SLOVAK UNIVERSITY OF TECHNOLOGY
IN BRATISLAVA

FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-184466-126940

Predictive Control of Dynamic Systems

BACHELOR THESIS

2026 **Michal Zatkalík**

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-184466-126940

Predictive Control of Dynamic Systems

BACHELOR THESIS

Study programme:	Process Control
Study field:	Cybernetics
Training workspace:	Institute of Information Engineering, Automation, and Mathematics
Thesis supervisor:	Ing. Marián Gall, PhD.

2026 **Michal Zatkalík**

Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation and Mathematics

Faculty of Chemical and Food Technology
Academic year: 2025/2026
Reg. No.: FCHPT-184466-126940

BACHELOR THESIS TOPIC

Student:	**Michal Zatkalík**
Student's ID:	126940
Study programme:	process control
Study field:	Cybernetics
Thesis supervisor:	Ing. Marián Gall, PhD.
Head of department:	prof. Ing. Miroslav Fikar, DrSc.
Workplace:	Institute of Information Engineering, Automation, and Mathematics (FCHPT)

Topic: **Predictive Control of Dynamic Systems**

Language of thesis: English

Specification of Assignment:

The objective of this thesis is to design and validate a controller based on the Model Predictive Control (MPC) approach for a selected dynamic system, for example a thermal process or a (bio)chemical reactor. Model predictive control is an advanced control strategy that uses a mathematical model of the process to predict its future behavior and optimize control actions while considering operational constraints. Within the project, a dynamic model of the system will be developed, on the basis of which a predictive controller will be designed. Its performance will be compared with a conventional (PID) controller through simulation experiments.

Tasks:
1. Study of model predictive control principles
2. Development of a model of the selected dynamic system
3. Design of a controller for the selected dynamic system
4. Evaluation of the controller performance

Selected bibliography:

1. RAWLINGS, James Blake – MAYNE, David Q. – DIEHL, Moritz. Model Predictive Control: Theory, Computation, and Design. 2nd ed. Nob Hill Publishing, 2020. 623 p. ISBN 978-0-9759377-5-4

Acknowledgment

I would like to thank my supervisor, Prof. Joaquim Blesa, PhD. and my co-supervisor, Prof. Vicenç Puig, PhD. from Polytechnic University of Catalonia (UPC), whose guidance, valuable inputs and regular meetings helped me complete this thesis. Additionally, I would like to thank my supervisor, Ing. Marián Gall, PhD. and co-supervisor, doc. Ing. Radoslav Paulen, PhD. from my home university (Slovak University of Technology in Bratislava) for regular check-ups on my progress, their ideas and suggestions and their great help with writing this thesis, by giving me pointers on what to change and what I missed.

Abstract

This bachelor's thesis is concerned with the design of an Economic Model Predictive Controller (Economic MPC) used on a dynamic model of the Richmond water distribution network. The thesis explains how an MPC works and how these concepts are applied to a non-linear system like a water distribution network (WDN). Additionally, it shows how a WDN is modelled for the controller to use. The main aim of this work is to compare the performance of non-linear MPC and linear parameter varying MPC, to determine which controller is better suited for problems like WDN control. Based on the achieved results, this work aims to find out if using LPV MPC offers any benefits compared to non-linear MPC. The results show that LPV MPC sufficiently replicates the non-linear MPC with a 0.6607% increase in operational costs but a lower simulation time of 10 seconds using a QP solver, compared to 125 seconds for the non-linear MPC.

Keywords: Model Predictive Control, Economic Model Predictive Control, Water Distribution Network, linear parameter varying

Abstrakt

Táto bakalárska práca sa zaoberá návrhom ekonomického regulátora prediktívneho riadenia založeného na modeli (Economic MPC) a jeho použitím na dynamickom modeli rozvodnej siete vody mesta Richmond. Práca vysvetľuje, ako MPC funguje a ako sa tieto koncepty aplikujú na nelineárny systém, ako je rozvodná sieť vody. Okrem toho ukazuje, ako je takáto sieť modelovaná pre použitie v regulátore. Hlavným cieľom tejto práce je porovnať výkonnosť nelineárneho MPC a LPV MPC, s cieľom určiť, ktorý regulátor je vhodnejší pre problémy ako riadenie rozvodných sietí. Na základe dosiahnutých výsledkov táto práca rozoberá, či používanie LPV MPC ponúka nejaké výhody v porovnaní s nelineárnym MPC. Je ukázané, že LPV MPC dosiahlo veľmi podobné trajektórie ako nelineárne MPC, malo o 0,6607 % vyššie ekonomické náklady prevádzky, ale simulácia s QP solverom trvala 10 sekúnd oproti 125 sekundám pre nelineárne MPC.

Kľúčové slová: prediktívne riadenie, ekonomické prediktívne riadenie, rozvodná sieť vody, menenie lineárnych parametrov

Contents

Chapter 1

Introduction

Model Predictive Control (MPC) is based on optimal control. MPC uses a dynamic model to forecast system behaviour and optimise the forecast to compute an optimal control action. It can handle large and complex multi-input and multi-output (MIMO) systems with many constraints, either physical or due to safety or demands. Its optimisation nature can also serve to aid economic and ecological safety performance [7]. This makes it very popular among advanced process control controllers. Its strengths are especially useful in systems like water distribution networks (WDN), where all these strengths are useful and sometimes even essential. In the last few years, important research efforts have been put towards exploring this control method [6].

Water is one of the most essential substances on our planet. Its use can be found in many areas that are commonly encountered, even by an average person. For instance, creating products that we use on a daily basis, like produce from farming, waste disposal and drinking. Water is also involved in less noticeable areas of daily life, like manufacturing and chemical processes and reactions, which still affect the entire modern world [3]. Water, being the basis of life on Earth, is an essential component of the human diet and needs to be replenished constantly. This is why it is important for human settlements to have a steady source of drinking water.

Throughout history, humans have built their settlements near sources of fresh water, mostly rivers, lakes, springs or places where they could dig a well to pull fresh water out of the earth [2]. However, with the populations of these villages and towns growing and also the need for higher standards and more sophisticated approaches for delivering water to the homes of people, larger and automated systems needed to be introduced. A city where one was not able to just use the tap in the kitchen to pour a glass of water and would instead need to walk to a river or a well in the town square to draw a bucket of water is far from the modern standards people are used to. This modern convenience is provided by large and complex water distribution networks, running under the cities we live in, consisting of hundreds of pipes, junctions, and tanks [4].

Water distribution networks need to be safe and stable, always providing a sufficient amount of clean water at the desired pressure. It would not be acceptable to have an outage in an institution like a hospital, where human lives could be put at stake because of an error in the control of the network. It is necessary to have these networks running all throughout the day, meaning their operation needs to be optimised to use as little electrical power as possible, lowering the operational costs of the network, as well as possible wear of the pumps themselves due to the constant use.

The aim of this thesis is to implement an Economic MPC for the Richmond WDN. Due to the nature of the WDN systems and their control which is constraint-driven and economically optimized without fixed setpoints a PID baseline would not address the same problem formulation. Instead, two different MPCs are compared - non-linear MPC and linear parameter-varying (LPV) MPC. These controllers are compared in terms of constraint satisfaction, operating cost, and computation time.

Chapter 2

Theory

2.1 Model Predictive Control

Model predictive control is a control technique utilising optimisation. This controller uses a dynamic model of the system to predict its behaviour in the future after applying calculated inputs for the finite forecast length. These inputs are optimised by minimising the objective function that is subject to certain constraints.

