SLOVAK UNIVERSITY OF TECHNOLOGY
IN BRATISLAVA

FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-5414-86663

Software Sensors for Industry

MASTER THESIS

2021

Bc. Darko Križan

SLOVAK UNIVERSITY OF TECHNOLOGY
IN BRATISLAVA

FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-5414-86663

Software Sensors for Industry

MASTER THESIS

Study programme:	Automation and Information Engineering in Chemistry and Food Industry
Study field:	Cybernetics
Training workspace:	Oddelenie informatizácie a riadenia procesov
Thesis supervisor:	doc. Ing. Radoslav Paulen, PhD.
Consultant:	Ing. Martin Mojto

2021 **Bc. Darko Križan**

Slovak University of Technology in Bratislava
Department of Information Engineering and Process Control

Faculty of Chemical and Food Technology
Academic year: 2020/2021
Reg. No.: FCHPT-5414-86663

MASTER THESIS TOPIC

Student: **Bc. Darko Križan**

Student's ID: 86663

Study programme: Automation and Information Engineering in Chemistry and Food Industry

Study field: Cybernetics

Thesis supervisor: doc. Ing. Radoslav Paulen, PhD.

Consultant: Ing. Martin Mojto

Workplace: Oddelenie informatizácie a riadenia procesov

Topic: **Software Sensors for Industry**

Language of thesis: English

Specification of Assignment:

The main goal of the project is the design of (inference) software sensors, which represent a cheap alternative for sensing problematic, difficult to measure quantities in industrial processes (e.g. product composition or equipment degradation). The design of inference sensors will be realized on selected examples. When designing inference sensors, the accuracy of several approaches will be compared using statistical methods and the most suitable software sensor for the given application will be determined.

Tasks:
- Design of structures of mathematical models of sensors
- Calculation of sensor parameters
- Evaluation of the quality of the designed software sensors

Selected bibliography:

1. Petr Kadlec, Bogdan Gabrys, Sibylle Strandt, Data-driven Soft Sensors in the process industry, Computers & Chemical Engineering 33(4), 2009, pp. 795-814, ISSN 0098-1354, https://doi.org/10.1016/j.compchemeng.2008.12.012.

Assignment procedure from: 15. 02. 2021

Date of thesis submission: 16. 05. 2021

Bc. Darko Križan
Student

Ing. Martin Klaučo, PhD.
Head of department

prof. Ing. Miroslav Fikar, DrSc.
Study programme supervisor

Acknowledgment

I would like to express my great gratitude to my supervisor doc. Ing. Radoslav Paulen, PhD. for all provided knowledge, professional advice, comments and effort invested during the master thesis. I would also like to thank my consultant Ing. Martin Mojto for the willingness and patience during the elaboration of the work.

Abstract

This master thesis is focused on the design of software sensors for industry. At the heart of every well-operated process is the accurate measurement of system variables. Instruments that measure these variables are very often difficult to implement, unreliable, or expensive. The aim is to propose an alternative solution for obtaining soft-measurements of system variables. The work is focused on the design of software sensors based on data. Software sensors are designed for the process of two tanks in series without interaction. Methods which are used for software sensor design are the ordinary least squares (OLS), ridge regression (RR), least absolute shrinkage and selection operator (LASSO) and elastic net (EN). The work describes the procedure of obtaining a difficult-to-measure variable (the liquid level height of the first tank) using software sensors. The designed software sensors are statistically evaluated. The final highlight of this work is software sensors implementation in the model of two tanks in series and comparison with Kalman filter as the best observer of difficult-to-measure variables.

Abstrakt

Táto diplomová práca je zameraná na návrh softvérových senzorov pre priemysel. Jadrom každého dobre fungujúceho procesu je presné meranie systémových premenných. Prístroje, ktoré tieto premenné merajú sú veľmi často ťažko implementovateľné, nespoľahlivé, alebo vyžadujú vysoké náklady. Cieľom je vyvinúť alternatívne riešenie získavania systémových premenných. Práca je zameraná na návrh softvérových senzorov na základe dát. Softvérové senzory sú navrhované pre procese dvoch zásobníkov kvapaliny bez interakcie. Na návrh softvérových senzorov sú použité metódy: metóda najmenších štvorcov, hrebeňová regresia, operátor najmenšieho absolútneho zmrštenia a výberu a elastická sieť. Práca opisuje postup získania ťažko merateľnej veličiny, ktorú reprezentuje výška hladiny kvapaliny prvého zásobníka, pomocou softvérových senzorov. Navrhnuté softvérové senzory sú následne štatisticky vyhodnocované. Vrcholom práce je implementácia softvérových senzorov do modelu dvoch zásobníkov kvapaliny a ich porovnanie s Kalmanovým filtrom ako najlepším pozorovateľom ťažko merateľných premenných.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

At present, almost all plants in the industry are equipped with a large number of sensors. The primary role of sensors is to provide data that ensures process monitoring and control [6]. At the end of the last century, researchers starts to work with a large amount of data. Data starts to be measured and stored in large quantities in the process industry. Based on these data, we can predict the behaviour of the process in the future and thus design predictive models [9]. Such predictive models in industry are called *software sensors* or *soft-sensors* (SSs). "Soft" because the models are software programs and "sensors" because they have the role of real physical sensors. Another term used in the process industry is *inferential sensors* etc. [6].

In general, there are two classes of SSs, namely model-based SSs and data-driven SSs. The model-based SSs family is most often based on first-principles models, which represents the physical and chemical background of the process. Such models are proposed for the planning and design of the processing plants. They are most often focused on the description of ideal steady states, which is one example of the disadvantage that makes it difficult to base SSs on them. There are also SSs based on an extended Kalman filter (KF), based on adaptive observer etc. [6]. Data-driven SSs have gained much bigger popularity in the process industry. These SSs are based on real industry data and thus describe the actual process conditions. Compared to model-based SSs, they are closer to reality and more accurately describe the actual process conditions [6]. On the other hand, they require more frequent revisions than model-based SSs to describe the current state of the process.

The range of tasks that SSs perform is wide. The primary and most important task of SSs is the prediction of process variables. These variables are very often related to the quality of the process output. Therefore, they are important for the manipulation and process control [6]. In the process industry it often happens that we have information about several easily measurable process variables. In reality, not all have affect on difficult-to-measure (DTM). Let us have a situation that the output variable ω is

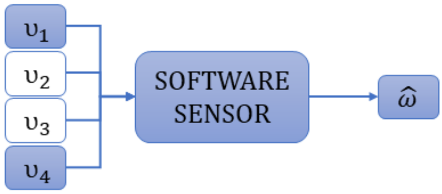

Figure 1.1: Structural estimation of the researched process.

affected only by the easy-to-measure (ETM) variables υ_1 and υ_4 from the available set of variables (υ_1, υ_2, υ_3, υ_4). Fig. 1.1 shows an ideal software sensor (SS). This SS in his structure correctly considers the influence of only mentioned variables υ_1 and υ_4.

Another application area of SSs in industry is the detection of malfunctioning sensors. As already mentioned, various plants in the industry are equipped with a large number of sensors. It is very likely that any of these sensors may fail from time to time. As soon as a faulty sensor is detected, it is reconstructed or replaced (by another sensor or by SS). The great advantage of SSs is that they are not subject to mechanical failures. They are easy to maintain and therefore much more cost-effective in this respect. Other important application areas of SSs are process monitoring and error detection, prognosis for preventing undesirable operation, optimisation of the operating conditions to ensure better productivity of the industrial plants, more energy-saving, less negative environmental impact etc. [9].

Despite all of the above applications and the benefits of SSs, there are still unresolved issues regarding the development and maintenance of SSs. Many key problems are caused by the process data on which the SSs are designed. Common problems in measuring data are measurement noise, extreme deviations of some values, missing values, different sampling rates, etc. Another problem is that industry is a dynamic environment where sudden process changes may occur. Some SSs find it difficult to work with daily changes in the process like this. This can usually lead to a deterioration in the accuracy of prediction [6].

The work consists of six chapters. The first chapter is introductory. This section explains the general features, reasons, and benefits of using SSs. Examples of the SSs use in the process industry, their tasks and shortcomings are also given in this section. The second chapter is the theoretical foundation part. This part of work describes various structures of SSs. It also involves methods for SS design and their effect on a simple example $\omega = 3\upsilon_1$. Third chapter describes case study of two tanks in series without interaction where the SSs will be proposed. This section also describes ways of liquid level height (LLH) estimation in tanks. The fourth chapter gives the most significant results of the case study. It presents the results of the SSs estimation of the LLH in the first tank. This chapter also contains the results of comparison KF and SSs estimation of LLH in the first tank as the greatest achievement of the work. The fifth chapter contains discussion of the obtained results. This chapter also describes possibilities for improvement. The last, sixth chapter concludes this thesis. This part presents a summary of the work, comments of the most significant results obtained through the whole work and and the possibility of developing work in the future.

Chapter 2

Theoretical Foundations

2.1 Introduction to Software Sensors

With the development of computer and automation technology a new era of monitoring and control of industrial plants begins. However, there are still variables which for some reason we cannot directly indicate [9]. These difficult-to-measure (DTM) variables can be estimated in the form of dependence on easy-to-measure (ETM) variables. This is the fundamental ideology of software sensor (SS) design. There are several ways to design software sensors (SSs). According to the structure, we distinguish data-driven SSs, model-based SSs and a hybrid of these two types.

2.1.1 Data-Driven Software Sensors

In general, the structure of SSs designed on the basis of real data can be interpreted as multiple linear regression. This means that the required variable is linearly dependent on several parameters of the process [15]. Generally, the structure of SS based on data can be written as follows

$$\hat{\omega} = \beta_0 + \beta_1 \upsilon_1 + \cdots + \beta_n \upsilon_n + \varepsilon, \tag{2.1}$$

or in vector form

$$\hat{\omega} = \boldsymbol{\beta}^T \boldsymbol{\upsilon} + \boldsymbol{\varepsilon}, \tag{2.2}$$

where vector $\boldsymbol{\upsilon}$ represents all ETM variables of the process, and $\boldsymbol{\varepsilon}$ represents random error of estimation. Vector $\boldsymbol{\beta}$ include the SS parameters [15]. We need to estimate this vector $\boldsymbol{\beta}$ to describe a required DTM variable ω, as it is the only unknown on the right side of the equation (2.2).

2.1.2 Model-Based Software Sensors

The performance of these SSs greatly depends on the reliability and accuracy of the identified process [2]. An example is the Kalman filter (KF) design based on the identified nominal state-space model. The main task of the KF is to estimate DTM variables. The main goal of this type of SSs is to find such a nominally optimal model

$$\hat{\omega} = M(\upsilon_1, \upsilon_2, \ldots, \upsilon_n), \tag{2.3}$$

where outputs of the SSs $\hat{\omega}$ are as close as possible to the recorded outputs of the observed process ω. This idea is defined by the following equation [2]:

$$J = \|\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}\|_2^2. \tag{2.4}$$

The objective function J from the equation (2.4) is expressed by ℓ_2 norm. It expresses the square of the difference between the vector of measured outputs of DTM variables $\boldsymbol{\omega}$ and the vector of predicted outputs of DTM variables $\hat{\boldsymbol{\omega}}$. This implies that the desired model gives predictions with minimum deviations from the process outputs.

2.1.3 Hybrid Structure of Software Sensors

These SSs use the knowledge of the model, but they are also designed on the basis of historically measured data.

$$\hat{\omega} = M(\upsilon_1, \upsilon_2, \ldots, \upsilon_n) + \sum_{i=1}^{n} \beta_i \upsilon_i. \tag{2.5}$$

The structure of these SSs is mathematically illustrated by the equation (2.5). We can take the case of a biochemical reactor as an example of using the model and data for SS design. In this case, the theoretical Michaelis-Menten model is used. As is already known, Michaelis-Menten model represents the relationship between the rate of the enzyme reaction and the substrate concentration. This model provides us information about substrate concentration [5]. The biochemical reactor also may contain physical equipment that monitors the substrate concentration. Using this knowledge we obtain information about the concentration of the substrate from two different sources, which can improve each other. With this, the quality of SS estimation might be dramatically increased.

2.2 Software Sensors Design

In this work, SSs will be designed by following methods: ordinary least squares (OLS), ridge regression (RR), least absolute shrinkage and selection operator (LASSO) and by elastic net (EN). To design SSs the following vectors are defined:

$$\boldsymbol{v_i} = \begin{bmatrix} v_{1,i} \\ v_{2,i} \\ \vdots \\ v_{n,i} \end{bmatrix}, \qquad \boldsymbol{\mathcal{V}} = \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \\ \vdots \\ \boldsymbol{v}_N^T \end{bmatrix}, \qquad \boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix}, \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}. \tag{2.6}$$

The vector $\boldsymbol{v_i}$ represents the values of the individual ETM variables measured at one time point ($v_{1,i}$, $v_{2,i}$, ..., $v_{n,i}$). The matrix $\boldsymbol{\mathcal{V}}$ contains all measuring points $\boldsymbol{v_i}$, where index N represents the number of measurements. All measured DTM variables are stored in the vector $\boldsymbol{\omega}$. The resulting vector of the SS design is the vector $\boldsymbol{\beta}$. This vector contains all SS parameters (β_0, β_1, ..., β_n).

2.2.1 Ordinary Least Squares

The OLS method is a standard method of regression analysis. It is used to find such regression parameters, which describe the regression line that is the most "closest" to all data points (v_i, ω_i) [15]. This problematic can be defined as optimisation formulation of minimising the sum of squares deviations

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} (\omega_i - \hat{\omega}_i)^2. \tag{2.7}$$

If we substitute $\hat{\omega}_i$ for expression from the equation (2.2), which represents the structure of SSs, we get the modified expression

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} (\omega_i - \boldsymbol{\beta}^T \boldsymbol{v}_i)^2. \tag{2.8}$$

or in the matrix form [15]

$$\min_{\boldsymbol{\beta}} \frac{1}{2} (\boldsymbol{\omega} - \boldsymbol{\mathcal{V}}\boldsymbol{\beta})^T (\boldsymbol{\omega} - \boldsymbol{\mathcal{V}}\boldsymbol{\beta}). \tag{2.9}$$

After appropriate modifications, the resulting objective function is in the form:

$$f(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta}} \frac{1}{2} \Big(\boldsymbol{\omega}^T \boldsymbol{\omega} - \boldsymbol{\omega}^T \boldsymbol{\mathcal{V}}\boldsymbol{\beta} - (\boldsymbol{\mathcal{V}}\boldsymbol{\beta})^T \boldsymbol{\omega} + (\boldsymbol{\mathcal{V}}\boldsymbol{\beta})^T \boldsymbol{\mathcal{V}}\boldsymbol{\beta} \Big), \tag{2.10}$$

$$f(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta}} \frac{1}{2} \Big(\boldsymbol{\omega}^T \boldsymbol{\omega} - 2\boldsymbol{\omega}^T \boldsymbol{\mathcal{V}}\boldsymbol{\beta} + \boldsymbol{\beta}^T \boldsymbol{\mathcal{V}}^T \boldsymbol{\mathcal{V}}\boldsymbol{\beta} \Big). \tag{2.11}$$

To solve this optimisation problem, the first-order derivative of the $f(\boldsymbol{\beta})$ must be equal to zero [15].

$$\frac{df(\boldsymbol{\beta})}{d\boldsymbol{\beta}} = \frac{1}{2} \Big(-2\boldsymbol{\mathcal{V}}^T \boldsymbol{\omega} + 2\boldsymbol{\mathcal{V}}^T \boldsymbol{\mathcal{V}}\boldsymbol{\beta} \Big) = 0, \tag{2.12}$$

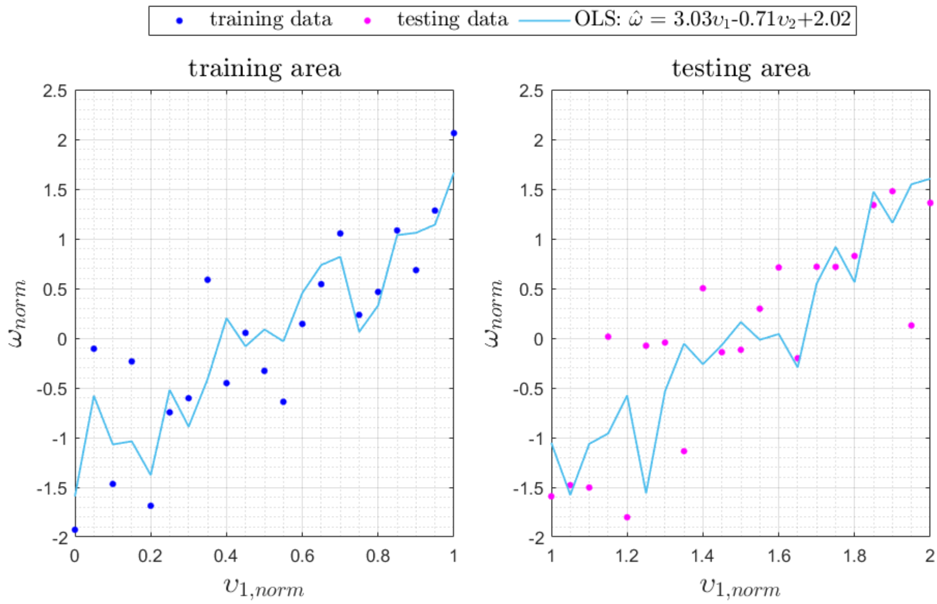

Figure 2.1: Comparison of OLS SS performance on training (left-hand plot) and testing (right-hand plot) data.

$$\boldsymbol{\mathcal{V}}^T\boldsymbol{\omega} = \boldsymbol{\mathcal{V}}^T\boldsymbol{\mathcal{V}}\boldsymbol{\beta}. \tag{2.13}$$

By multiplying both sides of the equation (2.13) with the matrix $(\boldsymbol{\mathcal{V}}^T\boldsymbol{\mathcal{V}})^{-1}$, we obtain the resulting vector of SS parameters

$$\boldsymbol{\beta} = (\boldsymbol{\mathcal{V}}^T\boldsymbol{\mathcal{V}})^{-1}\boldsymbol{\mathcal{V}}^T\boldsymbol{\omega}. \tag{2.14}$$

Example: Simple linear model The SS designed on the basis of the OLS method can be tested on a simple example of a linear model $\omega = 3\upsilon_1$. There are two ETM variables υ_1 and υ_2. Ideally, only υ_1 describes DTM variable ω. The design procedure is as follows: training and testing data are divided into two intervals. The training data is on interval [0, 1] with step 0.05 and the testing data is on interval [1, 2] with the same step 0.05. Thus, both intervals contains 20 data points each. In order to simulate the real situation, the data is corrupted by a white Gaussian noise. The data is also normalised for a better comparison of SSs in the training and testing area.

