CHALMERS
UNIVERSITY OF TECHNOLOGY

Moving Horizon Estimation for a Nonlinear Polyethylene Reactor

Master's thesis in System, Control & Mechatronics

OSKAR THOMASSON
PUMMARIN KHAMKHACHORN

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Moving Horizon Estimation for a Nonlinear Polyethylene Reactor

OSKAR THOMASSON
PUMMARIN KHAMKHACHORN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2025

Moving Horizon Estimation for a nonlinear polyethylene reactor
OSKAR THOMASSON
PUMMARIN KHAMKHACHORN

Supervisor:
Nikolce Murgovski, Professor, Department of Electrical Engineering, Chalmers
Staffan Skålén, Expert, Advanced Process Control, Borealis

Examiner:
Nikolce Murgovski, Professor, Department of Electrical Engineering, Chalmers

Master's Thesis 2025
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Gas phase reactor at Borealis with polyethylene end product.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Comparison of Moving Horizon Estimation and Single-Input Single-Output Methods for state and disturbance estimation, and formulation of Moving Horizon Estimation for solution via Sequential Quadratic Programming

OSKAR THOMASSON
PUMMARIN KHAMKHACHORN
Department of Electrical Engineering
Chalmers University of Technology

Abstract

In a closed-loop control system, measurements from sensors are used as feedback to improve the system's performance. However, these measurements can be affected by unknown disturbances or noise. Therefore, a state estimator plays a crucial role in the closed-loop system. Borealis, a polyolefins production company, currently uses a Single-Input Single-Output (SISO) estimation algorithm that updates the actual state, x, and the correction factor, e, which is an extended state included in the system model, separately. However, since some measurements are dependent on each other, it is more appropriate to use a multivariable estimation technique, such as Moving Horizon Estimation (MHE). Thus, the MHE algorithm was implemented to evaluate whether it can perform better than SISO.

Two systems were used to assess the performance of the estimators. The first is a simple system called the *Double Tank System*, and the second is a real polyethylene production system, the *Polyethylene Reactor System*, which is much more complex. The performance of MHE is clearly better than SISO for the *Double Tank System*. In the *Polyethylene Reactor System*, MHE performs at least as well as SISO but does not show a clearly better performance. Therefore, it can be concluded that MHE can match the performance of SISO, but it is not recommended to replace the SISO algorithm due to the significantly longer computation time required by MHE.

Keywords: Model Predictive Control, Moving Horizon Estimation, Sequential Quadratic Programming, Single-Input-Single-Output.

Acknowledgements

We would like to thank Staffan Skålén for his invaluable guidance and support. He helped us better understand polyethylene reactors and the application of control systems. There was a significant difference between working with a simplified model and the real-world system. We also appreciate Borealis for welcoming us as visitors. It was a great experience to see how industrial reactors operate and how professionals work in the field. Last but not least, a special thanks to Nikolce Murgovski for stepping in as our supervisor and examiner from Chalmers. His insights and discussions were invaluable to our work and played a key role in improving this thesis.

Oskar Thomasson, Gothenburg, June 2025
Pummarin Khamkhachorn, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

GPR	Gas Phase Reactor
LTI	Linear time-invariant
MHE	Moving Horizon Estimation
MPC	Model Predictive Control
NMHE	Nonlinear Moving Horizon Estimation
NMPC	Nonlinear Model Predictive Control
NLP	Nonlinear Programming
QP	Quadratic Programming
SQP	Sequential Quadratic Programming
SISO	Single-Input Single-Output

Nomenclature

Below is the nomenclature of indices, parameters, and variables that have been used throughout this thesis.

Indices

i	Indices for sample steps within the cost function
j	Indices for block number within the cost function
k	Index for current sample

Parameters

$\mathbf{Q_e}$	Weighting matrix for the process noise (for e)
q	Element in $\mathbf{Q_e}$ matrix
$\mathbf{R}$	Weighting matrix for the measurement noise
r	Element in R matrix
$\mathbf{P}$	Weighting matrix for the initial state penalty
J	Cost function or objective function
N	Length of horizon
$\mathbf{H}$	Hessian matrix in the Quadratic Programming subproblem
$c^\top$	Transpose of gradient vector in the Quadratic Programming subproblem
Δt	Time step size
r_{Dev}	Relative deviation
$eGain, xGain$	Gains used for updating the correction factor e and the state x in SISO estimator
n_{b}	Number of blocks
$n_{\text{b,len}}$	A vector represents the lengths of each block

$e_{\min}, e_{\max}$	Minimum and maximum value of the correction factor
$\Delta e_{\min}, \Delta e_{\max}$	Minimum and maximum value of the change of the correction factor
$\mathbf{q}_{\text{sens,unscale}}$	Unscaled sensitivity matrix used in the QP subproblem
$\mathbf{q}_{\text{sens,scale}}$	Scaled sensitivity matrix used in the QP subproblem
$\overline{\Delta e_{\text{opt}}}$	Δe_{opt} from previous QP iteration
$\overline{q_{\text{est}}}$	Estimated measurement from previous QP iteration
$\mathbf{e_{ss}}, \mathbf{q_{ss}}$	Scaling matrices
e_{perturb}	Predefined constant used to determine the magnitude of the perturbation

Variables

x	State
y	Output
u	Input
w	Process noise
n	Measurement noise
e	Correction factor (disturbance introduced in the system)
z	Extended state, $z = [x \quad e]^\top$
q_{meas}	Actual measurement
q_{est}	Estimated measurement
Δe	Change of the correction factor
Δe_{opt}	Change of the correction factor in reduced cost function (blocking technique)
$\Delta e_{\text{opt,scale}}$	Scaled Δe_{opt} in QP subproblem

Contents

1
Introduction

This chapter introduces the background of the project to explain why Moving Horizon Estimation is an interesting approach to explore for this application. It also includes a discussion of the ethical aspects of improving the estimation algorithm. The research questions are then presented, followed by a discussion of the limitations that define the scope of the project.

1.1 Background

A control system is designed to regulate the behavior or output of a dynamic system to achieve desired objectives. It manages the inputs of a system to produce specific outcomes while striving for stability, accuracy, and optimal performance[1]. Borealis employs a control system to control the process of producing advanced and sustainable polyolefins. The company uses nonlinear model predictive control (NMPC) to operate industrial-scale polyolefin reactors. For state and disturbance estimation, Borealis currently employs a simplified Single-Input, Single-Output (SISO) model update strategy. This is sufficient when all measurements are independent, but this strategy struggles when measurements are interdependent, leading to suboptimal estimation and reduced real-world accuracy.

In order to address this challenge, a true multivariable estimation approach should be explored, one that incorporates multiple interdependent measurements to enhance overall estimation accuracy. Moving Horizon Estimation (MHE) is one of several multivariable methods suitable for this purpose. MHE estimates system states and disturbances by solving a constrained optimization problem over a moving time horizon. It can directly integrate both physical and measurement constraints, making the estimation more robust.

1.2 Ethical Aspects

Improving precision in estimation enhances the control over polyolefin production in the factory. This improved control helps optimize the production process, which reduces the unnecessary use of energy and resources. As a result, the greenhouse gas emissions released during production are minimized, contributing to a reduction in global warming and promoting a more sustainable approach to manufacturing.

Moreover, by using energy and resources more efficiently, the same amount of material can yield a greater volume of product. This also reduces operational costs while aligning the production process with environmental and economic goals. Such improvements support sustainable development by balancing industrial growth with environmental responsibility. From an ethical perspective, improving precision in state estimation ensures that all the ingredients used in production are in optimal amounts. This guarantees the consistent quality of the products, meeting established standards and ensuring they are safe and reliable for consumers. High-quality products help reduce defective outputs, which could otherwise pose risks to consumer health and safety.

1.3 Purpose

The purpose of this project is to develop an advanced estimator based on the MHE algorithm. The primary objective of MHE is to improve the accuracy of model estimation by addressing inaccuracies inherent in the model itself. With this approach, the performance of the controller can be significantly enhanced, as the improved state estimation allows the controller to make more accurate and reliable control decisions. This will lead to more efficient system operation, reduced waste, and optimized production processes, which make production more effective and cost-efficient. Here are the research questions that this study will answer:

- How can the optimization problem of Moving Horizon Estimation be formulated to be compatible with a Sequential Quadratic Programming solver?

- Will Moving Horizon Estimation provide improved performance over Single-Input Single-Output estimation methods? What are the advantages and disadvantages of using MHE compared to SISO approaches?

1.4 Limitations

The performance of the designed estimator will be evaluated in a simulation environment. In the closed-loop control setup, the estimator's performance will be assessed in combination with the controller to evaluate the overall closed-loop control performance. Additionally, real measurement data from the industrial process will be used to evaluate the estimator's performance independently in a open-loop system. Using simulations and not tests on a running reactor ensures that the estimator can be tested under controlled and repeatable conditions, which allow a simple analysis of its behavior, accuracy, and robustness. The reason for not applying this in the real-world applications is that the estimator must first demonstrate reliable performance to ensure it does not cause any harm to the factory or its components.

2

Theory

In this chapter, information relevant to this project will be introduced and explained in detail. This includes the closed-loop control system and its components, such as the controller, estimator, and system's model. In the end of the chapter, the solver used for the optimization problem of the controller and estimator will be introduced.

2.1 Closed-Loop Control System

The purpose of a control system is to regulate the behavior of a real-world process (often referred to as the plant) to achieve desired performance, while maintaining stability and efficiency. In its simplest form, a control system consists of a controller that determines the control input to adjust the process variables (or states) so that the system output reaches the desired value. The desired output is known as the setpoint or reference, which serves as the target value that the control system aims to maintain. A system in which the control input is applied without using any feedback from the output is called an open-loop control system. In such a system, the controller does not consider whether the desired output has been achieved. It simply sends the input based on a predefined logic or schedule. In contrast, a closed-loop control system uses feedback to compare the actual output with the desired setpoint. The controller then automatically adjusts the control input to minimize the error between the actual and desired outputs[2]. This feedback mechanism significantly improves the robustness and accuracy of the control system, especially in the presence of disturbances or model uncertainties.

By integrating a mathematical model of the real-world process, the plant, into the control system, the system dynamics can be used to predict future behavior of the plant. This enables the controller to find the optimal input sequence to reach a certain setpoint. By driving the model with the same input as the process, deviations of the plant measurement and the predicted measurement can be handled by a state estimator, also called a state observer. The goal of the estimator is to determine the most likely current state given the models prediction and the measurements from the plant[3].

In a closed-loop control system, the actual output that is fed back into the system is typically measured by sensors. However, these measurements can be affected by unknown disturbances or noise. Therefore, a state estimator plays a crucial role in the closed-loop system. It estimates the true current state of the system by using

both output from the model, which represents the dynamics of the real-world system, and the sensor measurements. Based on this information, the estimator determines a more accurate estimate of the output. This estimated output is then fed to the controller, enabling more reliable and effective control despite the presence of noise and uncertainty. A holistic view of a closed-loop control system is shown in **Figure 2.1**.

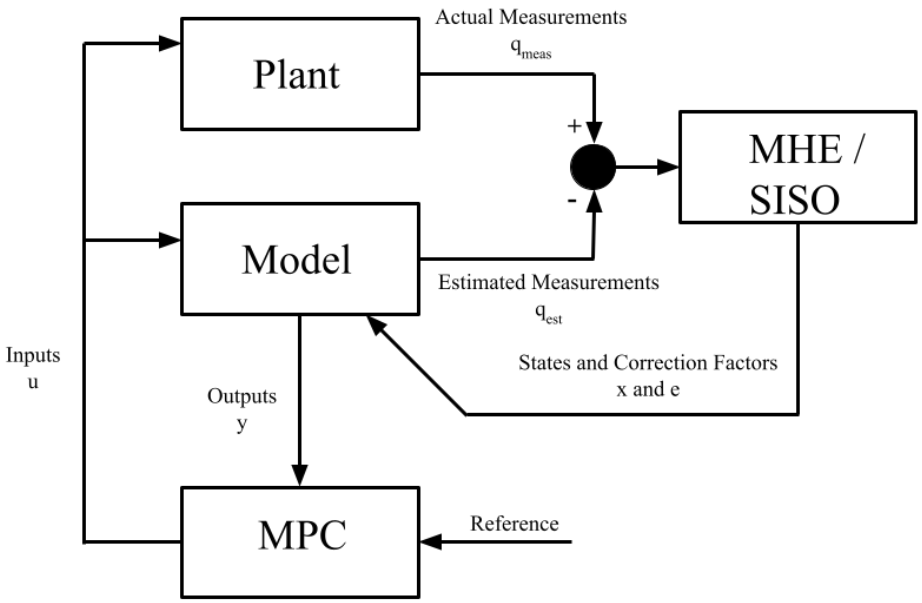


Figure 2.1: Closed-loop control system where the plant, model (used for both estimation and control of the system), estimator and controller interacts with one another.

2.2 Model and Plant

The plant is the physical process that the control system aims to regulate. It takes input in the form of control signal and outputs measurements according to its dynamics. Since it is a physical system it can be restricted by physical limits, both on input and on plant's variables. These limits are referred to as constraints.

A mathematical model in a control system aims to describe the plant process as accurately as possible. One common method of describing a system is state-space representation, where the behavior of a system is described using system variables, known as states. States are derived from the physics of the plant, often by describing the conservation of key quantities through balance equations. Some common quantities are mass, energy and momentum. The state equation in discrete time describes how states are updated as

$$x_{k+1} = f(x_k, u_k)$$

where the updated states x_{k+1} are a function f of the current states x_k and inputs u_k.

The states can the be used to calculate predict output, which refers to the output of the system. The output equation are defined by the function g

$$y_k = g(x_k, u_k)$$

where y_k is a function of the current states x_k and inputs u_k.

To reduce computational complexity and since many real-world factors are unknown, it is often necessary to make simplifications by neglecting certain dynamics, which means that the model representing the real-world behavior is imperfect[4]. This is especially important when calculations are needed in real time. Typical simplifications include linearizing nonlinear dynamics, discretizing continuous-time systems, and making assumptions such as ideal mixing or constant external conditions[5]. These simplifications, along with other external disturbances create differences between the predicted output from the model and the actual output from the plant. These differences will be used by the estimator to estimate the appropriate states.

2.3 Estimation

In control theory, estimation refers to the process of determining the current states of a system. An estimator uses the actual output from the plant and the predicted output, calculated using a model that describes the real-world system dynamics, as inputs[6]. The actual output is typically measured by sensors, but these measurements can be affected by noise, reducing their accuracy. Additionally, unknown disturbances may influence the plant, causing a mismatch between the plant and the model. Furthermore, the model itself may be simplified to reduce computational complexity, which introduce additional discrepancies. These sources of mismatch can degrade the overall performance of the control system. To address these issues, an estimator is employed to provide more accurate state estimates by comparing the actual output with the predicted output and updates its estimate based on this information. This approach will enable the controller to determine more appropriate control actions. There are several methods for state estimation. One approach is fixed-gain estimation, where the estimator uses predefined gains that do not change over time to estimate the current states. Another approach is Moving Horizon Estimation (MHE), which determines the current states by using actual output measurements over the previous N time steps.

2.3.1 Fixed-Gain Estimator

A fixed-gain estimator (also known as a predefined-gain estimator) is an estimation method in which the estimator's gain remains constant over time. The gain is typically precomputed offline using known information about the system's dynamics. This gain is used in real time to update or estimate the system states. One key advantage of this approach is its computational efficiency. The gain does not need to be recalculated at each time step, the estimator is very fast and well-suited for real-time applications with limited computational resources. However, a major

disadvantage of fixed-gain estimators is their lack of adaptability. Since the gain is fixed, the estimator can not adjust to changing system dynamics, modeling errors or unexpected disturbances, which makes it less robust in uncertain or varying environments. One approach that uses predefined gain to update the states is the *Luenberger observer*. This method estimates the system's internal states by comparing the predicted output with the actual measured output and multiplying the difference by a constant predefined gain matrix $\mathbf{L}$[7]. For linear time-invariant (LTI) systems represented in state-space form, the model update can be described as:

$$\begin{aligned} \hat{x}_{k+1} &= \mathbf{A}\hat{x}_k + \mathbf{B}u_k + \mathbf{L}\left(y_k - \hat{y}_k\right) \\ \hat{y}_k &= \mathbf{C}\hat{x}_k + \mathbf{D}u_k \end{aligned}$$

where $\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$ and $\mathbf{D}$ are the system matrices, $\hat{x}$ is estimated states and $\hat{y}$ is predicted outputs.

2.3.2 Nonlinear Moving Horizon Estimation

Nonlinear Moving Horizon Estimation (NMHE) is a multivariable state estimation technique that uses a sliding window of past measurements to estimate the system's states over a finite time horizon of length N. The sliding window moves forward at each sampling instant, keeping the number of data points constant and equal to the horizon length. At each sampling instant, NMHE solves a constrained optimization problem over this horizon by minimizing an objective function (or cost function) to provide a sequence of the optimal estimate of the system's states. However, only the last estimated state, the state at the current sample, from this sequence is applied to the system. The constraints in the optimization problem typically reflect the system's physical limitations and safety margins. The objective function in NMHE usually includes terms that penalize deviations between the system outputs and the actual measurements, as well as mismatches between the modeled and actual system dynamics[8, 9]. Let the time-invariant dynamic equations of the nonlinear modeled system in discrete time be formulated as:

$$\begin{aligned} x_{k+1} &= f\left(x_k, u_k\right) \\ y_k &= g\left(x_k, u_k\right) \end{aligned}$$

Assume that the actual dynamic system is the modeled system that is affected by process noise, w, and that measurement errors occur due to measurement noise, n

$$\begin{aligned} x_{k+1} &= f\left(x_k, u_k\right) + w_k \\ y_k &= g\left(x_k, u_k\right) + n_k \end{aligned}$$

The cost function that minimizes the process noise and measurement noise can be expressed as:

$$J = \left(x_{k-N} - x(k-N)\right)^\top \mathbf{P}\left(x_{k-N} - x(k-N)\right) + \sum_{i=k-N}^{k-1} w_i^\top \mathbf{Q} w_i + \sum_{i=k-N}^{k} n_i^\top \mathbf{R} n_i$$

The general formulation of NMHE optimization problem can be expressed as:

$$\begin{aligned}\underset{x_{k-N:x_k}}{\text{minimize}} \quad J = &\left(x_{k-N} - x(k-N)\right)^\top \mathbf{P} \left(x_{k-N} - x(k-N)\right) \\ &+ \sum_{i=k-N}^{k-1} \left(x_{i+1} - f\left(x_i, u_i\right)\right)^\top \mathbf{Q} \left(x_{i+1} - f\left(x_i, u_i\right)\right) \\ &+ \sum_{i=k-N}^{k} \left(y_i - g\left(x_i, u_i\right)\right)^\top \mathbf{R} \left(y_i - g\left(x_i, u_i\right)\right)\end{aligned} \tag{2.1}$$

subject to

$$x_{i+1} = f\left(x_i, u_i\right) + w_i \tag{2.2}$$

$$y_i = g\left(x_i, u_i\right) + n_i \tag{2.3}$$

$$x_i \in \mathbb{X},\ w_i \in \mathbb{W},\ n_i \in \mathbb{N}, \quad \forall i \in [k-N, k] \tag{2.4}$$

where

$$\begin{aligned} J &= \text{Cost function of the optimization problem} \\ \mathbf{R} &= \text{Weighting matrix for the measurement noise} \\ \mathbf{P} &= \text{Weighting matrix for the initial state penalty} \\ \mathbf{Q} &= \text{Weighting matrix for the process noise} \\ \mathbb{W} &= \text{Limitation of process noise} \\ \mathbb{N} &= \text{Limitation of measurement noise} \\ y_i &= \text{Measured output at sample } i \\ x_i &= \text{State at sample } i \\ u_i &= \text{Input at sample } i \\ n_i &= \text{Measurement noise at sample } i \\ x_{k-N} &= \text{Initial state estimate at sample } k-N \\ x(k-N) &= \text{Prior initial state at sample } k-N \end{aligned}$$

The first term in the cost function (2.1) is known as the *arrival cost*. It reflects the difference between the estimated initial state, x_{k-N}, and the prior initial state, $x(k-N)$. A larger difference indicates that the estimated state deviates more from the prior. Therefore, the arrival cost can be interpreted as a measure of how far the estimated state is from the expected or desired initial state[10]. This term is particularly important when the horizon, N, is short, since the cost of this term will dominate in the cost function. However, for longer horizons, other terms in the cost function become more significant, and this term can often be neglected. The arrival cost term is also commonly formulated as:

$$\left(x_0 - x(0)\right)^\top \mathbf{P} \left(x_0 - x(0)\right) \tag{2.5}$$

In this form, the term only accounts for the deviation at the initial sample, $i = 0$, rather than at every estimation time. It specifically measures how inaccurate the

guessed initial state, x_0, is compared to the prior estimate.

The second term in the cost function tries to penalize large values of process noise. This term prevents the optimizer from assuming large and unrealistic disturbances, which maintain the smoothness of the estimates. The last term measures how close the estimated output is to the actual measured output. Minimizing this term ensures that the system's estimation match the real measurements as closely as possible. The equations (2.2)-(2.4) represent the constraints of the problem. These constraints ensure that the optimization respects the physical and operational limits of the system and noises while estimating the states and disturbances as accurately as possible. They are crucial for ensuring the feasibility and safety of the system during the estimation process. Note that there are multiple ways to formulate the cost function. The one used in this project will be presented later in this report.

2.4 Controller

In control systems, a setpoint tracking controller is a mechanism or algorithm that aims to minimize the deviation between the system's output and a desired value, known as the reference or setpoint. The main objective of the controller is to ensure that the system behaves in a desired manner, despite disturbances and uncertainties[11]. The controller takes the estimated output of the system's model, y, and the reference signal, y_{ref}, as inputs to determine the deviation between them. The estimated output of the system's model is calculated using the outputs of the estimator, which typically include the estimated state, x. Based on these, it computes the control input, u, which is then applied to the system to drive the output toward the desired reference.

2.4.1 Nonlinear Model Predictive Control

Nonlinear Model Predictive Control (NMPC) is a type of controller designed for nonlinear systems. Similar to NMHE, NMPC solves a nonlinear optimal control problem at each sampling instant over a receding horizon, N.[12] The objective is to minimize a defined cost function, which is typically modeled to penalize deviations between the reference and system's output. At each sample instant, NMPC uses the current state to predict the future behavior of the system. It computes a sequence of optimal control inputs, u, over a finite time horizon. However, only the first control input, the input at current sample, from this sequence is applied to the system[13]. Let the time-invariant nonlinear system which depends on state, x and input, u, be modeled as:

$$x_{i+1} = f\left(x_i, u_i\right)$$
$$y_i = g\left(x_i, u_i\right)$$

The general formulation of NMPC optimization problem can be expressed as:

$$\underset{u_k : u_{k+N}}{\text{minimize}} \quad V$$

subject to

$$\begin{aligned} x_{i+1} &= f\left(x_i, u_i\right) \\ y_i &= g\left(x_i, u_i\right) \\ x_i \in \mathbb{X},\ u_i \in \mathbb{U}, &\quad \forall i \in [k, k+N] \end{aligned}$$

where V is the cost function and $\mathbb{X}$ and $\mathbb{U}$ describe the limits of the state and input, respectively.

2.5 Sequential Quadratic Programming

Sequential Quadratic Programming (SQP) is an iterative method used for solving nonlinear constrained optimization problems. The nonlinear optimization problem can be expressed as:

$$\underset{x}{\text{minimize}} \quad J$$

subject to

$$\begin{aligned} h(x) &= 0 \\ g(x) &\leq 0 \end{aligned}$$

Where J is the objective function to be optimized, x is the optimization variable, $h(x)$ represents the equality constraints, and $g(x)$ represents the inequality constraints. SQP solves a nonlinear optimization problem by approximating it as a series of Quadratic Programming (QP) subproblems[14]. This means that at each SQP iteration, one QP problem is solved. There are several ways to derive a QP subproblem, but the most common approach is to use a second-order Taylor expansion of the Lagrangian function combined with a linearization of the constraints. The Lagrangian function can be expressed as:

$$\mathcal{L}(x, \lambda, \mu) = J(x) + \sum_i \lambda_i h_i(x) + \sum_i \mu_i g_i(x)$$

The subproblem at iteration k can then be expressed as:

$$\underset{p}{\text{minimize}} \quad \frac{1}{2} p^\top \cdot \mathbf{H} \cdot p + c^\top \cdot p$$

subject to

$$\begin{aligned} \nabla h(x_k)^\top p + h(x_k) &= 0 \\ \nabla g(x_k)^\top p + g(x_k) &\leq 0 \end{aligned}$$

where $c^\top$ is the transposed gradient vector of the function f and $\mathbf{H}$ is an approximation of the Hessian of the Lagrangian

$$\begin{aligned} \mathbf{H} &= \nabla^2_{xx} \mathcal{L} \\ c^\top &= \nabla f(x_k)^\top \end{aligned}$$

It should be noted that if the Lagrangian is convex the Hessian does not need to be approximated. However, in general, the Lagrangian can be locally non-convex. Then, an approximation is needed since there the Hessian cannot be negative as it will lead the Newton direction towards a maximum, instead of minimum. The solution of the subproblem, p, will be used to update x

$$x_{k+1} = x_k - p$$

where x_k is the state of the previous iteration[15]. SQP iterates Newton's method to find critical points of the Lagrangian function to minimize the cost function. This means that SQP will perform the iteration until a termination criteria is met where the gradient of the Lagrangian is smaller than a threshold as

$$\|\nabla \mathcal{L}_k(x_k, \lambda_k, \mu_k)\| \leq \epsilon$$

where ϵ is the termination threshold.

3

Estimator Implementation and Result Interpretation in a Simulated Environment

In this chapter, an overview of the closed-loop control system implemented in this project will be presented. It provides detailed information about each estimator used in the project, beginning with the one currently implemented at Borealis, continuing with introducing the Moving Horizon Estimation method, followed by a description of the criteria used to evaluate estimator performance and how the results are interpreted.

3.1 Simulation Environment

To evaluate the performance of the implemented estimator, a simulation environment has been created, as shown in **Figure 3.1**. The control system consists of a plant, a model, an MPC controller, and an estimator, either Moving Horizon Estimation or a Single-Input Single-Output approach.

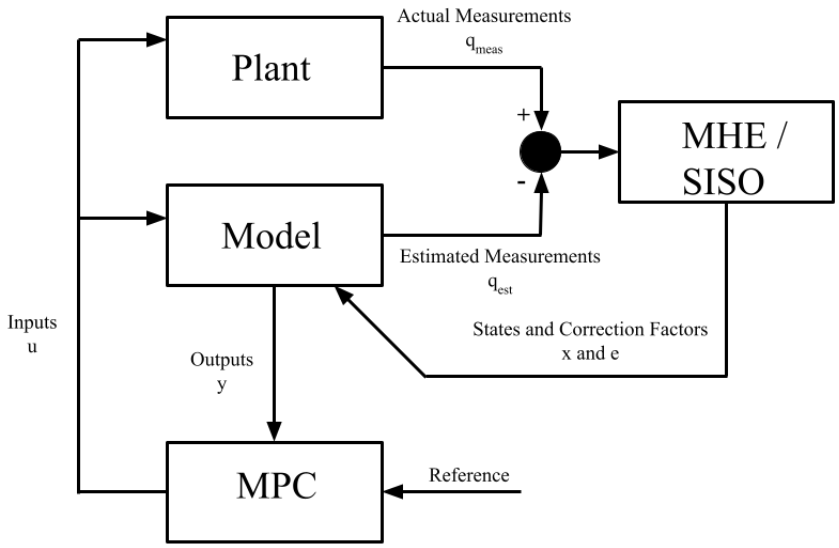


Figure 3.1: The closed-loop control system used in the project. Inputs to the estimator are the actual measurement, q_{meas}, and the estimated measurement, q_{est}. The outputs of the estimator are the updated state, x, and correction factor, e.

The plant is modeled similarly to the model, but with the addition of noise and other types of disturbances. These are deliberately introduced to create simulation scenarios in which the estimator must compensate for these imperfections. The specific scenarios will be presented in the following chapters. The model's nonlinear dynamic equations, which describe the system behavior, are based on a physical representation of the real-world process. In addition to the system states, x, and inputs, u, a disturbance variable, also referred to as a correction factor e, is included in the model's dynamics. The correction factor helps the model compensate for errors or unmeasured disturbances occurred in the plant, ensuring that the system's output reaches the desired setpoint. This is also known as implementing integral action into the model. It can be considered an extended state in the system as

$$z = \begin{bmatrix} x & e \end{bmatrix}^\top$$

The correction factor can be incorporated into the system's model either as an additive term or through multiplicative coupling with the system's actual states, x. Since the state vector is extended, the estimator must estimate or update both the actual system states and the correction factor. In this control system, the inputs to the estimator are not directly the system outputs that the controller aims to regulate. Instead, the inputs to the estimator are called *measurements*. These include q_{meas}, the actual measurement from the plant, and q_{est}, the estimated measurement from the model. These measurements are related to the system states, thus some of these variables serve as both measurements and system's outputs, y. However, some outputs are not directly measurable. In such cases, additional measurements, though not control outputs, can still be valuable for feedback, as they help estimate system states and apply necessary correction factors. This allows the estimator to utilize more information for improved state estimation. They are modeled in the same way, but q_{meas} derives from the states of the plant, while q_{est} uses the states from the model. After the estimation process, the estimated states and correction factor are fed into the model to calculate the predicted output. This output is then passed to the controller, which compares it with the reference value to determine the appropriate control input for the system. This input is then applied to the plant and model, and the cycle repeats. This creates a closed-loop cycle, where the system continuously updates its model and thus the control of the plant.

The nonlinear dynamic equations of the model, formulated to describe the system in this report, are expressed in the time domain

$$\underbrace{\begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix}}_{\dot{z}(t)} = \begin{bmatrix} f\left(x(t), e(t), u(t)\right) \\ 0 \end{bmatrix} \Leftrightarrow \dot{z}(t) = \underbrace{\begin{bmatrix} f\left(z(t), u(t)\right) \\ 0 \end{bmatrix}}_{\hat{f}(z(t),u(t))} = \hat{f}\left(z(t), u(t)\right) \tag{3.1}$$

$$q_{\text{est}}(t) = g\left(x(t), e(t), u(t)\right) = g\left(z(t), u(t)\right) \tag{3.2}$$

$$y(t) = h\left(x(t), e(t), u(t)\right) = h\left(z(t), u(t)\right) \tag{3.3}$$

The correction factor is treated as a constant variable because its future changes are unknown. Consequently, its time derivative is zero. In simulation, since the

system must be represented in the discrete domain, these continuous-time dynamics are converted to discrete-time form using Euler's method

$$\underbrace{\begin{bmatrix} x_{k+1} \\ e_{k+1} \end{bmatrix}}_{z_{k+1}} = \begin{bmatrix} x_k + f\left(z_k, u_k\right) \cdot \Delta t \\ e_k \end{bmatrix} \Leftrightarrow z_{k+1} = \underbrace{\begin{bmatrix} \tilde{f}\left(z_k, u_k\right) \\ e_k \end{bmatrix}}_{\hat{f}(z_k, u_k)} = \hat{f}\left(z_k, u_k\right) \tag{3.4}$$

$$q_{\text{est},k} = g\left(z_k, u_k\right) \tag{3.5}$$

$$y_k = h\left(z_k, u_k\right) \tag{3.6}$$

3.2 Single-Input Single-Output Estimation

The current estimation method Borealis uses is a predefined gain approach, known as Single-Input Single-Output (SISO) updating. In this approach, correction factors, e, and states, x, are updated independently. In a SISO estimator, the model updates its states and correction factors by adjusting the relative deviation, r_{Dev}. This deviation quantifies the difference between the actual measurement, q_{meas}, (from the real-world plant) and the estimated measurement, q_{est}, (calculated based on the system states).

$$r_{\text{Dev}} = \frac{q_{\text{meas}} - q_{\text{est}}}{q_{\text{meas}}}$$

As seen in the formulation above, r_{Dev} is valid only when q_{meas} is non-zero. In practice, a minimum value of q_{meas} is assigned to ensure that it is not too close to zero. The states and correction factors can be updated in two ways: additive update and multiplicative update.

Additive update at sample k is defined as:

$$\begin{aligned} e_k &= e_{\text{old},k} + eGain \cdot r_{\text{Dev}} \\ x_k &= x_{\text{old},k} + xGain \cdot r_{\text{Dev}} \end{aligned} \tag{3.7}$$

where, e_{old} and x_{old} represent the values of the correction factor and state before estimation at the corresponding sample. Here, the correction factor, e_k, and the state, x_k, are adjusted by directly adding the product of the relative deviation, r_{Dev}, and the corresponding gain, $eGain$ / $xGain$. In the case that multiple measurements are available, the relative deviation is calculated individually based on the difference between each actual measurement and estimated measurement associated with the relevant state and correction factor.

Multiplicative update at sample k is defined as:

$$\begin{aligned} e_k &= e_{\text{old},k} \cdot (1 + eGain \cdot r_{\text{Dev}}) \\ x_k &= x_{\text{old},k} \cdot (1 + xGain \cdot r_{\text{Dev}}) \end{aligned} \tag{3.8}$$

In this case, the update scales the current values based on the relative deviation.

eGain and *xGain* are predefined gains that can be chosen as desired. They can be seen as tuning parameters. The gains determine how quickly the system updates its estimates. They typically range between -1 and 1, where values close to zero, result in slower updates, and higher values closer to 1 or -1 allow faster adjustments.

The criteria for selecting the update approach depend on whether the updated variable can physically take negative values. If a negative value would be physically incorrect, the multiplicative update is used to ensure the result remains positive. On the other hand, if the variable can be negative, the additive update is applied.

3.3 Moving Horizon Estimation

In this section, the cost function used in the Moving Horizon Estimation (MHE) algorithm will be presented. Furthermore, it will be shown how the dimensionality of the cost function can be reduced to decrease the computational time required to solve the MHE optimization problem. Finally, the simplified cost function will be reformulated into a Quadratic Programming (QP) form, making it suitable for use with a Sequential Quadratic Programming (SQP) solver.

3.3.1 Cost Function and Constraints Formulation

The cost function used for the optimization problem, in both the *Double Tank System* and the *Polyethylene Reactor System*, is formulated in the same manner. These systems will later be explained in following chapters. As explained previously, the correction factor is also included in the state vector.

$$z = \begin{bmatrix} x & e \end{bmatrix}^\top$$

Therefore, the equations of the nonlinear dynamic system with the extended state are formulated as follows:

$$\begin{aligned} z_{k+1} &= \hat{f}\left(z_k, u_k\right) \\ q_{\text{est},k} &= g\left(z_k, u_k\right) \end{aligned}$$

Assume that the actual dynamic system, the plant, is the modeled system that is affected by process noises, $w = \begin{bmatrix} w_x & w_e \end{bmatrix}^\top$, and that measurement errors occur due to measurement noise, n.

$$\begin{aligned} z_{k+1} &= \hat{f}\left(z_k, u_k\right) + w_k \\ q_{\text{meas},k} &= g\left(z_k, u_k\right) + n_k = q_{\text{est},k} + n_k \end{aligned}$$

In the Nonlinear Moving Horizon Estimation (NMHE) algorithm, the optimization problem over a horizon of length, N, that minimizes the process noises and mea-

surement noise can be defined as[1]:

$$\underset{\Delta e_{k-N}:\Delta e_{k-1}}{\text{minimize}} \quad J = \sum_{i=k-N}^{k} \left(q_{\text{meas},i} - q_{\text{est},i}\right)^{\top} \mathbf{R} \left(q_{\text{meas},i} - q_{\text{est},i}\right) + \sum_{i=k-N}^{k-1} \Delta e_i^{\top} \mathbf{Q_e} \Delta e_i \tag{3.9}$$

subject to

$$q_{\text{meas},i} = g\left(x_i, e_i, u_i\right) + n_i = q_{\text{est},i} + n_i$$
$$e_{\min} \leq e_i \leq e_{\max}$$
$$\Delta e_{\min} \leq \Delta e_i \leq \Delta e_{\max}$$
$$n_i \in \mathbb{N}, \quad \forall i \in [k-N, k]$$

Note that the cost function (3.9) is different from the one introduced before in previous chapter, cost function (2.1).