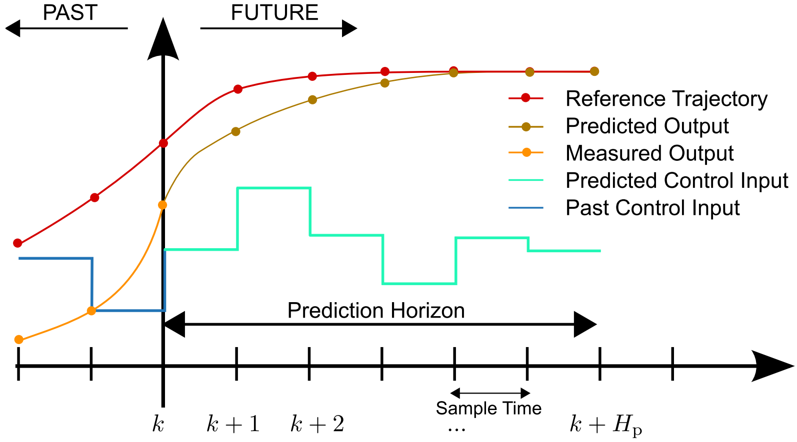

Figure 2.1: Visualisation of model predictive control with measured and predicted variables

A great benefit of MPC is that it can easily control multi-input multi-output systems (MIMO) with constraints. Additionally, this type of controller can be implemented quite easily with the right tools, making it very useful for the control of a large range of systems. However, on the other hand, MPC does have some drawbacks, like the necessity to have a model of the controlled system for calculations. Additionally, the calculations for finding the optimal values of manipulated variables can take a long time due to being expensive compared to time-invariant calculations using simple arithmetic operations in other types of controllers (e.g. PID controller).

Typical cost function for an MPC, which is minimised by an optimisation solver, could look as follows:

$$J = \sum_{i=1}^{N} w_{x_i}(r_i - x_i)^2 + \sum_{j=1}^{M} w_{u_j} \Delta u_j^2, \tag{2.1}$$

where the first summation term penalises the error (difference of the reference r_i and the controlled variable x_i) of the controlled variable by multiplying each squared error by a corresponding weight w_{x_i}. This term causes the controller to reach the reference, because its minimal value is equal to zero (the controlled variable is equal to the reference). The second summation term penalises large changes of individual inputs Δu_j (once again multiplied by corresponding weights w_{u_j}). This term prevents erratic changes in inputs, which can often lead to overshoots, instability or damage to the real system. The weights here can be used to target more important or problematic variables of a system in a very direct approach.

Equation (2.1) depicts the stage cost. To fully utilise the power of predictive control, the cost function is calculated for the following time steps as well, hence the name predictive. How far into the future the prediction goes depends on the length of the prediction horizon. The prediction horizon is another variable chosen when creating an MPC. A longer prediction horizon ensures the control will tend to be optimal; however, at the cost of increasing the computation costs. Using equation (2.1) for each step in the prediction horizon to calculate the cost and summing it yields the objective function to be minimised by an optimisation solver. The problem would then look as follows:

$$\min \sum_{k=1}^{H_\mathrm{p}} J_k, \tag{2.2}$$

where H_p is the prediction horizon.

Constraints given to the controller can not only ensure feasibility due to the physical limitations, but they can also be used for safety. For example, input change can be limited in the real world or sudden changes in controlled variables like temperature could lead to damaged equipment due to thermal stresses. The general form of these constraints could look like

$$a_\mathrm{lower} \leq f(a(k)) \leq a_\mathrm{upper}, \tag{2.3}$$

where the value of a function where variable a is the argument is bound between the selected lower and upper limit. Variable a can be any variable of the system, such as an input variable or controlled variable.

Separate but absolutely crucial constraints are the ones calculating values of controlled variables after applying inputs. This is the part where the internal system model comes in. The controller needs to have a model of the system implemented inside itself to be able to calculate these constraints. A simple example of this constraint, the objective function is subject to, can be a linear discrete time-invariant state equation

$$x(k+1) = Ax(k) + Bu(k), \tag{2.4}$$

with this equation being calculated for each step in the prediction horizon. These equations can take on any form to reflect the system accurately [7].

The entire optimisation problem of a standard MPC could therefore look as follows:

$$\begin{aligned} \min \quad & \sum_{k=1}^{H_\mathrm{p}} \left(\sum_{i=1}^{N} w_{x_i} (r_i(k) - x_i(k))^2 + \sum_{j=1}^{M} w_{u_j} \Delta u_j(k)^2 \right) \\ \text{s.t.} \quad & x(k+1) = f_1(x(k), u(k)) \\ & x_\mathrm{lower} \leq f_2(x(k)) \leq x_\mathrm{upper} \\ & u_\mathrm{lower} \leq f_3(u(k)) \leq u_\mathrm{upper}, \end{aligned} \tag{2.5}$$

with both controlled variables and inputs constrained between lower and upper bounds by an arbitrary function and each predicted value of the controlled variable being calculated by another arbitrary function.

2.2 Economic MPC

Standard MPC can optimise for general quality of control (steady-state error, overshoot, etc.). However, in some applications, optimising for performance of economic type like electrical power consumption or production can be desired. This means an Economic MPC aids in higher levels of automation pyramid by optimising costs and production efficiency on top of controlling a system. Mentioned criteria like electrical power consumption can be linearly or non-linearly related to the states or inputs of the system, and also to external factors. While these types of additions to the optimisation problem of MPC do not need to be directly related to the economy of operating the system, they often are, and that is how this MPC type got its name.

The economic terms affecting costs of operating the system can be added to the cost function (like the one in equation (2.1)) to effectively optimise not only control but also the economic problem of the operation. An example of these terms could be

$$\sum_{i=1}^{H_\mathrm{p}} w_{c_i} \cdot c^\mathrm{T}(t) \cdot f(u_i(t)), \tag{2.6}$$

where $c^\mathrm{T}(t)$ represents a vector of proportional costs of operation for each input at any given time. Function f can be any function that describes how the cost is affected by the value of the input. In cases when reaching a specific setpoint is not necessary, and instead the controlled variables only need to stay within certain limits, this operation cost function can replace the error term in equation (2.1) completely, serving as the primary optimisation driving force for the controller [9].

2.3 Non-linear MPC and LPV MPC

As it was already mentioned, cost functions as well as constraints can often be non-linear in nature. An example of this could be a model of a system where a pressure loss occurs inside a pipe with flowing fluids. This pressure loss is a function of the volume flow rate of the flowing fluid. This relation can be described by many empirical formulas (such as the Hazen-Williams formula), all of which are non-linear. If problems like these need to be solved, a non-linear solver needs to be used that can handle them and reach an optimal solution. However, due to the increased complexity of this problem, these solvers are slow compared to the usual linear and quadratic programming solvers that only allow linear constraints of type

$$A_\mathrm{ineq} a - b \leq 0. \tag{2.7}$$

Due to the increased times of computation for large and complex models with non-linear solvers, solutions to avoid non-linearities are often explored. A simple and intuitive solution could be linearisation using a derivative of the non-linear function at a selected point. Here a general non-linear function

$$f(a(k)) = 0, \tag{2.8}$$

is transformed into a linear function

$$f(a_1) + \left.\frac{\mathrm{d}f(a)}{\mathrm{d}a}\right|_{a=a_1} (a - a_1) = 0. \tag{2.9}$$

The drawback of this approach is that it is only accurate at one value (a_1) and in its close proximity. This is why it is often used in control of systems where only one steady state is present and/or needed to be reached (singular reference value). If there are more, this approach may become inaccurate for many of them and therefore more difficult to control. This can be solved by creating a lookup table with derivatives of the non-linear function at multiple points. However, this solution can cause the controller to rapidly grow in size, especially if the system is complex and contains many variables.