Fig. 2.1 shows a comparison of the results of SS based on OLS in training and testing area. The graphs show the dependence of the DTM variable ω_{norm} from the ETM

Table 2.1: Comparison of RMSE for OLS SS in training and testing area.

	training area	testing area
RMSE	0.4318	0.6935

variable $\upsilon_{1,norm}$. On the vertical axis is the estimated DTM variable ω_{norm} and on the horizontal axis is the ETM variable $\upsilon_{1,norm}$. The SS based on OLS tries to minimise the sum of squares deviations from training data. It uses all possible information (two ETM variables υ_1 and υ_2). In the testing area it tries to mimic its course from the training area. The accuracy of SS in training and testing area can be compared by Root Mean Square Error (RMSE) values. RMSE is often used to quantify the difference between a predicted or estimated model and an observed process [10].

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\omega_i - \hat{\omega}_i)^2}, \tag{2.15}$$

where ω_i is a real measured value of DTM variable, $\hat{\omega}_i$ represents the estimated value of a DTM variable [10] and N is a number of measurements.

The deviations of SS from the new testing data is larger than from training data. This is confirmed by Table 2.1, which shows the RMSE values for the training and testing area. The RMSE value in the testing area (0.6935) is much higher than in the training area (0.4318).

2.2.2 Ridge Regression

RR can be classified as a "shrinkage" method. This method has proven to be very useful for systems where a large number parameters is common. By penalising the magnitude of SS regression parameters, we ensure the stability of the estimate by reducing the SS parameters [15]. This is an extended form of the OLS method. The objective function of OLS method from equation (2.9) is extended by the penalty term that involves sum of squares of regression parameters (squared ℓ_2 norm: $\sum_{i=1}^{n}\beta_i^2 = \boldsymbol{\beta}^T\boldsymbol{\beta}$) [15]

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\Big[(\boldsymbol{\omega} - \boldsymbol{\mathcal{V}}\boldsymbol{\beta})^T(\boldsymbol{\omega} - \boldsymbol{\mathcal{V}}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}\Big]. \tag{2.16}$$

The parameter λ represents a weighting parameter. If we look at Figure 2.2, we can see that objective function of OLS is represented by contour lines and has a minimum at the point β_{OLS}. The penalty of RR moves the optimum closer to zero and placed

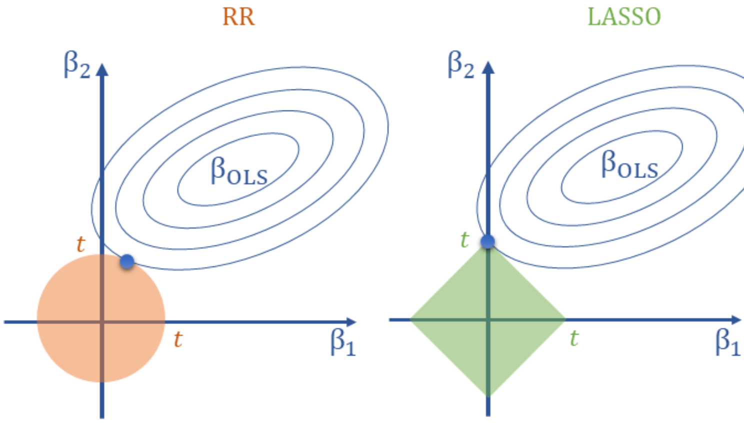

Figure 2.2: Optimal solution of RR and LASSO [7].

it on a circle ($\sum_{i=1}^{n} \beta_i^2 = t^2$) [7]. This graphically illustrates the minimisation of the OLS objective function by means of a squared ℓ_2 norm penalty.

Similar derivation procedure as with the OLS is used here

$$f(\boldsymbol{\beta}) = \frac{1}{2}\Big(\boldsymbol{\omega}^T\boldsymbol{\omega} - 2\boldsymbol{\omega}^T\boldsymbol{\mathcal{V}}\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{\mathcal{V}}^T\boldsymbol{\mathcal{V}}\boldsymbol{\beta} + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}\Big), \tag{2.17}$$

$$f(\boldsymbol{\beta}) = \frac{1}{2}\Big(\boldsymbol{\omega}^T\boldsymbol{\omega} - 2\boldsymbol{\omega}^T\boldsymbol{\mathcal{V}}\boldsymbol{\beta} + \boldsymbol{\beta}^T(\boldsymbol{\mathcal{V}}^T\boldsymbol{\mathcal{V}} + \lambda I)\boldsymbol{\beta}\Big). \tag{2.18}$$

The derivative of the function $f(\boldsymbol{\beta})$ must be equal to zero

$$\frac{df(\boldsymbol{\beta})}{d\boldsymbol{\beta}} = -\boldsymbol{\mathcal{V}}^T\boldsymbol{\omega} + (\boldsymbol{\mathcal{V}}^T\boldsymbol{\mathcal{V}} + \lambda\boldsymbol{I})\boldsymbol{\beta} = 0. \tag{2.19}$$

Resulting vector of regression parameters [15]:

$$\boldsymbol{\beta} = (\boldsymbol{\mathcal{V}}^T\boldsymbol{\mathcal{V}} + \lambda I)^{-1}\boldsymbol{\mathcal{V}}^T\boldsymbol{\omega}. \tag{2.20}$$

Example: Simple linear model (Continued) Similar as with SS based on OLS method, a SS based on RR can also be tested on a simple example of linear dependence $\omega = 3\upsilon_1$. The same data will be used as for the OLS SS (training area, testing area, noisy and normalised data). By solving the optimisation problem (2.16), where the weighting parameter takes the value $\lambda = 6$, the result of regression is shown in Fig. 2.3.

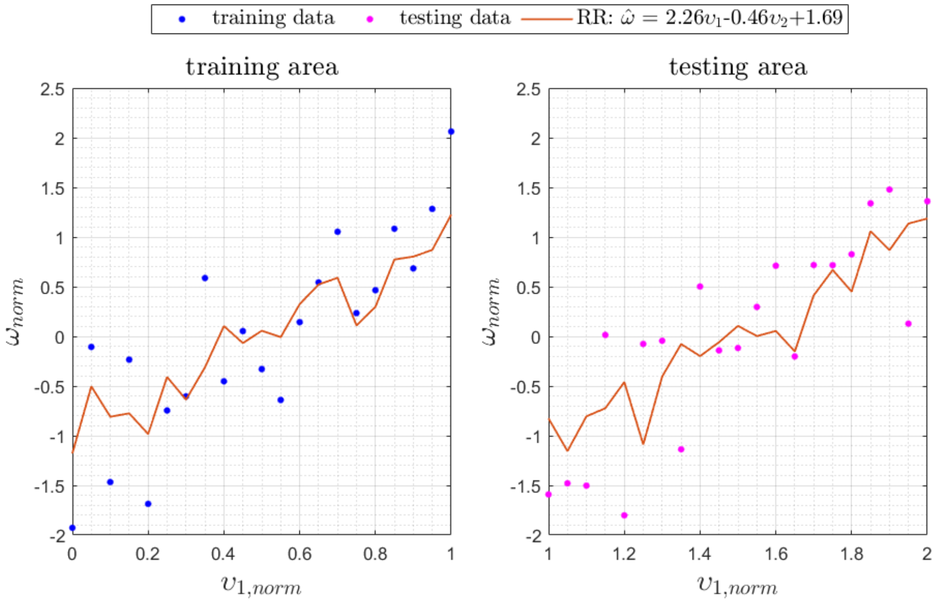

Figure 2.3: Comparison of RR SS performance on training and testing data.

Table 2.2: Comparison of RMSE for RR SS in training and testing area.

	training area	testing area
RMSE	0.4881	0.6397

The selection of the parameter λ plays a very important role in this method. With increasing value of λ, the parameters of the SS are more reduced [15]. The selected value for the training data is $\lambda = 6$. Compared to the OLS SS, the SS based on RR acquire smaller values of SS parameters. The result is a slightly smoother regression line.

Like the SS designed by OLS method, this SS also has large deviations from the testing data. This is shown by Table 2.2, where RMSE value for testing data (0.6397) has a larger value than RMSE for training data (0.4881).

Table 2.3: Comparison of RMSE for LASSO SS in training and testing area.

	training area	testing area
RMSE	0.5966	0.5972

2.2.3 Least Absolute Shrinkage and Selection Operator

This method intends to simplify the structure of SSs, therefore it belongs into the "shrinkage" methods as well. It is a regression analytical method that it naturally performs variable reduction to increase the accuracy of the estimate [15]. The main advantage compared to RR is that directly eliminates some regression parameters. Thus, in addition to estimating the parameters, it also makes an estimate of the model structure [7].

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\Big[(\boldsymbol{\omega} - \boldsymbol{\mathcal{V}}\boldsymbol{\beta})^T(\boldsymbol{\omega} - \boldsymbol{\mathcal{V}}\boldsymbol{\beta}) + \lambda||\boldsymbol{\beta}||_1\Big]. \tag{2.21}$$

As with the RR method, we extend the OLS method (2.9) by a penalty term. In this case, it is the sum of absolute values of the regression parameters (ℓ_1 norm: $\sum_{i=1}^{n}|\beta_i| = ||\boldsymbol{\beta}||_1$). As for RR, the parameter λ represents a weighting parameter intended to adjust the penalty. In Figure 2.2 we can see the difference between the RR and the LASSO. In the case of the LASSO method, the penalty ($\sum_{i=1}^{m}|a_i| = t$) shifts the optimum to a square. The optimum in this case lies on the vertical axis where the value of β_2 is equal to zero. The solution of the optimisation problem (2.21) can be approximated by an RR method

$$\boldsymbol{\beta} = (\boldsymbol{\mathcal{V}}^T\boldsymbol{\mathcal{V}} + \lambda\boldsymbol{W})^{-1}\boldsymbol{\mathcal{V}}^T\boldsymbol{\omega}, \tag{2.22}$$

where $\boldsymbol{W}$ is a diagonal matrix with elements $|\beta_i|$ on diagonal [15].

Example: Simple linear model (Continued) As in two previous cases, SS based on the LASSO can be tested on a simple linear example $\omega = 3\upsilon_1$. The same data with all properties (distribution, white noise, normalisation) is used as well.

By solving the optimisation problem (2.21), the resulting regression of the LASSO method is shown in Fig. 2.4. Weighting parameter λ also plays an important role in this method. By a suitable choice of the λ, we can eliminate some remaining parameters of the SS ($\lambda = 0.26$ in this case). The results show that LASSO is able to eliminate the ETM variable υ_2 and we obtain a description of the process, which structurally corresponds to the ideal case $\omega = 3\upsilon_1$.

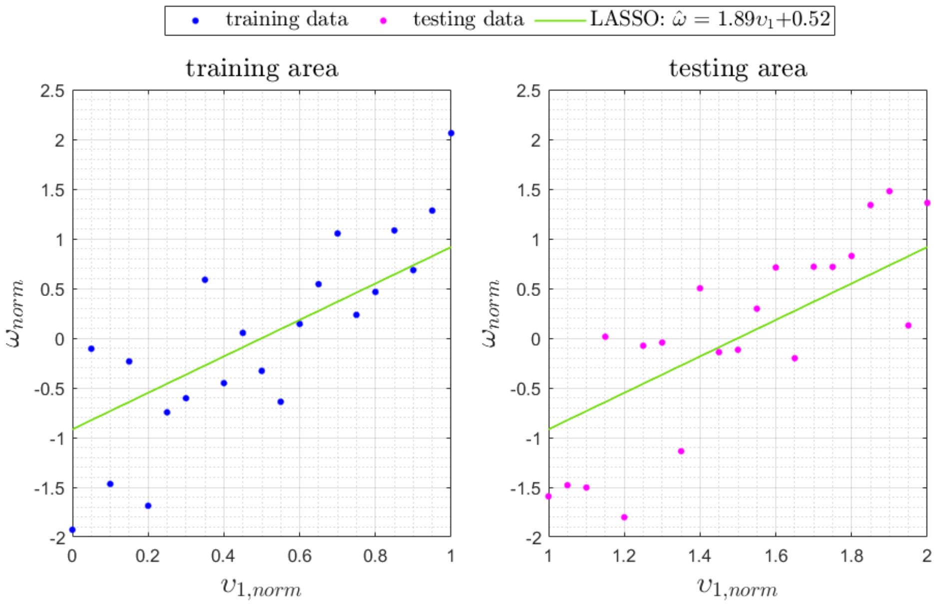

Figure 2.4: Comparison of LASSO SS performance on training and testing data.

In Table 2.4 we can see that RMSE values for the training and testing area differ only on the fourth decimal place. It follows that this SS is similarly accurate for newly measured testing data as for historical training data.

2.2.4 Elastic Net

Similar to the LASSO, the EN simultaneously does automatic variable selection and continuous shrinkage. It can select groups of correlated variables. It is a compromise between the RR and LASSO [16]. This method combines penalties, using the ℓ_1 norm (LASSO) and the squared ℓ_2 norm (RR)

$$\min_{\boldsymbol{\beta}} \left[\frac{1}{2}(\boldsymbol{\omega} - \boldsymbol{\mathcal{V}}\boldsymbol{\beta})^T(\boldsymbol{\omega} - \boldsymbol{\mathcal{V}}\boldsymbol{\beta}) + \lambda \left(\frac{1-\alpha}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + \alpha||\boldsymbol{\beta}||_1 \right) \right]. \tag{2.23}$$

The parameter α represents the weighting parameter which takes the values from interval [0, 1]. If α has value near to zero, the EN acquires the properties of the RR method. On the other hand, if the value of α approaches one, the EN behaves as a LASSO method. The weighting parameter λ must be non-negative as well [16].

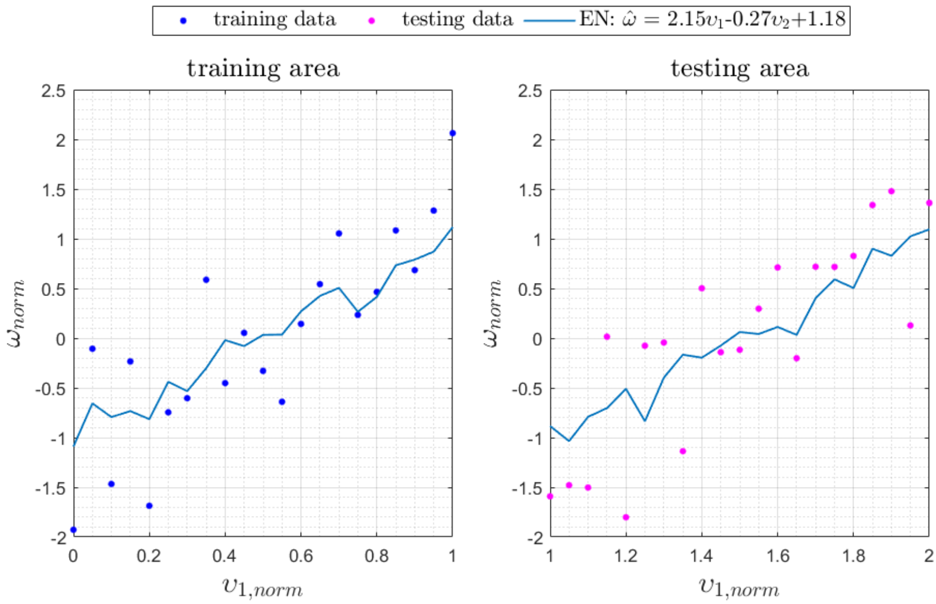

Figure 2.5: Comparison of EN SS performance on training and testing data.

Table 2.4: Comparison of RMSE for EN SS in training and testing area.

	training area	testing area
RMSE	0.5200	0.6071

Example: Simple linear model (Continued) As for all methods up to now, the SSs can also be designed based on the EN method. This method is also tested on a simple linear example $\omega = 3\upsilon_1$. The same noisy set of normalised data is divided into training and testing area.

Since the EN is a compromise between RR and LASSO methods, we expect the resulting performance to be in-between the performance of RR and LASSO SSs. Fig. 2.5 shows resulting regression of SS based on EN. There are two weighting parameters in this method. To make the compromise between RR and LASSO method visible, the parameter α takes value 0.5. Parameter λ takes the same value as in the case of pure LASSO ($\lambda = 0.26$).

