Università degli Studi di Padova

DIPARTIMENTO DI DI INGEGNERIA DELL'INFORMAZIONE
Corso di Laurea in Ingegneria dell'Automazione

TESI DI LAUREA MAGISTRALE

Applications Oriented Optimal Input Design: An Analysis of a quadruple water tank process

Laureando
Antonio Balsemin

Relatore
Prof. Giorgio Picci

Anno Accademico 2012-2013

Abstract

Model predictive control (MPC) has become an increasingly popular control strategy thanks to its ability to handle multivariable systems and constraints. This control technique makes use of a model of the system, therefore performances are highly dependent on the accuracy of the model chosen. The process of obtaining the model is often costly, for this reason system identification for MPC is an important topic. Applications oriented optimal input design enables optimization of the system identification experiments, leading to a set of models with the necessary accuracy for the intended application. In this thesis a method of system identification for MPC applications is simulated on a multivariable nonlinear system consisting of four interconnected water tanks. An analysis of the impact of MPC settings, such as active or no active constraints and different weight settings, is carried out. The initial hypothesis is that different MPC settings influence the obtained set of models. Simulations show that the hypothesis is correct and the result give rise to some interesting interpretations.

Acknowledgements

Writing my master thesis at KTH it was a great experience. I would like to thank many people for their support during all the time I worked on my thesis. First of all I want to thank my italian supervisor Prof. G. Picci who gave me the possibility to do my thesis at KTH. I would like to thank my examiner Prof. Bo Wahlberg who introduced me with really interesting research topics and gave me insightful comments. Special thanks to my supervisors Mariette Annergren and Christian Larsson for kindness and availability, for patience and help during all the time I worked on my thesis. This thesis would not exist without your guidance. Thanks to all the people of the Automatic Control Lab, for the exciting and inspiring work environment. It was really nice to attend your meetings. Thanks to Damiano Varagnolo for the useful LaTeXtips and support in general. Many thanks to the administration personal Gustafsson Kristina, Ström Anneli and Holmqvist Hanna for making sure that we always had what we need for our work. Thanks to Francesco and Sky, their company was fundamental during all the time I spent in Sweden.
Finally I want to thank my family for their love and support, for their faith and for giving me the possibility to have this beautiful experience.

Contents

1 Introduction

This master's thesis project was carried out at the Department of Automatic Control, Royal Institute of Technology and is based on the research of the PhD. Students Christian Larsson and Mariette Annergren. The needs to verify the theoretical results of their research on a concrete and interesting control problem is the motivation for this project work.

1.1 Main Topics

The main topics of this Master Thesis Project are *model predictive control*, *system identification*, *optimal input design* and *convex optimization*.

Model predictive control (MPC) is a particular type of controller which makes use of an internal model to predict the behaviour of the controlled system, starting at the current time, over a future prediction horizon. The predicted behavior depends on the input trajectory that is to be applied over the prediction horizon. The idea is to select the input that gives the best predicted behavior, that, at the same time, minimize a function cost. At each step, the whole input trajectory is computed but only the first element of it is applied to the

system. At the next step this procedure is repeated, [16]. As we said, the MPC controller has an internal model, that is used to describe the real system. The accuracy of the model, affects the control performance of the MPC. Procedures involving system identification are often used to construct the model.

"*System identification* deals with the problem of building mathematical models of dynamical systems based on observed data from the system", [15]. The goal is to find a model $\mathcal{M}$ that describes the relevant properties of the studied system $\mathcal{S}$. The main steps of the identification process is to:

- collect data from the system (inputs and outputs),
- define a set of candidate models, choosing a model structure,
- choose a criterion by which candidate models can be assigned using the data,
- estimate parameter values based on data, structure and criterion,
- validate the model.

The validation step involves various procedures to assess how the model is related to observed data. If a model does not pass the model validation test, we must go back and revise the various steps of the procedure.

Optimal input design relates to finding the input signal that assures that the estimates become as good as possible. The traditional way to obtain an informative experiment is to minimize the covariance matrix of the estimated parameter vector. However, in application oriented input design, [12], [8], et cetera, the focus is shifted towards the application of the estimated model. Instead of the covariance matrix of the estimated parameter vector, the distance between nominal performance and performance achieved with the model is considered.

Convex optimization is a class of mathematical optimization. In this class, the objective and the constraint functions have to be convex. General optimization problems can be very difficult to solve or even intractable. Moreover if they are solvable, they can require very long computation time. Convex optimization problems, on the other hand, are solvable and the solutions can be found efficiently and reliably. For these good qualities one should always rewrite an optimization problem in convex form, whenever possible.

1.2 Problem formulation

For controllers based on a model of the system, the accuracy of the model is crucial. Mathematical models can not perfectly describe the systems they are meant to describe. This discrepancy is called *plant-model mismatch*. When a specific application is considered, usually not all system properties have the same importance. Hence it is crucial that the model used by the controller captures the system properties important for the intended application. These, in fact, affect the control performance more evidently. The quality of the estimated models is affected by the input used in the identification experiment. Optimal input design is used as a tool, to obtain an input signal that uses as little resources as possible, while guaranteeing that the performance specifications in the intended application are met. Thus giving a set of acceptable models. This problem is formulated as follows [12]

$$\begin{aligned} \underset{\text{input}}{\text{minimize}} \quad & \text{Experimental effort} \\ \text{subject to} \quad & \text{Performance specifications} \end{aligned} \tag{1.1}$$

Within the framework described, we consider a particular nonlinear system composed by four interconnected water tanks controlled by MPC. This system has the ability to change one zero location, from minimum phase to non minimum phase by simply acting on a physical parameter. An analysis of the impact of MPC settings, such as active or no active constraints and different weight settings, is carried out.

1.3 Related Work

The research performed by Prof. Bo Wahlberg [18], and Prof. Håkan Hjalmarsson [8], deals with experiment design for system identification (more references in this field can be found in the references of [18] and [8]). They combine the criterion, that the model should fulfil in the system identification process, with the requirements on the system's behaviour in the control theory problem. In [8] the *cost of complexity* is defined as the minimum possible experimental effort required to obtain a parameter estimate that guarantees the desired performance in the control application. Furthermore, it was underlined that the identification experiments should reveal system properties important for

the application and conceal irrelevant properties.

The identification experiment can be performed in *open loop* or in *closed loop*. Open loop identification is when there is no feedback control of the system during the identification experiment. Closed loop identification is when there is feedback control. Methods for optimal input design in system identification for control with *open loop identification* have been extensively treated in [8], [14], [13], [12], [19].

In [19] model based control design methods, such as MPC, where the model is obtained by means of the prediction error system identification method, are studied. The accuracy of the model used in this control method is a main issue, as it is strictly related to degradation in control performance. Control performance is defined using a cost function that specifies which parameter values give acceptable performance. The objective is to find a minimum variance input signal to be used in the system identification experiment, such that the control performance specification is fulfilled with a given probability when using the estimated model in the control design. This problem can be reformulated as a convex optmization problem. Examples with FIR models show that up to a factor two in input power can be gained when using the optimal input compared to white noise. Other examples, show that much higher gain can be obtained in input power.

MPC applications are discussed more in [13], where a water tank process is considered. A scheme for optimal input design in a MPC context is presented, the major challenges with the practical implementation are highlighted, and an algorithm for identification experiments is proposed. It is also pointed out that there is no good way to consider time domain constraints in the identification part of the method, but they can be included in the calculation of the application cost.
Thanks to the work of Mariette Annergren and Christian Larsson a Model Based Optimal Input Design Toolbox for Matlab (MOOSE) has been developed, [4]. It simplifies the implementation of optimal input design problems, providing an extra layer between the user and `CVX`, a package for specifying and solving convex programs [7, 6]. At the time of writing, MOOSE only handles open loop identification.

1.4 Thesis outline

The thesis is organised in Chapters. In Chapter 2 treats the background. Chapter 3 focuses on the application oriented optimal input design in an MPC context. In Chapter 4 the quadruple water tank system is described. In Chapter 5 are presented the results obtained from the simulations, results are briefly described. A more detailed analysis and comparisons of the results, are presented in Chapter 6.

2

Background

2.1 Convex Optimization

A convex optimization problem can be written in the following form,

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i \leq 0, \quad i = 1, \ldots, m \end{aligned} \tag{2.1}$$

where the functions f and g are convex. A particular kind of these problems is the so called *Semi Definite Program* (SDP) in which the objective function f is linear and the inequality constraints $g_i \leq 0$ are called Linear Matrix Inequalities (LMI). These problems have the following form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^T x \\ \text{subject to} \quad & x_1 F_1 + x_2 F_2 + \cdots + x_n F_n + G \preceq 0, \end{aligned} \tag{2.2}$$

where $F_i, G \in \mathbf{S}^k$ and $\mathbf{S}^k$ is a set of $k \times k$ symmetric matrices. In (2.2) it is possible to include multiple LMI constraints. For example the two constraints,

$$x_1 F_1 + x_2 F_2 + \cdots + x_n F_n + G \preceq 0, \qquad x_1 H_1 + x_2 H_2 + \cdots + x_n H_n + R \preceq 0,$$

can be rewritten in a single LMI constraint as follows

$$x_1 \begin{bmatrix} F_1 & 0 \\ 0 & H_1 \end{bmatrix} + x_2 \begin{bmatrix} F_2 & 0 \\ 0 & H_2 \end{bmatrix} + \cdots + x_n \begin{bmatrix} F_n & 0 \\ 0 & H_n \end{bmatrix} + \begin{bmatrix} G & 0 \\ 0 & R \end{bmatrix} \preceq 0.$$

Convex problems have the important property that if a solution is found, it is guaranteed to be globally optimal. Convex optimization problems can be solved reliably and efficiently, using one of the available optimization solvers. A very extensive reference to deepen the knowledge about convex optimization is [3].

2.1.1 Optimization software

Prof. Stephen Boyd and Dr. Michael Grant at the Department of Electrical Engineering, Stanford University, have developed a programming framework for convex optimization problems called CVX, [7]. It is implemented in the software Matlab and uses Matlab-syntax.
The PhD. Students Mariette Annergren and Christian Larsson developed a Matlab Toolbox for optimal input design called MOOSE, [4]. MOOSE is a model based optimal input design toolbox which simplify implementation of optimization problems found in input design. It provides an extra layer between the user and a convex optimization environment. For the flexibility and easy setting up of optimization problems this toolbox was used in this master thesis project.

2.2 Model predictive control

Model predictive control is a control method which has made a significant impact on control of industrial processes, especially in the petrochemical industry. In fact, most petrochemical plants and refineries have implemented MPC [21]. Due to increased microprocessor speeds, MPC is spreading out into other application fields. For example, for controlling heating and ventilation systems in buildings. Main reasons that prompt the use of MPC are the simple treatment of multivariate processes and the ability to handle constraints on state variables and signals. Constraints can arise from physical limitations of the plant (consider for example saturation limits due to actuators), or they

can be of different nature as desired accuracy in product specifications or economical effort, et cetera.

At the core of any MPC implementation there is a model of the process that has to be controlled. Typically the model is a linear, in discrete time model, and can be written as,

$$\begin{aligned} x(t+1) &= \boldsymbol{A}x(t) + \boldsymbol{B}u(t) + v(t), \\ y(t) &= \boldsymbol{C}x(t) + w(t). \end{aligned} \tag{2.3}$$

Here $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the controlled input signal, $y(t) \in \mathbb{R}^p$ the measured output, and $v(t)$ and $w(t)$ are stationary, zero-mean, white signals commensurate with $x(t)$ and $y(t)$.
The MPC makes use of a model of the system and the measured data from the system to predict the system's future output signals, as a function of future input signals. The predicted inputs and outputs are used in an objective function, that penalizes undesired behavior, and it is minimized with respect to the future input sequence. When an optimal input trajectory is obtained, just the first element of the sequence is applied to the system. The procedure is then repeated at the next time step.

A key feature of MPC is that constraints on states, inputs and outputs can easily be taken into account using constrained optimization methods. The control input is computed, optimizing an objective function over a future *prediction horizon*. The *prediction horizon*, denoted N_y, defines the number of samples of the output that are predicted and the *control horizon*, denoted N_u, defines the number of samples of the input that are used in the optimization algorithm. The predicted behaviour of the system, that is $\hat{y}(t+i|t)$ for $i = 1, \cdots N_y$, depends on the assumed input trajectory $\hat{u}(t+i|t)$ for $i = 0, \cdots, N_u - 1$, that is used over the control horizon. At time t it is assumed to collect the measurement $y(t)$ and the input $\hat{u}(t|t)$ is computed at the same time sample. Once a future input trajectory has been chosen, only the first element of this it is applied to the system, that is $u(t) = \hat{u}(t|t)$. In the next time sample, a new measurement of the output is collected and the whole cycle of prediction of output and evaluation of the input trajectory is repeated. The length of the prediction and control horizons remain the same in each iteration, but they are shifted one time step ahead. This control technique is often called a *receding horizon* strategy.

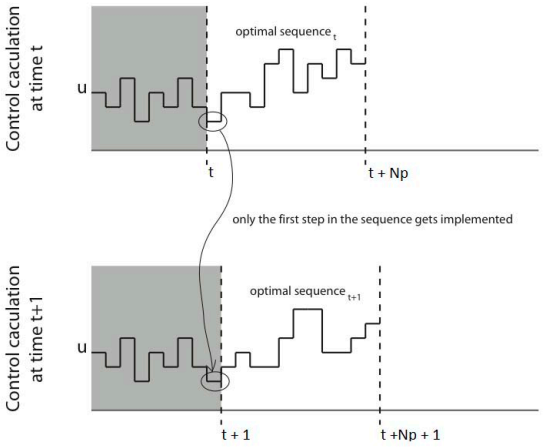

Figure 2.1: The receding horizon idea, [1]. In the upper figure at time t an input trajectory is evaluated (here p correspond to control horizon N_u) but only the first element is applied in the system as shown in the lower figure. At time $t+1$ a new optimal control is evaluated.

A common choice for the objective function used in MPC is [16]

$$J(t) \;=\; \sum_{i=0}^{N_y} \parallel \hat{y}(t+i|t) - r(t+i) \parallel^2_{\boldsymbol{Q}_y} + \sum_{i=0}^{N_u} \parallel \Delta\hat{u}(t+i) \parallel^2_{\boldsymbol{Q}_u} \tag{2.4}$$

where $\hat{y}(t+i|t)$ is the predicted output, $r(t)$ is the reference and $\Delta\hat{u}(t+i) = \hat{u}(t+i) - \hat{u}(t+i-1)$ is the increment of the input at time t. The norm $\parallel x \parallel_A$ is equal to $\sqrt{x^T A x}$. The weighting matrices $\boldsymbol{Q}_y$ and $\boldsymbol{Q}_u$, which are usually diagonal matrices, are tunable parameters.
The general optimization problem in MPC can be formulated as

$$\begin{aligned} \underset{u(t),u(t+1),\cdots,u(t+N_u)}{\text{minimize}} \quad & J(t) \\ \text{subject to} \quad & \hat{y}(t), \hat{y}(t+1), \cdots, \hat{y}(t+N_y) \in \mathcal{Y} \\ & u(t), u(t+1), \cdots, u(t+N_u) \in \mathcal{U} \end{aligned} \tag{2.5}$$

where $\mathcal{Y}$ and $\mathcal{U}$ represent the constraint sets for outputs and inputs respectively. The MPC problem as presented above can be formulated as a *Quadratic Problem* (QP) for which highly reliable and efficient solvers are available. The design parameters of the MPC are

- the model used in the MPC (matrices A, B and C);
- prediction and control horizon N_p and N_u;
- weighting matrices Q_y and Q_u;

- constraints on inputs, u_{min}, u_{max} (which are usually given by the system, e.g. saturations in actuators);

- constraints on outputs, y_{min}, y_{max} (which are usually imposed, e.g. by safety limits).

Most important to ensure good performance of the MPC is to have a model that describe with high accuracy the important properties of the system that has to be controlled. Plant-model mismatch can cause constraints violation or even instability in reference tracking applications. If a nonlinear system has to be controlled with MPC a linearized model, around an equilibrium point, has to be provided.

2.3 System identification

System identification is the process of constructing models of dynamic systems from experimental data. The goal is to find a model $\mathcal{M}$ belonging to a given class of parametric models that describes the relevant properties of the studied system $\mathcal{S}$. Identification experiments can be performed in open loop or in closed loop. In open loop identification there is no feedback control of the system during the identification experiment while in closed loop we have feedback control during the identification experiment. In this thesis all identification experiments are performed in open loop.

In the context considered in this thesis, the systems we want to identify are linear time-invariant, asymptotically stable multivariate systems on the form

$$\mathcal{S}: \quad \begin{aligned} x(t+1) &= \boldsymbol{A}x(t) + \boldsymbol{B}u(t) + v(t) \\ y(t) &= \boldsymbol{C}x(t) + w(t) \end{aligned} \tag{2.6}$$

with known input $u(t)$ and measured output $y(t)$. The noises $v(t)$ and $w(t)$ are white, zero-mean stationary processes with covariance matrices Λ_v and Λ_w. They are called *process noise* and *measurement noise*, respectively. Different identification methods are possible for finding the model $\mathcal{M}$. Such methods can be parametric or non-parametric and can be performed in both time domain or frequency domain. In this thesis we consider a parametric time domain

identification method. The model structure is

$$\mathcal{M}(\theta): \quad \begin{aligned} \hat{x}(t+1,\theta) &= \boldsymbol{A}(\theta)\hat{x}(t,\theta) + \boldsymbol{B}(\theta)u(t) + \boldsymbol{K}(\theta)e(t) \\ y(t,\theta) &= \boldsymbol{C}(\theta)\hat{x}(t,\theta) + e(t), \end{aligned} \tag{2.7}$$

where $\theta \in \mathbb{R}^n$ represents the unknown parameter vector to be identified and $e(t)$ is a zero-mean, white process with covariance matrix Λ_e. It is assumed that the model class $\mathcal{M}(\theta)$ include the true system. That is, there exists a true parameter vector θ_0 such that $\mathcal{S} = \mathcal{M}(\theta_0)$. The model structure in (2.7) is in innovation form. That it is not restrictive since it is always possible to transform a state space representation to the innovation form. Indeed a model on the same form as (2.6) can be rewritten in innovation form using spectral factorization. We must solve the Riccati equation

$$\begin{aligned} P(\theta) = A(\theta)P(\theta)A^T(\theta) + \Lambda_v(\theta) - [A(\theta)P(\theta)C^T(\theta) + S(\theta)] \\ \times [C(\theta)P(\theta)C^T(\theta) + \Lambda_w(\theta)]^{-1}[C(\theta)P(\theta)A^T(\theta) + S^T(\theta)]. \end{aligned}$$

Where $S(\theta) = Evw^T$. The solution $P(\theta)$ is a symmetric positive definite matrix. With $P(\theta)$ one can compute the Kalman gain

$$K(\theta) = [A(\theta)P(\theta)C^T(\theta) + S(\theta)][C(\theta)P(\theta)C(\theta)^T + \Lambda_w(\theta)]^{-1},$$

thus obtaining a model of the same form as (2.7). The estimates of θ are found using the prediction error method [15], this method and the properties of the resulting estimates are summarized next.