Linear parameter-varying (LPV) is a different solution to the problem of non-linearity. Here, no derivatives are calculated, and instead, a precalculated matrix of values is used to calculate the approximate value of the non-linear function. This way, calculation takes on a form of

$$\Phi(k) \cdot a(k) + b = 0, \tag{2.10}$$

where matrix Φ is a function of a scheduling variable θ

$$\Phi(k) = f(\theta(k)), \tag{2.11}$$

where $\theta = a$. Here, when one set of variables (a vector) of the matrix is taken, it can be multiplied by the vector of current values and a close approximation of the non-linear function value for each is achieved. This means that values of Φ are calculated as

$$\Phi_{ij} = \frac{f(a_i(k))}{a_i(k)}. \tag{2.12}$$

Note that equation (2.12) is only a representation of the value Φ_{ij} and not necessarily a real calculation. Should f be a simple quadratic function a^2, Φ_{ij} would simply be a to ensure equality as shown in equation (2.10). Since this matrix needs to be calculated ahead of time, the future value of a needs to be estimated. Finding or calculating good estimated values can often prove quite difficult; however, in the case of MPC, this is not the case. MPC already calculates predicted values of the system's variables all the way until the end of the prediction horizon. These values can be used to create the Φ matrix for each of the required steps of the optimisation problem. Once the matrix is created, it can be passed into the MPC as a constant parameter to be used in equations linearly. This way, non-linear constraints can be transformed into a linear version, allowing for the use of much faster QP solvers [8][5].

Chapter 3

Practical part

3.1 Richmond WDN

The Richmond water distribution network was selected as the case study in this work. This network is located in Richmond in the ceremonial county of North Yorkshire, England. The model used in this project consists of 6 tanks, 7 pumps, 11 demand sectors, 42 junctions and 44 pipes [10].

Tanks and junctions are together classified as nodes and in the network serve as points where a certain water pressure is required or stored (in the case of tanks - storage nodes). These nodes are connected by pipes, denoted as lines in the network diagram. Note that the demands are not connected to nodes by a pipe, but rather the node itself has the demand requirements and withdraws water from the network. Additionally, pumps are also not connected to nodes by pipes, but instead the pumps themselves serve as the connection between nodes. Pumps are located across the network to force water flow at the required value to different parts of the network. Water for the network is sourced from a reservoir shown in the lower right corner of the diagram.

For the purposes of modelling, controller creation and running the simulation, water levels/hydraulic heads in the storage nodes (tanks) will serve as states and manipulated flows (volume flow rate through the pumps) as inputs.

However, the reservoir of water has a high enough hydraulic head for the inflowing water to satisfy constraints of tank A, meaning pumps 1A, 2A, and 3A would never need to be turned on for the water to reach this tank, thanks to the two bypass pipe routes that can be seen in Figure 3.1. However, this would also lead to the loss of control of tank A and subsequently problems with controlling the rest of the network. Therefore, pipes allowing the water to bypass pumps 1A, 2A and 3A were removed from the model, leading to the modified network shown in Figure 3.2. Hence, Figure 3.2 shows the topology of the network actually used in this work.

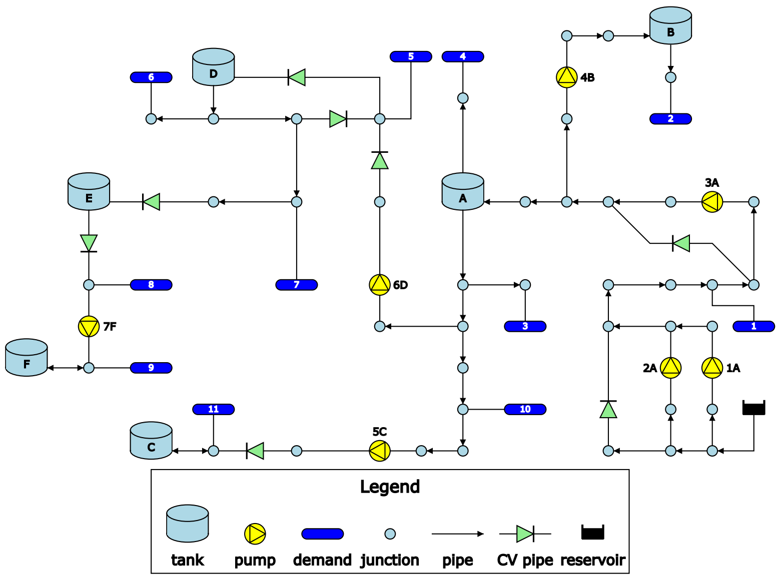

Figure 3.1: Original topology of the Richmond WDN

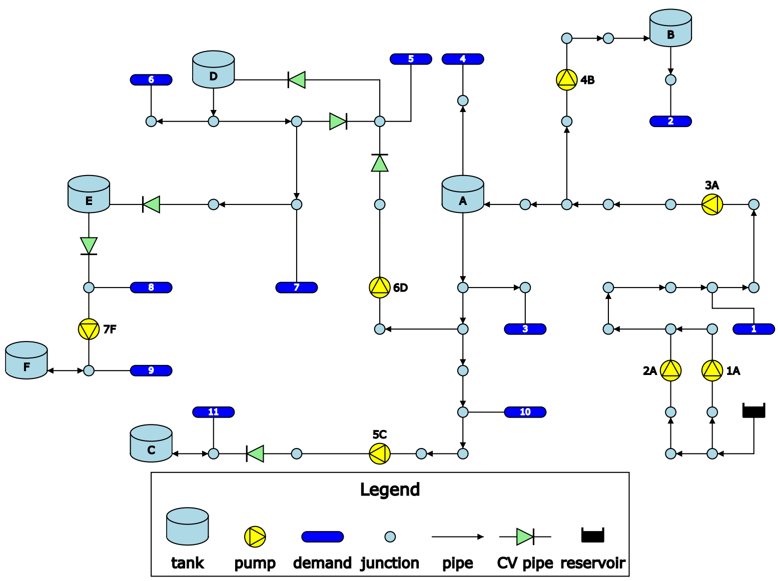

Figure 3.2: Modified WDN (after bypass pipes removal)

3.2 Mathematical Model and Optimization Problem

3.2.1 WDN Model

Storage tanks in the WDN serve two purposes. Storing water in case of emergency when the reservoir is unable to provide fresh water to the network, and for generating sufficient hydrostatic pressure in the pipes. Volume of the water stored in the i^{th} tank can also be expressed in terms of water level when assuming the horizontal cross-sectional area of the tank remains constant, as

$$h_i(k) = \frac{V_i(k)}{S_i}. \tag{3.1}$$

If pressure in the system is expressed as hydraulic head in meters, the calculation of pressure created by stored water becomes simple

$$x_i(k) = \frac{V_i(k)}{S_i} + E_i, \tag{3.2}$$

where E_i is the elevation of the tank. The hydraulic head generated is therefore the same as the height of an equivalent static column of water.

