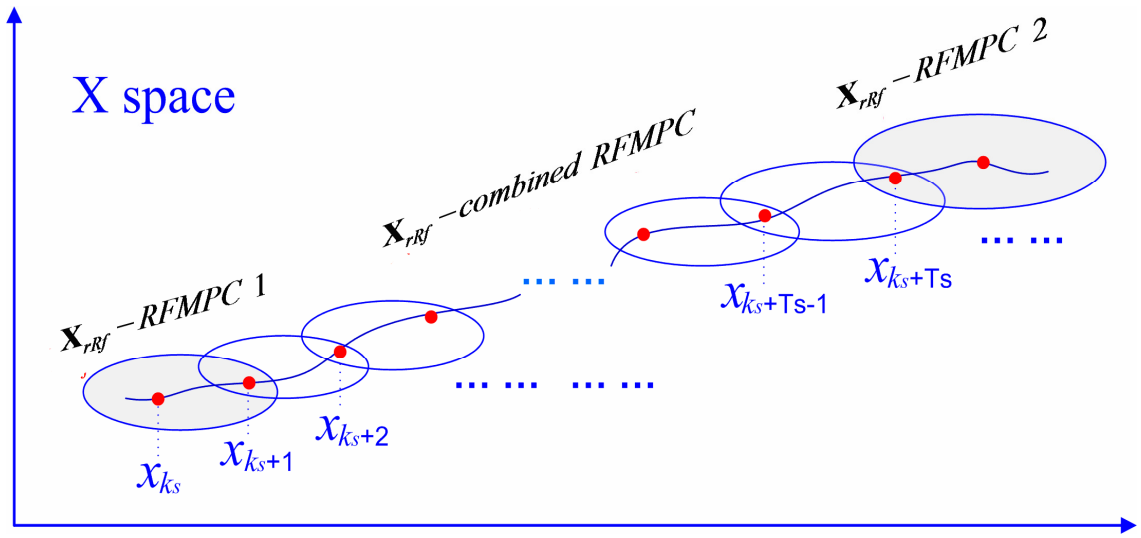


Figure 6.1 Soft switching mechanism: the ellipsoids from left to right represent robustly feasible invariant sets of old RFMPC, combined RFMPCs, and new RFMPC.

Suppose that the switching starts at time k_s. In order for the first combined RFMPC to take over, an effective method is to force the initial state at time k_s+1, not only the model but also from the real plant, to enter the intersection of the robustly feasible invariant sets of the old and the first combined RFMPC. Safety zones need to be embedded into the constraints to take care of the model-reality differences. Thus, an extra constraint (11b) is added to the old RFMPC optimization problem to produce the initial state for the first combined RFMPC, which belongs to $\mathbf{X}_{rRf}^{combinedRFMPC(1)}$:

$$\min_{s(\cdot|k_s),x(k_s+N|k_s)} \tilde{J}(s(\cdot\,|\,k_s), x(k_s+N\,|\,k_s))$$

subject to:

$$x_{rRf}^{\min,com(1)} + \varepsilon^l \le x(k_s+1\,|\,k_s) \le x_{rRf}^{\max,com(1)} - \varepsilon^u$$

$$x_{rRf}^{\min,old} + \varepsilon^l \le x(k_s+1\,|\,k_s) \le x_{rRf}^{\max,old} - \varepsilon^u$$

$$y^{\min,old} + \varepsilon_y^l \le y(k_s\,|\,k_s) \le y^{\max,old} - \varepsilon_y^u$$

$$F(s(k_s+i\,|\,k_s),d(k_s+i\,|\,k_s))=0 \tag{6.9}$$

$$x(k_s+1+i\,|\,k_s)=\widetilde{f}(x(k_s+i\,|\,k_s),s(k_s+i\,|\,k_s),d(k_s+i\,|\,k_s))$$

$$x(k_s+i+1\,|\,k_s)\in \mathbf{X}_{old}$$

$$y(k_s+i\,|\,k_s)\in \mathbf{Y}_{old}$$

$$u(k_s+i\,|\,k_s)\in \mathbf{U}_{old}$$

$$\forall i\in \overline{0:N-1}$$

where $x(k_s)\in \mathbf{X}_{rRf}^{oldRFMPC}$,

$[x_{rRf}^{\min,old},x_{rRf}^{\max,old}]=\mathbf{X}_{rRf}^{oldRFMPC}$, and $[x_{rRf}^{\min,com(1)},x_{rRf}^{\max,com(1)}]=\mathbf{X}_{rRf}^{combinedRFMPC(1)}$.

Applying the same idea to ensure robust feasibility of the second combined RFMPC, the optimization problem at time k_s+1 is formulated as follows:

$$\min_{s(\cdot|k_s+1),x(k_s+1+N|k_s+1)} J_{combined}^{1}(s(\cdot\,|\,k_s+1),x(k_s+1+N\,|\,k_s+1))$$

subject to:

$$x_{rRf}^{\min,com(2)}+\varepsilon^{l}\le x(k_s+2\,|\,k_s+1)\le x_{rRf}^{\max,com(2)}-\varepsilon^{u}$$

$$x_{rRf}^{\min,com(1)}+\varepsilon^{l}\le x(k_s+2\,|\,k_s+1)\le x_{rRf}^{\max,com(1)}-\varepsilon^{u}$$

$$y^{\min,com(1)}+\varepsilon_y^{l}\le y(k_s+1\,|\,k_s+1)\le y^{\max,com(1)}-\varepsilon_y^{u}$$

$$F(s(k_s+1+i\,|\,k_s+1),d(k_s+1+i\,|\,k_s+1))=0 \tag{6.10}$$

$$x(k_s+2+i\,|\,k_s+1)=\widetilde{f}(x(k_s+1+i\,|\,k_s+1),s(k_s+1+i\,|\,k_s+1),d(k_s+1+i\,|\,k_s+1))$$

$$x(k_s+i+2\,|\,k_s+1)\in \mathbf{X}_{combinedRFMPC(1)}$$

$$y(k_s+i+1\,|\,k_s+1)\in \mathbf{Y}_{combinedRFMPC(1)}$$

$$u(k_s+i+1\,|\,k_s+1)\in \mathbf{U}_{combinedRFMPC(1)}$$

$$\forall i\in \overline{0:N-1}$$

where $x(k_s+1)\in \mathbf{X}_{rRf}^{combinedRFMPC(1)}$, and $J_{combined}^{1}(\cdot)$, $\mathbf{X}_{combinedRFMPC(1)}$, $\mathbf{Y}_{combinedRFMPC(1)}$, and $\mathbf{U}_{combinedRFMPC(1)}$ are the designed performance, state constraints, output constraints, and input constraints for the first combined RFMPC respectively.

Note that the computation of the sets is carried out off-line. One needs to compute the robustly feasible invariant sets of these modified combined controllers. In order to obtain $\mathbf{X}_{rRf}^{combinedRFMPC(1)}$, one needs to compute $\mathbf{X}_{rRf}^{combinedRFMPC(2)}$ in advance since the optimization problem of the first combined RFMPC controller has already included the information about the second combined RFMPC controller. The robustly feasible invariant sets of the combined RFMPC have to be calculated backwards in time from the last RFMPC controller to the first one.

Suppose the duration time of switching process is T_s time steps. Before the switching process ends, in order to have recursive and robustly feasible initial state for the new RFMPC controller, the optimization problem at $k+T_s-1$ should be as follows:

$$\min_{s(\cdot|k_s+T_s-1),x(k_s+T_s-1+N|k_s+T_s-1)} J_{combined}^{1}(s(\cdot\,|\,k_s+T_s-1),x(k_s+T_s-1+N\,|\,k_s+T_s-1))$$

subject to:

$$x_{rRf}^{\min,new}+\varepsilon^{l}\le x(k_s+T_s\,|\,k_s+T_s-1)\le x_{rRf}^{\max,new}-\varepsilon^{u}$$

$$x_{rRf}^{\min,com(T_s-1)}+\varepsilon^{l}\le x(k_s+T_s\,|\,k_s+T_s-1)\le x_{rRf}^{\max,com(T_s-1)}-\varepsilon^{u}$$

$$
\begin{aligned}
& y^{\min,com(T_s-1)} + \varepsilon_y^l \le y(k_s+T_s-1 \mid k_s+T_s-1) \le y^{\max,com(T_s-1)} - \varepsilon_y^u \\
& F(s(k_s+T_s-1+i \mid k_s+T_s-1), d(k_s+T_s-1+i \mid k_s+T_s-1)) = 0 \\
& x(k_s+T_s+i \mid k_s+T_s-1) = \\
& \tilde{f}(x(k_s+T_s-1+i \mid k_s+T_s-1), s(k_s+T_s-1+i \mid k_s+T_s-1), d(k_s+T_s-1+i \mid k_s+T_s-1)) \\
& x(k_s+i+T_s \mid k_s+T_s-1) \in \mathbf{X}_{combinedRFMPC(T_s-1)} \\
& y(k_s+i+T_s-1 \mid k_s+T_s-1) \in \mathbf{Y}_{combinedRFMPC(T_s-1)} \\
& u(k_s+i+T_s-1 \mid k_s+T_s-1) \in \mathbf{U}_{combinedRFMPC(T_s-1)} \\
& \forall i \in \overline{0:N-1}
\end{aligned}
\tag{6.11}
$$

After the soft switching process has been completed, the new MPC will be robustly feasible since the initial state of the new controller has entered its robustly feasible invariant sets. Tuning the parameter α should ensures that the feasible sets of two neighbouring RFMPC controllers have nonempty intersection; hence the robustly feasible state transfer is possible.

6.3 An Algorithm for Fast Soft Switching

Ideally, in the event of needing to switch from one RFMPC controller to a different one as fast as possible, one would like to apply the hard switching. However, hard switching is not always possible. In such event, the fast soft switching can be used. The algorithm of the fast soft switching is illustrated by Algorithm 6.2. The algorithm iteratively searches

online to generate a sequence of combined constraint parameter α^k that can achieve a fast soft switching.

Algorithm 6.2: Fast switching algorithm

1) Compute the robustly feasible invariant sets

$$[x_{rRf}^{\min,newRFMPC}, x_{rRf}^{\max,newRFMPC}]$$

If $x(k_s) \notin [x_{rRf}^{\min,newRFMPC}, x_{rRf}^{\max,newRFMPC}]$, then go to step 2;

else, do hard switching at time k_s, and go to step 10;

2) Let flag $l = 0$ and $\alpha^l = 1$

3) Let $l = l + 1$ and $\alpha^l = ½$

4) Design the l-th combined RFMPC's constraints:

$$\begin{cases} x_{rRf}^{\min,newRFMPC} + \varepsilon^l \le x(k+1|k) \le x_{rRf}^{\max,newRFMPC} - \varepsilon^u \\ x_{rRf}^{\min,com(l)} + \varepsilon^l \le x(k+1|k) \le x_{rRf}^{\max,com(l)} - \varepsilon^u \\ y^{\min,com(l)} + \varepsilon_y^l \le y(k|k) \le y^{\max,com(l)} - \varepsilon_y^u \\ F(s(k+i|k), d(k+i|k)) = 0 \\ x(k+1+i|k) = \tilde{f}(x(k+i|k), s(k+i|k), d(k+i|k)) \\ x(k+i|k) \in (1-\alpha^l)\mathbf{X}_{old} + \alpha^l \mathbf{X}_{new} \quad \forall i \in \overline{1:N} \\ y(k+i|k) \in (1-\alpha^l)\mathbf{Y}_{old} + \alpha^l \mathbf{Y}_{new} \quad \forall i \in \overline{0:N-1} \\ u(k+i|k) \in (1-\alpha^l)\mathbf{U}_{old} + \alpha^l \mathbf{U}_{new} \quad \forall i \in \overline{0:N-1} \end{cases}$$

5) Calculate robustly feasible invariant set $[x_{rRf}^{\min,com(l)}, x_{rRf}^{\max,com(l)}]$ of the l-th combined MPC controller by using the Algorithm 4.1:

If $[x_{rRf}^{\min,com(l)}, x_{rRf}^{\max,com(l)}] \neq \varnothing$, then go to step 6;

else, let $\alpha^l = ½\left(\alpha^l + \alpha^{l-1}\right)$ and repeat step 4 and 5;

6) If $x(k_s) \notin [x_{rRf}^{\min,com(l)}, x_{rRf}^{\max,com(l)}]$, then go to step 7;

   else, let $T_s = l$, $\alpha^k \in \alpha^{\overline{T_s:1}}$, and go to step 10;

7) Let $l = l+1$ and $\alpha^l = \frac{1}{2}\alpha^{l-1}$

8) Design the l-th combined RFMPC's constraints:

$$\begin{cases} x_{rRf}^{\min,com(l-1)} + \varepsilon^l \le x(k+1|k) \le x_{rRf}^{\max,com(l-1)} - \varepsilon^u \\ x_{rRf}^{\min,com(l)} + \varepsilon^l \le x(k+1|k) \le x_{rRf}^{\max,com(l)} - \varepsilon^u \\ y^{\min,com(l)} + \varepsilon_y^l \le y(k|k) \le y^{\max,com(l)} - \varepsilon_y^u \\ F(s(k+i|k), d(k+i|k)) = 0 \\ x(k+1+i|k) = \tilde{f}(x(k+i|k), s(k+i|k), d(k+i|k)) \\ x(k+i|k) \in (1-\alpha^l)\mathbf{X}_{old} + \alpha^l \mathbf{X}_{new} \quad \forall i \in \overline{1:N} \\ y(k+i|k) \in (1-\alpha^l)\mathbf{Y}_{old} + \alpha^l \mathbf{Y}_{new} \quad \forall i \in \overline{0:N-1} \\ u(k+i|k) \in (1-\alpha^l)\mathbf{U}_{old} + \alpha^l \mathbf{U}_{new} \quad \forall i \in \overline{0:N-1} \end{cases}$$

9) Repeat step 5 – 8

10) End.

Algorithm 6.2 has two iterative loops. One is to build new combined RFMPC controllers moving backwards which robustly feasible invariant sets are not empty and are forced to intersect. The other is to decide whether more combined RFMPC controllers need to be built by checking the inclusion of the current state $x(k_s)$. The convergence rate of the first iteration loop depends on the distance in state space between the robustly feasible invariant sets of the old RFMPC and the new one. The convergence of the second iteration loop is determined by the current state $x(k_s)$ at the switching time k_s, which also implies that finding a proper switching time k_s can facilitate termination of the algorithm.

In step 5) of the Algorithm 6.2, the robustly feasible set $[x_{rRf}^{\min,com(l)}, x_{rRf}^{\max,com(l)}]$ of the l-combined MPC controller is calculated by using the Algorithm 4.1. Hence, the computational time of the algorithm 6.2 significantly depends on the convergence rate of the Algorithm 4.1. However, the convergence of the algorithm 4.1 has not been proven and only been verified by the simulation. In general, algorithm 4.1 cannot be ensured to terminate in finite time, but confining the iteration times and defining precision degree are usual ways to obtain an approximate solution. In the section 4.5.1, it takes about 10 hours in real time to calculate the robustly invariant set for the small water network. Hence, the computational time of Algorithm 6.2 is much more than that due to the repeatedly utilizing Algorithm 4.1. In order to achieve the results in the thesis in affordable time, the stopping criteria for Algorithm 6.2 are described as below:

- Generations is determined by the flag l and is set $l = 5$. It means that if $l > 5$, the algorithm stops.
- Time limit is set 48 hours. It means that if the computation takes more than 48 hours, the algorithm stops.

6.4 DWDS case study

In this section, four sets of simulation results of the DWDS are presented under MATLAB EPANET environment. The first one is to show the comparison between the soft switching and hard switching in the non-leaky DWDS example. The second one is to confirm that the hard switching is not always feasible and fast soft switching can be used in such events.

The leaky DWDS network is considered in the third set of simulation. The last set of simulation results shows the comparisons between hard switching and soft switching when they are both applied to the DWDS under full range of operational conditions.