Prediction error method

Given a model $\mathcal{M}(\theta)$ belonging to a parametric class of models and a sequence of input-output data

$$y^N = \{y(t); t = 1, 2, ..., N\}, \qquad u^N = \{u(t); t = 1, 2, ..., N\},$$

this method can be summarized as follows [17],

1. For a general value of θ the one step ahead predictor is constructed. If a model of the form (2.7) is used, the one step ahead predictor is given by

[15]

$$\begin{aligned} \hat{x}(t+1,\theta) &= \boldsymbol{A}(\theta)\hat{x}(t,\theta) + \boldsymbol{B}(\theta)u(t) + \boldsymbol{K}(\theta)(y(t) - \hat{y}(t,\theta)), \\ \hat{y}(t,\theta) &= \boldsymbol{C}(\theta)\hat{x}(t,\theta). \end{aligned} \tag{2.8}$$

2. The *prediction errors* are constructed

$$\varepsilon_\theta(t) = y(t) - \hat{y}(t,\theta), \qquad t = 1, 2, ...N.$$

3. The parameter estimate is found minimizing a cost function represented for example by

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon_\theta(t)^2 \tag{2.9}$$

minimizing (2.9). The parameter vector estimate is obtained as

$$\hat{\theta}_N = \underset{\theta}{\operatorname{argmin}}\, V_N(\theta) \tag{2.10}$$

4. The variance of innovation $\lambda = var\{e(t)\}$ estimate is obtained evaluating the (2.9) at the parameter estimate vector $\hat{\theta}_N$

$$\hat{\lambda}_N = V_N(\hat{\theta}_N)$$

To determine how good this method is, we look at its asymptotic properties. That is, when the number of samples used in the identification experiment goes to infinity.
As stated in [15], we assume that the (true) process can be described by a model $\mathcal{M}(\theta)$. Under mild assumptions on the data, and model, and the cost function chosen as in (2.9), then the PEM estimator is consistent, or in other terms,

$$\lim_{N\to\infty} \hat{\theta}_N = \theta_0 \qquad \text{w.p.1} \tag{2.11}$$

Furthermore the asymptotic distribution is given by [15]

$$\sqrt{N}(\hat{\theta}_N - \theta_0) \in \mathcal{N}(0, \mathbf{P}) \quad as \quad N \to \infty, \tag{2.12a}$$

$$\mathbf{P} = \left[E\left\{\psi(t,\theta_0)\Lambda_e^{-1}\psi^T(t,\theta_0)\right\}\right]^{-1}, \tag{2.12b}$$

$$\psi(t,\theta_0) = \left.\frac{d\hat{y}(t)}{d\theta}\right|_{\theta=\theta_0}. \tag{2.12c}$$

The $\mathcal{X}^2$-distribution can be used to describe the estimates convergence. In fact, if we consider a variable $y \sim \mathcal{N}(\mu, \Sigma)$ with mean $\mu \in \mathbb{R}^m$ and covariance matrix $\Sigma \in \mathbb{R}^{m\times m}$ positive definite, then it holds that

$$(y-\mu)^T\Sigma^{-1}(y-\mu) \sim \mathcal{X}^2(m),$$

where the number of degrees of freedom of the $\mathcal{X}^2$ distribution, is given by the dimension of y. An α-level *confidence ellipsoid* (that we call the *identification ellipsoid*) for the estimated parameters is given by [8]

$$\mathscr{E}_{SI}(\alpha) = \left\{\theta : [\theta-\theta_0]^T\mathbf{P}^{-1}[\theta-\theta_0] \le \frac{\mathcal{X}_\alpha^2(n)}{N}\right\}. \tag{2.13}$$

The constant $\mathcal{X}_\alpha^2(n)$ is the α-percentile of the $\mathcal{X}^2$ distribution with n degrees of freedom. This means that if the number of samples N is sufficiently large the identification ellipsoid will contain $\hat{\theta}_N$ with probability α. The confidence ellipsoids will prove to be useful in the optimal input design.

2.4 Identification with a control objective

A controller based on a model of the system will achieve good performance if the accuracy of the identified model used by the controller (in the important directions of the parameter space) is high. The concept of important direction will be clarified in Section 3.3.1. The objective of optimal input design is to deliver a model that when used in the intended application will give acceptable performance. To have a measure of degradation of control performance, and to be able to define acceptable performance in the intended application, an *application cost* function is used, see [8]. Since the models considered are parametrized by the vector θ, the application cost becomes a function of θ. If an exact mathematical model of the true system was available, that is, if θ_0 was

known, the desired performance would be obtained. However, when the model does not correspond to the true system, the mismatch can cause a degradation in the controller performance.

Definition 2.4.1. (Application cost, [12]) If $\theta \in \mathbb{R}^n$ is a parameter vector corresponding to the model $\mathcal{M}(\theta)$, and $\mathcal{S} = \mathcal{M}(\theta_0)$, the function

$$V_{app}(\theta) : \mathbb{R}^n \to \mathbb{R}^+, \tag{2.14}$$

is an *application cost* if it has the following properties,

$$V_{app}(\theta_0) = 0, \qquad V_{app}^{'}(\theta_0) = 0, \qquad V_{app}^{''}(\theta_0) \succeq 0, \tag{2.15}$$

The application cost gives a scalar number that shows how much performance degrades when a parameter vector θ is used, instead of the true parameter vector θ_0 in the intended application. In any application there is a maximal allowed performance degradation, this gives an upper limit of the application cost that we can express on the form

$$V_{app}(\theta) \leq \frac{1}{\gamma} \tag{2.16}$$

for some real-valued positive constant γ. This bound gives a set of parameters that correspond to acceptable application performance.

Definition 2.4.2. (Application set, [12]) If V_{app} is an application cost and $\gamma \in \mathbb{R}_+$, the *application set* is defined as

$$\Theta_{app}(\gamma) = \left\{ \theta : V_{app}(\theta) \leq \frac{1}{\gamma} \right\}. \tag{2.17}$$

This leads to the idea that the objective of applications oriented system identification should be to deliver parameter estimates that belong to the application set.

2.5 Applications oriented optimal input design

As stated in [12], the objective of applications oriented optimal input design is to find an input that with high probability α, results in acceptable parameters θ, that satisfy condition (2.16), while at the same time minimizing the experimental effort of the identification experiment. It is possible to formulate this problem as follows

$$\begin{aligned} \underset{\text{input}}{\text{minimize}} \quad & \text{Experimental effort} \\ \text{subject to} \quad & P\left\{\theta \in \Theta_{app}(\gamma)\right\} \geq \alpha. \end{aligned} \tag{2.18}$$

Meaning, that the aim of the application optimal input design the experimental effort is to find the input that minimize an experimental effort, while the estimates are acceptable parameters with high probability α. In the optimal input design problem it is necessary to quantify the experimental effort. Quantifying experimental effort is not obvious but some common possibilities are to look at

- experiment length, N,
- input power i.e. $var\left\{u\right\}$,
- input energy i.e. $N \cdot var\left\{.u\right\}$.

In the rest of the thesis focuses on input power. The objective of the optimization problem (2.18) is defined as [13]

$$\text{trace}\left(\frac{1}{2\pi}\int_{-\pi}^{\pi}\Phi_u(\omega)d\omega\right) \tag{2.19}$$

To see the way the input comes into the system identification problem, as a design variable, it is useful to have a look at the frequency domain expression of $\boldsymbol{P}^{-1}$. By Parseval's theorem, $\boldsymbol{P}^{-1}$ is given by the following lemma [12].

Lemma 2.5.1. *In open-loop identification, the inverse covariance matrix* $\boldsymbol{P}^{-1}$ *in (2.13) is an affine function of the input spectrum* Φ_u *given by*

$$
\begin{aligned}
\boldsymbol{P}^{-1} &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \Gamma_1(e^{j\omega})\Lambda_e^{-1} \otimes \Phi_u(w)\Gamma_1^H(e^{j\omega})d\omega \\
&\quad + \frac{1}{2\pi}\int_{-\pi}^{\pi} \Gamma_2(e^{j\omega})\Lambda_e^{-1} \otimes \Lambda_e(\omega)\Gamma_2^H(e^{j\omega})d\omega, \\
\Gamma_1(e^{j\omega}) &= \begin{bmatrix} vec\ F_u^1 \\ \vdots \\ vec\ F_u^n \end{bmatrix}, \qquad \Gamma_2(e^{j\omega}) = \begin{bmatrix} vec\ F_e^1 \\ \vdots \\ vec\ F_e^n \end{bmatrix} \\
F_u^i &= H(q,\theta)\frac{\partial G(q,\theta)}{\partial \theta_i^T}\bigg|_{\theta=\theta_0}, \qquad i = 1,\ldots,n, \\
F_e^i &= H(q,\theta)\frac{\partial H(q,\theta)}{\partial \theta_i^T}\bigg|_{\theta=\theta_0}, \qquad i = 1,\ldots,n, \\
G(q,\theta) &= \boldsymbol{C}(\theta)(q\boldsymbol{I} - \boldsymbol{A}(\theta))^{-1}\boldsymbol{B}(\theta), \\
H(q,\theta) &= \boldsymbol{C}(\theta)(q\boldsymbol{I} - \boldsymbol{A}(\theta))^{-1}\boldsymbol{K}(\theta) + \boldsymbol{I}.
\end{aligned}
\tag{2.20}
$$

Where θ_i *represent the i-th component of the vector* θ*. Furthermore vec* $\boldsymbol{X}$ *is the row vector with rows of* $\boldsymbol{X}$ *stacked next to each other.*

A proof of this Lemma can be found in [2]. According to Lemma 2.5.1 and expression (2.20), it turns out that, in open loop identification, $\boldsymbol{P}^{-1}$ is an affine function of the input spectrum Φ_u. Consequently by a linear parametrization of the input spectrum, input spectrum constraints can be formulated as LMIs, for details see [9].

Approximation of the application set

It is not guaranteed that in the problem formulation (2.18) the application set, defined by $V_{app}(\theta)$ and γ, is convex. But it is possible to use an approximate description of $\Theta_{app}(\gamma)$ instead. Two different possibilities to do it are presented in [12]:

- ellipsoidal approximation,
- scenario based approach.

Ellipsoidal approximation

Using a second order Taylor expansion of $V_{app}(\theta)$, this can be approximated around θ_0. According to Definition (2.4.1), $V_{app}(\theta_0) = V'_{app}(\theta_0) = 0$, then

$$\begin{aligned} V_{app}(\theta) &\approx V_{app}(\theta_0) + [\theta - \theta_0]^T V'_{app}(\theta_0) + \frac{1}{2} [\theta - \theta_0]^T V''_{app}(\theta_0) [\theta - \theta_0] \\ &= \frac{1}{2} [\theta - \theta_0]^T V''_{app}(\theta_0) [\theta - \theta_0] . \end{aligned} \tag{2.21}$$

Using expression (2.21) in (2.16), the inequality that defines the allowed performance degradation can be approximated by

$$[\theta - \theta_0]^T V''_{app}(\theta_0) [\theta - \theta_0] \leq \frac{2}{\gamma}. \tag{2.22}$$

Meaning, that the application set can be approximated by the ellipsoid

$$\Theta_{app}(\gamma) \approx \mathscr{E}_{app} = \left\{ \theta : [\theta - \theta_0]^T V''_{app}(\theta_0) [\theta - \theta_0] \leq \frac{2}{\gamma} \right\}. \tag{2.23}$$

The approximate application set $\mathscr{E}_{app}$ is called the *application ellipsoid*, [12]. In [12] it is also pointed out that for large values of γ (that is for high performance demands) the approximation will be better than for small values. An example of an application ellipsoid is provided in Figure 2.2. In this thesis the ellipsoidal approximation is used.

Scenario based approach

A different approximation is introduced in [14], in an experiment design context. The approximation is obtained by randomly selecting parameters from $\Theta_{app}(\gamma)$ to represent the set. Thus, the approximated application set is simply given by the selected parameters. The more parameters that are selected, the more accurate the approximation of $\Theta_{app}(\gamma)$ is. This method will not be explained more in detail, since in this thesis the ellipsoidal approximation will be used instead.

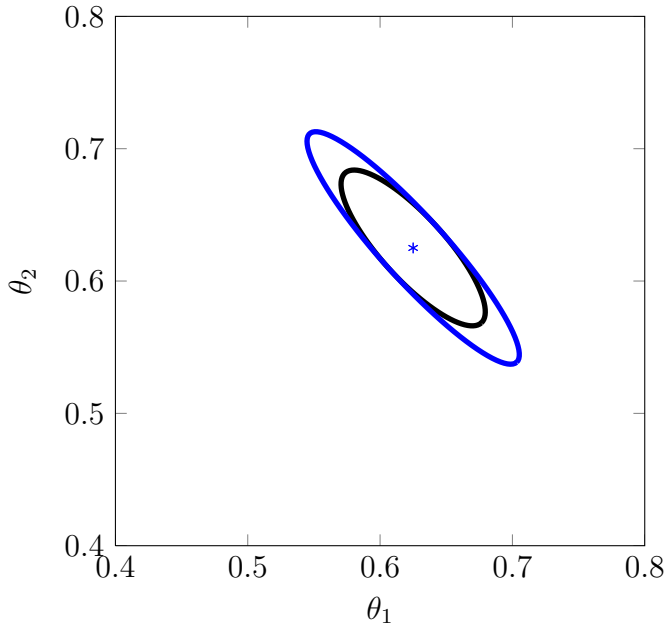

Figure 2.2: When the parameter vector has only two components, the application ellipsoid can be represented in a 2-D figure (Outer ellipse, represented in blue). The ellipse colored in black is the identification ellipse. The center of the ellipse is the parameter θ_0

Approximate optimization problem

Using the ellipsoidal approximation, the optimization problem (2.18) can be written as

$$\begin{aligned} \underset{\Phi_u}{\text{minimize}} \quad & \text{trace} \left(\tfrac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_u(\omega) d\omega \right) \\ \text{subject to} \quad & \mathscr{E}_{SI} \subseteq \mathscr{E}_{app} \\ & \Phi_u(\omega) \succeq 0 \quad \text{for all } \omega \end{aligned} \tag{2.24}$$

Where the constraint on the input spectrum in (2.24) is an infinite dimensional constraint which is possible to rewrite as an LMI constraint using finite dimensional parametrization of input spectrum and the KYP-lemma, see [12] for more details. The optimization problem (2.24) can be reformulated as a semi-definite program and solved efficiently, see e.g. [16].

3

Experiment design for model predictive control

3.1 Application cost

An application cost is a tool that permits to quantify how good is an estimated model, if used in a specific application. As seen in Section 2.3 the model structure of our estimated models is fixed. Hence, evaluating the application cost for a parameter vector θ, quantifies degradation of performance when that parameter is used. Furthermore, application cost shows, in which directions, in the parameter space, the performance is more sensitive to parameters variations. The analysis gives an insight into which parameters, or combination of them, that are more important to estimate with high accuracy. As stated in [13], by calculating the difference between the output of the system controlled by an MPC, based on a model using $\theta \neq \theta_0$ and one based on θ_0, denoted $y(t, \theta_0)$ and $y(t, \theta)$ respectively, a reasonable application cost for the MPC case is given by

$$V_{app}(\theta) = \frac{1}{N} \sum_{t=1}^{N} \| \, y(t, \theta_0) - y(t, \theta) \, \|^2 \, . \tag{3.1}$$

This function has the desired properties defined in Definition 2.4.1. Usually it is unlikely that one can know the value of θ_0 or try MPCs with different parameter vectors directly on the true system. Hence in [13] an approximate application cost, $\tilde{V}_{app}$, is introduced. The true output of the system, $y(t, \theta_0)$, is replaced by the output of a linear model that use an estimated parameter vector $\hat{\theta}$. This gives,

$$\tilde{V}_{app}(\theta, \hat{\theta}) = \frac{1}{N} \sum_{t=1}^{N} \parallel y(t, \hat{\theta}, \hat{\theta}) - y(t, \theta, \hat{\theta}) \parallel^2 . \tag{3.2}$$

Where the second argument in (3.2) represents the parameter vector of the model used by the MPC. The third argument is the parameter vector of the linear model replacing the true system.
As seen in Section 2.4, to define an application set, it is necessary to set an upper limit of the application cost, that is to choose a value for γ. The choice is highly application dependent. In [13], for reference tracking application using an MPC, allowing for a level of 1 % of degradation in the performance, it is chosen as

$$\gamma = \frac{100}{V(\theta_0)}. \tag{3.3}$$

Where

$$V(\theta_0) = \frac{1}{N} \sum_{t=1}^{N} \parallel y(t, \hat{\theta}, \hat{\theta}) - r(t) \parallel^2 . \tag{3.4}$$

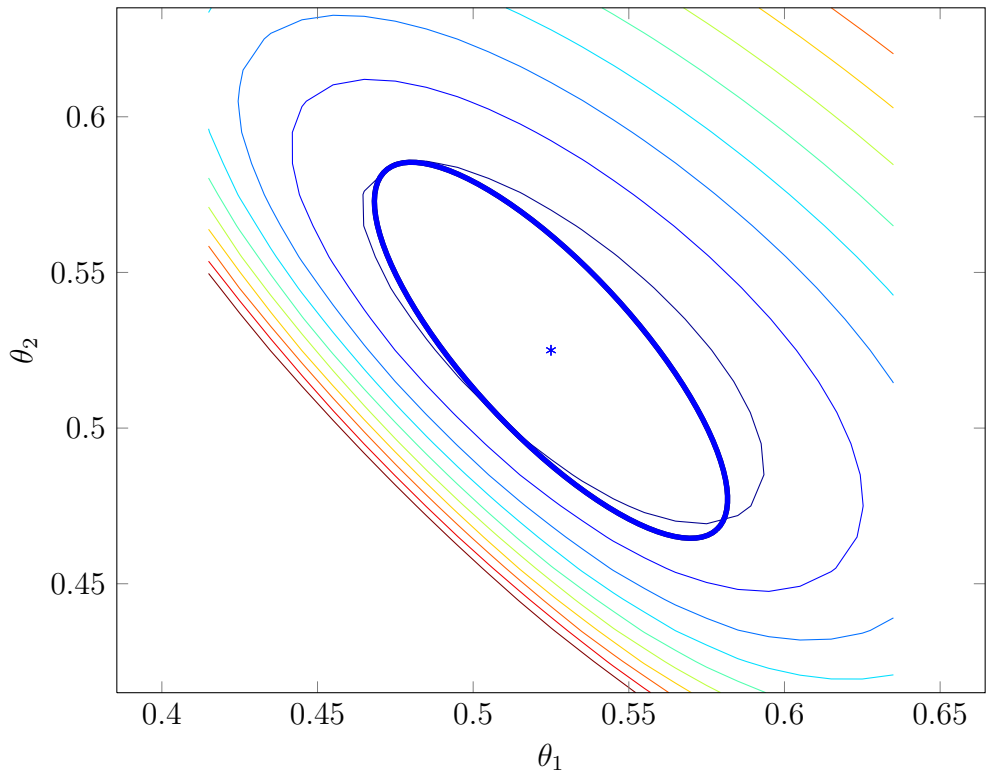

Figure 3.1: Level curves of $V_{app}(\theta)$ and ellipsoidal approximation of $\tilde{V}_{app}(\theta, \hat{\theta})$ centered in θ_0. The offset in the direction of the application ellipsoid, can be explained by the fact that, to evaluate $\tilde{V}_{app}(\theta, \hat{\theta})$ a linear approximation of the true model is used instead. It is possible to notice, that the level curves are not ellipses and are not centered in θ_0.

3.2 Identification experiments

In all simulations that will be presented in Chapter 5 the true system is represented by a nonlinear model. The nonlinear model is linearized around an equilibrium point and discretized. This gives a model $\mathcal{M}(\theta_0)$, in the same form as in (2.7). All identification experiments are performed in open loop. The models identified have a fixed structure, hence a grey box identification is performed using the IDGREY and PEM commands of MATLAB.

3.2.1 Identification algorithm

A complete application oriented identification method is presented in [13].

Algorithm

Step 0 Obtain an initial estimate of the model parameters using a white noise input sequence in the first identification experiment.

Step 1 Find the application cost based on simulations of the model with the parameter estimates.

Step 2 Design the optimal input signal based on the application cost and parameter estimates.

Step 3 Find a new estimate of the model parameters using the optimal input signal in the system identification experiment.