To model the dynamics of the tanks, a mass balance for the water needs to be created. Assuming constant density, the general discrete mass balance, which applies here, can be written in terms of volume as

$$V(k+1) = V(k) + T_{\text{s}}(\sum q_{\text{in}} - \sum q_{\text{out}}), \tag{3.3}$$

where q_{in} is volume flow rate coming into the tank and q_{in} is volume flow rate coming out of the tank. To relate the change in hydraulic head to the change in volume, division by the tank's cross-sectional area needs to be introduced, as can be seen in equation (3.1), to get the form

$$x(k+1) = x(k) + \frac{T_{\text{s}}(\sum q_{\text{in}} - \sum q_{\text{out}})}{S}. \tag{3.4}$$

Tanks can be directly connected to a pipe, a pump or a demand. These three types of connections are the source of in-flowing or out-flowing water. When using a vector for each of these system variables, head calculation for the tank requires multiplying them by matrices that correctly describe the network's topology. Then, the final form of the equation for the tanks takes on the form

$$x(k+1) = A \cdot x(k) + B_{\text{u}} \cdot u(k) + B_{\text{v}} \cdot v(k) + B_{\text{d}} \cdot d(k), \tag{3.5}$$

where $u(k)$ are manipulated flows (flows generated by pumps), $v(k)$ are non-manipulated flows (flows through the pipes due to conservation of mass) and $d(k)$ are flows directly to a demand and out of the network.

For each of the junctions in the network, mass needs to be conserved. Assuming an incompressible liquid with constant density, nodes need to follow the simple mass balance equation

$$0 = \sum q_{\mathrm{in}} - \sum q_{\mathrm{out}}. \tag{3.6}$$

Similar to the equation (3.5), the sum of all possible flows is used for each junction, where matrices used will this time also describe node connections. However, this time there is no accumulation, and the sum needs to be equal to zero. The resulting equation is therefore

$$0 = E_{\mathrm{u}} \cdot u(k) + E_{\mathrm{v}} \cdot v(k) + E_{\mathrm{d}} \cdot d(k). \tag{3.7}$$

Hydraulic head can be calculated at any place from the sum of heads created by storage nodes (tanks) and non-storage nodes (junctions) directly connected to it, and the head loss due to the water flow in the connecting pipe. This relation is also static, and this sum has to be equal to zero. Therefore, this relation can be expressed as

$$0 = P_{\mathrm{x}} \cdot x(k) + P_{\mathrm{z}} \cdot z(k) + \psi(v(k)), \tag{3.8}$$

where $x(k)$ is a vector of hydraulic heads of the tanks, $z(k)$ is a vector of hydraulic heads of junctions, and P_{x} and P_{z} are once again matrices denoting their interconnections in the network's topology. $\psi(v(k))$ is an empirical non-linear function describing the head loss in the connection (pipe) as a function of the water flow. For the purposes of this project, the Hazen-Williams formula was used because it is simple to implement but still provides acceptable accuracy for WDN systems. This formula describes head loss in pipes due to the volume flow rate as

$$h_{\mathrm{l}} = \frac{10.67 \cdot l}{c_{\mathrm{r}}^{1.852} \cdot d_{\mathrm{inner}}^{4.8704}} \cdot q^{1.852}, \tag{3.9}$$

where l is the length of the pipe, d_{inner} is the inner diameter, c_{r} is the pipe roughness coefficient and q is the volume flow rate inside the pipe. This equation introduces non-linearity to the system as the flows inside the pipes ($q = v(k)$) are exponentiated to the power of 1.852 [11].

In the case of LPV, the equation (3.8) is linearised by introducing another matrix Φ to take on a form

$$0 = P_{\mathrm{x}} \cdot x(k) + P_{\mathrm{z}} \cdot z(k) + \Phi(k) \cdot v(k), \tag{3.10}$$

where $\Phi(k)$ contains values calculated from previous predicted flows inside the pipes.

3.2.2 Optimisation Problem

The objective function for the WDN Economic MPC consists of four main parts. The first is the main economic problem of operating such a network. It calculates the cost of running the pumps of the WDN at the required power level. Expressing this problem mathematically looks like

$$\lambda_1 \cdot \alpha(k)^{\mathrm{T}} \cdot u(k), \tag{3.11}$$

where $\alpha(k)$ represents a vector of proportional costs of operation for each input at the current time. This is used to reflect the way the price of electricity may change over the 24 hours of the day.

Should it be desired to prefer using certain pumps over others in places where multiple pumps have the same effect on the network (like pumps 1A and 2A in Richmond WDN, as can be seen in Figure 3.2), this expression can use a diagonal matrix of individual weights for each pump instead of one weight constant to target specific pumps more. In this case, the expression would take on a form

$$\alpha(k)^{\mathrm{T}} \cdot W_{\alpha} \cdot u(k). \tag{3.12}$$

This same concept can be applied to each of the following parts of the objective function as well. However, as no such choices were desired in this project and also for simplicity's sake, only numerical constants will be used in the following equations, multiplying each individual element of the objective function term by the same value.

The second part of the objective function is used to penalise the controller for bringing the hydraulic head of the storage node below the chosen safety level. This safety level denoted x_{s} is introduced to solve possible irregular variations in demands between discrete time steps. It is implemented in the form of a soft limit, meaning it can be violated, but the cost function will increase by doing so. Expressing this soft limit mathematically can be done in the form of this constraint

$$x(k+1) \geq x_{\mathrm{s}} - \xi(k), \tag{3.13}$$

where $\xi(k)$ is a slack variable which will be used in the objective function as

$$\lambda_2 \cdot \xi(k)^{\mathrm{T}} \cdot \xi(k). \tag{3.14}$$

The third part of the objective function remains the same as in the original MPC example in equation (2.1), where the controller penalises sudden changes of inputs. This term takes on the form of

$$\lambda_3 \cdot \Delta u(k)^{\mathrm{T}} \cdot \Delta u(k). \tag{3.15}$$

Introducing this part of the objective function leads to smoother control of the network, which can prolong its lifespan.

The last part of the objective function does not have any physical representation or a way to be imagined in the real world. This part addresses the problems that arise with calculating pressure drops inside the pipes. Since these calculations are only approximate, setting equations (3.8) and (3.10) actually equal to zero would lead to an infeasible problem. Therefore, a slack variable is introduced, making up the final part of the objective function. This part has the form of

$$\lambda_4 \cdot e_{\mathrm{p}}(k)^{\mathrm{T}} \cdot e_{\mathrm{p}}(k), \tag{3.16}$$

where e_{p} is the slack variable which replaces zero in equations (3.8) and (3.10). By introducing this slack variable in the objective function, its value is minimised close to zero, allowing for the problem to be feasible and also close to reality.

Adding each of these individual parts yields the full objective function used for finding the optimal solution. The value of these terms of the objective function is added up for each step in the prediction horizon:

$$\begin{aligned}\sum_{k=1}^{H_{\mathrm{p}}} \lambda_1 \cdot \alpha(k)^{\mathrm{T}} \cdot u(k) + \lambda_2 \cdot \xi(k)^{\mathrm{T}} \cdot \xi(k) + \\ \lambda_3 \cdot \Delta u(k)^{\mathrm{T}} \cdot \Delta u(k) + \lambda_4 \cdot e_{\mathrm{p}}(k)^{\mathrm{T}} \cdot e_{\mathrm{p}}(k)\end{aligned} \tag{3.17}$$

This objective function is minimised by the optimisation solver and is subject to the following constraints:

$$x_{\min} \leq x(k+1) \leq x_{\max}, \tag{3.18}$$

$$u_{\min} \leq u(k) \leq u_{\max}, \tag{3.19}$$

$$v_{\min} \leq v(k) \leq v_{\max}, \tag{3.20}$$

$$z_{\min} \leq z(k) \leq z_{\max}, \tag{3.21}$$

$$x(k+1) \geq x_{\mathrm{s}} - \xi(k), \tag{3.22}$$

in addition to the equations (3.5), (3.7) and (3.8) (or (3.10) in the case of LPV).

3.3 Optimisation Solvers

For this work, CasADi is being used to create the optimisation problem for the controller. Originally, CasADi was a tool for algorithmic differentiation (AD) using a syntax borrowed from computer algebra systems (CAS), giving it its name. While not being an optimal control problem (OCP) solver, it provides the user with a set of building blocks that can be used to implement general-purpose or specific-purpose

OCP solvers efficiently with a modest programming effort. It is well-suited for solving non-linear programming problems by utilising gradient-based numerical optimisation. Additionally, it also offers a big assortment of LP and QP solvers (like Gurobi or qpOASES) if the problem that needs to be solved allows for their use [1].

CasADi's great performance in non-linear OCP using the built-in IPOPT solver is the main reason why it was chosen for this work. For comparison, when using YALMIP (MATLAB toolbox well suited for optimisation problems like LPs and QPs but also capable of utilising non-linear solvers) with the fmincon solver on this WDN, the solution could not be reached in a reasonable time (the time of simulating on a step exceeded the sample time itself). Ipopt's great speed (in comparison to other solvers) made it a perfect choice for this work. This way, it can be objectively judged if LPV MPC can not only reach similar results but also outperform these fast non-linear solvers in terms of speed.

3.4 Implementing Economic MPC for Richmond WDN in MATLAB

For the simulation, a sample time of one hour was selected and a prediction horizon of 24 hours. This is an ideal value for the prediction horizon of this controller as it is reasonably sized (not excessively long, causing unnecessarily long computation) and the demands and electricity prices repeat with a period of 24 hours. This means the controller can see every possible demand and operational cost it can encounter.

Richmond WDN model was loaded into MATLAB from an .inp file, which contains all necessary variables and information used in the system modelling and simulation, like the network topology, elevations, length, diameter and roughness of the pipes.

Matrices A, B_{u}, B_{v}, B_{d}, E_{u}, E_{v}, E_{d}, P_{x}, P_{z} were created based on these connections, where direct connection between two individual elements of the network is denoted as 1 (or -1 depending on directionality), otherwise 0. Additionally, these matrices were multiplied/divided by the required constants (for example, sample time of 3600 seconds) when needed to ensure the correct units were at every part of the equations. If a certain connection only allows for one direction of fluid flow (CV pipes seen in Figure 3.2), constraints for these pipes' minimal (or maximal) flow values are introduced, where putting one as 0 would lead to one-directional flow.

Using equation (3.17) and the following constraints as a template, the controller was implemented in the following manner (not the exact actual code, only here for illustrative purposes):

```
objective = 0
for k = 1:Hp
 delta_u = u(:,k) - past_u;

 objective += lambda(1) * alpha(:,k)' * u(:,k);
 objective += lambda(2) * ksi(:,k)' * ksi(:,k);
 objective += lambda(3) * delta_u' * delta_u;
 objective += lambda(4) * e_p(:,k)' * e_p(:,k);

 level = A * x(:,k) + B_u * u(:,k) + B_v * v(:,k) + B_d * d(:,k);
 mass = E_u * u(:,k) + E_v * v(:,k) + E_d * d(:,k);
 pressure = P_x * x(:,k) + P_z * z(:,k);

 if NL
  pressure += R' .* v(:,k) .* abs(v(:,k)) .^ 0.852;
 else
  pressure += Phi(:,k) .* v(:,k);
 end

 opti.subject_to(level == x(:,k+1));
 opti.subject_to(mass == zeros(size(E_u(:,1))));
 opti.subject_to(pressure == e_p(:,k));

 opti.subject_to(xmin <= x(:,k+1) <= xmax);
 opti.subject_to(umin <= u(:,k+1) <= umax);
 opti.subject_to(vmin <= v(:,k+1) <= vmax);
 opti.subject_to(zmin <= z(:,k+1) <= zmax);
 opti.subject_to(x(:,k+1) >= xs-ksi(:,k));

 past_u = u(:,k);
end

opti.solver(solver_name, opts);
opti.minimize(objective);
```

The main simulation loop was created next, where necessary variables like *past_u*, *alpha*, *d*, and *Phi* in the case of LPV were passed as parameters, and the optimiser was allowed to find an optimal solution. The first values of the variables returned by the optimiser were saved and used as current variables for the next step.

Creation of matrix *Phi* for the LPV approach was done for each step, before optimisation took place. To be able to multiply singular vector of values for each step in the prediction horizon, *Phi* matrix values were calculated exactly as in equation (3.9) with the only difference being that q (or v in this case) in the equation was exponentiated to the power of 0.852 instead of 1.852 as equation (2.12) would suggest. By multiplying *Phi* with v in the controller's optimisation problem, the original Hazen-Williams formula is achieved. The creation of the *Phi* matrix, therefore, looks as follows:

```
Phi = zeros(num_of_pipes, Hp);
for i = 2:Hp+1
 if i == Hp+1
  i = 1;
 end
 Phi(:,i) = R’ .* (abs(v_sol_last(:,i))).^0.852;
end
```

Here, the flow values are shifted by one since the first estimated value is a flow value of the previous step. Since the prediction horizon was 24 hours and the demands and electricity tariffs also periodically repeat every 24 hours, the first value of the flows was used to calculate the last values in the *Phi* matrix, because of the periodicity assumption for the whole system.

Chapter 4

Results

After the MPC was implemented in MATLAB, the simulation was set to run for several days in simulation for the system to reach a stable periodic cycle of 24 hours. This took roughly one week in the simulation, since the initial state values were set higher than the controller found optimal. The results are therefore collected over a period of three days, starting on day 9.

As can be seen in Figure 4.1, the non-linear MPC and LPV MPC achieve very similar results. The difference between the two approaches is negligible, and it can barely be seen on the individual plots. Both controllers successfully satisfy the minimal required pressure demands by keeping the hydraulic heads above the lower limit. The effect of the safety level can also be seen as the value is not staying at the lower limit, but is being increased by the controller when it is optimal to do so.

The same as was said for the Figure 4.1 can be said for Figure 4.2. Both MPC implementations calculate almost exactly the same inputs for the system, with the only exception being pump 1A. Here, the difference, while small, is at least noticeable with the controller taking a slightly different path to reach the optimal solution. The constraints are once again satisfied, and the controller keeps the input values inside the defined bounds.