$q_{\text{meas},i}$ in (3.9) is a vector containing the measurements at sample i

$$q_{\text{meas},i} = \left[q_{\text{meas},i}(1) \quad q_{\text{meas},i}(2) \ldots q_{\text{meas},i}(n_{\text{q}})\right]^{\top}, \quad n_{\text{q}} = \text{number of measurements}$$

$q_{\text{est},i}$ is a vector containing all the estimated measurements (related to states, inputs and correction factors calculated by the model at sample i

$$q_{\text{est},i} = \left[q_{\text{est},i}(1) \quad q_{\text{est},i}(2) \ldots q_{\text{est},i}(n_{\text{q}})\right]^{\top}$$

Δe_i represents a vector of the rate of change in the correction factor obtained by solving the optimization problem (3.9) at sample i

$$\Delta e_i = \left[\Delta e_i(1) \quad \Delta e_i(2) \ldots \Delta e_i(n_{\text{e}})\right], \quad n_{\text{e}} = \text{number of correction factors}$$

e_i represents a vector of correction factor at sample i

$$e(i) = \left[e_i(1) \quad e_i(2) \ldots \ e_i(n_{\text{e}})\right]$$

The weighting matrices, $\mathbf{R}$ and $\mathbf{Q_e}$, are defined as

$$\mathbf{R} = diag(r_1, r_2, \ldots, r_{n_{\text{q}}})$$
$$\mathbf{Q_e} = diag(q_1, q_2, \ldots, q_{n_{\text{e}}})$$

The first term in the cost function (3.9) represents the mismatch between the actual measurement and the estimated measurement along the horizon N. The second term is used to penalize movement in the correction factor estimates, preventing them from fluctuating excessively from sample to sample due to noise. Since the optimization problem only optimizes Δe, the responsibility for reducing the mismatch

[1]For full derivation of the cost function, please see Appendix A.1

lies entirely with the rate of change in the correction factor. The correction factor e is updated as follow:

$$e_i = e_{\text{old},i} + \Delta e_i$$

where $e_{\text{old},i}$ is the old correction factor before the estimation at sample i. Since Δe is only optimized up to sample $i = k - 1$, the last update for the state is x_k. For the update of the correction factor, the update at current sample $i = k$ corresponds to the update at the previous sample.

$$e_k = e_{k-1} = e_{\text{old},k-1} + \Delta e_{k-1}$$

3.3.2 Reducing the Computational Complexity of the Cost Function

As explained in the previous section, in the MHE algorithm, the change in correction factor is obtained by solving a constrained optimization problem over a finite horizon with length N. As the horizon length increases, the number of decision variables in the optimization problem also increases, which can lead to high computational costs and longer computational time, particularly for systems with many variables or fast sampling rates. To mitigate this issue, *blocking technique* is applied to reduce the dimension of the optimization problem. This technique divides the moving window (horizon length) into smaller blocks. Within each block, only the first variable is treated as an estimation variable, while the rest are ignored and not directly optimized. In other words, instead of optimizing the parameter at every sample along the horizon of length N, only one parameter is optimized in each block. This means that fewer parameters are necessary to perform the optimization over the entire horizon. This approach significantly reduces the number of optimizations from N to the number of blocks. As a result, the size of the optimization problem and computational time are reduced, while still using data from the entire horizon. The length of each block can be defined flexibly, but the sum of all block lengths must equal the total horizon length.

The cost function (3.9) can be reformulated by applying the *blocking technique*. Let the horizon length, N, be divided into n_b blocks, and let the new optimization parameters be denoted as Δe_opt. Thus, the correction factor is denoted as e_opt. Each element of Δe_opt corresponds to the optimal value of the change rate in correction factor for its respective block. Since only the first variable of each block is included in the optimization problem, the last estimated parameter in the sequence Δe_opt, estimation at sample $i = k-$length of the next last block, is equal to the optimized parameter at sample $i = k - 1$ (the last optimized parameter without reducing the number of optimized parameters). See **Figure 3.2** on the next page.

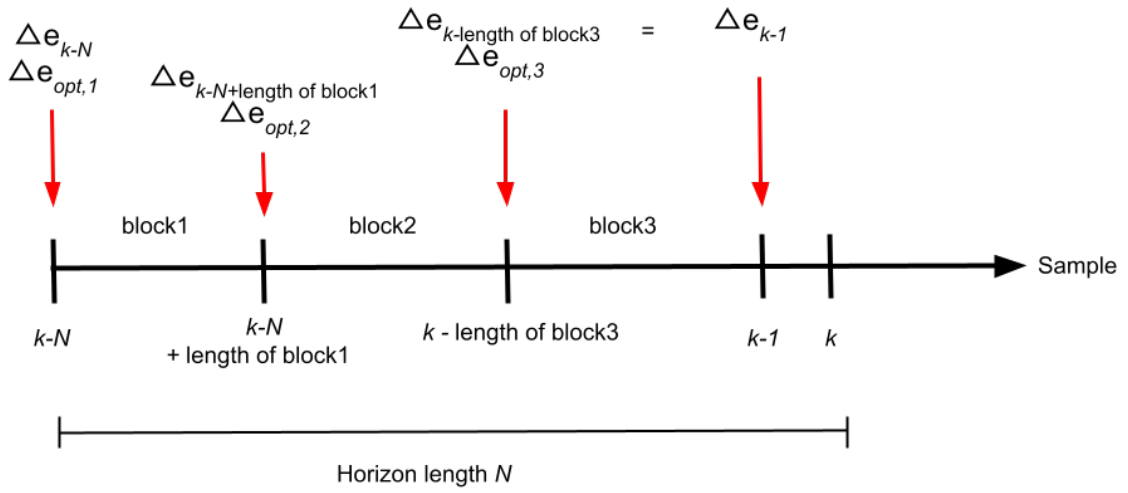


Figure 3.2: The horizon of N previous samples before the current time k,is divided into three blocks, where the variables Δe_{k-N}, $\Delta e_{k-N+\text{length of block1}}$ and $\Delta e_{k-\text{length of block3}}$ are treated as the optimized variables in the optimization problem. The change of the correction factor at sample $i = k-1$ is equal to $\Delta e_{k-\text{length of block3}}$.

The optimization problem (3.9) can be reformulated as:

$$\begin{aligned} \underset{\Delta e_{\text{opt},1}:\Delta e_{\text{opt},n_\text{b}}}{\text{minimize}} \quad J = &\sum_{i=k-N}^{k} \left(q_{\text{meas},i} - q_{\text{est},i}\right)^\top \mathbf{R} \left(q_{\text{meas},i} - q_{\text{est},i}\right) \\ &+ \sum_{j=1}^{n_\text{b}} \Delta e_{\text{opt},j}^\top \mathbf{Q_e} \Delta e_{\text{opt},j} \end{aligned} \tag{3.10}$$

subject to

$$\begin{aligned} q_{\text{meas},i} = g(x_i, e_i, u_i) + n_i = q_{\text{est},i} + n_i,\ n_i \in \mathbb{N} \qquad & \forall i \in [k-N, k] \\ e_{\min} \leq e_{\text{opt},j} \leq e_{\max} \qquad & \forall j \in [1, n_\text{b}] \\ \Delta e_{\min} \leq \Delta e_{\text{opt},j} \leq \Delta e_{\max} \qquad & j = 1 \\ \Delta e_{\min} \cdot n_{\text{b,len}}(j-1) \leq \Delta e_{\text{opt},j} \leq \Delta e_{\max} \cdot n_{\text{b,len}}(j-1) \qquad & \forall j \in [2, n_\text{b}] \end{aligned}$$

where n_b denotes the number of blocks and $n_{\text{b,len}}(j)$ represents the length of block j

$$n_{\text{b,len}} = \left[n_{\text{b,len}}(1) \quad n_{\text{b,len}}(2) \ldots n_{\text{b,len}}(n_\text{b})\right]^\top$$

$\Delta e_{\text{opt},j}$ is a vector that represents the optimal change rate of the correction factors in block j, which are obtained by solving the optimization problem (3.10) at sample $i = k - N + \sum_{b=1}^{j} n_{\text{b,len}}(b-1)$ (where block j begins)

$$\Delta e_{\text{opt},j} = \left[\Delta e_{\text{opt},j}(1) \quad \Delta e_{\text{opt},j}(2) \ldots \Delta e_{\text{opt},j}(n_e)\right]^\top$$

Note that $\sum_{b=1}^{j} n_{\text{b,len}}(b-1)$ represents the sum of the previous block length. For the first block, $j = 1$, the sum of the previous block length is set to zero, i.e. $n_{\text{b,len}}(j-1) = 0$, since the optimization starts at the beginning of the horizon.

Since the number of parameters in the optimization is decreased to n_b, it is important to limit the change rate of the correction factor. The constraint

$$\Delta e_\mathrm{min} \cdot n_\mathrm{b,len}(j-1) \leq \Delta e_\mathrm{opt,j} \leq \Delta e_\mathrm{max} \cdot n_\mathrm{b,len}(j-1)$$

ensures that the change rate stays within the allowed range, scaled by the length of the previous block. In other words, the change rate in block j must stay within the limit multiplied by the number of samples skipped since the last optimized parameter. This constraint is necessary because Δe_opt can not change too quickly, and it ensures that the solver does not propose estimates that behave too aggressively (either upwards or downwards).

For the first optimization step (i.e., when $j = 1$), there is no need to multiply by the block length since it occurs at the beginning of the horizon. Thus, the limitation remains the same, without needing to scale it by the block length

$$\Delta e_\mathrm{min} \leq \Delta e_\mathrm{opt,j} \leq \Delta e_\mathrm{max}$$

When reducing the number of optimization parameters, the number of updates for the state and correction factor is also decreased accordingly. The correction factor is updated as follows:

$$e_\mathrm{opt,i} = e_\mathrm{opt,old,i} + \Delta e_\mathrm{opt,j}$$

where $j \in [1, n_\mathrm{b}]$ and $i = k - N + \sum_{b=1}^{j} n_\mathrm{b,len}(b-1)$.

The update of correction factor at current sample $i = k$ can then be expressed as:

$$e_\mathrm{opt,k} = e_\mathrm{opt,k-1} = e_\mathrm{opt,old,k-1} + \Delta e_\mathrm{opt,}n_\mathrm{b}$$

Since the correction factor is optimized only once per block, the graph presenting (**Figure 3.3**) the correction factor along the horizon is generated using a linear update approach. This means that the optimized correction factor values at the beginning of each block are connected using linear interpolation across the blocks. Note that the correction factor in the last block remains constant, as its optimization point occurs at the beginning of the block (no more update after that sample).

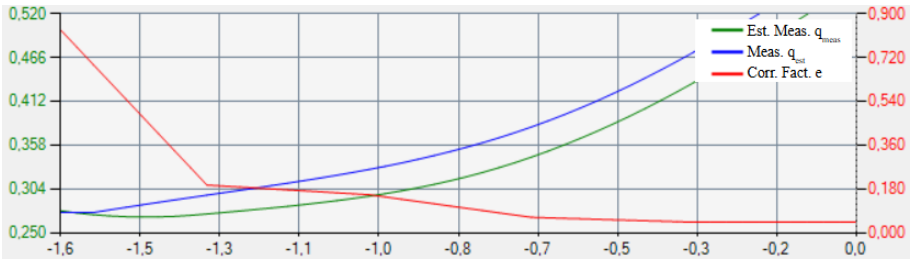


Figure 3.3: The correction factor e is updated using the linear update approach with 5 blocks. The x-axis represents time in hours, where -0.8 corresponds to 48 minutes in the past, and 0 indicates the current time. Magnitude of measurements and estimated measurements are shown on the left y-axis, correction factor magnitude is shown on the right y-axis.

3.3.3 Quadratic Programming Formulation

In this report, the optimization problem (3.10) is solved using Sequential Quadratic Programming (SQP) of Newton type to find a local minimum. To enable this, the problem must first be reformulated into a Quadratic Programming (QP) subproblem. The optimization problem (3.10) can then be expressed as follows[2]:

$$\underset{\Delta e_{\text{opt,scale}}}{\text{minimize}} \quad J = \frac{1}{2}\Delta e_{\text{opt,scale}}^{\top} \cdot \mathbf{H} \cdot \Delta e_{\text{opt,scale}} + c^{\top} \cdot \Delta e_{\text{opt,scale}} \tag{3.11}$$

subject to

$$\begin{gathered}\mathbf{max}\left(\mathbf{e_{ss}}^{-1} \cdot (e_{\min} - \overline{e_{\text{opt}}}),\ \mathbf{e_{ss}}^{-1}\left(\Delta e_{\min} \cdot n_{\text{b,len}}(j-1) - \overline{\Delta e_{\text{opt}}}\right)\right) \\ < \Delta e_{\text{opt,scale}} < \\ \mathbf{min}\left(\mathbf{e_{ss}}^{-1} \cdot (e_{\max} - \overline{e_{\text{opt}}}),\ \mathbf{e_{ss}}^{-1}\left(\Delta e_{\max} \cdot n_{\text{b,len}}(j-1) - \overline{\Delta e_{\text{opt}}}\right)\right) \\ \forall j \in [1, n_{\text{b}}],\ \text{but for } j = 1\,(\Delta e_{\text{opt,scale},1}) \longrightarrow n_{\text{b,len}}(j-1) = 1\end{gathered}$$

where

$$\begin{aligned}\mathbf{H} &= \mathbf{q_{sens,scale}}^{\top} \cdot \hat{\mathbf{R}} \cdot \mathbf{q_{sens,scale}} + \hat{\mathbf{Q}}_{\mathbf{e}} \\ c^{\top} &= \overline{\Delta e_{\text{opt}}}^{\top} \cdot \left(\mathbf{e_{ss}}^{-1}\right)^{\top} \cdot \hat{\mathbf{Q}}_{\mathbf{e}}{}^{\top} - (q_{\text{meas}} - \overline{q_{\text{est}}})^{\top} \cdot \left(\mathbf{q_{ss}}^{-1}\right)^{\top} \cdot \hat{\mathbf{R}}^{\top} \cdot \mathbf{q_{sens,scale}}\end{aligned}$$

and their dimensions are

$$\begin{gathered}\mathbf{q_{sens,scale}} \in \mathbb{R}^{n_q \cdot N \times n_e \cdot n_{\text{b}}},\ \hat{\mathbf{R}} \in \mathbb{R}^{n_q \cdot N \times n_q \cdot N},\ \hat{\mathbf{Q}}_{\mathbf{e}} \in \mathbb{R}^{n_e \cdot n_{\text{b}} \times n_e \cdot n_{\text{b}}} \\ \overline{\Delta e_{\text{opt}}} \in \mathbb{R}^{n_e \cdot n_{\text{b}}},\ \mathbf{e_{ss}} \in \mathbb{R}^{n_e \cdot n_{\text{b}} \times n_e \cdot n_{\text{b}}},\ \mathbf{q_{ss}} \in \mathbb{R}^{n_q \cdot N \times n_q \cdot N} \\ q_{\text{meas}} \in \mathbb{R}^{n_q \cdot N},\ \overline{q_{\text{est}}} \in \mathbb{R}^{n_q \cdot N},\ \Delta e_{\text{opt,scale}} \in \mathbb{R}^{n_e \cdot n_{\text{b}}}\end{gathered}$$

In this formulation, the optimization parameters are denoted as $\Delta e_{\text{opt,scale}}$, which represent the solution of QP problem and it will be used for updating Δe_{opt} and q_{est}.

$$\Delta e_{\text{opt,scale}} = [\Delta e_{\text{opt,scale},1,1} \ldots \Delta e_{\text{opt,scale},n_{\text{e}},1} \quad \Delta e_{\text{opt,scale},1,2} \ldots \Delta e_{\text{opt,scale},n_{\text{e}},n_{\text{b}}}]^{\top}$$

The first subscript of $\Delta e_{\text{opt,scale}}$ indicates the index of the correction factor element and the second subscript represents the block number. This scaling is applied to ensure that the outcomes of the QP iterations have the same order of magnitude regardless of units. The other variables introduced in this formulation are defined as follows:

$\mathbf{q_{sens,scale}}$ is the scaled version of $\mathbf{q_{sens,unscale}}$, which is a sensitivity matrix that is recalculated at every estimation step and used during each SQP iteration. It is used to define the direction of the update. It can be computed as:

$$\mathbf{q_{sens,scale}} = \underbrace{\frac{q_{\text{est,perturb}} - q_{\text{est}}}{e_{\text{perturb}}}}_{\mathbf{q_{sens,unscale}}} \cdot \frac{\mathbf{e_{ss}}}{\mathbf{q_{ss}}}$$

[2]For full derivation of QP formulation, please see Appendix A.2

Here, q_{est} is a vector of the estimated measurement at current sample instant and $q_{\text{est,perturb}}$ is a vector of perturbed estimated measurement which is calculated as:

$$q_{\text{est,perturb}} = g\left(x, e_{\text{opt}} + e_{\text{perturb}}, u\right)$$

where e_{perturb} is a vector of predefined constant used to determine the magnitude of the perturbation which has dimension $n_{\text{e}} \cdot n_{\text{b}}$. $\mathbf{e_{ss}}$ and $\mathbf{q_{ss}}$ are scaling matrices which represent e_{opt} and q_{est} corresponding to that sample instant before SQP iteration starts. They are diagonal quadratic matrices.

$\hat{\mathbf{R}}$ and $\hat{\mathbf{Q}}_{\mathbf{e}}$ are the weighting matrices corresponding to $\mathbf{R}$ and $\mathbf{Q_e}$

$$\hat{\mathbf{R}} = diag\left(\mathbf{R}_{k-N}, \ldots, \mathbf{R}_N\right)$$
$$\hat{\mathbf{Q}}_{\mathbf{e}} = diag\left(\mathbf{Q}_{\mathbf{e},1}, \ldots, \mathbf{Q}_{\mathbf{e},n_{\text{b}}}\right)$$

and $\overline{q_{\text{est}}}$ and $\overline{\Delta e_{\text{opt}}}$ define the trajectories at the previous QP iteration.

The optimization problem (3.11) is the one solved in every QP iteration. Since the solution $\Delta e_{\text{opt,scale}}$ is scaled, the unscaled update is defined as follows:

$$\begin{aligned} q_{\text{est}} &= \overline{q_{\text{est}}} + \mathbf{q_{ss}} \cdot \mathbf{q_{sens,scale}} \cdot \Delta e_{\text{opt,scale}} \\ \Delta e_{\text{opt}} &= \overline{\Delta e_{\text{opt}}} + \mathbf{e_{ss}} \cdot \Delta e_{\text{opt,scale}} \\ e_{\text{opt}} &= \overline{e_{\text{opt}}} + \mathbf{e_{ss}} \cdot \Delta e_{\text{opt,scale}} \end{aligned}$$

These values need to be updated at the end of each SQP iteration, as q_{est} and Δe_{opt} will become and $\overline{q_{\text{est}}}$ and $\overline{\Delta e_{\text{opt}}}$ for the next iteration and used to build the hessian matrix $\mathbf{H}$ and transposed gradient vector $c^{\top}$. e_{opt} is the updated correction factor.

When the SQP iterations are complete, the optimized correction factor, e_{opt}

$$e_{\text{opt}} = \left[e_{\text{opt},1,1} \ldots e_{\text{opt},n_{\text{e}},1} \quad e_{\text{opt},1,2} \ldots e_{\text{opt},n_{\text{e}},n_{\text{b}}}\right]^{\top}$$

is used to update the states, x, of the system. Note that the first subscript of e_{opt} indicates the index of the correction factor element and the second subscript represents the block number.

3.3.4 Sequential Quadratic Programming Iteration

Figure 3.4 illustrates how the Sequential Quadratic Program (SQP) iteration process works.

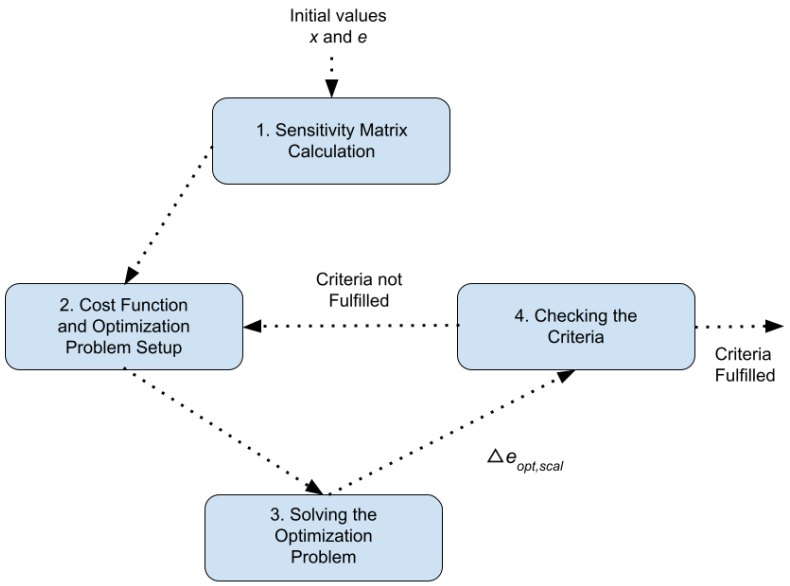


Figure 3.4: Sequential steps in the SQP algorithm, where the optimization problem is constructed, solved, and evaluated against the termination criteria.

The SQP iteration consists of the following five steps:

1. **Sensitivity Matrix Calculation:** In this step, the sensitivity matrix is calculated.

$$\mathbf{q_{sens,scale}} = \mathbf{q_{sens,unscale}} \cdot \frac{\mathbf{e_{ss}}}{\mathbf{q_{ss}}}$$

2. **Cost Function and Optimization Problem Setup:** In this step, all components of the cost function are prepared, and the hessian matrix $\mathbf{H}$ and transposed gradient vector $c^\top$ are constructed.

3. **Solving the Optimization Problem:** In this step, the optimization problem is solved, and $\Delta e_{\text{opt,scale}}$ is computed.

4. **Checking the Criteria:** In this step, the termination criteria, ϵ, is evaluated to determine whether the SQP iteration should stop.

$$\sqrt{\Delta e_{\text{opt,scale}}^\top \cdot \Delta e_{\text{opt,scale}}} \leq \epsilon$$

   If the termination criteria are fulfilled, the SQP algorithm ends. Otherwise, repeat from Step 2 using the updated variables.

3.3.5 State Update

Since the MHE algorithm only updates the correction factor, e_{opt}, these optimized values are then used to update the actual state x. The estimated state at the current sample k is updated using the values of the last block in e_{opt}, $e_{\text{opt},1,n_\text{b}} \dots e_{\text{opt},n_\text{e},n_\text{b}}$, as it corresponds to the correction factor at sample $k-1$.

$$x_k = x_{k-1} + f\Big(x_{k-1}, \underbrace{e_{\text{opt},n_\text{b}}}_{e_{\text{opt},k-1}}, u_{k-1}\Big) \cdot \Delta t$$

3.4 Performance Criteria

The criteria that will be used to evaluate the performance of the estimator are: the convergence sample, the variance of the correction factor, and the variance of the deviation between the measurement and estimated measurement.

- *Convergence sample:* indicates how quickly the outputs converges to a region around the defined reference or setpoint. In this problem, the output is considered to have converged when it enters and remains within ±1% of the reference value. A low number of convergence samples means that the system responds faster and tracks the reference more effectively.

- *Variance of the correction factor(Corr. Var):* reflects how much the correction factor fluctuates around its mean value. A high variance indicates that the correction factor oscillates significantly. This behavior is undesirable since the controller (MPC) must continuously adapt to these fluctuations, making it more difficult to maintain appropriate control performance. Variance of disturbance as evaluation criteria has been used in previous reasearch in the same area, for example in the work by Ramlal, Allsford and Hedengren[16].

- *Variance of the deviation between the measurement and estimated measurement (Dev. Var):* shows how closely the estimated measurement, q_{est}, follows the actual measurements, q_{meas}. A high variance indicates a large mismatch between the estimated measurement and the actual measurement, which suggests that the estimator is performing less effectively.

3.5 Graph Interpretation Guide

This section will provide a guide on how to interpret the graphs showing the results in the following chapters. Some graphs will have dual axes, meaning they feature two y-axes (one on the left and one on the right) sharing the same x-axis. An example of a dual-axis graph can be illustrated in following figures.

Figure 3.5 shows the system measurements: the blue line represents the estimated measurement, and the green line represents the actual measurement. Both lines correspond to the left y-axis, which is colored green. The red line represents the

correction factor associated with the measurement and corresponds to the right y-axis, which is colored red. Therefore, the color of each line indicates which y-axis it belongs to. Red line corresponds to the right y-axis (red), while green and blue lines correspond to the left y-axis (green). The x-axis represents the sample interval in the simulation.

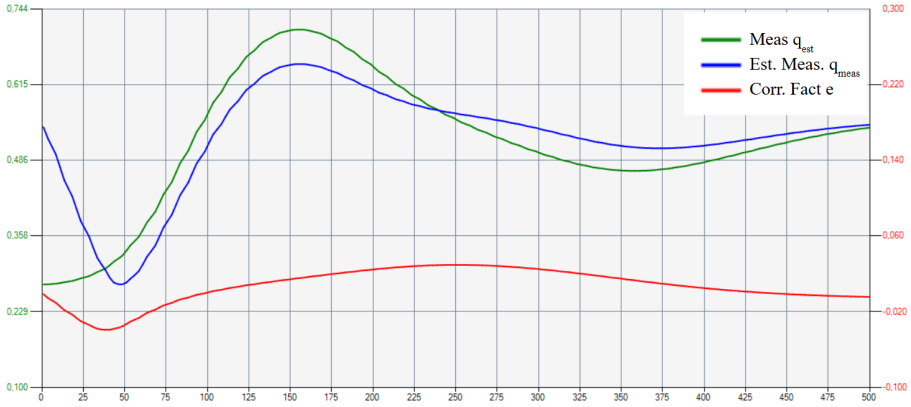


Figure 3.5: The blue and green lines, which represent the estimated and actual measurements respectively, correspond to the left y-axis. While the red line, which represents the correction factor, corresponds to the right y-axis. The x-axis represents the sample interval.

In the input-output graph, the output line is colored green and corresponds to the left y-axis, which is also colored green. The blue line represents the reference for the output and also belongs to the left y-axis. In this graph, the red line represents the input associated with the output and corresponds to the right y-axis, which is colored red. See **Figure 3.6** below.

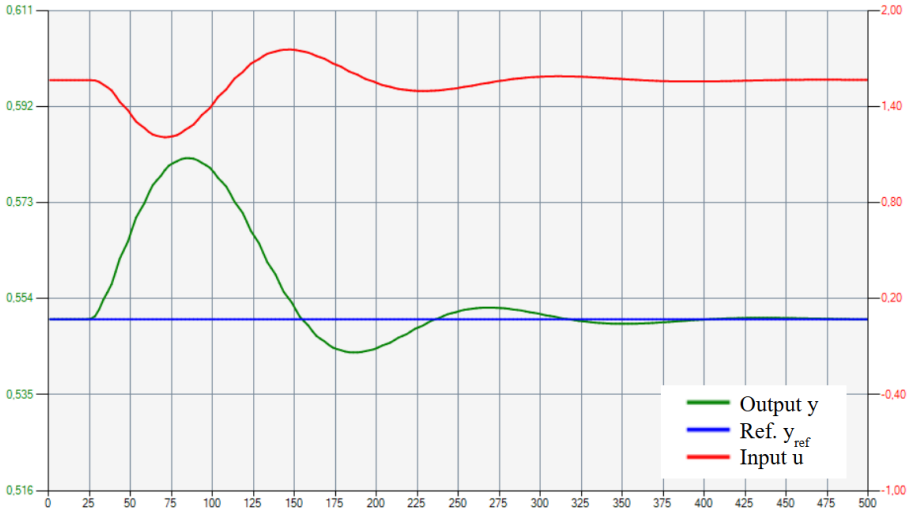


Figure 3.6: The blue and green lines, representing the reference and output respectively, correspond to the left y-axis, while the red line, representing the input, corresponds to the right y-axis. The x-axis represents the sample interval.

4

Double Tank System

This chapter will describe the *Double Tank System* and present simulations conducted to evaluate and compare the performance of the SISO and MHE approaches. The evaluation is done by finding the optimal tuning for MHE in two cases, one with initial state error and one with a model error. The performance of MHE is then compared with that of the SISO approach in the same two cases.

4.1 Problem Description of the Double Tank System

The double tank system consists of a pump connected to two tanks in series, an upper (tank 1) and a lower (tank 2), with the outflow from the lower tank exiting the system. The full system is shown in **Figure 4.1**

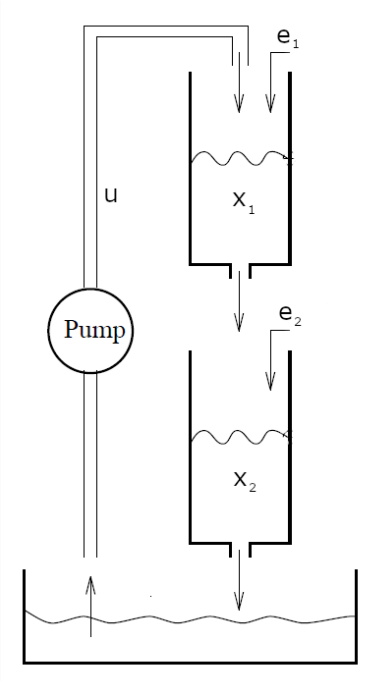


Figure 4.1: The flows between the pump and the two connected tanks in the *Double Tank System*. The disturbances e_1 and e_2 are modeled as additional inflows to respective tank.

The states representing the fluid levels in each tank are denoted as x_1 for the upper tank and x_2 for the lower tank. The changes in these levels define the state equations for the *Double Tank System*. These nonlinear equations derived from the mass balance can be expressed as

$$\dot{x_1}(t) = \underbrace{u(t)}_{\text{Inflow}} - \underbrace{\frac{A_{\text{out},1}}{A_1}\sqrt{2g \cdot x_1(t)}}_{\text{Outflow}} + e_1(t)$$

$$\dot{x_2}(t) = \underbrace{\frac{A_{\text{out},1}}{A_1}\sqrt{2g \cdot x_1(t)}}_{\text{Inflow}} - \underbrace{\frac{A_{\text{out},2}}{A_2}\sqrt{2g \cdot x_2(t)}}_{\text{Outflow}} + e_2(t)$$

where u is the flow created by the pump, $A_{\text{out},1,2}$ are the cross-sectional areas of the output holes in tank 1 and 2 respectively, $A_{1,2}$ are the cross sections of each tank and $e_{1,2}$ are the model disturbances or correction factors in the system. Both correction factors are modeled as additional inflows to each tank, i.e. as additive terms in the system's state equations. A positive disturbance corresponds to an increased inflow, whereas a negative disturbance represents an increased outflow from the corresponding tank.

The output of the system is the fluid level in the lower tank

$$y(t) = x_2(t)$$

The measurements represent the fluid levels in each tank.

$$\begin{aligned} q_{\text{meas1}}(t) = x_1(t),\ q_{\text{meas2}}(t) = x_2(t) \quad & \text{where } x_1 \text{ and } x_2 \text{ from the plant} \\ q_{\text{est1}}(t) = x_1(t),\ q_{\text{est2}}(t) = x_2(t) \quad & \text{where } x_1 \text{ and } x_2 \text{ from the model} \end{aligned}$$

There are two measurements used in the control system, q_{meas}, and two estimates of these measurements, q_{est}. They are calculated in the same way, but the estimated measurement uses the states from the model, while the actual measurement uses the states from the plant. Since the correction factors are introduced in the system, they are treated as part of the state vector. The dynamic equations above are formulated in the time (continuous) domain. To implement these equations in a simulation, they must be converted to the discrete domain. This conversion is performed using Euler's method. The discrete time dynamic equations can then be written as shown in equations (3.1)–(3.6).

$$\begin{aligned} z_{k+1} &= \hat{f}\left(z_k, u_k\right) \\ q_{\text{est},k} &= g\left(z_k, u_k\right) \\ y_k &= h\left(z_k, u_k\right) \end{aligned}$$

4.2 Test Cases and Simulation Setup

The following cases will be used to evaluate the performance of the estimators:

Case 1: Initial State Error
In this case, a closed-loop control simulation is performed with a constant output reference value of 0.55. This means that the water level in tank 2 is desired to be 55% of the tank. The model's initial states start at steady state, matching the reference. However, the plant states are initialized to half the value of the model states, which introduces an initial error. This case is therefore designed to evaluate the estimator's performance under incorrect initial state conditions.

Case 2: Model Error
This case is also a control simulation with a constant output reference value of 0.55. Both the model states and plants states are initially equal to the reference until a disturbance is introduced at sample 25 where the cross section of the outlet hole in the upper tank of the plant is enlarged by 10%. The purpose of this is to evaluate the performance of the estimator when the model's system differs from the real-world system.

Each simulation will run for 1000 samples, during which the correction factor, the deviation between the measurement and the estimated value, and the convergence sample will be recorded. The variance of both the correction factor and the deviation will then be calculated from this data. The sample time is set to one minute per sample, as the system is relatively slow and does not require high-frequency sampling.

4.3 Setup and Performance Evaluation for Single-Input Single-Output Estimation

The predefined gains for the Single-Input Single-Output (SISO) estimator, $eGain$ and $xGain$, will not be tuned, and the impact of tuning these gains will not be included in this report. The gains used here are the ones currently implemented at the company, which are already considered the best parameters. The value of these gains are presented below:

Table 4.1: The gains for the SISO estimator in the *Double Tank System*.

$xGain_1$	0.1
$xGain_2$	0.1
$eGain_1$	0.1
$eGain_2$	0.1

In this problem, the states x_1 and x_2 are updated using a multiplicative update approach, according to equation (3.8). The reason for this approach is that the

states are usefully positive as the fluid level can not be negative. The states update at sample k are:

$$x_{1,k} = x_{1,\text{old},k} \cdot \left(1 + xGain_1 \cdot \frac{q_{\text{meas1},k} - q_{\text{est1},k}}{q_{\text{meas1},k}}\right)$$
$$x_{2,k} = x_{2,\text{old},k} \cdot \left(1 + xGain_2 \cdot \frac{q_{\text{meas2},k} - q_{\text{est1},k}}{q_{\text{meas2},k}}\right)$$

and the correction factors are updated using an additive update approach, since they can be either positive or negative. According to equation (3.7), the update at sample k are:

$$e_{1,k} = e_{1,\text{old},k} + eGain_1 \cdot \frac{q_{\text{meas1},k} - q_{\text{est1},k}}{q_{\text{meas1},k}}$$
$$e_{2,k} = e_{2,\text{old},k} + eGain_2 \cdot \frac{q_{\text{meas2},k} - q_{\text{est2},k}}{q_{\text{meas2},k}}$$

Note that the denominator of the quotient can not be zero. To address this issue, a minimum values for q_{meas} are defined in case it becomes very close to zero. The performance of the SISO estimator will be presented in the following section.

4.3.1 Case 1: Initial States Error

The table below presents the performance of the SISO estimator using predefined gains in **Table 4.1**

Table 4.2: The resulting convergence sample, variance of correction factor and deviation when using the SISO-estimator in Case 1.

Convergence Sample	156
Corr. Var e_1	2.160×10^{-3}
Corr. Var e_2	2.426×10^{-3}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	1.968×10^{-4}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	2.118×10^{-4}

Figure 4.2 shows how the SISO estimator compensates for the initial states error using the correction factor. The SISO estimator only took 75 samples to converge, corresponding to 75 minutes since the sample time is one minute per sample. This indicates that the estimator performs well in correcting initial state errors. The correction factor e_2 also found its steady state around that sample.

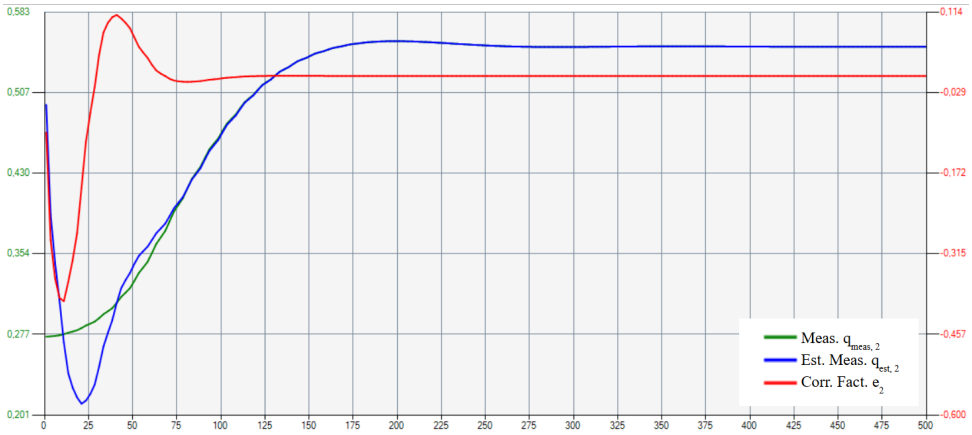


Figure 4.2: The measurement q_{meas2}, the estimated measurement from the model q_{est2} describing the filled portion of the lower tank, and the correction factor e_2 for the SISO estimator in Case 1. The figure shows the first 500 samples of the 1000 sample long simulation.

The resulting controlled output, i.e. fluid level in lower tank, is shown in **Figure 4.3**. As the fluid level in the lower tank, was lower than the setpoint, the input, fluid being pumped into the system, was increased to reach the desired setpoint. Since the plant's fluid level in the lower tank (q_{meas2}) was only half of the model's value (q_{est2}), the system first lowered the output y, to match the plant's value before tracking the setpoint.

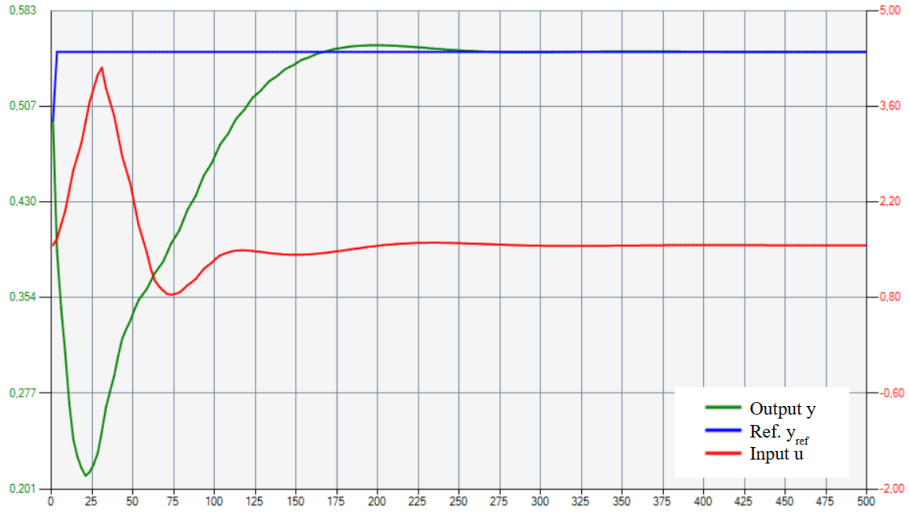


Figure 4.3: Controlled output and the reference signal describing fraction of filled water level in lower tank, and control input (pump flow) for the SISO estimator in Case 1. The figure shows the first 500 samples of the 1000 sample long simulation.