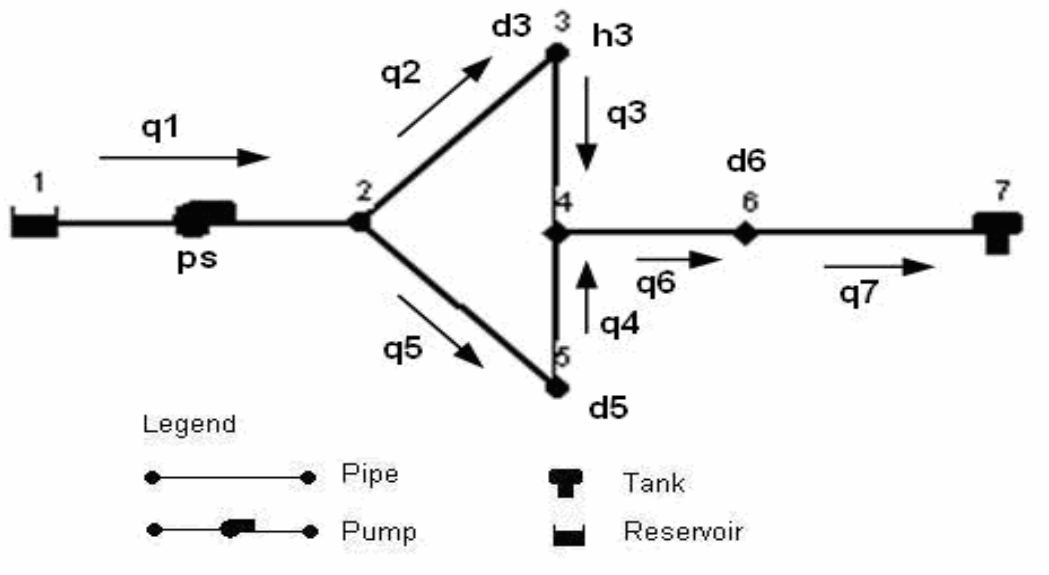


Figure 6.2 Diagram of the DWDS example

The DWDS example is depicted in Figure 6.2 which has the same structure as the one in Chapter 4. The detail profile of this DWDS example can be viewed in the Appendix D.

6.4.1 SSRFMPC designs

In order to apply SSRFMPC to the DWDS, the components of SSRFMPC need to be designed. Specifically, RFMPC controller needs to be designed for each control strategy Moreover, the intermediated RFMPC also needs to be design for the soft switching can take action. In this section, three typical control strategies of the DWDS and their

corresponding robustly feasible invariant sets are shown. The intermediated RFMPC is constructed by producing convex combined performance indexes and convex combined constraints as described in Section 6.2.

6.4.1.1 Predictive control strategies formulation

The general formulations of three typical control strategies that have been described in the Section 2.6.6 are applied to the DWDS example.

Control strategy of *normal operational states*:

In order to achieve a sustainable operation day after day, it is expected that tank levels can come back to their original states after a certain period of operation. Hence, the overall objective function for the normal control strategy is:

$$\begin{aligned} J_n &= \sum_{k=k_0}^{k_0+23} \frac{\gamma(k)}{\eta(k)} q_1(k)(h_2(k)-h_1(k)) + \rho\left|h_7(k_0+N)-h_7(k_0)\right| \\ &= \sum_{k=k_0}^{k_0+23} \overline{J}_n(k) + \rho\left|h_7(k_0+N)-h_7(k_0)\right| \end{aligned} \tag{6.12}$$

where $\gamma(k)$ is the power unit charge in £/kWh, $\eta(k)$ is the pump efficiency and is set $\eta(k)=0.8$ for all k, $h_1(k)$ is the head of the source and is set $h_1(k)=5$ for all k.

As the RFMPC optimization task in the reduced space of decision variables is implemented, the equality constraints i.e. nodal flow continuity equations, head-low equations, volume mass balance equations are automatically embedded and handled by the EPANET. In addition to that, the inequality constraints need to be also embedded into the optimization:

- Physical tank limit is $[2,8]$ and is equivalent to $12 = h_7^{\min} \le h_7(k) \le h_7^{\max} = 18$
- Monitored pressure limit: $13 = h_4^{\min} \le h_4(k) \le h_4^{\max} = 16$
- Relative pump speed limit: $0.2 = u^{\min} \le u(k) \le u^{\max} = 1.1$

Control strategy of *disturbed operational states*:

$$\begin{aligned} J_d &= \xi_n J_n + \xi_d \sum_{k=k_0}^{k_0+23} \sum_{1}^{7} h_i(k) + \rho \left| h_7(k_0+N) - h_7(k_0) \right| \\ &= \sum_{k=k_0}^{k_0+23} \left(\xi_n \overline{J}_n(k) + \xi_d \sum_{1}^{7} h_i(k) \right) + \rho \left| h_7(k_0+N) - h_7(k_0) \right| \end{aligned} \tag{6.13}$$

where ξ_n and ξ_d are the suitably chosen weights.

The optimization constraints for this control strategy are the same as those for the *normal* control strategy.

Control strategy of *emergency operational states*:

$$\begin{aligned} J_d &= \xi_n J_n + \xi_d \sum_{k=k_0}^{k_0+23} \sum_{1}^{2} lq_i(k) + \rho \left| h_7(k_0+N) - h_7(k_0) \right| \\ &= \sum_{k=k_0}^{k_0+23} \left(\xi_n \overline{J}_n(k) + \xi_d \sum_{1}^{2} lq_i(k) \right) + \rho \left| h_7(k_0+N) - h_7(k_0) \right| \end{aligned} \tag{6.14}$$

where ξ_n and ξ_d are the suitably chosen weights.

The optimization inequality constraints for this control strategy are the same as those for the *normal* control strategy. Although the equality constraints are no longer the same as normal control strategy due to the existence of the leaky nodes, all equality constraints are

automatically embedded and handled by EPANET. The reason has been mentioned before i.e. the optimization task is formulated in the reduced spaces of decision variables.

6.4.1.2 Soft switching between control strategies

In the previous subsection, predictive control strategies corresponded to three different operational states of the DWDS example have been presented. In order to softly switch between those three corresponding control strategies, the intermediate combined strategies have the following combined performance index:

- Soft switching between *normal* control strategy and *disturbed* control strategy:

$$J = \sum_{k=k_0}^{k_0+23} \left\{ w_1(k,k_0)\overline{J}_n(k) + w_2(k,k_0)\left(\xi_n \overline{J}_n(k) + \xi_d \sum_{1}^{7} h_i(k) \right) \right\} + \rho \left| h_7(k_0+N) - h_7(k_0) \right| \tag{6.15}$$

- Soft switching between *disturbed* control strategy and *emergency* control strategy:

$$J = \sum_{k=k_0}^{k_0+23} \left\{ w_1(k,k_0)\left(\xi_n \overline{J}_n(k) + \xi_d \sum_{1}^{7} h_i(k) \right) + w_2(k,k_0)\left(\xi_n \overline{J}_n(k) + \xi_d \sum_{1}^{7} h_i(k) \right) \right\} + \rho \left| h_7(k_0+N) - h_7(k_0) \right| \tag{6.16}$$

- Soft switching between *emergency* control strategy and *normal* control strategy:

$$J = \sum_{k=k_0}^{k_0+23} \left\{ w_1(k,k_0) \left(\xi_n \overline{J}_n(k) + \xi_d \sum_{1}^{7} h_i(k) \right) + w_2(k,k_0) \overline{J}_n(k) \right\} + \rho \left| h_7(k_0+N) - h_7(k_0) \right| \tag{6.17}$$

where w_1 and w_2 are dynamic weighting vectors. Their values are not only different within the prediction horizon but also varying with every new time steps. The values of w_1 and w_2 can be generated by Algorithm 6.1 and is illustrated as follows:

The prediction horizon, time duration of soft switching, tuning knob respectively are $N = 24$, $T_s = 2$, $\lambda = 0.5$. The switching time is assumed to be $k_s = 10$ for the demonstration. In the simulation, switching time instant can be selected differently.

At time step $k = 10$ (switching process begins):

$$w_1 = [1\ 0.5\ 0\ 0\ 0\ 0 \ldots 0]$$

$$w_2 = [0\ 0.5\ 1\ 1\ 1\ 1 \ldots 1]$$

At time step $k = 11$

$$w_1 = [0.5\ 0\ 0\ 0\ 0 \ldots 0\ 0]$$

$$w_2 = [0.5\ 1\ 1\ 1\ 1 \ldots 1\ 1]$$

At time step $k = 12$ (switching process ends)

$$w_1 = [0\ 0\ 0\ 0 \ldots 0\ 0\ 0]$$

$$w_2 = [1\ 1\ 1\ 1 \ldots 1\ 1\ 1]$$

The inequality constraints of the control strategies are the same during operational control of the DWDS. Therefore, those constraints are also used as the intermediate combined constraints during soft switching process.

6.4.1.3 Robustly feasible invariant sets

One of the important steps in designing SSRFMPC is to compute in advance the robustly feasible invariant sets for *normal operational states,* disturbed *operational states,* and *emergency operational states*. The computer implementation utilises MATLAB-EPANET simulation environment (Rossmand, 2000) which has been described in Chapter 3 and is not repeated here. By applying the Algorithm 4.1, the sets $\mathbf{X}_{rRf}^{normal} = [3.35, 7.21]$, $\mathbf{X}_{rRf}^{disturbed} = [3.13, 6.05]$, and $\mathbf{X}_{rRf}^{emergency} = [3.25, 6.18]$ were calculated and the control strategies were designed for each of the operational states. The algorithm iterations are illustrated in Table 6.1- 6.3. It is worth of mentioning that in order to decrease the computing time the optimisation problems were performed in the space of decision variables reduced to the control inputs. However, it is still very time consuming to carry such heavy computations. Specifically, the computing time to achieve the results as described in each of the Table 6.1– 6.3 takes approximately 8 hours to 10 hours in real time. The following stopping criteria have been used in order to achieve the results in affordable time:

- Maximum change in function value is set to be 0.05. It means that when the change in value of the optimized function is less than 0.05, the algorithm stops.

- Stall time limit is set to be 7200 seconds. It means that if there is no improvement in the value of the optimized function for an interval of time in seconds specified by stall time limit, the algorithm stops.
- Generations is set to be 10. It means that the maximum number of generations the algorithm performs is 10.

Normal control strategy – Pumping cost minimization		
Number of iterations	$\tilde{x}_{Rf}^{\min}$	$\tilde{x}_{Rf}^{\max}$
0	2.00	8.00
1	2.41	7.70
2	2.54	7.63
3	2.85	7.34
4	2.93	7.21
5	3.33	7.21
6	3.35	7.21

Table 6.1 Iterations resulting in robustly feasible invariant sets for *normal* control strategy

Disturbed control strategy – Excessive pressure minimization		
Number of iterations	$\tilde{x}_{Rf}^{\min}$	$\tilde{x}_{Rf}^{\max}$
0	2.00	8.00
1	2.45	7.66
2	2.80	7.20
3	2.85	6.63
4	3.10	6.18
5	3.13	6.05

Table 6.2 Iterations resulting in robustly feasible invariant sets for *disturbed* control strategy

Emergency control strategy –Leakage minimization		
Number of iterations	$\tilde{x}_{Rf}^{\min}$	$\tilde{x}_{Rf}^{\max}$
0	2.00	8.00
1	2.45	7.66
2	2.80	7.22
3	2.82	6.73
4	3.17	6.17
5	3.20	6.18

Table 6.3 Iterations resulting in robustly feasible invariant sets for *emergency* control strategy

6.4.2 Non-leaky network: Soft switching and hard switching

In the event of no leakage is involved, the control strategies were applied separately to control DWDS over 24h period and the results were compared with regard to the overall energy cost and soft pressure constraints. Comparative simulation results of relative pump speeds, nodal pressures, tank level, and flows between those two control strategies are illustrative in Figure 6.3 – 6.9. It can be clearly seen that the pumping cost minimization control strategy utilizes the electricity tariff and the pumping speed is more active during off-peak hours (see Figure 6.3). However, the excessive pressure minimization control strategy does not take the electricity tariff into account and the pumping action is mainly affected by the daily demand profile. Therefore, regarding the nodal pressures the excessive pressure minimization achieves smaller peak values than pumping cost minimization (see Figure 6.4 – 6.5).

To demonstrate the soft switching, node 4 which is the critical node is selected for the assessment. Although the normal control strategy achieves less energy cost, it can be seen in Figure 6.4 that the pressure at node 4 reaches high value and violates the soft constraint $[13.6, 15.6]$ during 5-8 hours. In order to prevent against the pipe burst, it is required that at 5 hour, the normal control strategy is to be switched to the disturbed one in order to reduce the pressure profile. Then at 12 hours, when the pressure is back to normal, the disturbed control strategy is to be switched back to the normal one. One could apply either hard or soft switching. Designing the latter the soft switching time was chosen as $T_s = 2$ time steps. The combined RFMPC was constructed by using (6.15) and $w(i,k)$ was generated by the Algorithm 6.1 with $\lambda = 0.5$. The invariant sets of the combined RFMPCs were also computed off-line to obtain $\mathbf{X}_{rRf}^{CombinedMPC(1)} = [3.25, 6.88]$ and $\mathbf{X}_{rRf}^{CombinedMPC(2)} = [3.21, 6.26]$. The SSRFMPC and hard switching RFMPC were applied to control DWDS. The resulting pressure trajectory at the node 4 is shown in Figure 6.11. It can be seen that the switched strategies nicely managed to reduce the pressure of node 4 and satisfy the soft constraint during 5-8 hours. However, the hard switching strategy produces the switching pressure transients, which are rapid as opposed to the soft switching transients. It is so because the SSRFMPC distributes the actions over the switching period, hence it smoothes the unwanted transient and achieves smaller peak values. The comparison between the impact of hard switching and soft switching on pump speed, tank level, and nodal pressures are illustrated in Figure 6.10 -6-16.

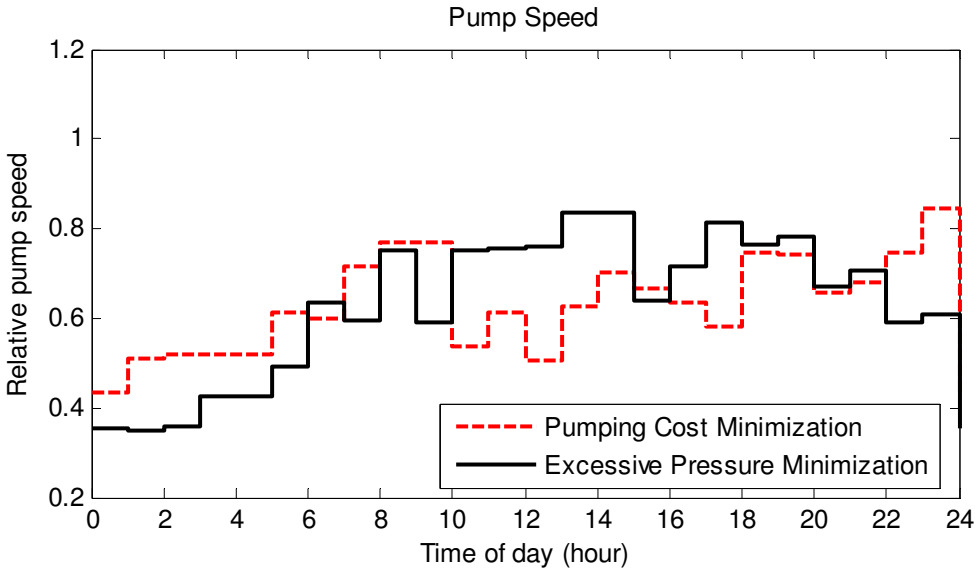


Figure 6.3 Non-leaky operational scenarios – Relative pump speed: the least pumping cost control (red dashed line) and the least excessive pressure control (black solid line)

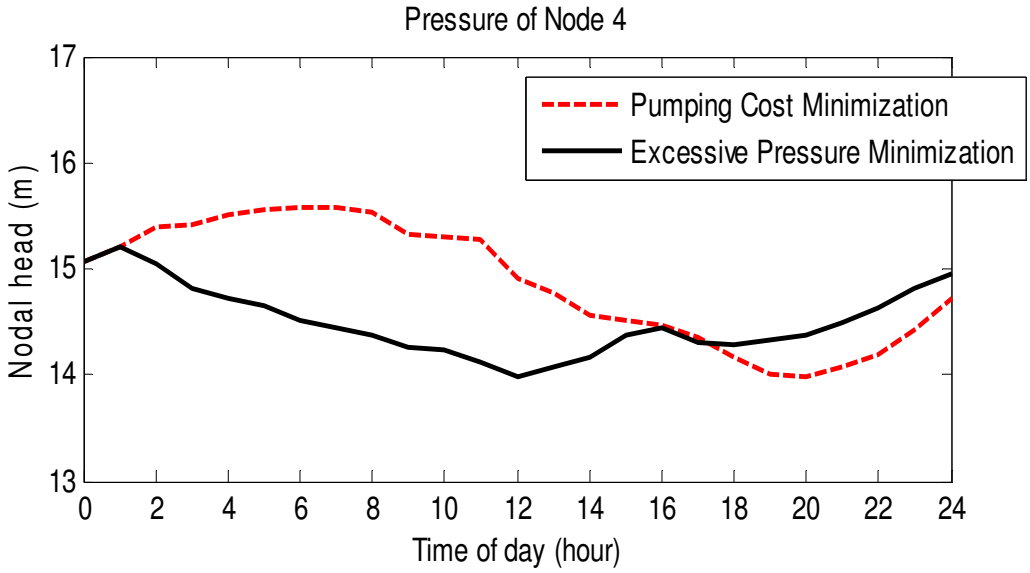


Figure 6.4 Non-leaky operational scenarios - Pressure at node 4: the least pumping cost control (red dashed line) and the least excessive pressure control (black solid line)