As stated in [13], if a good initial guess of the parameters is available, for example coming from physical insight of the process, this guess can replace the initial estimation in Step 0. Furthermore, in [13] it is pointed out that the algorithm can be iterated. In fact the estimate from Step 3 can be used, instead of the initial guess, in Step 1 and 2.
The algorithm is also represented in a block diagram in Figure 3.2. The non linear model is excited by a white Gaussian noise n with low variance, around an equilibrium point, to find an initial estimate $\hat{\theta}$. This initial estimate is used by an MPC to control a linear model $\mathcal{M}(\hat{\theta})$ that should represents the linearized model of the true system. This gives the trajectory $y(\hat{\theta}, \hat{\theta})$, see Section 3.1. Another MPC using a parameter $\theta \neq \hat{\theta}$ is then used to control the same linear model $\mathcal{M}(\hat{\theta})$. This gives the trajectory $y(\theta, \hat{\theta})$, see Section 3.1. These two

trajectory are used to evaluate $\tilde{V}_{app}(\theta, \hat{\theta})$. Considering many simulations, for different values of θ, the hessian matrix $\tilde{V}''_{app}(\hat{\theta}, \hat{\theta})$ can be calculated. Based on the hessian matrix the optimal input design problem is solved. The resulting input design is optimal for $\mathcal{M}(\hat{\theta})$ but may not be optimal for the true nonlinear model.

In Chapter 4 this algorithm will be applied to a multivariable process of four interconnected tanks. In Chapter 5 simulations will be performed, both using optimal input and white noise input, with different system setups allowing for comparisons.

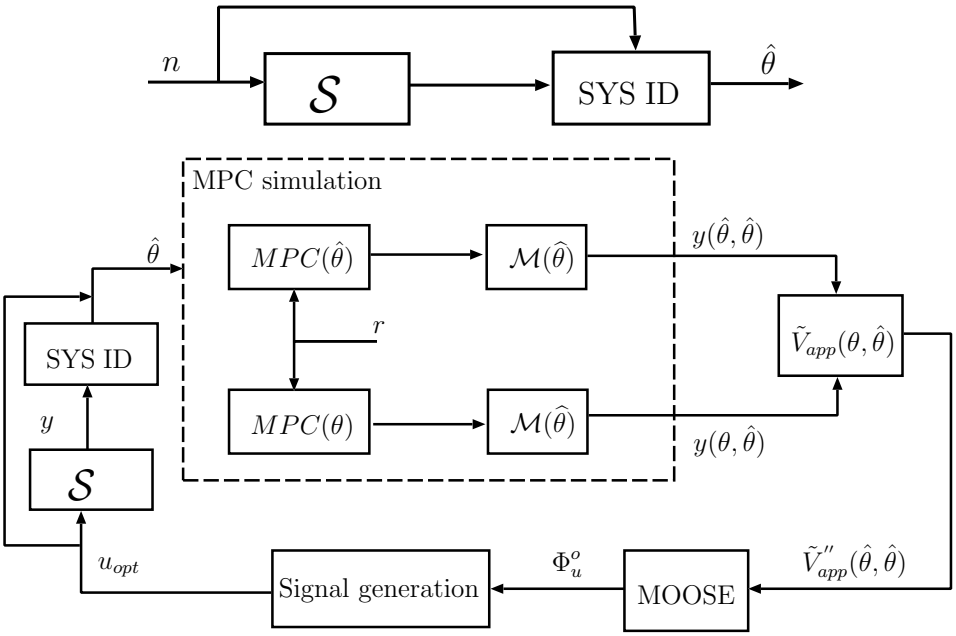

Figure 3.2: Identification Algorithm. An initial identification experiment using a white gaussian noise n gives an initial parameter vector $\hat{\theta}$ (upper figure). In the lower figure Step 1, 2 and 3 of the algorithm are schematically represented. The control performance are evaluated for different values of θ. This is done simulating a model of the system ($\mathcal{M}(\hat{\theta})$) controlled by an MPC using $\hat{\theta}$ and another MPC using θ. The output trajectories $y(\hat{\theta}, \theta)$ and $y(\theta, \theta)$ are used to evaluate the approximate application cost V''_{app}. This is used by the toolbox MOOSE to evaluate the optimal input spectrum Φ^o_u and the with that the optimal input u_{opt} can be realised. The latter is then used for a new identification experiment giving a new estimated parameter.

3.3 Sensitivity analysis

3.3.1 Important directions

When an estimated parameter vector is used, instead of the true one, the performance degrades. Variations of the estimated parameters from the true ones cause plant-model mismatch, thus giving worse performance. It is interesting to notice that not every variation, give the same performance degradation. As seen in Section 3.1, the application cost can be used, to get in which directions the performance is more sensitive to parameters' variations. The analysis is done considering the application ellipsoid $\mathscr{E}_{app}$. As defined in (2.23) the application ellipsoid is defined by using the hessian of the application cost. When the latter can not be evaluated (see Section 3.1), $\tilde{V}''_{app}(\theta, \hat{\theta})$ is used instead.
When we consider a two dimensional parameters vector, the application ellipsoid can be represented in a 2-D figure as an ellipse. The lengths of the semi-axes are given by $\frac{1}{\sqrt{\lambda_i}}$, where λ_i are the eigenvalues of $\tilde{V}''_{app}(\theta, \hat{\theta})$. The directions of the axis are given by the respective eigenvectors. An example of an application ellipsoid is represented in Figure 3.3. The two arrows represent the semi-axis of the ellipsoid. From Figure 3.3 one can easily understand, that just a little variation of the parameters in the direction of the smallest semi-axis (represented by the red arrow in Figure 3.3), can be tolerated to obtain the desired performance. While, in the direction of the largest semi-axis, a bigger variation of the parameters can be tolerated.
Summarizing, looking at the eigenvalues of $\tilde{V}''_{app}(\theta, \hat{\theta})$, one can understand which combinations of the parameters influence more the performance. Coefficients of these combinations of parameters are given by the respective eigenvectors.

3.3.2 Area

It is also interesting, to evaluate the volume of the application ellipsoid. This in the case that $\theta \in \mathbb{R}^2$ corresponds to the area of the ellipse. The general expression to evaluate the volume of an ellipsoid, is stated for example in [3]. Given an ellipsoid centered in the origin, defined in the following form,

$$\mathscr{E}_X = \left\{ z \mid z^T X^{-1} z \leq 1 \right\}, \tag{3.5}$$

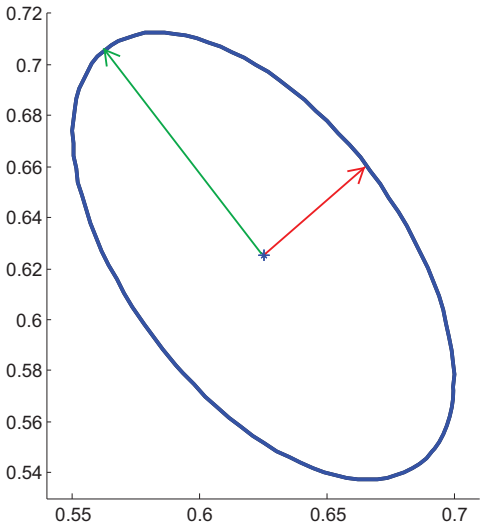

Figure 3.3: Application ellipsoid $\mathscr{E}_{app}$.

where $X = X^T \succ 0$, that is, X is symmetric and positive definite. The volume of the ellipsoid is proportional to $(detX^{-1})^{\frac{1}{2}} = \frac{1}{\sqrt{\lambda_1 \cdot \lambda_2 \cdots \lambda_n}}$, where n is the dimension of X.

The area will be used in Chapter 5 to compare application ellipsoids obtained using different MPC settings.

4

The quadruple water tanks process

4.1 Process description

The process is composed by four interconnected water tanks, two pumps and two valves, that divide the water flow in upper and lower tanks. The tanks are stacked one over each other in couples and numbered as schematically showed in Figure 4.1. Each tank has a hole at the bottom where water can flow out. The flow of the two upper tanks goes into the respective lower tank, while the flow of the two lower tanks goes in a common water container arranged below the tanks. The pumps are connected in such a way that pump 1 delivers water to tank 1 and 4, while pump 2 delivers water to tank 2 and 3. The process has some physical constraints, as described in Table 4.1. The water level in each tank is measured by pressure sensors at the bottom of the tank. The sensors are characterized by a conversion constant which is known to be $k_c = 0.2$ V/cm. The outputs of the process are the pressure sensor signals from the lower tanks. The inputs of the process are the voltages applied to the two pumps. Depending on the setting of the two valves one can decide the fraction of water that will be delivered to the upper tank and to the lower. The valve setting determines if the process is minimum phase or not.

Parameter	Limit	Description
$h_{i,max}$	25 cm	Maximum water level of tank i
$h_{i,min}$	0 cm	Minimum water level of tank i
$u_{j,max}$	15 V	Maximum voltage of pump i
$u_{j,min}$	0 V	Minimum voltage of pump i

Table 4.1: Physical constraints of the quadruple tank process.

Nonlinear model

The process is very well described by a system of nonlinear differential equations. One can express the variation of the volume V of water in each tank with the following expression

$$\frac{dV}{dt} = q_{in} - q_{out} \tag{4.1}$$

where q_{in} and q_{out} represent the total inflow and outflow of water respectively. The outflow of water of each tank can be found, using Torricelli's principle[1] and given the cross sectional area a of the outlet hole of the tank, to be

$$q_{out} = a\sqrt{2gx}, \tag{4.2}$$

where x is the level of water in the tank and $g = 981$ cm/s^2 is the gravitational constant.

The inflow of water to the upper tanks, comes from the flow generated from a pump. For the lower tanks, in addition to the pump flow, there is also the flow coming from the upper tanks. The outflow q of a pump is proportional to the applied voltage u, that is $q = ku$ where k is the constant of the pump. The flow q is then divided into the respective upper and lower tanks, according to the setting of the valves. Hence flows to upper and lower tanks are given by

$$q_L = \gamma ku, \quad q_U = (1-\gamma)ku, \quad \gamma \in [0,1] \tag{4.3}$$

where γ represents the setting of the valve that is connected to the pump. In (4.3) q_L denotes the flow to the lower tank and q_U is the flow to the upper tank. The measurement noises are modelled as zero mean white Gaussian noise. The measurement noises of each output are uncorrelated with each other.

[1] Torricelli's law states that the speed, v, of a fluid through a sharp-edged hole at the bottom of a tank filled to a depth h is the same as the speed that a body would acquire in falling freely from a height h, [20]

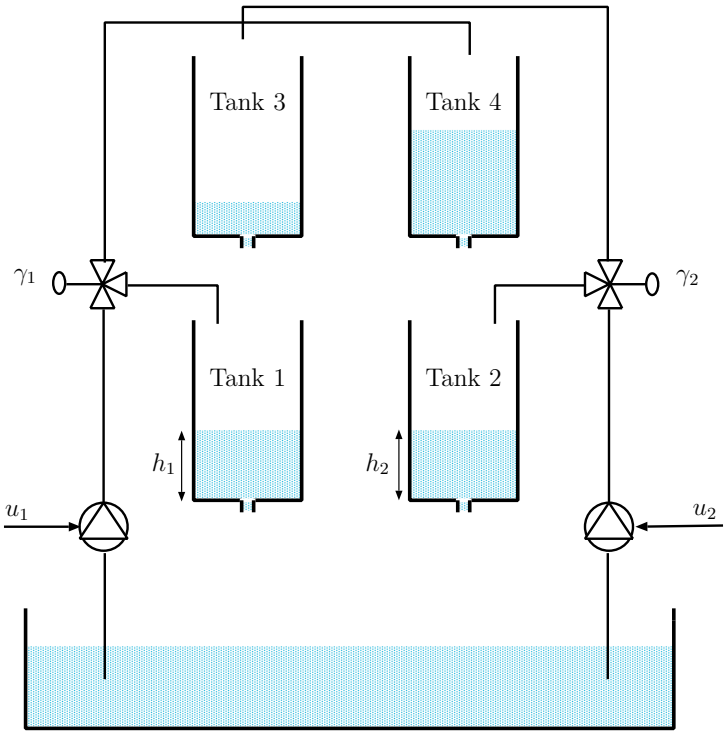

Figure 4.1: The quadruple water tanks process.

The previous expressions lead to the following system of nonlinear differential equations

$$\begin{aligned}
\frac{dx_1}{dt} &= -\frac{a_1}{A}\sqrt{2gx_1(t)} + \frac{a_3}{A}\sqrt{2gx_3(t)} + \frac{\gamma_1 k_1}{A}u_1(t),\\
\frac{dx_2}{dt} &= -\frac{a_2}{A}\sqrt{2gx_2(t)} + \frac{a_4}{A}\sqrt{2gx_4(t)} + \frac{\gamma_2 k_2}{A}u_2(t),\\
\frac{dx_3}{dt} &= -\frac{a_3}{A}\sqrt{2gx_3(t)} + \frac{(1-\gamma_2)k_2}{A}u_2(t),\\
\frac{dx_4}{dt} &= -\frac{a_1}{A}\sqrt{2gx_4(t)} + \frac{(1-\gamma_1)k_1}{A}u_1(t),\\
y_1(t) &= k_c x_1(t) + e_1(t),\\
y_2(t) &= k_c x_2(t) + e_2(t).
\end{aligned} \tag{4.4}$$

In (4.4), x_i is the water level in each tank, A is the cross sectional area of the tanks (assumed to be the same for all tanks). All the physical parameters used in the dynamic model are summarized in Table 4.2. From the nonlinear model (4.4) it is possible to obtain the mathematical expressions for the cross sectional

Parameter	Unit	Description
a_i	cm^2	Cross sectional area of water outlet hole of tank i
A	cm^2	Cross sectional area of the four tanks
γ_i	$[\cdot]$	Fraction of water flow of pump i that goes into lower tank
k_i	$\text{cm}^3/(\text{s}\cdot\text{V})$	Voltage to volumetric pump constant of pump i
k_c	V/cm	Water level to voltage proportionality constant of sensors

Table 4.2: Physical parameters of the quadruple tank process.

areas of the outlet holes. For a stationary working point (x,u) it holds [11],

$$\begin{aligned}
\frac{a_1}{A}\sqrt{2gx_1} &= \frac{\gamma_1 k_1}{A}u_1 + \frac{(1-\gamma_2)k_2}{A}u_2,\\
\frac{a_2}{A}\sqrt{2gx_2} &= \frac{\gamma_2 k_2}{A}u_2 + \frac{(1-\gamma_1)k_1}{A}u_1,\\
\frac{a_3}{A}\sqrt{2gx_3} &= \frac{(1-\gamma_2)k_2}{A}u_2,\\
\frac{a_4}{A}\sqrt{2gx_4} &= \frac{(1-\gamma_1)k_1}{A}u_1.
\end{aligned} \tag{4.5}$$

Rewriting expressions (4.5) we can obtain

$$\begin{aligned}
a_1 &= \frac{\gamma_1 k_1 u_1 + (1-\gamma_2)k_2 u_2}{\sqrt{2gx_1}},\\
a_2 &= \frac{\gamma_2 k_2 u_2 + (1-\gamma_1)k_1 u_1}{\sqrt{2gx_2}},\\
a_3 &= \frac{(1-\gamma_2)k_2 u_2}{\sqrt{2gx_3}},\\
a_4 &= \frac{(1-\gamma_1)k_1 u_1}{\sqrt{2gx_4}}.
\end{aligned} \tag{4.6}$$

Linear model

The MPC needs a discrete time, linear model of the process on the form (2.3). Therefore, the nonlinear dynamic model (4.4) will be linearized around a working point x^0 and u^0 and then discretized.

Linearization, made using first order Taylor expansion, gives

$$\begin{aligned}
\boldsymbol{A}_C &= \begin{bmatrix} -\tau_1 & 0 & \tau_3 & 0 \\ 0 & -\tau_2 & 0 & \tau_4 \\ 0 & 0 & -\tau_3 & 0 \\ 0 & 0 & 0 & -\tau_4 \end{bmatrix}, \quad \boldsymbol{B}_C = \begin{bmatrix} \frac{k_1\gamma_1}{A} & 0 \\ 0 & \frac{k_2\gamma_2}{A} \\ 0 & \frac{k_2(1-\gamma_2)}{A} \\ \frac{k_1(1-\gamma_1)}{A} & 0 \end{bmatrix}, \\
\boldsymbol{C}_C &= \begin{bmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{bmatrix}, \\
\tau_i &= \frac{a_i}{A}\sqrt{\frac{g}{2x_i^0}}
\end{aligned} \tag{4.7}$$

It is useful for the following analysis to also consider an expression of the transfer matrix of the system. The Laplace transform of (4.7) yields the transfer matrix of the system,

$$\begin{aligned}
&G(s) = C_C(sI - A_C)^{-1}B_C, \quad \begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = G(s) \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}, \\
&G(s) = \begin{bmatrix} \frac{\gamma_1 c_1}{1+s\tau_1} & \frac{1-\gamma_2 c_1}{(1+s\tau_3)(1+s\tau_1)} \\ \frac{(1-\gamma_1)c_2}{(1+s\tau_4)(1+s\tau_2)} & \frac{\gamma_2 c_2}{1+s\tau_2} \end{bmatrix},
\end{aligned} \tag{4.8}$$

where $c_i = \frac{\tau_i k_i k_c}{A}$ for $i = 1, 2$.
Discretization, using zero-order hold with a sampling rate of 1 Hz, gives

$$\boldsymbol{A} = e^{A_C}, \qquad \boldsymbol{B} = \int_0^1 e^{\boldsymbol{A}_C(1-t)}\boldsymbol{B}_C dt, \qquad \boldsymbol{C} = \boldsymbol{C}_C. \tag{4.9}$$

These matrices are used by the MPC to predict system outputs.

4.2 Valves setting and physical interpretation

Systems that are described by a transfer function

$$G(s) = \frac{N(s)}{D(s)}$$

that have zeros in the right half s plane, are called *non minimum phase* systems. On the other hand systems that have all zeros in the left half s plane, are

called *minimum phase* systems. The quadruple water tanks process permits to analyse both minimum phase and non minimum phase behaviors, by modifying the valves setting. Furthermore the relation between valves setting and position of the system's zeros has a straightforward physical interpretation.

The definition of zeros of a multivariable system is given by the following theorem [5].

Theorem 4.2.1. ***The zero polynomial*** *of $G(s)$ is the greatest common divisor for the numerators of the maximal minors of $G(s)$, normalized to have the pole polynomial as denominator. The* ***zeros*** *of the system are the zeros of the zero polynomial.*

Hence the zeros of the transfer matrix in (4.8) are the zeros of the numerator polynomial of the following [10],

$$\begin{aligned} \det\ G(s) \ &= \frac{c_1 c_2}{\gamma_1\gamma_2 \prod_{i=1}^{4}(1+s\tau_i)} \\ &\times \left[(1+s\tau_3)(1+s\tau_4) - \tfrac{(1-\gamma_1)(1-\gamma_2)}{\gamma_1\gamma_2}\right]. \end{aligned} \tag{4.10}$$

Thus for $\gamma_1, \gamma_2 \in (0,1)$ the transfer matrix $G(s)$ has two finite zeros. One of these zeros is always in the left half s plane, while the the other one, depending on values of γ_1, γ_2, can be in the left or right half plane. Thus, the system can be minimum phase or non minimum phase depending on the position of the moving zero. In [10], it is pointed out that the system is non minimum phase for

$$0 < \gamma_1 + \gamma_2 < 1,$$

and it is minimum phase for

$$1 < \gamma_1 + \gamma_2 < 2.$$

Expressions for the total flow of water that goes to upper and lower tanks can be given by

$$q_{lower} = q_1\gamma_1 + q_2\gamma_2 \qquad q_{upper} = (1-\gamma_1)q_1 + (1-\gamma_2)q_2. \tag{4.11}$$

Where $q_1 = k_1 \cdot u_1$ and $q_2 = k_2 \cdot u_2$. If we assume that $q_1 = q_2 = q$, then the expressions in (4.11) can be rewritten as follow,

$$q_{lower} = q(\gamma_1 + \gamma_2) \qquad q_{upper} = q\left[2 - (\gamma_1 + \gamma_2)\right]. \tag{4.12}$$

From (4.12) can be seen that $(\gamma_1 + \gamma_2)$ determines if the total flow of water to the upper tanks, is higher or lower compared to the total flow to the lower tanks. Hence, with the assumption that $q_1 = q_2 = q$, the system is non minimum phase if the total flow of water that goes into the upper tanks is higher than the the total flow that goes into the lower tanks. We can notice that if both flow ratios γ_i are big, most of the water flows directly into the lower tanks. If γ_i are small, water flows first to the upper tanks and after that into the lower tanks. In the latter case, pump 1 indirectly fills tank 2 and pump 2 indirectly fills tank 1. It is intuitively clear that it is easier to control the lower water levels if the water flows directly into lower tanks.