4.3.2 Case 2: Model Error

The performance evaluation results of the SISO estimator in Case 2 are shown in the table below

Table 4.3: The resulting convergence sample, variance of correction factor and deviation when using the SISO-estimator in Case 2

Convergence Sample	202
Corr. Var e_1	3.138×10^{-5}
Corr. Var e_2	3.166×10^{-5}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	6.690×10^{-7}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	6.545×10^{-7}

Figure 4.4 shows how the SISO estimator compensates for the disturbance at sample 25 using the correction factor. The estimator finds the correct estimation around sample 85, as the blue line starts matching the green line. This occurs only 50 samples after the disturbance introduced into the system.

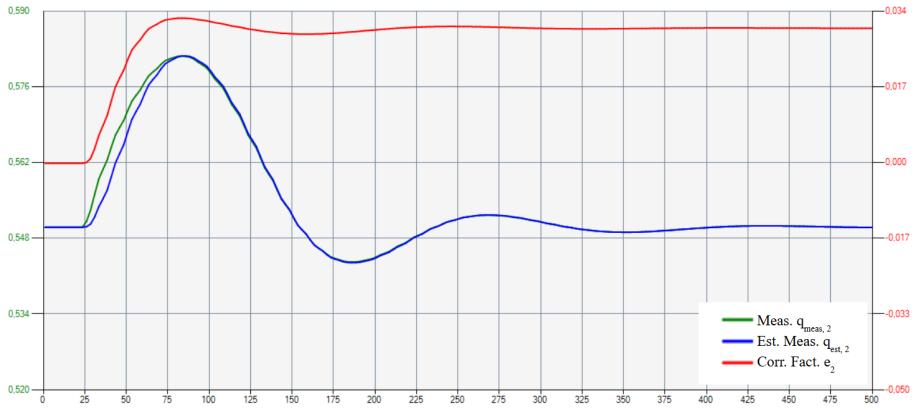


Figure 4.4: The measurement q_{meas2} of filled water level in the lower tank, the estimates of these measurements from the model q_{est2}, and the correction factor e_2 for the SISO estimator in Case 2. The figure shows the first 500 samples of the 1000 sample long simulation.

The resulting controlled output is shown in **Figure 4.5**. When the disturbance is introduced into the system at sample 25, it drives the output out of the convergence region. Even though the estimator finds the correct estimation value at sample 85, it still takes a while for the system to converge. As shown in the figure, the output starts decreasing around sample 85, where the estimator finds the correct value. After that, the output exhibits an undershoot before converging to the setpoint.

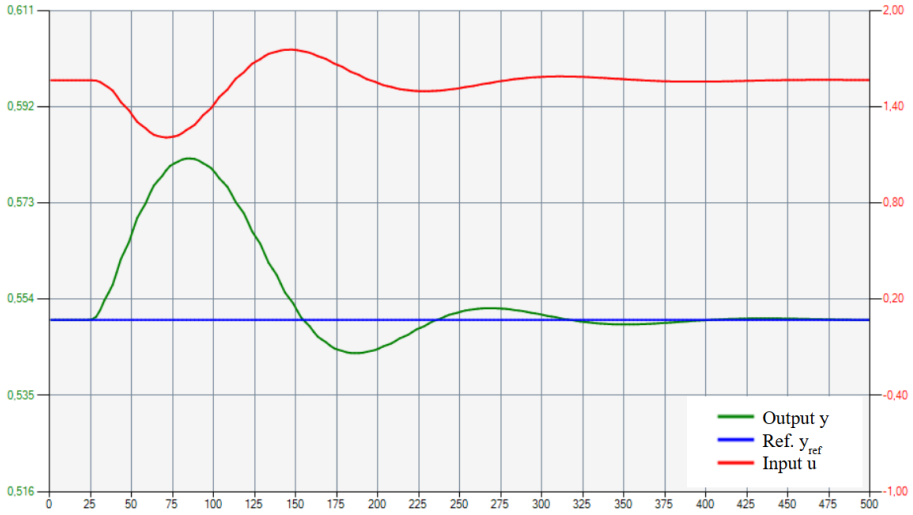


Figure 4.5: The controlled output, i.e. the filled tank portion in the lower tank, the reference signal, and control input (pump flow) when using the SISO estimator in Case 2. The figure shows the first 500 samples of the 1000 sample long simulation.

4.4 Tuning and Performance Evaluation of Moving Horizon Estimation

In the *Double Tank System*, several parameters are involved in the optimization process for MHE and they need to be defined and tuned. The parameters included are shown in **Table 4.4** below.

Table 4.4: Parameters in Moving Horizon Estimation for *Double Tank System*.

Category	Parameter	Description
To be Defined	$e_{1,\min}, e_{2,\min}$ $e_{1,\max}, e_{2,\max}$	Minimum and maximum value of correction factors e_1 and e_2
	$\Delta e_{1,\min}, \Delta e_{2,\min}$ $\Delta e_{1,\max}, \Delta e_{2,\max}$	Minimum and maximum value of the change rate in correction factor Δe_1 and Δe_2
	$e_{1,\text{perturb}}, e_{2,\text{perturb}}$	Constants used to calculate $\mathbf{q_{sens}}$ in SQP iterations
To be Tuned	$\mathbf{Q_e}$=diag(q_1, q_2)	Weighting matrix for process noise
	$\mathbf{R}$=diag(r_1, r_2)	Weighting matrix for measurement noise
	N	Horizon length
	n_b	Number of blocks
	$n_{\text{b,len}}$	Length of each block

where the defined parameters are given in **Table 4.5** below.

Table 4.5: The value of the defined parameters in MHE for *Double Tank System*.

$e_{1,\min}, e_{2,\min}$	-10
$e_{1,\max}, e_{2,\max}$	10
$\Delta e_{1,\min}, \Delta e_{2,\min}$	-0.01
$\Delta e_{1,\max}, \Delta e_{2,\max}$	0.01
$e_{1,\text{perturb}}, e_{2,\text{perturb}}$	0.01

In the tuning process, combinations of different tuning parameters will be tested. This makes it difficult to isolate and understand the effect of changing individual parameters. Therefore, a set of standard parameters is introduced, allowing one parameter to be changed at a time, to more easily determine its effect. Note that the standard values are initial estimates that are considered reasonable. The parameters of the standard tuning are given in **Table 4.6** below.

Table 4.6: Standard parameters in MHE for *Double Tank System*.

q_1, q_2	1
r_1, r_2	100
n_b	5
N	100
$n_{\text{b,len}}$	uniform distribution 20,20,20,20,20

These parameters will be used as standard values for simulation in different cases of the problem. The impact of tuning parameters will be presented in the following sections.

4.4.1 Case 1: Initial States Error

The impact of tuning different parameters in the MHE algorithm is presented below.

Weight Tuning:

The impact on convergence sample, the variance of the correction factors, and the variance of the deviation when tuning the weight matrices $\mathbf{Q_e}$ and $\mathbf{R}$ in different combinations is shown in **Table 4.7** and **Table 4.8** below. When tuning the weight matrices, the horizon length, horizon distribution, and number of blocks were kept at their standard values as shown **Table 4.6**. It is the ratio between the matrices that determines the estimator's behavior, rather than the absolute values of elements in the matrices. This means that lowering $\mathbf{Q_e}$ while keeping $\mathbf{R}$ constant yields the same result as keeping $\mathbf{Q_e}$ constant and increasing $\mathbf{R}$. The results of increasing the elements of $\mathbf{R}$, while keeping the $\mathbf{Q_e}$ matrix constant, is shown in **Table 4.7**, whereas the results of keeping $\mathbf{R}$ constant while increasing $\mathbf{Q_e}$ is shown in **Table 4.8**.

Table 4.7: Effect of tuning the elements in weighting matrix $\mathbf{R}$ while keeping $\mathbf{Q_e}$ constant on convergence sample, correction factor variance, and measurement deviation variance in Case 1. The result of the standard tuning is shown in bold text.

Weighting Matrices	$\mathbf{q_1, q_2 = 1}$ $\mathbf{r_1, r_2 = 100}$	$q_1, q_2 = 1$ $r_1, r_2 = 10^4$	$q_1, q_2 = 1$ $r_1, r_2 = 10^6$
Convergence Sample	**258**	258	258
Corr. Var e_1	$\mathbf{1.087 \times 10^{-4}}$	1.094×10^{-4}	1.094×10^{-4}
Corr. Var e_2	$\mathbf{3.181 \times 10^{-7}}$	3.593×10^{-7}	3.610×10^{-7}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	$\mathbf{3.485 \times 10^{-6}}$	3.428×10^{-6}	3.427×10^{-6}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	$\mathbf{4.432 \times 10^{-8}}$	4.769×10^{-8}	4.785×10^{-8}

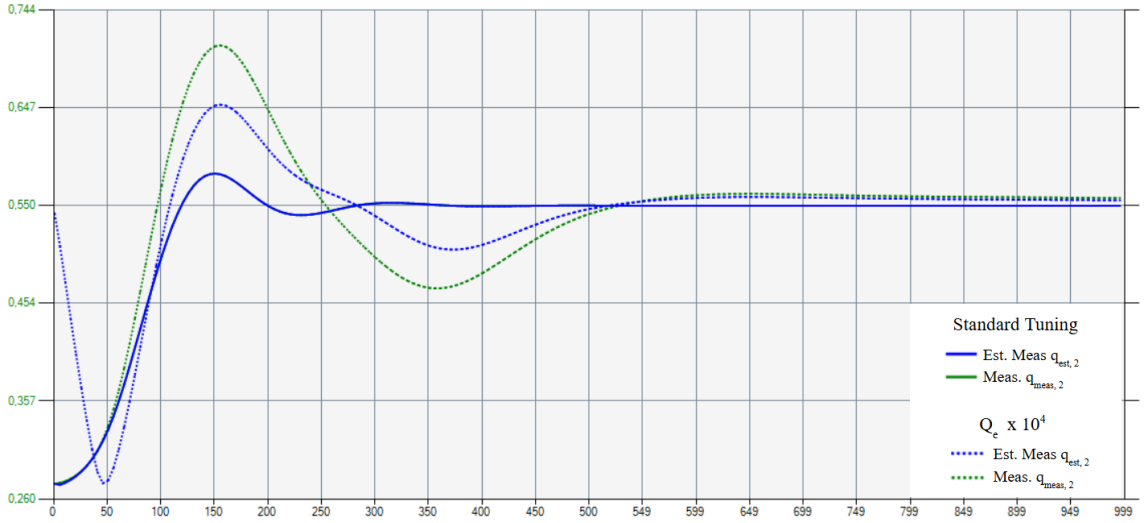
According to **Table 4.7**, the results demonstrate that increasing the values of the elements in the weighting matrix $\mathbf{R}$ does not lead to any improvement in performance compared to the standard parameter tuning. Higher ratio of $\mathbf{R}$ to $\mathbf{Q_e}$ result in almost the same variances for both the correction factors and the deviations, and the convergence sample remains unchanged. This indicates that increasing $\mathbf{R}$ beyond 100 provides no significant benefit.

The results from increasing the elements in $\mathbf{Q_e}$ show that higher values increases the correction factor and deviation variance and lead to slower convergence. When the $\mathbf{Q_e}$ values are 100 or greater, the update of the correction factor is so heavily penalized that the output does not converge within the simulation duration. The results are shown in **Table 4.8** on the next page.

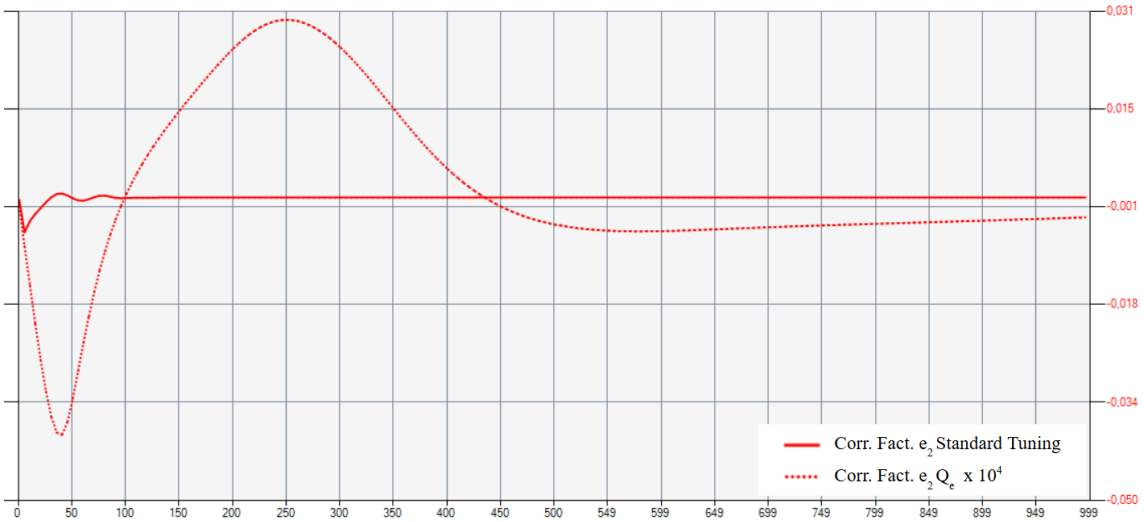
Table 4.8: Effect of tuning parameters in weighting matrices $\mathbf{Q_e}$ and $\mathbf{R}$ on convergence sample, correction factor variance, and measurement deviation variance in case 1. The result of the standard tuning is shown in bold text.

Weighting Matrices	$\mathbf{q_1}, \mathbf{q_2} = \mathbf{1}$ $\mathbf{r_1}, \mathbf{r_2} = \mathbf{100}$	$q_1, q_2 = 100$ $r_1, r_2 = 100$	$q_1, q_2 = 10^3$ $r_1, r_2 = 100$	$q_1, q_2 = 10^4$ $r_1, r_2 = 100$
Convergence Sample	**258**	264	529	No Convergence
Corr. Var e_1	$\mathbf{1.087 \times 10^{-4}}$	1.179×10^{-4}	1.354×10^{-4}	-
Corr. Var e_2	$\mathbf{3.181 \times 10^{-7}}$	3.238×10^{-6}	1.662×10^{-5}	-
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	$\mathbf{3.485 \times 10^{-6}}$	9.142×10^{-6}	1.330×10^{-4}	-
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	$\mathbf{4.432 \times 10^{-8}}$	4.305×10^{-7}	7.579×10^{-5}	-

The measurement estimates when using the standard tuning and a tuning with $\mathbf{Q_e}$-values of 10^4 are shown in **Figure 4.6 (a)**. The figure shows that a tuning with higher $\mathbf{Q_e}$ does not estimate the measurement very accurately. The estimator can not estimate correctly until around sample 500 because the penalty for changing the correction factor is higher. This is also shown in **Figure 4.6 (b)** where the correction factor does not reach a steady state value during the simulation. As a result, the output in **Figure 4.7** with the higher $\mathbf{Q_e}$ tuning also does not convergence fully during the simulation. It causes a larger overshoot of the output before the output decreases again.



(a) The true and estimated measurement describing the filled portion in the lower tank with standard tuning parameters for $\mathbf{Q_e}$ and higher values of 10^4.



(b) The corresponding correction factor with standard tuning parameters for $\mathbf{Q_e}$ and higher values of 10^4.

Figure 4.6: The measurements of the fluid level in the lower tank and correction factor e_2 using the standard tuning and a tuning with $\mathbf{Q_e}$-values of 10^4.

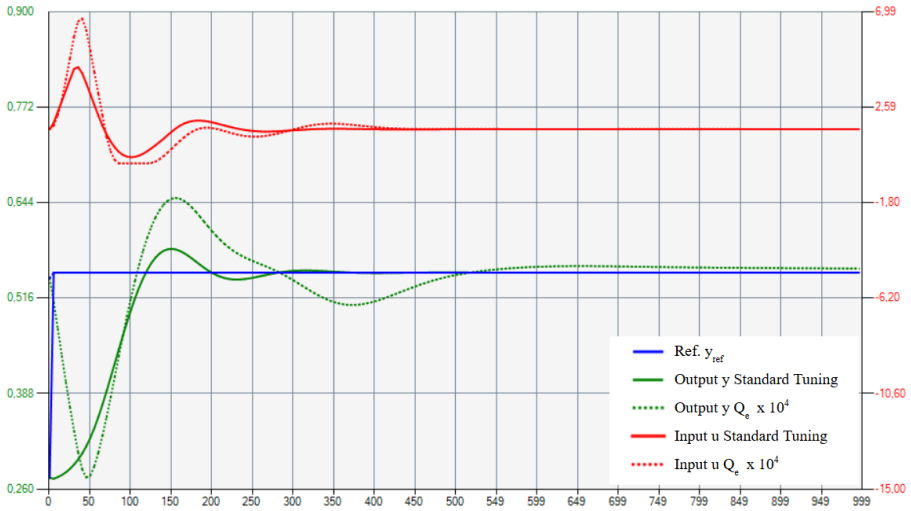


Figure 4.7: The output y describing filled portion in the lower tank using the standard tuning and a tuning with $\mathbf{Q_e}$-values of 10^4. The corresponding pump feed u is also shown.

Horizon Length:

When tuning horizon length, the other parameters are kept as standard values. The results of 4 different horizon lengths on convergence sample, correction factor variance and deviation variance are shown in **Table 4.9** and **Table 4.10** below. In **Table 4.9** all horizons are divided into 5 blocks evenly distributed whereas in **Table 4.10** all block lengths are of length 10, while the amount of blocks change between horizon lengths.

Table 4.9: Effect of tuning horizon length N, the uniformly distributed over 5 blocks, on convergence sample, correction factor variance, and measurement deviation variance. The result of the standard tuning is shown in bold text.

Horizon Length	Horizon = 10	Horizon = 50	**Horizon = 100**	Horizon = 200
Block Length	2	10	**20**	40
Convergence Sample	278	260	**258**	278
Corr. Var e_1	4.209×10^{-4}	4.462×10^{-5}	$\mathbf{1.094 \times 10^{-4}}$	2.199×10^{-4}
Corr. Var e_2	3.920×10^{-4}	8.765×10^{-7}	$\mathbf{3.593 \times 10^{-7}}$	5.778×10^{-7}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	4.048×10^{-5}	3.838×10^{-7}	$\mathbf{3.428 \times 10^{-6}}$	3.185×10^{-5}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	3.727×10^{-5}	2.378×10^{-8}	$\mathbf{4.769 \times 10^{-8}}$	5.871×10^{-7}

According to **Table 4.9**, the system achieves the fastest convergence when using a horizon length of 100 with 5 uniformly distributed blocks. Using a longer horizon,

e.g. 200, or a very short horizon, e.g. 10, does not improve the variance or convergence sample. However, when the horizon length is reduced to 50 while keeping the same number of uniformly distributed blocks, the variance of the correction factors and the deviations decreases by nearly one-tenth. Although the variance of e_2 increases slightly, the system still converges almost at the same time, only two samples slower. This result suggests that using a shorter horizon tend to advantageous, as it maintains acceptable estimation performance. Although, a longer horizon includes more historical information, it is important to consider how the number of blocks affects the recency of the last update. For example, with a horizon of 100 and 5 uniformly distributed blocks, the last update occurs 20 samples before the current sample, whereas with a horizon of 50, the last update is only 10 samples earlier. This means that, with the same number of blocks, a shorter horizon provides updates that are closer in time to the current sample, which is an important benefit. To ensure a fair comparison, the number of blocks will also be adjusted so that the last update occurs at the same sample for different horizon lengths. The results can be seen in **Table 4.10**.

Table 4.10: Effect of tuning horizon length N on convergence sample, correction factor variance, and measurement deviation variance. All blocks are of length 10, thus increasing horizon lengths increases the amount of blocks. Thereby, the last update occurs 10 samples before the current sample.

Horizon length	Horizon = 10	Horizon = 50	Horizon = 100	Horizon = 200
Number of Blocks	1	5	10	20
Convergence Sample	263	260	260	260
Corr. Var e_1	3.426×10^{-3}	4.462×10^{-5}	5.086×10^{-5}	5.082×10^{-5}
Corr. Var e_2	3.475×10^{-3}	8.765×10^{-7}	7.531×10^{-9}	6.150×10^{-9}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	1.609×10^{-4}	3.838×10^{-7}	4.164×10^{-7}	4.154×10^{-7}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	1.653×10^{-4}	2.378×10^{-8}	3.607×10^{-10}	3.087×10^{-10}

The results from **Table 4.10** show that increased horizon lengths with more blocks lower both correction factor and deviation variances. Even though, the estimator performance improves, convergence sample remains unchanged. Thus, if the main requirement is minimize the computation time and cost, it is more beneficial to use a shorter horizon and fewer blocks, as less parameters included in the optimization problem while maintaining acceptable performance. On the other hand, if low variance is also a critical requirement, it is preferable to use a longer horizon and more blocks, as this generally provides more accurate estimates. However, increasing the

horizon to 200 and the number of blocks to 20 does not yield significant improvements compared to using a horizon of 100 with 10 blocks. Therefore, a horizon length of 100 with 10 blocks can be considered a good balance between accuracy and computational efficiency. Only the correction factor variance of e_2 and the the deviation variance of $q_{\text{est2}} - q_{\text{meas2}}$ decrease as the horizon length and the number of blocks increase, starting from a horizon of 50 with 5 blocks. While the other values remain unchanged. The difference in measurement estimates between a horizon length of 50 and 200, across 5 and 20 blocks is shown in **Figure 4.8** below.

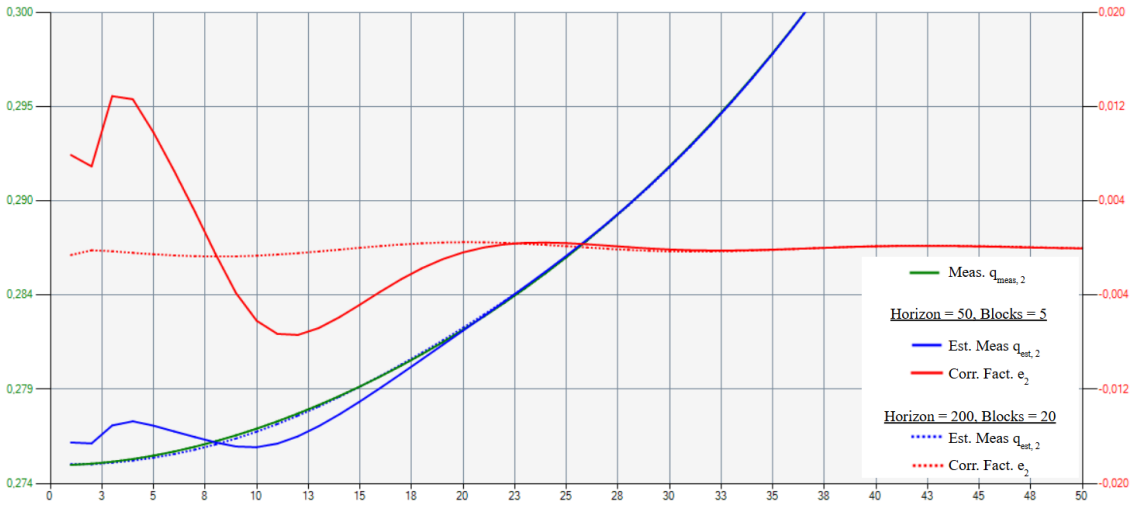


Figure 4.8: Measurement estimates describing the filled portion of the lower tank, as well as the corresponding correction factors, when using a horizon length of 50 and 200 samples, across 5 and 20 blocks. The figure shows the first 50 samples of the simulations.

The figure shows that even though the estimator performance is increased with longer horizon and more blocks, the difference is only noticeable in the first 25 samples.

Block Distribution:

The result of three different block distributions with total horizon length of 100 is shown in **Table 4.11** below.

Table 4.11: Effect of tuning block distribution with horizon length $N = 100$ on convergence sample, correction factor variance, and measurement deviation variance. The result of the standard tuning is shown in bold text.

Block length	**20, 20, 20, 20, 20**	10, 10, 20, 20, 40	40, 20, 20, 10, 10
Convergence Sample	**258**	267	267
Corr. Var e_1	$\mathbf{1.094 \times 10^{-4}}$	1.708×10^{-4}	7.610×10^{-5}
Corr. Var e_2	$\mathbf{3.593 \times 10^{-7}}$	1.435×10^{-7}	5.254×10^{-6}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	$\mathbf{3.428 \times 10^{-6}}$	1.435×10^{-5}	9.170×10^{-7}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	$\mathbf{4.769 \times 10^{-8}}$	1.101×10^{-7}	1.459×10^{-7}

The results show that modifying the block distribution do not lead to any improvements compared to the uniform distribution case. Using longer blocks toward the end of the horizon results in a slower response to initial errors. This is reasonable, as the last update occurs further away from the current sample. Therefore, if a disturbance arises at that time, the system is less able to adapt quickly. Using shorter blocks near the end of the horizon was expected to improve performance by allowing more frequent updates close to the current sample. However, this configuration actually leads to worse results, see **Figure 4.9**. The system reacts too aggressively to early errors, resulting in overshooting of the output. This also causes a larger mismatch between the estimated and actual measurement, see **Figure 4.10**.

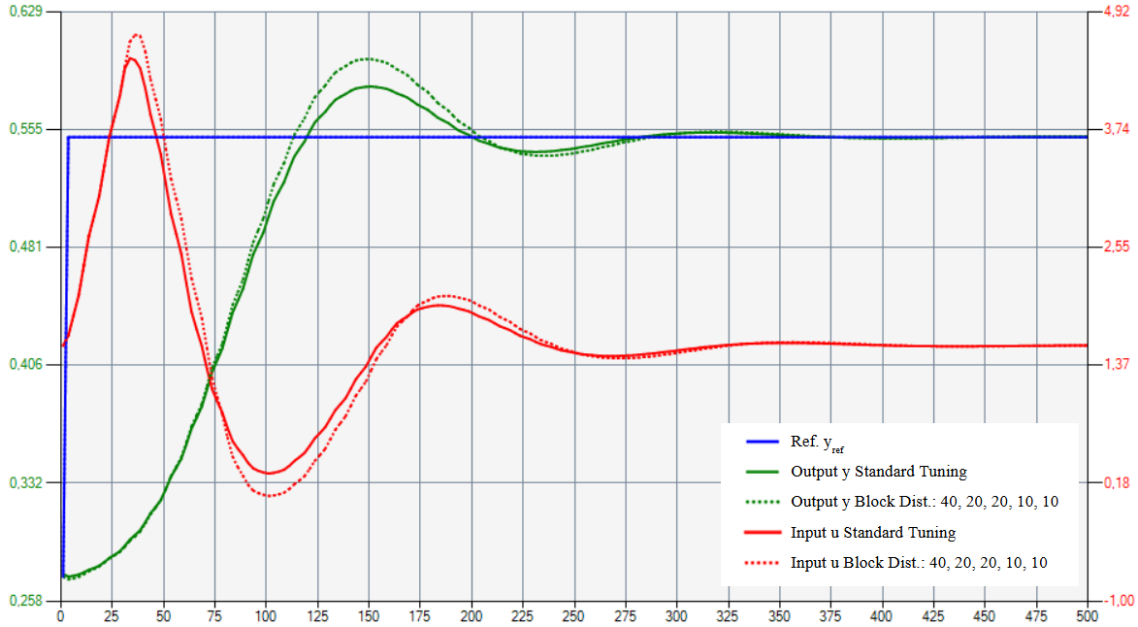


Figure 4.9: The outputs y describing the filled portion in the lower tank and corresponding inputs u (pump feed) when using standard tuning and a tuning using a block distribution with shorter blocks towards the end of the horizon in Case 1.

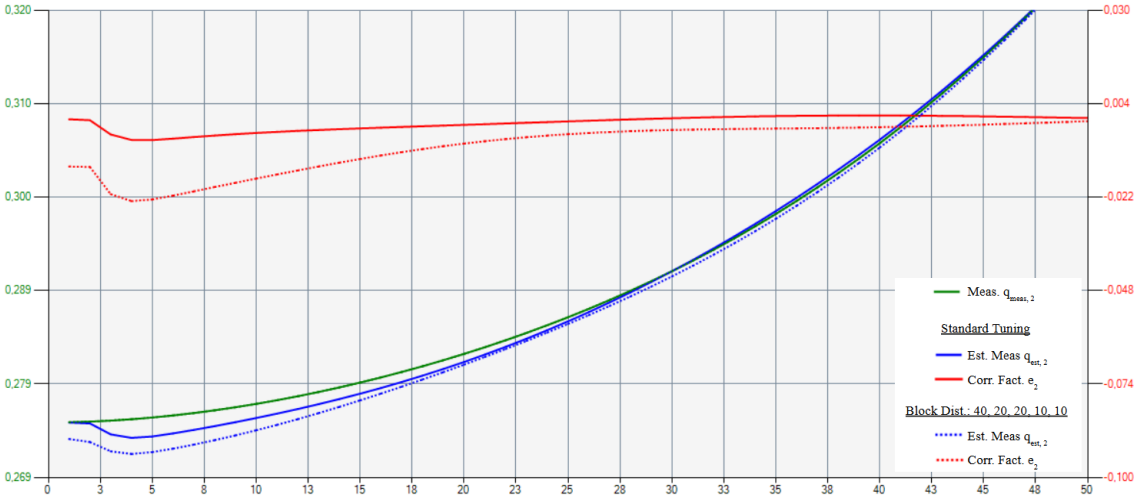


Figure 4.10: The measurements q_{meas2} describing the filled portion in the lower tank and estimates of these measurements q_{est2}, as well as correction factors when using standard tuning and a tuning using a block distribution with shorter blocks towards the end of the horizon in Case 1.

Amount of Blocks:

The results of 4 different amounts of blocks in 100 sample long horizons are shown in **Table 4.12**. The blocks in each individual horizon is of equal length.

Table 4.12: Effect of tuning number of blocks with horizon length $N = 100$ on convergence sample, correction factor variance, and measurement deviation variance. The result of the standard tuning is shown in bold text.

Number of Blocks	1	2	**5**	10
Convergence Sample	430	276	**258**	260
Corr. Var e_1	1.004×10^{-3}	1.805×10^{-4}	$\mathbf{1.094 \times 10^{-4}}$	5.086×10^{-5}
Corr. Var e_2	3.850×10^{-4}	1.525×10^{-5}	$\mathbf{2.593 \times 10^{-7}}$	7.531×10^{-9}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	6.524×10^{-4}	3.519×10^{-5}	$\mathbf{3.428 \times 10^{-6}}$	4.164×10^{-7}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	1.415×10^{-5}	9.787×10^{-6}	$\mathbf{4.769 \times 10^{-8}}$	3.607×10^{-10}

The results show that increasing the number of blocks dividing the horizon leads to improved performance. Specifically, both the correction factor and the deviation variances decrease with each additional block. A higher number of blocks means more optimization steps are performed along the horizon, which provide the estimator with more flexibility to adapt to the stored data, q_{meas}. This results in more accurate state estimation. However, the drawback is that increasing the number of blocks also increases the number of optimization parameters, which in turn raises the computational cost. The difference in output when using 1 and 5 blocks in the horizon of 100 samples is shown in **Figure 4.11** on the next page.

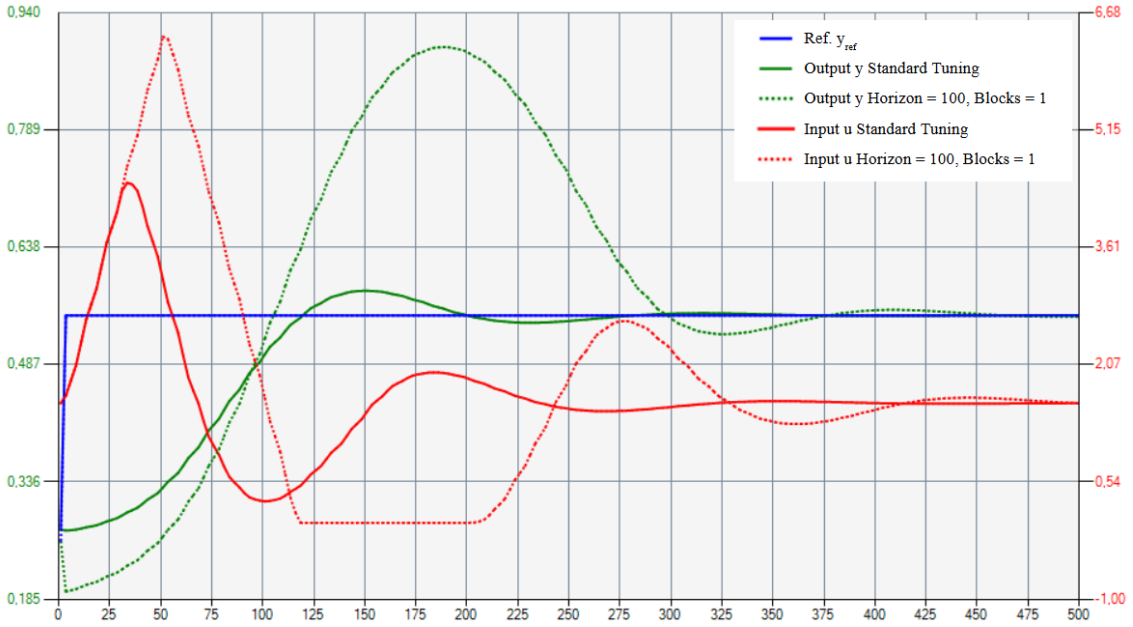


Figure 4.11: The output y depicting the filled portion of the lower tank with corresponding input u, pump feed, when using standard tuning and a tuning with the same horizon length with only 1 block.

With fewer blocks, the output results in a much larger overshoot, causing the system to converge more slowly and resulting in higher variances.

Optimal Tuning Parameters for Case 1:

From the tuning results presented in previous sections, a conclusion can be made that longer horizon and increased amounts of blocks improves the performance of the estimator in this case. The best tuning from the previous sections can be found in **Table 4.10** with the horizon length of 200. However by comparing the results in **Table 4.9** with **Table 4.12**, it is evident that increasing the number of blocks improves performance more effectively than increasing the horizon length. The tuning with horizon length 200 was therefore compared with 3 other combinations of number of blocks and horizon lengths. The results of these combinations are shown in **Table 4.13** below.

Table 4.13: Effect of tuning number of blocks with different horizon lengths on convergence sample, correction factor variance, and measurement deviation variance. The blocks in each individual horizon is of equal length. All other tuning parameters are set to the standard tuning in **Table 4.6**.

Horizon Length	200	20	40	60
Number of Blocks	20	20	20	30
Indiv. Block Length	10	1	2	2
Weighting Matrices	$q_1, q_2 = 1$ $r_1, r_2 = 100$	$q_1, q_2 = 1$ $r_1, r_2 = 100$	$q_1, q_2 = 1$ $r_1, r_2 = 10^4$	$q_1, q_2 = 1$ $r_1, r_2 = 10^4$
Convergence Sample	260	261	261	261
Corr. Var e_1	5.082×10^{-5}	1.019×10^{-5}	7.475×10^{-6}	7.518×10^{-6}
Corr. Var e_2	6.150×10^{-9}	8.484×10^{-6}	1.052×10^{-10}	1.068×10^{-10}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	4.154×10^{-7}	3.506×10^{-7}	2.317×10^{-9}	2.379×10^{-9}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	3.087×10^{-10}	6.827×10^{-10}	2.235×10^{-12}	2.730×10^{-12}

In the first combination, shown in column 3 in **Table 4.13**, where the horizon was shortened while keeping the number of blocks constant, resulted in a very small difference in performance. However, when the horizon length and thus the individual block length was doubled, the results improved significantly. Extending the horizon further and adding more blocks did not result in any further improvements. Thus the best tuning found in this case was the combination in column 3, as it achieves fast convergence while maintaining low correction factor and deviation variances.

4.4.2 Case 2: Model Error

The impact of tuning different parameters are presented below.

Weight Tuning:

The impact on converge sample, correction factor variance and deviation variance when tuning the weights matrices, $\mathbf{Q_e}$ and $\mathbf{R}$, in different combinations is shown in **Table 4.14** and **Table 4.15** below. Horizon length, horizon distribution and number of blocks were kept to the standard values shown in **Table 4.6**.

Table 4.14: Effect of increasing the elements in weighting matrix $\mathbf{R}$ while keeping $\mathbf{Q_e}$ constant on convergence sample, correction factor variance, and measurement deviation variance in Case 2. The result of the standard tuning is shown in bold text.

Weighting Matrices	$\mathbf{q_1}, \mathbf{q_2} = \mathbf{1}$ $\mathbf{r_1}, \mathbf{r_2} = \mathbf{100}$	$q_1, q_2 = 1$ $r_1, r_2 = 10^4$	$q_1, q_2 = 1$ $r_1, r_2 = 10^6$
Convergence Sample	**194**	194	194
Corr. Var e_1	$\mathbf{3.110 \times 10^{-5}}$	3.093×10^{-5}	3.093×10^{-5}
Corr. Var e_2	$\mathbf{3.083 \times 10^{-5}}$	3.067×10^{-5}	3.067×10^{-5}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	$\mathbf{3.522 \times 10^{-7}}$	3.354×10^{-7}	3.352×10^{-7}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	$\mathbf{3.208 \times 10^{-7}}$	3.059×10^{-7}	3.057×10^{-7}

Table 4.15: Effect of tuning parameters in weighting matrices $\mathbf{Q_e}$ and $\mathbf{R}$ on convergence sample, correction factor variance, and measurement deviation variance in Case 2.