The comparison of methods for SS design is summarised in Fig. 2.6. In this graph we can deduce the behaviour of SS, designed on a simple example of linear dependence

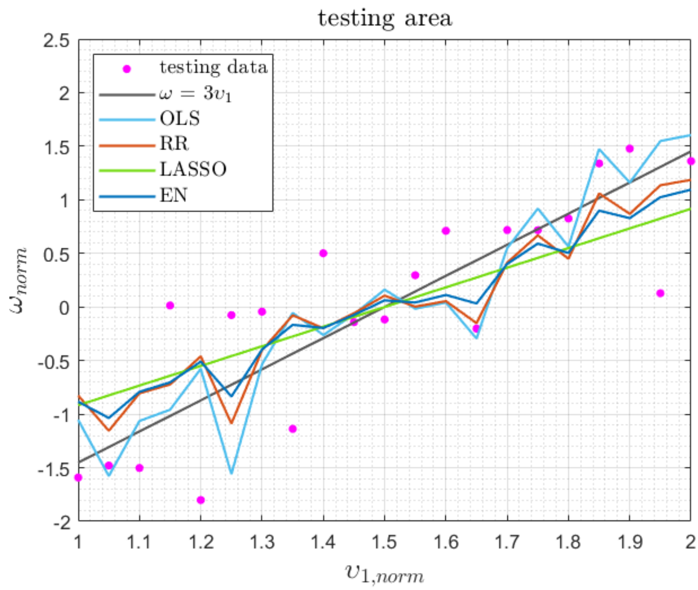

Figure 2.6: Comparison of SSs on testing data.

Table 2.5: Comparison of RMSE for all SSs in training and testing area.

RMSE	OLS	RR	LASSO	EN
training area	0.4318	0.4881	0.5966	0.5200
testing area	0.6935	0.6397	0.5972	0.6071

$\omega = 3v_1$ (in particular on testing data). The SS based on the OLS has large deviations from testing data. The values of the OLS SS parameters are reduced by squared ℓ_2 norm penalisation of RR method. With the penalty of LASSO method, we can reduce certain number of ETM variables. In this case the value of the parameter β_2 is zero and the SS designed by LASSO acquires a linear course and structurally corresponds to the ideal case $\omega = 3v_1$ (grey line in Fig. 2.6). The weighting parameter α of the EN was chosen to be exactly half ($\alpha = 0.5$), not preferring either of RR or LASSO methods. In Fig. 2.6 we can observe the expected results. The blue line (EN SS) is a compromise between orange (RR SS) and green lines (LASSO SS). According to Table 2.5 we can conclude that the smallest difference between RMSE in the training and testing area has SS based on LASSO. Although this SS has the largest RMSE value in training area, in testing area this value is the smallest. It follows that for the newly measured data, LASSO SS has the most accurate estimate.

All methods for SS design have their advantages and disadvantages. Depending on the process requirements, the optimal method can be chosen to ensure the best results.

CHAPTER 3

Case Study

3.1 Two Tanks in Series

We study a system of two tanks in series without interaction (Fig. 3.1). There are two inputs into the process. One inlet flow enters the first tank $q_{0,1}(t)$. Another inlet flow $q_{0,2}(t)$ enter the second tank. The liquid level heights (LLHs) of the first tank $h_1(t)$ and the second $h_2(t)$ represents state variables. The controlled variable is the liquid level height (LLH) of the second tank $h_2(t)$. To ensure quality of control and manipulate the process, we need information about the states of the process. Since $h_2(t)$ is a measurable and controllable state, we need to find out information about the unmeasured state $h_1(t)$. In this work $h_1(t)$ represents a DTM variable, which need to be estimated using SSs. The ETM variables that are used to describe $h_1(t)$: $q_{0,1}(t)$, $q_{0,2}(t)$, $h_2(t)$.

The system parameters are given in the Table 3.1. Parameter k_{nn} represents the valve constant of the corresponding tank and F_n is the cross-sectional area of the corresponding tank.

Table 3.1: System parameters.

Parameter	Value	Unit
k_{11}	1.15	$\mathrm{m}^{2.5}/\mathrm{s}$
k_{22}	1.3	$\mathrm{m}^{2.5}/\mathrm{s}$
F_1	0.25	m^2
F_2	0.8	m^2
$q^s_{0,1}$	0.3	m^3/s
$q^s_{0,2}$	0.5	m^3/s

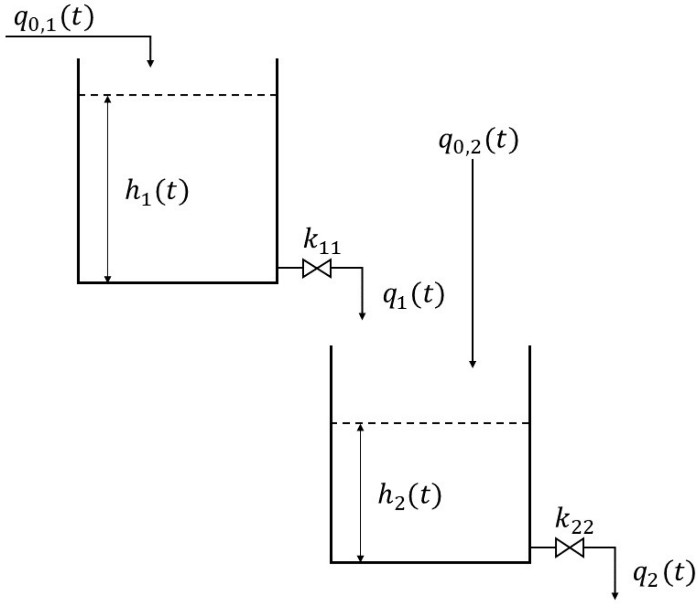

Figure 3.1: The schematic diagram of two tanks in series without interaction.

3.1.1 Mathematical Model

Assuming that the density of the liquid does not change over time, the mass balance of the process can be written as follows [1]:

$$\frac{dh_1(t)}{dt} = \frac{q_{0,1}(t)}{F_1} - \frac{k_{11}}{F_1}\sqrt{h_1(t)}, \tag{3.1}$$

for the first tank. Mass balance for the second tank:

$$\frac{dh_2(t)}{dt} = \frac{q_{0,2}(t)}{F_2} + \frac{k_{11}}{F_2}\sqrt{h_1(t)} - \frac{k_{22}}{F_2}\sqrt{h_2(t)}. \tag{3.2}$$

The inlet flows $q_{0,1}(t)$ and $q_{0,2}(t)$ are independent manipulated variables [1]. To achieve a steady state condition, we set the input flows to the constant values. Then the LLHs in the tanks are also steady. Initial steady values of inlet flows ($q^s_{0,1}$, $q^s_{0,2}$) and LLHs (h^s_1, h^s_2) are given in Table 3.1. Mathematically expressed, this means that the derivatives of LLHs are zero according to time [1]

$$\frac{dh^s_1}{dt} = \frac{dh^s_2}{dt} = 0. \tag{3.3}$$

The relations of the steady LLHs values h_1^s, h_2^s can be expressed from the mass balances (3.1) and (3.2)

$$h_1^s = \left(\frac{q_{0,1}^s}{k_{11}}\right)^2 = \left(\frac{0.3}{1.15}\right)^2 = 0.0681\ m, \tag{3.4}$$

$$h_2^s = \left(\frac{q_{0,1}^s}{k_{22}} + \frac{q_{0,2}^s}{k_{22}}\right)^2 = \left(\frac{0.3}{1.3} + \frac{0.5}{1.3}\right)^2 = 0.3787\ m. \tag{3.5}$$

The derived model is non-linear due to the presence of the square root of state variables. In order to design an observer and a controller, we have linearise the model. We introduce deviation variables for linearisation:

$$x_1(t) = h_1(t) - h_1^s, \qquad u_1(t) = q_{0,1}(t) - q_{0,1}^s, \tag{3.6}$$

$$x_2(t) = h_2(t) - h_2^s, \qquad u_2(t) = q_{0,2}(t) - q_{0,2}^s. \tag{3.7}$$

The input-output state-space model can be written

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}}_{\boldsymbol{A}} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \underbrace{\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}}_{\boldsymbol{B}} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}, \tag{3.8}$$

$$y(t) = \underbrace{\begin{pmatrix} c_{11} & c_{12} \end{pmatrix}}_{\boldsymbol{C}} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \underbrace{\begin{pmatrix} d_{11} & d_{12} \end{pmatrix}}_{\boldsymbol{D}} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}, \tag{3.9}$$

where

$$a_{11} = \frac{\partial\left(\frac{q_{0,1}(t)}{F_1} - \frac{k_{11}}{F_1}\sqrt{h_1(t)}\right)}{\partial h_1(t)} = -\frac{k_{11}}{2F_1\sqrt{h_1^s}}, \tag{3.10}$$

$$a_{12} = \frac{\partial\left(\frac{q_{0,1}(t)}{F_1} - \frac{k_{11}}{F_1}\sqrt{h_1(t)}\right)}{\partial h_2(t)} = 0, \tag{3.11}$$

$$a_{21} = \frac{\partial\left(\frac{q_{0,2}(t)}{F_2} + \frac{k_{11}}{F_2}\sqrt{h_1(t)} - \frac{k_{22}}{F_2}\sqrt{h_2(t)}\right)}{\partial h_1(t)} = \frac{k_{11}}{2F_2\sqrt{h_1^s}}, \tag{3.12}$$

$$a_{22} = \frac{\partial\left(\frac{q_{0,2}(t)}{F_2} + \frac{k_{11}}{F_2}\sqrt{h_1(t)} - \frac{k_{22}}{F_2}\sqrt{h_2(t)}\right)}{\partial h_1(t)} = -\frac{k_{22}}{2F_2\sqrt{h_2^s}}. \tag{3.13}$$

The first inlet flow enters the first tank, the second inlet flow to the second tank. It follows that only b_{11} and b_{22} of matrix $\boldsymbol{B}$ are nonzero

$$b_{11} = \frac{\partial\left(\frac{q_{0,1}(t)}{F_1} - \frac{k_{11}}{F_1}\sqrt{h_1(t)}\right)}{\partial q_{0,1}(t)} = \frac{1}{F_1}, \tag{3.14}$$

$$b_{22} = \frac{\partial\left(\frac{q_{0,2}(t)}{F_2} + \frac{k_{11}}{F_2}\sqrt{h_1(t)} - \frac{k_{22}}{F_2}\sqrt{h_2(t)}\right)}{\partial q_{0,2}(t)} = \frac{1}{F_2}. \tag{3.15}$$

Then we can linearised model as follows:

$$\frac{dx_1(t)}{dt} = a_{11}x_1(t) + b_{11}u_1(t), \tag{3.16}$$

$$\frac{dx_2(t)}{dt} = a_{21}x_1(t) + a_{22}x_2(t) + b_{22}u_2(t), \tag{3.17}$$

As our controlled variable is the LLH of the second tank

$$y(t) = h_2(t), \tag{3.18}$$

the matrix $\boldsymbol{C}$ looks like

$$\boldsymbol{C} = \begin{pmatrix} 0 & 1 \end{pmatrix}. \tag{3.19}$$

The inputs do not appear in the output equation, so the matrix $\boldsymbol{D}$ has zero values

$$\boldsymbol{D} = \begin{pmatrix} 0 & 0 \end{pmatrix}. \tag{3.20}$$

The final version of the input-output state-space model is numerically given as

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} -4.4083 & 0 \\ 2.7552 & -1.3203 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 2.00 & 0 \\ 0 & 1.25 \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}, \tag{3.21}$$

$$y(t) = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{3.22}$$

3.1.2 LQR Control Strategy

The system is controllable if the rank of *controllability matrix* $\boldsymbol{Q_c}$ is equal to the dimension of the vector of states $\boldsymbol{x}$. $\boldsymbol{Q_c}$ is defined as [11]:

$$\boldsymbol{Q_c} = (\boldsymbol{B}\ \boldsymbol{AB}\ \boldsymbol{A}^2\boldsymbol{B}\ \ldots\ \boldsymbol{A}^{k-1}\boldsymbol{B}), \tag{3.23}$$

where k represents dimension of the vector $\boldsymbol{x}$. The resulting $\boldsymbol{Q_c}$ matrix for our process looks as follows

$$\boldsymbol{Q_c} = (\boldsymbol{B}\ \boldsymbol{AB}) = \begin{pmatrix} 2 & 0 & -8.82 & 0 \\ 0 & 1.25 & 5.51 & -1.65 \end{pmatrix}. \tag{3.24}$$

Rank of *controllability matrix* $\boldsymbol{Q_c}$ is 2. Our process has two state variables. This means that two tanks in series are controllable.

System of two tanks in series is controlled by LQR controller with integral action. LQR is a type of optimal control. For initial conditions $t_0 = 0$ s:

$$\boldsymbol{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{3.25}$$

the objective function is defined as

$$J = \int_0^\infty \left(\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t) \right) dt, \tag{3.26}$$

where $\boldsymbol{Q}$ is a real symmetric positive semidefinite weighting matrix and $\boldsymbol{R}$ is a real symmetric positive definite weighting matrix [11]. These two matrices are tuning matrices of the LQR controller. According to the Bryson's rule $\boldsymbol{Q}$ and $\boldsymbol{R}$ are diagonal matrices whose diagonal elements (Q_{kk} for $\boldsymbol{Q}$ and R_{jj} for $\boldsymbol{R}$) are expressed as the reciprocals of the squares of maximum acceptable values of the state variables ($\boldsymbol{x}(t)$) and the input control variables ($\boldsymbol{u}(t)$) [12]. The diagonal elements Q_{kk}, R_{jj} can be written as

$$Q_{kk} = \frac{1}{\max(x_k^2)}, \tag{3.27}$$

$$R_{jj} = \frac{1}{\max(u_j^2)}, \tag{3.28}$$

where k represents controlled state number and j represents input number. Numerically for our system

$$\boldsymbol{Q} = \begin{bmatrix} \frac{1}{h_1^{s\,2}} & 0 \\ 0 & \frac{1}{h_2^{s\,2}} \end{bmatrix} = \begin{bmatrix} 215.9 & 0 \\ 0 & 6.97 \end{bmatrix}, \tag{3.29}$$

$$\boldsymbol{R} = \begin{bmatrix} \frac{1}{{q_{0,1}^s}^2} & 0 \\ 0 & \frac{1}{{q_{0,2}^s}^2} \end{bmatrix} = \begin{bmatrix} 11.11 & 0 \\ 0 & 4 \end{bmatrix}. \tag{3.30}$$

This is a good starting point for tuning $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices. Feedback optimal control law is a given by the equation [11]

$$\boldsymbol{u}(t) = -\boldsymbol{K}(t)\boldsymbol{x}(t), \tag{3.31}$$

where $\boldsymbol{K}$ represents gain of LQR controller. It is given as [11]:

$$\boldsymbol{K}(t) = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}(t). \tag{3.32}$$

The matrix $\boldsymbol{P}$ can be obtained from algebraic Riccati equation [11]

$$\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{Q} = 0. \tag{3.33}$$

LQR controller is a proportional state-feedback controller. Under reference changes or due to disturbances, one might encounter a steady-state control error. This problem

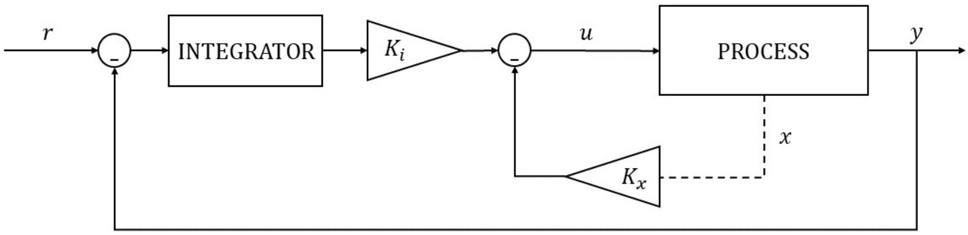

Figure 3.2: LQR control scheme.

can be solved by adding integral action to the closed-loop system [11]. The procedure is such that new states $\dot{\boldsymbol{x}}_{\boldsymbol{int}}(t)$ is added as integrator to the closed-loop system

$$\dot{\boldsymbol{x}}_{\boldsymbol{int}}(t) = \boldsymbol{r}(t) - \boldsymbol{y}(t) = \boldsymbol{r}(t) - \boldsymbol{C}\boldsymbol{x}(t), \tag{3.34}$$

where $\boldsymbol{r}(t)$ represents a reference. Feedback optimal control law with integral action is given as

$$\boldsymbol{u}(t) = -\boldsymbol{K}_{\boldsymbol{x}}(t)\boldsymbol{x}(t) - \boldsymbol{K}_{\boldsymbol{int}}(t)\boldsymbol{x}_{\boldsymbol{int}}(t). \tag{3.35}$$

$\boldsymbol{K}_{\boldsymbol{x}}$ represents a gain of proportional feedback controller and $\boldsymbol{K}_{\boldsymbol{int}}$ is a gain of integral action. In Fig. 3.2 is shown illustration scheme of state-feedback control with integral action, where states ($\boldsymbol{x}$) will be estimated by state observer.