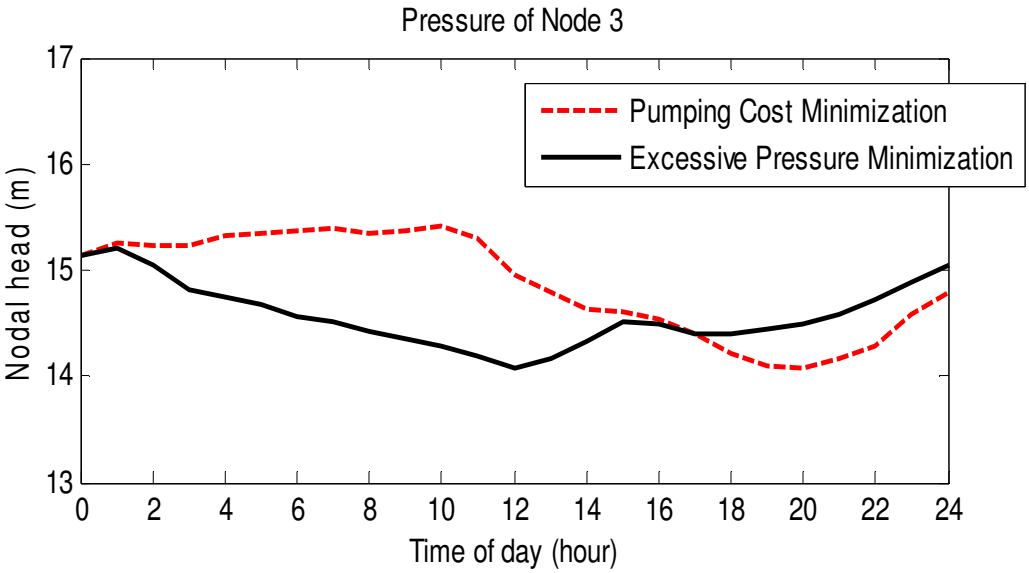


Figure 6.5 Non-leaky operational scenarios - Pressure at node 3: the least pumping cost control (red dashed line) and the least excessive pressure control (black solid line)

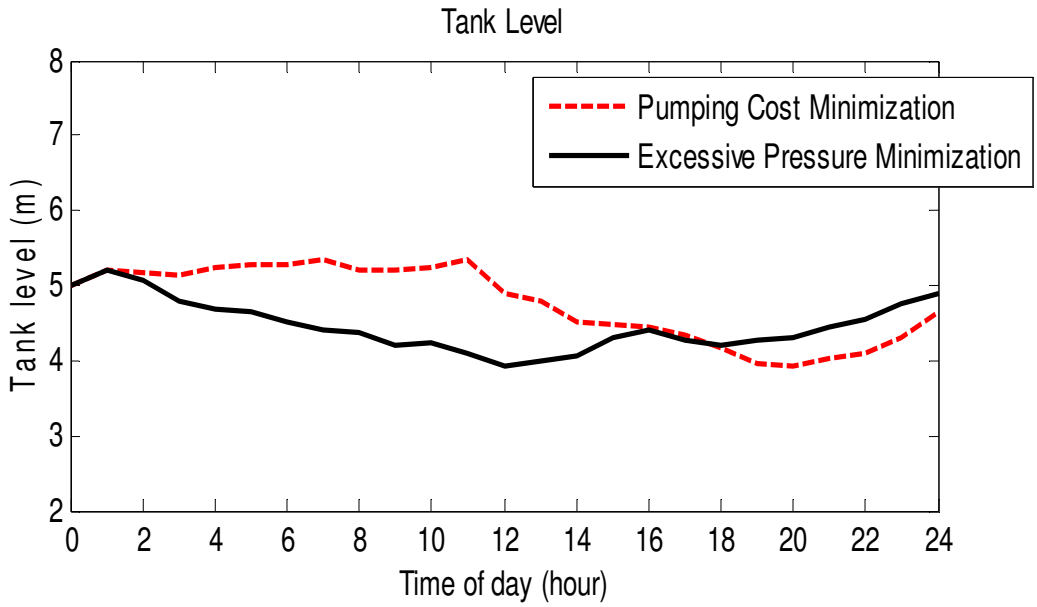


Figure 6.6 Non-leaky operational scenarios - Tank Level: the least pumping cost control (red dashed line) and the least excessive pressure control (black solid line)

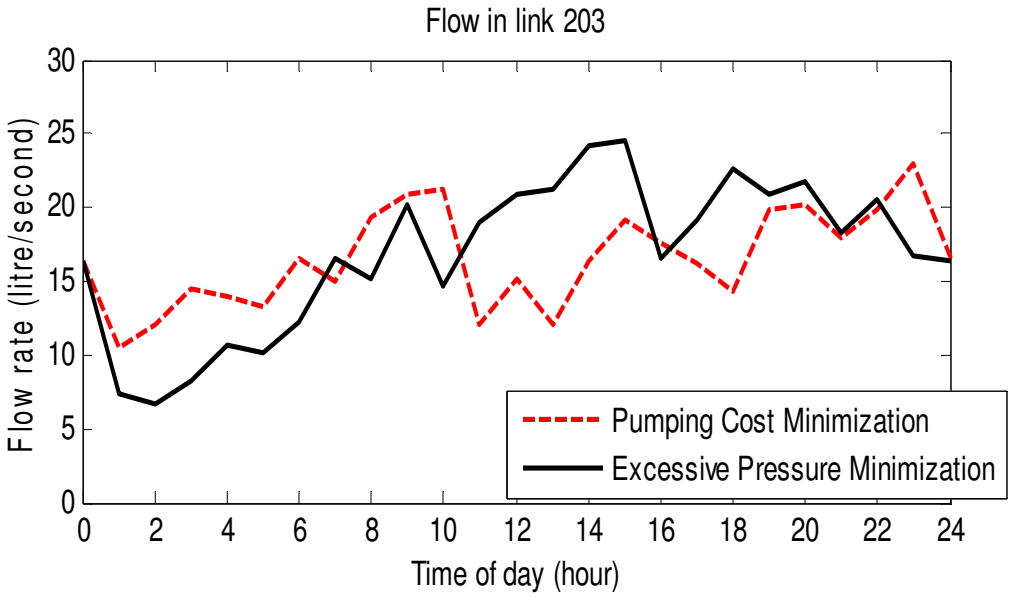


Figure 6.7 Non-leaky operational scenarios - Flow in link 203: the least pumping cost control (red dashed line) and the least excessive pressure control (black solid line)

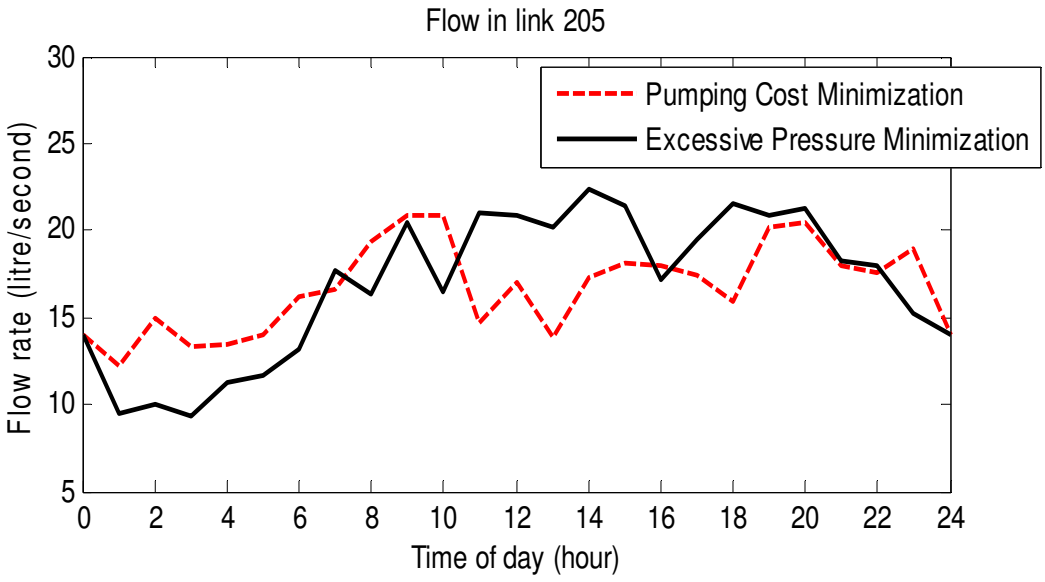


Figure 6.8 Non-leaky operational scenarios - Flow in link 205: the least pumping cost control (red dashed line) and the least excessive pressure control (black solid line)

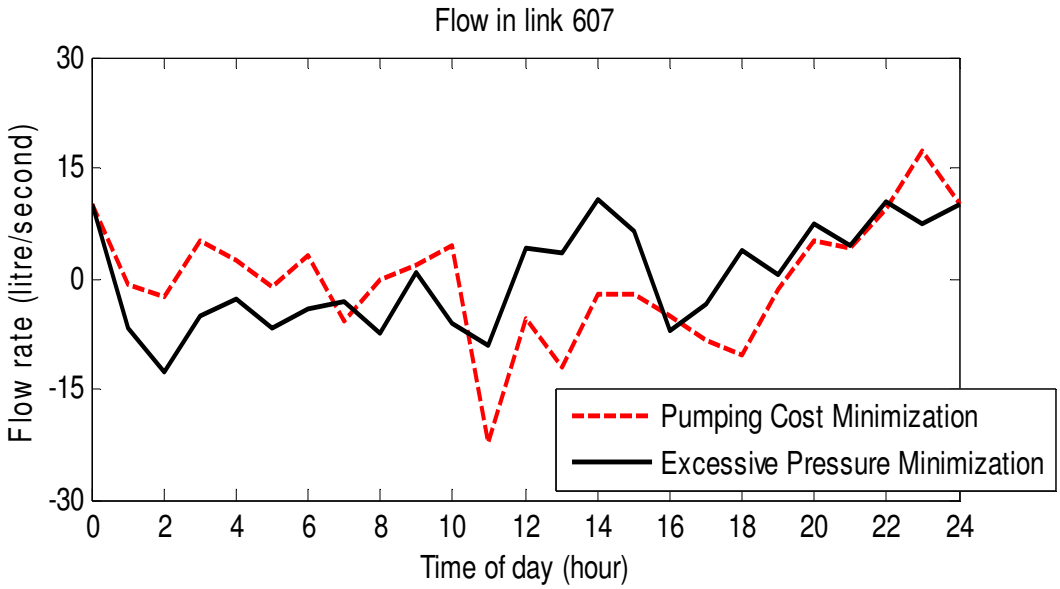


Figure 6.9 Non-leaky operational scenarios - Flow in link 607: the least pumping cost control (red dashed line) and the least excessive pressure control (black solid line)

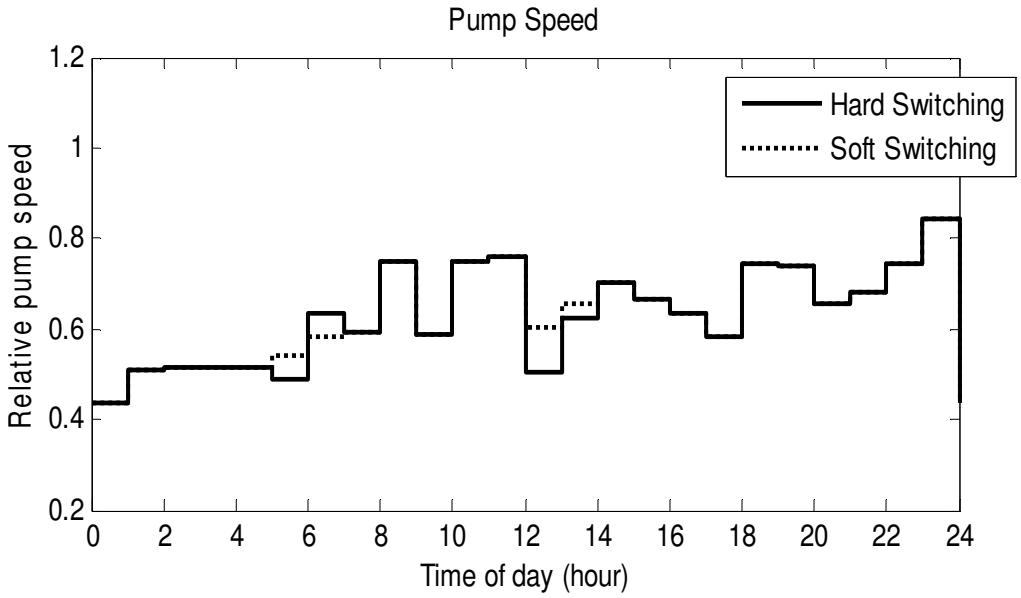


Figure 6.10 Hard switching and soft switching – Relative pump speed: hard switching (solid line) and soft switching (dotted line)

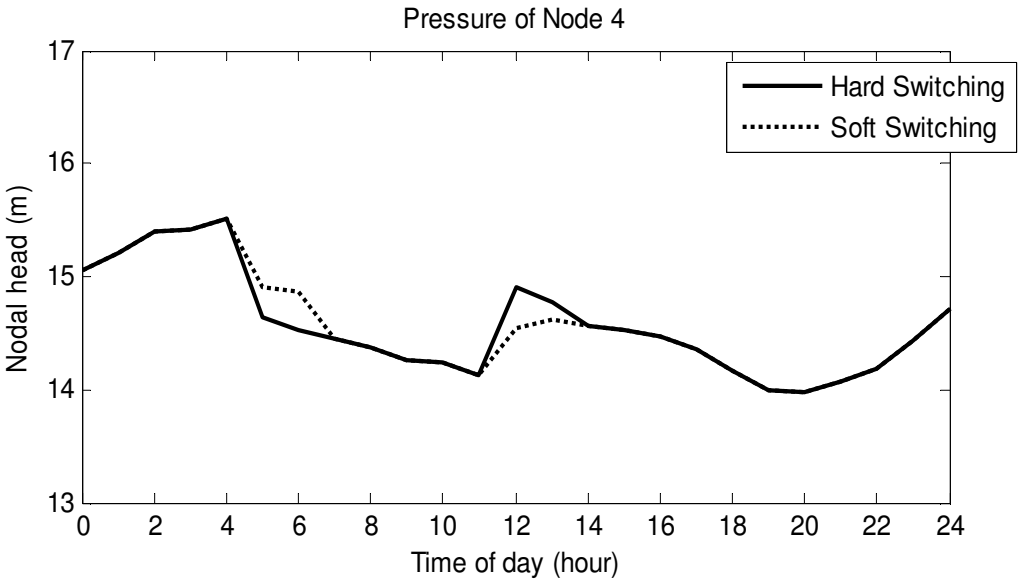


Figure 6.11 Hard switching and soft switching – Pressure of Node 4:

hard switching (solid line) and soft switching (dotted line)

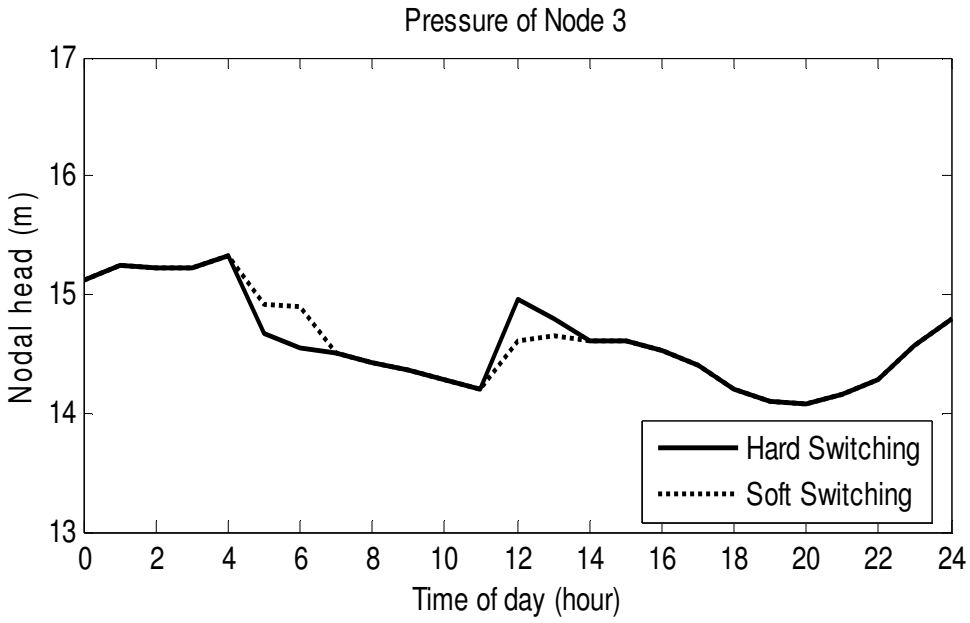


Figure 6.12 Hard switching and soft switching – Pressure of Node 3:

hard switching (solid line) and soft switching (dotted line)