Weighting Matrices	$\mathbf{q_1}, \mathbf{q_2} = \mathbf{1}$ $\mathbf{r_1}, \mathbf{r_2} = \mathbf{100}$	$q_1, q_2 = 100$ $r_1, r_2 = 100$	$q_1, q_2 = 10^3$ $r_1, r_2 = 100$	$q_1, q_2 = 10^4$ $r_1, r_2 = 100$
Convergence Sample	**194**	206	179	No Convergence
Corr. Var e_1	$\mathbf{3.110 \times 10^{-5}}$	3.802×10^{-5}	4.989×10^{-5}	-
Corr. Var e_2	$\mathbf{3.083 \times 10^{-5}}$	3.783×10^{-5}	4.823×10^{-5}	-
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	$\mathbf{3.522 \times 10^{-7}}$	1.810×10^{-6}	1.526×10^{-5}	-
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	$\mathbf{3.208 \times 10^{-7}}$	1.569×10^{-6}	1.135×10^{-5}	-

The results of increasing the elements of $\mathbf{R}$ while keeping the $\mathbf{Q_e}$ matrix constant, i.e. increasing the ratio of $\mathbf{R}$ relative to $\mathbf{Q_e}$, are shown in **Table 4.14**. The results indicate that increasing this ratio does not significantly improve performance. The convergence sample remains unchanged, while the variances of both the correction factors and the deviations decrease slightly but not substantially. On the other hand, increasing the ratio of $\mathbf{Q_e}$ relative to $\mathbf{R}$ leads to higher variances in the correction factor and the deviation. This effect is shown in **Table 4.15**. However, increasing $\mathbf{Q_e}$ to 1000 while keeping $\mathbf{R}$ at 100 results in faster convergence. Even though the

measurement estimates are not converging with the true measurements as fast as using standard tuning, shown in **Figure 4.12**, the resulting overall control output has a smaller undershoot of the reference, which makes it converge faster. See **Figure 4.13**.

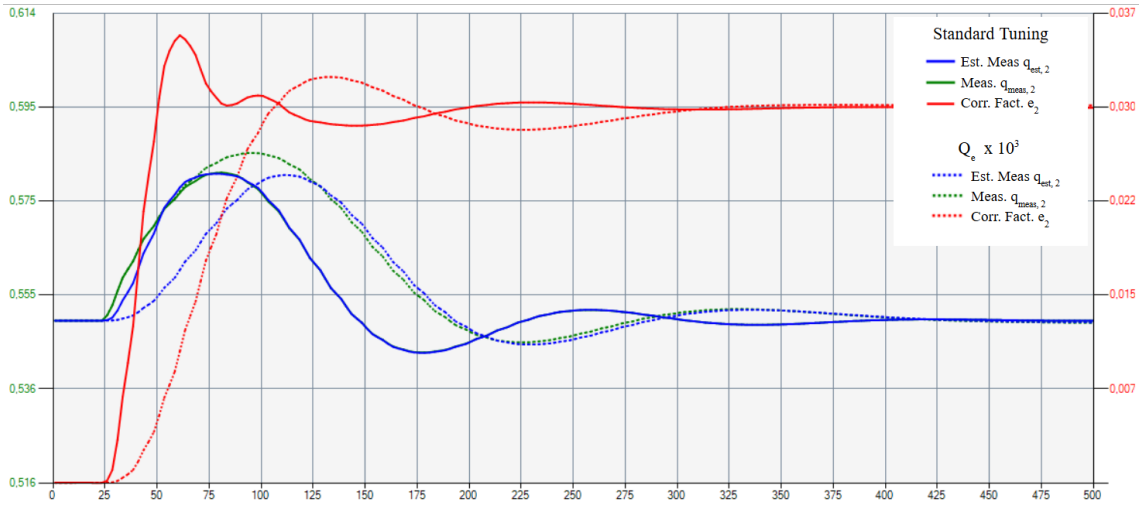


Figure 4.12: The measurements describing the filled portion of the lower tank and estimates of these measurements, as well as correction factors when using the standard tuning and a tuning with a higher $\mathbf{Q_e}$ weight of 10^3.

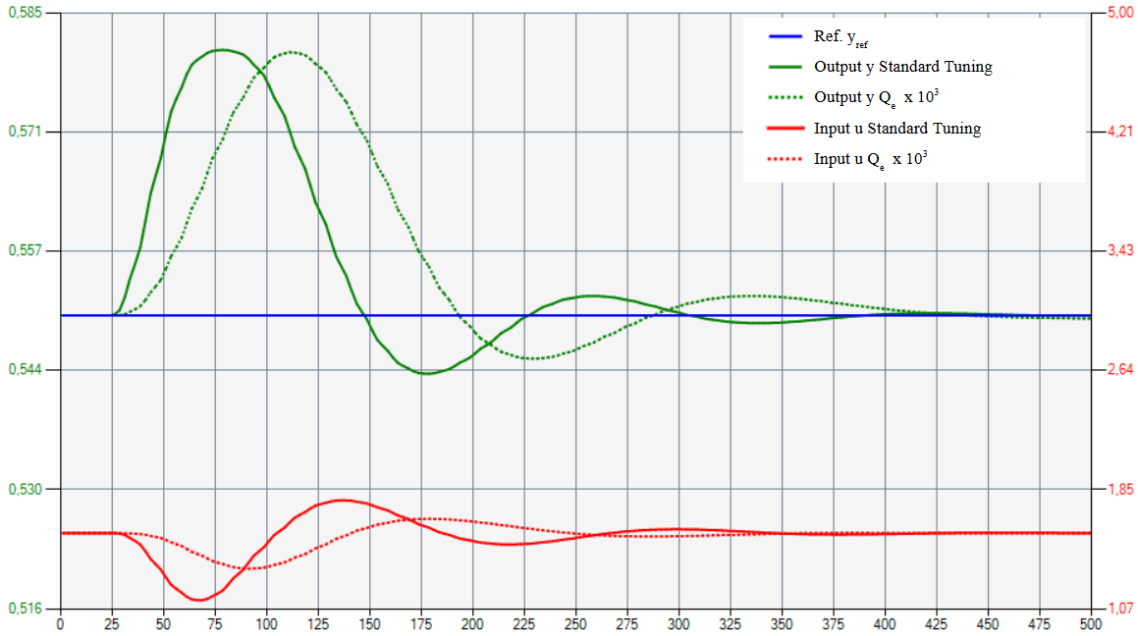


Figure 4.13: The outputs and reference describing the filled portion of the lower tank, as well as corresponding inputs using standard tuning and a tuning with a higher $\mathbf{Q_e}$ weight of 10^3.

Horizon Length:

The results for four different horizon lengths are shown in **Table 4.16** below. All horizon lengths have a uniform block distribution across 5 blocks, meaning each block within a given tuning case is of equal length. Other parameters are kept as standard values.

Table 4.16: Effect of tuning, the uniformly block distributed, horizon length N on convergence sample, correction factor variance, and measurement deviation variance in case 2. The result of the standard tuning is shown in bold text.

Horizon length	Horizon = 10	Horizon = 50	**Horizon = 100**	Horizon = 200
Convergence Sample	184	185	**194**	213
Corr. Var e_1	2.652×10^{-5}	2.725×10^{-5}	$\mathbf{3.110 \times 10^{-5}}$	4.008×10^{-5}
Corr. Var e_2	2.652×10^{-5}	2.716×10^{-5}	$\mathbf{3.083 \times 10^{-5}}$	3.909×10^{-5}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	1.418×10^{-7}	6.766×10^{-8}	$\mathbf{3.522 \times 10^{-7}}$	2.751×10^{-6}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	1.359×10^{-7}	6.421×10^{-8}	$\mathbf{3.208 \times 10^{-7}}$	2.265×10^{-6}

The results from comparing different horizon lengths show that shorter horizons generally lead to faster convergence and lower correction factor and deviation variances, except for the horizon length of 10, which resulted in slightly higher deviation variances. However, the overall trend indicates that shorter horizons yield better performance.

As explained earlier, since each horizon is divided into 5 blocks, changing the horizon length also changes the length of each block. Therefore, it is important to consider cases where the last update occurs at the same point in time. In **Table 4.17**, the results for four different horizon lengths are presented, with block lengths adjusted to ensure that the last update occurs 10 samples before the current sample.

Table 4.17: Effect of tuning horizon length N on convergence sample, correction factor variance, and measurement deviation variance. All blocks in all the horizon lengths have a length of 10 samples.

Horizon length	Horizon = 10	Horizon = 50	Horizon = 100	Horizon = 200
Number of Blocks	1	5	10	20
Convergence Sample	186	185	185	185
Corr. Var e_1	2.628×10^{-5}	2.725×10^{-5}	2.726×10^{-5}	2.726×10^{-5}
Corr. Var e_2	2.622×10^{-5}	2.716×10^{-5}	2.717×10^{-5}	2.717×10^{-5}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	4.813×10^{-8}	6.766×10^{-8}	6.769×10^{-8}	6.769×10^{-8}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	4.600×10^{-8}	6.421×10^{-8}	6.425×10^{-8}	6.425×10^{-8}

When all the different horizon lengths use the same block lengths, the results are very similar. With a horizon length of 10, the system converges one sample slower but yields slightly lower variances for both the correction factor and the deviation. Therefore, using a longer horizon and more blocks does not provide any significant benefit, as it only increases the computational time and cost.

Block Distribution:

The results for three different block distributions over a 100-sample-long horizon are shown in **Table 4.18** below, one with uniform distribution, one with longer blocks at the end of the horizon, and one with shorter blocks at the end. Weight matrices $\mathbf{Q_e}$ and $\mathbf{R}$, as well as number of blocks in the horizon were kept to the standard values.

Table 4.18: Effect of tuning block distribution with horizon length $N = 100$ on convergence sample, correction factor variance, and measurement deviation variance. The result of the standard tuning is shown in bold text.

Block length	**20, 20, 20, 20, 20**	10, 10, 20, 20, 40	40, 20, 20, 10, 10
Convergence Sample	**194**	206	185
Corr. Var e_1	$\mathbf{3.110 \times 10^{-5}}$	3.649×10^{-5}	2.758×10^{-5}
Corr. Var e_2	$\mathbf{3.083 \times 10^{-5}}$	3.587×10^{-5}	2.746×10^{-5}
Dev. Var $q_{est1} - q_{meas1}$	$\mathbf{3.522 \times 10^{-7}}$	1.500×10^{-6}	7.677×10^{-8}
Dev. Var $q_{est2} - q_{meas2}$	$\mathbf{3.208 \times 10^{-7}}$	1.286×10^{-6}	7.282×10^{-8}

The results show that longer blocks at the end of the horizon increase the correction factor variance, deviation variance, and convergence sample compared to a uniform distribution. In contrast, shorter blocks towards the end of the horizon decrease all these values. This is reasonable, as the final update occurs closer to the current sample, allowing the system to better adapt to more recent information. This suggests that simply adjusting the block distribution can enhance system performance without increasing computational time or cost. The different outputs and measurement estimates using standard tuning and the tuning with shorter blocks toward the end of the horizon are shown in **Figure 4.14** and **Figure 4.15** below.

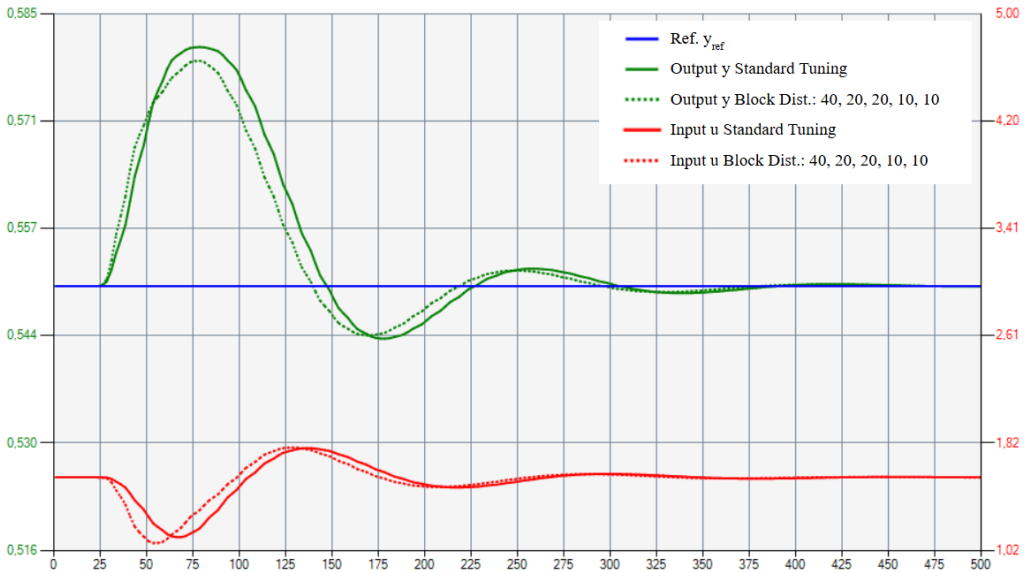


Figure 4.14: The outputs describing the filled portion in the lower tank and inputs describing the pump feed when using the standard tuning and a tuning with a horizon with shorter blocks toward the end. Both horizons are of length 100.

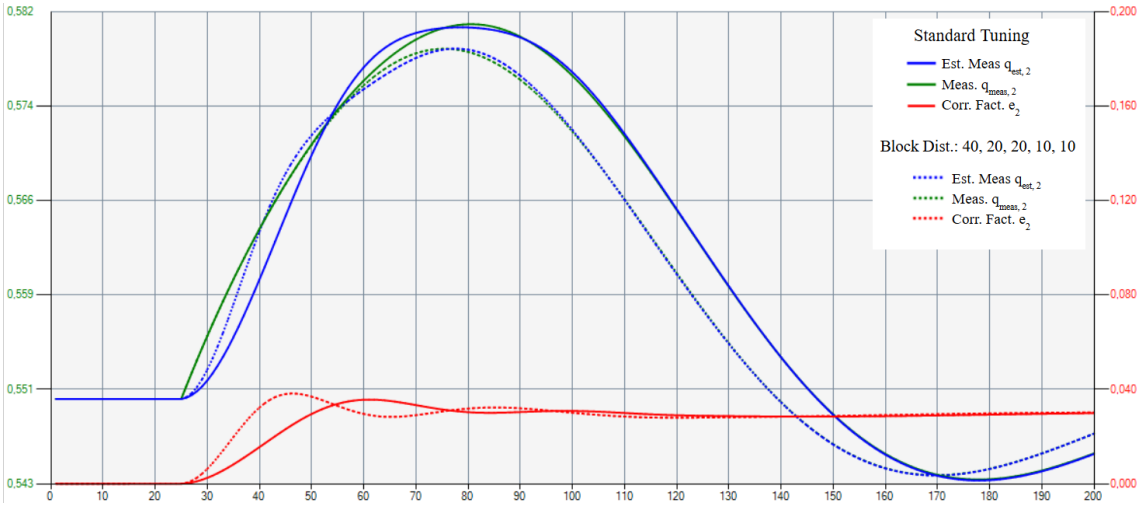


Figure 4.15: Measurements of the filled portion in the lower tank, along with their estimates and correction factors, using two tunings: standard tuning and a tuning with shorter blocks toward the end of the horizon. Both horizons are of length 100.

The figures show improved estimates of the measurements when using a tuning with shorter blocks toward the end of the horizon. The tuning makes the estimator react faster to the disturbance, while also converging to the setpoint faster, resulting in a better overall control performance.

Amount of Blocks:

The results of four different amounts of blocks with horizon length of 100 samples are presented in **Table 4.19** below. Other parameters are kept as standard values.

Table 4.19: Effect of tuning number of blocks with horizon length $N = 100$ on convergence sample, correction factor variance, and measurement deviation variance. The result of the standard tuning is shown in bold text.

Number of Blocks	1	2	**5**	10
Convergence Sample	185	221	**194**	185
Corr. Var e_1	5.850×10^{-5}	4.300×10^{-5}	$\mathbf{3.110 \times 10^{-5}}$	2.726×10^{-5}
Corr. Var e_2	5.445×10^{-5}	4.174×10^{-5}	$\mathbf{3.083 \times 10^{-5}}$	2.717×10^{-5}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	2.309×10^{-5}	4.548×10^{-6}	$\mathbf{3.522 \times 10^{-7}}$	6.769×10^{-8}
Dev. Var $q_{est,2} - q_{meas,2}$	1.415×10^{-5}	3.613×10^{-6}	$\mathbf{3.208 \times 10^{-7}}$	6.425×10^{-8}

When comparing different numbers of blocks within the horizon, increasing the number of blocks reduced both the deviation and correction factor variances. Additionally, results with more blocks showed faster convergence. This is reasonable, as a higher number of blocks introduces more parameters into the optimization problem, allowing for more accurate estimation. However, using a single block led to the same convergence time as using ten blocks. This indicates a trade-off: if there are requirements on how much the correction factor is allowed to oscillate or how closely the estimated measurement should track the actual measurement, then a higher number of blocks may be preferred. The different outputs and measurement estimates using standard tuning and the tuning with 10 blocks in the horizon are shown in **Figure 4.16** and **Figure 4.17** below.

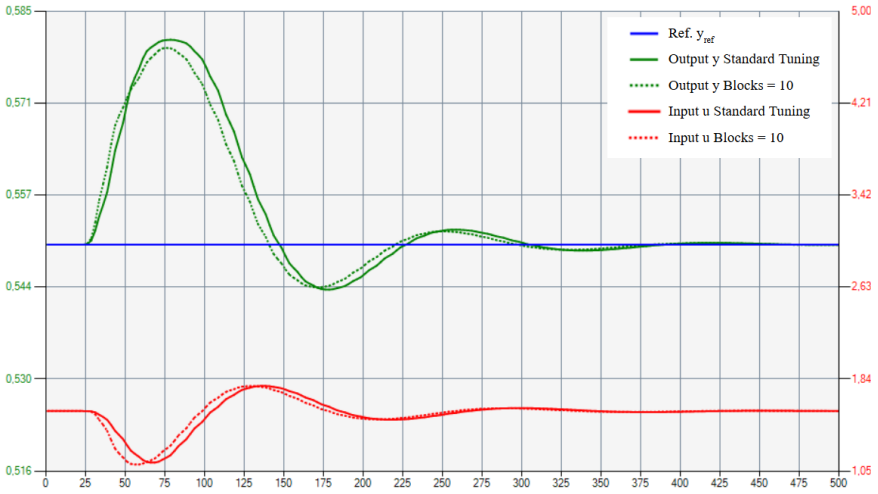


Figure 4.16: Outputs depicting the filled portion in the lower tank and inputs (pump flow) when using standard tuning and a tuning with 10 blocks in the horizon. Both horizons are of length 100 with uniform block distribution.

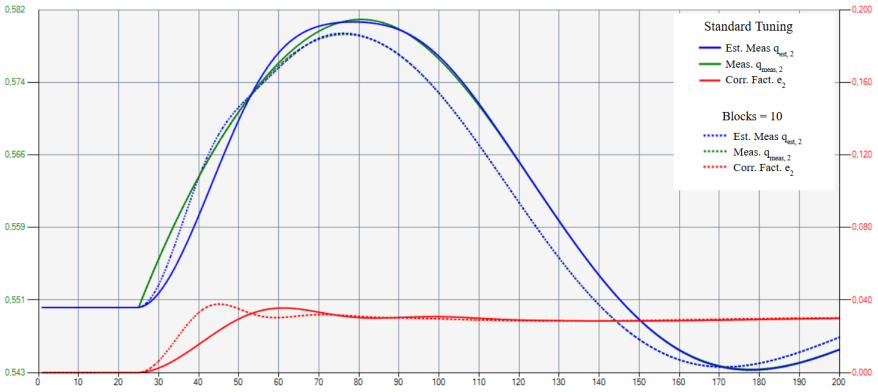


Figure 4.17: Measurements depicting the filled portion of the lower tank and the estimates of these, as well as correction factors using standard tuning and a tuning with 10 blocks in the horizon. Both horizons are of length 100 with uniform block distribution.

Optimal Tuning Parameters for Case 2:

Based on the previous results from tuning the parameters of the moving horizon estimation algorithm, four parameter combinations were selected to determine the optimal tuning for Case 2. Since the trends in horizon length, number of blocks, and block distribution indicated that shorter horizon length improved the results, the first combination was set to the shortest possible horizon length of 1. To stabilize the estimator behavior, the elements of weighting matrix $\mathbf{R}$ was increased in the second combination. Lastly, the number of blocks was increased to 5 and then to 10, while keeping the block lengths constant at 1. This means that the horizon length was also increased to 5 and 10, respectively. The purpose of this setup was to investigate whether increasing the number of blocks, even with fixed short block lengths, would further stabilize the estimator. Since the blocks were of the shortest possible length, resulting in a very short total horizon length, different block distributions would have a very small impact on the result. This can be seen as the block lengths are uniformly distributed. The results of the four combinations is shown in **Table 4.20** below.

Table 4.20: Effect of four different tuning combinations on convergence sample, correction factor variance, and measurement deviation variance.

Weighting Matrices	$q_1, q_2 = 1$ $r_1, r_2 = 100$	$q_1, q_2 = 1$ $r_1, r_2 = 10^4$	$q_1, q_2 = 1$ $r_1, r_2 = 10^4$	$q_1, q_2 = 1$ $r_1, r_2 = 10^4$
Number of Blocks	1	1	5	10
Horizon length	1	1	5	10
Convergence Sample	304	175	176	176
Corr. Var e_1	6.290×10^{-5}	2.345×10^{-5}	2.293×10^{-5}	2.293×10^{-5}
Corr. Var e_2	5.201×10^{-5}	2.342×10^{-5}	2.292×10^{-5}	2.293×10^{-5}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	9.652×10^{-6}	4.703×10^{-9}	7.060×10^{-10}	7.069×10^{-10}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	8.391×10^{-6}	4.669×10^{-9}	6.980×10^{-10}	6.987×10^{-10}

The first combination, shown in the second column in **Figure 4.20**, did not result in any improvements, as the estimator behavior was too aggressive. Both the correction factor variance and deviation variance increased, along with a later convergence sample, compared to the previous results. However increasing the elements of $\mathbf{R}$ from the standard values of 100 to 1000 improved the results drastically. By adding blocks of length 1 to the horizon, the deviation variance decreased while keeping the same convergence sample and correction factor variance as the previous

combination. Thereby, having more blocks than 5 in the horizon did not result in any improvements. Thus, the parameter combination shown in column 3 is selected as the optimal configuration for this case, as it achieves fast convergence while maintaining low correction factor and deviation variances.

4.4.3 Optimal Tuning Parameters for Moving Horizon Estimation in the Double Tank System

To determine the overall optimal tuning for MHE, each optimal tuning from the individual cases was applied to the other case. The result of applying the optimal tuning for Case 1 in Case 2 is shown in **Table 4.21**, with the optimal tuning for Case 2 provided as a reference.

Table 4.21: Results of applying the optimal tuning in Case 1 to Case 2 problem on convergence sample, correction factor variance, and measurement deviation variance. The optimal tuning for Case 2 is provided as reference marked in bold text.

Horizon length	**5**	40
Number of Blocks	**5**	20
Weighting Matrices	$\mathbf{q_1}, \mathbf{q_2} = \mathbf{1}$ $\mathbf{r_1}, \mathbf{r_2} = \mathbf{10^4}$	$q_1, q_2 = 1$ $r_1, r_2 = 10^4$
Convergence Sample	**176**	176
Corr. Var e_1	$\mathbf{2.293 \times 10^{-5}}$	2.302×10^{-5}
Corr. Var e_2	$\mathbf{2.292 \times 10^{-5}}$	2.301×10^{-5}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	$\mathbf{7.060 \times 10^{-10}}$	7.173×10^{-10}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	$\mathbf{6.980 \times 10^{-10}}$	7.071×10^{-10}

The results show that the optimal tuning parameters for Case 1 yield very similar correction factor and deviation variances, as well as convergence sample, compared to the optimal tuning for Case 2 in Case 2 problem. Therefore, it can not be determined which combination performs better. The results of applying the optimal tuning for Case 2 in Case 1 problem is shown in **Table 4.22**.

Table 4.22: Results of applying the optimal tuning of Case 2 to Case 1 on convergence sample, correction factor variance, and measurement deviation variance. The optimal tuning for Case 1 is provided as reference marked in bold text.

Horizon length	**40**	5
Number of Blocks	**20**	5
Weighting Matrices	$\mathbf{q_1}, \mathbf{q_2} = \mathbf{1}$ $\mathbf{r_1}, \mathbf{r_2} = \mathbf{10^4}$	$q_1, q_2 = 1$ $r_1, r_2 = 10^4$
Convergence Sample	**261**	259
Corr. Var e_1	$\mathbf{7.475 \times 10^{-6}}$	2.291×10^{-3}
Corr. Var e_2	$\mathbf{1.052 \times 10^{-10}}$	2.291×10^{-3}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	$\mathbf{2.317 \times 10^{-9}}$	1.548×10^{-4}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	$\mathbf{2.235 \times 10^{-12}}$	1.629×10^{-4}

Applying the optimal tuning parameters from Case 2 to the Case 1 problem results in poorer performance. Although the convergence is slightly faster and the computation time is reduced, due to a shorter horizon and fewer blocks, the correction factor and deviation variances are significantly higher compared to using the optimal parameters for Case 1. The difference in measurement estimates for Case 1 using the optimal tuning and the optimal tuning for Case 2 is shown in **Figure 4.18**.

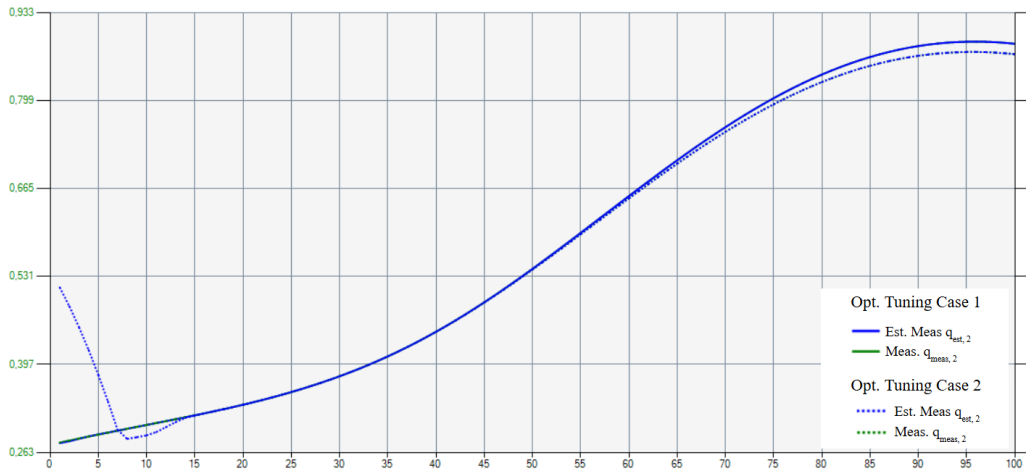


Figure 4.18: Difference in estimates of the filled portion measurements in the lower tank for Case 1 using the optimal tuning for Case 1 and the optimal tuning for Case 2. The figure displays the first 100 samples of the simulation.

As shown in the figure, the optimal tuning for Case 1 estimates the measurements faster than the optimal tuning for Case 2, as the blue line consistently aligns with the green line from the start of the estimation. Since the MPC controller prefers lower deviations and reduced oscillation in the correction factor, the optimal parameters for Case 1 are chosen as the best parameters for HME in *Double Tank System*.

Table 4.23: Best tuning parameters for *Double Tank System*.

q_1, q_2	1
r_1, r_2	1000
n_b	20
N	40
$n_\mathrm{b,len}$	uniform distribution

4.5 Performance Evaluation and Analysis of the Estimators

In this section, the performance of each estimator, MHE and SISO, will be compared for each test case to evaluate their effectiveness.

4.5.1 Case 1: Initial States Error

The results of comparing the two estimator algorithms for Case 1 in the *Double Tank System* are presented in **Table 4.24** below.

Table 4.24: The impact of the two estimators on convergence sample, correction factor variance, and measurement deviation variance in Case 1.

Estimator	SISO	MHE
Convergence Sample	156	261
Corr. Var e_1	2.160×10^{-3}	7.475×10^{-6}
Corr. Var e_2	2.426×10^{-3}	1.052×10^{-10}
Dev. Var $q_{\mathrm{est1}} - q_{\mathrm{meas1}}$	1.968×10^{-4}	2.317×10^{-9}
Dev. Var $q_{\mathrm{est2}} - q_{\mathrm{meas2}}$	2.118×10^{-4}	2.235×10^{-12}

The results in the table above suggest that while MHE results in lower correction factor and deviation variances, but leads to significantly slower convergence. The

system output, reference signal, and input for both estimators in Case 1 are presented in **Figure 4.19** below. The corresponding estimated measurements and true measurements are shown in **Figure 4.20** along with the correction factor.

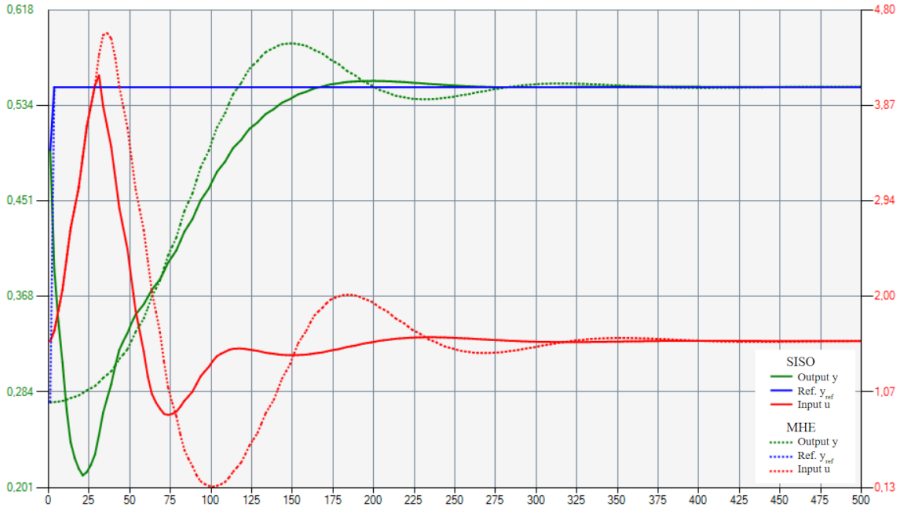
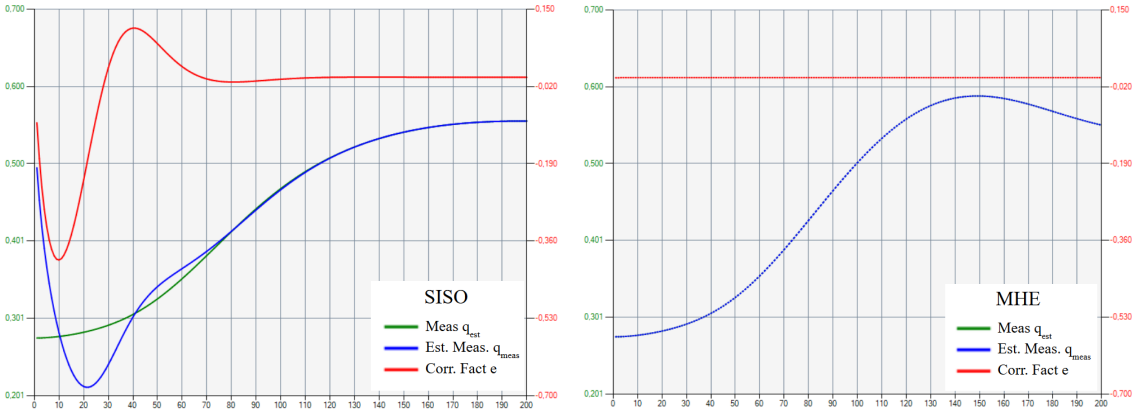


Figure 4.19: Shows the output and reference depicting the filled portion of the lower tank, as well as input (pump flow) for SISO (solid) and MHE (dotted) in Case 1. The figure shows the first 500 samples of the 1000 sample long simulation.



(a) True measurements, estimated measurements and correction factor when using the SISO estimator for Case 1.

(b) True measurements, estimated measurements and correction factor when using the MHE estimator for Case 1.

Figure 4.20: The true and estimated measurements depicting the filled level in the lower tank when simulating Case 1 with SISO and MHE. The figures shows the first 200 samples of the 1000 sample long simulation.

Figure 4.19 shows that, although MHE quickly estimates the true measurements and reaches the setpoint earlier than the SISO, it causes the output to overshoot the reference, resulting in slower convergence because the output leaves the convergence

region before it enters again and remains within it. Comparing the measurements, q_{meas} and q_{est}, in **Figure 4.20**, it is clear that the estimated measurements converge with the true measurements faster when using MHE. When using the SISO estimator in **Figure 4.20 (a)**, it takes around 70-80 samples for the estimated measurements to converge with the true measurements. This is further supported by the correction factor, shown in red in the same figure, which also reaches a steady state after about 70–80 samples into the simulation. However when using MHE, shown in **Figure 4.20 (b)**, the estimated measurements closely match the true measurements already from the beginning and the correction factor remains nearly constant throughout the simulation. This indicates that MHE provides better estimation performance, as it quickly reaches the steady state for the correction factor, resulting in very low correction factor variance.

These findings indicate that the controller is too aggressive for fast estimation, leading to an overshoot of the reference signal. This can be observed as the output reaches the reference quickly but exceeds the convergence region before settling back within it, leading to an overshoot. However, since the SISO estimator updates the estimated states, x, and correction factors, e, more slowly than MHE, it inadvertently compensates for the controller's limitations. It can therefore be concluded that MHE provides better estimation performance in general. However, this advantage is not fully realized when combined with the current controller tuning. To fully leverage MHE's capabilities, the controller can be tuned to better adapt to the estimator's behavior.

4.5.2 Case 2: Model Error

The results of comparing the two estimator algorithms for Case 2 in the *Double Tank System* are presented in **Table 4.25** below.

Table 4.25: The impact of the two estimators on convergence sample, correction factor variance, and measurement deviation variance in Case 2.

Estimator	SISO	MHE
Convergence Sample	202	176
Corr. Var e_1	3.138×10^{-5}	2.302×10^{-5}
Corr. Var e_2	3.166×10^{-5}	2.301×10^{-5}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	6.690×10^{-7}	7.173×10^{-10}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	6.545×10^{-7}	7.071×10^{-10}

The results in **Table 4.25** show that MHE provides more accurate measurement estimates, as evidenced by the reduced variance in both the correction factor and the deviation. Particularly, the deviation variance decreases significantly. The system also converges faster when using MHE, indicating that the MHE estimator outperforms the SISO estimator in this case. The simulations with each estimator are compared in **Figures 4.21** and **Figure 4.22** below.

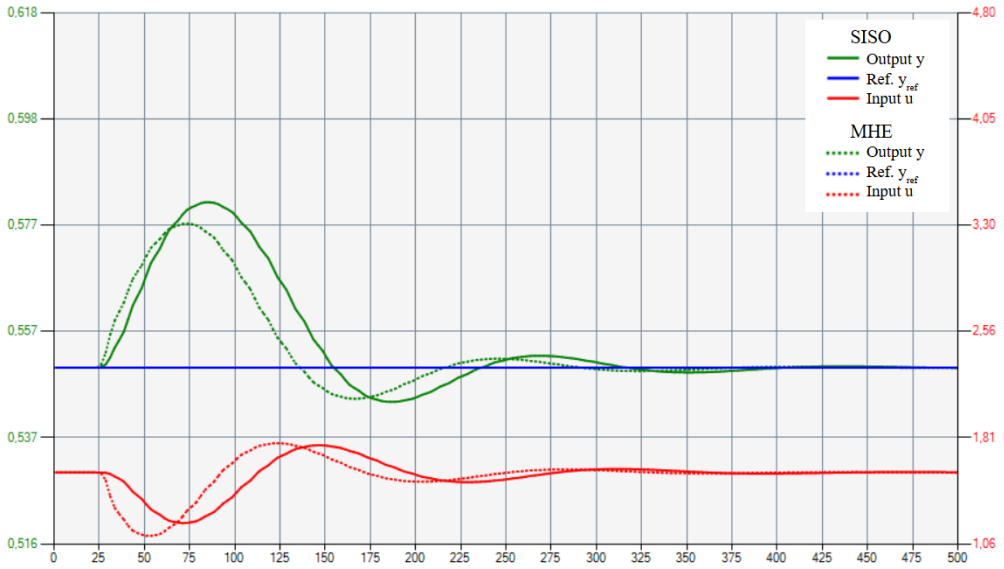
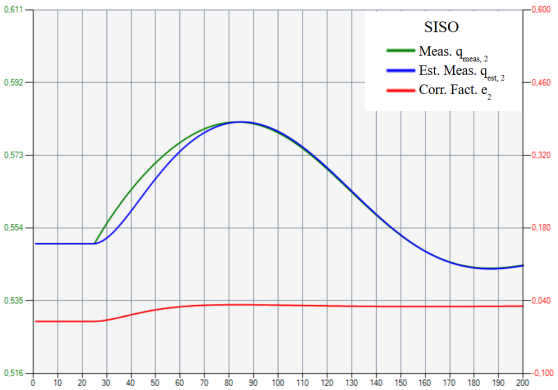
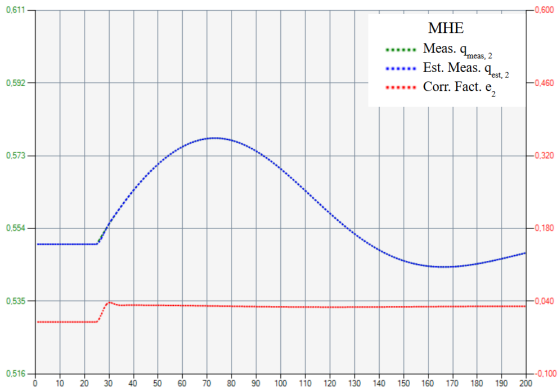


Figure 4.21: Shows the output and reference describing the filled portion of the lower tank, as well as input (pump flow) for SISO (solid) and MHE (dotted) in Case 2. The figure shows the first 500 samples of the 1000 sample long simulation.