4.3 Water level control using MPC

An MPC controller is used to control water levels of the two lower tanks (tank 1 and 2 as defined in Figure 4.1). The MPC makes use of the linearized model of the plant around an equilibrium point (x^0, u^0), on the form (2.3), to predict the future outputs of the system, using the following formula

$$\hat{y}(t+k|t) = \boldsymbol{C}\boldsymbol{A}^k x(t) + \sum_{i=0}^{k-1} \boldsymbol{A}^{k-i-1}\boldsymbol{B}u(t+i). \tag{4.13}$$

One can see that in the previous formula the complete knowledge of the state $x(t)$ is needed. When there is no possibility to measure it or if just a part of it is available for measurement, a state estimator is needed. From the $\boldsymbol{C}$ matrix expression, see (4.9), can be seen that just two components of the state can be reconstructed from the outputs. A Kalman filter has to estimate the states not known.
In Figure 4.2 the diagram block of the system is represented. The cost function $J(t)$ minimized by the MPC is on the form as in (2.4), $v(t)$ and $w(t)$ are process and measurement noise respectively.

4.3.1 MPC weights

The weights, that we use to compute the cost function 2.4, are represented by diagonal matrices with the following form: $\boldsymbol{Q}_y = q_y I$ and $\boldsymbol{Q}_u = q_u I$. That is, both outputs and both inputs have respectively the same weights. The weights q_y and q_u have been set manually. Different combinations of the weights

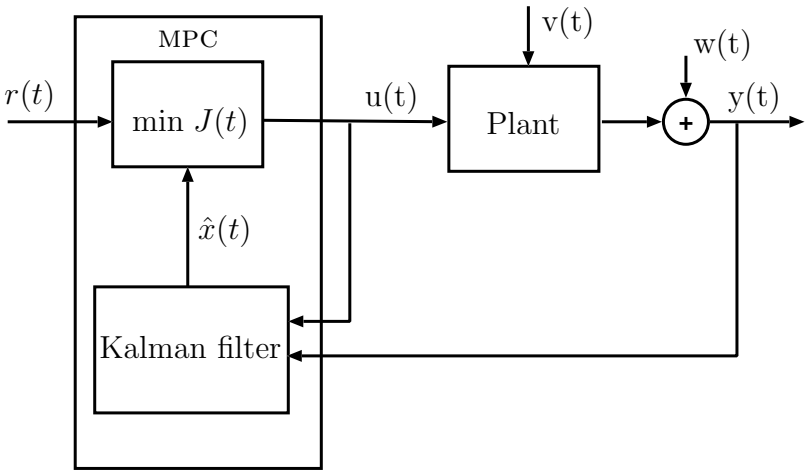

Figure 4.2: System and controller scheme.

give different response of the system. Thus, in Figure 4.3 we can see that the response of the system is faster and the input signal variation is sharper, when an higher weight is given to the outputs compared to the inputs. Figure 4.4, shows that when using higher weight on inputs instead, the response is slower but the input signals variation is smoother.

4.3.2 Reference

The reference trajectory consists of steps of 1 cm, around the equilibrium point. The reference signal is constructed in a suitable way to show if the outputs are coupled. For this reason when the reference of one output change, the reference for the other one is kept to the same value. The reference trajectory and the nominal response of the system, when the model perfectly describe the system is shown in Figure 4.3.

4.3.3 handling constraints

The inputs in all simulations are constrained to never go below 0.5 V, to ensure that there is always water in the tubes. If the tubes were drained, an additional delay would be introduced into the system, that is not captured by the models used. As has been described in Section 4.2, some values of γ_1, γ_2 make the system harder to control. As an example of this in Figure 4.4 a linear model with $\gamma_1 + \gamma_2 = 0.98$, is simulated. Figure 4.4 shows that the outputs are coupled and the constraints on the outputs can still be handled with no problems if the

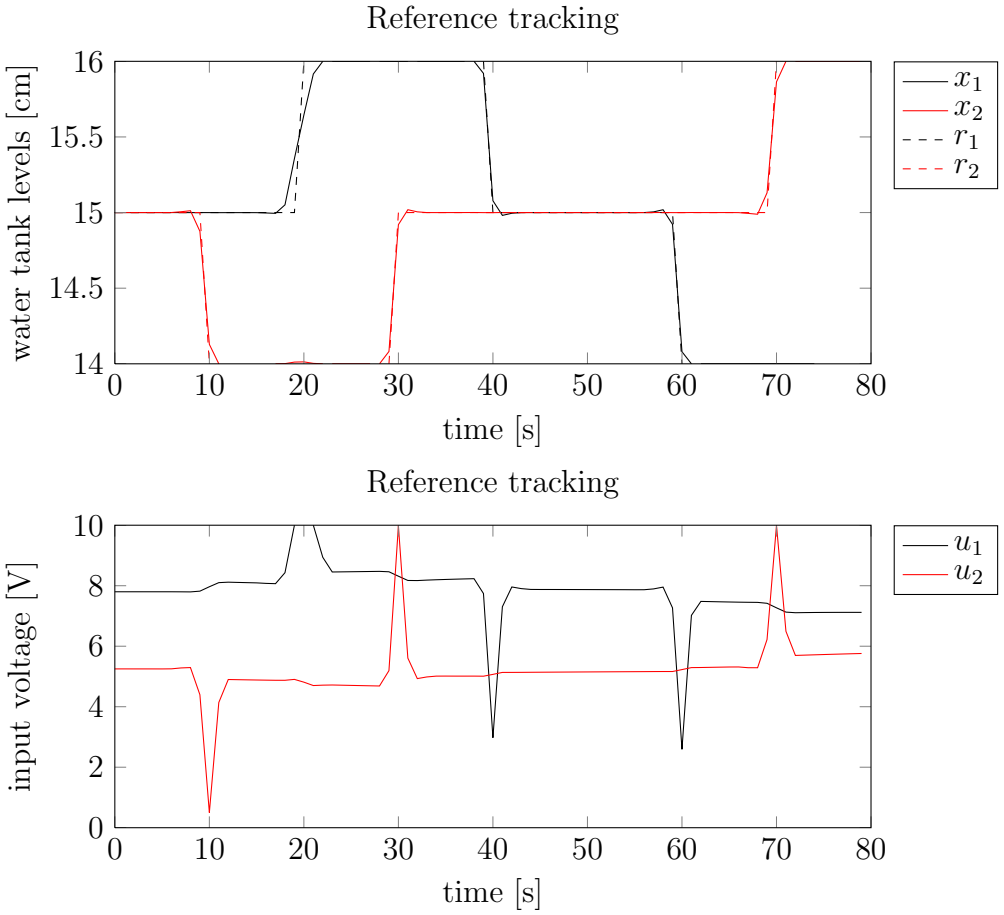

Figure 4.3: Nominal response of a linear model, with $\gamma_1 + \gamma_2 = 1.23$, controlled with a MPC using exactly the same model. Inputs are constrained to never be below 0.5 V and over 10 V. Outputs are constrained to be between 14 cm and 16 cm.

inputs are unconstrained. Figure 4.5 shows instead the response of the system when the inputs are constrained to be between 0.5 V and 10 V.

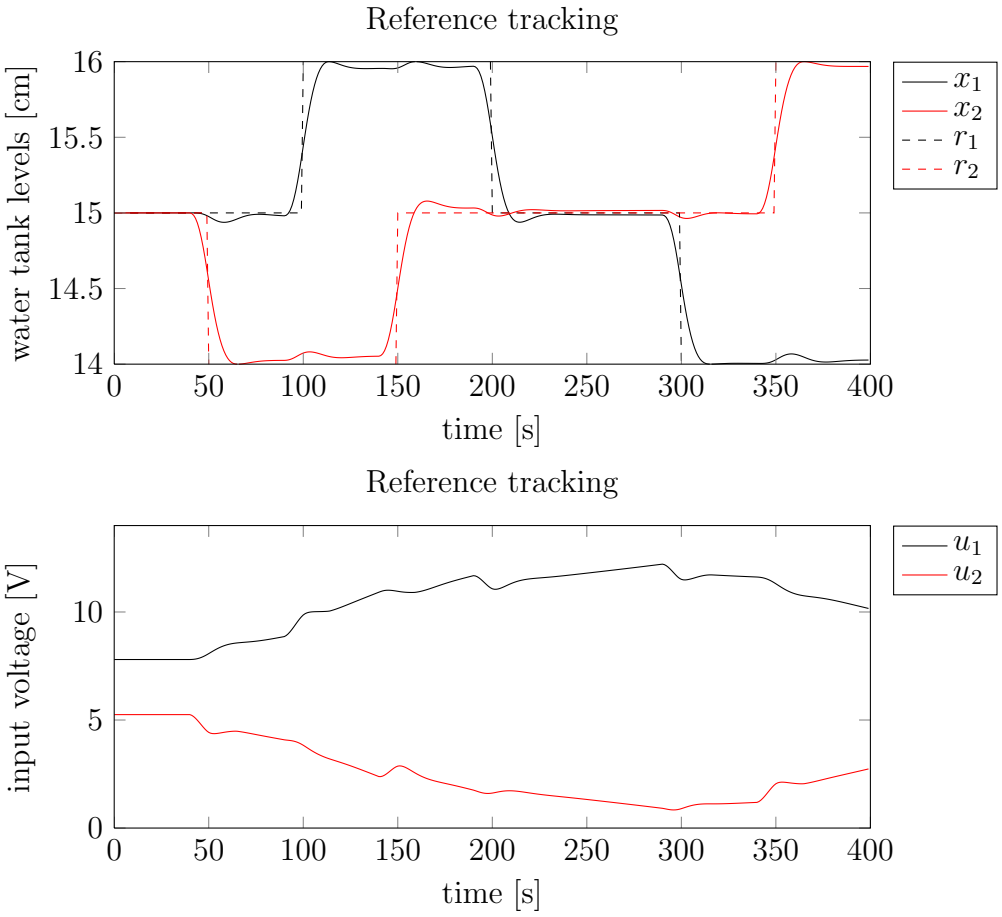

Figure 4.4: Nominal response of a linear model, with $\gamma_1 + \gamma_2 = 0.98$ controlled with a MPC using exactly the same model. Inputs are unconstrained. Outputs are constrained to be between 14 *cm* and 16 *cm*.

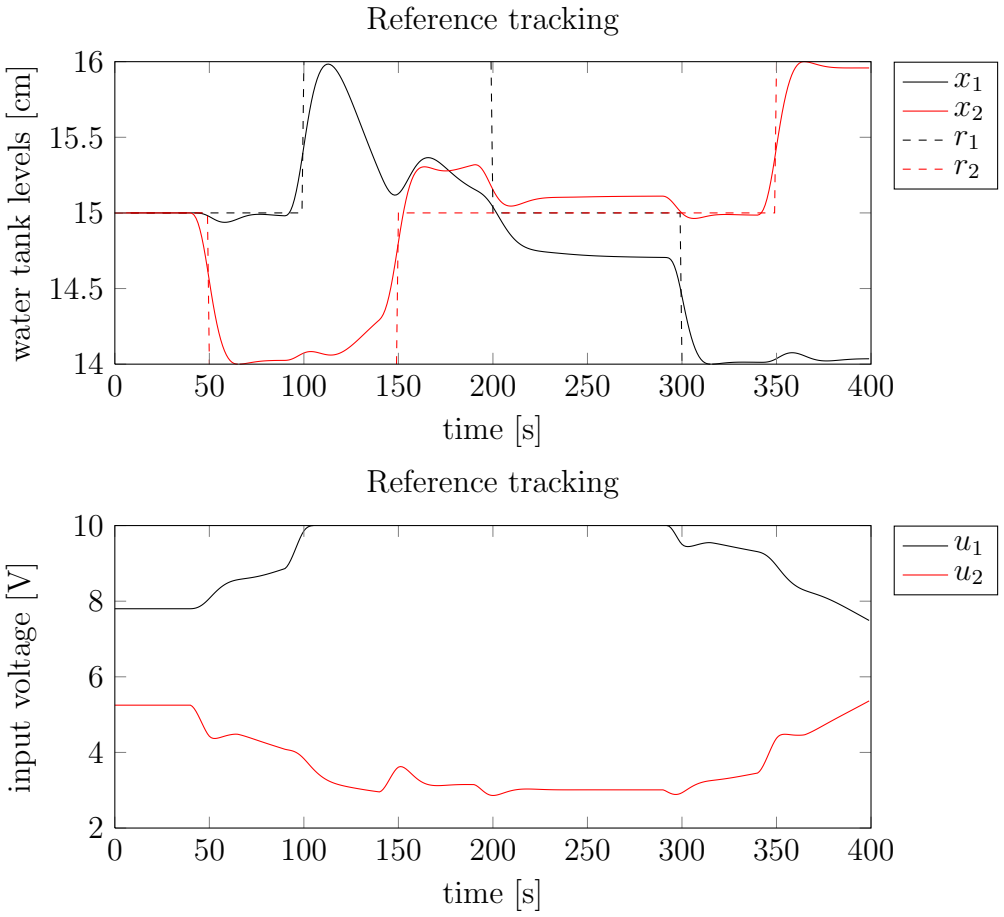

Figure 4.5: Nominal response of a linear model, with $\gamma_1 + \gamma_2 = 0.98$, controlled with a MPC using exactly the same model. Inputs are constrained to never be below 0.5 V and over 10 V. Outputs are constrained to be between 14 cm and 16 cm.

5
Simulations

In this Chapter are presented the results of the simulations performed with the quadruple water tank process described in Section 4.1. The results come with a brief description leaving for Chapter 6 a more detailed analysis and comparisons.

Nonlinear model

To perform simulations, we need to choose a model with fixed parameters that represents the true system. In all the simulations the nonlinear model (4.4) is considered to be the true system. The physical parameters of the nonlinear model are described in Table 4.2. In all simulations the set of physical parameters considered is described in Table 5.1. The estimated models, as described in Section 3.2, have a fixed structure. The parameter vector θ, consists of the physical parameters γ_1 and γ_2. The remaining physical parameters are the one considered in Table 5.1. The physical parameters can be obtained for example, with an initial long identification experiment.

The values of the sectional areas of tank's outlets, are obtained from expressions (4.6).

Table 5.1: Physical parameters

Parameter	Value
x_0	$[15, 15, 3, 12]$ cm
u_0	$[7.8, 5.25]$ V
γ_1	0.625
γ_2	0.625
A	15.52 cm^2
a_i	$\{0.17, 0.15, 0.11, 0.08\}$ cm^2
k_1	4.14 cm^3/(s·V)
k_2	4.14 cm^3/(s·V)
k_c	1 V/cm

MPC settings

As described in Chapter 2 MPC can be tuned, using weights and constraints. It is interesting to see, if different MPC settings, affect in some way the sensitivity analysis described in Chapter 3. Different MPC settings that we can consider are the following:

- no active constraints. That is, the inputs and outputs do not reach their maximal and minimal allowed values, during the prediction part of the optimization problem solved in the MPC (see Section 4.3),
- active constraints. That is, the inputs and outputs actually reach their maximal and minimal allowed values, during the prediction part of the optimization problem solved in the MPC,
- large weight on outputs compared to the weight on inputs. That is, the MPC punishes more variations on outputs, allowing for stronger control action,
- large weight on inputs compared to the weight on outputs. That is, the freedom of the action of the MPC is limited. This should increase the robustness of the MPC.

The MPC settings that are analysed in the sensitivity analysis, are summarized in Table 5.2. We call these particular settings *Scenarios*.

	N_y	N_u	q_y	q_u	y_{min}	y_{max}	u_{min}	u_{max}
Scenario 1	10	10	1	0.0001	0 cm	25 cm	0.5 V	15 V
Scenario 2	10	10	1	0.0001	14 cm	16 cm	0.5 V	10 V
Scenario 3	10	10	0.1	1	0 cm	25 cm	0.5 V	15 V
Scenario 4	10	10	0.1	1	14 cm	16 cm	0.5 V	10 V

Table 5.2: *Scenarios.*

Valve settings

As described in Section 4.2 depending on the value of $\gamma_1 + \gamma_2$, the system is minimum phase or non-minimum phase. In particular when $\gamma_1 + \gamma_2$ is close to 1, the system is shifting from minimum phase to non-minimum phase. Thus, it is interesting to compare the sensitivity analysis when the system is clearly minimum phase (and clearly non minimum phase), to the sensitivity analysis when the system is close to shifting between minimum phase and non-minimum phase. This is performed for each scenario.

Procedure

1. First we choose the value of the true parameter.

2. An initial identification experiment with 1000 samples are performed. The input used is a Gaussian white noise with covariance matrix $\Lambda_u = 0.1\boldsymbol{I}$.

3. A reference tracking simulation of the initial estimated model is performed. The parameter used in the MPC is the same used in the model. This trajectory is then used to define the approximate application cost $\tilde{V}_{app}(\theta)$, see Section 3.2.

4. The Hessian of the approximate application cost, $\tilde{V}''_{app}(\hat{\theta}, \hat{\theta})$, is evaluated.

5. The Hessian is then used by the toolbox MOOSE to design the optimal input spectrum that minimizes the input power. With the optimal input spectrum the optimal input signal u^* can be realized.

6. 100 identification experiments, using the optimal input u^*, are performed. For each identification experiment 600 samples are used.

7. 16 reference tracking simulations of the true system are performed. The system is controlled by an MPC based on linear models, with parameter vectors randomly chosen from the ones obtained in the previous identification experiments.

8. Step 6 is repeated, using a white Gaussian noise as input instead. The variance is set to be equal to the one of the optimal input u^*.

9. Step 7 is repeated, using the estimates obtained in step 8.

5.1 Minimum phase setting: $\gamma_1 = \gamma_2 = 0.625$

With this setting the zeros of the system are $s_1 = -0.1040$, $s_2 = -0.0160$. The simulation time is set to 400 s . The value of N used to calculate $\tilde{V}_{app}(\theta, \hat{\theta})$ is equal to the simulation time.

5.1.1 Scenario 1

Reference tracking

Figure 5.1 shows the response of the linear model, with the initial parameter estimate, controlled by an MPC using the same model. The response is fast and due to the simulation time, it is difficult to distinguish the output signals from the respective references. Figure 5.2 shows the input signals used to control the system. Due to the little input weight, the input signal variations are big.

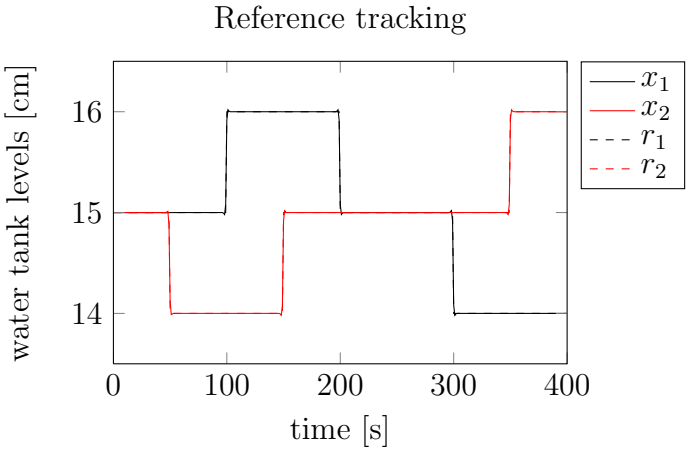

Figure 5.1: Scenario 1: reference tracking simulation. x_1 and x_2 represents the output signals. The dashed lines r_1 and r_2 the reference signals of x_1 and x_2 respectively.