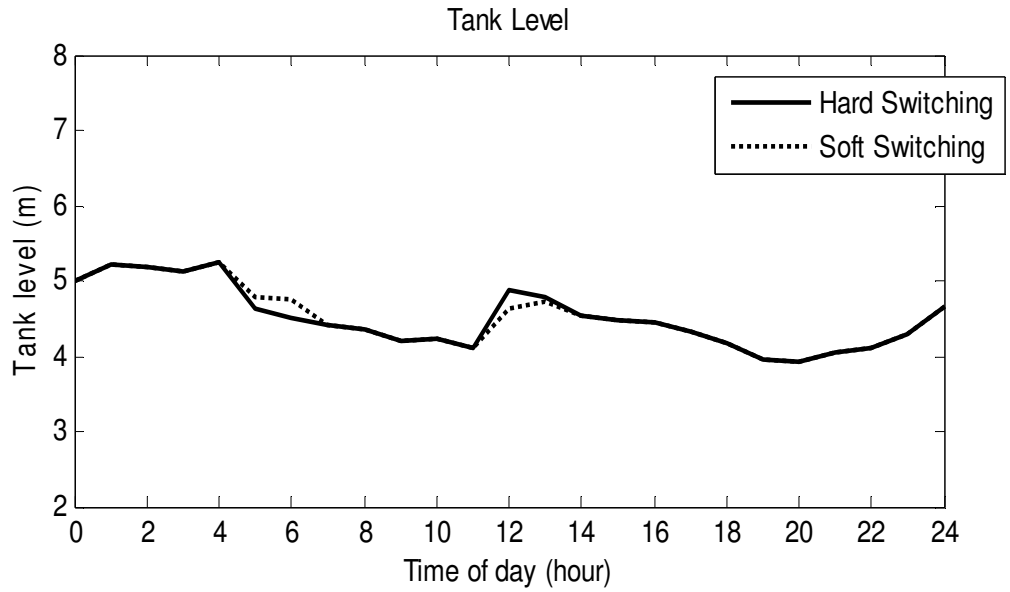


Figure 6.13 Hard switching and soft switching – Tank Level:

hard switching (solid line) and soft switching (dotted line)

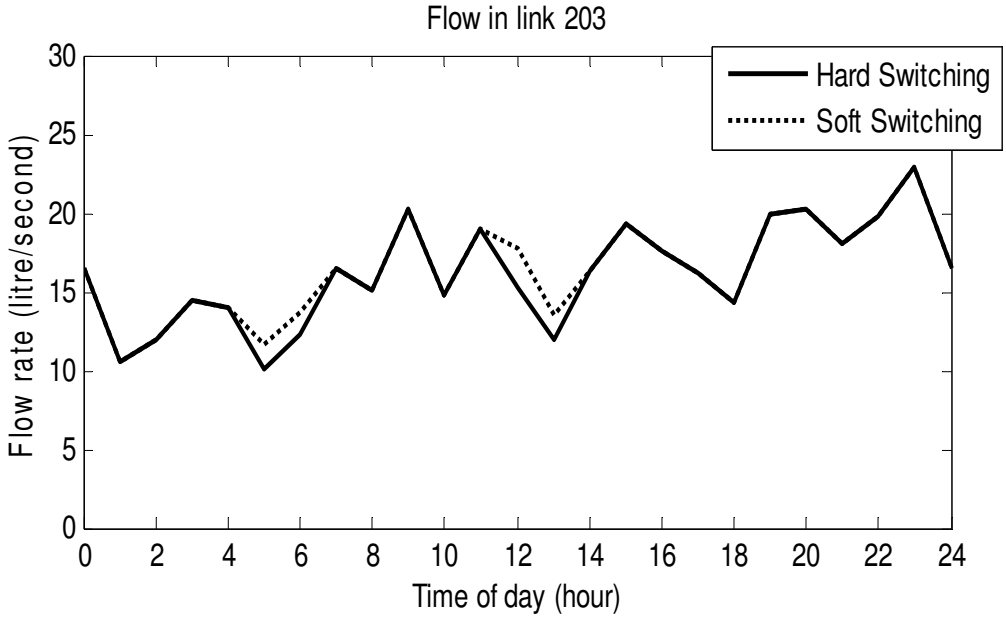


Figure 6.14 Hard switching and soft switching – Flow in link 203:

hard switching (solid line) and soft switching (dotted line)

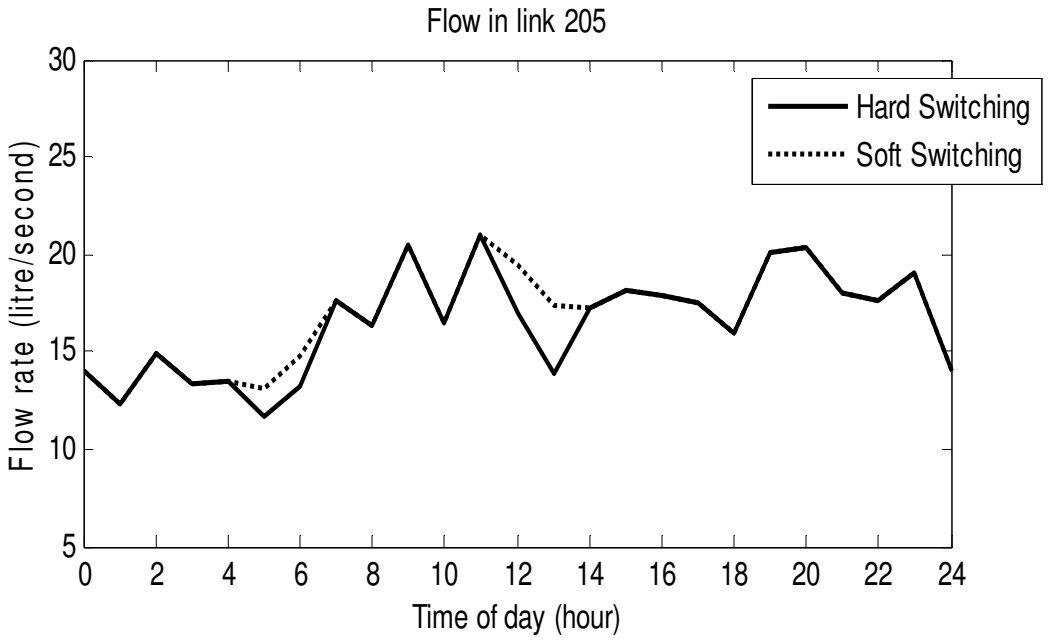


Figure 6.15 Hard switching and soft switching – Flow in link 203:

hard switching (solid line) and soft switching (dotted line)

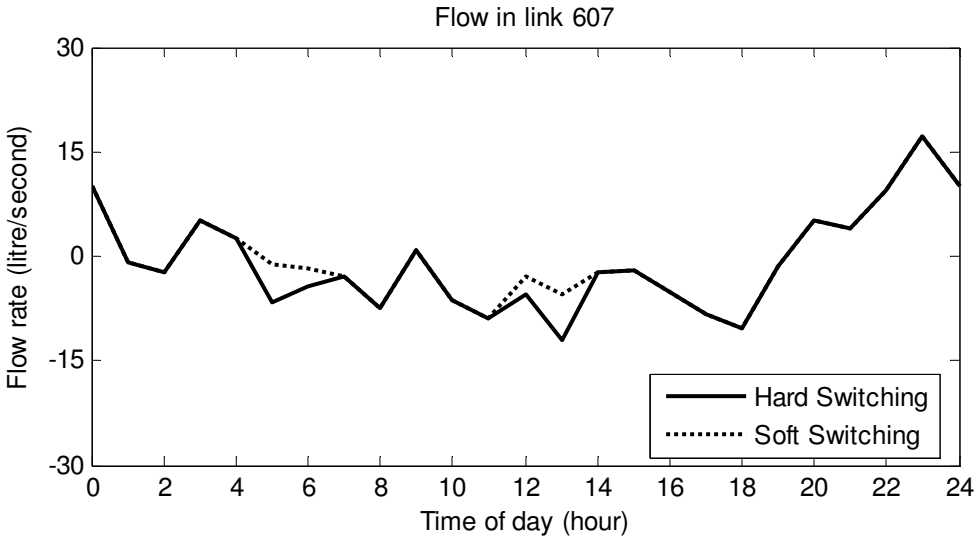


Figure 6.16 Hard switching and soft switching – Flow in link 607:

hard switching (solid line) and soft switching (dotted line)

6.4.3 Infeasible hard switching and fast soft switching

The hard switching is no longer feasible if the initial tank level is 7[m] and the switching time starts at 3h. This situation is illustrated in Figure 6.15 and 6.16 where the simulation breaks down at 4h. However, applying Algorithm 6.2 the fast soft switching can be designed that needs only one step and the operation of MPC is guaranteed to continue. The resulting pump speed and tank level trajectories are shown in Figure 6.17 and Figure 6.18.

The invariant set $\mathbf{X}_{rRf}^{CombinedMPC(1)} = [3.41, 6.95]$ is calculated off-line by the Algorithm 6.2 and the computational time is approximately 18 hours in real time. Having the invariant set $\mathbf{X}_{rRf}^{CombinedMPC(1)}$ calculated, the soft switching is used and the computational time for the online control is approximately 4 hours in real time.

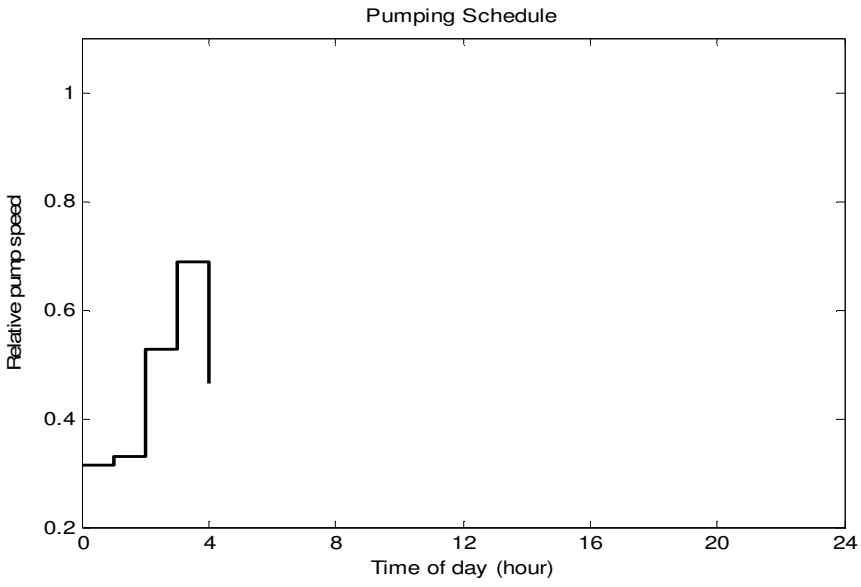


Figure 6.17 Infeasible hard switching – Relative pump speeds

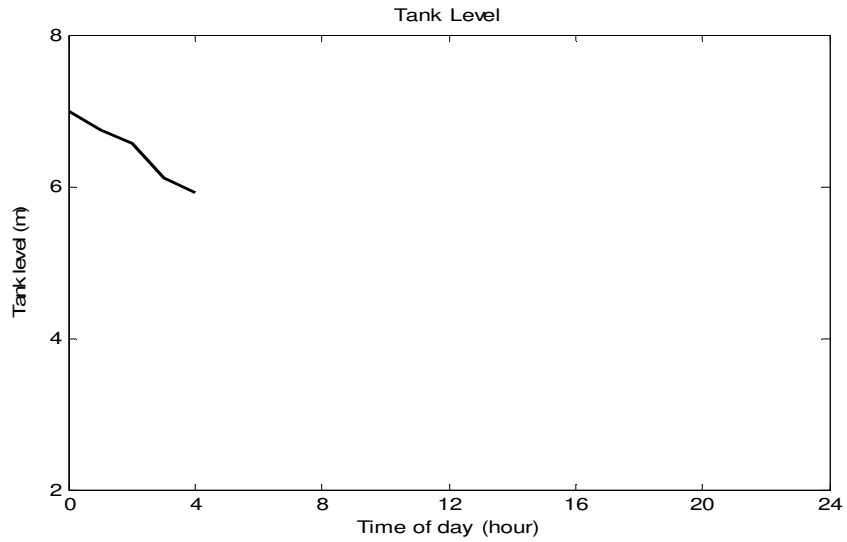


Figure 6.18 Infeasible hard switching – Tank Level

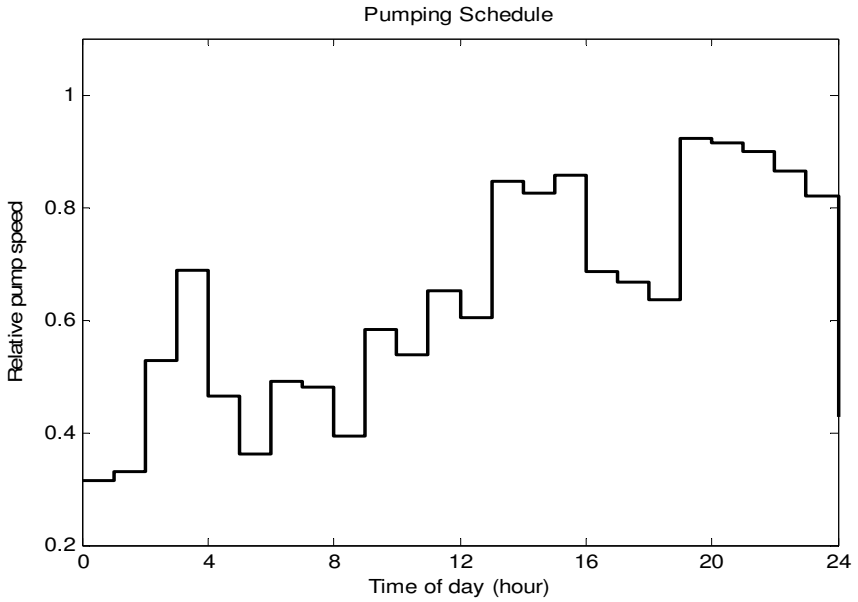


Figure 6.19 Fast soft switching in case of

infeasible hard switching – Relative pump speeds

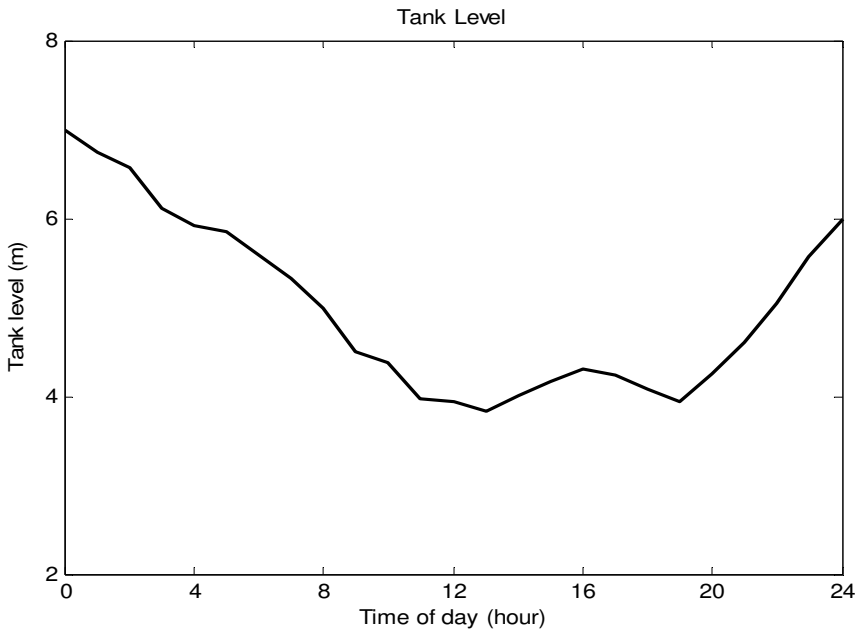


Figure 6.20 Fast soft switching in case of infeasible hard switching – Tank Level

6.4.4 Leaky network

When there is leakage occurred, e.g. across node 4 and node 5 with the emitter coefficients $C_E = 0.1$, the DWDS is then seen as in emergency operational state. Hence, the leakage minimization control strategy needs to be applied to the network. Although the pumping cost control strategy and the excessive pressure minimization control strategy technically still can be applied to the DWDS, the total amount of leakage obtained by the leakage minimization control strategy is smaller than by the other two control strategies.