(a) True measurements, estimated measurements and correction factor when using the SISO estimator for Case 2.



(b) True measurements, estimated measurements and correction factor when using the MHE estimator for Case 2.

Figure 4.22: The true and estimated measurements describing the filled portion of the lower tank when simulating Case 2 with SISO and MHE. The figures shows the first 200 samples of the 1000 sample long simulation.

In **Figure 4.22 (b)**, the estimated measurements follow the true measurements very accurately and the correction factor reaches steady state more quickly than with the SISO estimator shown in **Figure 4.22 (a)**. MHE quickly adapts to the model error introduced at sample 25, as the estimated measurement closely matches the actual measurement, as the green doted line is barely visible. When using the SISO estimator the estimated measurements are lower than the true measurement from when the error occur at sample 25 to around sample 70. This deviation shows that the input, u, with the SISO estimator shown in **Figure 4.21** responds less aggressively and slower to the error compared to MHE, resulting in slower convergence and higher variances.

4.6 A New Strategy of Implementing the Correction Factor

The correction factors used in this control system are implemented in a general and somewhat non-precise way, with one correction factor affecting each measurement as an additional inflow. This is a very effective way of dealing with disturbances since almost all disturbances can ultimately be described as an inflow/outflow. An alternative approach to estimating disturbances in the system is to implement the correction factors across multiple measurements, where the purpose of the correction factor is not general, but specific to a certain type of disturbance. For example, in the *Double Tank System*, a correction factor can be implemented directly to the outlet hole from the upper tank to the lower tank. In other words, the correction factor becomes a factor multiplied with the flow between the tanks, instead of a general inflow/outflow to the tank. The model's dynamic equations can then be formulated as

$$\dot{x_1}(t) = \underbrace{u(t)}_{\text{Inflow}} - \underbrace{\frac{A_{\text{out},1}}{A_1}\sqrt{2g \cdot x_1(t)}}_{\text{Outflow}} \cdot (1 + e_1(t))$$

$$\dot{x_2}(t) = \underbrace{\frac{A_{\text{out},1}}{A_1}\sqrt{2g \cdot x_1(t)}}_{\text{Inflow}} \cdot (1 + e_1(t)) - \underbrace{\frac{A_{\text{out},2}}{A_2}\sqrt{2g \cdot x_2(t)}}_{\text{Outflow}} + e_2(t)$$

where the same correction factor, e_1, affects both states, which means it also impacts both measurements, since the measurements correspond to the states x_1 and x_2 as

$$q_1 = x_1$$
$$q_2 = x_2$$

A comparison was made between the disturbance model used in this report and the disturbance model shown above. The tuning applied to the new disturbance model was set to the same as the optimal tuning from the performance evaluation of MHE in the *Double Tank System* given in **Table 4.23**. The results of simulations made in both Case 1 and Case 2 are shown below.

Table 4.26: The resulting convergence sample, deviation- and correction variance when using the two disturbance models in Case 1. Both estimators use the same tuning.

Disturbance Model	New	Old
Convergence Sample	245	261
Corr. Var e_1	6.934×10^{-5}	7.475×10^{-6}
Corr. Var e_2	5.212×10^{-6}	1.052×10^{-10}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	8.754×10^{-7}	2.317×10^{-9}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	2.078×10^{-9}	2.235×10^{-12}

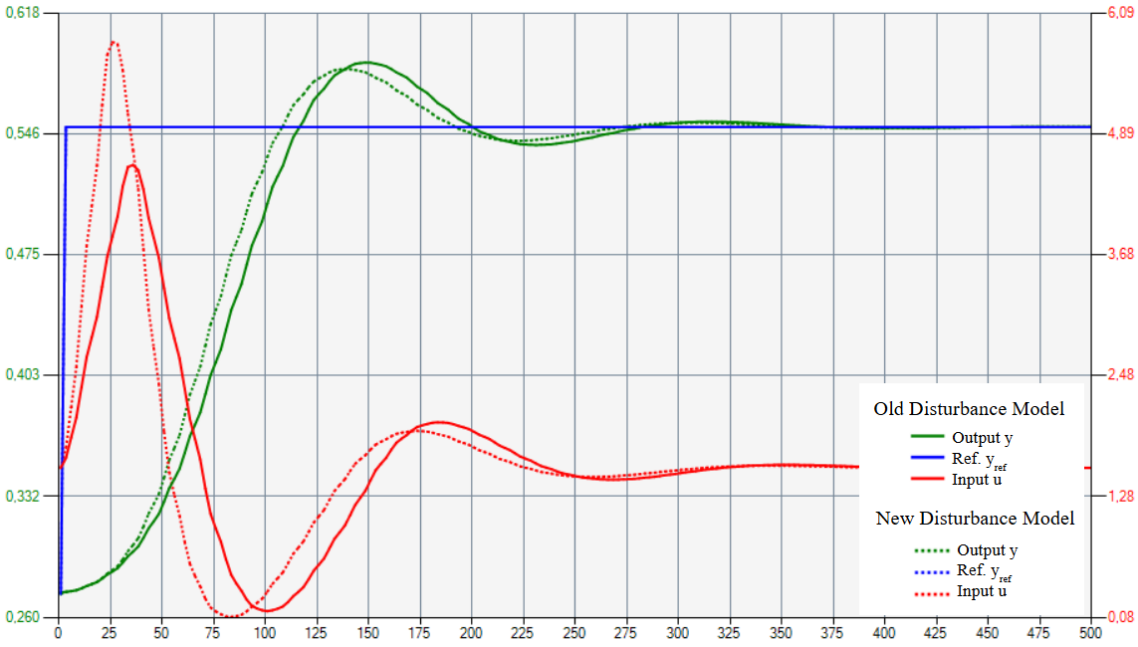


Figure 4.23: The outputs and reference describing the filled portion in the lower tank, as well as inputs (pump flows) using the old and new disturbance models in Case 1. The figure shows the first 500 samples of the simulation.

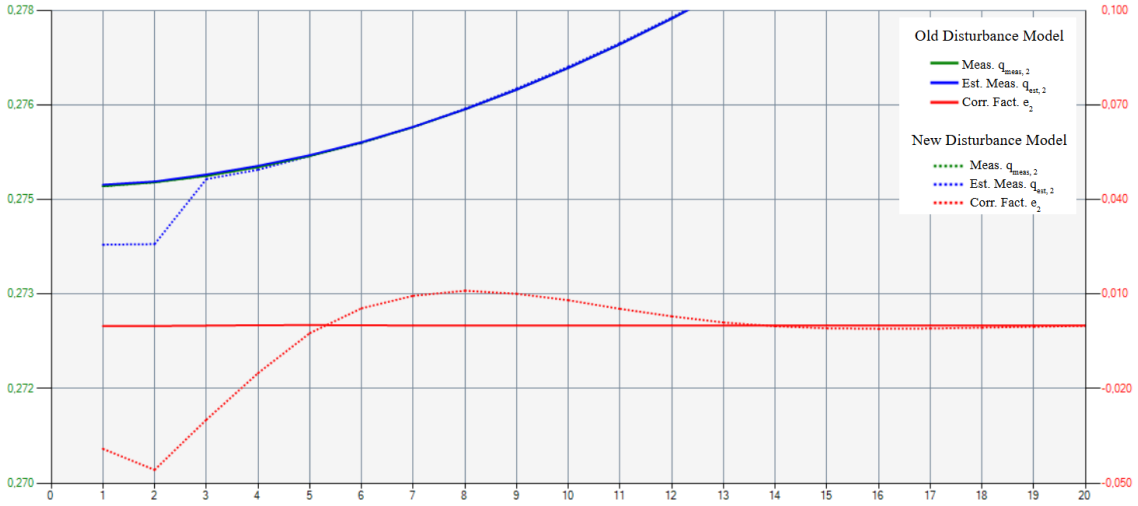


Figure 4.24: The measurements and estimates describing the filled portion in the lower tank, as well as the correction factors when using the old and new disturbance models in Case 1. The figure shows the first 20 samples of the simulations.

The results of the simulations in Case 1 show no improvement when using the new disturbance model. As can be seen in **Figure 4.24** the estimated measurements are not as accurate using the new disturbance model. Just as when comparing MHE with the SISO estimator in Case 1, the decreased accuracy in measurement estimation counteracts the aggressive tuning of the controller, leading to an earlier convergence sample but higher correction factor- and deviation variances (see **Table 4.26**). The results for Case 2 are shown below.

Table 4.27: The resulting convergence sample, deviation- and correction variance when using the two disturbance models in Case 2. Both estimators use the same tuning.

Disturbance Model	New	Old
Convergence Sample	127	176
Corr. Var e_1	2.642×10^{-4}	2.302×10^{-5}
Corr. Var e_2	8.374×10^{-7}	2.301×10^{-5}
Dev. Var $q_{\text{est1}} - q_{\text{meas1}}$	5.876×10^{-9}	7.173×10^{-10}
Dev. Var $q_{\text{est2}} - q_{\text{meas2}}$	5.937×10^{-10}	7.071×10^{-10}

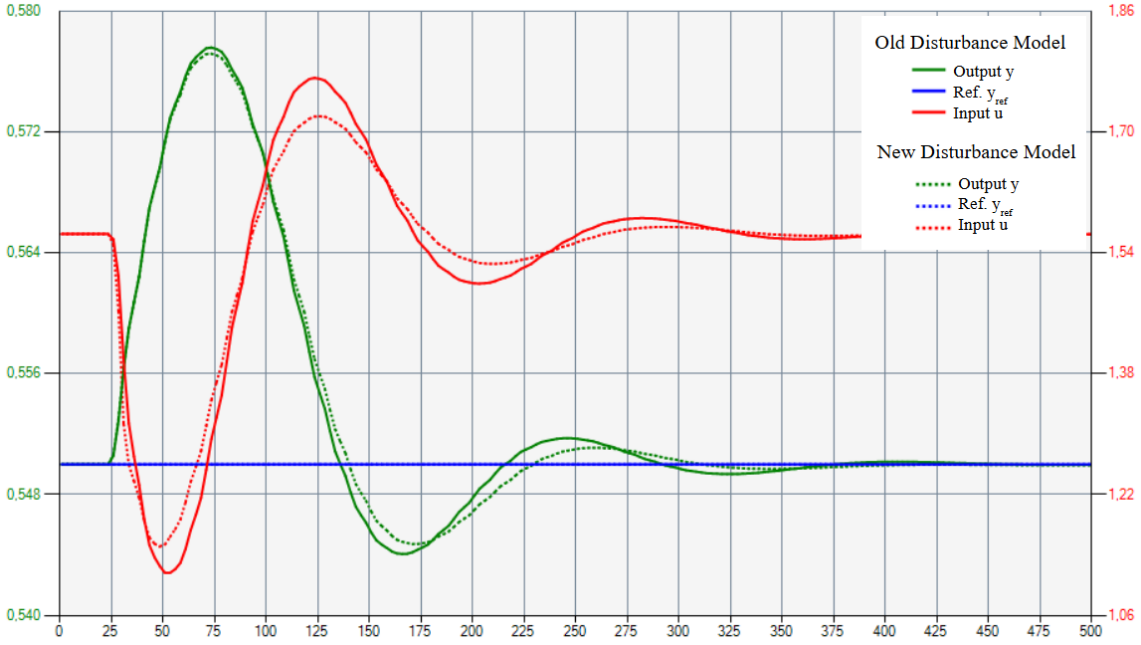


Figure 4.25: The outputs and reference describing the filled portion in the lower tank, as well as the inputs (pump flows) using the old and new disturbance models in Case 2. The figure shows the first 500 samples of the simulation.

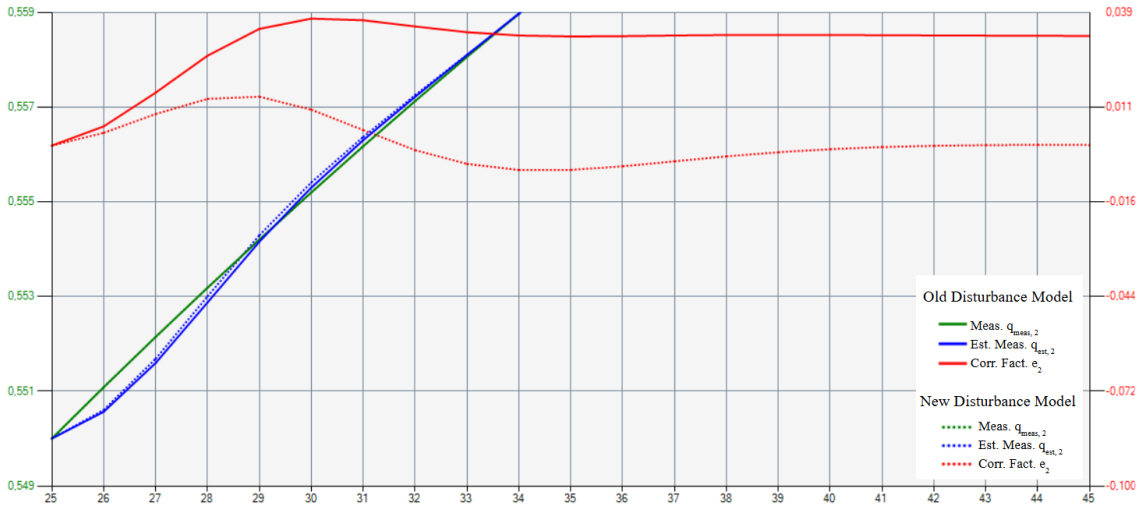


Figure 4.26: The measurements and estimates describing the filled portion in the lower tank, as well as correction factors, when using the old and new disturbance models in Case 2. The figure shows the first 20 samples after the disturbance introduced at sample 25.

Simulations in Case 2 show that using the new disturbance model results in an earlier convergence sample. The estimated measurements of q_{meas2} and the correction factor e_2 shown in **Figure 4.26** lead to lower correction factor variance and lower deviation variance shown in **Table 4.27**. This shows that the new disturbance model results in better control performance while having very similar estimation performance.

4.7 Conclusion and Discussion on the Double Tank System

The comparison between the SISO and MHE estimators leads to the conclusion that an optimally tuned MHE estimator provides better estimation performance. In both cases, the use of MHE results in lowered correction factor and deviation variance. However, as demonstrated in Case 1, improved estimation does not necessarily translate to better overall control performance. Since the control system in these simulations use full feedback, the performance is determined by both the controller and the estimator combined. However, since this report focuses on evaluating the performance of the estimator, rather than optimizing its tuning to compensate for control tuning, MHE is found to yield better results. To improve the control system, the MPC controller can be tuned to be less aggressive. As shown in **Figure 4.19**, the controller drives the output to reach the reference level very quickly. However, due to the fast updates, this results in an overshoot beyond the convergence region, which in turn slows down the overall convergence. Additionally, the convergence criterion was quite strict, $\pm 1\%$ around the setpoint. If the convergence region was instead extended to $\pm 10\%$, the resulting convergence sample evaluation would accept a larger overshoot. This would in turn benefit a more aggressive and accurate estimator. However, the appropriate convergence region also depends on the system requirements and how much deviation from the setpoint was acceptable for the given application. Since the *Double Tank System* is a very simple and stable system with no measurement noise, a relatively small convergence region of $\pm 1\%$ was chosen.

5

Polyethylene Reactor System

This chapter will describe the *Polyethylene Reactor System* and present simulations conducted to evaluate and compare the performance of SISO and MHE approaches. The evaluation is done by finding the optimal tuning for MHE under a case with catalyst error. The performance of MHE is then compared with that of the SISO approach by simulations using measurements from 3 different real-world datasets.

5.1 Problem Description of the Polyethylene Reactor System

The polyolefin reactor used in this report is a low pressure gas phase polymerization reactor (GPR). Polymerization is the process of reacting monomer molecules together in a chemical reaction to form polymer chains.[17] A co-monomer is a secondary monomer introduced during this process to modify the structure or properties of the resulting polymer. While the monomer in this reactor is ethylene, the co-monomer can be either butene or hexene depending on the desired end product, referred to as the grade. Changing the desired grade will be referred to as a transition.

A simplified flowchart of the polymerization reactor system is shown in **Figure 5.1** below.

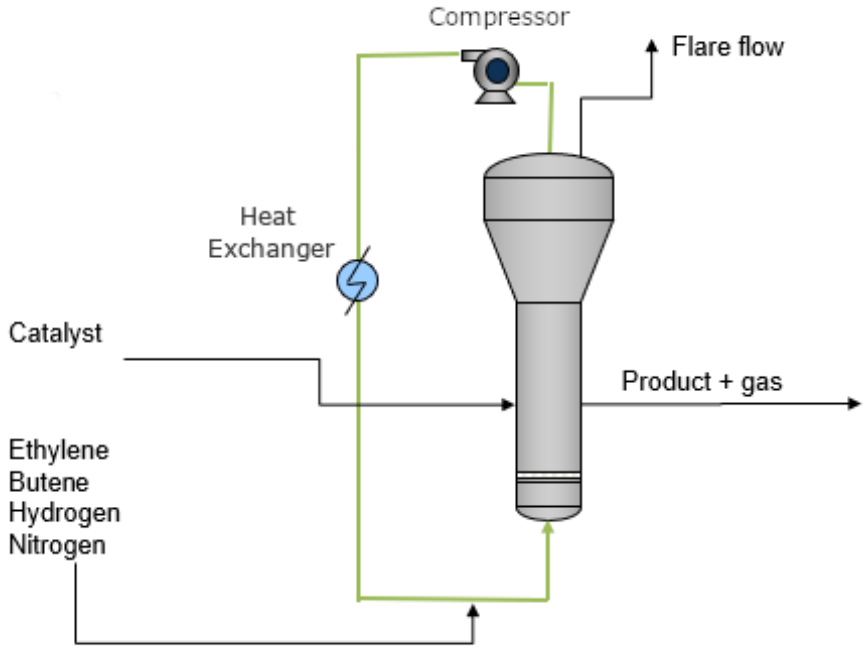


Figure 5.1: Simplified flowchart of the polymerization reactor system with the most essential components for the polymerization process.

Just as in the double tank system, the dynamics of the GPR system is derived from mass balances, meaning that the derivative of gas is modeled as the difference between inflow and outflow. The inflows of all the gases to the GPR are independent feeds of respective gas. Whereas the outflow of each gas is dependent on the controlled flare flow, consumption in the polymerization process and a gas loss caused by the retrieval of end product. The flare flow vents the gas from the GPR. This means that the individual outflow of a gas, caused by the flare flow, is dependent on the concentration of that gas in the reactor. The amount of gases consumed in the polymerization process; hydrogen, monomers and co monomers are dependent on reaction kinetics. The reaction rate at which the gases are consumed can be described by the non-linear function R

$$r = R(c, P_{\text{tot}}, \boldsymbol{m}, \boldsymbol{X}) \exp\left(\frac{k_1}{T} + k_2\right) \tag{5.1}$$

where c is the catalyst properties, P_{tot} is the total pressure in the reactor, $\boldsymbol{m}$ are the molar concentrations, $\boldsymbol{X}$ are the masses of the gases, k_1 and k_2 two empirical constants and T the temperature in the reactor. The correction factors for the monomers and co monomers are integrated in the state equations as multiplicative disturbances affecting the reaction rates. While the correction factors for the other gases are modeled as additive disturbances as extra inflows.

The states, i.e. the masses of each gas, with respective state equation can be written as

$$\begin{aligned}
\text{Hydrogen}: &\quad \dot{x}_{\mathrm{H}_2}(t) = u_{\mathrm{H}_2}(t) - w_{\mathrm{H}_2}(t) - r_{\mathrm{H}_2}(t) + e_{\mathrm{H}_2}(t) \\
\text{Ethylene}: &\quad \dot{x}_{\mathrm{C}_2}(t) = u_{\mathrm{C}_2}(t) - w_{\mathrm{C}_2}(t) - r_{\mathrm{C}_2}(t) \cdot e_{\mathrm{C}_2}(t) \\
\text{Butene}: &\quad \dot{x}_{\mathrm{C}_4}(t) = u_{\mathrm{C}_4}(t) - w_{\mathrm{C}_4}(t) - r_{\mathrm{C}_4}(t) \cdot e_{\mathrm{C}_4}(t) \\
\text{Nitrogen}: &\quad \dot{x}_{\mathrm{N}_2}(t) = u_{\mathrm{N}_2}(t) - w_{\mathrm{N}_2}(t) + e_{\mathrm{N}_2}(t)
\end{aligned}$$

where u is the input feed of each gas, w is the individual outflow caused by the flare flow and gas loss caused by extraction of product combined, r is the reaction rate and e the correction factor.

The total pressure in the GPR is determined by the properties of the reactor contents and temperature through a non-linear function

$$P_{\mathrm{tot}} = f(\boldsymbol{X}_k, \boldsymbol{m}_k, \boldsymbol{\rho}_k, T) \tag{5.2}$$

where $\boldsymbol{X}$ are the masses of the gases, $\boldsymbol{m}$ are the molar concentrations, $\boldsymbol{\rho}$ are the densities and T is the temperature in the reactor. The total pressure can be controlled by increasing the flare flow to release excess gas from the reactor or by adding nitrogen to raise the pressure.

The measurements from the reactor are

Table 5.1: Measurements in the GPR system.

Measurement	Model Name	Description
$q_{\mathrm{meas},\mathrm{H}_2}$	$q_{\mathrm{est},\mathrm{H}_2}$	Hydrogen concentration [mol%] in reactor
$q_{\mathrm{meas},\mathrm{C}_2}$	$q_{\mathrm{est},\mathrm{C}_2}$	Ethylene concentration [mol%] in reactor
$q_{\mathrm{meas},\mathrm{C}_4}$	$q_{\mathrm{est},\mathrm{C}_4}$	Butene concentration [mol%] in reactor
$q_{\mathrm{meas},P_{\mathrm{tot}}}$	$q_{\mathrm{est},P_{\mathrm{tot}}}$	Total pressure [barg] in reactor

The properties of the resulting polymer depend on the ratios of monomer, co-monomer, and hydrogen. This report will therefore foucs on the outputs of the system given in **Table 5.2**.

Table 5.2: Outputs in the GPR system.

Output	Description
y_{HE}	Hydrogen/Ethylene concentration ratio in reactor
y_{BE}	Butene/Ethylene concentration ratio in reactor
$y_{P_{\mathrm{tot}}}$	Total pressure in reactor
y_{Epp}	Ethylene partial pressure in reactor
y_{Flare}	Flare flow from reactor

As a clarification, **Table 5.3** shows a summary of all the signals in the GPR system.

Table 5.3: A summary of all signals in the GPR system.

Controlled Outputs y	Control Inputs u	Measurements q	Correction Factors e
y_{HE}	$u_{\mathrm{H_2}}$	$q_{\mathrm{meas,H_2}}, q_{\mathrm{est,H_2}}$	$e_{\mathrm{H_2}}$
y_{BE}	$u_{\mathrm{C_4}}$	$q_{\mathrm{meas,C_4}}, q_{\mathrm{est,C_4}}$	$e_{\mathrm{C_4}}$
$y_{P_{\mathrm{tot}}}$	$u_{\mathrm{C_2}}$	$q_{\mathrm{meas,C_2}}, q_{\mathrm{est,C_2}}$	$e_{\mathrm{C_2}}$
y_{Epp}	$u_{\mathrm{N_2}}$	$q_{\mathrm{meas},P_{\mathrm{tot}}}, q_{\mathrm{est},P_{\mathrm{tot}}}$	$e_{\mathrm{N_2}}$
y_{Flare}	u_{Flare}		

5.2 State Interactions and System Dynamics

To better illustrate how the states in this multivariable system influence each other, a simple test case will be introduced to help understand the system dynamics and interactions between feeds and outputs. In the simulation controlled inputs are sent to an independent process model, which means some degree of model error will occur.

At the beginning of the simulation, the setpoints for the outputs are set according to the values shown in column 2 of the **Table 5.4**. To clearly observe the effect of changing setpoints, the simulation maintains these values until sample 500. At that point, a new setpoint for pressure is introduced. It is increased from 20 to 20.2, while the other reference values remain unchanged. This change is introduced to evaluate the response of the GPR system to a small setpoint variation. The complete set of setpoints used throughout the simulation is summarized in the **Table 5.4**.

Table 5.4: The setpoints used to clarify the dynamics of the system. At sample 30, the initial setpoints are introduced, and a change in the setpoints occurs at sample 500

Setpoints	$t = 30$	$t = 500$
y_{HE}	60	60
y_{BE}	1.5	1.5
y_{Epp}	6	6
$y_{P_{\mathrm{tot}}}$	20	20.2
y_{Flare}	0	0

As the pressure setpoint at sample 500 is increased, the ethylene feed is briefly raised. The ethylene feed is then gradually decreased until it returns to a similar steady state value. The relation between total pressure in the GPR and it's reference, as well as ethylene flow can be seen in **Figure 5.2**.

Figure 5.2: Total pressure in the reactor expressed in barg (gauge pressure in bars), with reference change at sample 500 together, as well as ethylene feed in ton/h.

Since ethylene is added to the reactor, increasing its concentration, both the Hydrogen/Ethylene and Butene/Ethylene ratios decrease. To maintain these ratios at their reference values, the controller compensates by adding Hydrogen and Butene. Hydrogen/Ethylene and Butene/Ethylene with hydrogen and butene flows are shown in **Figure 5.3** and **Figure 5.4** below.

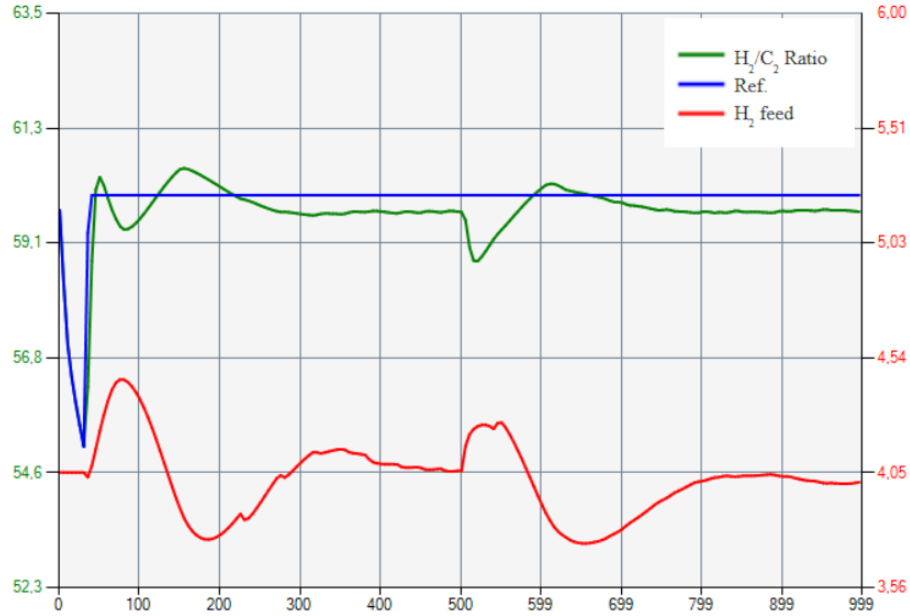


Figure 5.3: Hydrogen/Ethylene ratio expressed in mol/kmol together with hydrogen feed in kg/h. The hydrogen correction factor does not fully reach a steady state, hence the deviation between controlled output and reference before and after the disturbance.

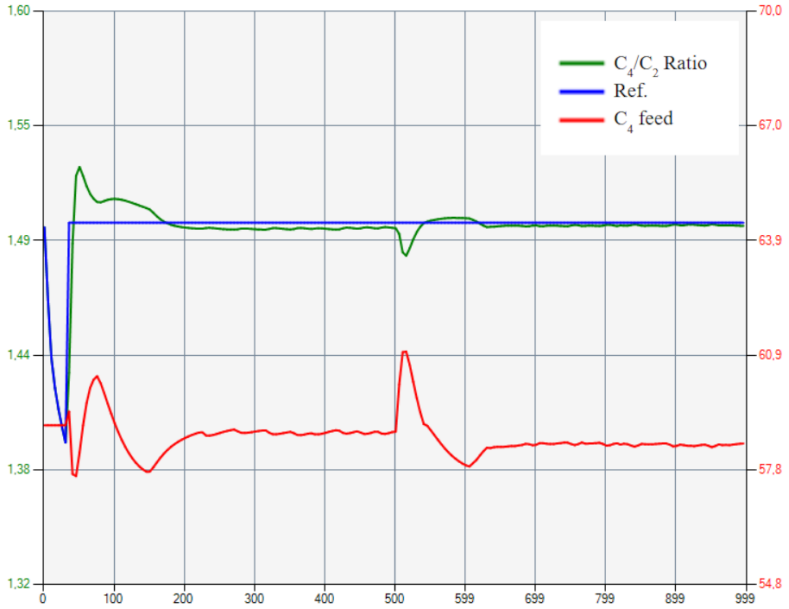


Figure 5.4: Butene/Ethylene ratio expressed in mol/kmol together with butene feed measured in kg/h. The butene correction factor does not fully reach a steady state, hence the deviation between controlled output and reference before the disturbance.

The added ethylene will also increase the overall partial pressure of ethylene. To compensate for this, nitrogen is added to the reactor, increasing the total pressure in the reactor, while not affecting the ethylene concentration. Ethylene Partial Pressure and Nitrogen flow is shown in **Figure 5.5**.

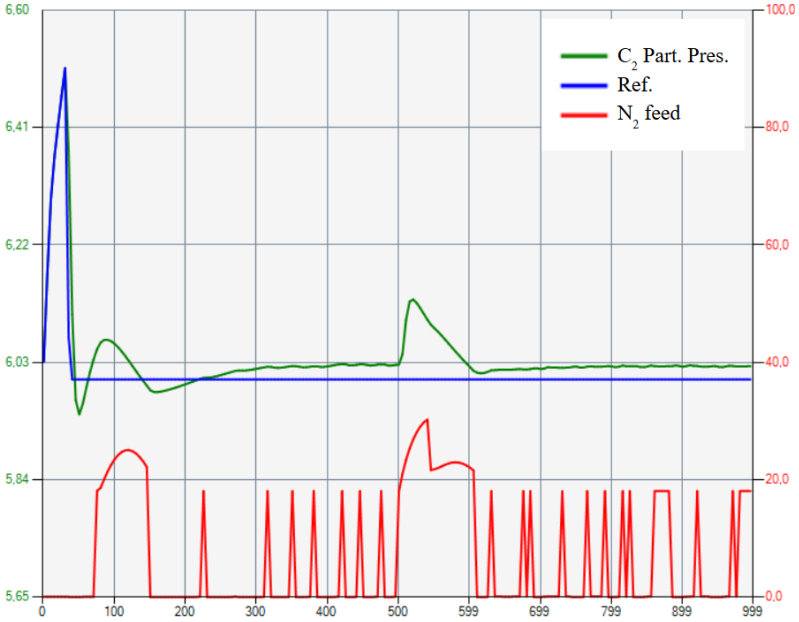


Figure 5.5: Ethylene partial pressure in barg together with nitrogen feed in kg/h. The nitrogen correction factor does not fully reach a steady state, hence the deviation between controlled output and reference before and after the disturbance.

Although flare flow is undesirable (with a setpoint of 0), there will almost always be some present to stabilize drastic changes in the system. In this system, both flare and nitrogen flow are non-ideal, as the measurements are either 18 or zero and cannot pick up any value in between. The flare flow for the test case is shown in **Figure 5.6** below.

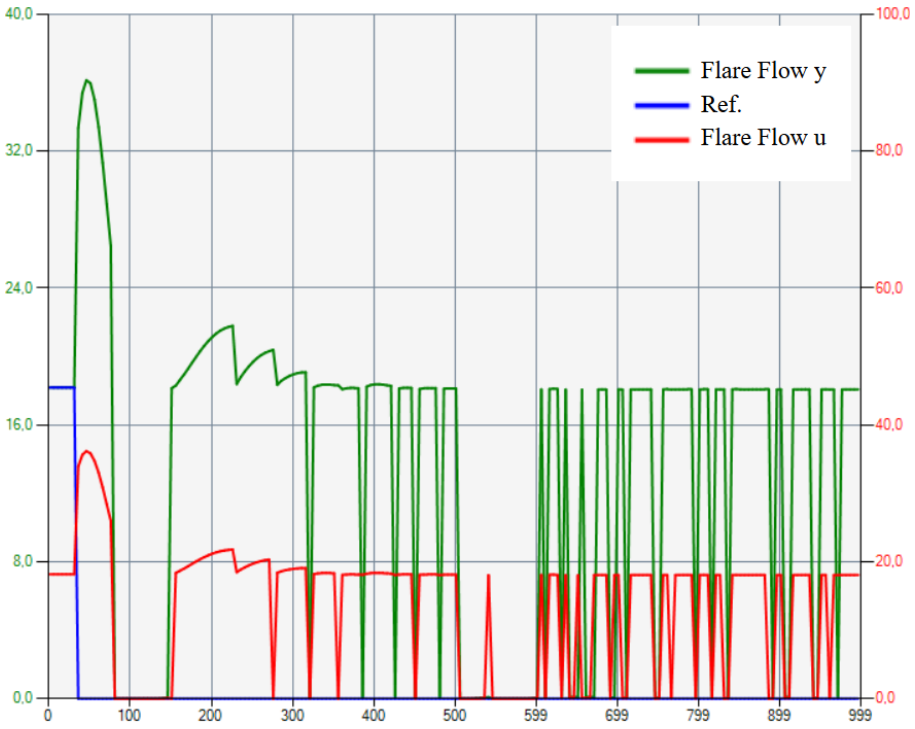


Figure 5.6: Flare flow with input signal, both expressed in kg/h.

5.3 Test Case and Simulation Setup

In this system, only one case will be used to evaluate the performance of the estimators. The case is described below:

Catalyst Error:
In this case, a closed-loop control simulation is performed. At the beginning of the simulation, the reference values for the outputs are set to

Outputs	y_{HE}	y_{BE}	y_{Epp}	$y_{P_{\text{tot}}}$	y_{Flare}
Setpoints	60	1.5	6	20	0

Since the initial values of the system are not at steady state, it is necessary to allow the system to reach steady state before introducing any disturbances. This ensures a clearer evaluation of the estimators performance. The disturbance is applied as an excessive catalyst input, the catalyst feed of the plant is increased by 10% at sample 500. The effect of the catalyst is an increased reaction rate, which the system must

then regulate.

Since the system is not at steady state at the beginning, it is necessary to wait until steady state is reached before introducing the disturbance into the system. Therefore, the outcomes of the first samples will not be used to calculate the variance of the correction factors and the deviations. Based on testing, the system reaches steady state at approximately sample 100. Thus, data collection begins from sample 100 onward. To compensate for the discarded samples, each simulation runs for 1100 samples in total, ensuring a sufficient number of valid data, 1000 data samples.

The disturbance will then be added to the system at sample 500. If the disturbance does not cause the outputs to deviate beyond ±1% of their setpoints, the system is considered capable of handling the disturbance effectively, and the convergence sample is set to 500. However, if the disturbance drives the outputs out of the convergence region, the convergence sample is defined as the point at which the outputs re-enter and remain within the convergence region. The setpoint of y_{Flare} is set to zero, but it is not intended to be actively tracked. The purpose of the flare flow is to rapidly reduce the pressure in the reactor when necessary. Therefore, if gas needs to be released to reduce the pressure, it must be released regardless of the setpoint. Thus, the flare's setpoint is important, but it is not of primary significance. This is achieved by setting the weight value low for controlled y_{Flare}. The sample time is set to one minute per sample, as the system is relatively slow and does not require high-frequency sampling.

5.4 Setup and Performance Evaluation for Single-Input Single-Output Estimation

The gains in this system will not be tuned either, but instead, the gains currently used by the company will be applied.

Table 5.5: The gains for the SISO estimator in the *Polyethylene Reactor System*.