The number of integrators is equal to the dimension of the control error [11]. We only control the LLH in the second tank, so there is only one integrator. The extended matrices looks as follows:

$$\begin{pmatrix} \dot{\boldsymbol{x}}(t) \\ \dot{x}_{int}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} & 0 \\ -\boldsymbol{C} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}(t) \\ x_{int}(t) \end{pmatrix} + \begin{pmatrix} \boldsymbol{B} \\ 0 \end{pmatrix} \boldsymbol{u}(t), \tag{3.36}$$

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_{int}(t) \end{pmatrix} = \underbrace{\begin{pmatrix} -4.4083 & 0 & 0 \\ 2.7552 & -1.3203 & 0 \\ 0 & -1 & 0 \end{pmatrix}}_{\boldsymbol{A_e}} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_{int}(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 2 & 0 \\ 0 & 1.25 \\ 0 & 0 \end{pmatrix}}_{\boldsymbol{B_e}} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}. \tag{3.37}$$

To realise equation (3.26), the weight matrix $\boldsymbol{Q}$ also has a new parameter (equal to one) on the diagonal

$$\boldsymbol{Q_e} = \begin{pmatrix} \boldsymbol{Q} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 215.9 & 0 & 0 \\ 0 & 6.97 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.38}$$

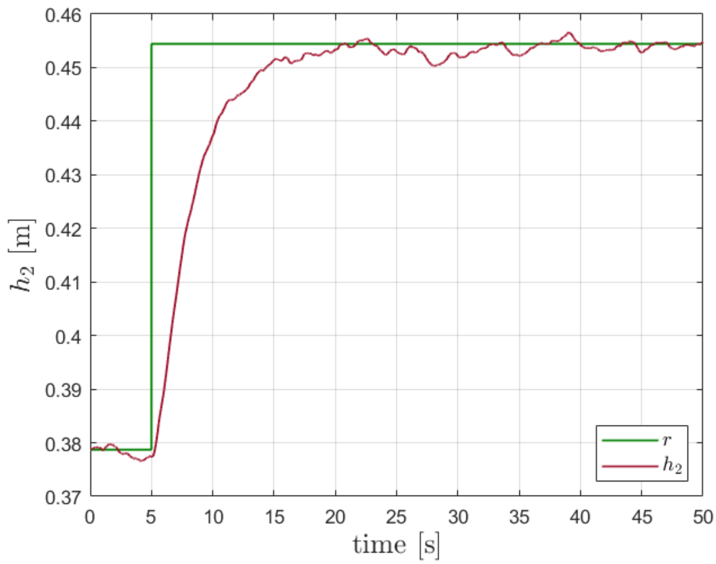

Figure 3.3: LQR control of two tanks in series.

The gain of LQR control with integral action $\boldsymbol{K_e}(t)$ is obtained from equations (3.32) and (3.33), where instead of matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{Q}$ we use their extended form $\boldsymbol{A_e}$, $\boldsymbol{B_e}$, $\boldsymbol{Q_e}$. The resulting gain $\boldsymbol{K_e}(t)$ looks as follows:

$$\boldsymbol{K_e}(t) = \Bigg(\underbrace{\begin{matrix} 2.7549 & 0.1144 \\ 0.1986 & 0.8411 \end{matrix}}_{\boldsymbol{K_x}(t)} \quad \underbrace{\begin{matrix} 0.0707 \\ 0.4859 \end{matrix}}_{\boldsymbol{K_{int}}(t)} \Bigg) . \tag{3.39}$$

The LQR controller adjusts the manipulated variables to steer the controlled variable h_2 to the reference quite quickly. In Fig. 3.3 we can see that controlled variables h_2 of non-linear process (two tanks in series) reach reference (h_2 increase by 20 %). The great advantage of this controller is the integral action. The integral action helps the controlled variable to reach the reference regardless of any model inaccuracies. Since height h_1 can not be measured and the measurement of h_2 contains a white Gaussian noise, we use KF to estimate and smooth information about these states. The state LQR controller uses states estimated using KF, which design will be described in the following section.

3.2 Observation of State Variables

Liquid Level Measurement makes roughly over 50 % of all the process measurements in current process industry [4]. The environment in which the measurement is performed is most often polluted, harsh and inaccessible. LLH sensors may require special abilities to work under these extreme circumstances (to resist high temperature, pressure and electromagnetic interference etc.) [4]. They also need to have good applicability, reliability, high resolution and precision, large sensitivity etc. [4]. These special requirements can be very expensive. Software alternatives are more used due these demanding requirements.

3.2.1 Kalman Filter

The system is observable if the rank of *observability matrix* $\boldsymbol{Q_o}$ is equal to the dimension of the vector of states $\boldsymbol{x}$. $\boldsymbol{Q_o}$ is defined as [11]:

$$\boldsymbol{Q_o} = \begin{pmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^{k-1} \end{pmatrix} \tag{3.40}$$

where k represents dimension of the vector $\boldsymbol{x}$. The resulting $\boldsymbol{Q_o}$ matrix for our process looks as follows

$$\boldsymbol{Q_o} = \begin{pmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2.76 & -1.32 \end{pmatrix} \tag{3.41}$$

Rank of *observability matrix* $\boldsymbol{Q_c}$ is 2. Our process has two state variables. This means that two tanks in series are observable.

The LLH of the first tank h_1 is DTM state variable in our case of two tanks in series. A possible option of states (DTM variables) estimation is to express them as dependence on inputs ($\boldsymbol{u}(t)$) and outputs ($\boldsymbol{y}(t)$) measurements [11]. This method of estimation can be performed by using a KF as a state observer (Fig. 3.4).
In general, states (DTM variables) can be mathematically described using

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{Ax}(t) + \boldsymbol{Bx}(t) + \xi_x(t), \qquad \xi_x(t) \sim \mathcal{N}(0, \boldsymbol{Q}), \tag{3.42}$$

where $\xi_x(t)$ represents properties of a white Gaussian noise with normal distribution. Its mean is equal to zero and its standard deviation can be represented by a covariance matrix $\boldsymbol{Q}$ [11].

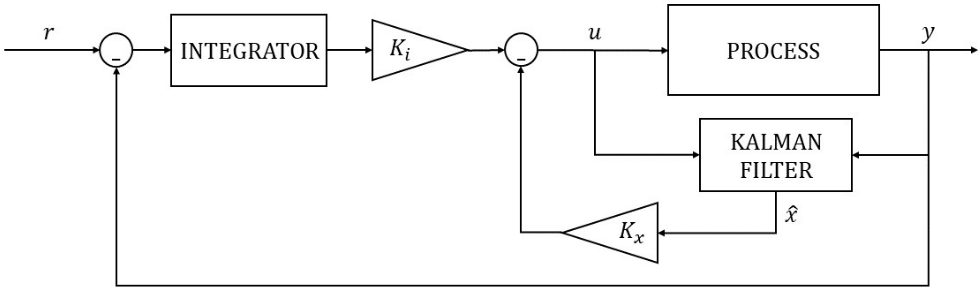

Figure 3.4: Implementation of KF in LQR control scheme.

Initial condition is also approximately known

$$\boldsymbol{x}(0) = \bar{\boldsymbol{x}}_0 + \xi_0, \qquad \xi_0 \sim \mathcal{N}(0, \boldsymbol{P}_0). \tag{3.43}$$

Value of $\bar{\boldsymbol{x}}_0$ is an initial guess and ξ_0 represents with Gaussian noise with zero mean and a covariance matrix $\boldsymbol{P}_0$ [11].
The deviation of the output measurements can be mathematically expressed

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \xi(t), \qquad \xi(t) \sim \mathcal{N}(0, \boldsymbol{R}), \tag{3.44}$$

with a white Gaussian noise $\xi(t)$ with zero mean and a covariance matrix $\boldsymbol{R}$ [11].

Since the KF is the optimal state observer, the objective function of state estimation is defined as follows [11]:

$$\begin{aligned} J = &\frac{1}{2}[\boldsymbol{x}(0) - \bar{\boldsymbol{x}}_0]^T \boldsymbol{P}_0^{-1}[\boldsymbol{x}(0) - \bar{\boldsymbol{x}}_0] \\ &+ \frac{1}{2}\int_0^t \Big([\dot{\boldsymbol{x}}(t) - \boldsymbol{A}\boldsymbol{x}(t)]^T \boldsymbol{Q}^{-1}[\dot{\boldsymbol{x}}(t) - \boldsymbol{A}\boldsymbol{x}(t)]\Big)dt \\ &+ \frac{1}{2}\int_0^t \Big([\boldsymbol{y}(t) - \boldsymbol{C}\boldsymbol{x}(t)]^T \boldsymbol{R}^{-1}[\boldsymbol{y}(t) - \boldsymbol{C}\boldsymbol{x}(t)]\Big)dt. \end{aligned} \tag{3.45}$$

Covariance matrices $\boldsymbol{P}_0$, $\boldsymbol{Q}$, $\boldsymbol{R}$ are weighting matrices whose tuning can improve state estimation performance. By solving the optimisation problem (3.45), we get the following equation of estimation:

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{L}(t)[\boldsymbol{y}(t) - \boldsymbol{C}\hat{\boldsymbol{x}}(t)], \qquad \hat{\boldsymbol{x}}(0) = \bar{\boldsymbol{x}}_0. \tag{3.46}$$

$\boldsymbol{L}$ represents the gain of time-varying KF and it is given as [11]

$$\boldsymbol{L}(t) = \boldsymbol{P}(t)\boldsymbol{C}^T\boldsymbol{R}^{-1}. \tag{3.47}$$

$\boldsymbol{P}(t)$ can be obtain from differential Riccati equation [11]

$$\dot{\boldsymbol{P}}(t) = \boldsymbol{P}(t)\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{P}(t) - \boldsymbol{P}(t)\boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{C}\boldsymbol{P}(t) + \boldsymbol{Q}, \qquad \boldsymbol{P}(0) = \boldsymbol{P}_0. \tag{3.48}$$

To simplify the design, we introduce an asymptotic KF. The derivative of covariance matrix $\boldsymbol{P}(t)$ converges to zero. The $\boldsymbol{P}$ gain is no longer time dependent, so the gain $\boldsymbol{L}$ is no longer either [14]

$$\boldsymbol{0} = \boldsymbol{P}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{P} - \boldsymbol{P}\boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{C}\boldsymbol{P} + \boldsymbol{Q}, \qquad \boldsymbol{P}(0) = \boldsymbol{P}_0, \tag{3.49}$$

$$\boldsymbol{L} = \boldsymbol{P}\boldsymbol{C}^T\boldsymbol{R}^{-1}. \tag{3.50}$$

State estimation equation and estimation error is given as [14]

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{L}[\boldsymbol{y}(t) - \boldsymbol{C}\hat{\boldsymbol{x}}(t)], \qquad \hat{\boldsymbol{x}}(0) = \bar{\boldsymbol{x}}_0, \tag{3.51}$$

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t) \sim \mathcal{N}(0, \boldsymbol{P}). \tag{3.52}$$

The deviation of linear KF from the non-linear model of two tanks in series, can be described by disturbance $\boldsymbol{d}(t)$. Then process can be written as follows [13]

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{B}\boldsymbol{d}(t), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{3.53}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t). \tag{3.54}$$

The disturbance is mathematically expressed as [13]:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}}(t) = \boldsymbol{A}_{\boldsymbol{d}}\boldsymbol{x}_{\boldsymbol{d}}(t), \qquad \boldsymbol{x}_{\boldsymbol{d}}(0) = \boldsymbol{x}_{\boldsymbol{d}0}, \tag{3.55}$$

$$\boldsymbol{d}(t) = \boldsymbol{C}_{\boldsymbol{d}}\boldsymbol{x}(t). \tag{3.56}$$

An extended asymptotic KF can be used to eliminate the effect of disturbance. Estimation equation (3.51) can be extend as follows [13]

$$\dot{\hat{\boldsymbol{x}}}(t) = \underbrace{\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B}\boldsymbol{C}_d \\ \boldsymbol{0} & \boldsymbol{A}_d \end{pmatrix}}_{\boldsymbol{A}_{\boldsymbol{KF}}} \hat{\boldsymbol{x}}(t) + \underbrace{\begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{pmatrix}}_{\boldsymbol{B}_{\boldsymbol{KF}}} \boldsymbol{u}(t) + \begin{pmatrix} \boldsymbol{K}_f \\ \boldsymbol{K}_d \end{pmatrix} \Big(\boldsymbol{y}(t) - \underbrace{\begin{pmatrix} \boldsymbol{C} & \boldsymbol{0} \end{pmatrix}}_{\boldsymbol{C}_{\boldsymbol{KF}}} \hat{\boldsymbol{x}}(t) \Big), \tag{3.57}$$

where $\boldsymbol{K}_f$ and $\boldsymbol{K}_d$ represent parts of the $\boldsymbol{L}$ gain from equation (3.51).
Estimation error is given as [13]:

$$\dot{\boldsymbol{e}}(t) = \left(\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B}\boldsymbol{C}_d \\ 0 & \boldsymbol{A}_d \end{pmatrix} - \begin{pmatrix} \boldsymbol{K}_f \\ \boldsymbol{K}_d \end{pmatrix} \begin{pmatrix} \boldsymbol{C} & \boldsymbol{0} \end{pmatrix} \right) \boldsymbol{e}(t). \tag{3.58}$$

For initial conditions

$$\boldsymbol{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \boldsymbol{P}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.59}$$

and following weighting matrices

$$\boldsymbol{Q} = \begin{bmatrix} 10^4 & 0 & 0 \\ 0 & 10^4 & 0 \\ 0 & 0 & 10^4 \end{bmatrix}, \qquad \boldsymbol{R} = 10^4, \tag{3.60}$$

we can obtain gain $\boldsymbol{L}$ from equations (3.49) and (3.50)

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{K}_f \\ K_d \end{pmatrix} = \begin{pmatrix} 0.0003 \\ 0.9999 \\ 0.0001 \end{pmatrix}. \tag{3.61}$$

Equation (3.57) numerically can be written

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}}(t) = &\begin{pmatrix} -4.4083 & 0 & 2 \\ 2.7552 & -1.3203 & 1.25 \\ 0 & 0 & 0 \end{pmatrix} \hat{\boldsymbol{x}}(t) + \begin{pmatrix} 2 & 0 \\ 0 & 1.25 \\ 0 & 0 \end{pmatrix} \boldsymbol{u}(t) \\ &+ \begin{pmatrix} 0.0003 \\ 0.9999 \\ 0.0001 \end{pmatrix} \Big(\boldsymbol{y}(t) - \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \hat{\boldsymbol{x}}(t) \Big). \end{aligned} \tag{3.62}$$

The last element of matrix $\boldsymbol{A_{KF}}$ on diagonal is zero ($A_d = 0$). This extension reduce deviation of linear KF from non-linear model of two tanks in series.

In Fig. 3.5 we can see the comparison of states from non-linear process and from KF for increased h_2^s value by 20 % . The second state x_{h_2} is measurable and represent output variable $y(t)$ of non-linear process of two tanks in series (NL: x_{h_2}). We simulate the measurement noise by a white Gaussian noise

$$y(t) = \boldsymbol{C}\boldsymbol{x}(t) + \xi(t), \qquad \xi(t) \sim \mathcal{N}(0,\ 0.001). \tag{3.63}$$

The noisy signal of the second state x_{h_2} is filtered almost ideally by KF (KF: x_{h_2}). The first state x_{h_1} is unmeasured, but we have the ability to simulate its course in software MATLAB Simulink (NL: x_{h_1}). In this way we can evaluate the estimation accuracy of the first state x_{h_1} by using KF (KF: x_{h_1}).

In the previous example we saw almost ideal state estimation. This is the case when state-space model of the process is correct. If there is a change in process or the

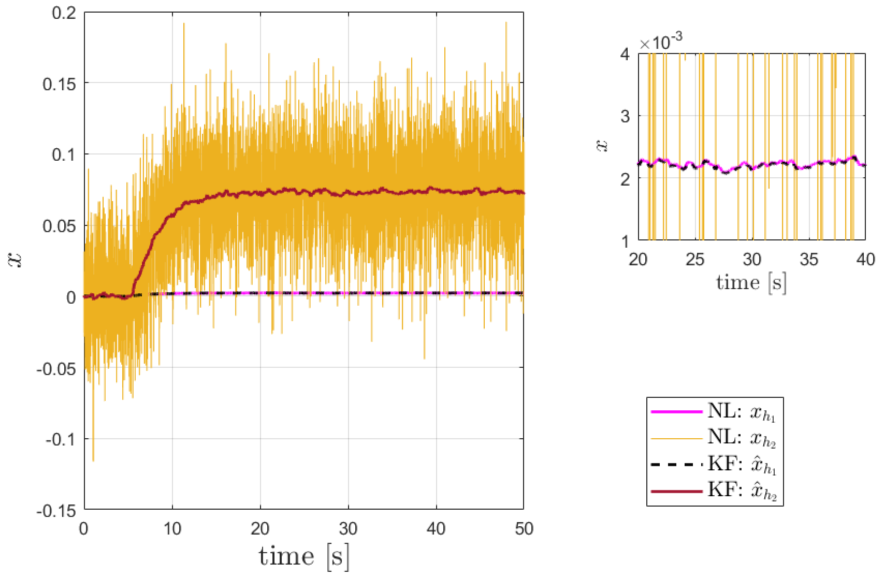

Figure 3.5: States estimation by extended asymptotic KF.

state-space model gets easily inaccurate, the state estimation is no longer accurate. As a result, there are deviations in estimation of states.

Let us make a small changes in matrix $\boldsymbol{A}$. With an increase a_{11} value by 30 %, the estimation equation look as follows

$$\begin{aligned}\dot{\hat{\boldsymbol{x}}}(t) &= \begin{pmatrix} \mathbf{-5.7308} & 0 & 2 \\ 2.7552 & -1.3203 & 1.25 \\ 0 & 0 & 0 \end{pmatrix} \hat{\boldsymbol{x}}(t) + \begin{pmatrix} 2 & 0 \\ 0 & 1.25 \\ 0 & 0 \end{pmatrix} \boldsymbol{u}(t) \\ &+ \begin{pmatrix} 0.0003 \\ 0.9999 \\ 0.0001 \end{pmatrix} \Big(\boldsymbol{y}(t) - \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \hat{\boldsymbol{x}}(t) \Big). \end{aligned} \tag{3.64}$$

This change in the state-space model cause inaccurate estimation of KF. In Fig (3.6) we can see the difference in states between non-linear process of two tanks in series (NL: x_{h_1}) and KF (KF: x_{h_1}). The conclusion is that KF is unequivocally dependent on the model of process.

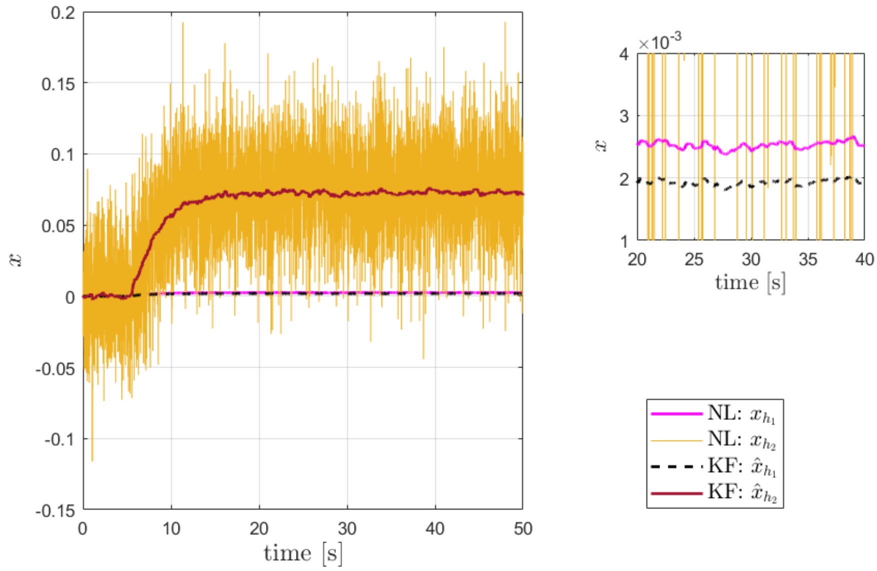

Figure 3.6: State estimation by outdated extended asymptotic KF.

3.2.2 Software Sensor

The main task is to design SSs which can replace a complicated procedure of state observers (KF) designing. We deal with the estimation of the state x_{h_1}. This state is unmeasurable and its estimate is in the form of the steady LLH h_1^s. Mass balance of the first tank under steady state conditions can be obtained from equation (3.1)

$$0 = \frac{q_{0,1}^s}{F_1} - \frac{k_{11}}{F_1}\sqrt{h_1^s}. \tag{3.65}$$

By expressing the DTM variable h_1^s, we get an expression

$$h_1^s = \left(\frac{q_{0,1}^s}{k_{11}}\right)^2. \tag{3.66}$$

We consider the simplest possible structure for SS design (linear SSs). To be able to design linear SS, we need to transform equation (3.66) into a form of linear dependence (2.3)

$$\hat{h}_1^s = \beta_1 q_{0,1}^s, \qquad \beta_1 = \frac{q_{0,1}^s}{k_{11}^2}. \tag{3.67}$$

The steady-state LLH of the first tank h_1^s can also be expressed from the mass balance of the second tank

$$0 = \frac{q_{0,2}^s}{F_2} + \frac{k_{11}}{F_2}\sqrt{h_1^s} - \frac{k_{22}}{F_2}\sqrt{h_2^s}, \tag{3.68}$$

$$h_1^s = \left(\frac{q_{0,2}^s}{k_{11}}\right)^2 - 2\frac{q_{0,2}^s}{k_{11}}\frac{k_{22}\sqrt{h_2^s}}{k_{11}} + \left(\frac{k_{22}\sqrt{h_2^s}}{k_{11}}\right)^2, \tag{3.69}$$

$$\hat{h}_1^s = \beta_2 q_{0,2}^s + \beta_3 h_2^s. \tag{3.70}$$

The task is to determine the parameters β_1, β_2, β_3 by using methods for SS design. Input flows $q_{0,1}^s$, $q_{0,2}^s$ are an ETM variables and h_2^s is measured controlled variable. Based on these SS parameters we get information about the steady state LLH in the first tank h_1^s.

Chapter 4

Case Study Results

4.1 Software Sensors as a Liquid Level Height Estimators

This section shows the most important results of the SS design based on the methods described in Section 2. As we can see from equations (3.67) and (3.70), LLH of the first tank can be described by all ETM variables of our process

$$h_1^s = f(q_{0,1}^s,\ q_{0,2}^s,\ h_2^s). \tag{4.1}$$

In general, SSs can be described by the following linear dependence

$$\hat{h}_1^s = \beta_0 + \beta_1 q_{0,1}^s + \beta_2 q_{0,2}^s + \beta_3 h_2^s. \tag{4.2}$$

Relation (4.2) represents fundamental structure of whose parameters β_0, β_1, β_2, β_3 will be estimated.

4.1.1 Data Obtaining and Distribution

The accuracy and reliability of SSs is highly dependent on the data which they use for training. Noise is always present in the measurements. It is convenient to train SSs with as much available data as possible. With a large amount of data, the estimation accuracy of the SSs tends to be higher.

The data of the ETM variables are obtained by performing step responses of the controlled variable h_2^s in software MATLAB, with respect to its reference. Initial steady state LLH value of the second tank h_2^s was calculated by equation (3.5). Steady LLH has value $h_2^s = 0.3787\ m$. This is the initial value for step responses of LLH h_2.

Step responses are performed in the range ± 25 % of the initial value of h_2^s. We assumed

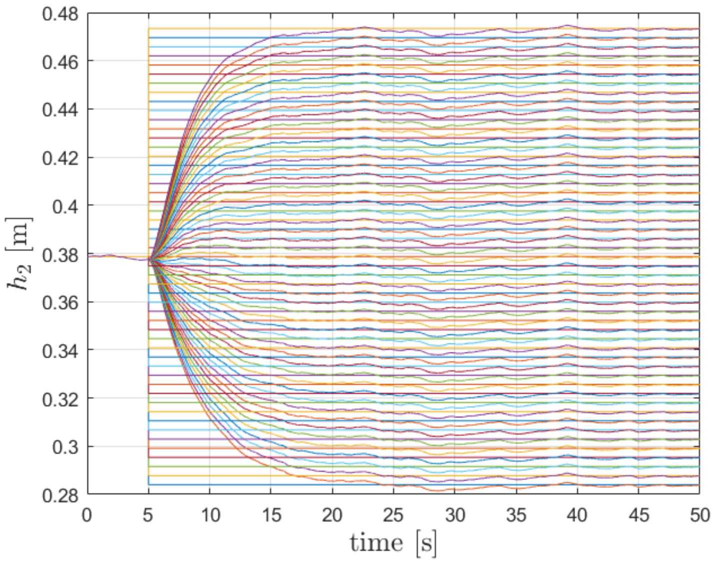

Figure 4.1: 50 step responses of h_2^s in range ±25 %.

that in a given range, the DTM variable h_1^s historical data contains 50 step responses of reference (the same number of measurement are for ETM variables).

Steady values of ETM variables are recorded for each step response. The steady values of all ETM variables were filtered and averaged over the impact of noise. The data is stored in vectors of ETM variables $\boldsymbol{q_{0,1}^s}$, $\boldsymbol{q_{0,1}^s}$, $\boldsymbol{h_2^s}$. These vectors contain 50 values each.

From 50 step responses, 30 were randomly selected. The data corresponding to these 30 step responses are used to train the SSs. The remaining 20 values of ETM variables are considered as testing data. Conclusions of SS accuracy is drawn based on the behavior of the SSs in the testing area.

Due to better interpretation of the results of method for SS design, we assume the simulated reality with noisy data

$$h_{1,i}^s = h_{1,i}^s + \xi_{h_{1,i}}, \qquad \xi_{h_{1,i}} \sim \mathcal{N}(h_{1,i}^s,\ 0.001) \tag{4.3}$$

$$h_{2,i}^s = h_{2,i}^s + \xi_{h_{2,i}}, \qquad \xi_{h_{2,i}} \sim \mathcal{N}(h_{2,i}^s,\ 0.001). \tag{4.4}$$

Gaussian noise $\xi_{h_{n,i}}$ with the mean equal to the corresponding value of the LLH $h_{n,i}^s$ and standard deviation 0.001.

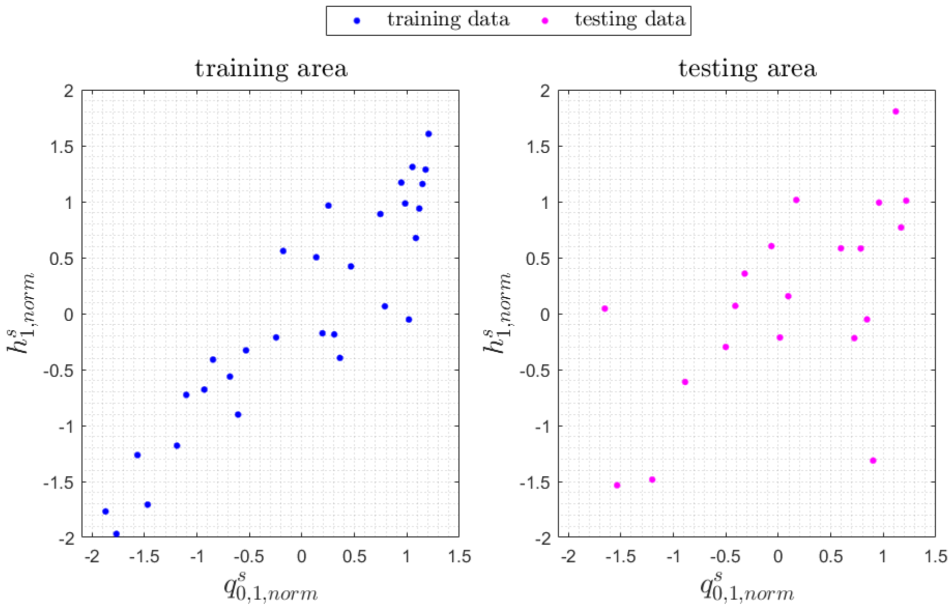

Figure 4.2: Training and testing area of SSs.

Fig. 4.2 shows the graphical dependence of $\boldsymbol{h_1^s}$ on $\boldsymbol{q_{0,1}^s}$ for the training and testing area of the SSs. For better comparison of SSs in training and testing area all data are normalised. Example of normalisation is given for $h^s_{1,i}$

$$h^s_{1,i,norm} = \frac{h^s_{1,i} - \mu_{h^s_{1,i}}}{\sigma_{h^s_{1,i}}}. \tag{4.5}$$

Mean $\mu_{h^s_{1,i}}$ and standard deviation $\sigma_{h^s_{1,i}}$ are calculated as follows [8]

$$\mu_{h^s_{1,i}} = \frac{1}{N}\sum_{i=1}^{N} h^s_{1,i}, \tag{4.6}$$

$$\sigma_{h^s_{1,i}} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(h^s_{1,i} - \mu_{h^s_{1,i}})^2}, \tag{4.7}$$

where N represents number of measurements.

4.1.2 Training and Testing of the Software Sensors

SSs are trained and tested based on the data from Fig 4.2. They are designed according to the methods for SS design, described in the Section 2. According to equation (4.2),

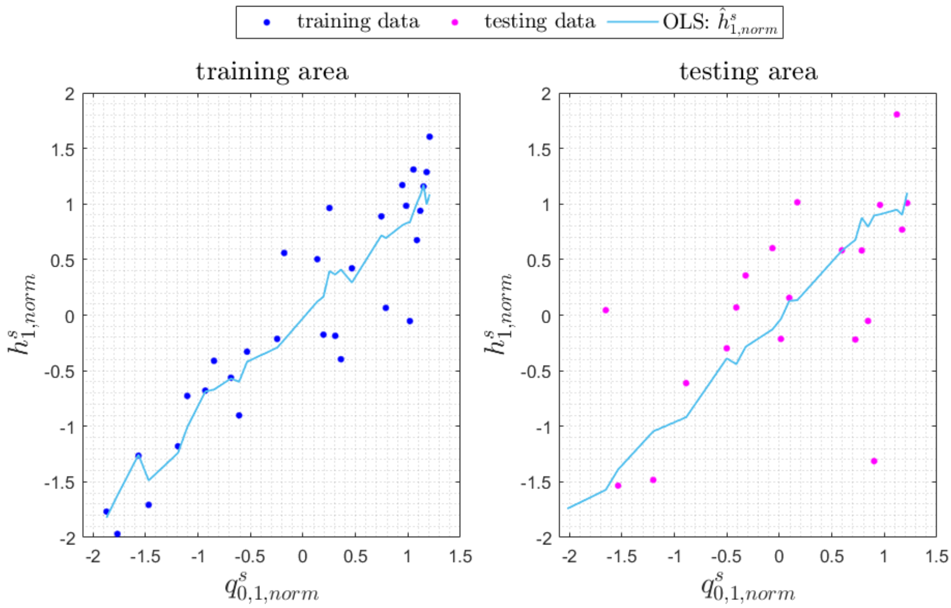

Figure 4.3: Comparison of OLS SS performance in training (left-hand plot) and testing (right-hand plot) area.

SSs have three ETM input variables ($\boldsymbol{q^s_{0,1}}$, $\boldsymbol{q^s_{0,2}}$, $\boldsymbol{h^s_2}$).

The first SS is designed by the OLS. By solving the optimisation problem from equation (2.9) using training data (30 measurements), we get the vector of SS parameters. Since we worked with normalised data, we obtained the normalised SS parameters. After conversion to real values, we get the following result

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 0.0001 \\ 0.1257 \\ 0.2098 \\ -0.1825 \end{bmatrix}. \tag{4.8}$$

The resulting equation of the OLS SS looks as follows

$$\hat{h}^s_1 = 0.0001 + 0.1257 q^s_{0,1} + 0.2098 q^s_{0,2} - 0.1825 h^s_2. \tag{4.9}$$

Figure 4.3 shows that the SS designed by the OLS tries to make a linear regression, while minimising deviations from the measured noisy data. We can see in the training area that regression line try to be as close as possible to measured data. According

Table 4.1: RMSE for OLS SS in training and testing area.

	training area	testing area
RMSE	0.3913	0.7703

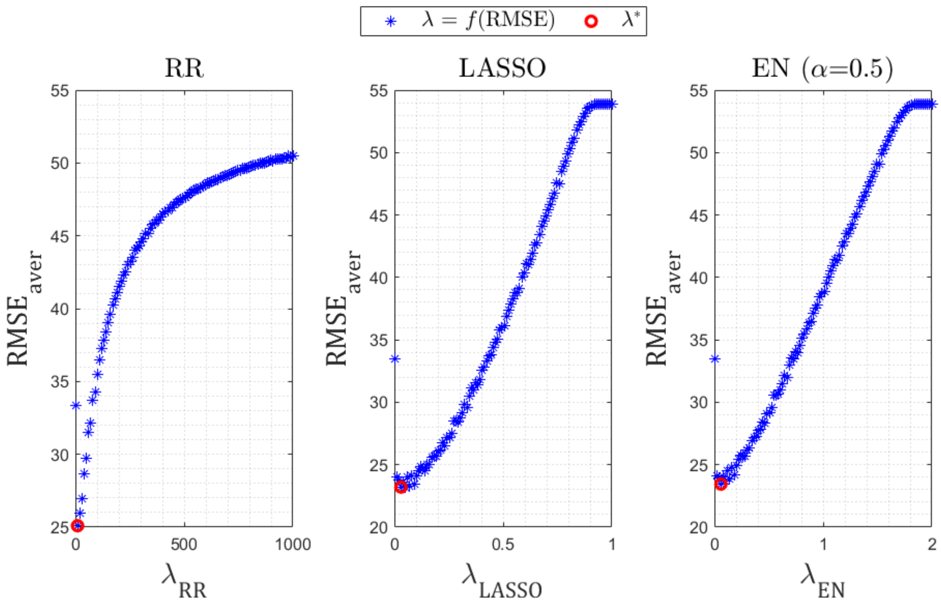

Figure 4.4: Optimal weighting parameters.

to Table 4.1, the RMSE for the testing area (0.7703) has almost double value of the RMSE for the training area (0.3913). This is the expected result, since the testing data is not used for OLS SS training. From the training area, we expect β_1 to be positive. The resulting value of β_1 is 0.1257, which corresponds to regression in the training area.

The following are three "shrinkage" methods, which use penalisation of SS parameters to improve the OLS method. The question was how to choose the best weighting parameters. The methodology we have chosen is to express the dependence of the RMSE value on the weighting parameter λ.

The initial interval of weighing parameters λ for all "shrinkage" methods (RR, LASSO, EN) was [0, 1000]. In order to obtain a graphically representative dependence RMSE $= f(\lambda)$, the intervals are narrowed. For RR the width remained the same

Table 4.2: RMSE for RR SS in training and testing area.

	training area	testing area
RMSE	0.3959	0.7461

[0, 1000], LASSO [0, 1], EN [0, 2] as is shown by Fig. 4.4.

As already mentioned, we simulate the measurement noise in the data by adding the white Gaussian noise (equations (4.3), (4.4)). This means the different data for each new value from the λ interval. To neutralise this factor, 100 data sets were generated for the aforementioned ranges of the λ interval. The RMSE values of the SSs was calculated for each set. In total, there were 100 RMSE values for one value of λ. We get the average RMSE value from these 100 values ($\mathrm{RMSE}_{\mathrm{aver}}$) and this represents one point in the graphs in Fig 4.4. There are 100 points evaluated in this way for each of three λ intervals. The resulting dependencies RSME on λ are shown in Figure 4.4. The optimal λ is the one that gives the lowest RMSE value. We can see that for all "shrinkage" methods optimal weighting parameter λ^* converge near to zero (in Figure 4.4 marked with a red circle). Since the weighting parameters cannot take a negative value, the λ^* values of all three methods (RR, LASSO, EN) are

$$\lambda^*_{\mathrm{RR}} = 0.01, \qquad \lambda^*_{\mathrm{LASSO}} = 0.01, \qquad \lambda^*_{\mathrm{EN}} = 0.01. \tag{4.10}$$

The second weighing parameter of the EN method has value $\alpha = 0.5$. This value is chosen to reach a trade-off between RR and LASSO methods.

The second SS which is designed for the steady LLH estimation of the first tank is the SS based on RR. As in the previous case, using the training data and solving the optimisation from equation (2.16), where $\lambda = \lambda^*_{\mathrm{RR}} = 0.01$, we obtained the following vector of SS parameters

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 0.0002 \\ 0.4282 \\ 0.0643 \\ -0.0591 \end{bmatrix}. \tag{4.11}$$

The resulting equation of the steady LLH estimation in the first tank by SS based on RR

$$\hat{h}^s_1 = 0.0002 + 0.4282 q^s_{0,1} + 0.0643 q^s_{0,2} - 0.0591 h^s_2. \tag{4.12}$$

In Figure 4.5, we can see that by penalising the vector of SS parameters by squared ℓ_2 norm is possible to smooth the course of the regression line. If we compare it with

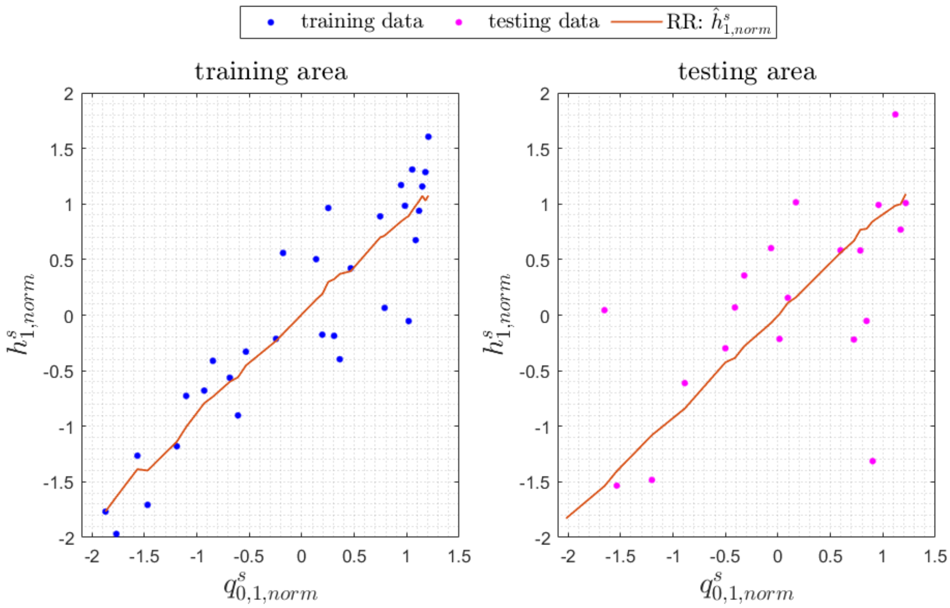

Figure 4.5: Comparison of RR SS performance in training and testing area.

Fig. 4.3, we can say that the linear dependence is much better described by a SS based on RR. The parameter β_1 in this case is also positive. From Table 4.2, we can see that similar to the OLS SS, RMSE for testing data (0.7461) is much bigger than for training data (0.3959).

Another is SS based on LASSO. By solving the objective function from equation 2.21, where the weighting parameter λ has a value $\lambda = \lambda^*_{\text{LASSO}} = 0.01$, we get the vector of SS parameters

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 0.0001 \\ 0.4918 \\ 0 \\ 0 \end{bmatrix}. \tag{4.13}$$

Resulting SS equation looks like

$$\hat{h}^s_1 = 0.0001 + 0.4918 q^s_{0,1}. \tag{4.14}$$

With a suitable weighing parameter $\lambda = 0.01$, two SS parameters are eliminated and we get a smooth linear dependence as is shown by Fig. 4.6. By eliminating two ETM variables ($q^s_{0,2}$, h^s_2), we get the same structure as given by the equation (3.67).

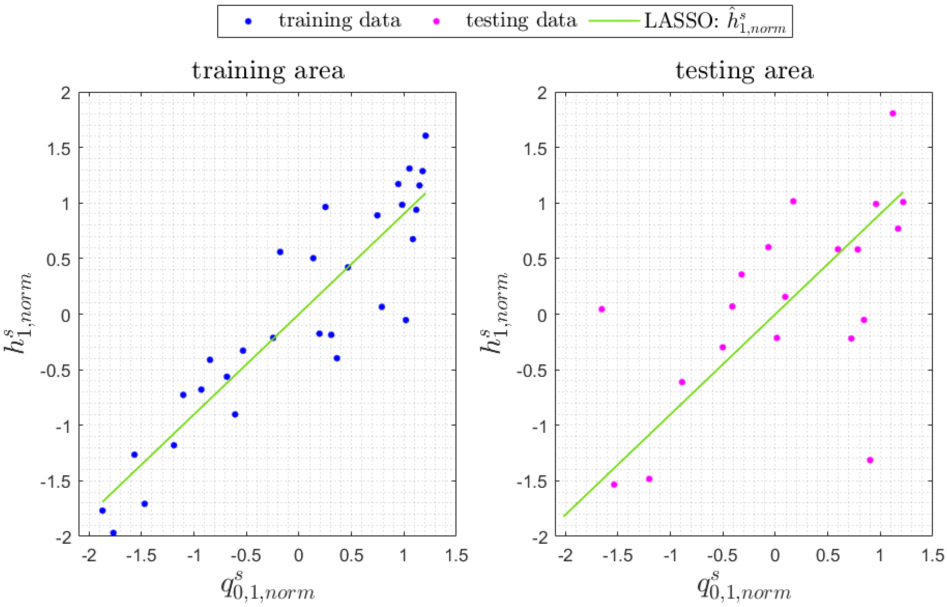

Figure 4.6: Comparison of LASSO SS performance in training and testing area.

Table 4.3: RMSE for LASSO SS in training and testing area.

	training area	testing area
RMSE	0.4018	0.7349

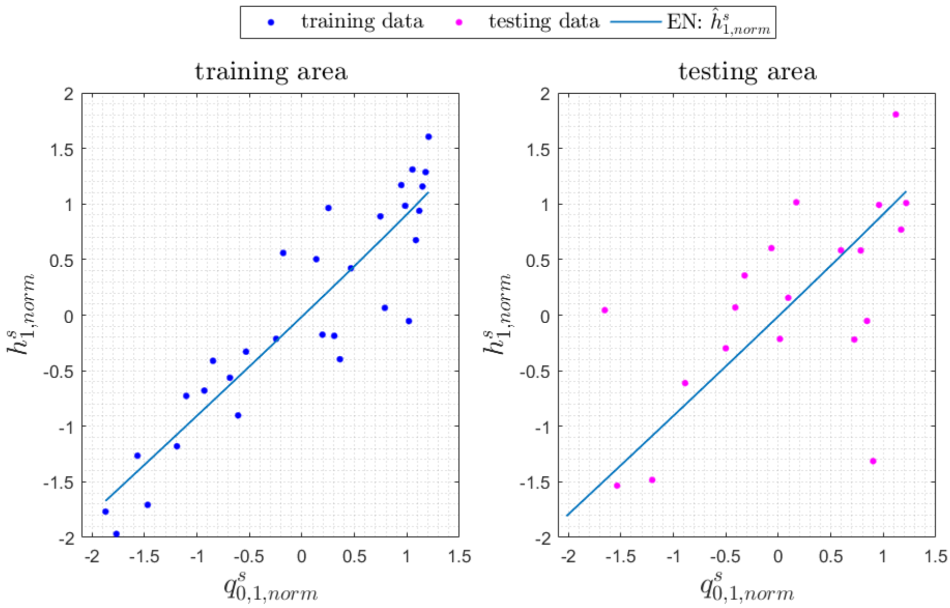

Figure 4.7: Comparison of EN SS performance in training and testing area.

According to this structure, it is expected that β_1 has a positive value, which is also achieved by this method. The difference in RMSE values for the training (0.4018) and testing area (0.7349) is similar to previous SSs (Table 4.3).

The last is SS based on EN. By solving the optimisation problem (2.23), where $\alpha = 0.5$ and $\lambda = \lambda^*_{\mathrm{EN}} = 0.01$, the vector of SS parameters looks as follows

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 0.0001 \\ 0.4494 \\ 0.0031 \\ 0 \end{bmatrix}. \tag{4.15}$$

The SS equation has form

$$h^s_1 = 0.0001 + 0.4494 q^s_{0,1} + 0.0031 q^s_{0,2}. \tag{4.16}$$

With $\alpha = 0.5$, this SS should have a regression line between the RR SS and LASSO SS. Since we were looking for the optimal λ (Fig. 4.4), the EN SS is closer to the more accurate SS from the pair RR SS and LASSO SS. SS based on EN has similar course as SS designed by LASSO. This is the confirmation that LASSO is the best method

Table 4.4: RMSE for EN SS in training and testing area.

	training area	testing area
RMSE	0.4023	0.7366

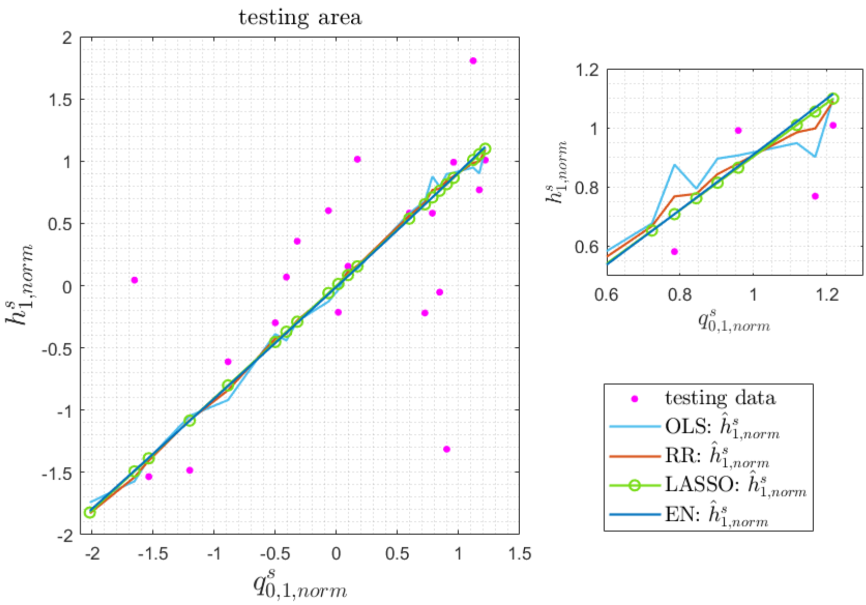

Figure 4.8: SSs comparison in testing area.

for SS design to a system of two tanks in series without interaction. SS based on EN is trying to imitate or even improve the SS based on LASSO. RMSE values of EN SS for training and testing area are given in Table 4.4.

Figure 4.8 shows a comparison of SSs in the testing area. We can say that the best linear dependence is given by SSs proposed by the LASSO and EN methods. If we look at the Table 4.5, we can see that only LASSO SS estimates a structure identical to the equation (3.67), where h_1^s is depended just from $q_{0,1}^s$. The intercept β_0 of all SSs has almost zero value. This is the verification that our SSs are well designed. The accuracy of the estimate can be quantified using RMSE values. According to Table 4.6, LASSO SS has the smallest deviation from the testing data. This also confirms the title of the most accurate SS.

Table 4.5: Parameters of SSs.

	OLS	RR	LASSO	EN
β_0	0.0001	0.0002	0.0001	0.0001
β_1	0.1257	0.4282	0.4918	0.4494
β_2	0.2098	0.0643	0	0.0031
β_3	-0.1825	-0.0591	0	0

Table 4.6: Comparison of RMSE for all SSs in training and testing area.

RMSE	OLS	RR	LASSO	EN
training area	0.3913	0.3959	0.4018	0.4023
testing area	0.7703	0.7461	0.7349	0.7366

4.1.3 Statistical Evaluation of the Software Sensors

The effect of noise in the measured data, which is in our case a white Gaussian noise (expressed by equations (4.3) and (4.4)), cannot be predicted. The randomness of the noise can be neutralised with the statistical evaluation. If a large amount data is collected, data distribution can be approximated. From this point of view, the results based on statistical evaluation are much more reliable.

The first statistical evaluation is the percentage occurrence of ETM variables in the SSs structures. In the previous results, we have seen that LASSO SS is able to find the derived structure from equation (3.67). Using statistics, we can evaluate how many times SSs are able to find this structure which follows from the physical nature. Other possibilities of the structure can also be statistically evaluated. This evaluation can be useful for the future SS design. We can find out the occurrence of ETM variables and possibly not include some of them. This approach can simplify the structure of the SSs and thus facilitate SS design in the future.

Table 4.7: Occurrence of $q^s_{0,1}$ in the structure of SSs.

	$\beta_1 \neq 0,\ \beta_2 = 0,\ \beta_3 = 0$			
	OLS	RR	LASSO	EN
n_{occ}	0	0	476	124
[%]	0	0	47.6	12.4

Table 4.8: Occurrence of $q^s_{0,2}$ in the structure of SSs.

	$\beta_1 = 0,\ \beta_2 \neq 0,\ \beta_3 = 0$			
	OLS	RR	LASSO	EN
n_{occ}	0	0	46	10
[%]	0	0	4.6	1

Table 4.9: Occurrence of h^s_2 in the structure of SSs.

	$\beta_1 = 0,\ \beta_2 = 0,\ \beta_3 \neq 0$			
	OLS	RR	LASSO	EN
n_{occ}	0	0	36	6
[%]	0	0	3.6	0.6

Statistical evaluation was performed for 1000 different randomly generated data sets. Value of n_{occ} represents the occurrences number of a defined structure per 1000 different cases. The Tables 4.7, 4.8 and 4.9 also contain a percentage representation of the defined structure for 1000 different cases. Since the SS based on LASSO has the best results in previous section, the weight of the results is given to LASSO as well. According to Tables 4.7, 4.8 and 4.9, we can see that the probability of occurrence only $q^s_{0,1}$ is much higher (47.6 %) than occurrence of only $q^s_{0,2}$ (4.6 %) or only h^s_2 (3.6 %) in the structure of LASSO SS. To confirm this hypothesis, we statistically calculated how many times out of 1000 cases each SS parameter is equal to zero. In Table 4.10 we can see that β_1 is equal to zero in only 6.8 % of cases. On the other hand, β_2 and β_3 are equal to zero for 71.9 % and 65.2 % of cases. From this we can conclude and confirm the equation (3.67) as ideal structure for SS design.

According to table 4.11, we can conclude that the inlet flow $q^s_{0,1}$ has the greatest influence on the LLH in the first tank h^s_1. The SS corresponding to equation (3.67) will be considered as an ideal SS further in the work.

Table 4.10: Occurrence of zero parameters in the structure of LASSO SS.

LASSO	$\beta_1 = 0$	$\beta_2 = 0$	$\beta_3 = 0$
n_{occ}	68	719	652
[%]	6.8	71.9	65.2

Table 4.11: Percentage dependence of h_1^s from ETM variables.

	LASSO [%]
$h_1^s = f(q_{0,1}^s)$	47.6
$h_1^s = f(q_{0,2}^s)$	4.6
$h_1^s = f(h_2^s)$	3.6
combined dependencies	44.2
$\sum$	100

The following is a statistical evaluation of the SS accuracy. The RMSE values of the SSs was calculated for 1000 different data sets (based on equation (2.15)). To neutralise the effect of noise on the statistical evaluation, we proceeded as follows. We subtract the corresponding RMSE of the ideal SS from each of 1000 RMSE values. Ideal sensor can be expressed from equation (3.67)

$$h_{1,\mathrm{ID}}^s = \frac{q_{0,1}^s}{k_{11}^2} q_{0,1}^s. \tag{4.17}$$

Then the normalised RMSE value is given as

$$\mathrm{RMSE}_{\mathrm{norm}} = \mathrm{RMSE} - \mathrm{RMSE}_{\mathrm{ID}}. \tag{4.18}$$

With normalisation like this, it is very likely that the normalised RMSE has negative value. This is a consequence to the presence of noise in the measured data. The conclusion is that with a more negative value of RMSE, the accuracy of SSs increases.

Statistical evaluation of SSs accuracy is shown by *boxplots*. *Boxplots* can be considered as a graphical interpretation of the normal distribution. Red pluses in *boxplots* charts represent outliers, the blue rectangle represents the interquartile range (50 % of data) and the black dashed lines represent the lower 25 % and upper 25 % from the interquartile range. The median is represented by a red line inside the blue rectangles. This red line represents the highest probability of data occurrence. SSs accuracy are evaluated based on these medians.

If we look at Fig. 4.9, where the SSs are statistically evaluated on the basis of training data, we see that the SS designed by the OLS method has the lowest value. This of course makes sense, since the OLS SS trying to minimise the deviation from the training data as much as possible. The other three SSs, designed by the "shrinkage" methods have higher RMSE values for training data which is also numerically shown in Table 4.12. This is caused by a penalty where the regression line is smoothed and thus causes higher deviations from the training data.

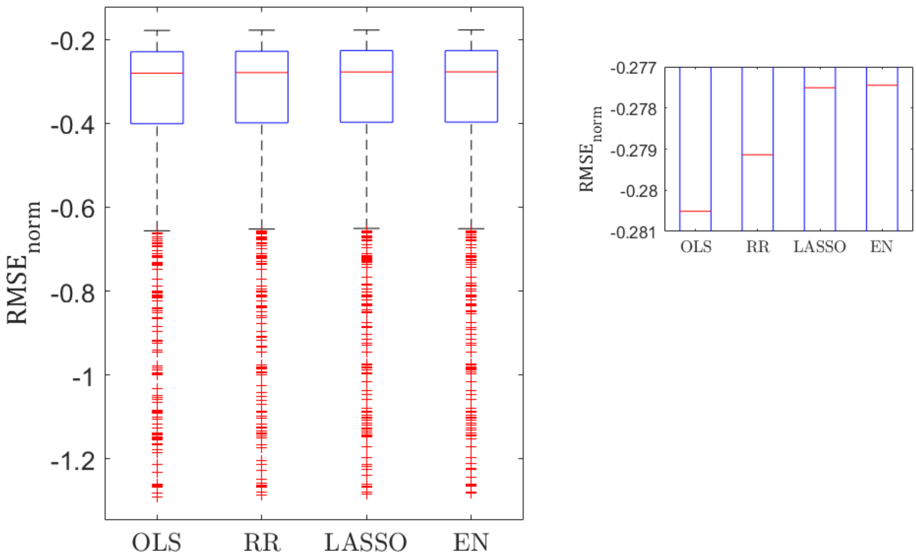

Figure 4.9: Statistical evaluation of SSs accuracy on training data.

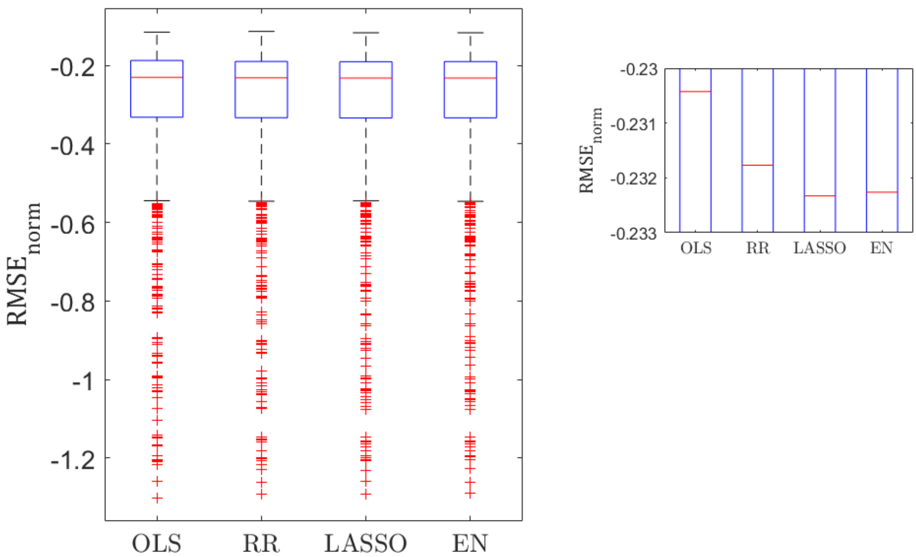

Figure 4.10: Statistical evaluation of SSs accuracy on testing data.

Table 4.12: *Boxplot* medians of SSs for training data.

	LS	RR	LASSO	EN
Median $\times 10^2$	-28.05	-27.91	-27.75	-27.75

Table 4.13: *Boxplot* medians of SSs for testing data.

	LS	RR	LASSO	EN
Median $\times 10^2$	-23.04	-23.18	-23.23	-23.22

What interests us more is how SSs behave in the testing area. In Fig. 4.10 we can see a mirror reflection of the previous study. The statistically most accurate SS in the testing area is the SS designed by the LASSO method and at the same time by the EN method (Table 4.13). The SS which had the statistically smallest accuracy error in the training area (OLS SS) is now statistically the least accurate. This SS tries to imitate the rough data course from the training area. Since this SS works with new testing data, it cause very large deviations. The SS designed by the LASSO method eliminates some parameters and thus smoothens the course of the regression line. This ensures a smaller statistical error of accuracy in the testing area. The SS designed by the RR method in both cases reaches an average performance. The penalisation of this SS reduced SS parameters which smoothes the regression line. SS based on RR is advantageous when we do not want to eliminate any input variable. For SS designed by the EN method, the idea from Chapter 4.1.2 applies. There it was said that by finding a suitable weighting parameter λ, this SS behaves similarly to a better SS from the pair of RR and LASSO SSs.

Finally, the SS designed by the LASSO method is the best choice for obtaining information about steady state value of the LLH h_1^s. In some cases, statistically better results could be reached by EN method with suitable choice of weighing parameters.

4.2 Software Sensors Implementation in the Model of Two Tanks in Series

In this work, SSs are designed to estimate the steady state LLH of the first tank h_1^s. It is a process state variable. To obtain information about states we often used state observers. SSs can also be used for this purpose. This section contains the most

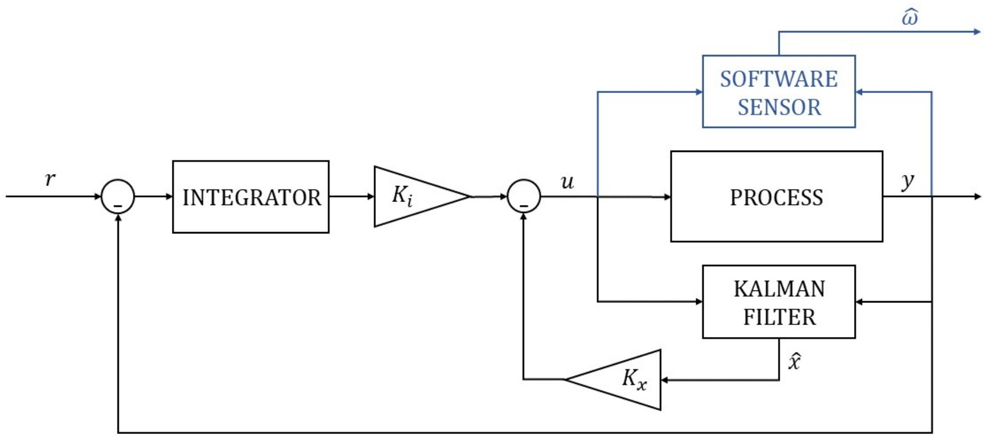

Figure 4.11: Implementation of SS in LQR control scheme.

significant results of estimating the steady state value h_1^s using SSs. The results of the comparison of SSs and KF are also included in this section.

4.2.1 Software Sensors as Steady State Estimators

The new steady states of h_1 can be found basis on SSs designed for the range ± 25 % of the initial value of $h_2^s = 0.3787$ m. Monitoring of the LLH h_1 using SSs can be implemented in a controlled process of two tanks in series (described in Section 3.1.2). The SSs are designed in a similar way as the implementation of KF.

Figure 4.11 shows the implementation of SS in the controlled process. The input u and output y process variables enter the SS. The input variable is the vector of inlet flows $q_{0,1}$ and $q_{0,2}$ for our case. The controlled output variable is h_2. The estimation equation for monitoring of th LLH h_1 looks as follows

$$\hat{h}_1^s = \begin{bmatrix} 1 & \boldsymbol{u} & y \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 1 & q_{0,1} & q_{0,2} & h_2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}. \tag{4.19}$$

The SS parameters for all SSs are given in Table 4.5. The reference increases from the initial value of h_2^s by 20 %.

$$r = 1.2 h_2^s. \tag{4.20}$$

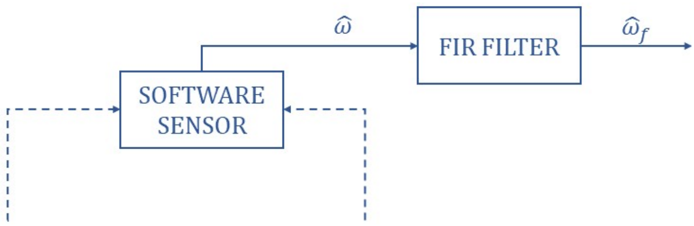

Figure 4.12: Implementation of the FIR filter.

The discrete finite impulse response (FIR) filter is used to smooth the signals from the SSs (Fig. 4.12). It is the moving-average filter. The general form of the discrete FIR filter is [3]

$$F(z) = f_0 + \frac{f_1}{z} + \frac{f_2}{z^2} + \cdots + \frac{f_i}{z^i}, \tag{4.21}$$

where coefficients of the delayed input f_i can vary [3].

The selected filter for our case has the following structure

$$F(z) = 0.1 + \frac{0.1}{z} + \cdots + \frac{0.1}{z^9}. \tag{4.22}$$

In Fig. 4.13 we can observe the efficiency of the FIR filter given by equation (4.22). The resulting course $\hat{h}_1$ (blue lines) for all SSs is smoothed ($\hat{h}_{1,f}$, black lines) using the FIR filter. The pink line represents the simulated result of non-linear process (NL: h_1).

Fig. 4.14 displays an estimate of h_1^s by SSs. Pink line represents simulation of real h_1 course (NL: h_1). SS based on OLS obtain large parameters in this case which causes very large oscillations. This is the reason why we did not use this SS for comparison. The SSs based on “shrinkage” methods performed very well in estimating the steady state of h_1. The smallest difference from the real steady state value (NL: h_1) has LASSO SS. This SS (green line) acquires a very similar course of reality (pink line) as it has the ability to eliminate some parameters. SS based on RR (orange line) has the least accurate h_1^s estimation results in this comparison. This method can only reduce the weight of SS parameters, but it cannot eliminate. This can be observed by the noisy course of the orange line in Fig. 4.14. EN SS is delivered as expected. The output of this SS (blue line) is average from the pair of RR and LASSO SSs.

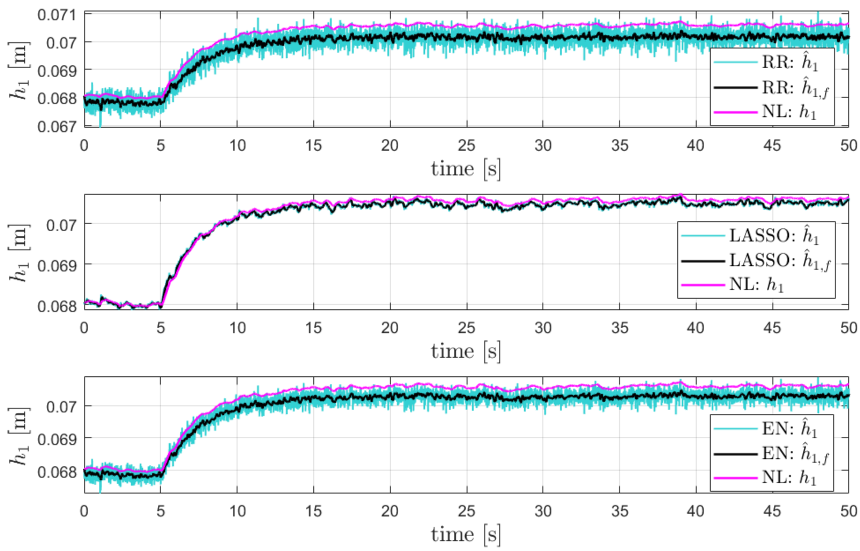

Figure 4.13: Filtering the outputs of SSs using discrete FIR filter.

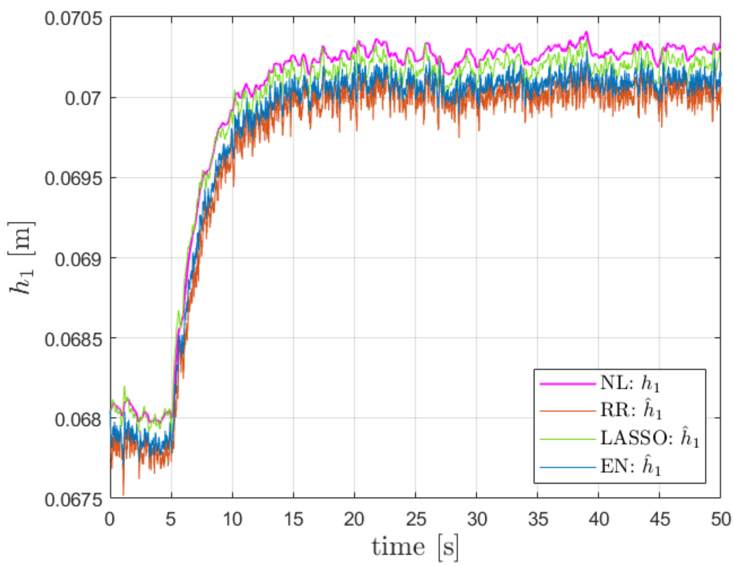

Figure 4.14: Steady state estimation of h_1^s by SSs.

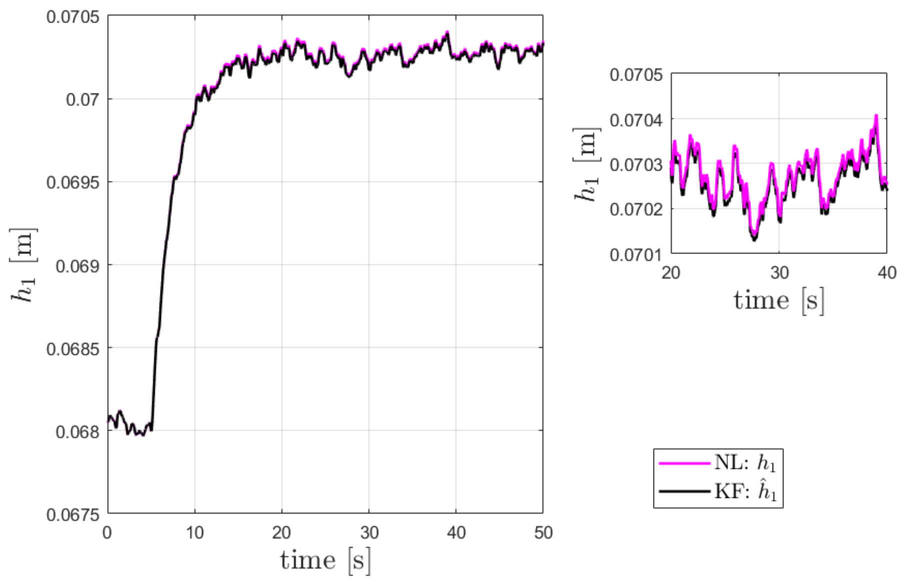

Figure 4.15: Monitoring steady state of h_1 by KF.

4.2.2 Comparison of the Software Sensors with Kalman Filter

The KF is considered to be the best state observer in the process industry. We tried to simulate this efficiency with the SSs.

We deal with the observation of the LLH h_1. With appropriate choice of weighing matrices

$$\boldsymbol{P}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \boldsymbol{Q} = \begin{bmatrix} 10^4 & 0 & 0 \\ 0 & 10^4 & 0 \\ 0 & 0 & 10^4 \end{bmatrix}, \qquad \boldsymbol{R} = 10^4, \tag{4.23}$$

we provide an estimate of the LLH h_1 with KF. Fig. 4.15 shows an almost ideal estimate of the LLH h_1 using KF.

Now we can compare SSs estimation from previous section and estimation using KF. In Fig. 4.16 we can confirm the hypothesis that the KF is the best state observer in process industry. SSs can also be evaluated as sufficiently accurate estimators of the LLH h_1. The estimation error of the least accurate SS (RR SS) does not exceed 1 % from simulated steady value h_1 of the non-linear process (NL: h_1).

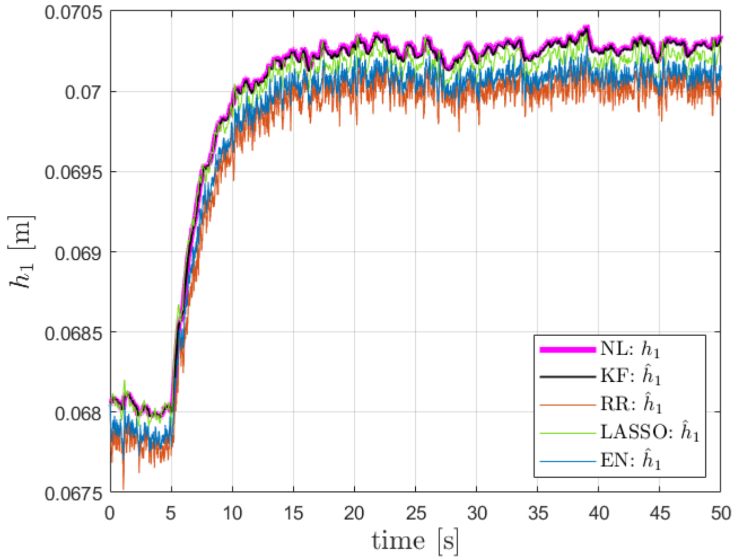

Figure 4.16: Comparison of h_1 level height monitoring using SSs and KF.