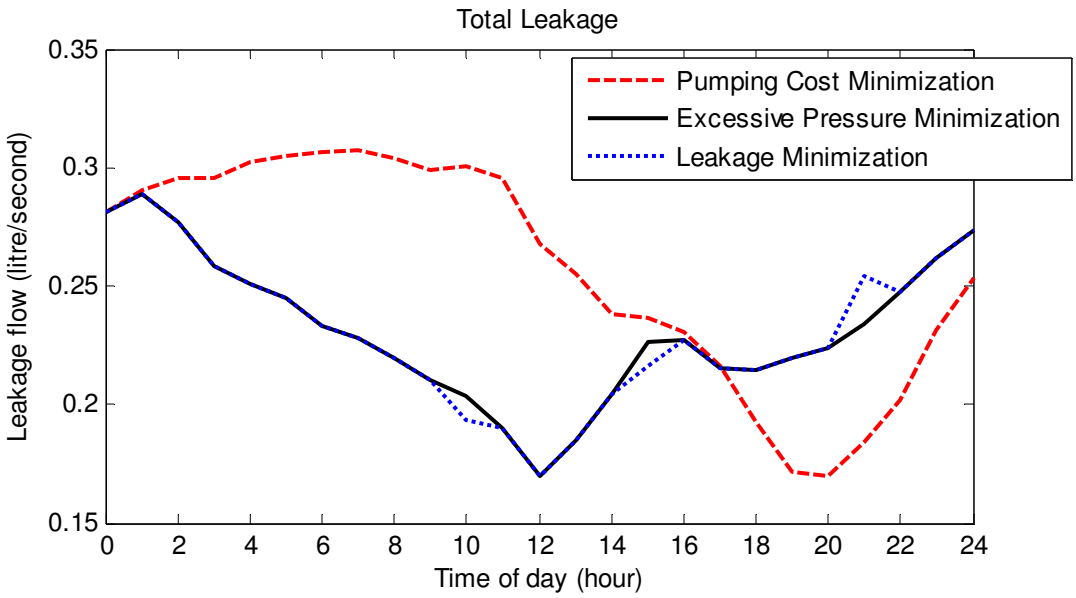


Figure 6.21 Leakage operational scenarios – Total leakage: leakage control (blue dotted line), pressure control (black solid line), and pumping cost control (red dashed line)

The simulation shows that the total amount of leakage by using pumping cost control strategy and excessive pressure minimization control strategy respectively are 0.6949 Ml and 0.6250Ml in a month, whereas it is only 0.6203 ML by the leakage minimization control strategy. In Figure 6.21, it is pointed out that the leakage obtained by the excessive pressure minimization control strategy might be smaller than by the leakage minimization control strategy over certain time steps. However, it does not contradict the fact the leakage control strategy achieves overall least amount of water loss. Simulation results are shown in Figure 6.22 and Figure 6.23 for the pump speed and tank level respectively.

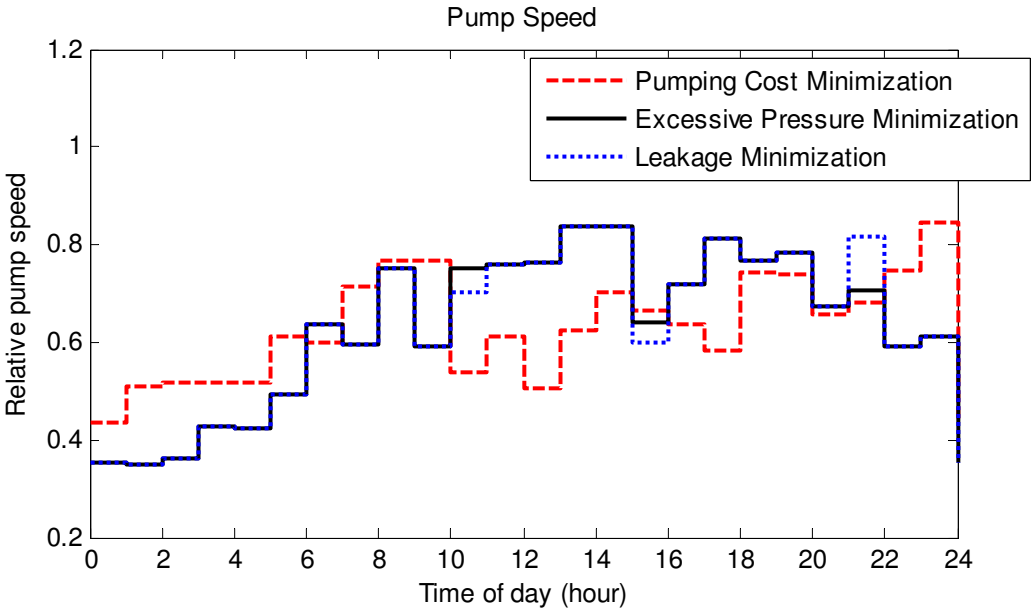


Figure 6.22 Leakage operational scenarios – Relative pump speed: leakage control (blue dotted line), pressure control (black solid line), and pumping cost control (red dashed line)

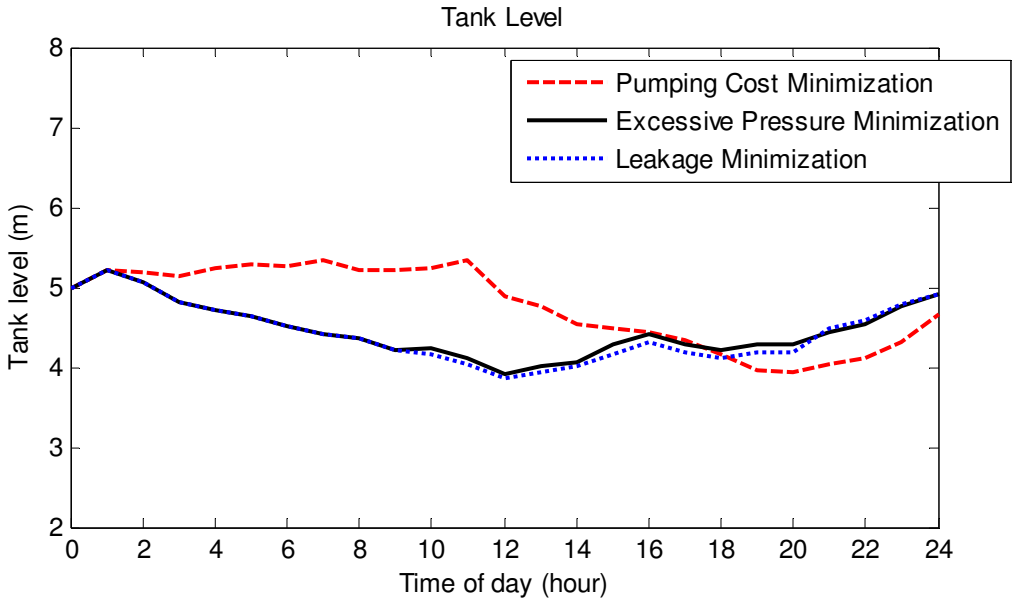


Figure 6.23 Leakage operational scenarios – Tank level: leakage control (blue dotted line), pressure control (black solid line), and pumping cost control (red dashed line)

6.4.5 Full range of operational conditions

An optimal operation of the DWDS under full range of operational conditions can be achieved by applying SSRFMPC which is depicted in Figure 6.24. As the operational conditions vary, the control strategies can be softly switched by SRFMPC.

It is worth noticing that the direction of switching is recommended by the direction of arrows in the Figure 6.24. For example if during the normal operational states there occurs leakage, then the switching direction can go from normal operational states to emergency operational states to minimize the water loss . However if the operational states is being seen in emergency, switching into normal operational states cause the pump become more active and likely to create even bigger leakage. Hence, switching from emergency operational states to normal one is not recommended. Similar reason is applied when switching from disturbed operational states to normal. The recommended directions of switching as shown in Figure 6.24 is important in practice and needs to be taken into consideration in the operational control of DWDS.

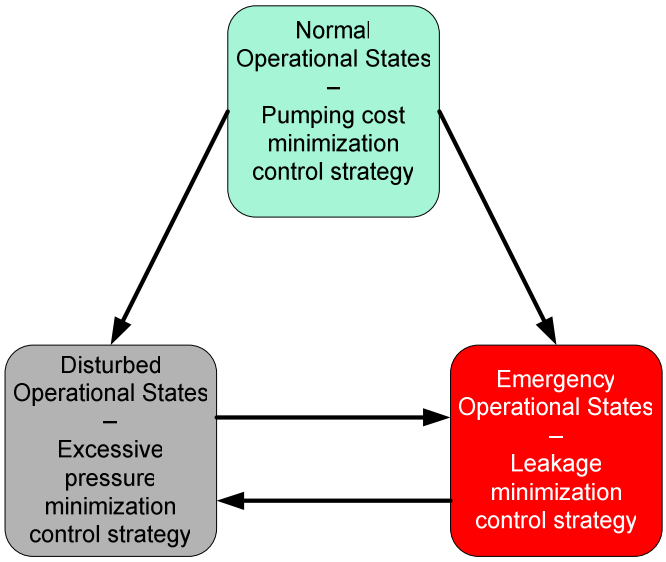


Figure 6.24 Optimal operations by SSRFMPC

It is assumed that from 0h to 6h the DWDS is in *normal operational state* and the *normal* control strategy is firstly applied in saving electrical pumping cost. During 6h-15h, the network suffers from the leakage, the DWDS is then seen to be in the *emergency operational state*. Consequently, the leakage minimization control strategy is applied. After the leakage has been fixed at 15h, the network starts recovering to the normal operational state through the disturbed state. This is achieved by applying first the excessive pressure minimization control strategy for the purpose of leakage prevention until the supervisor commands to minimize the electrical pumping cost at 21h.

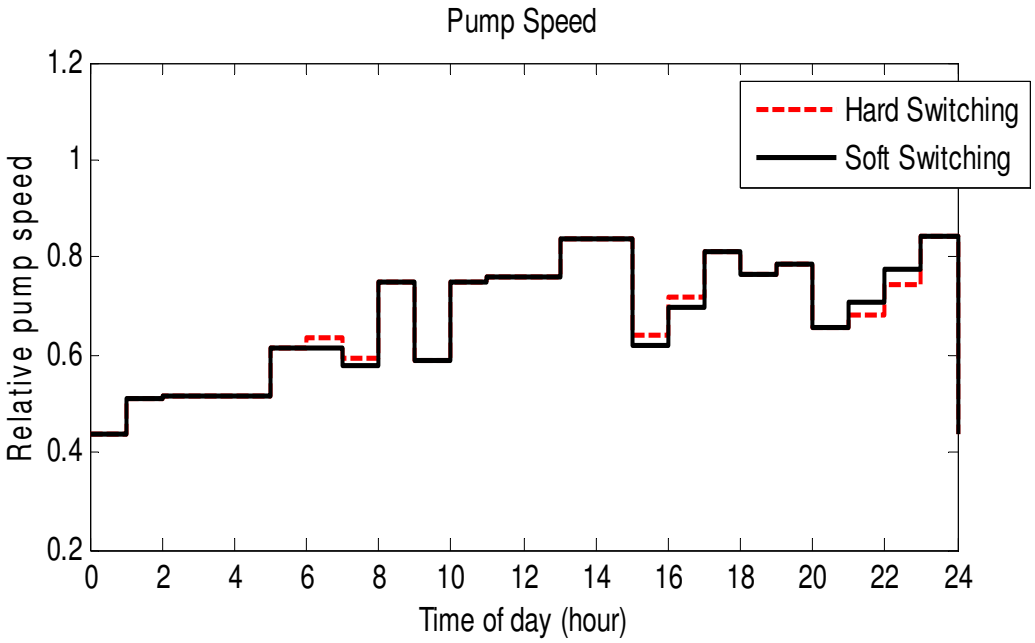


Figure 6.25 Full range of operational conditions – Relative pump speed: hard switching (red dashed line) and soft switching (black solid line)

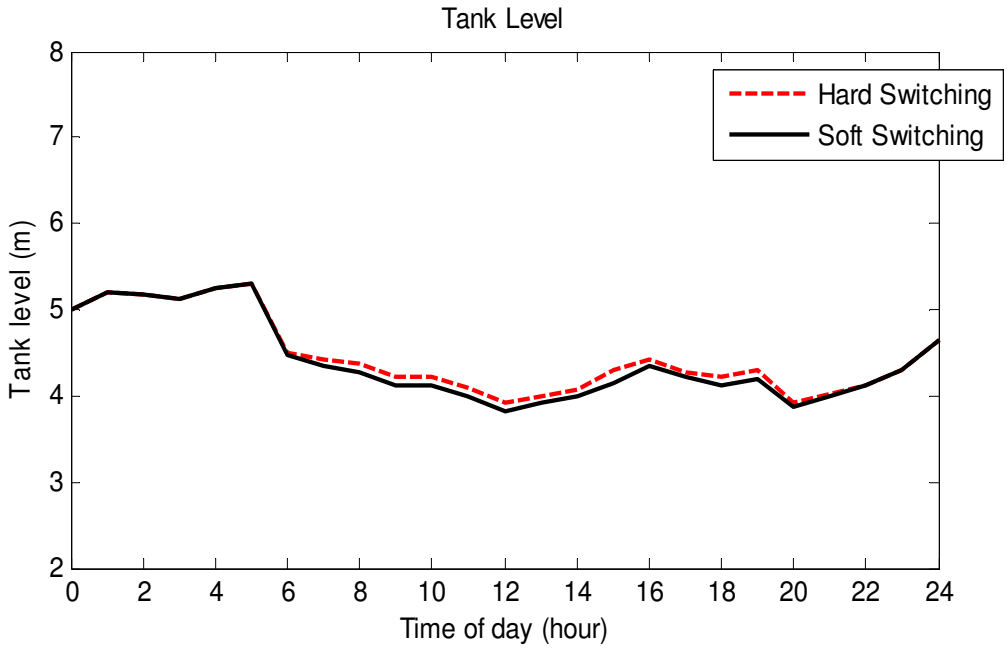


Figure 6.26 Full range of operational conditions – Tank Level: hard switching (red dashed line) and soft switching (black solid line)

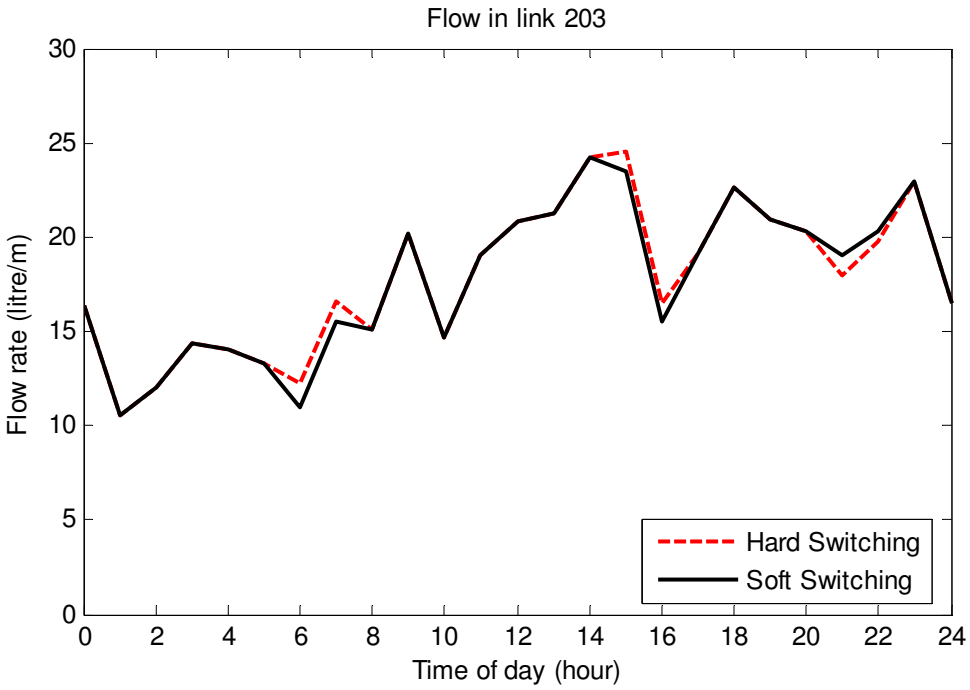


Figure 6.27 Full range of operational conditions – Flow in link 203: hard switching (red dashed line) and soft switching (black solid line)