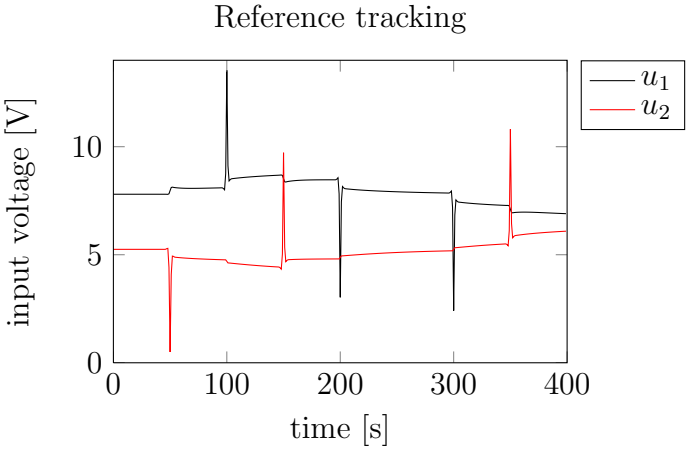

Figure 5.2: Scenario 1: input signals calculated by the MPC to control the system in the reference tracking simulation.

Sensitivity analysis

Figure 5.3 shows $\mathscr{E}_{app}$ and $\mathscr{E}_{SI}$. Calculation of the Hessian of $\tilde{V}_{app}(\theta, \hat{\theta})$ gives,

$$\tilde{V}''_{app}(\hat{\theta}, \hat{\theta}) = \begin{bmatrix} 16.1822 & 13.2851 \\ 13.2851 & 13.2237 \end{bmatrix}. \tag{5.1}$$

The corresponding eigenvalues are $s_1 = 1.3358$ and $s_2 = 28.0701$, while $v_1 = [0.6668 \ -0.7452]^T$ and $v_2 = [-0.7452 \ -0.6668]^T$ are the respective eigenvectors. This gives that the longest semi-axis correspond to the eigenvector v_1, while the smallest one correspond to the eigenvector v_2. This means that a bigger variation of the parameters is tolerated in the direction of v_1, that correspond

to the straight line,

$$\begin{bmatrix}\theta^1\\ \theta^2\end{bmatrix} = \begin{bmatrix}\theta_0^1\\ \theta_0^2\end{bmatrix} + \delta \begin{bmatrix}0.6668\\ -0.7452\end{bmatrix}. \tag{5.2}$$

The area of $\mathscr{E}_{app}$ is 0.1633.
Figure 5.3 shows the resulting estimates from the 100 identification experiments, using the optimal input. The estimates are spread around the true parameter, following the shape of $\mathscr{E}_{app}$ and 96% of these estimates lie inside the identification ellipse. Figure 5.4 shows the reference tracking simulation of the true system,

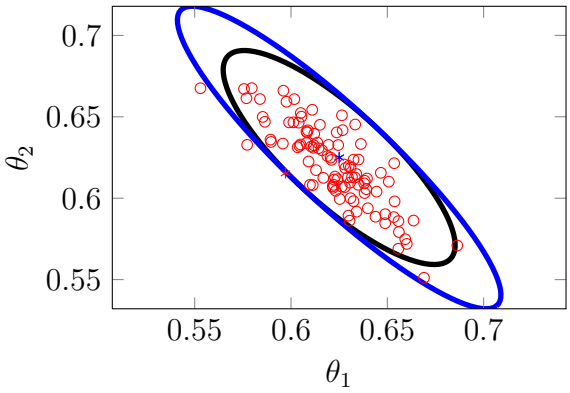

Figure 5.3: Scenario 1: results of step 6 of the procedure. 96% of the estimates lie inside the identification ellipsoid.

see 4.4, controlled with an MPC based on linear models, that use the estimates represented in Figure 5.3. Figure 5.5 shows the resulting estimates, from 100

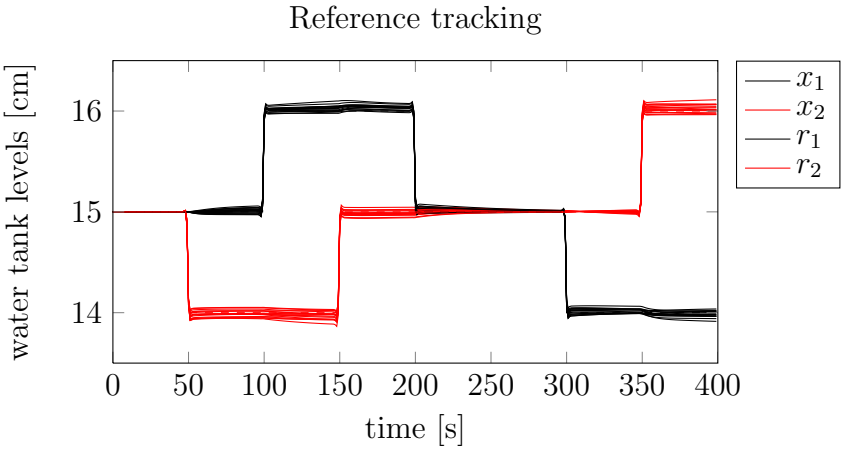

Figure 5.4: Scenario 1: results of step 7 of the procedure.

identification experiments, when a white Gaussian input with the same energy

of the optimal input is used. In this case only 13% of the estimates lie inside the ellipse. Figure 5.6 shows the reference tracking simulation of the true system,

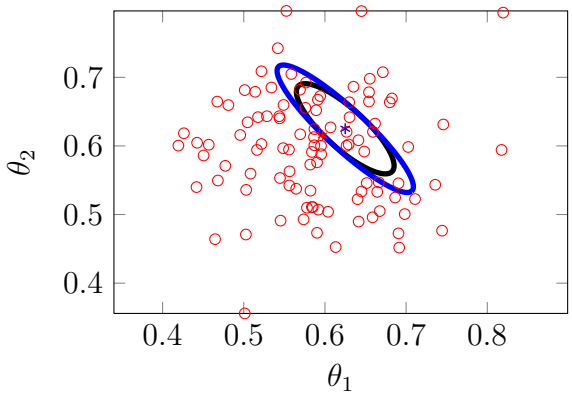

Figure 5.5: Scenario 1: results of step 8 of the procedure. 13% of the estimates lie inside the identification ellipsoid.

see 4.4, controlled with an MPC based on linear models, that use the estimates represented in Figure 5.5.

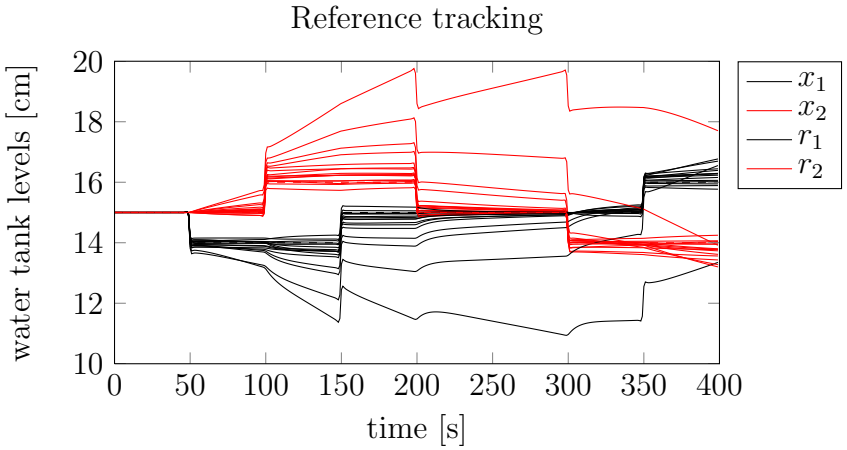

Figure 5.6: Scenario 1: results of step 9 of the procedure.

Observation 1

This example demonstrate that using the optimal input signal in the identification experiments, give an higher percentage of estimates that lie inside the application ellipsoid, compared to the number of estimates inside the application ellipsoid that we obtain when a white Gaussian noise with the same energy is used.

5.1.2 Scenario 2

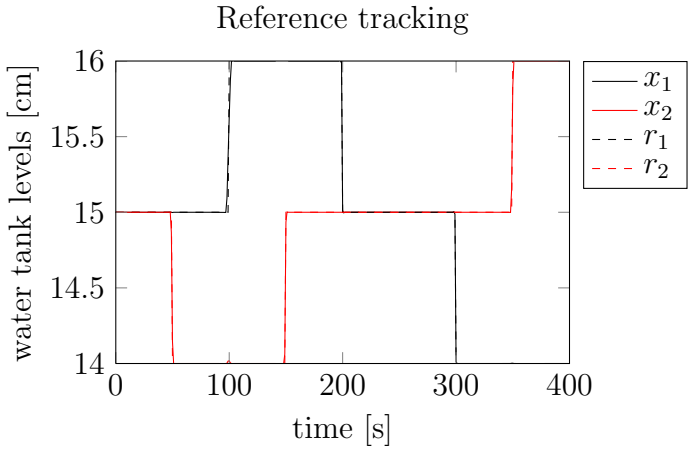

Figure 5.7: Scenario 2: reference tracking simulation. x_1 and x_2 represents the output signals. The dashed lines r_1 and r_2 the reference signals of x_1 and x_2 respectively.

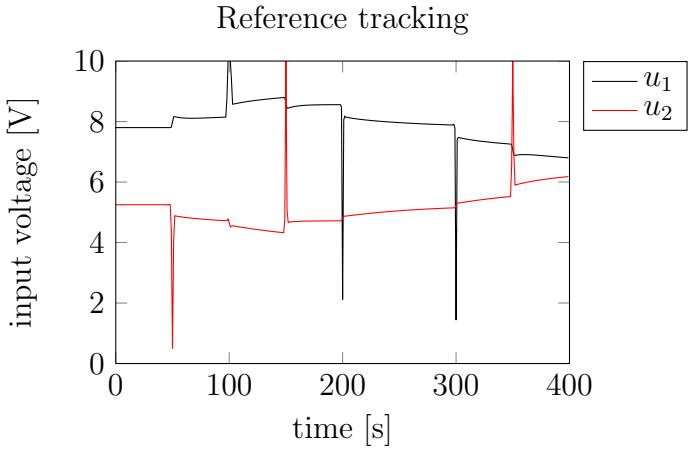

Figure 5.8: Scenario 2: input signals calculated by the MPC to control the system in the reference tracking simulation.

Sensitivity analysis

Figure 5.9 shows $\mathscr{E}_{app}$ and $\mathscr{E}_{SI}$. Calculation of the Hessian of $\tilde{V}_{app}(\theta, \hat{\theta})$ gives,

$$\tilde{V}''_{app}(\hat{\theta}, \hat{\theta}) = \begin{bmatrix} 19.6585 & 16.4728 \\ 16.4728 & 16.3953 \end{bmatrix}. \tag{5.3}$$

The corresponding eigenvalues are $s_1 = 1.4735$ and $s_2 = 34.5803$, while $v_1 = [0.6714 \ -0.7411]^T$ and $v_2 = [-0.7411 \ -0.6714]^T$ are the respective eigenvectors. This gives that the longest semi-axis correspond to the eigenvector v_1, while the smallest one correspond to the eigenvector v_2. This means that a bigger variation of the parameters is tolerated in the direction of v_1, that correspond to the straight line,

$$\begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix} = \begin{bmatrix} \theta_0^1 \\ \theta_0^2 \end{bmatrix} + \delta \begin{bmatrix} 0.6714 \\ -0.7411 \end{bmatrix}. \tag{5.4}$$

The area of $\mathscr{E}_{app}$ is 0.1401.

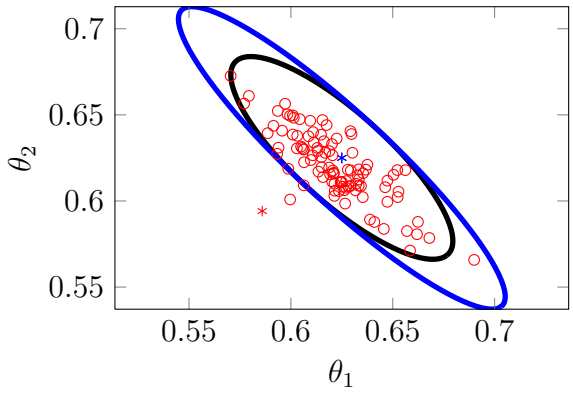

Figure 5.9: Scenario 2: results of step 6 of the procedure. 96% of the estimates lie inside the identification ellipsoid

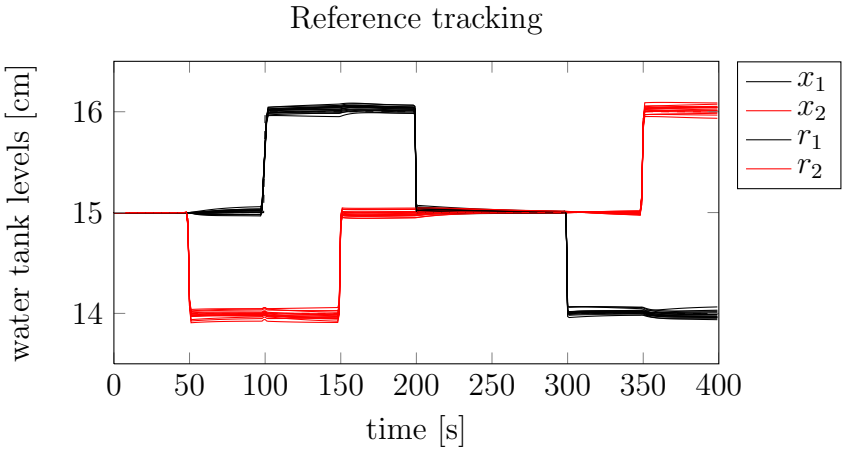

Figure 5.10: Scenario 2: results of step 7 of the procedure.

Observation 2

In Figure 5.11 can be noticed that some parameters, identified using the white Gaussian noise, can make the system instable (we underline that the parameter used in step 9 of the procedure described, are chosen randomly from all the estimates obtained in step 8).

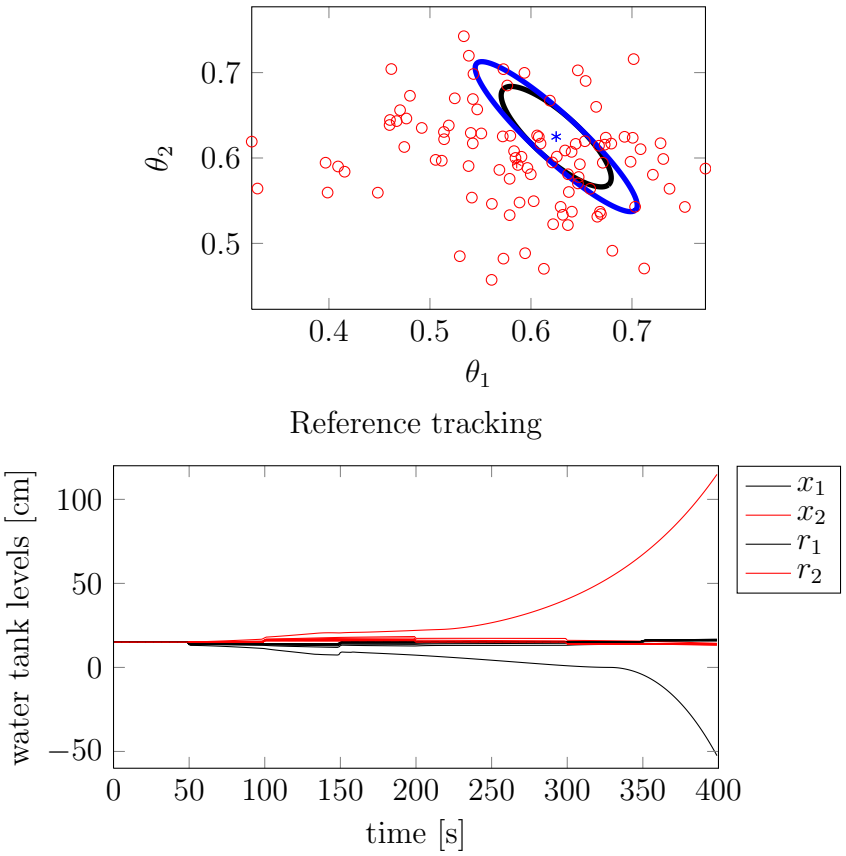

Figure 5.11: Scenario 2: results of step 8 (in the upper figure), and step 9 (in the lower figure)

5.1.3 Scenario 3

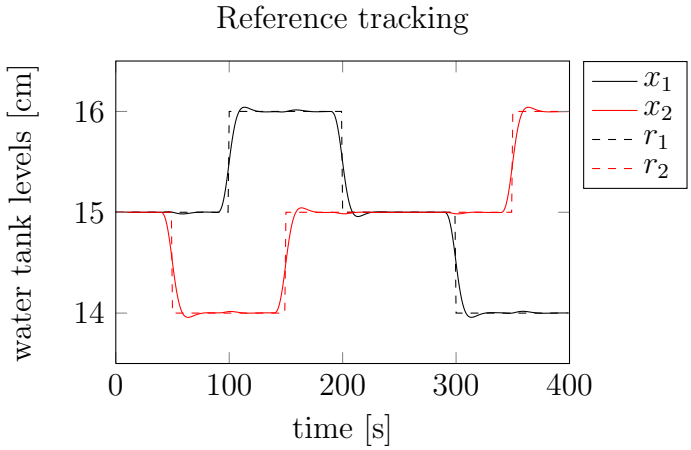

Figure 5.12: Scenario 3: reference tracking simulation. x_1 and x_2 represents the output signals. The dashed lines r_1 and r_2 the reference signals of x_1 and x_2 respectively.

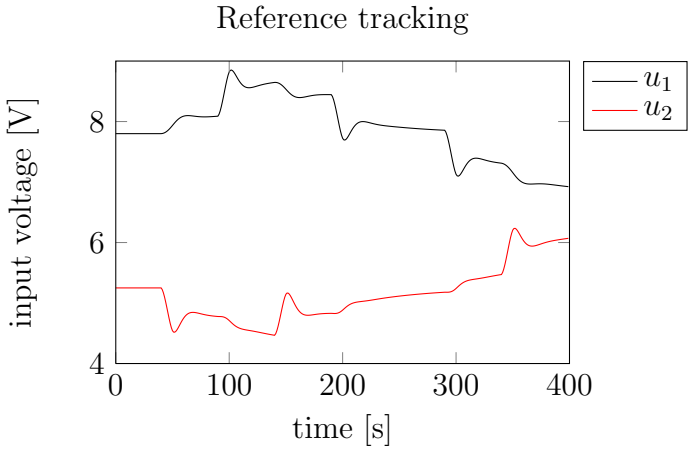

Figure 5.13: Scenario 3: input signals calculated by the MPC to control the system in the reference tracking simulation.

Sensitivity analysis

Figure 5.15 shows $\mathscr{E}_{app}$ and $\mathscr{E}_{SI}$. Calculation of the Hessian of $\tilde{V}_{app}(\theta, \hat{\theta})$ gives,

$$\tilde{V}''_{app}(\hat{\theta}, \hat{\theta}) = \begin{bmatrix} 15.1718 & 12.5269 \\ 12.5269 & 12.4377 \end{bmatrix}. \tag{5.5}$$

The corresponding eigenvalues are $s_1 = 1.2035$ and $s_2 = 26.4060$, while $v_1 = [0.6677 \ -0.7445]^T$ and $v_2 = [-0.7445 \ -0.6677]^T$ are the respective eigenvectors. This gives that the longest semi-axis correspond to the eigenvector v_1, while the smallest one correspond to the eigenvector v_2. This means that a bigger variation of the parameters is tolerated in the direction of v_1, that correspond to the straight line,

$$\begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix} = \begin{bmatrix} \theta_0^1 \\ \theta_0^2 \end{bmatrix} + \delta \begin{bmatrix} 0.6677 \\ -0.7445 \end{bmatrix}. \tag{5.6}$$

The area of $\mathscr{E}_{app}$ is 0.1774.