$xGain_{\text{H}_2}$	0.1	$eGain_{\text{H}_2}$	0.05
$xGain_{\text{C}_2}$	0.1	$eGain_{\text{C}_2}$	-0.05
$xGain_{\text{C}_4}$	0.1	$eGain_{\text{C}_4}$	-0.05
$xGain_{\text{N}_2}$	0.1	$eGain_{\text{N}_2}$	30

In this problem, the states are updated using multiplicative approach since the mass

of gases can not be negative. The states at sample k are updated as follow:

$$x_{\mathrm{H_2},k} = x_{\mathrm{H_2,old},k} \cdot \left(1 + xGain_{\mathrm{H_2}} \cdot \frac{q_{\mathrm{meas,H_2},k} - q_{\mathrm{est,H_2},k}}{q_{\mathrm{meas,H_2},k}}\right)$$
$$x_{\mathrm{C_2},k} = x_{\mathrm{C_2,old},k} \cdot \left(1 + xGain_{\mathrm{C_2}} \cdot \frac{q_{\mathrm{meas,C_2},k} - q_{\mathrm{est,C_2},k}}{q_{\mathrm{meas,C_2},k}}\right)$$
$$x_{\mathrm{C_4},k} = x_{\mathrm{C_4,old},k} \cdot \left(1 + xGain_{\mathrm{C_4}} \cdot \frac{q_{\mathrm{meas,C_4},k} - q_{\mathrm{est,C_4},k}}{q_{\mathrm{meas,C_4},k}}\right)$$
$$x_{\mathrm{N_2},k} = x_{\mathrm{N_2,old},k} \cdot \left(1 + xGain_{\mathrm{N_2}} \cdot \frac{q_{\mathrm{meas},P_{\mathrm{tot}},k} - q_{\mathrm{est},P_{\mathrm{tot}},k}}{q_{\mathrm{meas},P_{\mathrm{tot}},k}}\right)$$

The correction factors at sample k are updated as follow:

$$e_{\mathrm{H_2},k} = e_{\mathrm{H_2,old},k} + eGain_{\mathrm{H_2}} \cdot \frac{q_{\mathrm{meas,H_2},k} - q_{\mathrm{est,H_2},k}}{q_{\mathrm{meas,H_2},k}}$$
$$e_{\mathrm{C_2},k} = e_{\mathrm{C_2,old},k} \cdot \left(1 + eGain_{\mathrm{C_2}} \cdot \frac{q_{\mathrm{meas,C_2},k} - q_{\mathrm{est,C_2},k}}{q_{\mathrm{meas,C_2},k}}\right)$$
$$e_{\mathrm{C_4},k} = e_{\mathrm{C_4,old},k} \cdot \left(1 + eGain_{\mathrm{C_4}} \cdot \frac{q_{\mathrm{meas,C_4},k} - q_{\mathrm{est,C_4},k}}{q_{\mathrm{meas,C_4},k}}\right)$$
$$e_{\mathrm{N_2},k} = e_{\mathrm{N_2,old},k} + eGain_{\mathrm{N_2}} \cdot \frac{q_{\mathrm{meas},P_{\mathrm{tot}},k} - q_{\mathrm{est},P_{\mathrm{tot}},k}}{q_{\mathrm{meas},P_{\mathrm{tot}},k}}$$

Note that minimum values for q_{meas} are also defined to prevent division by zero during the update. The performance of the SISO estimator for the test case will be presented in the following subsection.

5.4.1 Test Case: Catalyst Error

The table below shows the performance of SISO estimator using the gains shown in **Table 5.5**

Table 5.6: The resulting variance of correction factor and deviation when using the SISO-estimator in the test case. All outputs stayed within the convergence region, thus they converge by sample 500.

Corr. Var $e_{\mathrm{H_2}}$	4.412×10^{-4}
Corr. Var $e_{\mathrm{C_2}}$	1.260×10^{-1}
Corr. Var $e_{\mathrm{C_4}}$	1.766×10^{-2}
Corr. Var $e_{\mathrm{N_2}}$	2.968×10^{-1}
Dev. Var $q_{\mathrm{est,H_2}} - q_{\mathrm{meas,H_2}}$	2.990×10^{-5}
Dev. Var $q_{\mathrm{est,C_2}} - q_{\mathrm{meas,C_2}}$	4.480×10^{-3}
Dev. Var $q_{\mathrm{est,C_4}} - q_{\mathrm{meas,C_4}}$	6.144×10^{-7}
Dev. Var $q_{\mathrm{est},P_{\mathrm{tot}}} - q_{\mathrm{meas},P_{\mathrm{tot}}}$	2.196×10^{-5}

5.5 Tuning and Performance Evaluation of Moving Horizon Estimation

The parameters involved in this problem are presented in the table below:

Table 5.7: Parameters in Moving Horizon Estimation for *Polyethylene Reactor System*.

Category	Parameter	Description
To be Defined	$e_{H_2,min}, e_{C_4,min}$ $e_{C_2,min}, e_{N_2,min}$ $e_{H_2,max}, e_{C_4,max}$ $e_{C_2,max}, e_{N_2,max}$	Minimum and maximum value of correction factors
	$\Delta e_{H_2,min}, \Delta e_{C_4,min}$ $\Delta e_{C_2,min}, \Delta e_{N_2,min}$ $\Delta e_{H_2,max}, \Delta e_{C_4,max}$ $\Delta e_{C_2,max}, \Delta e_{N_2,max}$	Minimum and maximum value of the change of correction factors
	$e_{H_2,perturb}, e_{C_4,perturb}$ $e_{C_2,perturb}, e_{N_2,perturb}$	Constants used to calculate $\mathbf{q}_{sens}$ in SQP iteration
To be Tuned	$\mathbf{Q}_e$=diag$(q_{H_2}, q_{C_4}, q_{C_2}, q_{N_2})$	Weighting matrix for process noise
	$\mathbf{R}$=diag$(r_{H_2}, r_{C_4}, r_{C_2}, r_{P_{tot}})$	Weighting matrix for measurement noise
	N	Horizon length
	n_b	Number of blocks
	$n_{b,len}$	Length of each block

where the defined parameters are set to be:

Table 5.8: The value of the defined parameters in MHE for *Polyethylene Reactor* System.

	e_{min}	e_{max}	Δe_{min}	Δe_{max}	$e_{perturb}$
H_2	-5	5	-0.01	0.01	0.1
C_4	0.1	10	-0.05	0.05	0.1
C_2	0.1	30	-0.05	0.05	0.1
N_2	-500	500	-5	5	1

In the tuning process, a set of standard parameters is introduced to make it easier to isolate and understand the effect of changing individual parameters. These values are:

Table 5.9: Standard parameters in MHE for *Polyethylene Reactor System*.

$q_{H_2}, q_{C_4}, q_{C_2}, q_{N_2}$	0.1, 10, 10, 1
$r_{H_2}, r_{C_4}, r_{C_2}, r_{P_{tot}}$	10, 10, 100, 100
n_b	5
N	100
$n_{b,len}$	uniform distribution 20,20,20,20,20

These parameters will be used as standard values for simulation in different cases of the problem. The impact of tuning parameters will be presented in the following sections.

5.5.1 Test Case: Catalyst Error

The impact of tuning different parameters are presented below.

Weight Tuning:

The impact on converge sample, correction factor variance and deviation variance when tuning the weights matrices, $\mathbf{Q_e}$ and $\mathbf{R}$, in different combinations is shown in **Table 5.11** and **Table 5.10** below. Horizon length, horizon distribution and number of blocks were kept to the standard values shown in **Table 5.9**.

Table 5.10: Effect of tuning the elements in weighting matrix $\mathbf{R}$ while keeping $\mathbf{Q_e}$ constant on convergence sample, correction factor variance, and measurement deviation variance in test case. All combinations of the weight matrix keep the output within the convergence region after the error is introduced. Thus, all outputs converge by sample 500. The result of the standard tuning is shown in bold text.

Weighting Matrices	$\mathbf{q_{H_2} = 0.1}$ $\mathbf{q_{N_2} = 1}$ $\mathbf{q_{C_2}, q_{C_4} = 10}$ $\mathbf{r_{H_2}, r_{C_4} = 10}$ $\mathbf{r_{C_2}, r_{P_{tot}} = 100}$	$q_{H_2} = 0.1$ $q_{N_2} = 1$ $q_{C_2}, q_{C_4} = 10$ $r_{H_2}, r_{C_4} = 10^3$ $r_{C_2}, r_{P_{tot}} = 10^5$	$q_{H_2} = 0.1$ $q_{N_2} = 1$ $q_{C_2}, q_{C_4} = 10$ $r_{H_2}, r_{C_4} = 10^5$ $r_{C_2}, r_{P_{tot}} = 10^7$
Corr. Var e_{H_2}	$\mathbf{8.624 \times 10^{-3}}$	4.279×10^{-3}	4.217×10^{-3}
Corr. Var e_{C_2}	$\mathbf{1.284 \times 10^{-1}}$	1.272×10^{-1}	1.272×10^{-1}
Corr. Var e_{C_4}	$\mathbf{2.141 \times 10^{-2}}$	2.155×10^{-2}	2.154×10^{-2}
Corr. Var e_{N_2}	$\mathbf{1.046 \times 10^{1}}$	1.046×10^{1}	1.046×10^{1}
Dev. Var $q_{est,H_2} - q_{meas,H_2}$	$\mathbf{4.151 \times 10^{-5}}$	1.248×10^{-5}	1.254×10^{-5}
Dev. Var $q_{est,C_2} - q_{meas,C_2}$	$\mathbf{5.178 \times 10^{-4}}$	6.747×10^{-4}	6.794×10^{-4}
Dev. Var $q_{est,C_4} - q_{meas,C_4}$	$\mathbf{4.794 \times 10^{-10}}$	9.667×10^{-11}	1.000×10^{-10}
Dev. Var $q_{est,P_{tot}} - q_{meas,P_{tot}}$	$\mathbf{3.838 \times 10^{-5}}$	1.844×10^{-5}	1.864×10^{-5}

The results of increasing the elements of $\mathbf{R}$ while keeping the $\mathbf{Q_e}$ matrix constant, i.e. increasing the ratio of $\mathbf{R}$ relative to $\mathbf{Q_e}$, are shown in **Table 5.10**. Similarly to

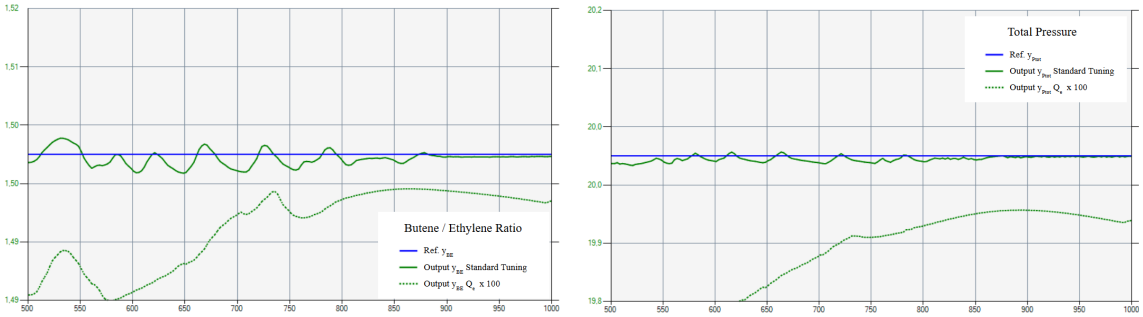
the observations from the *Double Tank System* the results indicate that increasing this ratio does not significantly improve performance. To determine if this was the case for a large region of magnitudes of $\mathbf{R}$, the increment of elements in $\mathbf{R}$ for each simulation was increased. The convergence sample remains unchanged. All outputs stayed within the convergence region, while the variances of both the correction factors and the deviations decrease slightly but not substantially.

Table 5.11: Effect of tuning the elements in weighting matrix $\mathbf{Q_e}$ while keeping $\mathbf{R}$ constant on convergence sample, correction factor variance, and measurement deviation variance in test case. The result of the standard tuning is shown in bold text.

Weighting Matrices	$\mathbf{q_{H_2} = 0.1}$ $\mathbf{q_{N_2} = 1}$ $\mathbf{q_{C_2}, q_{C_4} = 10}$ $\mathbf{r_{H_2}, r_{C_4} = 10}$ $\mathbf{r_{C_2}, r_{P_{tot}} = 100}$	$q_{H_2} = 1$ $q_{N_2} = 10$ $q_{C_2}, q_{C_4} = 100$ $r_{H_2}, r_{C_4} = 10$ $r_{C_2}, r_{P_{tot}} = 100$	$q_{H_2} = 10$ $q_{N_2} = 100$ $q_{C_2}, q_{C_4} = 10^3$ $r_{H_2}, r_{C_4} = 10$ $r_{C_2}, r_{P_{tot}} = 100$
Convergence Sample y_{HE}	**500**	500	500
Convergence Sample y_{BE}	**500**	500	583
Convergence Sample y_{Epp}	**500**	500	500
Convergence Sample $y_{P_{tot}}$	**500**	500	624
Corr. Var e_{H_2}	$\mathbf{8.624 \times 10^{-3}}$	3.984×10^{-2}	6.633×10^{-2}
Corr. Var e_{C_2}	$\mathbf{1.284 \times 10^{-1}}$	1.304×10^{-1}	2.165×10^{-1}
Corr. Var e_{C_4}	$\mathbf{2.141 \times 10^{-2}}$	2.371×10^{-2}	3.476×10^{-2}
Corr. Var e_{N_2}	$\mathbf{1.046 \times 10^{1}}$	1.169×10^{2}	3.265×10^{3}
Dev. Var $q_{est,H_2} - q_{meas,H_2}$	$\mathbf{4.151 \times 10^{-5}}$	7.747×10^{-4}	1.721×10^{-1}
Dev. Var $q_{est,C_2} - q_{meas,C_2}$	$\mathbf{5.178 \times 10^{-4}}$	1.498×10^{-3}	3.274×10^{-2}
Dev. Var $q_{est,C_4} - q_{meas,C_4}$	$\mathbf{4.794 \times 10^{-10}}$	1.320×10^{-8}	3.149×10^{-6}
Dev. Var $q_{est,P_{tot}} - q_{meas,P_{tot}}$	$\mathbf{3.838 \times 10^{-5}}$	8.023×10^{-4}	5.568×10^{-2}

On the other hand, increasing the ratio of $\mathbf{Q_e}$ relative to $\mathbf{R}$ leads to higher variances in the correction factor and the deviation. When the elements of $\mathbf{Q_e}$ increase by a

factor of 100, column 3, the correction factor update is so heavily penalized that the outputs y_{BE} and $y_{P_{tot}}$ exit the convergence interval. These outputs are compared with the standard tuning in **Figure 5.7**



(a) Butene / ethylene ratio output in mol/kmol when standard tuning and when the $\mathbf{Q_e}$ matrix is multiplied with 100, together with the setpoint of 1.5 mol/kmol.

(b) The total pressure output in barg with standard tuning and with a higher $\mathbf{Q_e}$-weight, together with the setpoint of 20 barg.

Figure 5.7: The butene / ethylene ratio output and total pressure output with standard tuning compared with a higher $\mathbf{Q_e}$-weight. The figures show the results from sample 500, where the disturbance occurred.

Figure 5.7 (a) shows that the butene / ethylene ratio output slightly goes outside the convergence region, causing the system to converge slower. Looking at **Figure 5.7 (b)**, the tuning parameters makes the estimator so slow, that the total pressure output does not converge before the disturbance occurs in the system. This indicates that a higher $\mathbf{Q_e}$ worsens the overall system performance.

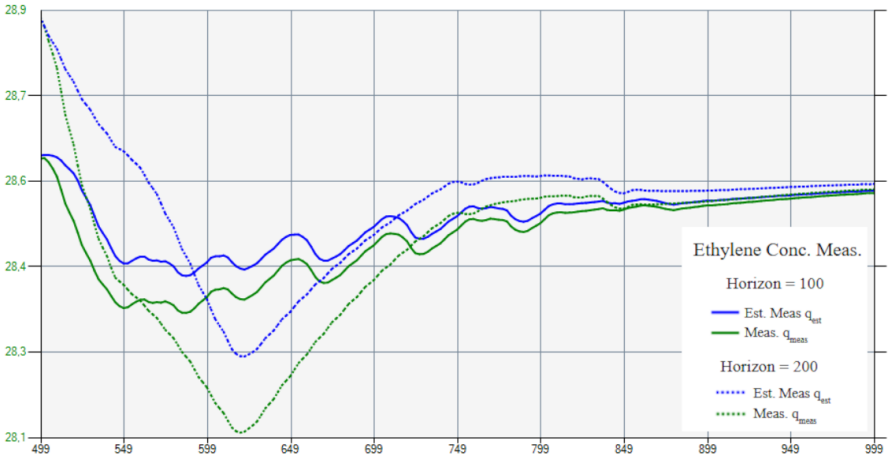
Horizon Length:

The results for four different horizon lengths are shown in **Table 5.12** below. All horizon lengths have a uniform block distribution across 5 blocks, meaning each block within a given tuning case is of equal length. Other parameters are kept as standard values, see **Table 5.9**.

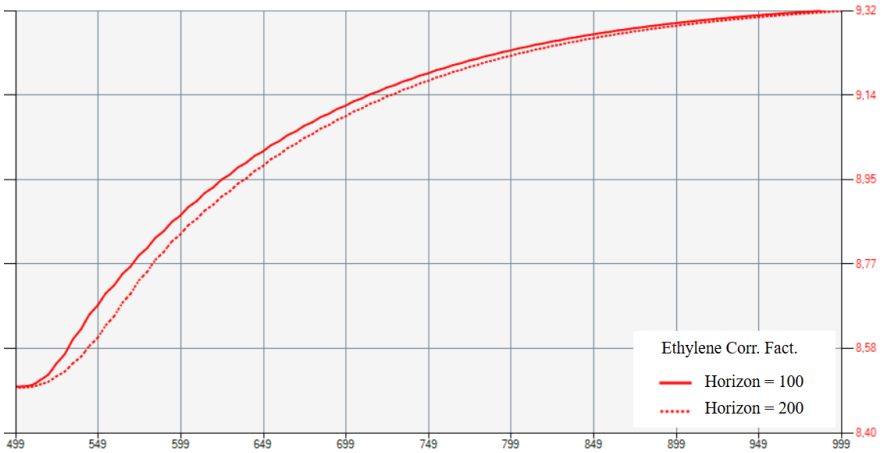
Table 5.12: Effect of different horizon lengths, with uniform block distribution across 5 blocks, on convergence sample, correction factor variance, and measurement deviation variance in test case. The result of the standard tuning is shown in bold text.

Horizon Length	Horizon = 10	Horizon = 50	**Horizon = 100**	Horizon = 200
Block Length	2, 2, 2 2, 2	10, 10, 10 10 10	**20, 20, 20** **20, 20**	40, 40, 40 40, 40
Convergence Sample y_{HE}	800	500	**500**	500
Convergence Sample y_{BE}	500	500	**500**	500
Convergence Sample y_{Epp}	No Convergence	500	**500**	628
Convergence Sample $y_{P_{\mathrm{tot}}}$	500	500	**500**	500
Corr. Var $e_{\mathrm{H_2}}$	1.707×10^{-1}	9.147×10^{-3}	$\mathbf{8.624 \times 10^{-3}}$	1.772×10^{-2}
Corr. Var $e_{\mathrm{C_2}}$	1.854×10^{-1}	1.289×10^{-1}	$\mathbf{1.284 \times 10^{-1}}$	1.241×10^{-1}
Corr. Var $e_{\mathrm{C_4}}$	1.912×10^{-2}	2.231×10^{-2}	$\mathbf{2.141 \times 10^{-2}}$	2.263×10^{-2}
Corr. Var $e_{\mathrm{N_2}}$	1.179×10^{3}	3.452×10^{1}	$\mathbf{1.046 \times 10^{1}}$	2.809×10^{2}
Dev. Var $q_{\mathrm{est,H_2}} - q_{\mathrm{meas,H_2}}$	1.550×10^{-1}	2.543×10^{-4}	$\mathbf{4.151 \times 10^{-5}}$	1.004×10^{-4}
Dev. Var $q_{\mathrm{est,C_2}} - q_{\mathrm{meas,C_2}}$	2.522×10^{-2}	1.162×10^{-4}	$\mathbf{5.178 \times 10^{-4}}$	1.012×10^{-2}
Dev. Var $q_{\mathrm{est,C_4}} - q_{\mathrm{meas,C_4}}$	3.427×10^{-7}	6.601×10^{-10}	$\mathbf{4.794 \times 10^{-10}}$	9.755×10^{-8}
Dev. Var $q_{\mathrm{est},P_{\mathrm{tot}}} - q_{\mathrm{meas},P_{\mathrm{tot}}}$	7.136×10^{-2}	1.003×10^{-4}	$\mathbf{3.838 \times 10^{-5}}$	3.313×10^{-4}

The results across different horizon lengths show that a short horizon of 10 exhibits higher deviation and greater correction factor variance compared to the standard tuning. Additionally, two outputs, y_{HE} and y_{Epp}, exits the convergence region during the simulation. Increasing the horizon to 50 samples yields performance equivalent to that at the standard 100-sample length. However, the performance declines again when the horizon is extended to 200 samples, suggesting that longer horizons do not improve the performance of the estimator. Measurements with estimated measurements of ethylene from the MHE tuning with horizon lengths 100 and 200 are shown in **Figure 5.8** below.



(a) The true and estimated measurements of ethylene concentration expressed in mol% from simulating test case with uniformly distributed horizon lengths 100 and 200 after the disturbance is introduced at sample 500.



(b) The ethylene rate correction factor (unitless) for the same two horizon lengths.

Figure 5.8: The true and estimated measurements of ethylene concentration in, along with the ethylene rate correction factor, are compared between horizon lengths of 100 and 200. The figures show the results from sample 500, where the disturbance occurred.

By observing **Figure 5.8 (b)** it is clear that the ethylene correction factor is updated slower when the horizon is extended to 200. This causes an increased gap between measurements and estimated measurements in **Figure 5.8 (a)**, which also match slower than using a horizon length of 100. This leads to a downswing causing the ethylene partial pressure output to exceed the convergence interval of $\pm$ 1% at around sample 625, as illustrated in **Figure 5.9** below. Thereby, with a horizon of 200, y_{Epp} converges very late, only a few samples before the disturbance occurs at sample 500. This causes it to start from a high value of 6.06.

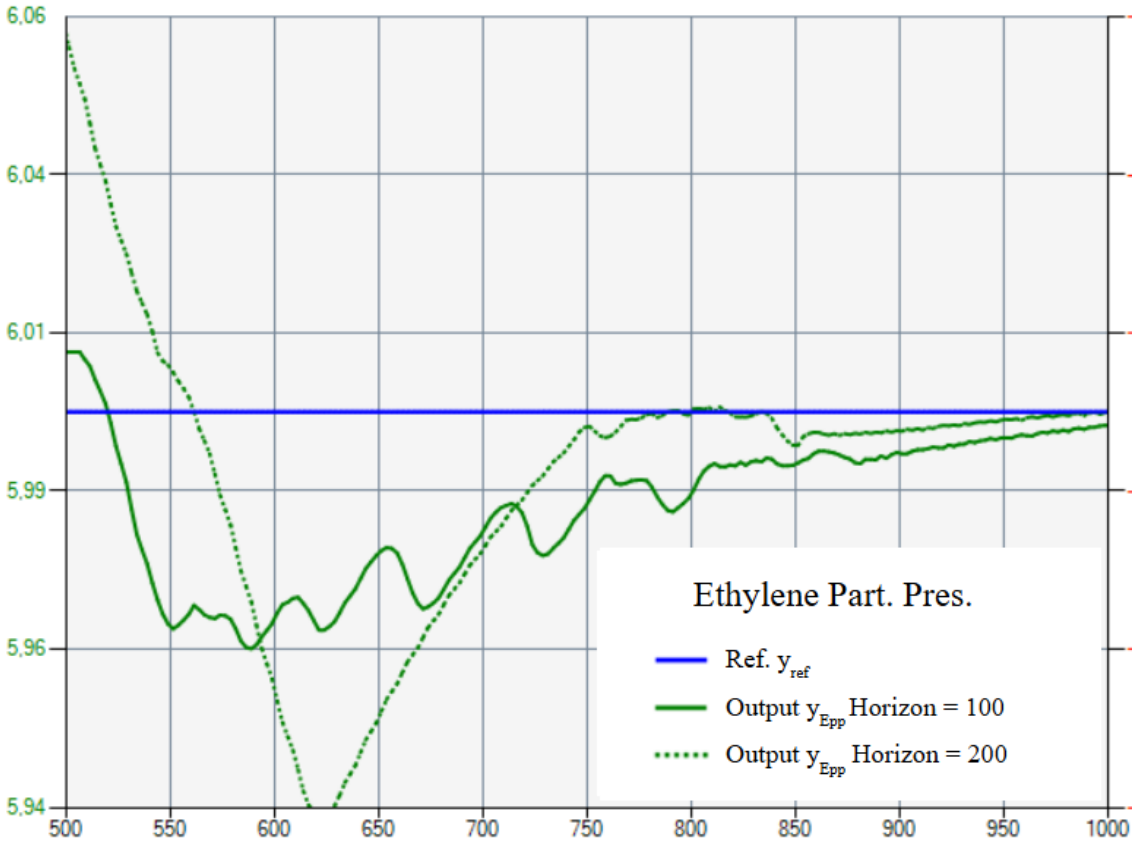


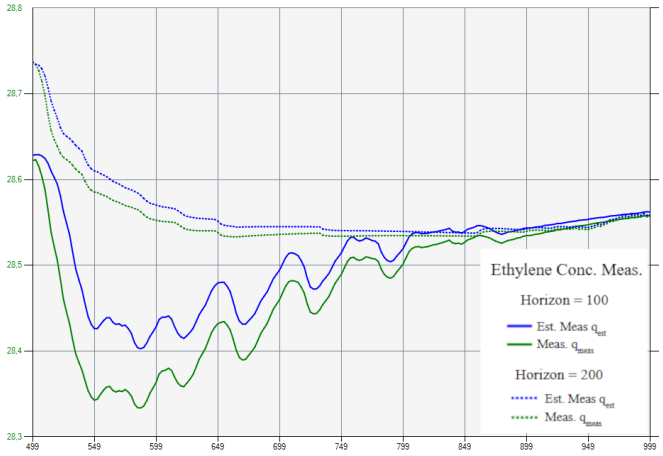
Figure 5.9: Outputs of ethylene partial pressure and the reference, both in barg, when using horizon lengths 100 and 200 together with the reference y_{Epp}. The figure shows the results from sample 500, when the disturbance occurred.

In the previous tunings, all horizons were divided into five uniformly distributed blocks, which led to an unfair comparison, since a longer horizon results in updates that are further from the current sample. Alternatively, the same horizons can be divided so that all blocks have a constant length. The results for the same horizon lengths, but with a constant block length of 10 across all horizons, are presented in **Table 5.13**.

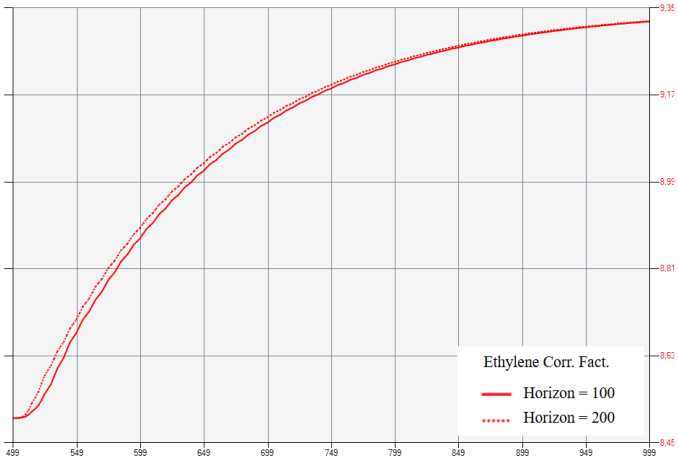
Table 5.13: Effect of different horizon lengths on convergence sample, correction factor variance, and measurement deviation variance in test case. All combinations of the parameters keep the output within the convergence region after the error is introduced. Thus, all outputs converge by sample 500.

Horizon Length	Horizon = 10	Horizon = 50	Horizon = 100	Horizon = 200
Number of Blocks	1	5	10	20
Corr. Var e_{H_2}	2.303×10^{-2}	9.147×10^{-3}	3.529×10^{-3}	2.728×10^{-3}
Corr. Var e_{C_2}	1.293×10^{-1}	1.289×10^{-1}	1.287×10^{-1}	1.275×10^{-1}
Corr. Var e_{C_4}	2.199×10^{-2}	2.231×10^{-2}	2.177×10^{-2}	2.154×10^{-2}
Corr. Var e_{N_2}	3.387×10^{1}	3.452×10^{1}	4.527	1.251
Dev. Var $q_{\text{est},H_2} - q_{\text{meas},H_2}$	1.715×10^{-3}	2.543×10^{-4}	4.855×10^{-5}	3.044×10^{-6}
Dev. Var $q_{\text{est},C_2} - q_{\text{meas},C_2}$	2.184×10^{-4}	1.162×10^{-4}	5.951×10^{-5}	5.401×10^{-5}
Dev. Var $q_{\text{est},C_4} - q_{\text{meas},C_4}$	2.925×10^{-9}	6.601×10^{-10}	6.582×10^{-10}	7.992×10^{-10}
Dev. Var $q_{\text{est},P_{\text{tot}}} - q_{\text{meas},P_{\text{tot}}}$	4.101×10^{-4}	1.003×10^{-4}	2.284×10^{-5}	6.828×10^{-6}

The results across different horizon lengths with constant block length show that a short horizon of 10 still exhibits higher deviation and greater correction factor variance compared to the longer horizons. However, none of the output trajectories from any combination of tuning parameters exit the convergence region during the simulation. Increasing the horizon length leads to improved performance, as indicated by a decrease in variances. The ethylene measurements and correction factor from the MHE tuning with horizon lengths of 100 and 200 are shown in **Figure 5.10**.



(a) The true and estimated measurements of ethylene concentration in mol% from simulating test case with horizon lengths 100 and 200, consisting of 10 and 20 blocks respectively, after the disturbance is introduced at sample 500.



(b) The ethylene rate correction factor (unitless) for the same two horizon lengths.

Figure 5.10: The true and estimated measurements of ethylene concentration, along with the ethylene rate correction factor, are compared between horizon lengths of 100 with 10 blocks and 200 with 20 blocks. The figures show the results from sample 500, where the disturbance occurred.

By observing **Figure 5.10 (b)** it is the horizon of 200 that updates the ethylene correction factor faster than when the horizon is 100. This reduces the deviation between the actual and estimated measurements, as shown in **Figure 5.10 (a)**, where the estimated measurements match the actual measurements more quickly and closely compared to using a horizon length of 100. Using these parameters does not cause significant downswings, instead, it results in a more stable y_{Epp} output compared to the shorter horizon, as illustrated in **Figure 5.11** below.

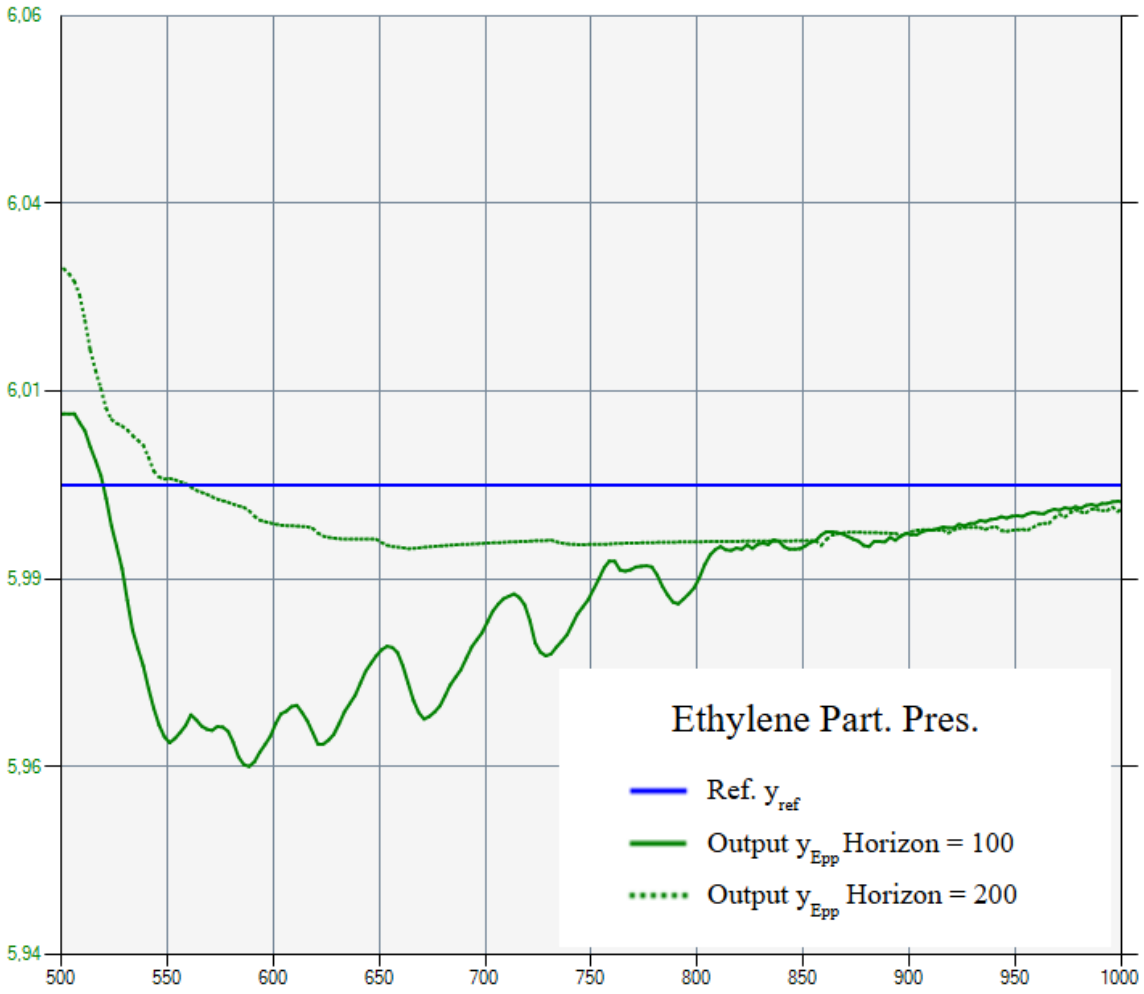


Figure 5.11: Ethylene partial pressure outputs in barg when using horizon lengths 100 and 200 together with the reference y_{Epp}. Aside from horizon length and block distribution all other tuning parameters were kept to the standard tuning. The figure shows the results from sample 500, where the disturbance occurred.

Block Distribution:

The results for three different block distributions over a 100-sample-long horizon are shown in **Table 5.14** below, one with uniform distribution, one with longer blocks at the end of the horizon, and one with shorter blocks at the end. Weight matrices $\mathbf{Q_e}$ and $\mathbf{R}$, as well as number of blocks in the horizon were kept to the standard values.

Table 5.14: Effect of tuning the elements in weighting matrix $\mathbf{R}$ while keeping $\mathbf{Q_e}$ constant on convergence sample, correction factor variance, and measurement deviation variance in test case. All combinations of the parameters keep the output within the convergence region after the error is introduced. Thus, all outputs converge by sample 500. The result of the standard tuning is shown in bold text.

Block Length	20, 20, 20 20, 20	10, 10, 20 20 40	40, 20, 20 10, 10
Corr. Var e_{H_2}	$\mathbf{8.624 \times 10^{-3}}$	6.952×10^{-3}	1.994×10^{-2}
Corr. Var e_{C_2}	$\mathbf{1.284 \times 10^{-1}}$	1.267×10^{-1}	1.285×10^{-1}
Corr. Var e_{C_4}	$\mathbf{2.141 \times 10^{-2}}$	6.952×10^{-3}	2.222×10^{-2}
Corr. Var e_{N_2}	$\mathbf{1.046 \times 10^{1}}$	8.312×10^{1}	2.749×10^{1}
Dev. Var $q_{\text{est},H_2} - q_{\text{meas},H_2}$	$\mathbf{4.151 \times 10^{-5}}$	2.755×10^{-4}	5.860×10^{-6}
Dev. Var $q_{\text{est},C_2} - q_{\text{meas},C_2}$	$\mathbf{5.178 \times 10^{-4}}$	3.319×10^{-3}	9.780×10^{-5}
Dev. Var $q_{\text{est},C_4} - q_{\text{meas},C_4}$	$\mathbf{4.794 \times 10^{-10}}$	1.939×10^{-8}	7.983×10^{-10}
Dev. Var $q_{\text{est},P_{\text{tot}}} - q_{\text{meas},P_{\text{tot}}}$	$\mathbf{3.838 \times 10^{-5}}$	6.318×10^{-5}	2.687×10^{-5}

The results show that using longer blocks at the end of the horizon reduces the variance of some correction factors, all but not e_{N_2} which increase slightly. However, all deviation variances increase, indicating worse estimation performance compared to using uniform blocks. In contrast, using shorter blocks toward the end of the horizon decreases all deviation variances, except for butene, C_4. On the other hand, all correction factor variances increase, except for hydrogen, H_2. This suggests that achieving lower deviation variance comes at the cost of greater variance of the correction factors.

Amount of Blocks:

The results of four different amounts of blocks with horizon length of 100 samples are presented in **Table 5.15**below. Other parameters are kept as standard values.

Table 5.15: Effect of different amounts of blocks in an horizon with $N = 100$ on convergence sample, correction factor variance, and measurement deviation variance in test case. All block distributions are uniform. The result of the standard tuning is shown in bold text.