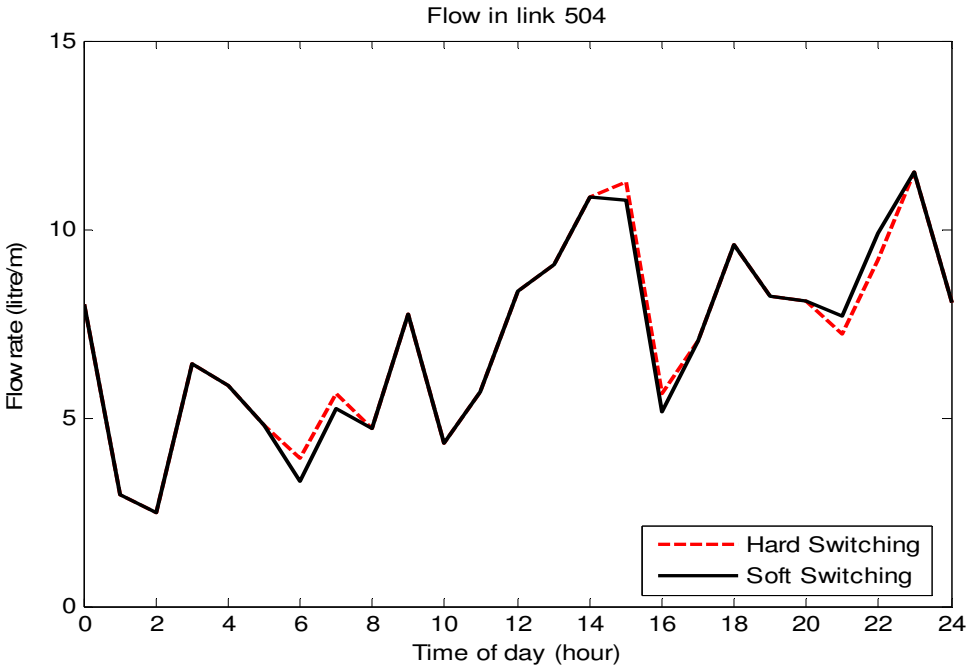


Figure 6.28 Full range of operational conditions – Flow in link 504: hard switching (red dashed line) and soft switching (black solid line)

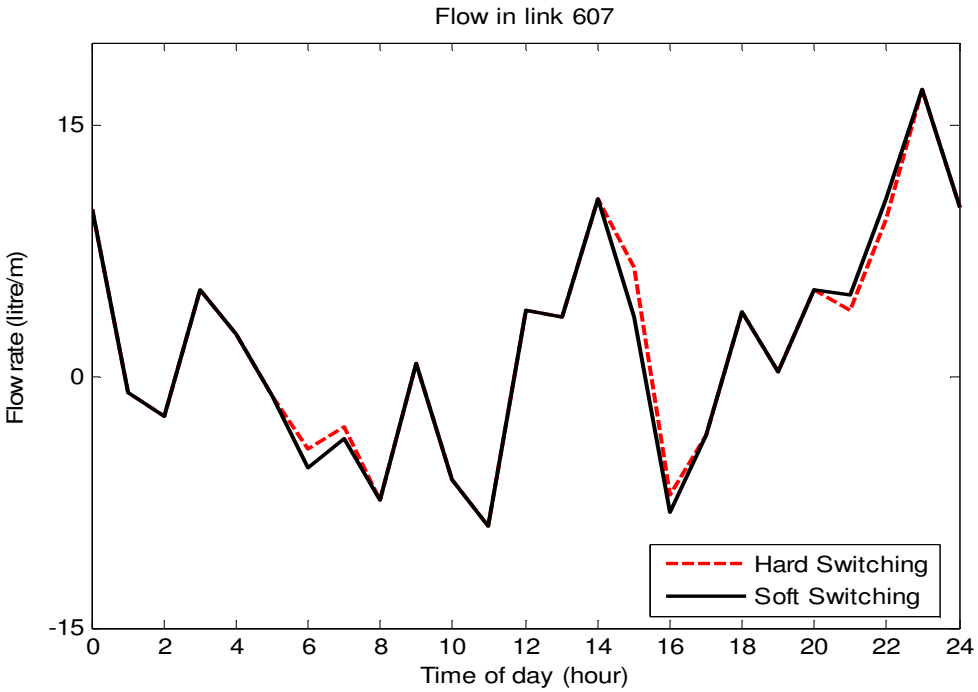


Figure 6.29 Full range of operational conditions – Flow in link 607: hard switching (red dashed line) and soft switching (black solid line)

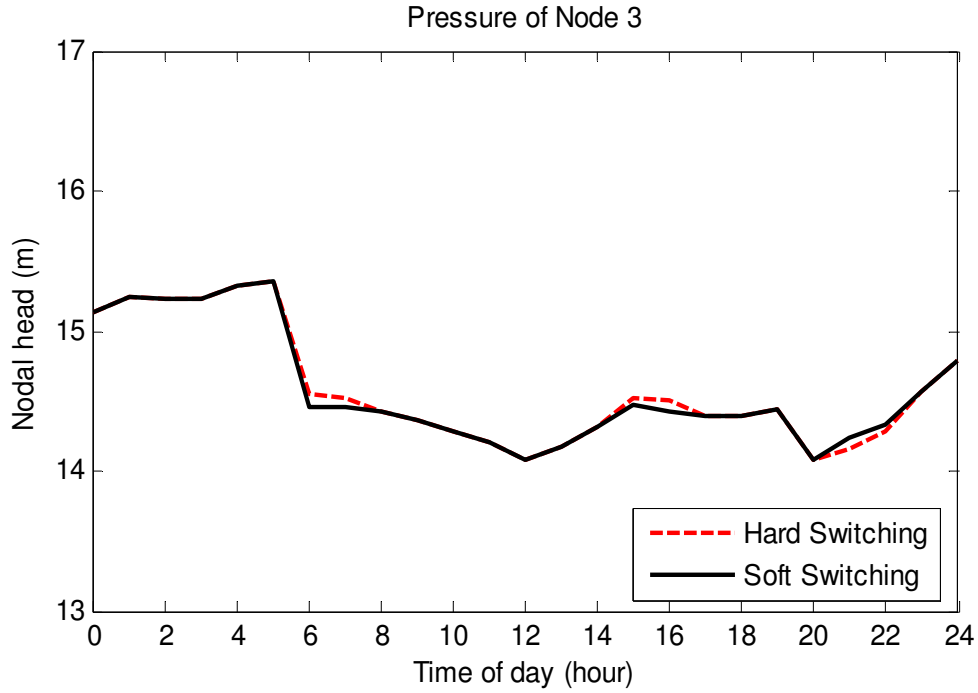


Figure 6.30 Full range of operational conditions – Pressure of Node 3: hard switching (red dashed line) and soft switching (black solid line)

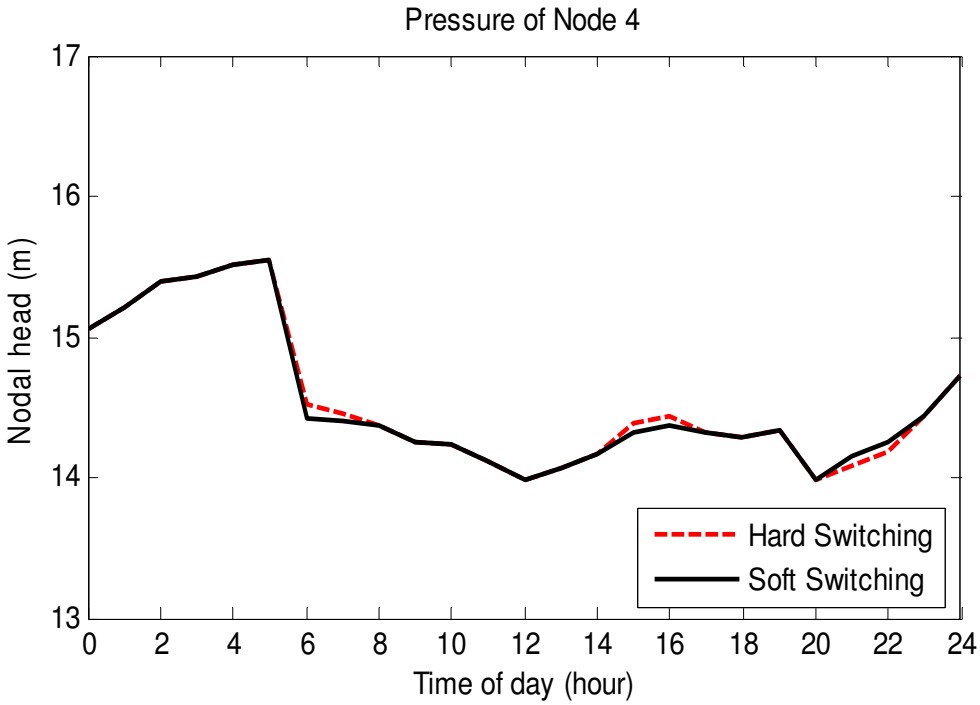


Figure 6.31 Full range of operational conditions – Pressure of Node 4: hard switching (red dashed line) and soft switching (black solid line)

In can be seen that the soft switching achieves smaller peak values and smoother transient trajectory than switching the control strategies in the hard manner on the pipe flows (see Figure 6.25 – 6.27) and nodal pressures (see Figure 6.28 – 6.29). Moreover, with the considered three control strategies that have been applied to the DWDS, a long term sustainable operation of the distribution system is guaranteed by adding the terminal cost on the tank level whish is shown as the second term in objective functions (6.12) – (6.14). Consequently, control the tank level back to around its original status (see Figure 6.24) after a period of 24 hours.

6.5 Summary

The synthesis, analysis, and soft switching mechanism between RFMPCs have been presented in the chapter. The soft switching design is applied to nonlinear network systems in which the impact of disturbance on state is nonlinear and implicit. The new iterative algorithm has also been proposed for the switching mechanism parameter design to achieve fast soft switching in the event of hard switching being not feasible.

Simulation results have been presented to show the feasibility of applying SSRFMPC in the operational control of DWDS. Comparison between soft switching and hard switching in regard to nodal pressures and pipe flows has shown that the hard switching gives more aggressive transients. Although the soft switching gives smoother transients it takes more time to fully engage in to new control strategy. The simulation results also have shown that the hard switching is not always feasible and in such event the fast soft switching has been applied. The SSRFMPC has been applied to the DWDS under full range of operational conditions. In the hierarchical decision support control structure, the SuCL must be involved to make the appropriate decision of when to switch a control strategy. As mentioned in Chapter 5, designing SuCL is nevertheless very complex and beyond the scope the thesis.

Chapter 7

Conclusions and future works

7.1 Conclusions

The thesis has presented the synthesis and analysis of Robustly Feasible MPC (RFMPC) controllers and the soft switching mechanism between them. The Softly Switched RFMPC has been proposed and applied to the DWDS example under full range of operational conditions.

As the DWDS is the main application of the thesis and is used repeatedly in several chapters. All of the modelling components, hydraulic laws, and nodal model of the DWDS have been presented in the Chapter 2 and used as our tool in designing and achieving the simulation results. Three main operational states of the DWDS under wide range of

operational conditions have been distinguished and their corresponding control strategies have been also formulated.

The iterated safety zones based RFMPC (Brdys and Chang 2000) has been briefly presented the in Chapter 3. Some further improvements have been made in the design of the RFMPC. The first improvement was to apply the stepwise robust output prediction instead of least conservative one. In order to do so, two types of scenarios have been declared: (i) stepwise robust output prediction lies inside least conservative one (ii) least conservative robust output prediction lies inside the stepwise one. The technique of using stepwise robust output prediction is only valid if the systems have the property (i). In thesis, the RFMPC has been applied to the operational control of the DWDS example and only simulation results have been used to classify which type of the DWDS example belongs to. The second improvement of this RFMPC is to shorten the input robust feasibility during iterating the safety zones. This has been done by reducing the robust feasibility horizon from 24 hours to 2 hours in the simulation implementation. In term of computational aspect, different types of optimizations solvers i.e. Genetic Algorithm and Sequential Quadratic Programming have been compared. Although the GA was easy to implement, it did not give promising computing efficiency due to the nonlinearities. The computing efficiency has been approximately doubled when applying the SQP. The SQP however required user to supply gradient information which is not an easy task for some circumstances. When applied to the DWDS example, the Hamiltonian based technique has been utilized to calculate the suitable gradients of performance function and the derivatives of constraints. In the implementation of the DWDS, the water network simulator EPANET is used as a ‘real’ water distribution system. The accuracy of the obtained simulation

results heavily depend on the simulator, which may not accurately reflect the real-world situation. Developing a more accurate water network simulator is beyond the scope of the thesis and this is recommended as a possible future research direction. In this thesis, the computing time has been reduced to the affordable and reasonable amount of times i.e. 0.5-2.5 hours to solve one RFMPC optimization task. Moreover, we concentrate on the control side and utilize the currently available optimization methods such as GA and SQP mainly for process control purposes while leaving development of more efficient solvers untouched.

The explicit safety zones based technique to design RFMPC has been proposed in Chapter 4. The technique utilizes the Karush-Kuhn-Tucker optimality condition to approximate the robustly feasible invariant sets of initial states. In the thesis, only the box type approximation in the state space is applied due to the problem complexity. One of the clear advantages of this technique is to allow the RFMPC achieving recursive robust feasibility which cannot be achieved by the technique in Chapter 3. Although it has been pointed out in Chapter 4 that the robustly feasible invariant sets have been computed offline, the computations are very time-consuming. Moreover, there is no guarantee that those invariant sets are exist. Consequently the methods developed in the thesis are only applicable to small scale water network due to the computing difficulty.

The Softly Switched RFMPC and its components have been presented in Chapter 5. The soft switching mechanism is presented and the feasibility of the hard switching is discussed. The functionalities of supervisory control level (SuCL) and its importance with

respect to switching are briefly discussed. A systematic design of such SuCL requires a comprehensive knowledge of plant operational states which is beyond the scope of the thesis

In chapter 6, the soft switching mechanism has been proposed and the SSRFMPC has been successfully achieved. The fast soft switching algorithm has been derived to use in the event of the hard switching is not feasible. The SSRFMPC has been applied to the DWDS example under full range of operational conditions. There possible operational states are described and the switching directions between them have been distinguished. The comparative simulation results have shown that the soft switching achieves smoother transients than the hard switching.

7.2 Future works

Some possible directions and propositions for future research are outlined below.

- **Computing robustly feasible sets for nonlinear predictive controllers**

  Efficient algorithms need to be developed for the computation of robustly feasible invariant sets for predictive controllers of nonlinear systems. The class of systems for which these sets can be computed should also be expanded. It can be expected that efficient computation of such robustly feasible invariant sets can lead to wider applicability of soft switching methods.

- **Implementing a DWDS with multi-tanks and time-varying disturbances**

  With the better computing facilities to compute invariant sets, the DWDS example should be extended to bigger network that consists of multi-tanks/reservoirs. In order to calculate the robustly feasible invariant set of one tank, other tanks will be treated as disturbances and the procedure shown in Chapter 4 will be used. Output constraints e.g. pressure constraints will be added as well. The disturbance will be extended to time varying. The robustly feasible invariant sets $\mathbf{X}_{rRf}(k)$, $k = k_0, ..., N_c$ will be designed over the control period. Then the stationary $\mathbf{X}_{rRf}$ which is suitable for the whole control period is produced by taking the intersection of $\mathbf{X}_{rRf}(k)$ for $k = k_0, ..., N_c$.

- **Integrating Softly Switched RFMPC into supervisory control level**

  Integrating the proposed Softly Switched RFMPC in supervisory hierarchical control of large-scale complex systems needs to be investigated. The supervisory control level needs to be designed in order to assess plant operational states or process performance, and make decisions of control strategy soft switching.

- **Improvement on DWDS simulation under uncertainty**

  EPANET was used in this thesis as the simulator that solved the quantity dynamics. The uncertainties introduced in the simulation case study were demand prediction errors. However, no uncertainty was handled directly in this simulator, e.g. output disturbances. A DWDS simulator needs to be developed to handle uncertainty

quantitatively. Moreover, the water leakage modelling of the simulator does not take the leakage position on the pipe into account but distributing the leakage among certain nodes. A more realistic simulator needs then to be developed by incorporating these two improvements.

Appendix A

Karush - Kuhn - Tucker conditions

Let x^* is a local minimizer of the following optimization problem

$$\text{minimize } f(x)$$

$$\text{subject to: } a_i(x) = 0 \quad \text{for } i = 1, 2, ..., p$$

$$c_j(x) \geq 0 \quad \text{for } j = 1, 2, ..., q$$

satisfying regularity conditions for the constraints that are active at x^*. Then the following hold:

(a) $a_i(x^*) = 0$ for $1 \leq i \leq p$

(b) $c_j(x^*) \geq 0$ for $1 \leq j \leq q$

(c) there exist Lagrange multipliers λ_1^* for $1 \leq i \leq p$ and μ_1^* for $1 \leq j \leq q$ such that

$$\nabla f(x^*) = \sum_{i=1}^{p} \lambda_i^* \nabla a_i(x^*) + \sum_{j=1}^{q} \mu_j^* \nabla c_j(x^*)$$

(d) $\lambda_i^* a_i(x^*) = 0$ for $1 \leq i \leq p$

$\mu_j^* c_j(x^*) = 0$ for $1 \leq j \leq q$

(e) $\mu_j^* \geq 0$ for $1 \leq j \leq q$

Conditions (a) and (b) simply mean that x^* must be a feasible point. The $p + q$ equations in (d) are often referred to as the complementarity KKT conditions. They state that λ_i^* and $a_i(x^*)$ cannot be nonzero simultaneously.