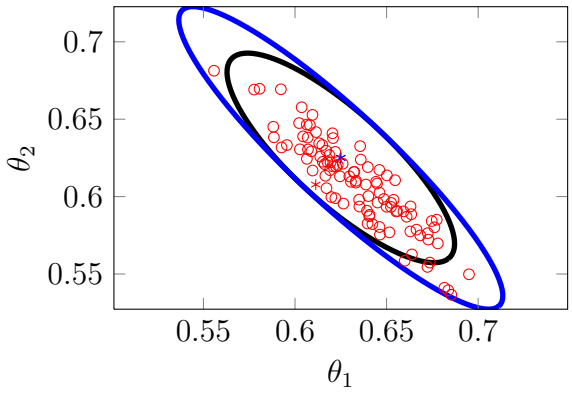

Figure 5.14: Scenario 3: results of step 6 of the procedure. 92% of the estimates lie inside the identification ellipsoid

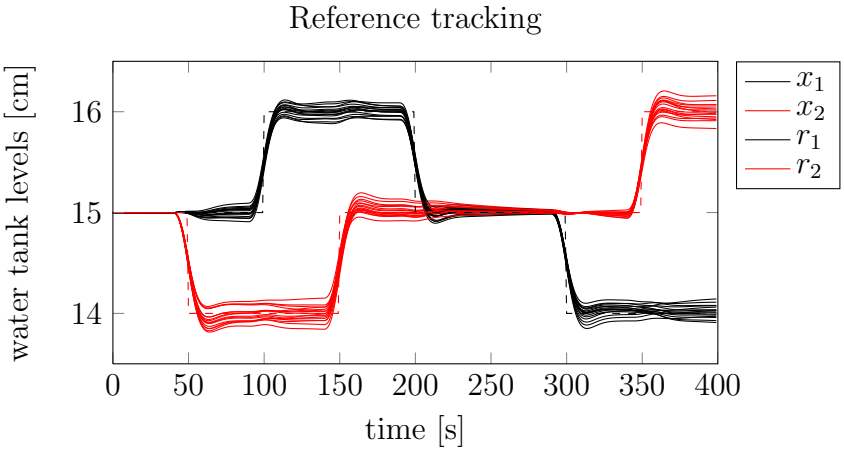

Figure 5.15: Scenario 3: results of step 7 of the procedure.

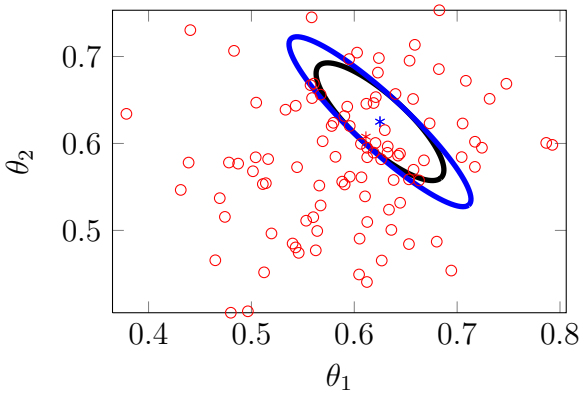

Figure 5.16: Scenario 3: results of step 8 of the procedure. 14% of the estimates lie inside the identification ellipsoid.

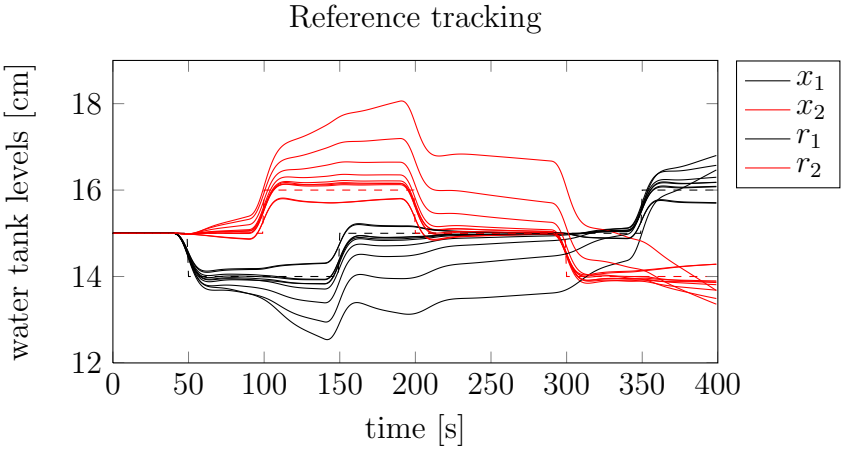

Figure 5.17: Scenario 3: results of step 9 of the procedure.

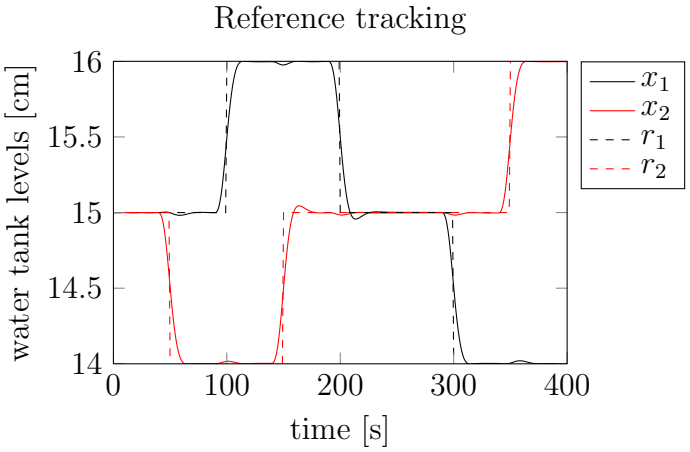

Figure 5.18: Scenario 4: reference tracking simulation. x_1 and x_2 represents the output signals. The dashed lines r_1 and r_2 the reference signals of x_1 and x_2 respectively.

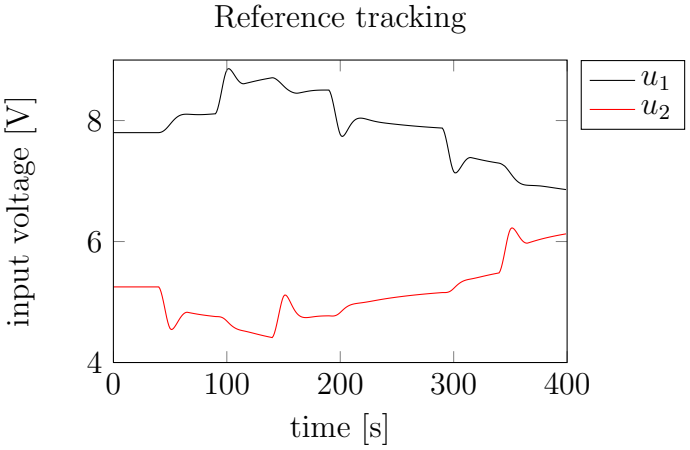

Figure 5.19: Scenario 4: input signals calculated by the MPC to control the system in the reference tracking simulation.

5.1.4 Scenario 4

Sensitivity analysis

Figure 5.21 shows $\mathscr{E}_{app}$ and $\mathscr{E}_{SI}$. Calculation of the Hessian of $\tilde{V}_{app}(\theta, \hat{\theta})$ gives,

$$\tilde{V}''_{app}(\hat{\theta}, \hat{\theta}) = \begin{bmatrix} 17.0033 & 14.1985 \\ 14.1985 & 14.0111 \end{bmatrix}. \tag{5.7}$$

The corresponding eigenvalues are $s_1 = 1.2301$ and $s_2 = 29.7843$, while $v_1 = [0.6690 \ -0.7432]^T$ and $v_2 = [-0.7432 \ -0.6690]^T$ are the respective eigenvectors. This gives that the longest semi-axis correspond to the eigenvector v_1, while the smallest one correspond to the eigenvector v_2. This means that a bigger

variation of the parameters is tolerated in the direction of v_1, that correspond to the straight line,

$$\begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix} = \begin{bmatrix} \theta_0^1 \\ \theta_0^2 \end{bmatrix} + \delta \begin{bmatrix} 0.6690 \\ -0.7432 \end{bmatrix}. \tag{5.8}$$

The area of $\mathscr{E}_{app}$ is 0.1652.

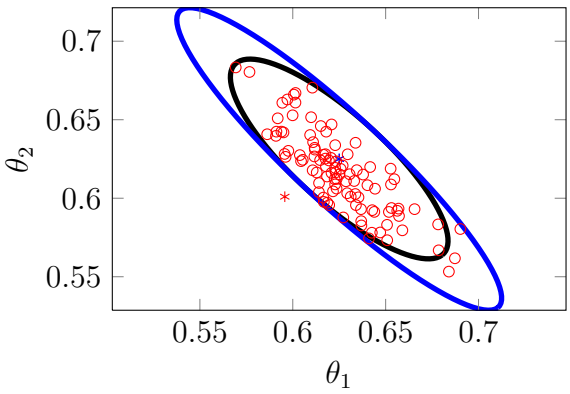

Figure 5.20: Scenario 4: results of step 6 of the procedure. 96% of the estimates lie inside the identification ellipsoid

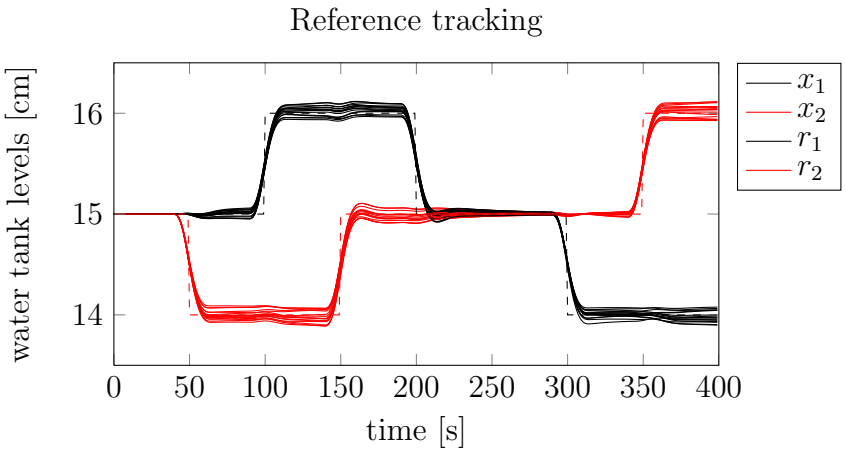

Figure 5.21: Scenario 4: results of step 7 of the procedure.

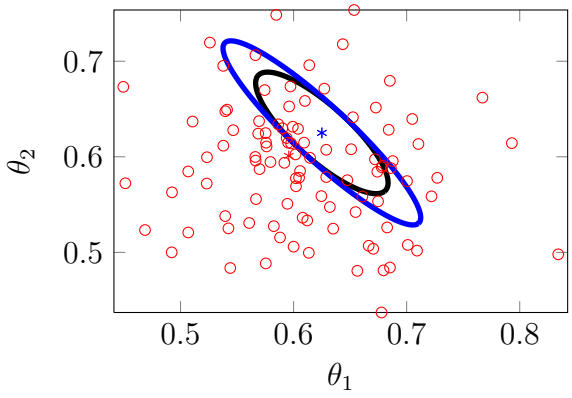

Figure 5.22: Scenario 4: results of step 8 of the procedure. 17% of the estimates lie inside the identification ellipsoid.

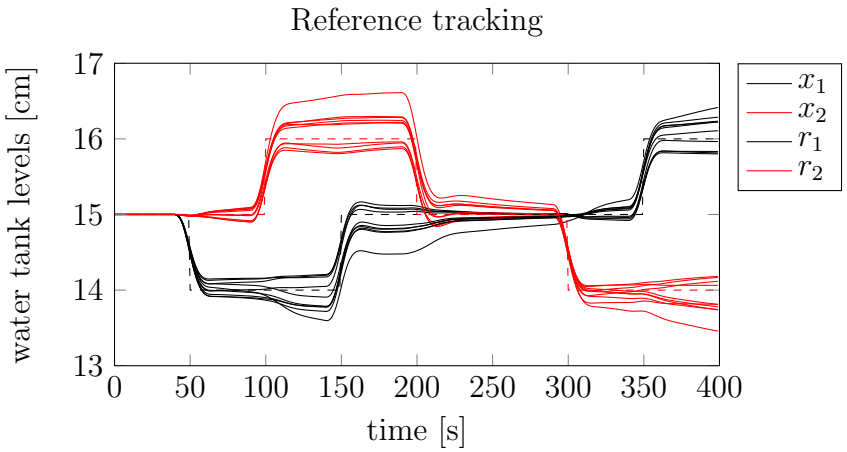

Figure 5.23: Scenario 4: results of step 9 of the procedure.

5.2 Minimum phase setting: $\gamma_1 = \gamma_2 = 0.505$

With this setting the zeros of the system are: $s_1 = -0.1572$, $s_2 = -0.0012$

5.2.1 Scenario 1

Reference tracking

Figure 5.24 shows the response of the linear model, with the initial parameter estimate, controlled by an MPC using the same model. The response is fast and outputs follow easily the reference. Figure 5.25 shows the input signals used to control the system. Due to the little weight used, the inputs are allowed to vary a lot. From Figure 5.26 we can notice that one of the state reach its maximal allowed value. In fact Figure 5.26 represent the variation from the equilibrium point. The fourth state, reach the value of 13 cm, that is it reaches the maximal value of 25 cm (see Table 4.1).

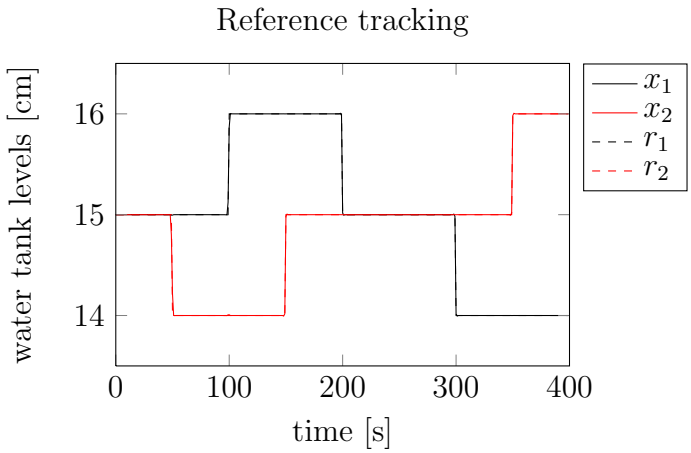

Figure 5.24: Scenario 1: reference tracking simulation. x_1 and x_2 represents the output signals. The dashed lines r_1 and r_2 the reference signals of x_1 and x_2 respectively.

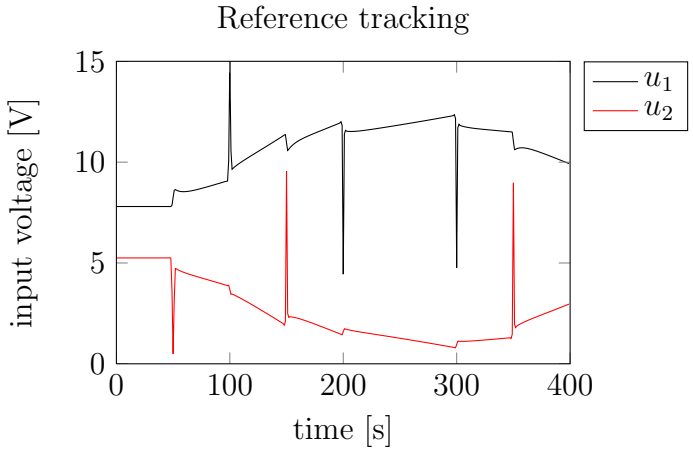

Figure 5.25: Scenario 1: input signals calculated by the MPC to control the system in the reference tracking simulation.

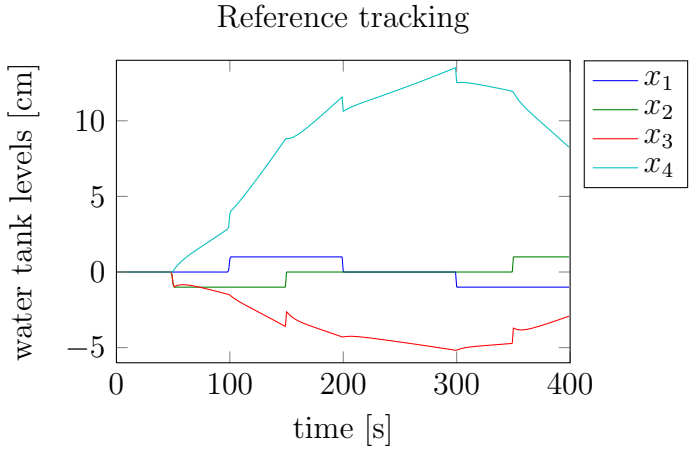

Figure 5.26: Scenario 1: simulation of the water levels in the four tanks. The water level x_4 reaches its maximal allowed value.

Sensitivity analysis

Figure 5.27 shows $\mathscr{E}_{app}$ and $\mathscr{E}_{SI}$ which are overlapped. Calculation of the Hessian of $\tilde{V}_{app}(\theta, \hat{\theta})$ gives,

$$\tilde{V}''_{app}(\hat{\theta}, \hat{\theta}) = \begin{bmatrix} 726.5258 & 725.3857 \\ 725.3867 & 731.5592 \end{bmatrix}. \tag{5.9}$$

The corresponding eigenvalues are $s_1 = 3.65$ and $s_2 = 1454.4$, while $v_1 = [-0.7083\ 0.7059]^T$ and $v_2 = [0.7059\ 0.7083]^T$ are the respective eigenvectors. This gives that the longest semi-axis correspond to the eigenvector v_1, while the smallest one correspond to the eigenvector v_2. This means that a bigger

variation of the parameters is tolerated in the direction of v_1, that correspond to the straight line,

$$\begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix} = \begin{bmatrix} \theta_0^1 \\ \theta_0^2 \end{bmatrix} + \delta \begin{bmatrix} -0.7083 \\ 0.7059 \end{bmatrix}. \tag{5.10}$$

The area of $\mathscr{E}_{app}$ is 0.0137.

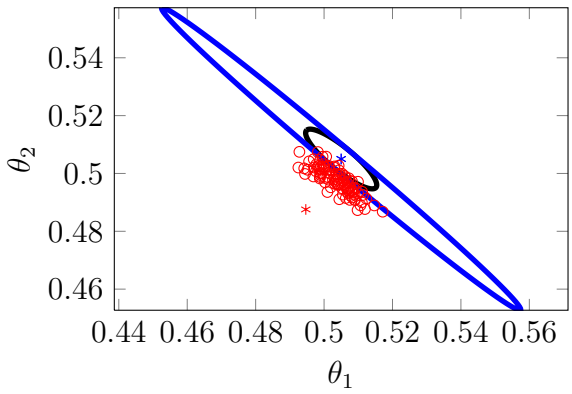

Figure 5.27: Scenario 1: results of step 6 of the procedure. 11% of the estimates lie inside the identification ellipsoid.

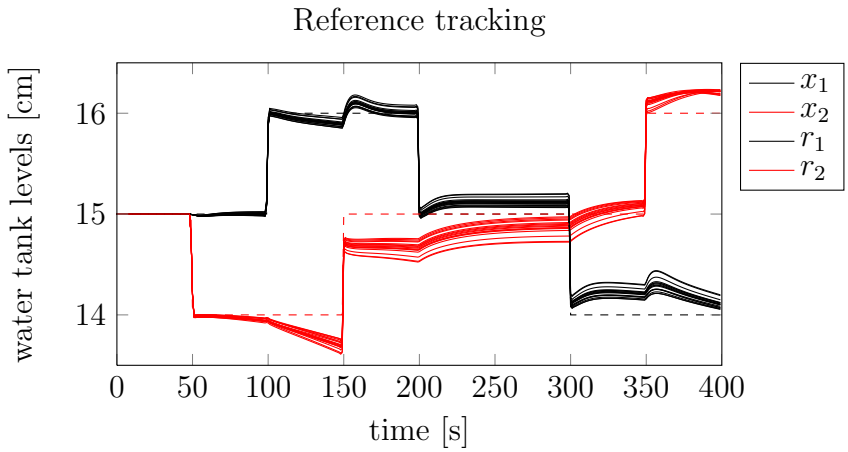

Figure 5.28: Scenario 1: results of step 7 of the procedure.

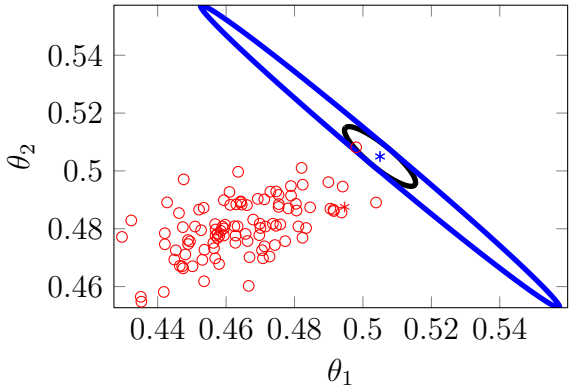

Figure 5.29: Scenario 1: results of step 8 of the procedure. 1% of the estimates lie inside the identification ellipsoid.

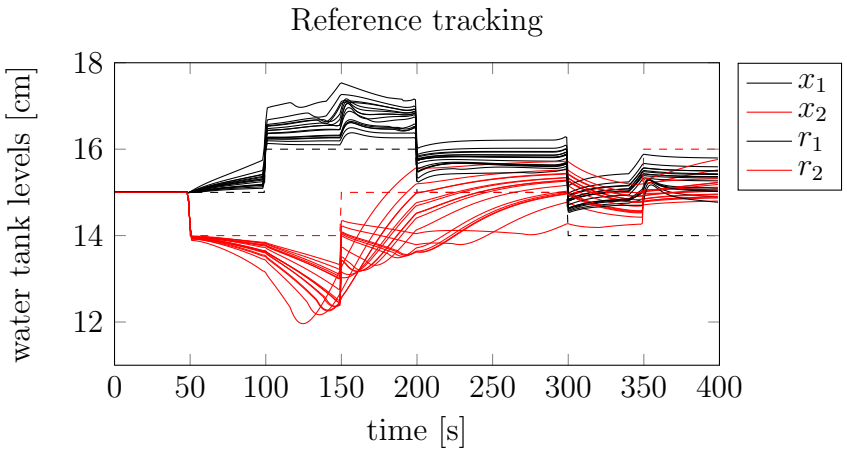

Figure 5.30: Scenario 1: results of step 9 of the procedure.

Observation 3

Figure 5.27 shows that when the sum of γ is close to 1 the application ellipsoid stretch out in the direction of the longer semi-axes more than in the other cases.

Observation 4: offset in the estimates

Figures 5.27,5.33,5.38 and 5.43 show that the estimates make a blob that is not centered around the true parameter θ_0. It is interesting to notice that this behavior was already present in the minimum phase case, just not so evident. See for example figures 5.9, 5.15. This behavior is discussed more in detail in Chapter 6.

5.2.2 Scenario 2

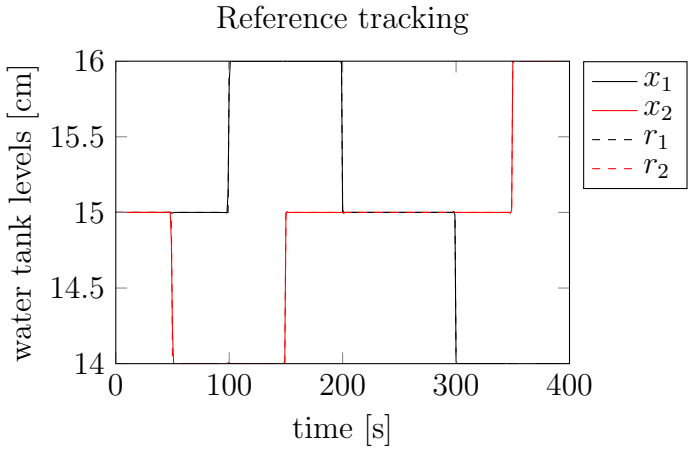

Figure 5.31: Scenario 2: reference tracking simulation. x_1 and x_2 represents the output signals. The dashed lines r_1 and r_2 the reference signals of x_1 and x_2 respectively.

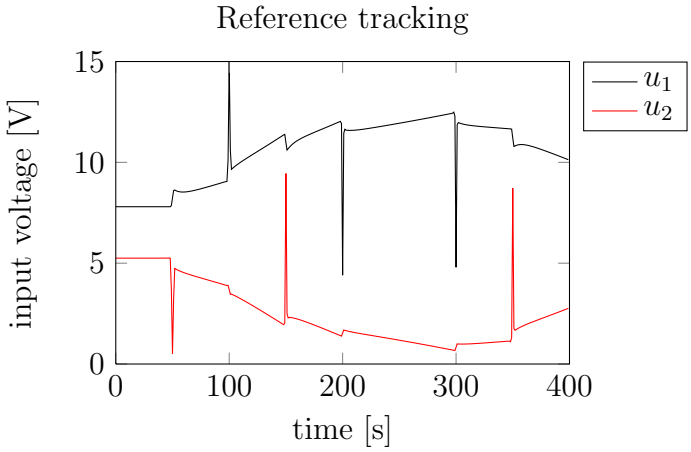

Figure 5.32: Scenario 2: input signals calculated by the MPC to control the system in the reference tracking simulation.

Sensitivity analysis

Figure 5.33 shows $\mathscr{E}_{app}$ and $\mathscr{E}_{SI}$. Calculation of the Hessian of $\tilde{V}_{app}(\theta, \hat{\theta})$ gives,

$$\tilde{V}''_{app}(\hat{\theta}, \hat{\theta}) = \begin{bmatrix} 762.3064 & 761.0394 \\ 761.0394 & 767.1592 \end{bmatrix}. \tag{5.11}$$

The corresponding eigenvalues are $s_1 = 3.68$ and $s_2 = 1525.8$, while $v_1 = [-0.7082\ 0.7060]^T$ and $v_2 = [0.7060\ 0.7082]^T$ are the respective eigenvectors.

This gives that the longest semi-axis correspond to the eigenvector v_1, while the smallest one correspond to the eigenvector v_2. This means that a bigger variation of the parameters is tolerated in the direction of v_1, that correspond to the straight line,

$$\begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix} = \begin{bmatrix} \theta_0^1 \\ \theta_0^2 \end{bmatrix} + \delta \begin{bmatrix} -0.7082 \\ 0.7060 \end{bmatrix}. \tag{5.12}$$

The area of $\mathscr{E}_{app}$ is 0.0133.

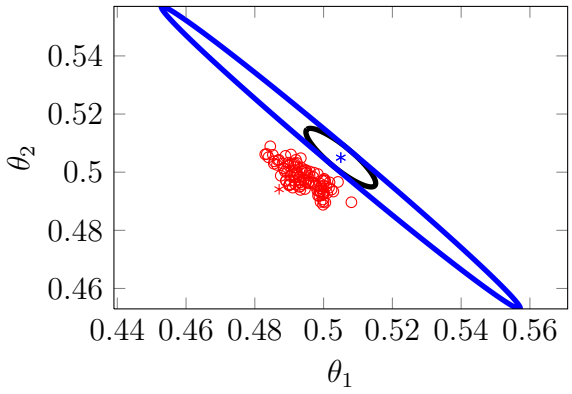

Figure 5.33: Scenario 2: results of step 6 of the procedure. No estimates lie inside the identification ellipsoid.

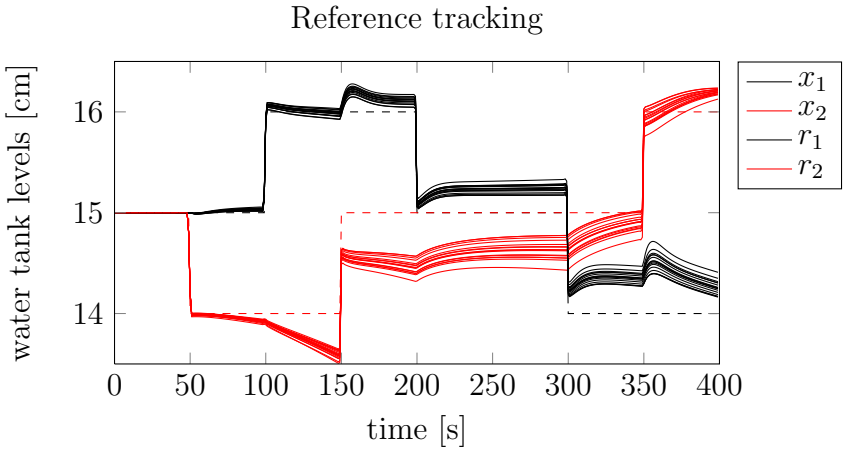

Figure 5.34: Scenario 2: results of step 7 of the procedure.

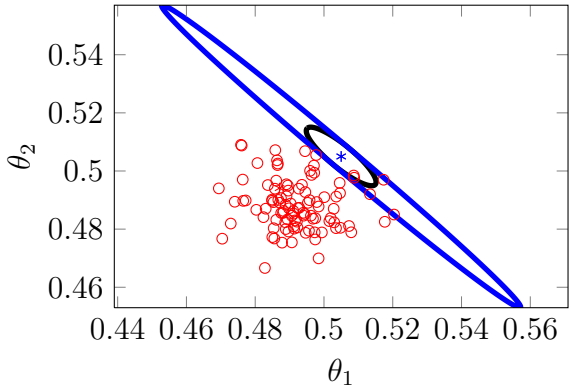

Figure 5.35: Scenario 2: results of step 8 of the procedure. 2% of the estimates lie inside the identification ellipsoid.

5.2.3 Scenario 3

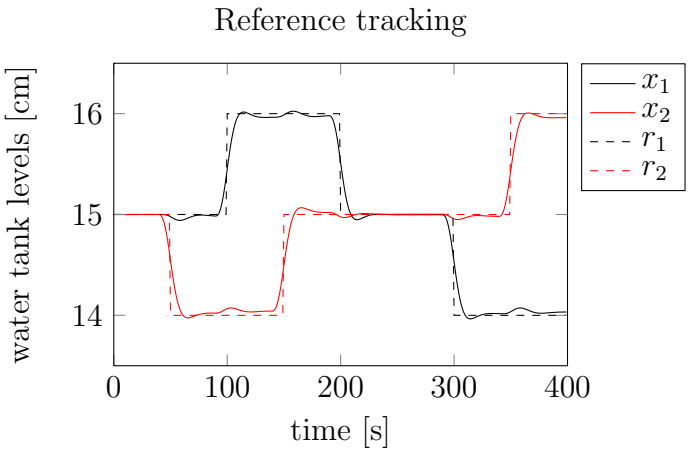

Figure 5.36: Scenario 3: reference tracking simulation. x_1 and x_2 represents the output signals. The dashed lines r_1 and r_2 the reference signals of x_1 and x_2 respectively.

Sensitivity analysis

Figure 5.38 shows $\mathscr{E}_{app}$ and $\mathscr{E}_{SI}$ which are overlapped. Calculation of the Hessian of $\tilde{V}_{app}(\theta, \hat{\theta})$ gives,

$$\tilde{V}''_{app}(\hat{\theta}, \hat{\theta}) = \begin{bmatrix} 338.3409 & 335.6563 \\ 335.6563 & 338.5342 \end{bmatrix}. \tag{5.13}$$

The corresponding eigenvalues are $s_1 = 2.7812$ and $s_2 = 674.09$, while $v_1 = [-0.7072\ 0.7070]^T$ and $v_2 = [0.7070\ 0.7072]^T$ are the respective eigenvectors. This gives that the longest semi-axis correspond to the eigenvector v_1, while

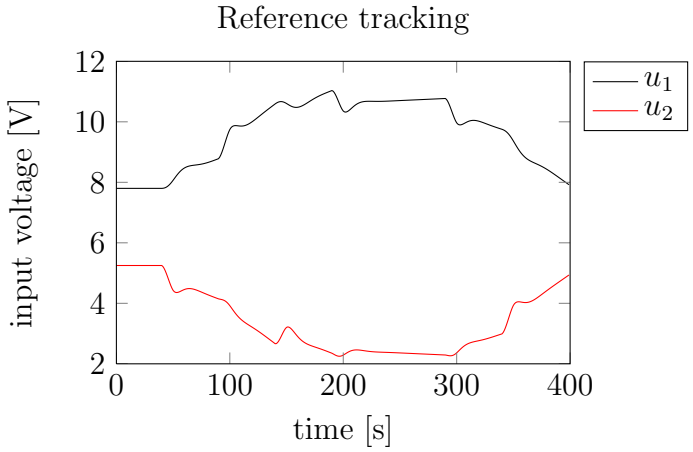

Figure 5.37: Scenario 3: input signals calculated by the MPC to control the system in the reference tracking simulation.

the smallest one correspond to the eigenvector v_2. This means that a bigger variation of the parameters is tolerated in the direction of v_1, that correspond to the straight line,

$$\begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix} = \begin{bmatrix} \theta_0^1 \\ \theta_0^2 \end{bmatrix} + \delta \begin{bmatrix} -0.7072 \\ 0.7070 \end{bmatrix}. \tag{5.14}$$

The area of $\mathscr{E}_{app}$ is 0.0231.

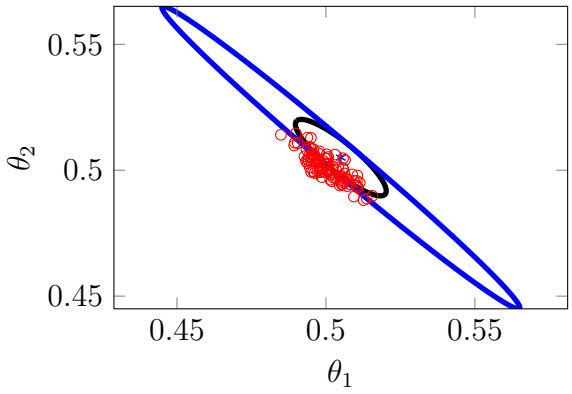

Figure 5.38: Scenario 3: results of step 6 of the procedure. 22% of the estimates lie inside the identification ellipsoid.

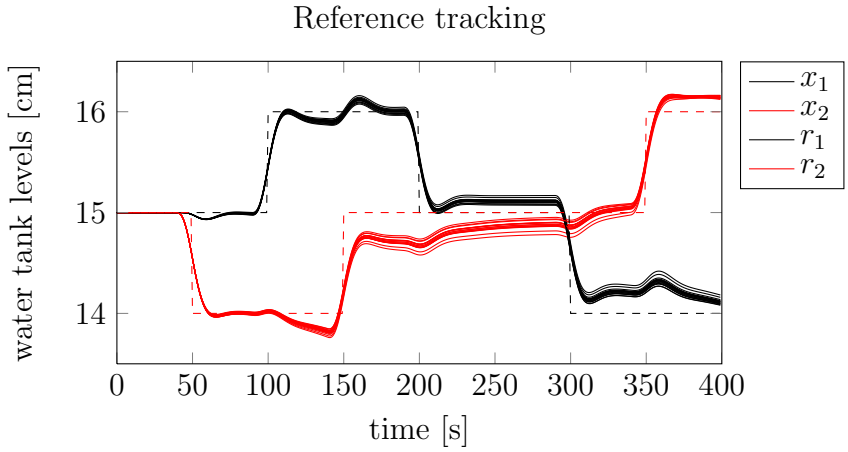

Figure 5.39: Scenario 3: results of step 7 of the procedure.

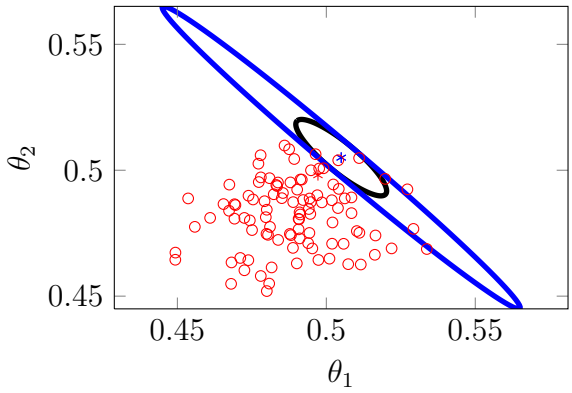

Figure 5.40: Scenario 3, check of theory - u^*.

5.2.4 Scenario 4

Sensitivity analysis

Figure 5.43 shows $\mathscr{E}_{app}$ and $\mathscr{E}_{SI}$. Calculation of the Hessian of $\tilde{V}_{app}(\theta, \hat{\theta})$ gives,

$$\tilde{V}''_{app}(\hat{\theta}, \hat{\theta}) = \begin{bmatrix} 640.14 & 638.5481 \\ 638.5481 & 642.8231 \end{bmatrix}. \tag{5.15}$$

The corresponding eigenvalues are $s_1 = 2.932$ and $s_2 = 1280$, while $v_1 = [-0.7078\ 0.7064]^T$ and $v_2 = [0.7064\ 0.7078]^T$ are the respective eigenvectors. This gives that the longest semi-axis correspond to the eigenvector v_1, while the smallest one correspond to the eigenvector v_2. This means that a bigger variation of the parameters is tolerated in the direction of v_1, that correspond

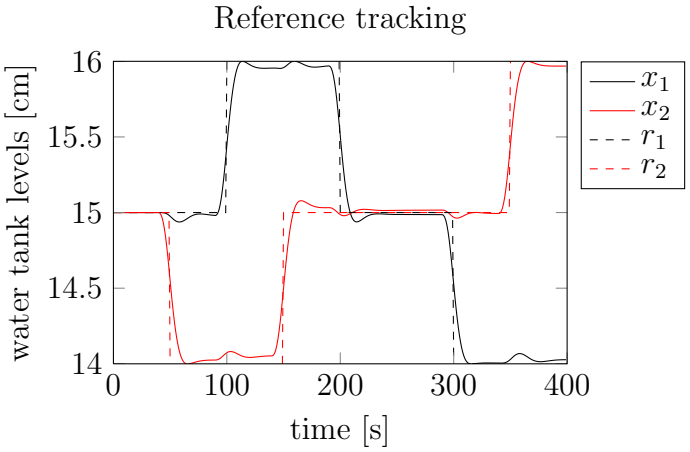

Figure 5.41: Scenario 4: reference tracking simulation. x_1 and x_2 represents the output signals. The dashed lines r_1 and r_2 the reference signals of x_1 and x_2 respectively.

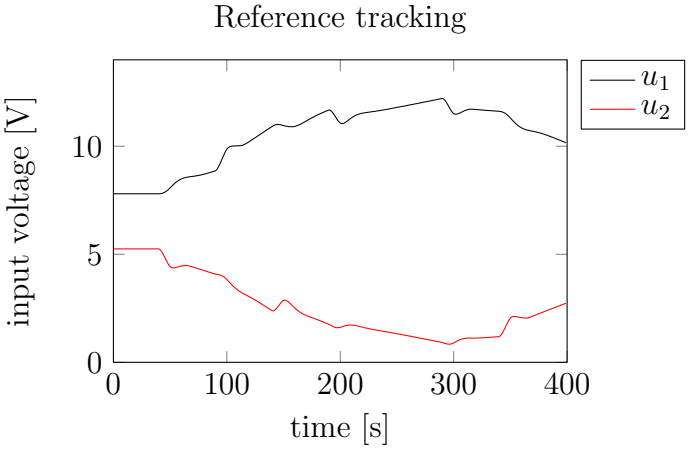

Figure 5.42: Scenario 4: input signals calculated by the MPC to control the system in the reference tracking simulation.

to the straight line,

$$\begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix} = \begin{bmatrix} \theta_0^1 \\ \theta_0^2 \end{bmatrix} + \delta \begin{bmatrix} -0.7078 \\ 0.7064 \end{bmatrix}. \tag{5.16}$$

The area of $\mathscr{E}_{app}$ is 0.0163.

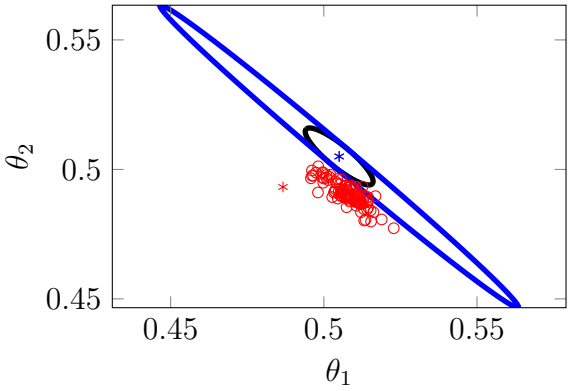

Figure 5.43: Scenario 4: results of step 6 of the procedure. No estimates lie inside the identification ellipsoid.

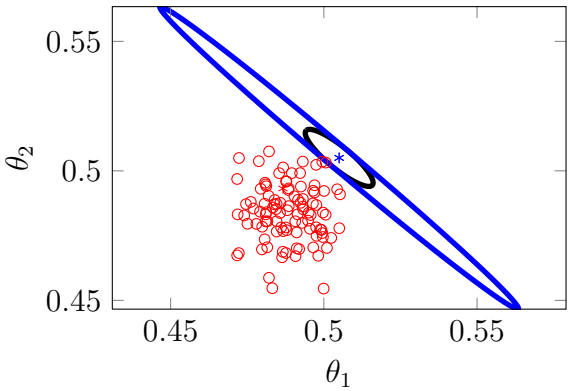

Figure 5.44: Scenario 4: results of step 8 of the procedure. No estimates lie inside the identification ellipsoid.

5.3 Non minimum phase case: $\gamma_1 = \gamma_2 = 0.25$

With this setting the zeros of the system are: $s_1 = -0.4449$, $s_2 = 0.2047$. For this case we do not show any figures of the results because could be repetitive and not much interesting. All the results of this setting are used instead to make comparisons in Chapter 6.

5.3.1 Observation

For this setting, the reference tracking was quite hard when the constraints on inputs were applied. So for this setting Scenario 2 and 4 are simulated with only output constraints.

6
Discussion

The numerical results obtained from the simulations presented in Chapter 5 are summarized in Table 6.1,6.2,6.3. From the analysis of the obtained results it is possible to make the following observations:

1. The area of the application ellipsoid is smaller when $\gamma_1 + \gamma_2 \approx 1$. Meaning that this setting is actually harder to control, in the sense that the requirements on the model are much higher. See for example Figure 6.2.

2. The eigenvalue s_2 becomes very high when $\gamma_1 + \gamma_2 \approx 1$, namely, the semi-axis in the direction of v_2 becomes very small. That is, $\mathscr{E}_{app}$ becomes much narrower in the direction of v_2. This means that performance degrades mostly when estimated parameters vary in that direction.

3. The area of the application ellipsoid is bigger when we use higher weights on inputs compared to outputs. Meaning that this setting results in a larger number of models that meet the performance specifications. Thus, the robustness of the controller is higher.

4. The longer semi-axis corresponds to the direction of the eigenvector v_1,

while the smallest one corresponds to the eigenvector v_2. This means that a bigger variation of the estimates in the direction of the eigenvector v_1 is tolerated.

5. Parameter variations in the direction of v_2 give the higher degradation of performance. From Table 6.2 we can see that, when $\gamma_1 + \gamma_2 \approx 1$, that direction is defined by the line $y = x + (\gamma_1 + \gamma_2)$.

6. The area of the application ellipsoid is usually smaller when we analyse scenarios with active constraints compared to the area obtained with scenarios with no active constraints. See for example Figure 6.3.

	Scenario 1	**Scenario 2**	**Scenario 3**	**Scenario 4**
Area	0.1633	0.1401	0.1774	0.1652
$\boldsymbol{s}_1$	1.358	1.4735	1.2035	1.2301
$\boldsymbol{s}_2$	28.07	34.58	26.406	29.7843
$\boldsymbol{v}_1$	$\begin{bmatrix} 0.668 \\ -0.7452 \end{bmatrix}$	$\begin{bmatrix} 0.6714 \\ -0.7411 \end{bmatrix}$	$\begin{bmatrix} 0.6677 \\ -0.7432 \end{bmatrix}$	$\begin{bmatrix} 0.669 \\ -0.7432 \end{bmatrix}$
$\boldsymbol{v}_2$	$\begin{bmatrix} -0.7452 \\ -0.668 \end{bmatrix}$	$\begin{bmatrix} -0.7411 \\ -0.6714 \end{bmatrix}$	$\begin{bmatrix} -0.7432 \\ -0.6677 \end{bmatrix}$	$\begin{bmatrix} -0.7432 \\ -0.669 \end{bmatrix}$

Table 6.1: $\gamma_1 = \gamma_2 = 0.625$

	Scenario 1	**Scenario 2**	**Scenario 3**	**Scenario 4**
Area	0.0137	0.0133	0.0231	0.0163
$\boldsymbol{s}_1$	3.650	3.6800	2.7812	2.932
$\boldsymbol{s}_2$	1454.4	1525.8	674.09	1280
$\boldsymbol{v}_1$	$\begin{bmatrix} 0.7083 \\ -0.7059 \end{bmatrix}$	$\begin{bmatrix} 0.7082 \\ -0.7060 \end{bmatrix}$	$\begin{bmatrix} 0.7072 \\ -0.7070 \end{bmatrix}$	$\begin{bmatrix} 0.7078 \\ -0.7064 \end{bmatrix}$
$\boldsymbol{v}_2$	$\begin{bmatrix} -0.7059 \\ -0.7083 \end{bmatrix}$	$\begin{bmatrix} -0.7060 \\ -0.7082 \end{bmatrix}$	$\begin{bmatrix} -0.7070 \\ -0.7072 \end{bmatrix}$	$\begin{bmatrix} -0.7064 \\ -0.7078 \end{bmatrix}$

Table 6.2: $\gamma_1 = \gamma_2 = 0.505$

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Area	0.2671	0.2823	0.3439	0.2042
$\boldsymbol{s}_1$	1.1121	1.0867	0.7953	1.0556
$\boldsymbol{s}_2$	12.6086	11.5450	10.6335	22.7166
$\boldsymbol{v}_1$	$\begin{bmatrix} 0.6733 \\ -0.7394 \end{bmatrix}$	$\begin{bmatrix} 0.6704 \\ -0.7420 \end{bmatrix}$	$\begin{bmatrix} 0.6821 \\ -0.7312 \end{bmatrix}$	$\begin{bmatrix} 0.5871 \\ -0.8095 \end{bmatrix}$
$\boldsymbol{v}_2$	$\begin{bmatrix} -0.7394 \\ -0.6733 \end{bmatrix}$	$\begin{bmatrix} -0.7420 \\ -0.6704 \end{bmatrix}$	$\begin{bmatrix} -0.7312 \\ -0.6821 \end{bmatrix}$	$\begin{bmatrix} -0.8095 \\ -0.5871 \end{bmatrix}$

Table 6.3: $\gamma_1 = \gamma_2 = 0.25$

6.1 Estimates offset

In Chapter 5 we observed that the estimates obtained from the identification experiments are not spread out around the true parameter value (see figures 5.27,5.33,5.38 and 5.43). This behavior might be explained by the fact that the true system is nonlinear and the estimated models are linear instead. The PEM method assures that the estimates will converge to the true parameter if the true system is linear. To verify this idea, we identified the nonlinear system using 6000 samples using the corresponding optimal input. The results are shown in Figure 6.1. Figure 6.1 shows that even increasing the number of samples in the identification experiment the estimates do not converge to the true parameter for the nonlinear system. It is possible to notice instead, that the estimates are more clearly centered around a different parameter value. This enforces the idea that the true parameter for the linear model differs from the true one of the nonlinear system.

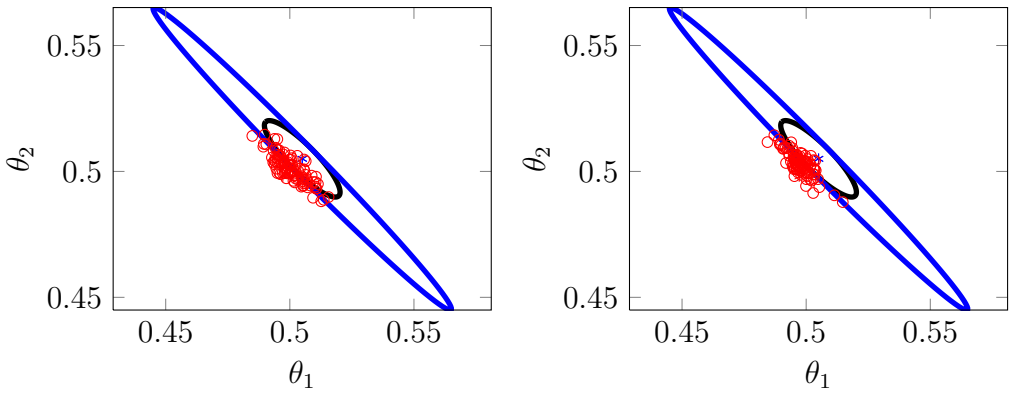

Figure 6.1: Comparison that show the estimates offset when 600 samples (left figure) and when 6000 samples (right figure) are used in the system identification experiment.

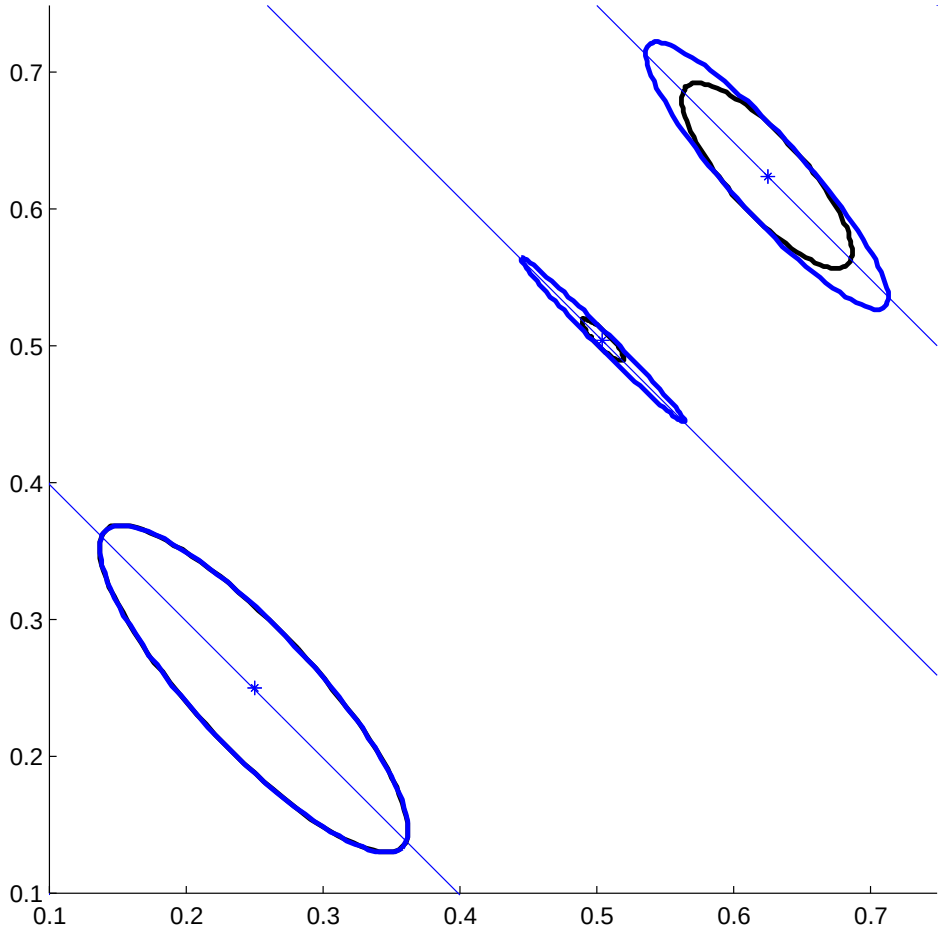

Figure 6.2: Comparison of the application ellipses obtained for different values of the true parameter vector. The application ellipse becomes very small when $\gamma_1 + \gamma_2 \approx 1$. The rects represent the straight line $y = -x + (\gamma_1 + \gamma_2)$ for the respective value of γ_1, γ_2. The rects help to see the orientation of the ellipse in the parameter space.

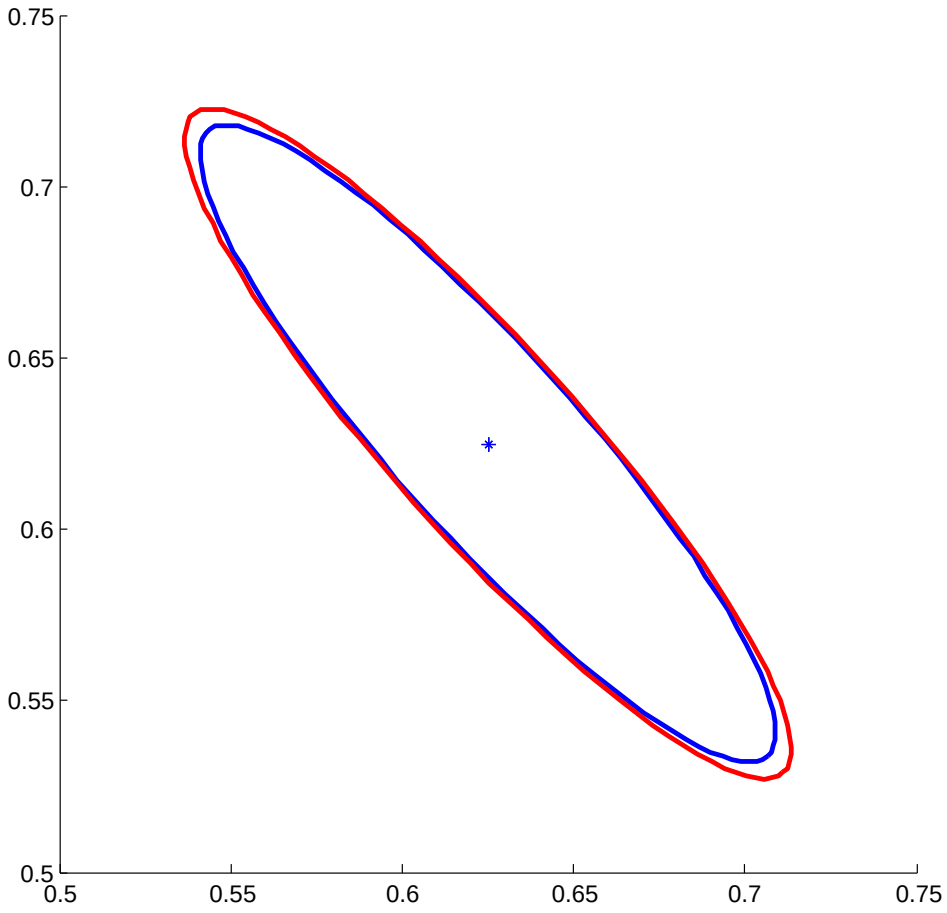

Figure 6.3: Comparison between the application ellipse obtained from scenario 1 (inner ellipse) and the application ellipse obtained from scenario 3 (outer ellipse). With scenario 3 (higher weight on inputs), the ellipse's area is bigger. The center of the ellipses is the true parameter vector.

7

Conclusions and future work

In this chapter we make some remarks of the obtained results and suggest some ideas for future work.

The main objective of this master thesis work was to investigate if MPC settings affect the shape and the important directions of the application ellipsoid $\mathscr{E}_{app}$. The quadruple water tank process has the characteristic property of changing the position of one zero, to be minimum phase or non minimum phase, using the physical parameters γ_1, γ_2 (see Section 4.2). Hence, MPC tuning parameters and constraints are used to define four scenarios. Simulations are carried out for every scenario considering the system to be minimum phase, non minimum phase and close to shifting from minimum phase to non minimum phase. The latter case corresponds to when one system's zero is close to the origin, that is $\gamma_1 + \gamma_2 \approx 1$. This situation was really hard to control so constraints on input were not imposed. Obtained results are summarized with the following remarks:

- Different MPC settings affect the area of the application ellipsoid. When we consider higher weights on inputs the area of $\mathscr{E}_{app}$ becomes bigger. This means that limiting the freedom of the controller requires less accurate models. Active constraints increase controller's performance demand and

reduce the robustness of the controller resulting in a smaller application ellipsoid's area.

- Different positions of the system's zero have a strong influence on $\mathscr{E}_{app}$. The closer the zero is to the origin, the narrower becomes the $\mathscr{E}_{app}$ in the smaller semi-axis direction. The position of the system's zero depends on the value assumed by $\gamma_1 + \gamma_2$.

Future work

This master thesis project does not represent a complete solution and further researches are recommended in the following areas:

- The number of parameters: our analysis takes only two parameters, γ_1 and γ_2, into account. In this case $\mathscr{E}_{app}$ has only 2 dimensions and it can be graphically represented. Furthermore, the two parameters we consider have a straightforward physical interpretation. For the future work we suggest to take into consideration all the parameters of the nonlinear model of the water tanks.

- Application cost: in this work we consider performance degradation in reference tracking applications. Another possibility is to focus on the disturbance rejection. A different application cost for this application should be defined.

- Real plant: we suggest to verify the obtained results of the simulations on the real water tank process.

Bibliography

[1] Sciencedirect.

[2] Märta Barenthin Syberg. *Complexity Issues, Validation and Input Design for Control in System Identification*. PhD thesis, KTH, Automatic Control, 2008. QC 20100702.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. CUP, 2004.

[4] M. Annergren C. A. Larsson. Moose: Model based optimal input design toolbox for matlab, 2011.

[5] Torkel Glad and Lennart Ljung. *Control Theory : Multivariable & Nonlinear Methods*. CRC, March 2000.

[6] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[7] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `../../cvx`, apr 2011.

[8] Håkan Hjalmarsson. System identification of complex and structured systems. *European Journal of Control*, 15(3-4):275–310, 2009. QC 20110112.

[9] Henrik Jansson and Håkan Hjalmarsson. Input design via lmis admitting frequency-wise model specifications in confidence regions. *IEEE Transactions on Automatic Control*, 50(10):1534–1549, 2005. QC 20100525.

[10] Karl H. Johansson and J. L. R. Nunes. A multivariable laboratory process with an adjustable zero. In *Proceedings of the American Control Conference, Philadelphia, Pennsylvania June 1998*, pages 2045–2049. AACC, 1998. QC 20120227.

[11] Karl Henrik Johansson. The quadruple-tank process-a multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control Systems Technology*, 8(3):456–465, 2000. QC 20120227.

[12] Christian Larsson. Toward applications oriented optimal input design with focus on model predictive control, 2011. QC 20111013.

[13] Christian A. Larsson, Mariette Annergren, and Håkan Hjalmarsson. On optimal input design in system identification for model predictive control. In *CDC-ECE*, pages 805–810. IEEE, 2011.

[14] Christian A. Larsson, Cristian R. Rojas, and Håkan Hjalmarsson. MPC oriented experiment design. In *Proceedings of the 18th IFAC World Congress*, pages 9966–9971, 2011. QC 20111122.

[15] Lennart Ljung. *System Identification: Theory for the User*. Prentice-Hall, Upper Saddle River, New Jersey, 2nd edition, 1999.

[16] Jan Marian Maciejowski. *Predictive control: with constraints*. Prentice-Hall, pub-PH:adr, 2002.

[17] G. Picci. Metodi statistici per l'identificazione di sistemi lineari, 2011.

[18] Bo Wahlberg, Håkan Hjalmarsson, and Mariette Annergren. On optimal input design in system identification for control. In *49TH IEEE CONFERENCE ON DECISION AND CONTROL (CDC)*, pages 5548–5553. IEEE, 2010.

[19] Bo Wahlberg, Håkan Hjalmarsson, and Mariette Annergren. On optimal input design in system identification for control. In *CDC*, pages 5548–5553. IEEE, 2010.

[20] Wikipedia. Plagiarism — Wikipedia, the free encyclopedia, 2004. [Online; accessed 22-July-2004].

[21] Yucai Zhu. System identification for process control: recent experience and outlook. *International Journal of Modelling, Identification and Control*, 6, 2009.