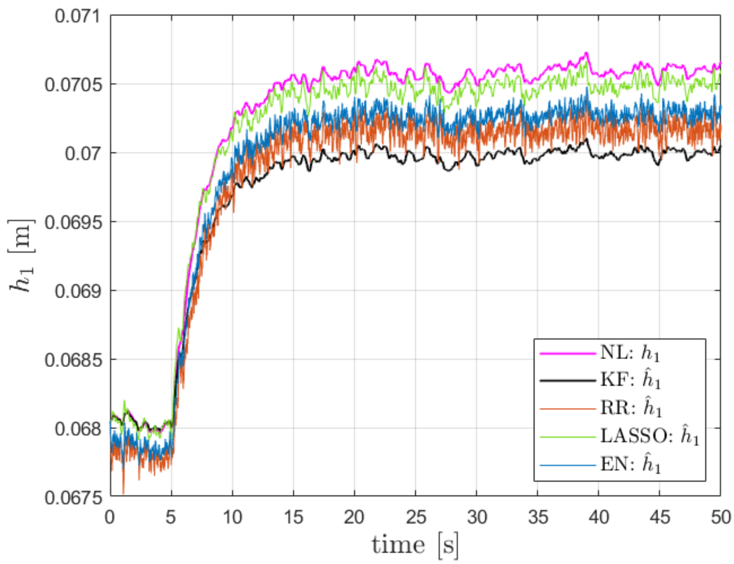

Figure 4.17: Comparison of h_1 level height monitoring using SSs and inaccurate KF.

The disadvantage of KF is that it very much depends on the process (state-space) model. Process parameters may change over time. The state-space model cannot monitor these changes. This can cause variations in the monitoring of DTM variables. Fig. 4.17 shows the change in estimation state h_1 using KF. The same conditions apply as from Section 3.2.1, where matrix $\boldsymbol{A}$ is changed by member a_{11} (increased by 30 %). In Fig. 4.17 we can see that in this case KF has the least accurate estimate of the LLH h_1. We can say that the great advantage of SSs over KF is that it does not depend on the model of the process of interest. A rigorous dynamic model can be difficult and costly to build and keep up to date. From this point of view, the great advantage of SSs is that they can be designed only on the basis of measured data. To maintain the accuracy of the SSs, it is necessary to re-train the SSs on new data. However, this issue is much more practical compared to the identification of rigorous dynamic models in the process industry.

Chapter 5

Discussion

With the SSs, we were able to describe the DTM variable (LLH h_1) using the data of ETM variables ($q_{0,1}$, $q_{0,2}$, h_2). We also had success with implementation of the SSs into the process of two tanks in series, where they have the role of estimating DTM variables. There is always space for improvement. The results can be improved in the following ways.

With SSs proposing, we worked with step responses of the steady state value h_2^s in the range ± 25 %. We could increase this range to improve the LLH estimate at extremely large changes from the initial steady value $h_2^s = 0.0681\ m$. There were also 50 step responses in the mentioned range (± 25 %). The accuracy of the SSs estimate also increases with increasing number of data which they use for training.

In a statistical study we found that $q_{0,2}^s$ and h_2^s had small effect on DTM variable h_1^s. In the future, these variables may not be necessary to propose the SSs for h_1^s estimate. Non-linear SSs could also be designed, as it is a nonlinear process of two tanks in series.

With the EN SS, the procedure was such the weighting parameter α is chosen to be 0.5. We wanted to achieve behaviour between the SSs based on RR and LASSO methods. The λ weighting parameter were obtained by optimisation similarly to the RR and LASSO SSs. The α can also be found by optimisation methods for two optimisation variables. This would provide optimal weighting parameters λ and α, which would theoretically make it possible to defeat the most accurate SS based on LASSO.

Chapter 6

Conclusions

In this master thesis, we focused on the design of SSs based on data from process of two tanks in series without interaction. The work was focused on the theoretical base, in order to compare and obtain information about different methods for SS design. On the basis of the obtained results, general properties of SSs were derived. These properties can be very useful for the design of SSs in the industry. The design of SSs, their comparison, statistical evaluation and implementation into the case study was processed in the MATLAB and Simulink software environments.

The procedure was such that using data from ETM vaiables ($q^s_{0,1}$, $q^s_{0,2}$, h^s_2) we designed SSs to estimate the DTM variable the LLH h^s_1. Based on the expected results, we evaluated the SSs. We found out that the "shrinkage" methods had a very accurate estimate of the DTM variable h^s_1. Using RR, we reduce the influence of some parameters and thus smooth out the dependence of the DTM variable on ETM variables. Using the LASSO method, we can eliminate some parameters and simplify the structure of the SS. The advantage of the EN method is that it contains two penalisation of SS parameters by which we can ensure a compromise between the RR and the LASSO methods. This method also can eliminate some ETM variables from the SS structure.

SSs were also statistically evaluated. First, we evaluated the statistics of the occurrence of ETM variables in the structure of SSs. We found that the highest probability of occurrence has the inlet flow $q^s_{0,1}$. SS, which contained only this ETM variable in his structure was considered as SS with an ideal structure. This conclusion also follows from the physical nature represented by the mass balance of the first tank. Since the data were noisy, the accuracy of the SSs was evaluated statistically using the *boxplot* functions. The highest statistical accuracy was demonstrated by SSs based on LASSO and EN methods. This result confirmed the success of the LASSO method. SS designed by this method was considered the most accurate. EN SS has a very similar statistical performance. The difference between these two SSs was in the structure, where the EN SS was slightly more complex.

At the end, we implemented the proposed SSs into our process of two tanks in series without interaction. The SSs performed the role of state observers and estimated the steady state LLH of the first tank h_1^s. According to the results, we found that OLS SS is not suitable for such an implementation. The parameters of this SS can have very large values, which causes an unstable estimate of the steady-state value of the DTM variable. The other three SSs proposed by "shrinkage" methods had very accurate results of estimating the steady state value of h_1. We compared the estimate of h_1^s using SSs with the estimate of the KF. In a well-identified process, where we obtain an correct model, KF is an invincible state estimator. As soon as a change in the process occurs, the identified model becomes inaccurate. This results in inaccuracies in the estimation using KF. The advantage of the SSs is that they can be easily overtrained on new data and thus ensure the update of estimate.

From the results interpreted above, it follows that the most preferred methods for SS design for our process of two tanks in series is LASSO and EN. This methods can eliminate some ETM variables which has small or no effect on estimating the DTM variable. In this way, the structure of the SS is simplified and the accuracy of the estimate is increased. Due to the requirements, other methods for SS design can also be very useful. The most significant result achieved is the fact that in addition to real sensors, SSs can also replace state observers.

The work can be developed on the optimising described methods for SS design but also on the design of SSs based on new methods. Another direction in which work can continue is the design of SSs in online modules. These SSs would instantly record the measured data and use it directly to describe the DTM variables. SSs could also be used for process control. The controllers will be able to work using states estimated by SSs.

Appendix A

Resumé

Táto práca je zameraná na návrh softvérových senzorov pre procesný priemysel. V súčasnosti sú takmer všetky priemyselné prevádzky vybavené veľkým počtom senzorov. Primárnou úlohou senzorov je poskytovať údaje, pomocou ktorých môžeme proces monitorovať a riadiť. Koncom minulého storočia sa začalo pracovať s veľkým množstvom údajov. Dáta sa merajú a ukladajú vo veľkom množstve v procesnom priemysle. Na základe týchto dát môžeme predpovedať správanie sa procesu v budúcnosti a navrhnúť tak prediktívne modely. Takéto prediktívne modely v priemysle sa nazývajú softvérové senzory (SS). SS môžu byť všeobecne rozdelené na dve triedy, SS na základe dát a SS na základe modelu. Rozsah úloh, ktoré SS vykonávajú, je široký. Primárnou a najdôležitejšou úlohou SS je predikcia procesných premenných. Na základe informácií ľahko merateľných (ĽM) veličín vieme opísať niektoré ťažko merateľné (ŤM) veličiny. Ďalšími dôležitými oblasťami aplikácie SS sú monitorovanie procesov a detekcia chýb, optimalizácia prevádzkových podmienok, zvýšenie výkonu prevádzky, zabezpečenie väčšej úspory energie a menej negatívnych dopadov na životné prostredie atď. Napriek všetkým vyššie uvedeným aplikáciám a výhodám SS, stále existujú nevyriešené otázky týkajúce sa vývoja a údržby SS. Bežným problémom pri zaznamenávaní dát je šum merania, extrémne odchýlky niektorých hodnôt, chýbajúce hodnoty atď. Práca sa skladá zo šiestich kapitol. Prvá kapitola je úvodná. Táto časť vysvetľuje všeobecné vlastnosti, dôvody a výhody používania SS. V tejto časti sú uvedené aj príklady využitia SS v procesnom priemysle, ich úlohy a nedostatky.

Druhá kapitola predstavuje teoretický základ práce. Táto kapitola je rozdelená na dve časti. Prvá časť opisuje rozdelenie SS na základe štruktúry:

- SS na základe dát,
- SS na základe modelu,
- Hybridná štruktúra SS.

Popísané sú ich najdôležitejšie vlastnosti a uvádza sa všeobecný matematický zápis týchto typov SS. Druhá časť tejto kapitoly opisuje metódy pre návrh SS. Štyri metódy pre návrh SS sú:

- metóda najmenších štvorcov (OLS),
- hrebeňová regresia (RR),
- operátor najmenšieho absolútneho zmrštenia a výberu (LASSO),
- elastická sieť (EN).

Pre všetky metódy je uvedená matematické formulácia. Jedná sa o optimalizačné metódy, ktoré získavajú optimálne hodnoty parametrov regresie, respektíve, v našom prípade parametrov senzora. Metóda OLS vyhľadáva také parametre, ktoré zabezpečia najmenšiu odchýlku regresnej priamky od nameraných dát. Zvyšné tri metódy sú rozšírenou verziou tejto metódy. Metóda RR v účelovej funkcii obsahuje penalizáciu ℓ_2 normy parametrov senzora. Takouto penalizáciou vieme úspešne znížiť hodnoty parametrov senzora. Metóda LASSO funguje na podobnom princípe, pričom používa penalizáciu ℓ_1 normy parametrov senzora. Táto metóda dokáže nie len zmenšiť, ale aj vynulovať niektoré parametre. Obe tieto metódy obsahujú váhovací parameter λ pomocou ktorého vieme tieto penalizácie ladiť. Posledná je metóda EN. Táto metóda je kompromisom medzi RR a LASSO. Okrem λ, obsahuje ďalší váhovací parameter α pomocou ktorého vieme váhovať vlastnosti tejto metódy. α nadobúda hodnoty v rozmedzí 0 až 1. Ak je α bližšie k jednotke, EN nadobúda vlastnosti LASSO metódy. V opačnom prípade sa správa ako RR. Okrem teoretickej interpretácii, v tejto časti je zahrnutý aj jednoduchý príklad lineárnej závislosti. Všetky SS sú otestované na príklade $\omega = 3\upsilon_1$ a ich vlastnosti a výhody sú zdôraznené v tejto časti práce.

Tretia kapitola opisuje prípadovú štúdiu na základe ktorej SS budú navrhované. Jedná sa o systém dvoch zásobníkov kvapaliny bez interakcie. V tejto časti je odvodený matematický model tohto procesu na základe preddefinovaných parametrov z tabuľky 3.1. Ďalej sa popisuje stratégia riadenia. Proces je riadený stavovým LQR regulátorom s integračnou činnosťou. V tejto časti je opísaná riaditeľnosť procesu, návrh a výpočet zosilnenia LQR regulátora. Naďalej sú opísané možnosti odhadu výšky hladiny kvapaliny. V tomto procese výška hladiny druhého zásobníka je riaditeľná ĽM veličina. Výška hladiny prvého zásobníka je ŤM veličina, ktorú sa snažíme odhadnúť. Opísané sú dve možnosti odhadovania. Prvá možnosť je pomocou Kalmanovho filtra (KF). Opísaný je spôsob návrhu KF pre systém dvoch zásobníkov kvapaliny. Graficky je zobrazený výsledok odhadu pomocou KF. Následne je uvedený príklad KF, ktorý je navrhovaný na základe neaktualizovaného modelu. Druhou možnosťou odhadovania výšky hladiny prvého zásobníka je pomocou SS. Táto časť obsahuje návrh štruktúry SS pomocou matematického modelu procesu.

Výsledky práce sú opísané v štvrtej kapitole. Táto kapitola pozostáva z dvoch sekcií. Prvá sekcia je zameraná na návrh SS pre prípadovú štúdiu dvoch zásobníkov kvapaliny. Matematicky je vyjadrená závislosť výšky hladiny prvého zásobníka h_1 od ĽM veličín $q^s_{0,1}$, $q^s_{0,2}$ a h^s_2. Na základe tejto informácii je definovaná inicializačná štruktúra SS. Ďalej sa opisuje postup získavania dát pomocou prechodových charakteristík riadenej veličiny h_2 a rozdelenia dát na trénovaciu a testovaciu oblasť. Práca pokračuje trénovaním a testovaním SS. Pre všetky metódy opísané v teoretickej časti sú získané parametre SS, pomocou ktorých boli SS otestované na testovacích dátach. Váhovacie parametre metód RR, LASSO a EN boli optimalizačným postupom získane, okrem váhovacieho parametra α ktorého hodnota bola zvolená (0.5). Keďže namerané dáta obsahovali šum merania, SS boli štatisticky vyhodnotené kvôli presnejším výsledkom. Štatistické vyhodnotenie zahŕňalo získanie informácii o percentuálnom zastúpení každej jednej ĽM veličiny v štruktúre SS. Metóda LASSO dokáže odstrániť niektoré parametre senzora a takto aj niektoré ĽM veličiny zo svojej štruktúry. Na základe tejto metódy sme interpretovali výsledky štatistického vyhodnotenia. Zistili sme, že ĽM veličiny $q^s_{0,2}$ a h^s_2 majú pomerne nízky vplyv na odhadovanú ŤM veličinu h^s_1. Ďalej sme štatistický vyhodnocovali presnosť SS na základe RMSE hodnôt. Výsledky sme vyjadrili pomocou funkcie *boxplot*. Štatistickým vyhodnotením sme zistili, že dva najpresnejšie SS sú na základe metód LASSO A EN. Druhou časťou výsledkov bola implementácia SS do procesu dvoch zásobníkov kvapaliny. V tejto časti je vysvetlený spôsob implementácie SS a výsledky odhadu ustáleného stavu h^s_1 pomocou SS. Pri odhade h^s_1 bol použitý FIR filter, pomocou ktorého výstupné signály SS boli vyhladené. Všetky SS okrem senzora navrhnutého na základe OLS, odhadovali ustálený stav s veľmi veľkou presnosťou. Poslednou časťou práce je porovnanie SS s KF. KF predstavuje najpresnejší odhad stavových veličín v procesnom priemysle. Výsledky monitorovania, ktoré KF poskytuje vhodnou voľbou váhovacích matíc sú takmer ideálne. Pomocou SS sme sa snažili túto presnosť napodobniť. Model na základe ktorého je KF navrhovaný je potrebné často aktualizovať. Identifikácia modelu môže byť zložitá a preto sa v procesnom priemysle často nerobí. Dáta ĽM veličín sú častejšie merané. Toto nám udáva možnosť pretrénovať SS ak je to potrebné. SS sme porovnali s KF navrhnutým na základe neaktualizovaného modelu skúmaného procesu. Výsledkom bolo, že chyba odhadu pomocou KF narástla, pričom odhad SS zostal rovnaký.

Posledné dve kapitoly sú diskusia a záver. Tieto dve kapitoly obsahujú zhrnutie výsledkov práce, možnosti vylepšenia a návrh pokračovania s prácou v budúcnosti.

Bibliography

[1] M. Bakošová and M. Fikar. *Riadenie procesov.* Faculty of Chemical and Food Technology STU, Bratislava, Slovakia, 2nd edition, 2012.

[2] R. Doraiswami and L. Cheded. Robust model-based soft sensor: design and application. *IFAC Proceedings Volumes*, pages 5491–5496, 2014.

[3] G. Ellis. *Control System Design Guide.* Elsevier Academic Press, 525 B Street, Suite 1900, San Diego, California 92101-4495, USA, 3rd edition, 2004.

[4] J. R. Hanni and Venkata S. K. Does the existing liquid level measurement system cater the requirement of future generation? *Measurement*, 2020.

[5] K. Johnson and R. Goody. The Original Michaelis Constant: Translation of the 1913 Michaelis - Menten Paper. *Biochemistry*, page 8264–8269, 2011.

[6] P. Kadlec, B. Gabrys, and S. Strandt. Data-driven Soft Sensors in the process industry. *Computers & Chemical Engineering*, pages 795–814, 2009.

[7] L. Lafférs. Scvrkávacie metódy. *Moderná Aplikovaná Regresia 1*, 2016.

[8] D. M. Lane, D. Scott, M. Hebl, R. Guerra, D. Osherson, and H. Zimmer. *Introduction to Statistics.* University of Houston, Downtown Campus, 1 Main St, Houston, TX 77002, United States, Online Edition.

[9] Y. Liu and M. Xie. Rebooting data-driven soft-sensors in process industries: A review of kernel methods. *Journal of Process Control*, pages 58–73, 2020.

[10] T. Meyer. Root Mean Square Error Compared to, and Contrasted with, Standard Deviation. *Surveying and Land Information Science*, pages 107–108, 2012.

[11] J. Mikleš and M. Fikar. *Process Modelling, Identification, and Control.* Springer Science & Business Media, Berlin, Germany, 1st edition, 2007.

[12] E. Okyere, A. Bousbaine, Poyi G. T., Joseph A. K., and Andrade J. M. LQR controller design for quad-rotor helicopters. *The Journal of Engineering*, pages 4003–4007, 2019.

[13] D. Simon. *Optimal State Estimation.* John Wiley & Sons, Inc., Hoboken, New Jersey, 1st edition, 2006.

[14] J. Walrand and A. Dimakis. *Random Processes in Systems - Lecture Notes.* Department of Electrical Engineering and Computer Sciences University of California, Berkeley CA 94720, 2006.

[15] X. Yan. *Linear Regression Analysis: Theory and Computing.* World Scientific, Kansas City, USA, 2009.

[16] H. Zou and T. Hastie. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society*, pages 301–320, 2004.