Appendix B

Calling Genetic Algorithm solver

The Genetic Algorithm (GA) is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. The interface of GA is embedded in MATLAB environment and can be called directly from the MATLAB command window.

GA attempts to solve problems of the form:

$$\text{minimize } f(X)$$

$$\text{subject to: } AineqX \le Bineq,\ AeqX = Beq \text{ (linear constraints)}$$

$$C(X) \le 0,\ C(X) = 0 \text{ (nonlinear constraints)}$$

$$LB \le X \le UB$$

Calling syntax:

[X,FVAL,EXITFLAG]=GA(FITNESSFCN,NVARS,A,B,Aeq,Beq,LB,UB,

NONLCON,options)

Input:

FITNESSFCN: Fitness function

NVARS: Number of design variables

Aineq: A matrix for inequality constraints

Bineq:	B vector for inequality constraints
Aeq:	Aeq matrix for equality constraints
Beq:	Beq vector for equality constraints
LB:	Lower bound on X
UB:	Upper bound on X
NONLCON:	nonlinear constraint function

Optional input:

options:	Options structure created with GAOPTIMSET

Output:

X	Solution vector with decision variable values
FVAL	The value of the fitness function FITNESSFCN at the solution X
EXITFLAG	The corresponding exit conditions are

1 Average change in value of the fitness function over options.StallGenLimit generations less than options.TolFun and constraint violation less than options.TolCon.

3 The value of the fitness function did not change in options.StallGenLimit generations and constraint violation less than options.TolCon.

4 Magnitude of step smaller than machine precision and constraint violation less than options.TolCon. This exit condition applies only to nonlinear constraints.

5 Fitness limit reached and constraint violation less than options.TolCon.

0 Maximum number of generations exceeded.

-1 Optimization terminated by the output or plot function.

-2 No feasible point found.

-4 Stall time limit exceeded.

-5 Time limit exceeded.

More detailed information on using GA solver can be found at the following links:

http://www.mathworks.com/help/toolbox/gads/f6187.html (December 20th 2010)

Appendix C

Calling SQP in MATLAB

The Optimization Toolbox is called by the function *fmincon* under the MATLAB environment. The function of Optimization Toolbox is to solve both constrained and unconstrained optimization problems.

fmincon attempts to solve problems of the form:

$$\text{minimize } f(X)$$

$$\text{subject to: } AineqX \le Bineq,\ AeqX = Beq \text{ (linear constraints)}$$

$$C(X) \le 0,\ C(X) = 0 \text{ (nonlinear constraints)}$$

$$LB \le X \le UB$$

Calling syntax:

[X,FVAL,EXITFLAG]=GA(FITNESSFCN,NVARS,A,B,Aeq,Beq,LB,UB,

NONLCON,options)

Input:

FITNESSFCN: Fitness function

NVARS: Number of design variables

Aineq: A matrix for inequality constraints

Bineq:	B vector for inequality constraints
Aeq:	Aeq matrix for equality constraints
Beq:	Beq vector for equality constraints
LB:	Lower bound on X
UB:	Upper bound on X
NONLCON:	nonlinear constraint function

Optional input:

options:	Options structure created with GAOPTIMSET

Output:

X	Solution vector with decision variable values
FVAL	The value of the fitness function FITNESSFCN at the solution X
EXITFLAG	The corresponding exit conditions are

1 Average change in value of the fitness function over options.StallGenLimit generations less than options.TolFun and constraint violation less than options.TolCon.

3 The value of the fitness function did not change in options.StallGenLimit generations and constraint violation less than options.TolCon.

4 Magnitude of step smaller than machine precision and constraint violation less than options.TolCon. This exit condition applies only to nonlinear constraints.

5 Fitness limit reached and constraint violation less than options.TolCon.

0 Maximum number of generations exceeded.

-1 Optimization terminated by the output or plot function.

-2 No feasible point found.

-4 Stall time limit exceeded.

-5 Time limit exceeded.

More detailed information on using GA solver can be found at the following links:

http://www.mathworks.com/help/toolbox/gads/f6187.html (December 20th 2010)

Appendix D

Network file of the DWDS example

```
[TITLE]
Network File of the Example Distribution Network
Under Normal Operating Scenario without Leakage
This network is used in Chapter 3

[JUNCTIONS]
;ID     Elev    Demand      Pattern
2       15      5       1       ;
3       14      5       1       ;
4       12      5       1       ;
5       15      5       1       ;
6       8       30      1       ;

[RESERVOIRS]
;ID     Head    Pattern
1       10              ;

[TANKS]
;ID     Elevation       InitLevel       MinLevel        MaxLevel        Diameter        MinVol
        VolCurve
7       10       5.00   0       10      15      0               ;

[PIPES]
;ID     Node1   Node2   Length  Diameter        Roughness       MinorLoss       Status
203     2       3       1000    400     100     0       Open    ;
205     2       5       1000    400     100     0       Open    ;
304     3       4       1000    400     100     0       Open    ;
504     5       4       1000    300     100     0       Open    ;
406     4       6       1000    500     100     0       Open    ;
607     6       7       1000    500     100     0       Open    ;

[PUMPS]
;ID     Node1   Node2   Parameters
1       1       2       HEAD 1          SPEED 1         PATTERN 2   ;
```

```
[VALVES]
;ID      Node1  Node2  Diameter        Type    Setting MinorLoss

[TAGS]

[DEMANDS]
;Junction        Demand          Pattern Category

[STATUS]
;ID      Status/Setting

[PATTERNS]
;ID      Multipliers
;Demand pattern
1         0.50    0.50    0.40    0.40    0.60    0.80
1         1.00    1.20    1.20    1.30    1.40    1.50
1         1.50    1.40    1.40    1.40    1.50    1.80
1         1.80    1.70    1.60    1.00    0.80    0.60
;Pump Speed
2         0.90    0.90    1.00    1.00    0.90    0.90
2         0.70    1.00    1.20    0.80    1.00    1.00
2         0.71    0.97    0.92    1.16    1.00    1.40
2         1.20    1.50    0.90    0.90    0.50    0.80

[CURVES]
;ID      X-Value         Y-Value
;PUMP:
1         0       200
1         38      120
1         60      0

[CONTROLS]


[RULES]


[ENERGY]
Global Efficiency       75
Global Price            0
Demand Charge           0

[EMITTERS]
;Junction        Coefficient
```

```
[QUALITY]
;Node  InitQual

[SOURCES]
;Node  Type    QualityPattern

[REACTIONS]
;Type   Pipe/Tank       Coefficient


[REACTIONS]
Order Bulk      1
Order Tank      1
Order Wall      1
Global Bulk     0
Global Wall     0
Limiting Potential      0
Roughness Correlation           0

[MIXING]
;Tank   Model

[TIMES]
Duration        24:00
Hydraulic Timestep      1:00
Quality Timestep        0:05
Pattern Timestep        1:00
Pattern Start   0:00
Report Timestep         1:00
Report Start    0:00
Start Clocktime         12 am
Statistic       NONE

[REPORT]
Status  No
Summary         No
Page    0

[OPTIONS]
Units    LPS
Headloss        H-W
Specific Gravity        1
Viscosity       1
Trials  40
Accuracy        0.001
Unbalanced      Continue 10
Pattern 1
Demand Multiplier       1.0
```

```
Emitter Exponent      0.5
QualityNone mg/L
Diffusivity     1
Tolerance       0.01

[COORDINATES]
;Node  X-Coord        Y-Coord
2       -395.57       7713.61
3       450.95 8560.13
4       450.95 7658.23
5       446.99 6878.96
6       1257.91       7658.23
1       -1795.89      7705.70
7       2333.86       7654.27

[VERTICES]
;Link   X-Coord        Y-Coord

[LABELS]
;X-Coord       Y-Coord       Label & Anchor Node

[BACKDROP]
DIMENSIONS      0.00    0.00    10000.00        10000.00
UNITSMeters
FILE
OFFSET          0.00    0.00

[END]



[TITLE]
Network File of the Example Distribution Network
Under Normal Operating Scenario without Leakage
This network is used in Chapter 4 and Chapter 6


[JUNCTIONS]
;ID     Elev    Demand          Pattern
2       14      0               ;
3       14      0               ;
4       12      2               ;
5       14      15      2       ;
6       8       30      1       ;

[RESERVOIRS]
;ID     Head    Pattern
```

```
1       10              ;

[TANKS]
;ID     Elevation       InitLevel       MinLevel        MaxLevel        Diameter        MinVol
        VolCurve
7       10      5.00    0       20      15      0               ;

[PIPES]
;ID     Node1   Node2   Length  Diameter        Roughness       MinorLoss       Status
203     2       3       1000    400     100     0       Open    ;
205     2       5       1000    400     100     0       Open    ;
304     3       4       1000    400     100     0       Open    ;
504     5       4       1000    300     100     0       Open    ;
406     4       6       1000    500     100     0       Open    ;
607     6       7       1000    500     100     0       Open    ;

[PUMPS]
;ID     Node1   Node2   Parameters
1       1       2       HEAD 1          SPEED 1         PATTERN 3   ;

[VALVES]
;ID     Node1   Node2   Diameter        Type    Setting MinorLoss

[TAGS]

[DEMANDS]
;Junction       Demand          Pattern Category

[STATUS]
;ID     Status/Setting

[PATTERNS]
;ID      Multipliers
;Demand pattern
1       0.40    0.40    0.35    0.40    0.45    0.50
1       0.60    0.70    0.70    0.60    0.80    0.60
1       0.65    0.65    0.70    0.70    0.70    0.75
1       0.70    0.60    0.55    0.50    0.40    0.40
;Demand pattern
2       0.60    0.50    0.50    0.55    0.60    0.60
2       0.85    0.80    0.90    0.80    1.00    0.80
2       0.70    0.80    0.65    0.75    0.85    0.80
2       0.85    0.80    0.75    0.55    0.50    0.55
;Pump speed
3       0.35    0.35    0.36    0.43    0.43    0.49
3       0.64    0.60    0.75    0.59    0.75    0.76
3       0.76    0.84    0.84    0.64    0.72    0.81
3       0.77    0.78    0.67    0.71    0.59    0.61
```

```
[CURVES]
;ID     X-Value         Y-Value
;PUMP:
1       0       200
1       38      119.6
1       60      0

[CONTROLS]


[RULES]


[ENERGY]
Global Efficiency       75
Global Price            0
Demand Charge           0

[EMITTERS]
;Junction       Coefficient

[QUALITY]
;Node  InitQual

[SOURCES]
;Node  Type    QualityPattern

[REACTIONS]
;Type   Pipe/Tank       Coefficient


[REACTIONS]
Order Bulk      1
Order Tank      1
Order Wall      1
Global Bulk     0
Global Wall     0
Limiting Potential      0
Roughness Correlation           0

[MIXING]
;Tank   Model

[TIMES]
Duration        24:00
```

```
Hydraulic Timestep    1:00
Quality Timestep      0:05
Pattern Timestep      1:00
Pattern Start    0:00
Report Timestep       1:00
Report Start    0:00
Start Clocktime       12 am
Statistic             NONE

[REPORT]
Status  No
Summary       No
Page    0

[OPTIONS]
Units    LPS
Headloss         H-W
Specific Gravity      1
Viscosity       1
Trials   40
Accuracy         0.001
Unbalanced       Continue 10
Pattern 1
Demand Multiplier     1.0
Emitter Exponent      0.5
QualityNone mg/L
Diffusivity      1
Tolerance        0.01

[COORDINATES]
;Node  X-Coord        Y-Coord
2         -395.57         7713.61
3         450.95 8560.13
4         450.95 7658.23
5         446.99 6878.96
6         1257.91         7658.23
1         -1795.89        7705.70
7         2333.86         7654.27

[VERTICES]
;Link   X-Coord        Y-Coord

[LABELS]
;X-Coord        Y-Coord         Label & Anchor Node

[BACKDROP]
DIMENSIONS       0.00    0.00    10000.00        10000.00
UNITSMeters
```

```
FILE
OFFSET          0.00     0.00

[END]



[TITLE]
Network File of the Example Distribution Network
Under Emergency Operational Conditions with Leakage
This network is used in Chapter 6


[JUNCTIONS]
;ID     Elev    Demand          Pattern
2       14      0               ;
3       14      0               ;
4       12      2               ;
5       14      15      2       ;
6       8       30      1       ;

[RESERVOIRS]
;ID     Head    Pattern
1       10              ;

[TANKS]
;ID     Elevation       InitLevel       MinLevel        MaxLevel        Diameter        MinVol
        VolCurve
7       10        5.00  0       20      15      0               ;

[PIPES]
;ID     Node1   Node2   Length  Diameter        Roughness       MinorLoss       Status
203     2       3       1000    400     100     0       Open    ;
205     2       5       1000    400     100     0       Open    ;
304     3       4       1000    400     100     0       Open    ;
504     5       4       1000    300     100     0       Open    ;
406     4       6       1000    500     100     0       Open    ;
607     6       7       1000    500     100     0       Open    ;

[PUMPS]
;ID     Node1   Node2   Parameters
1       1       2       HEAD 1          SPEED 1         PATTERN 3   ;

[VALVES]
;ID     Node1   Node2   Diameter        Type    Setting MinorLoss

[TAGS]
```

```
[DEMANDS]
;Junction        Demand          Pattern Category

[STATUS]
;ID      Status/Setting

[PATTERNS]
;ID      Multipliers
;Demand pattern
1         0.40    0.40    0.35    0.40    0.45    0.50
1         0.60    0.70    0.70    0.60    0.80    0.60
1         0.65    0.65    0.70    0.70    0.70    0.75
1         0.70    0.60    0.55    0.50    0.40    0.40
;Demand pattern
2         0.60    0.50    0.50    0.55    0.60    0.60
2         0.85    0.80    0.90    0.80    1.00    0.80
2         0.70    0.80    0.65    0.75    0.85    0.80
2         0.85    0.80    0.75    0.55    0.50    0.55
;Pump speed
3         0.35    0.35    0.36    0.43    0.43    0.49
3         0.64    0.60    0.75    0.59    0.75    0.76
3         0.76    0.84    0.84    0.64    0.72    0.81
3         0.77    0.78    0.67    0.71    0.59    0.61

[CURVES]
;ID      X-Value         Y-Value
;PUMP:
1         0       200
1         38      119.6
1         60      0

[CONTROLS]


[RULES]


[ENERGY]
Global Efficiency       75
Global Price            0
Demand Charge           0

[EMITTERS]
;Junction        Coefficient
4         0.1
```

```
5       0.1

[QUALITY]
;Node  InitQual

[SOURCES]
;Node  Type    QualityPattern

[REACTIONS]
;Type   Pipe/Tank       Coefficient


[REACTIONS]
Order Bulk      1
Order Tank      1
Order Wall      1
Global Bulk     0
Global Wall     0
Limiting Potential      0
Roughness Correlation           0

[MIXING]
;Tank   Model

[TIMES]
Duration        24:00
Hydraulic Timestep      1:00
Quality Timestep        0:05
Pattern Timestep        1:00
Pattern Start   0:00
Report Timestep         1:00
Report Start    0:00
Start Clocktime         12 am
Statistic       NONE

[REPORT]
Status  No
Summary         No
Page    0

[OPTIONS]
Units   LPS
Headloss        H-W
Specific Gravity        1
Viscosity       1
Trials  40
Accuracy        0.001
Unbalanced      Continue 10
```

```
Pattern 1
Demand Multiplier     1.0
Emitter Exponent      0.5
QualityNone mg/L
Diffusivity     1
Tolerance       0.01

[COORDINATES]
;Node  X-Coord        Y-Coord
2       -395.57       7713.61
3       450.95 8560.13
4       450.95 7658.23
5       446.99 6878.96
6       1257.91       7658.23
1       -1795.89      7705.70
7       2333.86       7654.27

[VERTICES]
;Link   X-Coord        Y-Coord

[LABELS]
;X-Coord       Y-Coord       Label & Anchor Node

[BACKDROP]
DIMENSIONS      0.00    0.00    10000.00        10000.00
UNITSMeters
FILE
OFFSET        0.00    0.00

[END]
```

Bibliography

Antoniou A. and Lu W.S, (2007) "*Practical Optimization: Algorithms and Engineering Applications*", Springer

Behrang, M., Navid, M., Farhang, J., "A hybrid GA–SQP optimization technique for determination of kinetic parameters of hydrogenation reactions," *Comp. and Chem. Eng*, vol.32, pp.1447–1455, 2008.

Bemporad A., Mosca E., Fulfilling hard constraints in uncertain linear systems by reference managing', in "Automatica", April, 1998.

Bemporad, A. and Morari, M. (1999a). "Robust model predictive control: A Survey". In Garulli, A., Tesi, A. and Vicino, A. (Eds.), *Robustness in Identification and Control*, *Lecture Notes in Control and Information Sciences*, vol. 245, pp. 207-226. Springer-Verlag.

Bentley Systems. (2003). *WaterCAD: Water Distribution System Modeling Software. 6.* Watertown, CT: Haestad Methods Solution Center.

Blanchini, F. (1999). "Set Invariance in control". *Automatica*, **35**(11), 1747–1767.

Blanchini, F. (1994). "Ultimate boundedness control for discrete-time uncertain system via set-induced Lyapunov functions". *IEEE Transactions on Automatic Control*, **39**(2), 428–433.

Brdys, M. A. and P. Tatjewski (2005). *Iterative Algorithms for Multilayer Optimizing Control*, Imperial College Press, UK

Brdys, M. A., Grochowski, M., Chotkowski, W., K. Duzinkiewicz (2002). Design of control structure for integrated wastewater treatment plant- sewer systems. I International Conference on Technology, Automation and Control of Wastewater and Drinking Water Systems-TiASWiK'02, Gdansk - Sobieszewo, June 19 - 21, 2002–Poland- plenary paper.

Brdys, M.A and Ulanicki, B. (1994). *Operational Control of Water Systems: Structures, Algorithms and Applications*. London: Prentice Hall International (UK) Ltd., 1994.

Brdys, M.A. and Chang, T. (2002). "Robust model predictive control under output constraints". In *Proceedings of the 15th Triennial IFAC World Congress,* Barcelona, Spain, July 2002.

Brdys, M.A., Arnold, E., Puta, H., Chen, K. Hopfgarten, S. (1999a) "Integration of quantity and quality issues in operational control of water systems. Part I: Modelling and integrated operational control". (Research Report / Paper Draft Provided by the Authors)

Brdys, M.A., Arnold, E., Puta, H., Chen, K., Hopfgarten, S. (1999b) "Integration of quantity and quality issues in operational control of water systems. Part II: Methods, solvers and case-study". (Research Report / Paper Draft Provided by the Authors)

Brdys, M.A., Malinowski, K., -editors. (1994). Computer Aided Control System Design. World Scientific Publishing, 1994.

Brdys, M.A., Tran, V.N. (2011) Safety zones based robustly feasible model predictive control for nonlinear network systems. In *Proceedings of 18th IFAC World Congress,* Milan, Italy, August 2011. (accepted)

Brdys, MA and Grochowski, M and Gminski, T and Konarczak, K and Drewa, M (2007) *Hierarchical predictive control of integrated wastewater treatment systems.* Control Engineering Practice . ISSN 0967-0661

Camacho, E.F. and Bordons, C. (1999). *Model Predictive Control*. London: Springer-Verlag, 1999.

Cannon, M., Deshmukh, V. and Kouvaritakis, B. (2002). "Nonlinear model predictive control with polytopic invariant sets". In *Proceedings of the 15th Triennial IFAC World Congress,* Barcelona, Spain, July 2002.

Casavola, A., Giannelli, M. and Mosca, E. (2000). "Min-max predictive control strategies for input-saturated polytopic uncertain systems". *Automatica*, **36**(1), 125–133.

Chang, T. (2003). *Robust Model Predictive Control of Water Quality in Drinking Water Distribution Systems*. PhD Thesis. Department of Electronic, Electrical and Computer Engineering, University of Birmingham, United Kingdom, 2003.

Chen, K. (1997). Set Membership Estimation of State and Parameters and Operational Control of Integrated Quantity and Quality Models of Water Supply and Distribution Systems. PhD Thesis. Department of Electronic, Electrical and Computer Engineering, University of Birmingham, United Kingdom, 1997.

Chisci, L., Falugi, P. and Zappa, G. (2001). "Predictive control for constrained systems with polytopic uncertainty". In *Proceedings of American Control Conference*, Arlington, VA, June 2001.

Chisci, L., Rossiter, J.A. and Zappa, G. (2001). "Systems with persistent disturbances: predictive control with restricted constraints". *Automatica*, **37**(7), 1019–1028.

Coulbeck, B. (1988). "A review of methodologies for modelling and control of water supply". *Computer Applications in Water Supply* vol. 2 (Coulbeck, B. and Orr, C.H. Eds.). Research Studies Press Ltd, John Wiley, Chichester.

Cutler, C.R. and Ramaker, B.L. (1980). "Dynamic matrix control - a computer control algorithm". In *Proceedings of Joint Automatic Control Conference*, San Francisco, CA, June 1980.

Dam, A.A and Nieuwenhuis, J.W. (1995). “A linear programming algorithm for invariant polyhedral sets of discrete-time linear systems”. *Systems and Control Letters*, **25**(5), 337–341.

Dorea, C. and Hennet, J.C. (1999). “(A, B) - invariant polyhedral sets of linear discrete-time systems”. *Journal of Optimization Theory and Application*, **103**(3), 521–542.

Findeisen, W., Bailey, F. N., Brdys, M, Malinowski, K., Tatjewski, P., Wozniak, A., (1980). Control and coordination in hierarchical systems. A Wiley International Institute for Applied Systems Analysis, 1980.

Fisher, M. E. and L. S. Jennings (1992). "Discrete-time optimal control problems with general constraints." *ACM Transactions on Mathematical software* **18**(4): 13.
Fletcher, R. (1987). *Practical Methods of Optimization*, Second Edition. Chichester: John Wiley, 1987.

G. Ewald, W. Kurek, M.A. Brdys. Grid implementation of parallel multi-objective genetic algorithm for optimized allocation of chlorination stations in drinking water distribution systems: Chojnice case study. *IEEE Trans. on System, Man and Cybernetics – Part C: Applications and Reviews*, Vol. 38, No. 4, 2008, pp. 497 - 509.

Garcia, C.E., Prett, D.M. and Morari, M. (1989). “Model predictive control: theory and practice - a survey”. *Automatica*, **25**(3), 335–348.

Germanopoulos, G., (1985). “A technical note on the inclusion of pressure dependent demand and leakage terms in water supply network models.” Civil Engineering Systems, 2(September), 171–179.

Gilbert, E.G. and Tan, K.T. (1991). “Linear systems with state and control constraints: the theory and application of maximal output admissible sets”. *IEEE Transactions on Automatic Control*, **36**(9), 1008–1020.

Gill, P.E., W. Murray, and M.H.Wright, Practical Optimization, London, Academic Press, 1981.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Kluwer Academic Publishers, Boston, MA, USA

Grieder, P., Parrilo, P.A. and Morari, M. (2003). "Robust receding horizon control – analysis and synthesis". In *Proceedings of the 42th IEEE Conference on Decision and Control*, Hawaii, December 2003.

Griva I., Nash S.G., Sofer A., (2009) "*Linear and nonlinear optimization*", Society for Industrial Mathematics; 2nd edition

Grochowski, M. (2003). *Intelligent control of integrated wastewater treatment system under full range of operating conditions*. PhD thesis, Faculty of Electrical and Control Engineering, Gdansk University of Technology, Gdansk, Poland, 2003.

Grochowski, M., Brdys, M.A. Gmiński, T. (2004). Intelligent control structure for control of integrated wastewater systems. IFAC 10th Symposium Large Scale Systems: Theory and Applications. July 26-28 2004, Osaka – Japan.

Grochowski, M., Brdys, M.A. Gmiński, T., Deinrych, P. (2004). Softly switched model predictive control for control of integrated wastewater treatment systems at medium time scale. IFAC 10th Symposium Large Scale Systems: Theory and Applications. July 26-28 2004, Osaka – Japan

Haestad Methods. (2002). *Computer Applications in Hydraulic Engineering*, Fifth Edition. London: Haestad Press, 2002.

Hock, W. and K. Schittkowski, "A Comparative Performance Evaluation of 27 Nonlinear Programming Codes," Computing, Vol. 30, p. 335, 1983

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, USA

Jingsong Wang and Mietek A. Brdys. (2006a). "Supervised robustly feasible soft switching model predictive control with bounded disturbances". In *Proceedings of the 6th IEEE Biennial World Congress on Intelligent Control and Automation* (*WCICA'06*), Dalian, China, June 2006.

Jingsong Wang, Michal Grochowski and Mietek A. Brdys. (2005). "Analysis and design of softly switched model predictive control". In *Proceedings of the 16th IFAC Triennial World Congress*, Prague, Czech Republic, July 2005.

Jonkergouw P. M. R., Khu S. T. and Savic D. A. (2004): Chlorine: An indicator for intentional chemical and biological contamination of a water distribution, Proc. AutMoNet Conf. 2004, Vienna, Austria.

Jowitt, P.W. and Xu, C. (1990). "Optimal valve control in water distribution networks". *Journal of water Resource Planning and Management*, **116**(4), 455-472.

K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering,* vol. 186, 2000, pp. 311-338.

K.L. Teo, C.J. Goh, and K.H. Wong, *A* (1991) *Unified Computational Approach to Optimal Control Problems,* John Wiley, New York, USA

Kerrigan, E.C. (2000). *Robust constraints satisfaction: invariant sets and predictive control*. PhD thesis, Department of Engineering, University of Cambridge, 2000.

Kerrigan, E.C. and Maciejowski, J.M. (2001). "Robust feasibility in model predictive control: necessary and sufficient conditions". In *Proceedings of the 40th IEEE Conference on Decision and Control*, Florida, December 2001.

Kolmanovsky, I. and Gilbert, E.G. (1998). "Theory and computation of disturbance invariant sets for discrete-time linear systems". Mathematical Problems in Engineering, 4(4), 317-367.

Kothare, M.V., Balakrishnan, V. and Morari, M. (1996). "Robust constrained model predictive control using linear matrix inequalities". *Automatica*, **32**(10), 1361–1379.

Kouvaritakis, B. and Cannon, M. (2001). *Nonlinear Predictive Control: Theory and Practice*. London: IEE Publishing, 2001.

Kwon, W.H. and Han, S. (2005). *Receding Horizon Control*. London: Springer-Verlag, 2005.

Lambert, A., Myers, S. and Trow, S. (1998). *Managing water leakage, economic and technical issues*. London: Financial Times Energy Ltd.

Lee, E.B. and Markus, L. (1968). *Foundations of Optimal Control Theory*. New York: John Wiley, 1968.

Lee, J.H. and Cooley, B.L. (2000). "Min-max predictive control techniques for a linear state-space system with a bounded set of input matrices". *Automatica*, **36**(3), 463–473.

Lee, J.H. and Yu, Z. (1997). "Worst-case formulations of model predictive control for systems with bounded parameters". *Automatica*, **33**(5), 763–781.

Lewis, F. L. and Syrmos, V.L. (1995) "*Optimal control*", Wiley, New York, USA
Maciejowski, J.M. (2002). *Predictive control with constraints*. London: Prentice Hall, 2002.

May, J. (1994). "Pressure-depended leakage". *World Water and Environmental Engineering Magazine*, **17**(5/12).

Mayne, D.Q., Rawlings, J.B., Rao, C.V. and Scokaert, P.O.M. (2000). “Constrained model predictive control: stability and optimality”. *Automatica*, **36**(6), 789–814.

Mietek A. Brdys and Jingsong Wang. (2005). “Invariant set-based robust softly switched model predictive control”. In *Proceedings of the 16th IFAC Triennial World Congress*, Prague, Czech Republic, July 2005.

Mosca, E. (1995). *Optimal, Predictive, and Adaptive Control*. New York: Prentice Hall, 1995.

Muske, K.R. and Rawlings, J.B. (1993). “Model predictive control with linear models”. *A.I.CH.E Journal*, **39**(2), 262–297.

Ogata, K. (2005). Modern control engineering, Prince Hall.

Ostfeld A. and Salomons E., "Optimal layout of early warning detection stations for water distribution systems security," *Journal of Water Resources Planning and Management-Asce,* vol. 130, 2004, pp. 377-385.

Pearce, H. (1991). “The rising price of diminishing returns”. *Water Services*, **95**(1149), 14-16.

Powell, M.J.D., "Variable Metric Methods for Constrained Optimization," Mathematical Programming: The State of the Art, (A. Bachem, M. Grotschel and B. Korte, eds.) Springer Verlag, pp 288-311, 1983.

Primbs, J.A and Nevistic, V. (2000b). “Feasibility and stability of constrained finite receding horizon control”. *Automatica*, **36**(7), 965-971.

Propoi, A.I. (1963). “Use of linear programming methods for synthesizing sampled data automatic systems”. *Automation and Remote Control*, **24**(7), 837–844.

Pudar, R.S. and Liggett, J.A. (1992). "Leaks in pipe networks". *Journal of Hydraulic Engineering*, **118**(7), 1031-1045.

Qin, S.J. and Badgwell, T.A. (2003). "A survey of industrial model predictive control technology". *Control Engineering Practice*, **11**(7), 733-764.

Rawlings, J. (2000). "Tutorial overview of model predictive control". *IEEE Control Systems Magazine*, **20**(3), 38–52.

Richalet, J., Rault, A., Testud, J.L. and Papon, J. (1978). "Model predictive heuristic control: applications to industrial processes". *Automatica*, **14**(5), 413–428.

Rossiter, J.A. (2003). *Model-based predictive control: a practical approach*. Florida: CRC press, 2003.

Rossman, L.A. (2000). *EPANET 2.0 for Windows*. Water Supply and Water Resourses Devision, National Risk Management Research Laboratory, Cincinnati, OH.

Sage, A. P. (1968) "*Optimum systems control*", Prentice Hall, Electrical Engineering Series, England

Santis, D.E. (1998). "On invariant sets for constrained discrete time linear systems with disturbances and parametric uncertainties". *Automatica*, **33**(2), 2033-2039.

Savic, D. A. and G. A. Walters. (1997). "Genetic algorithms for least-cost design of water distribution networks." *Journal of Water Resources Planning and Management-Asce*, 123(2), 67-77.

Scokaert, P.O. and Mayne, D.Q. (1998). "Min–max feedback model predictive control for constrained linear systems". *IEEE Transactions on Automatic Control*, **43**(8), 1136–1142.

Scokaert, P.O.M. and Rawlings, J.B. (1999). "Feasibility issues in linear model predictive control". *A.I.CH.E Journal*, **45**(8), 1649–1659.

Sterling, M. and Bargiela, A. (1984). "Leakage reduction by optimised control of valves in water networks". *Transactions of the Institute of Measurement and Control*, **6**(6), 293-298.

Tolson, B. A., H. R. Maier, A. R. Simpson and B. J. Lence. (2004). "Genetic algorithms for reliability-based optimization of water distribution systems." *Journal of Water Resources Planning and Management-Asce*, 130(1), 63-72.

Tran, V.N and Brdys, M.A (2009). "Optimizing Control by Robustly Feasible Model Predictive Control and Application to Drinking Water Distribution Systems". In *Proceedings of 19th International Conference on Artificial Neural Networks (ICANN)*, Limassol, Cyprus, September 2009.

Tran, V.N and Brdys, M.A (2010). "Robustly Feasible Optimizing Control of Network Systems under Uncertainty and Application to Drinking Water Distribution Systems". In *Proceedings of 12th IFAC symposium on Large Scale Complex System: Theory and Application*, Lille, France, July 2010.

Tran, V.N and Brdys, M.A (2011). "Optimizing Control by Robustly Feasible Model Predictive Control and Application to Drinking Water Distribution Systems" Journal of Artificial Intelligence and Soft Computing Research, 2011, Vol. 1, No. 1, pp.43-57.

Vairavamoorthy, K. and Lumbers, J. (1998). "Leakage reduction in water distribution systems: optimal valve control". *Journal of Hydraulic Engineering*, **123**(11), 1146-1154.

Wang, J. (2006). *Softly switched model predictive control : generic development and application to water supply and distribution systems*. PhD Thesis. Department of Electronic, Electrical and Computer Engineering, University of Birmingham, United Kingdom, 2003.

Water Authorities Association. (1985). "Leakage control policy and practice". *National Water Council Standing Technical Committee Report*, No. 26, NWC, London.

WRc-UK Water Industry. (1994). *Managing Leakage: Managing Water Pressure – Report G*. Wiltshire: WRc Publications, 1994.

Wu, Z. Y. and A. R. Simpson. (2001). "Competent genetic-evolutionary optimization of water distribution systems." *Journal of Computing in Civil Engineering*, 15(2), 89-101.