From the economic standpoint, it is also clear that the controller tries to pump water into the network when the cost of electrical power is low, as the lowest value of the control variables is usually located in the high tariff region, while the highest value of the control variables is located in the low tariff region. Figure 4.3 shows the value of the objective function and its individual terms. It can be seen that the overall value is mostly determined by the slack variable e_{p} used in the pressure equation. Since this term is essential for ensuring feasibility but does not have a physical representation or effect on the real system, its contribution to the objective function here has minor significance. Of the three remaining objective function terms, only the economic u

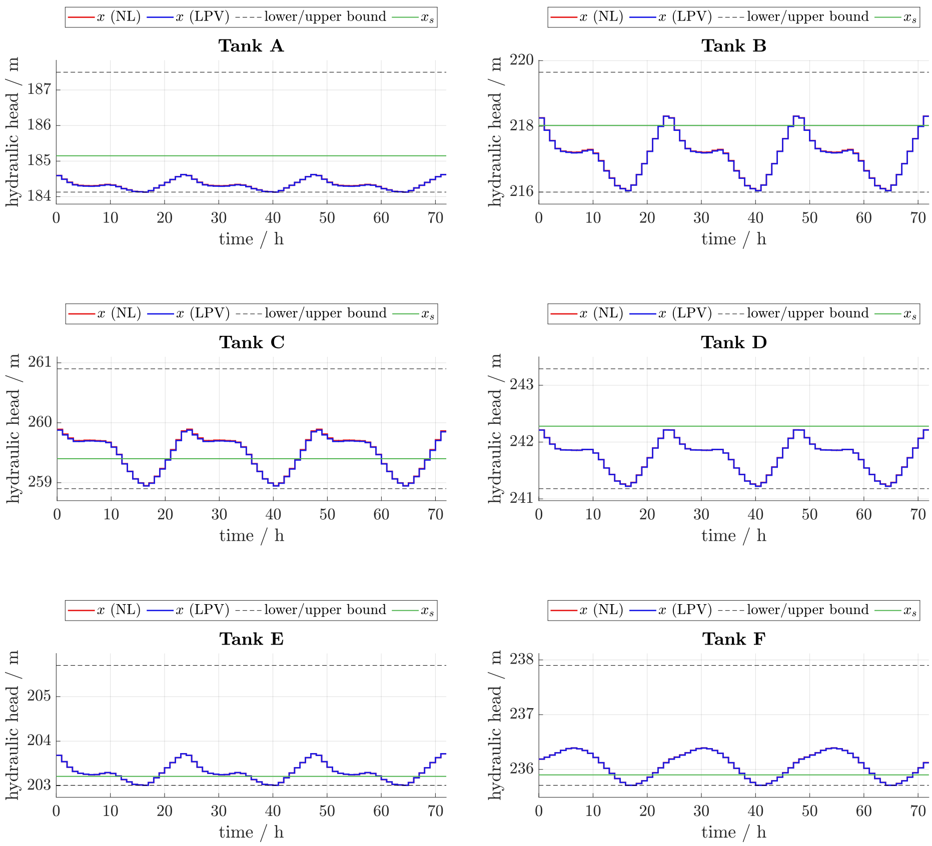

Figure 4.1: Hydraulic heads inside the network tanks (system states)

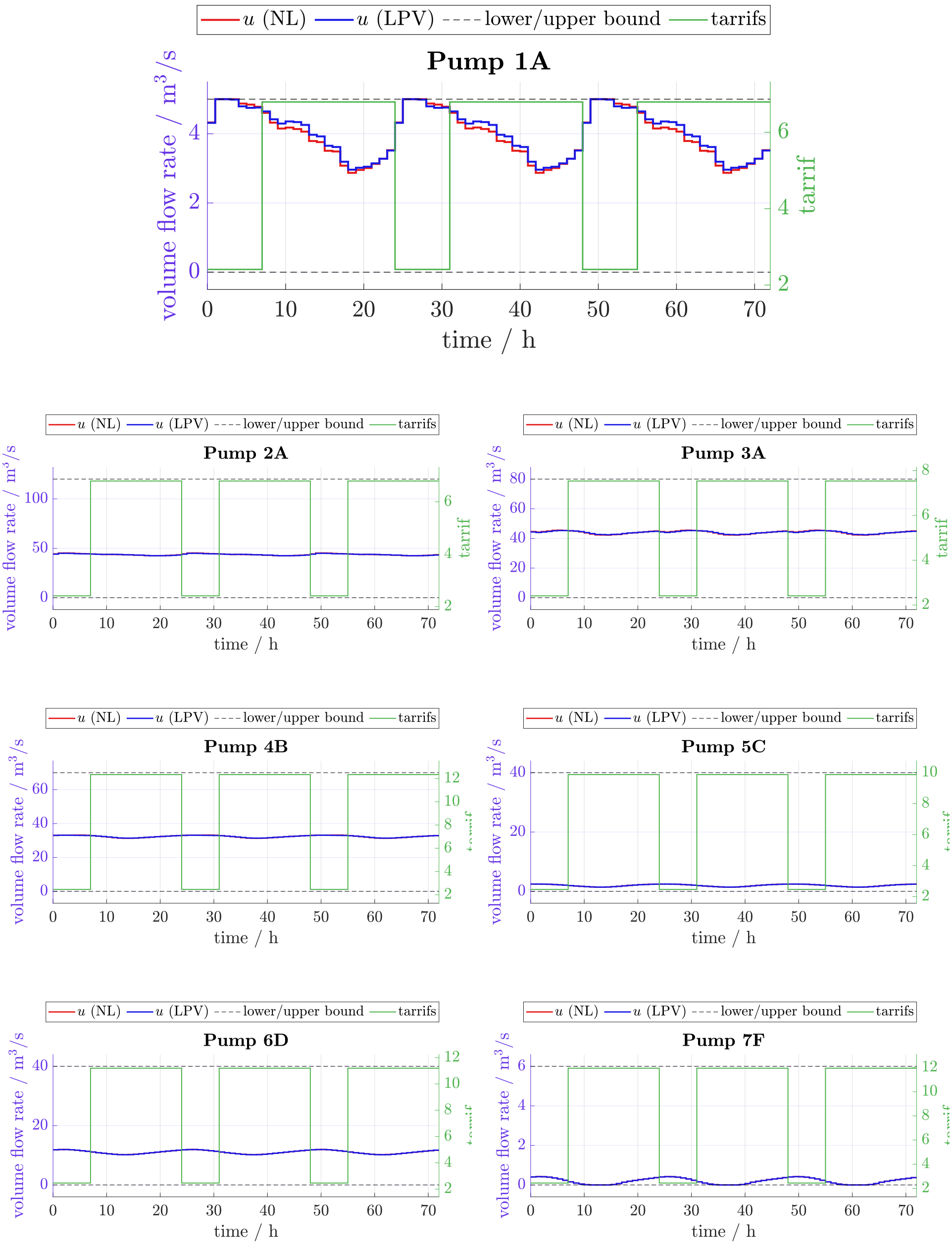

Figure 4.2: Volume flow rate through the network's pumps (system inputs)

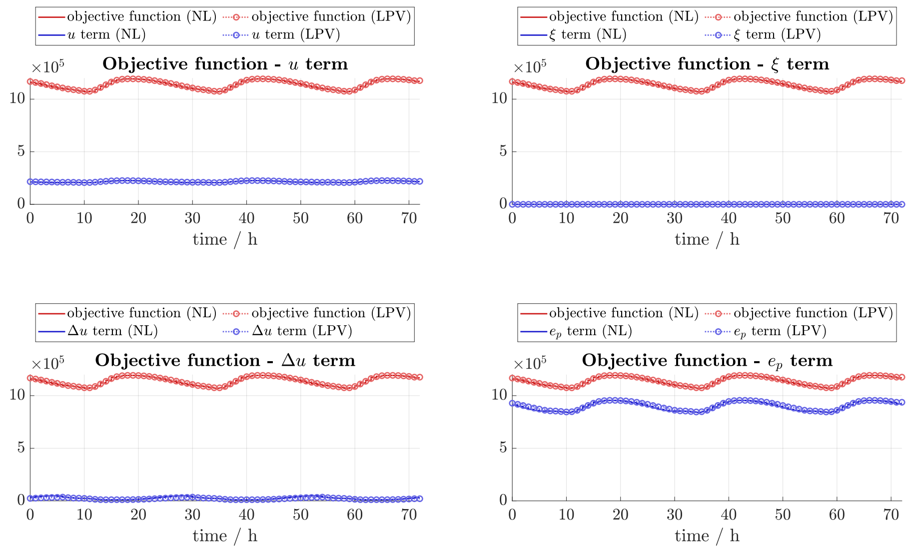

Figure 4.3: Values of the objective function and each term of the objective function for both control approaches

term has a noticeably larger value. The plot clearly shows that the LPV approach manages to follow the non-linear approach very closely in terms of operational costs. To determine how well LPV did from the economic standpoint, the u term values over a 24-hour period were added for both controllers.

Table 4.1: Sum of u term over a period of 24 hours

Non-linear MPC	LPV MPC
$5.1522 \cdot 10^6$	$5.1863 \cdot 10^6$

Table 4.1 shows the sum of these values. It can be seen that LPV MPC does have slightly higher operational costs compared to the non-linear MPC. The LPV approach in this case caused an increase of 0.6607% in costs. While not being as economically optimal as a non-linear controller, the LPV MPC is much faster.

Table 4.2: Time needed to complete a simulation of 24 hours on a computer with AMD Ryzen 5 7537U CPU

	Non-linear solver (IPOPT)	QP solver (qpOASES)
Non-linear	125 seconds	-
LPV	89 seconds	10 seconds

As can be seen from Table 4.2, the LPV MPC significantly outperforms non-linear MPC in terms of speed, even when the same non-linear solver is used. The real improvement with using this method comes from using a QP solver, which is possible due to using LPV, which turns the constraints of the optimisation problem linear. In this case, the simulation for finding an optimal solution takes a fraction of the previous runs. With a higher number of constraints, the optimisation problem becomes more difficult and takes longer to calculate. The WDN used in this work is quite small and simple, so the time to solve optimal control using a non-linear MPC controller is still acceptable. However, with larger WDNs that have hundreds or thousands of different nodes and pipes, the time it takes to calculate optimal input values for them can grow larger than the sample time itself. In this case, it would become necessary to use LPV.

Chapter 5

Conclusions

This work aimed to model a dynamic system and then design and create a controller based on the Model Predictive Control (MPC) approach. The chosen dynamic system is a water distribution network (WDN), which is non-linear. Therefore, as a possible solution for the drawbacks of this controller, a second controller using linear parameter-varying (LPV) was created so that their results and performance could be compared to determine if the LPV approach is viable.

In the theoretical part, the basics of MPC are discussed, explaining how it works and what its benefits and drawbacks are. Building on this knowledge, a type of MPC called Economic MPC, which was used in this work, was explained. This approach is very beneficial for use in systems, where their economic performance can vary greatly depending on the chosen path of control. Finally, non-linear optimisation problems for MPC were discussed. These problems tend to be computationally expensive, and therefore, ways to avoid non-linear optimisation are used, such as LPV, which was explained more closely in the chapter.

In the practical part, Richmond WDN, the case study for this work, was introduced. This network (or its modified version, as mentioned in the chapter) serves as a relatively simple WDN system on which the results of this work can be shown. Following the introduction of this network, the creation of its model is discussed next. Each of the modelled individual elements of which the network consists was discussed in detail, explaining how the equations used later were made, starting with the storage nodes (tanks). These are the dynamic parts of the network, and the hydraulic heads of these tanks are the states of the system that can be used to determine the quality of the control. After the tanks, the mass balance equation for the network junctions and the pressure equations for the network pipes were explained. The pressure equation is the source of non-linearity in this system, due to the use of the Hazen-Williams formula to describe the drop in pressure inside the pipes. A way of utilising LPV was shown as a solution for this non-linear equation. Next, the optimisation problem for this Economic MPC was shown, starting with the objective function to be minimised by an optimisation solver. This objective function consists of four parts that were shown and explained in detail. Afterwards, optimisation solvers and CasADi (the tool used to implement this MPC in MATLAB) were shortly mentioned and explained. Finally, the actual implementation of the model and MPC was explained, also showing pseudo-code to help with the explanation.

The last part of this thesis shows the achieved results from this work. In two figures, the resulting plots of hydraulic heads of the tanks and volume flow rates through the pumps of the network were shown. On these plots, both the non-linear MPC and the LPV MPC results can be seen for comparison. These plots show that LPV is able to very closely replicate the results of non-linear MPC. The following figure of the objective function and its terms supports this and shows that both controllers reached a similarly optimal solution.

While the LPV MPC did have a slight increase in operational costs compared to the non-linear MPC, it is far superior in calculation speed due to the removal of non-linearity. If the complexity of a system causes the non-linear approach to not be viable due to the calculation time exceeding the sample time, LPV would need to be used instead. If that is not the case, however, to determine which option is optimal calls for another economic optimisation problem. Here, the costs of the hardware performing the calculations and the cost of powering it need to be considered in addition to the operational costs of the system. A faster and therefore more expensive computer is needed to calculate the non-linear problem, while a linearised problem would have lower requirements for computational hardware. Hence, depending on the system, the LPV MPC might not only be faster but also more economically viable due to the lower initial costs as well as lower costs of powering the controller.

Appendix A

Resumé

Cieľom tejto práce bolo modelovať dynamický systém a následne navrhnúť a vytvoriť regulátor prediktívneho riadenia založeného na modeli (MPC). Zvoleným dynamickým systémom je rozvodná sieť vody (WDN), ktorá je nelineárna. Preto ako možné riešenie nevýhod tohto regulátora bol vytvorený druhý regulátor využívajúci lineárne menenie parametrov (LPV), aby sa ich výsledky a výkon dali porovnať, a dosiahlo sa teda záveru, či je prístup LPV rentabilný.

V teoretickej časti sú rozobraté základy MPC, vysvetľuje sa ako funguje a aké sú jeho výhody a nevýhody. Nadväzujúc na tieto poznatky bol vysvetlený ekonomický typ MPC, ktorý bol použitý v tejto práci. Tento prístup je vysoko prospešný pre použitie v systémoch, kde sa ich ekonomický výkon môže značne líšiť v závislosti od zvoleného spôsobu riadenia. Nakoniec sa rozobrali optimalizačné problémy nelineárneho MPC. Tieto problémy bývajú výpočtovo nákladné, a preto sa používajú spôsoby, ako sa vyhnúť nelineárnej optimalizácii, ako napríklad LPV, ktorá bola bližšie vysvetlená v tejto kapitole.

V praktickej časti bola predstavená Richmond WDN, ktorá bola použitá ako problémová štúdia pre túto prácu. Táto sieť (presnejšie jej upravená verzia, ako je uvedené v tejto kapitole) slúži ako pomerne jednoduchý systém rozvodnej siete, na ktorom je možné ukázať výsledky tejto práce.

Po predstavení tejto siete sa ďalej diskutuje o vytvorení jej modelu. Každý z jednotlivých modelovaných častí, z ktorých sieť pozostáva, bol podrobne rozobraný, pričom bolo vysvetlené, ako boli vytvorené rovnice použité neskôr, počnúc zásobnými uzlami (nádržami). Tie sú dynamické časti siete a hydraulický tlak vytváraný týmito nádržami predstavuje stavy systému, ktoré možno použiť na určenie kvality riadenia. Po nádržiach bola vysvetlená rovnica hmotnostnej bilancie pre uzly a tlaková rovnica pre potrubia siete. Tlaková rovnica je zdrojom nelinearity v tomto systéme v dôsledku použitia Hazen-Williamsovho vzorca na opis straty tlaku vo vnútri potrubia. Ako riešenie pre túto nelineárnu rovnicu sa využil prístup LPV. Ďalej bol ukázaný optimalizačný problém pre tento ekonomický MPC regulátor, počnúc účelovou funkciou, ktorú má minimalizovať optimalizačný riešiteľ. Táto účelová funkcia pozostáva zo štyroch častí, ktoré boli ukázané a podrobne vysvetlené. Potom boli krátko spomenuté a vysvetlené optimalizačné riešitele a CasADi (nástroj používaný na implementáciu tohto MPC v MATLABe). Nakoniec bola vysvetlená skutočná implementácia modelu a MPC, pričom sa tiež ukázal pseudokód, ako pomôcka k vysvetleniu.

V poslednej časti práce sú uvedené dosiahnuté výsledky tejto práce. Na dvoch obrázkoch sú znázornené výsledné grafy hydraulických tlakov nádrží a objemových prietokov cez čerpadlá siete. Na týchto grafoch možno na porovnanie vidieť výsledky nelineárneho MPC aj LPV MPC regulátora. Tieto grafy ukazujú, že prístup LPV je schopný veľmi presne replikovať výsledky nelineárneho MPC. Nasledujúci obrázok účelovej funkcie a jej častí to potvrdzuje a ukazuje, že oba regulátory dosiahli podobné optimálne riešenie.

Kým LPV MPC regulátor mal mierne zvýšenie prevádzkových nákladov v porovnaní s nelineárnym, je oveľa rýchlejší vďaka odstráneniu nelinearity. Ak zložitosť systému spôsobí, že nelineárny prístup nie je realizovateľný v dôsledku času výpočtu presahujúceho čas vzorkovania, použiť LPV by bolo nevyhnutné. Ak tomu tak však nie je, určiť optimálny prístup, si vyžaduje ďalší problém ekonomickej optimalizácie. Tu je potrebné okrem prevádzkových nákladov systému zvážiť aj náklady na hardvér vykonávajúci výpočty a náklady na jeho napájanie. Na riešenie nelineárneho problému sú spravidla potrebné vyššie výpočtové nároky, zatiaľ čo linearizovaný problém môže byť riešený s nižšími požiadavkami na výpočtové zdroje. V závislosti od systému môže byť LPV MPC regulátor teda nielen rýchlejší, ale aj ekonomicky výhodnejší vďaka nižším počiatočným nákladom, ako aj nižším nákladom na jeho napájanie.

Appendix B

List of variables and their units

a - arbitrary variable
A - matrix of the system dynamics
A_{ineq} - inequality constraint matrix
α - vector of electricity costs for corresponding network pumps
b - arbitrary constant
B_{d} - matrix of connections between storage nodes and demand nodes
B_{u} - matrix of connections between storage nodes and pumps
B_{v} - matrix of connections between storage nodes and pipes
c - vector of proportional costs for each input
c_{r} - pipe roughness
d - volume flow demand [m^3/s]
d_{inner} - inner pipe diameter [m]
Δu - input change [$\mathrm{m}^3/\mathrm{s}/T_{\mathrm{s}}$]
E_{d} - matrix of connections between non-storage nodes and demand nodes
E_i - i^{th} tank elevation
e_{p} - slack variable for the pressure constraint [m]
E_{u} - matrix of connections between non-storage nodes and pumps
E_{v} - matrix of connections between non-storage nodes and pipes
f - arbitrary function
Φ - LPV matrix
h - height of the water stored inside a tank
h_{l} - head loss inside a pipe [m]
H_{p} - prediction horizon
J - objective function
l - pipe length [m]
λ_i - weight for i^{th} term of the objective function
ψ - arbitrary non-linear function describing pressure drop inside a pipe
P_{x} - matrix of connections between pipes and storage nodes
P_{z} - matrix of connections between pipes and non-storage nodes

q - volume flow rate [m^3/s]
r - reference/setpoint
R - constant part of the Hazen-Williams formula (used in code)
S - cross-sectional area of a tank
θ - scheduling variable
T_s - sample time
u - manipulated flows (flows inside the pumps)/inputs of a system [m^3/s]
v - unmanipulated flows (flows inside the pipes) [m^3/s]
V - volume of water inside a tank
w - weight
W - diagonal weight matrix
x - hydraulic heads of the storage nodes/controlled variables (states) of a system [m]
x_s - safety level [m]
ξ - slack variable for the safety level soft constraint [m]
z - hydraulic head at a non-storage node [m]

Bibliography

[1] ANDERSSON, Joel - GILLIS, Joris - HORN, Greg - RAWLINGS, James Blake - DIEHL, Moritz. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2018.

[2] FAINSTEIN, Susan S. Urban planning. *Britannica*, 2026.

[3] HORDON, Robert M., B.A., M.A., Ph.D. - KÄHLER, Karen N. Water use. *EBSCO*, 2024.

[4] NATHANSON, Jerry A. Water supply system. *Britannica*, 2026.

[5] POUR, Fatemeh K. - PUIG, Vicenç - CEMBRANO, Gabriela. Economic mpc-lpv control for the operational management of water distribution networks. *IFAC-PapersOnLine*, 52(23):88–93, 2019.

[6] PUIG, Vicenç - OCAMPO-MARTÍNEZ, Carlos. - PÉREZ, Ramon - CEMBRANO, Gabriela - QUEVEDO, Joseba - ESCOBET, Teresa. *Real-Time Monitoring and Operational Control of Drinking Water Systems*. Springer, 2017. ISBN 978-3-319-50750-7.

[7] RAWLINGS, James Blake - MAYNE, David Q. - DIEHL, Moritz. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2nd edition, 2020. 623p. ISBN 978-0-9759377-5-4.

[8] ROTONDO, Damiano - SÁNCHEZ, Helem S. - NEJJARI, Fatiha - PUIG, Vicenç. Análisis y diseño de sistemas lineales con parámetros variantes utilizando lmis. *Revista Iberoamericana de Automática e Informática industrial*, 2019.

[9] Sigurd SKOGESTAD. Advanced control using decomposition and simple elements. *Annual Reviews in Control*, 2023.

[10] TELLO, Andrés - TRUONG, Huy - LAZOVIK, Alexander - DEGELER, Victoria. Water distribution networks (wdns) - network operation state snapshots, 2024. Zenodo.

[11] WANNAPOP, Rangsan - JEARSIRIPONGKUL, Thira - JIAMJIROCH, Krit. Adaptive urban drinking water supply model using the effect of node elevation and head loss formula: A case study. *Heliyon*, 2024.