Number of Blocks	1	2	**5**	10
Convergence Sample y_{HE}	500	628	**500**	500
Convergence Sample y_{BE}	500	500	**500**	500
Convergence Sample y_{Epp}	500	648	**500**	500
Convergence Sample $y_{P_{\mathrm{tot}}}$	500	500	**500**	500
Corr. Var $e_{\mathrm{H_2}}$	3.764×10^{-3}	5.470×10^{-3}	$\mathbf{8.624 \times 10^{-3}}$	3.529×10^{-3}
Corr. Var $e_{\mathrm{C_2}}$	1.242×10^{-1}	1.236×10^{-1}	$\mathbf{1.284 \times 10^{-1}}$	1.287×10^{-1}
Corr. Var $e_{\mathrm{C_4}}$	2.042×10^{-2}	2.092×10^{-2}	$\mathbf{2.141 \times 10^{-2}}$	2.177×10^{-2}
Corr. Var $e_{\mathrm{N_2}}$	8.611×10^{-1}	2.460×10^{1}	$\mathbf{1.046 \times 10^{1}}$	4.527
Dev. Var $q_{\mathrm{est,H_2}} - q_{\mathrm{meas,H_2}}$	2.090×10^{-4}	8.047×10^{-5}	$\mathbf{4.151 \times 10^{-5}}$	4.855×10^{-5}
Dev. Var $q_{\mathrm{est,C_2}} - q_{\mathrm{meas,C_2}}$	1.077×10^{-2}	6.248×10^{-3}	$\mathbf{5.178 \times 10^{-4}}$	5.951×10^{-5}
Dev. Var $q_{\mathrm{est,C_4}} - q_{\mathrm{meas,C_4}}$	1.053×10^{-6}	1.078×10^{-7}	$\mathbf{4.794 \times 10^{-10}}$	6.582×10^{-10}
Dev. Var $q_{\mathrm{est},P_{\mathrm{tot}}} - q_{\mathrm{meas},P_{\mathrm{tot}}}$	3.595×10^{-4}	1.171×10^{-4}	$\mathbf{3.838 \times 10^{-5}}$	2.284×10^{-5}

When comparing different numbers of blocks within the horizon, increasing the number of blocks reduced the deviation variances, while the correction factor variance showed only a very small change across all horizons. Only the outputs y_{HE} and y_{Epp} when dividing the horizon in 2 blocks reacted enough to the catalyst error to leave the convergence region of ± 1%. However, increasing the number of blocks from 5 to 10 does not result in improvement. This indicates that while adding more blocks can improve performance, the increase in computational time eventually outweighs the benefits. In simulations performed using one block in the horizon, the computational time was 0.4 seconds per sample, while in simulations using 10 blocks, the computational time was 1.1 seconds per sample.

Optimal Tuning Parameters:

The trends from tuning the previous parameters of MHE shows that longer horizons in combination with more blocks increase the performance of the estimator. Additionally, block distributions with shorter blocks towards the end of the horizon showed lower deviation variance on the expense of higher correction factor variances. The MHE tuning with horizon of length 200 across 20 blocks shown in **Table 5.13** that had the best performance of previous tests, was therefore compared with a smaller horizon of 100 across the same amount of blocks, leading to overall shorter block lengths. The result is shown in **Table 5.16** below.

Table 5.16: The effect of different combinations of tuning parameters given by the trends.

Horizon Length	Horizon = 200	Horizon = 100	Horizon = 200
Number of Blocks	20	20	10
Block Dist.	Uniform	Uniform	30, 30, 30, 20, 20, 20, 20 10, 10
Corr. Var e_{H_2}	2.728×10^{-3}	3.929×10^{-3}	9.651×10^{-3}
Corr. Var e_{C_2}	1.275×10^{-1}	1.392×10^{-1}	1.279×10^{-1}
Corr. Var e_{C_4}	2.154×10^{-2}	2.133×10^{-2}	2.316×10^{-2}
Corr. Var e_{N_2}	1.251	2.534	9.433×10^{1}
Dev. Var $q_{est,H_2} - q_{meas,H_2}$	3.044×10^{-6}	2.856×10^{-5}	2.580×10^{-3}
Dev. Var $q_{est,C_2} - q_{meas,C_2}$	5.401×10^{-5}	5.753×10^{-5}	1.461×10^{-4}
Dev. Var $q_{est,C_4} - q_{meas,C_4}$	7.992×10^{-10}	6.382×10^{-10}	6.345×10^{-10}
Dev. Var $q_{est,P_{tot}} - q_{meas,P_{tot}}$	6.828×10^{-6}	2.087×10^{-5}	3.600×10^{-5}

The results indicate that shortening both the horizon length and the block lengths does not lead to a performance improvement. Therefore, the optimal tuning for MHE in test case was determined to use a horizon length of 200 and 20 blocks, with all other parameters set to their standard values.

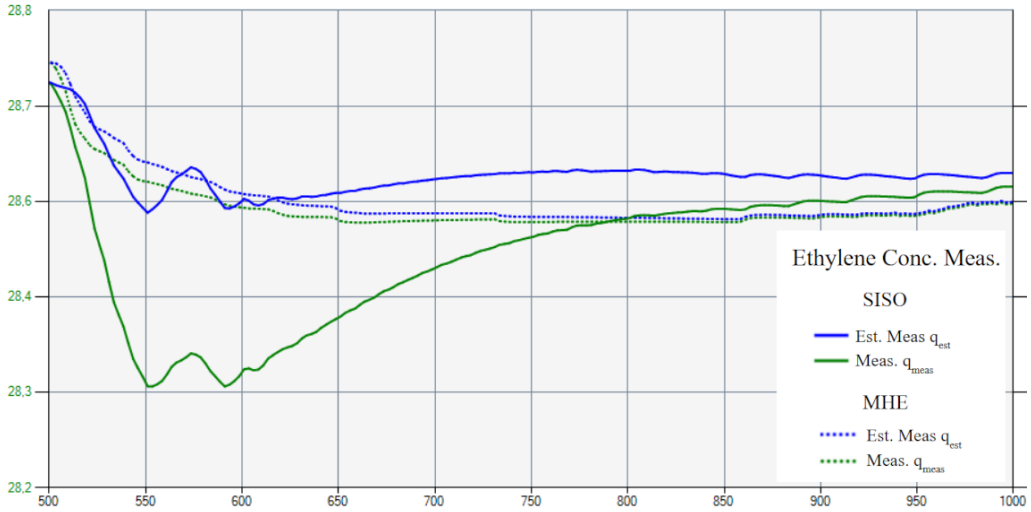
5.6 Performance Evaluation and Analysis of the Estimators on the Test Case

The results of comparing the optimal tuning of MHE with SISO estimator is shown in **Table 5.17** below.

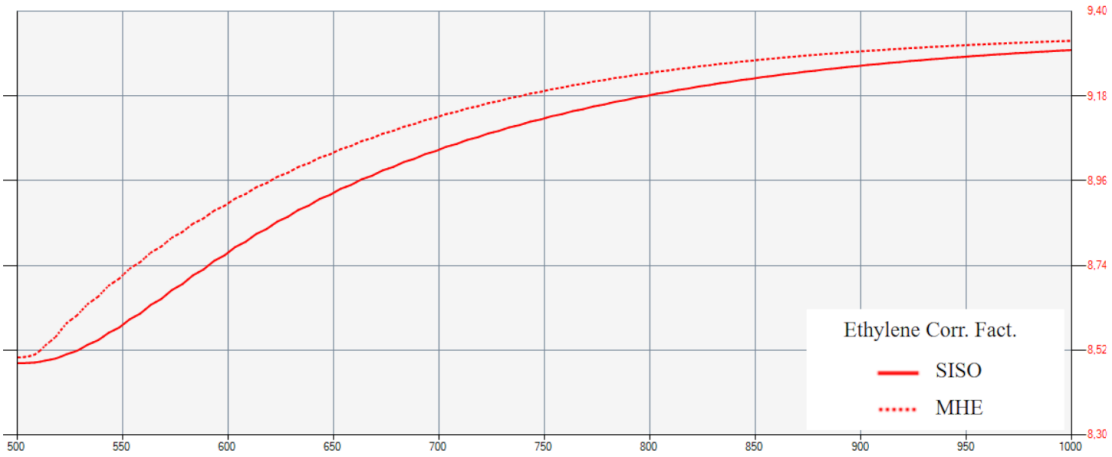
Table 5.17: Comparison of results between the SISO estimator and MHE for test case, showing convergence sample, correction factor variance, and measurement deviation variance. None of the estimator tunings left the convergence interval of ± 1%. All outputs converge at sample 500.

Estimator	SISO	MHE
Corr. Var e_{H_2}	4.412×10^{-4}	2.728×10^{-3}
Corr. Var e_{C_2}	1.260×10^{-1}	1.275×10^{-1}
Corr. Var e_{C_4}	1.766×10^{-2}	2.154×10^{-2}
Corr. Var e_{N_2}	2.968×10^{-1}	1.251
Dev. Var $q_{est,H_2} - q_{meas,H_2}$	2.990×10^{-5}	3.044×10^{-6}
Dev. Var $q_{est,C_2} - q_{meas,C_2}$	4.480×10^{-3}	5.401×10^{-5}
Dev. Var $q_{est,C_4} - q_{meas,C_4}$	6.144×10^{-7}	7.992×10^{-10}
Dev. Var $q_{est,P_{tot}} - q_{meas,P_{tot}}$	2.196×10^{-5}	6.828×10^{-6}

Aside from the hydrogen correction factor variance, the results show that both the SISO and MHE estimators yield similarly low correction factor variances. However, MHE achieves significantly lower deviation variances compared to SISO, which suggests that MHE performs better than the SISO estimator in test case. This is expected since a multivariable estimator should perform better than a SISO estimator because MHE uses data from previous samples to estimate the current state. The estimates of the ethylene concentration measurements for both estimators are compared in **Figure 5.12** below.



(a) The true and estimated measurements of ethylene concentration in mol% from simulating test case with SISO and MHE, after the disturbance is introduced at sample 500.



(b) The ethylene rate correction factor (unitless) for SISO and MHE.

Figure 5.12: Comparison of the performance of MHE and SISO in estimating ethylene concentration and the ethylene rate correction factor. The figures show the results from sample 500, where the disturbance occurred.

The results show that MHE handles the catalyst disturbance a lot faster than using SISO. In **Figure 5.12 (b)** it is shown that the ethylene correction factor approaches the new correction factor steady state faster than using SISO. This minimizes the deviation between estimated and true measurements shown in **Figure 5.12 (a)**. This leads to much less impact on, for example, the total pressure output shown in **Figure 5.13** below.

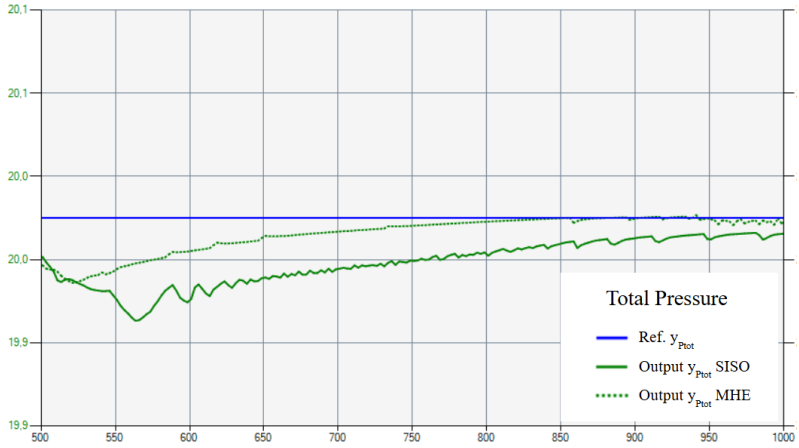


Figure 5.13: Total pressure outputs in barg when using SISO and MHE with the reference $y_{P_{\text{tot}}}$. The figure shows the results from sample 500, where the disturbance occurred.

5.7 Performance Evaluation and Analysis of the Estimators on True Industrial Measurements

In this section, the performance of each estimator, MHE and SISO, will be compared under different datasets based on real industrial measurements to evaluate their effectiveness. Since the real measurements are used, it will be an open-loop simulation, i.e. no control actions will be applied. Only the estimation performance will be evaluated. The figure below shows the control system used for the estimation when no controller is included.

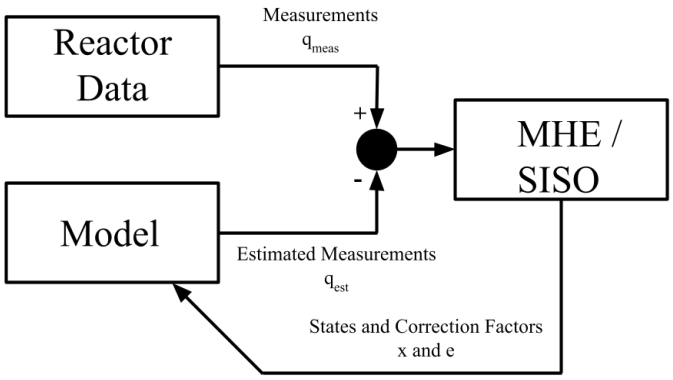


Figure 5.14: Open loop system for estimator evaluation, where the controller is decoupled. At each sample, real measurements from the dataset are compared to the estimated measurements produced by the estimator model. Simulations with each estimator algorithm enables performance evaluation.

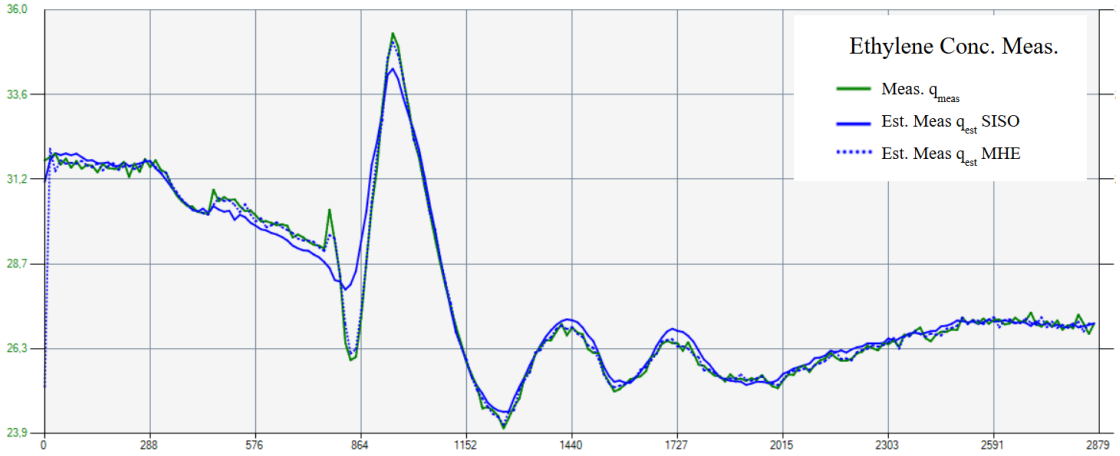
5.7.1 Dataset 1: Transition

This datasets shows a transition to a new end product where the setpoints of y_{Epp} and y_{HE} are decreased, y_{BE} is increased, while $y_{P_{\mathrm{tot}}}$ is kept constant. The transition occurs at around sample 700. The purpose of this dataset is to evaluate the performance of the estimators during product changeovers, when transitions occur. The performance of the estimators is presented in **Table 5.18** below.

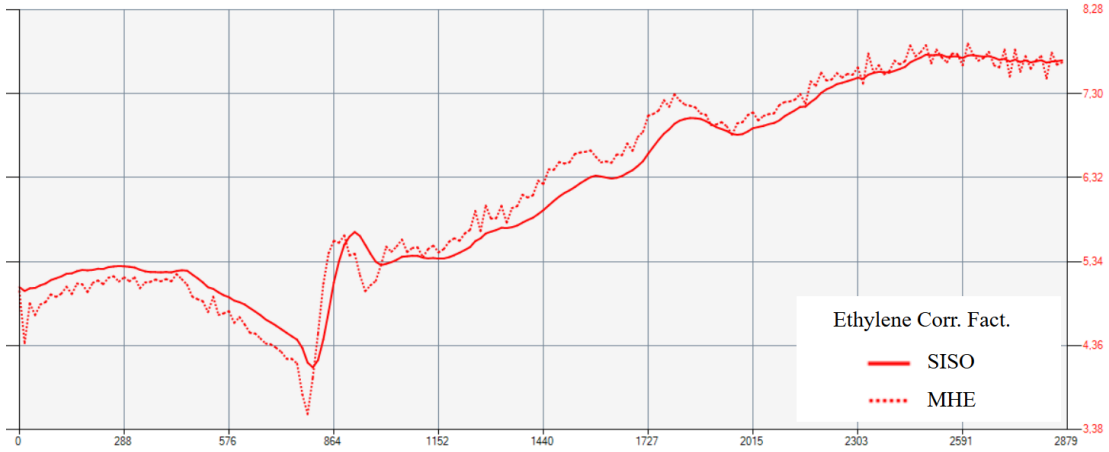
Table 5.18: The performance of the estimators with correction factor variance and measurement deviation variance in dataset 1.

Estimator	SISO	MHE
Corr. Var $e_{\mathrm{H_2}}$	2.013×10^{-2}	2.787×10^{-2}
Corr. Var $e_{\mathrm{C_2}}$	1.166	1.376
Corr. Var $e_{\mathrm{C_4}}$	1.497×10^{-1}	2.498×10^{-1}
Corr. Var $e_{\mathrm{N_2}}$	4.532×10^{1}	9.124×10^{1}
Dev. Var $q_{\mathrm{est,H_2}} - q_{\mathrm{meas,H_2}}$	6.926×10^{-4}	5.867×10^{-4}
Dev. Var $q_{\mathrm{est,C_2}} - q_{\mathrm{meas,C_2}}$	1.707×10^{-1}	7.559×10^{-2}
Dev. Var $q_{est,\mathrm{C_4}} - q_{\mathrm{meas,C_4}}$	2.879×10^{-3}	3.897×10^{-3}
Dev. Var $q_{\mathrm{est},P_{\mathrm{tot}}} - q_{\mathrm{meas},P_{\mathrm{tot}}}$	3.904×10^{-3}	6.657×10^{-3}

The results from **Table 5.18** shows clearly that the correction factor variances of the MHE estimator is higher than with the SISO estimator. This can also be seen in **Figure 5.15 (b)** and **Figure 5.16 (b)** where the correction factor of ethylene and nitrogen are more fluctuating using MHE. While the deviation variances for hydrogen and ethylene slightly decrease with the use of MHE, those for butene and total pressure increase. Measurements of ethylene concentration and total pressure are shown in **Figure 5.15 (a)** and **Figure 5.16 (a)** below.

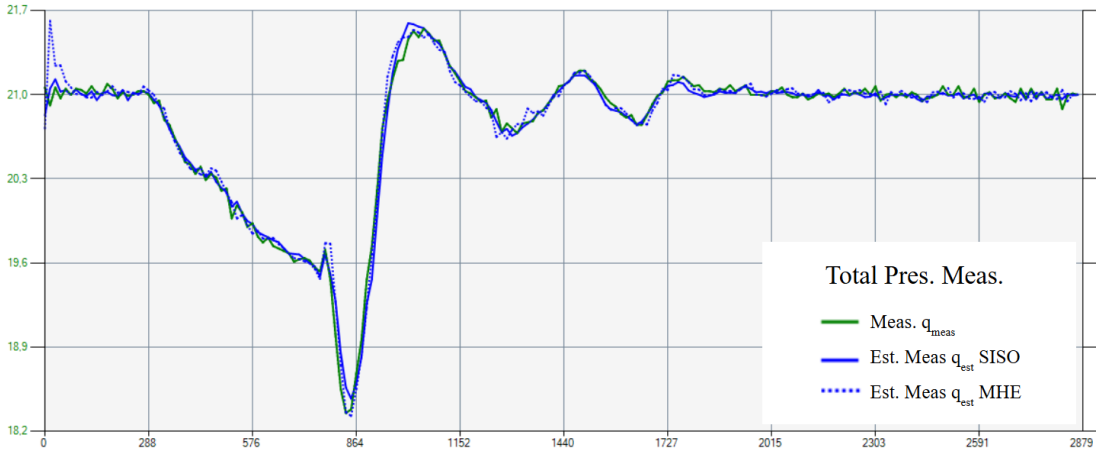


(a) The true and estimated measurements of ethylene concentration in mol% from simulating dataset 1 with SISO and MHE.

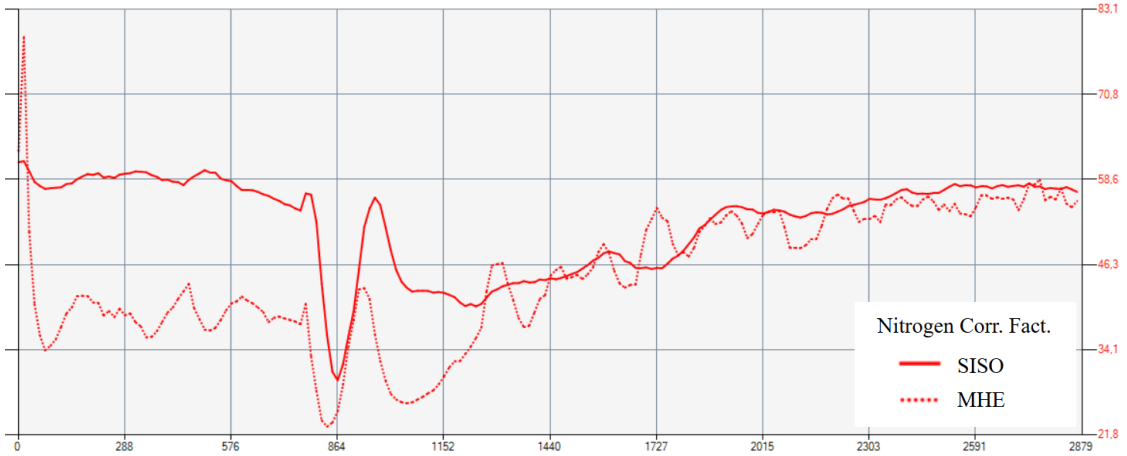


(b) The ethylene rate correction factor (unitless) for SISO and MHE.

Figure 5.15: The true and estimated measurements of ethylene concentration and ethylene rate correction factor for SISO and MHE in dataset 1.



(a) The true and estimated measurements of total pressure in the reactor given in barg with SISO and MHE.



(b) The nitrogen correction factor in kg/h for SISO and MHE.

Figure 5.16: The true and estimated measurements of total pressure and the nitrogen correction factor for SISO and MHE in dataset 1.

The figures comparing the estimated measurements of both estimators show that MHE performs significantly worse than SISO at the first samples of the simulation. This suggests that MHE is more sensitive to deviations in the initial state estimates. However, overall the performance of the two estimators are very similar for test case.

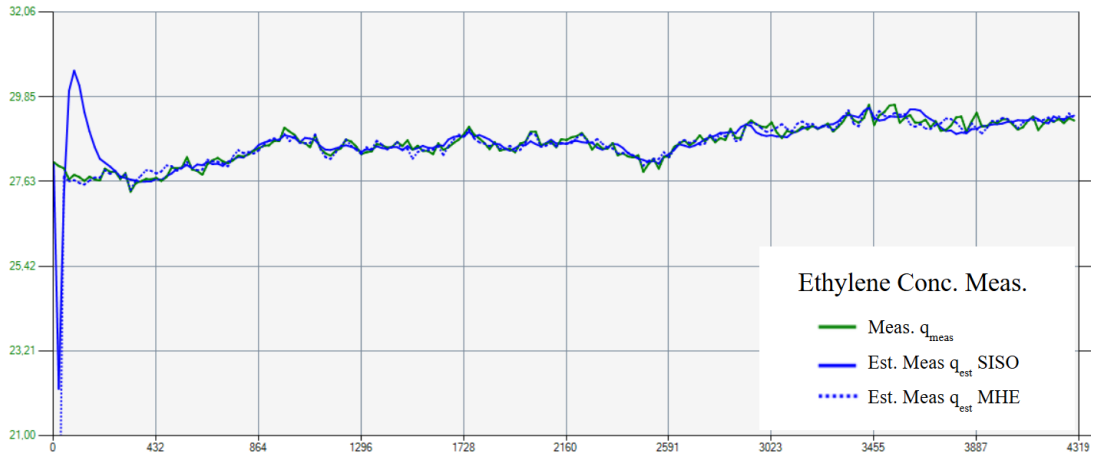
5.7.2 Dataset 2: Start-Up Process of the Reactor

The purpose of this dataset is to evaluate the performance of the estimators during the start-up process, in order to determine the best initial values to use in the MPC controller. The performance of the estimators in tracking the current states is presented in **Table 5.19** and **Figures 5.17** and **5.18** below.

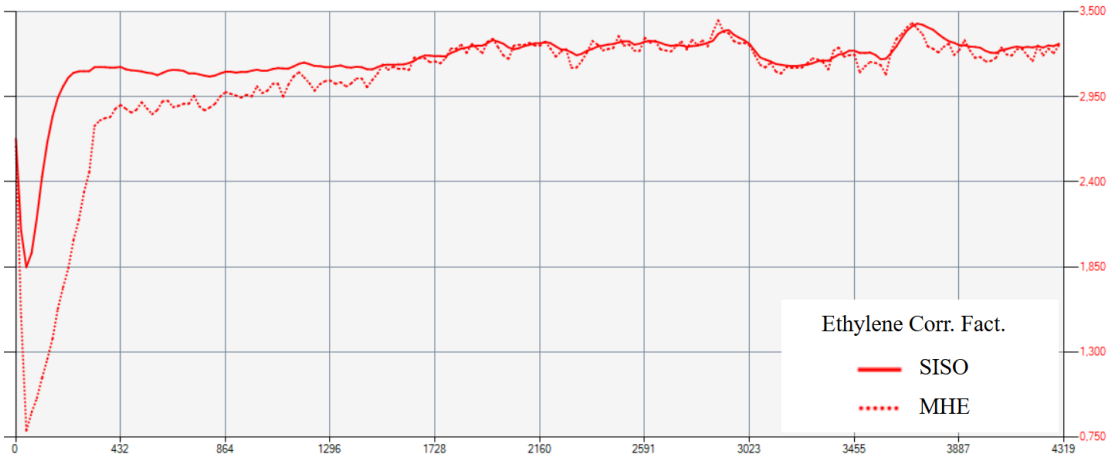
Table 5.19: The performance of the estimators under correction factor variance and measurement deviation variance on the dataset 2.

Estimator	SISO	MHE
Corr. Var $e_{\mathrm{H_2}}$	6.967×10^{-1}	4.168×10^{-1}
Corr. Var $e_{\mathrm{C_2}}$	4.152×10^{-2}	2.029×10^{-1}
Corr. Var $e_{\mathrm{C_4}}$	3.877×10^{-2}	8.702×10^{-2}
Corr. Var $e_{\mathrm{N_2}}$	1.725×10^{2}	4.241×10^{2}
Dev. Var $q_{\mathrm{est,H_2}} - q_{\mathrm{meas,H_2}}$	6.971×10^{-1}	2.527
Dev. Var $q_{\mathrm{est,C_2}} - q_{\mathrm{meas,C_2}}$	3.980×10^{-1}	1.819
Dev. Var $q_{est,\mathrm{C_4}} - q_{\mathrm{meas,C_4}}$	7.932×10^{-5}	1.369×10^{-5}
Dev. Var $q_{\mathrm{est},P_{\mathrm{tot}}} - q_{\mathrm{meas},P_{\mathrm{tot}}}$	1.114×10^{-1}	5.173×10^{-2}

According to **Table 5.19**, the MHE estimator results in higher correction factor variance for all cases except hydrogen, where the variance is slightly lower. The deviation variances for butene and total pressure using MHE are slightly higher, but slightly lower for hydrogen and ethylene. Since the differences are not significant, it can be concluded that their performances are similar. Although both estimators are exposed to initial errors, MHE begins with a very low estimate, whereas SISO overshoots the measurements, which can be seen in **Figure 5.17 (a)** and **Figure 5.18 (a)**. As shown in **Figure 5.17 (b)**, the initially low estimate from MHE results in a slower convergence to the steady-state value of the ethylene correction factor compared to SISO, which reaches it more quickly. However in **Figure 5.18 (b)**, even though the initial estimate of the nitrogen correction factor is less accurate with MHE, it reaches the steady-state value of around 70 faster than with SISO.

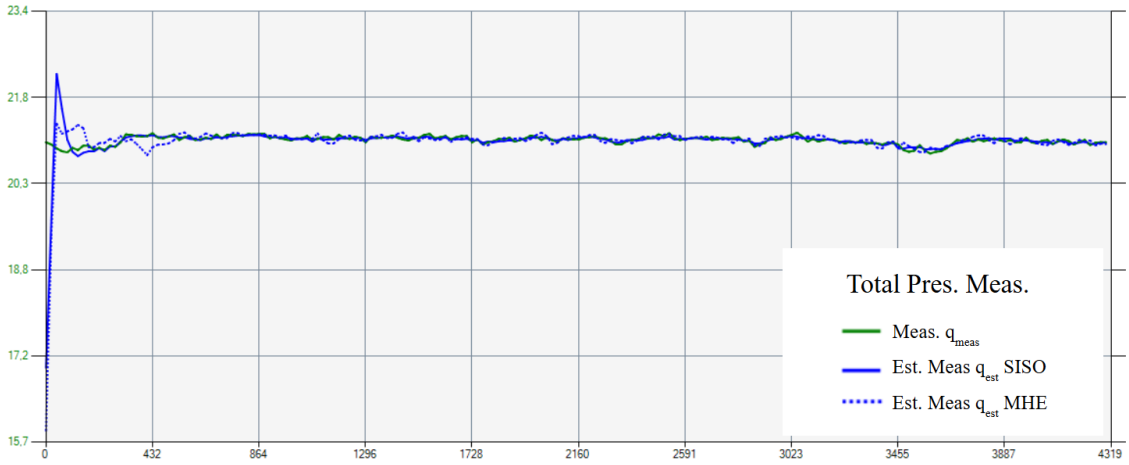


(a) The true and estimated measurements of ethylene concentration in mol% with SISO and MHE.

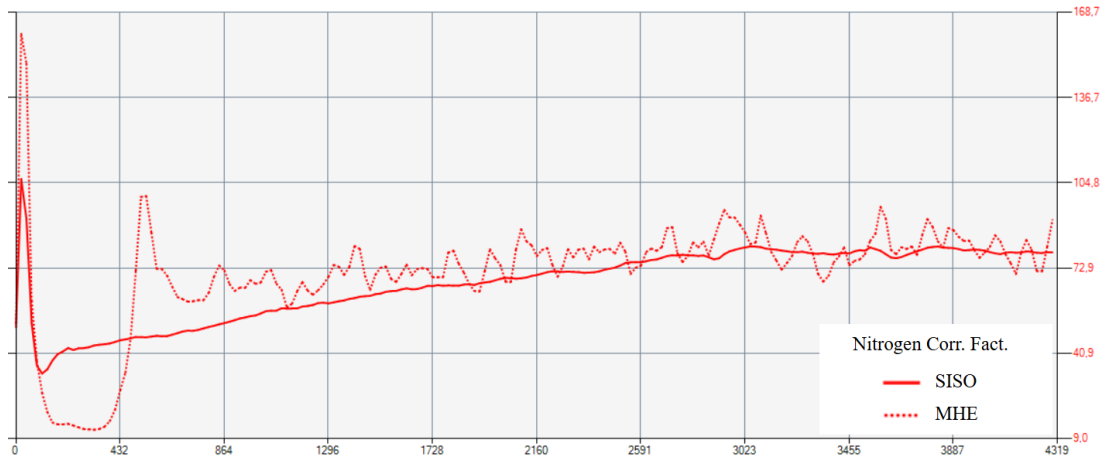


(b) The ethylene rate correction factor (unitless) for SISO and MHE.

Figure 5.17: The true and estimated measurements of ethylene concentration and the ethylene rate correction factor for SISO and MHE in dataset 2.



(a) The true and estimated measurements of total pressure in the reactor given in barg with SISO and MHE.



(b) The nitrogen correction factor in kg/h for SISO and MHE.

Figure 5.18: The true and estimated measurements of total pressure and the nitrogen correction factor for SISO and MHE in dataset 2.

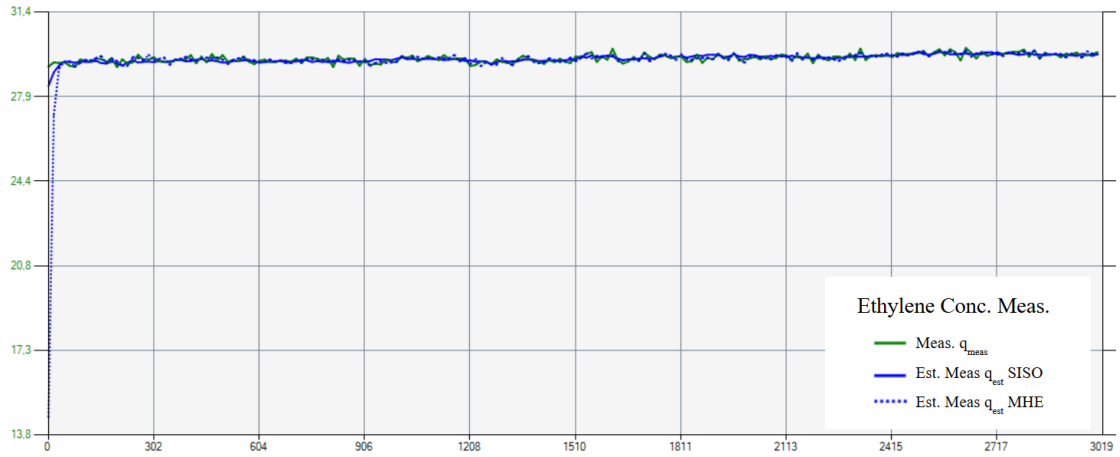
5.7.3 Dataset 3: Steady State

In this dataset, the system is in steady state, meaning that the same grade (same product) of polyethylene is being produced. The purpose of using this dataset is to evaluate the performance when the same product is consistently produced. The performance of the estimators is presented below.

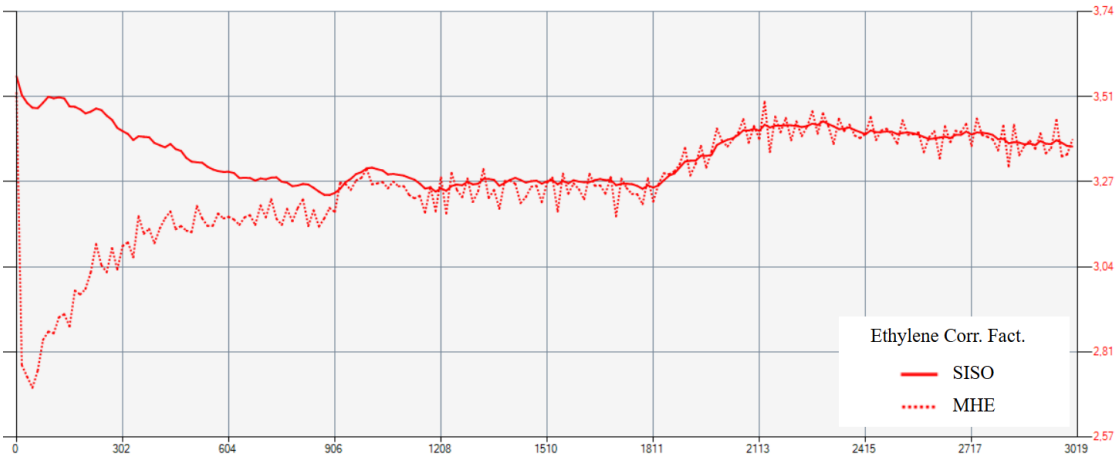
Table 5.20: The performance of the estimators under correction factor variance and measurement deviation variance on the dataset 3.

Estimator	SISO	MHE
Corr. Var $e_{\mathrm{H_2}}$	1.701×10^{-1}	6.785×10^{-2}
Corr. Var $e_{\mathrm{C_2}}$	5.552×10^{-3}	2.272×10^{-2}
Corr. Var $e_{\mathrm{C_4}}$	2.320×10^{-3}	1.528×10^{-2}
Corr. Var $e_{\mathrm{N_2}}$	1.020×10^{2}	6.353×10^{1}
Dev. Var $q_{\mathrm{est,H_2}} - q_{\mathrm{meas,H_2}}$	9.186×10^{-4}	7.193×10^{-3}
Dev. Var $q_{\mathrm{est,C_2}} - q_{\mathrm{meas,C_2}}$	1.705×10^{-2}	3.782×10^{-1}
Dev. Var $q_{est,\mathrm{C_4}} - q_{\mathrm{meas,C_4}}$	1.716×10^{-6}	3.465×10^{-6}
Dev. Var $q_{\mathrm{est},P_{\mathrm{tot}}} - q_{\mathrm{meas},P_{\mathrm{tot}}}$	1.189×10^{-3}	8.095×10^{-3}

From the results shown in **Table 5.20** using the SISO estimator results in lower correction factor and deviation variance compared to MHE. A comparison of ethylene concentration measurements using both MHE and SISO is shown in **Figure 5.19** below.



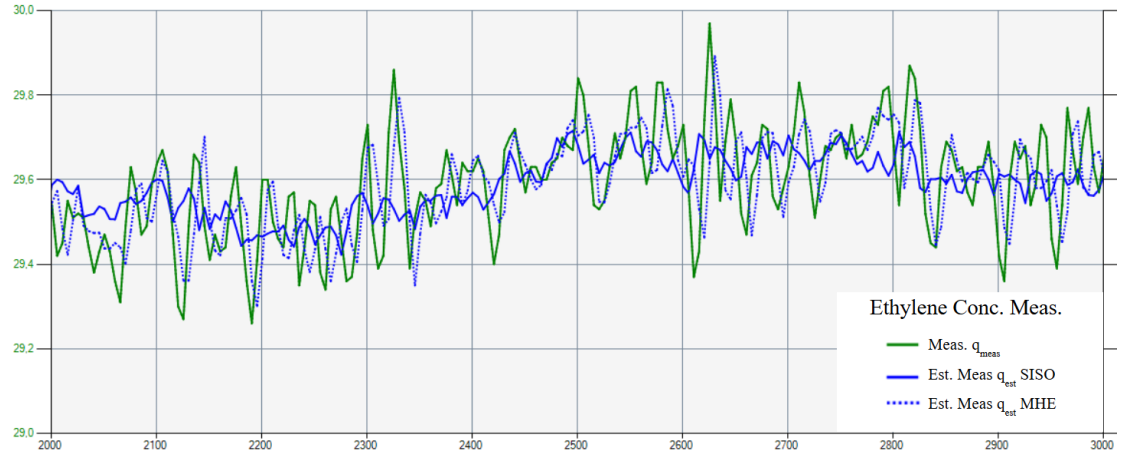
(a) The true and estimated measurements of ethylene concentration in mol% with SISO and MHE.



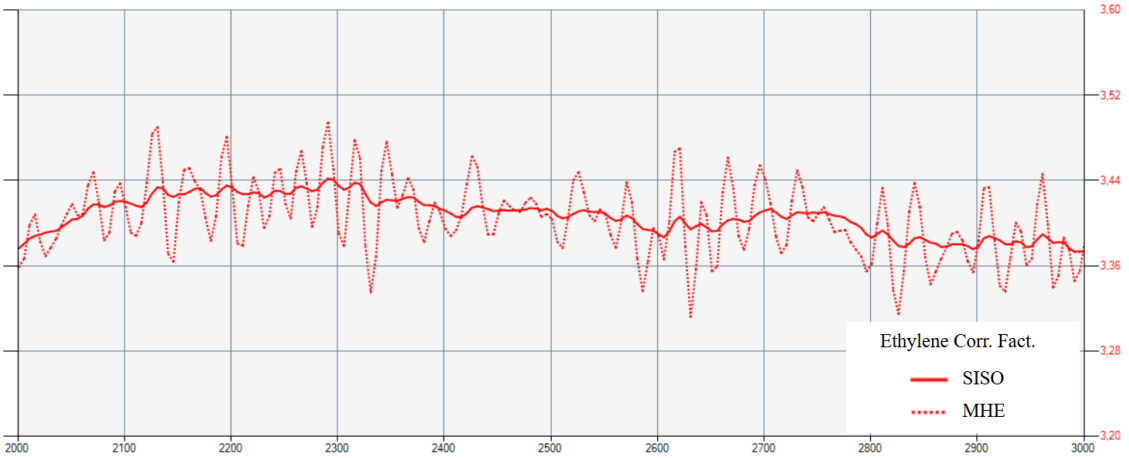
(b) The ethylene correction factor (unitless) for SISO and MHE.

Figure 5.19: The true and estimated measurements of ethylene and the ethylene correction factor for SISO and MHE in dataset 3.

The results show that, once again, MHE begins with a very low estimate of the ethylene concentration. The deviation between the estimated and true measurements is also shown in **Figure 5.19 (b)**, where the ethylene correction factor similarly starts at a low value before converging with the SISO estimation of the correction factor. However, a closer look at the ethylene concentration measurements between samples 2000 and 3000 (shown in **Figure 5.20 (a)**), reveals that the estimates using MHE are closer to the true values than those obtained using SISO. This suggests that MHE outperforms the SISO estimator. However, because of the low initial estimate, it results in an overall higher deviation variance.



(a) True and estimated measurements of ethylene concentration in mol% with SISO and MHE.



(b) Ethylene correction factor (unitless) for SISO and MHE.

Figure 5.20: The true and estimated measurements of ethylene and the ethylene correction factor for SISO and MHE in dataset 3.

5.8 Conclusion and Discussion of the Polyethylene Reactor System

Overall, the performance of both algorithms shows similar behavior. In the test case where a catalyst error occurs, it is clear that MHE performs better as an estimator, as all deviation variances are much lower, although some correction factor variances are slightly higher or at the same level as those from SISO. This indicates that MHE provides better estimation while maintaining the same variance level for the correction factor and the same convergence sample. However, when examining the actual measurements from the industrial process, MHE does not show a clear advantage, as the correction factor variances are slightly higher. This can be seen in **Tables 5.18** to **5.20**. The results are even clearer in **Figures 5.15** to **5.20**, where the correction factor graphs oscillate more. Despite the larger variation in

correction factors, MHE still performs estimation as well as SISO, since the deviation variances of MHE are similar to those of SISO. This is shown in the graphs, where the estimated output q_{est} from MHE matches the actual measurements as closely as the estimates from SISO. This suggests that MHE can perform just as well as SISO. Even though the performance is similar, replacing SISO with MHE is still not recommended, as MHE takes much longer to run. Using the MHE tuning with a horizon length of 200 across 20 blocks resulted in a computational time of 4.4 seconds per sample due to the complexity of the optimization problem, while the computational time using the SISO estimator was only 0.2 seconds per sample. Therefore, replacing SISO with MHE is not worthwhile, as the performance gain is negligible and the computation time is significantly longer.

Another factor that can affect the performance of MHE on the actual industrial dataset is that MHE has only been tuned to handle errors in the catalyst. This means that when other issues arise, which may not be related to the catalyst, it is not surprising that MHE does not perform well in those cases. Therefore, it may be worthwhile to tune MHE on a wider range of cases and select the optimal tuning parameters, as was done in the *Double Tank* case. This could potentially improve MHE's performance.

6

Conclusion and Discussion

The comparison between the SISO and MHE estimators in both the *Double Tank System* and the *Polyethylene Reactor System* leads to the conclusion that MHE has the potential to perform better than SISO in simpler systems like the *Double Tank System*. This is supported by the results, which show that the deviation variances are much lower for MHE than for SISO, indicating that MHE is a better estimator. Even in the more complex polyethylene reactor system, MHE can still perform better in test case, as the deviation variances are lower, although the correction factor variances are not lower than those from SISO. However, when using real industrial data, there are clear signs that MHE does not outperform SISO. Their performances are similar, which means that MHE can at least perform as well as SISO. However, the complexity of MHE is still much higher, which increases the computation time required to solve the MHE problem. Thus, MHE is not yet a suitable replacement for SISO due to these disadvantages. Even so, the performance of MHE in the test cases provides important evidence that it has good potential to outperform SISO, since MHE can handle multiple variables simultaneously, which is a significant advantage. Nevertheless, further work is needed to improve the algorithm.

7
Future Work

Working through this project provides perspectives from various angles. There are some positive aspects that are worth building upon, as well as negative aspects that could be improved. The following parts will present the possibilities for improving the performance of the Moving Horizon Estimation.

A New Strategy for Implementing the Correction Factor:

The correction factors used in both systems, *Double Tank* and *Polyethylene Reactor*, are implemented in a general and somewhat non-precise way, with one correction factor affecting each measurement. As already mentioned this is a very effective way of dealing with disturbances, since almost all disturbances can ultimately be described as an inflow/outflow of a state. An alternative approach to estimating disturbances in the system was tested in section 4.6. Where the purpose of the correction factor was not general, but specific to a certain type of disturbance. The disturbance model tested in the *Double Tank System* used a correction factor implemented directly to the outlet hole from the upper tank to the lower tank. In other words, the correction factor became a factor multiplied with the flow between the tanks, instead of a general inflow/outflow to the tank.

Future work in this area would mean finding a more disturbance-specific disturbance model for the *Polyethylene Reactor System*, where correction factor could be implemented to a disturbance source instead of as an additional inflow/outflow. An example of this is a disturbance model with a correction factor affecting the catalyst activity, which will in turn affect all gases in the polymerization reaction. The challenge in implementing new disturbance models in more complex systems is to find where the disturbances originate from. This demands more information of potential disturbances which should be investigated further.

Full Cost Function:

The cost function (3.9) used in this project is a simplified version. The full cost function is expressed as:

$$\begin{aligned} J = & \left(x_{k-N} - x(k-N)\right)^\top \mathbf{P} \left(x_{k-N} - x(k-N)\right) \\ & + \sum_{i=k-N}^{k-1} \left(x_{i+1} - \tilde{f}\left(x_i, e_i, u_i\right)\right)^\top \mathbf{Q_x} \left(x_{i+1} - \tilde{f}\left(x_i, e_i, u_i\right)\right) \\ & + \sum_{i=k-N}^{k-1} \Delta e_i^\top \mathbf{Q_e} \Delta e_i \\ & + \sum_{i=k-N}^{k} \left(q_{\text{meas},i} - q_{\text{est},i}\right)^\top \mathbf{R} \left(q_{\text{meas},i} - q_{\text{est},i}\right) \end{aligned}$$

In the simplified cost function, the first and second terms are neglected. The advantage of not including these two terms is that the optimization problem becomes much easier to solve. However, a major disadvantage is that the estimation then relies only on changes in the correction factor. In other words, only the correction factor can be optimized to compensate for disturbances occurring in the system. For the *Double Tank System*, it can be appropriate since the system is not complex. However, for a more complex problem, it can then be inappropriate. To address this problem, the second term

$$\sum_{i=k-N}^{k-1} \left(x_{i+1} - \tilde{f}\left(x_i, e_i, u_i\right)\right)^\top \mathbf{Q_x} \left(x_{i+1} - \tilde{f}\left(x_i, e_i, u_i\right)\right)$$

will play a crucial role. This term includes the state, x, which also needs to be optimized. By including the states in the optimization problem, the overall performance of optimization will improve, as the states are also optimized. This makes the optimization parameters, both x and e, more adaptive to the unknown disturbances that occurred to the system.

Bibliography

[1] Paul Kirvan
Article: *What is a control system?*, March, 2023. [Online]. Available: https://www.techtarget.com/whatis/definition/control-system

[2] Gavin Wright
Article: *Closed loop control system*, May, 2022. [Online]. Available: https://www.techtarget.com/whatis/definition/closed-loop-control-system

[3] Prof. Scott Moura
Book: *Energy Systems and Control*, Spring, 2018. [Online]. Available: https://ecal.studentorg.berkeley.edu

[4] Lennart Ljung
Book: *System Identification: Theory for the User*, 1999, 142-143. [Online]. Available: https://www.mit.bme.hu

[5] Per-Ola Larsson
Master thesis: *Optimization of Low-Level Controllers and High-Level Polymer Grade Changes*, 2011, 162. [Online]. Available: https://lucris.lub.lu.se/ws/portalfiles/portal/3857116/2174455.pdf

[6] Jan Drgona, Javier Arroyo, Iago Cupeiro Figueroa, David Blum
Review article: *All you need to know about model predictive control for buildings*, 29 September, 2020, pages 202-203. [Online]. Available: https://pdf.sciencedirectassets.com

[7] Peter Hippe, Joachim Deutscher
Book: *Design of Observer-based Compensators*, 2009, pages 28-31. [Online]. Available: https://link.springer.com/book/10.1007/978-1-84882-537-6

[8] James B. Rawlings, David Q. Mayne, Moritz M. Diehl
Book: *Model Predictive Control: Theory, Computation, and Design 2nd Edition*, October, 2018, pages 39-41. [Online]. Available: https://sites.engineering.ucsb.edu

[9] Jørgensen, John Bagterp
Doctoral thesis: *Moving Horizon Estimation and Control*, December, 2004, pages 111-121. [Online]. Available: https://backend.orbit.dtu.dk

[10] Tor Larsson
Master thesis: *Moving Horizon Estimation for JModelica.org*, 2015, pages 9-66. [Online]. Available: https://lup.lub.lu.se

[11] Vipin Jain
Article: *Types of Controllers | Proportional Integral and Derivative Controllers*, 6 May, 2024. [Online]. Available: https://www.electrical4u.com/types-of-controllers-proportional-integral-derivative-controllers/

[12] Borreggine, Simone and Monopoli, Vito Giuseppe and Rizzello, Gianluca and Naso, David and Cupertino, Francesco and Consoletti, Rinaldo
Article: *A Review on Model Predictive Control and its Applications in Power Electronics*, 2019. [Online]. Available: https://ieeexplore.ieee.org

[13] Tor A. Johanse
PDF document: *Chapter: Introduction to Nonlinear Model Predictive Control and Moving Horizon Estimation*, pages 1-47. [Online]. Available: https://torarnj.folk.ntnu.no/nonlinear.pdf

[14] Loubna Bouzenzen
Article: *What is Sequential Quadratic Programming (SQP)?*, October 6, 2024. [Online]. Available: https://medium.com/@Loubna-DS/what-is-sequential-quadratic-programming-sqp-4d5a34b4d53b

[15] Ben Goodman
Article: *Sequential quadratic programming*, Spring 2016. [Online]. Available: https://optimization.cbe.cornell.edu

[16] Jasmeer Ramlal, Kenneth V. Allsford, John D. Hedengren
Article: *MOVING HORIZON ESTIMATION FOR AN INDUSTRIAL GAS PHASE POLYMERIZATION REACTOR*, Spring 2016, [Online]. Available: https://www.sciencedirect.com

[17] Romeo M. Flores
Book: *Coal and Coalbed Gas*, 2013, Pages 597-649, [Online]. Available: https://www.sciencedirect.com

A

Appendix

A.1 Derivation of Cost Function

According to extended state, the equations of nonlinear modeled dynamic system are formulated as:

$$\begin{aligned} z_{k+1} &= \hat{f}\left(z_k, u_k\right) \\ q_{\text{est},k} &= g\left(z_k, u_k\right) \end{aligned}$$

Assume that the actual dynamic system, the plant, is the modeled system that is affected by process noises, $w = [w_x \quad w_e]^\top$, and that measurement errors occur due to measurement noise, n.

$$\begin{aligned} z_{k+1} &= \hat{f}\left(z_k, u_k\right) + w_k \\ q_{\text{meas},k} &= g\left(z_k, u_k\right) + n_k = q_{\text{est},k} + n_k \end{aligned}$$

The cost function that minimizes the process noises and measurement noise is defined as:

$$J = \left(x_{k-N} - x(k-N)\right)^\top \mathbf{P}\left(x_{k-N} - x(k-N)\right) + \sum_{i=k-N}^{k-1} w_i^\top \mathbf{Q} w_i + \sum_{i=k-N}^{k} n_i^\top \mathbf{R} n_i$$

combining with the system's dynamic equations, the cost function can be reformulated as:

$$\begin{aligned} J = {} & \left(x_{k-N} - x(k-N)\right)^\top \mathbf{P}\left(x_{k-N} - x(k-N)\right) \\ & + \sum_{i=k-N}^{k-1} \underbrace{\left(z_{i+1} - \hat{f}\left(z_i, u_i\right)\right)^\top}_{\begin{bmatrix} \mathbf{Q_x} & 0 \\ 0 & \mathbf{Q_e} \end{bmatrix}} \underbrace{\left(z_{i+1} - \hat{f}\left(z_i, u_i\right)\right)}_{\begin{bmatrix} x_{i+1} \\ e_{i+1} \end{bmatrix} - \begin{bmatrix} \tilde{f}\left(x_i, e_i, u_i\right) \\ e_i \end{bmatrix}} \\ & + \sum_{i=k-N}^{k} \left(q_{\text{meas},i} - q_{\text{est},i}\right)^\top \mathbf{R}\left(q_{\text{meas},i} - q_{\text{est},i}\right) \end{aligned}$$

which can be expanded as:

$$\begin{aligned} J = &\left(x_{k-N} - x(k-N)\right)^\top \mathbf{P} \left(x_{k-N} - x(k-N)\right) \\ &+ \sum_{i=k-N}^{k-1} \left(x_{i+1} - \tilde{f}\left(x_i, e_i, u_i\right)\right)^\top \mathbf{Q_x} \left(x_{i+1} - \tilde{f}\left(x_i, e_i, u_i\right)\right) \\ &+ \sum_{i=k-N}^{k-1} \left(e_{i+1} - e_i\right)^\top \mathbf{Q_e} \left(e_{i+1} - e_i\right) \\ &+ \sum_{i=k-N}^{k} \left(q_{\text{meas},i} - q_{\text{est},i}\right)^\top \mathbf{R} \left(q_{\text{meas},i} - q_{\text{est},i}\right) \end{aligned}$$

Since Borealis does not have any requirements or references related to the state, the first term, *arrival cost*, of the cost function can be neglected. The another common form of the arrival cost

$$\left(x_0 - x(0)\right)^\top \mathbf{P} \left(x_0 - x(0)\right)$$

includes the initial state as an optimization parameter. However, this is not considered significant in this case. One reason is that the company already has another controller running continuously, which makes it easy to track the current situation of the factory, the current states are always known.

$$x_0 \approx x(0)$$

Therefore, to reduce the complexity of the optimization problem, this term can be neglected. To further simplify the optimization problem, Borealis chooses to optimize only the disturbance parameter or correction factor, e, which is used to adjust the model to better match real-world behavior. The mismatch between the estimated model and the real-world plant can be considered a result of unknown disturbances. Therefore, adjusting the correction factor that represents these disturbances should be sufficient enough to address the mismatch. Thus, the matrix $\mathbf{Q_x}$ is set to be zero. This approach will also reduce computation time, as the number of parameters being optimized is decreased. The optimization problem can be formulated as:

$$\begin{aligned} \underset{\Delta e_{k-N}:\Delta e_{k-1}}{\text{minimize}} \quad J = &\sum_{i=k-N}^{k} \left(q_{\text{meas},i} - q_{\text{est},i}\right)^\top \mathbf{R} \left(q_{\text{meas},i} - q_{\text{est},i}\right) \\ &+ \sum_{i=k-N}^{k-1} \Delta e_i^\top \mathbf{Q_e} \underbrace{\Delta e_i}_{e_{i+1} - e_i} \end{aligned}$$

subject to

$$\begin{gathered} q_{\text{meas},i} = g\left(z_i, u_i\right) + n_i = q_{\text{est},i} + n_i \\ e_{\min} \leq e_i \leq e_{\max} \\ \Delta e_{\min} \leq \Delta e_i \leq \Delta e_{\max} \\ n_i \in \mathbb{N}, \quad \forall i \in [k-N, k] \end{gathered}$$

A.2 Derivation of the Quadratic Programming Formulation

To enable the use of SQP as a solver for the optimization problem, the problem

$$\begin{aligned}\underset{\Delta e_{\text{opt},1}:\Delta e_{\text{opt},n_\text{b}}}{\text{minimize}} \quad J = &\sum_{i=k-N}^{k} \left(q_{\text{meas},i} - q_{\text{est},i}\right)^\top \mathbf{R} \left(q_{\text{meas},i} - q_{\text{est},i}\right) \\ &+ \sum_{j=1}^{n_\text{b}} \Delta e_{\text{opt},j}^\top \mathbf{Q_e} \Delta e_{\text{opt},j}\end{aligned} \tag{A.1}$$

subject to

$$\begin{aligned} q_{\text{meas},i} = g(x_i, e_i, u_i) + n_i = q_{\text{est},i} + n_i,\ n_i \in \mathbb{N} \qquad & \forall i \in [k-N, k] \\ e_{\min} \leq e_{\text{opt},j} \leq e_{\max}, \qquad & \forall j \in [1, n_\text{b}] \\ \Delta e_{\min} \leq \Delta e_{\text{opt},j} \leq \Delta e_{\max} \qquad & j = 1 \\ \Delta e_{\min} \cdot n_{\text{b,len}}(j-1) \leq \Delta e_{\text{opt},j} \leq \Delta e_{\max} \cdot n_{\text{b,len}}(j-1) \qquad & \forall j \in [2, n_\text{b}] \end{aligned}$$

must first be reformulated into standard QP form, subproblem. The cost function (A.1) can be reformulated into standard matrix QP form as follows:

$$\begin{aligned} J = &\frac{1}{2} \left(q_{\text{meas}} - q_{\text{est}}\right)^\top \hat{\mathbf{R}} \left(q_{\text{meas}} - q_{\text{est}}\right) \\ &+ \frac{1}{2} \Delta e_{\text{opt}}^\top \hat{\mathbf{Q}}_\mathbf{e} \Delta e_{\text{opt}} \end{aligned} \tag{A.2}$$

where

$$\begin{aligned} q_{\text{meas}} &= \begin{bmatrix} q_{\text{meas},1,k-N} \cdots q_{\text{meas},n_\text{q},k-N} & q_{\text{meas},1,k-N+1} \cdots q_{\text{meas},n_\text{q},k} \end{bmatrix}^\top \\ q_{\text{est}} &= \begin{bmatrix} q_{\text{est},1,k-N} \cdots q_{\text{est},n_\text{q},k-N} & q_{\text{est},1,k-N+1} \cdots q_{\text{est},n_\text{q},k} \end{bmatrix}^\top \\ \Delta e_{\text{opt}} &= \begin{bmatrix} \Delta e_{\text{opt},1,1} \ldots \Delta e_{\text{opt},n_\text{e},1} & \Delta e_{\text{opt},1,2} \ldots \Delta e_{\text{opt},n_\text{e},n_\text{b}} \end{bmatrix}^\top \\ \hat{\mathbf{R}} &= diag\left(\mathbf{R}_{k-N}, \ldots, \mathbf{R}_N\right) \\ \hat{\mathbf{Q}}_\mathbf{e} &= diag\left(\mathbf{Q}_{\mathbf{e},1}, \ldots, \mathbf{Q}_{\mathbf{e},n_\text{b}}\right) \end{aligned}$$

The first subscript of q_{meas} and q_{est} indicates the index of the measurement element, while the second subscript denotes the sample number within the horizon. The first subscript of Δe_{opt} indicates the index of the correction factor element and the second subscript represents the block number.

In each SQP iteration, q_{est} and e_{opt} can be updated as a linear prediction model:

$$q_{\text{est}} = \overline{q_{\text{est}}} + \Delta q_{\text{est}} = \overline{q_{\text{est}}} + \mathbf{q_{sens,unscale}} \cdot \Delta e_{\text{opt,unscale}} \tag{A.3}$$

$$e_{\text{opt}} = \overline{e_{\text{opt}}} + \Delta e_{\text{opt,unscale}} \tag{A.4}$$

where $\overline{q_{\text{est}}}$ defines a vector of trajectories at the previous iteration and $\Delta e_{\text{opt,unscale}}$ is the solution in QP subproblem. Here, $\mathbf{q_{sens,unscale}}$ represents the unscaled sensitivity

matrix used to define the direction of the update. It contains the coefficients that describe how the estimated measurement responds to changes in e_{opt}. The unscaled sensitivity matrix, $\mathbf{q_{sens,unscale}}$, is recalculated at every estimation step and used during each SQP iteration. It is computed as follows:

$$\mathbf{q_{sens,unscale}} = \frac{q_{\text{est,perturb}} - q_{\text{est}}}{e_{\text{perturb}}}$$

where $q_{\text{est,perturb}}$ is a vector of perturbed estimated measurement which is calculated as:

$$q_{\text{est,perturb}} = g\left(x, e_{\text{opt}} + e_{\text{perturb}}, u\right), \quad q_{\text{est,perturb}} \in \mathbb{R}^{n_{\text{q}} \cdot N}$$

Here e_{perturb} is a vector of predefined constant used to determine the magnitude of the perturbation and it has dimension $n_{\text{e}} \cdot n_{\text{b}}$.

To ensure that all parameters have the same order of magnitude regardless of units, $q_{\text{est}}, \Delta q_{\text{est}}, \Delta e_{\text{opt,unscale}}$ and q_{meas} need to be scaled before solving the QP problem. The scaled Δq_{est} and $\Delta e_{\text{opt,unscale}}$ are defined as:

$$\Delta q_{\text{est,scale}} = \mathbf{q_{sens,scale}} \cdot \Delta e_{\text{opt,scale}} \tag{A.5}$$

$$\Delta e_{\text{opt,scale}} = \frac{\Delta e_{\text{opt,unscale}}}{\mathbf{e_{ss}}} \tag{A.6}$$

$$q_{\mathbf{sens,scale}} = \underbrace{\frac{q_{\text{est,perturb}} - q_{\text{est}}}{e_{\text{perturb}}}}_{\mathbf{q_{sens,unscale}}} \cdot \frac{\mathbf{e_{ss}}}{\mathbf{q_{ss}}} \tag{A.7}$$

i.e they are scaled by diagonal quadratic matrix $\mathbf{e_{ss}}$ and $\mathbf{q_{ss}}$ which represent the matrices of e_{opt} and q_{est} corresponding to that sample instant before SQP iteration starts.

Now apply (A.5)-(A.7) into (A.3) and (A.4). q_{est} and e_{opt} can then be expressed as:

$$q_{\text{est}} = \overline{q_{\text{est}}} + \mathbf{q_{ss}} \cdot \underbrace{\mathbf{q_{sens,scale}} \cdot \Delta e_{\text{opt,scale}}}_{\Delta q_{\text{est,scale}}}$$

$$e_{\text{opt}} = \overline{e_{\text{opt}}} + \mathbf{e_{ss}} \cdot \Delta e_{\text{opt,scale}}$$

Note that to preserve the original form of the equations, $\mathbf{q_{ss}}$ and $\mathbf{e_{ss}}$ are also added as multiplicative factors.

The solution at the current QP iteration of the SQP procedure is a function of the solution at the previous iteration plus the solution from the QP solver, i.e. e_{opt} and Δe_{opt} at the current iteration is expressed as:

$$\Delta e_{\text{opt}} = \overline{\Delta e_{\text{opt}}} + \Delta e_{\text{opt,unscale}} = \overline{\Delta e_{\text{opt}}} + \mathbf{e_{ss}} \cdot \Delta e_{\text{opt,scale}}$$

$q_{\text{est}}, \Delta e_{\text{opt}}$ and q_{meas} can then be scaled as:

$$q_{\text{est,s}} = \frac{q_{\text{est}}}{\mathbf{q_{ss}}} = \mathbf{q_{ss}}^{-1} \cdot \overline{q_{\text{est}}} + \mathbf{q_{sens,scale}} \cdot \Delta e_{\text{opt,scale}} \tag{A.8}$$

$$q_{meas,s} = \frac{q_{\text{meas}}}{\mathbf{q_{ss}}} = \mathbf{q_{ss}}^{-1} \cdot q_{\text{meas}} \tag{A.9}$$

$$\Delta e_{\text{opt,s}} = \frac{\Delta e_{\text{opt}}}{\mathbf{e_{ss}}} = \mathbf{e_{ss}}^{-1} \cdot \overline{\Delta e_{\text{opt}}} + \Delta e_{\text{opt,scale}} \tag{A.10}$$

$$e_{\text{opt,s}} = \frac{e_{\text{opt}}}{\mathbf{e_{ss}}} = \mathbf{e_{ss}}^{-1} \cdot \overline{e_{\text{opt}}} + \Delta e_{\text{opt,scale}} \tag{A.11}$$

Note that $\Delta e_{\text{opt,s}}$ and $\Delta e_{\text{opt,scale}}$ are not the same. $\Delta e_{\text{opt,scale}}$ is a scaled solution from QP while $\Delta e_{\text{opt,s}}$ is a scaled update in each SQP iteration. To prevent division by zero, as some elements in $\mathbf{q_{ss}}$ and $\mathbf{e_{ss}}$ are very close to zero, the minimum values for them are defined.

The scaled version of the cost function (A.2) can then be expressed as:

$$\begin{aligned} J = &\frac{1}{2} \left(q_{\text{meas,s}} - q_{\text{est,s}}\right)^{\top} \hat{\mathbf{R}} \left(q_{\text{meas,s}} - q_{\text{est,s}}\right) \\ &+ \frac{1}{2} \Delta e_{\text{opt,s}}^{\top} \hat{\mathbf{Q}}_{\mathbf{e}} \Delta e_{\text{opt,s}} \end{aligned}$$

Now insert (A.8)-(A.10) into the cost function above

$$\begin{aligned} J = &\frac{1}{2} \left(\mathbf{q_{ss}}^{-1} \cdot (q_{\text{meas}} - \overline{q_{\text{est}}}) - \mathbf{q_{sens,scale}} \cdot \Delta e_{\text{opt,scale}}\right)^{\top} \hat{\mathbf{R}} \left(\dots\right) \\ &+ \frac{1}{2} \left(\mathbf{e_{ss}}^{-1} \cdot \overline{\Delta e_{\text{opt}}} + \Delta e_{\text{opt,scale}}\right)^{\top} \hat{\mathbf{Q}}_{\mathbf{e}} \left(\dots\right) \\ &= \frac{1}{2} \left(q_{\text{meas}} - \overline{q_{\text{est}}}\right)^{\top} \cdot \mathbf{q_{ss}}^{-1} \cdot \hat{\mathbf{R}} \cdot \mathbf{q_{ss}}^{-1} \cdot (q_{\text{meas}} - \overline{q_{\text{est}}}) \\ &- \Delta e_{\text{opt,scale}}^{\top} \cdot \mathbf{q_{sens,scale}}^{\top} \cdot \hat{\mathbf{R}} \cdot \mathbf{q_{ss}}^{-1} \cdot (q_{\text{meas}} - \overline{q_{\text{est}}}) \\ &+ \frac{1}{2} \Delta e_{\text{opt,scale}}^{\top} \cdot \mathbf{q_{sens,scale}}^{\top} \cdot \hat{\mathbf{R}} \cdot \mathbf{q_{sens,scale}} \cdot \Delta e_{\text{opt,scale}} \\ &+ \frac{1}{2} \overline{\Delta e_{\text{opt}}}^{\top} \cdot \mathbf{e_{ss}}^{-1} \cdot \hat{\mathbf{Q}}_{\mathbf{e}} \cdot \mathbf{e_{ss}}^{-1} \cdot \overline{\Delta e_{\text{opt}}} \\ &+ \Delta e_{\text{opt,scale}}^{\top} \cdot \hat{\mathbf{Q}}_{\mathbf{e}} \cdot \mathbf{e_{ss}}^{-1} \cdot \overline{\Delta e_{\text{opt}}} \\ &+ \frac{1}{2} \Delta e_{\text{opt,scale}}^{\top} \cdot \hat{\mathbf{Q}}_{\mathbf{e}} \cdot \Delta e_{\text{opt,scale}} \end{aligned}$$

which can be shorten as:

$$J = \frac{1}{2} \Delta e_{\text{opt,scale}}^{\top} \cdot \mathbf{H} \cdot \Delta e_{\text{opt,scale}} + c^{\top} \cdot \Delta e_{\text{opt,scale}}$$

where

$$\begin{aligned} \mathbf{H} &= \mathbf{q_{sens,scale}}^{\top} \cdot \hat{\mathbf{R}} \cdot \mathbf{q_{sens,scale}} + \hat{\mathbf{Q}}_{\mathbf{e}} \\ c^{\top} &= \overline{\Delta e_{\text{opt}}}^{\top} \cdot \left(\mathbf{e_{ss}}^{-1}\right)^{\top} \cdot \hat{\mathbf{Q}}_{\mathbf{e}}{}^{\top} - \left(q_{\text{meas}} - \overline{q_{\text{est}}}\right)^{\top} \cdot \left(\mathbf{q_{ss}}^{-1}\right)^{\top} \cdot \hat{\mathbf{R}}^{\top} \cdot \mathbf{q_{sens,scale}} \end{aligned}$$

and their dimensions are

$$\mathbf{q_{sens,scale}} \in \mathbb{R}^{n_\mathrm{q}\cdot N\times n_\mathrm{e}\cdot n_\mathrm{b}},\ \hat{\mathbf{R}} \in \mathbb{R}^{n_\mathrm{q}\cdot N\times n_\mathrm{q}\cdot N},\ \hat{\mathbf{Q}}_\mathbf{e} \in \mathbb{R}^{n_\mathrm{e}\cdot n_\mathrm{b}\times n_\mathrm{e}\cdot n_\mathrm{b}}$$
$$\overline{\Delta e_\mathrm{opt}} \in \mathbb{R}^{n_\mathrm{e}\cdot n_\mathrm{b}},\ \mathbf{e_{ss}} \in \mathbb{R}^{n_\mathrm{e}\cdot n_\mathrm{b}\times n_\mathrm{e}\cdot n_\mathrm{b}},\ \mathbf{q_{ss}} \in \mathbb{R}^{n_\mathrm{q}\cdot N\times n_\mathrm{q}\cdot N}$$
$$q_\mathrm{meas} \in \mathbb{R}^{n_\mathrm{q}\cdot N},\ \overline{q_\mathrm{est}} \in \mathbb{R}^{n_\mathrm{q}\cdot N},\ \Delta e_\mathrm{opt,scale} \in \mathbb{R}^{n_\mathrm{e}\cdot n_\mathrm{b}}$$

There is also a constant term

$$K = \frac{1}{2}\left(q_\mathrm{meas} - \overline{q_\mathrm{est}}\right)^\top \cdot \mathbf{q_{ss}}^{-1} \cdot \hat{\mathbf{R}} \cdot \mathbf{q_{ss}}^{-1} \cdot \left(q_\mathrm{meas} - \overline{q_\mathrm{est}}\right) + \frac{1}{2}\overline{\Delta e_\mathrm{opt}}^\top \cdot \mathbf{e_{ss}}^{-1} \cdot \hat{\mathbf{Q}}_\mathbf{e} \cdot \mathbf{e_{ss}}^{-1} \cdot \overline{\Delta e_\mathrm{opt}}$$

However, this term is not significant and can therefore be neglected.

Since the optimization parameters are scaled, the constraints must also be scaled accordingly. For the scaled correction factor $e_\mathrm{opt,s}$ (using (A.11)):

$$\begin{aligned} & \mathbf{e_{ss}}^{-1} \cdot e_\mathrm{min} < e_\mathrm{opt,s} < \mathbf{e_{ss}}^{-1} \cdot e_\mathrm{max} \\ \Leftrightarrow \quad & \mathbf{e_{ss}}^{-1} \cdot e_\mathrm{min} < \mathbf{e_{ss}}^{-1} \cdot \overline{e_\mathrm{opt}} + \Delta e_\mathrm{opt,scale} < \mathbf{e_{ss}}^{-1} \cdot e_\mathrm{max} \\ \Leftrightarrow \quad & \mathbf{e_{ss}}^{-1} \cdot \left(e_\mathrm{min} - \overline{e_\mathrm{opt}}\right) < \Delta e_\mathrm{opt,scale} < \mathbf{e_{ss}}^{-1} \cdot \left(e_\mathrm{max} - \overline{e_\mathrm{opt}}\right) \end{aligned} \tag{A.12}$$

For the scaled deviation of correction factor $\Delta e_\mathrm{opt,s}$ (using (A.10)):

$$\begin{aligned} & \mathbf{e_{ss}}^{-1} \cdot \Delta e_\mathrm{min} \cdot n_\mathrm{b,len}(j-1) < \Delta e_\mathrm{opt,s} < \mathbf{e_{ss}}^{-1} \cdot e_\mathrm{max} \cdot n_\mathrm{b,len}(j-1) \\ \Leftrightarrow \quad & \mathbf{e_{ss}}^{-1} \cdot \Delta e_\mathrm{min} \cdot n_\mathrm{b,len}(j-1) < \mathbf{e_{ss}}^{-1} \cdot \overline{\Delta e_\mathrm{opt}} + \Delta e_\mathrm{opt,scale} < \mathbf{e_{ss}}^{-1} \cdot e_\mathrm{max} \cdot n_\mathrm{b,len}(j-1) \\ \Leftrightarrow \quad & \mathbf{e_{ss}}^{-1} \left(\Delta e_\mathrm{min} \cdot n_\mathrm{b,len}(j-1) - \overline{\Delta e_\mathrm{opt}}\right) \\ & < \Delta e_\mathrm{opt,scale} < \\ & \mathbf{e_{ss}}^{-1} \left(\Delta e_\mathrm{max} \cdot n_\mathrm{b,len}(j-1) - \overline{\Delta e_\mathrm{opt}}\right) \end{aligned} \tag{A.13}$$

Thus, there are two lower bounds and two upper bounds that need to be satisfied, (A.12) and (A.13).

The optimization problem that will be solved in QP iteration can then be formulated as:

$$\underset{\Delta e_\mathrm{opt,scale}}{\text{minimize}} \quad J = \frac{1}{2}\Delta e_\mathrm{opt,scale}^\top \cdot \mathbf{H} \cdot \Delta e_\mathrm{opt,scale} + c^\top \cdot \Delta e_\mathrm{opt,scale}$$

subject to

$$\begin{aligned} & \mathbf{max}\left(\mathbf{e_{ss}}^{-1} \cdot \left(e_\mathrm{min} - \overline{e_\mathrm{opt}}\right),\ \mathbf{e_{ss}}^{-1} \left(\Delta e_\mathrm{min} \cdot n_\mathrm{b,len}(j-1) - \overline{\Delta e_\mathrm{opt}}\right)\right) \\ & < \Delta e_\mathrm{opt,scale} < \\ & \mathbf{min}\left(\mathbf{e_{ss}}^{-1} \cdot \left(e_\mathrm{max} - \overline{e_\mathrm{opt}}\right),\ \mathbf{e_{ss}}^{-1} \left(\Delta e_\mathrm{max} \cdot n_\mathrm{b,len}(j-1) - \overline{\Delta e_\mathrm{opt}}\right)\right) \\ & \forall j \in [1, n_\mathrm{b}],\ \text{for } j = 1\,(\Delta e_\mathrm{opt,scale,1}) \longrightarrow n_\mathrm{b,len}(j-1) = 1 \end{aligned}$$

Since the solution $\Delta e_\mathrm{opt,scale}$ is scaled, the unscaled update is defined as follows:

$$\begin{aligned} q_\mathrm{est} &= \overline{q_\mathrm{est}} + \mathbf{q_{ss}} \cdot \mathbf{q_{sens,scale}} \cdot \Delta e_\mathrm{opt,scale} \\ \Delta e_\mathrm{opt} &= \overline{\Delta e_\mathrm{opt}} + \mathbf{e_{ss}} \cdot \Delta e_\mathrm{opt,scale} \\ e_\mathrm{opt} &= \overline{e_\mathrm{opt}} + \mathbf{e_{ss}} \cdot \Delta e_\mathrm{opt,scale} \end{aligned}$$

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY