Master's thesis

NTNU
Norwegian University of Science and Technology

Emil Oppegård

Optimization and Control of Battery Electrode Drying Process

Master's thesis in Industrial Cybernetics

October 2020

NTNU
Norwegian University of
Science and Technology

Emil Oppegård

Optimization and Control of Battery Electrode Drying Process

Master's thesis in Industrial Cybernetics
October 2020

Norwegian University of Science and Technology

NTNU
Norwegian University of
Science and Technology

MASTER'S THESIS

TTK4900-Engineering Cybernetics

Optimization and Control of Battery Electrode Drying Process

Emil Oppegård
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Supervisor:
Associate Prof. Jon Are Suul
Department of Engineering Cybernetics
Co-Supervisor:
Prof. Odne S. Burheim
Department of Energy and Process Engineering
PhD. Asanthi Jinasena
Department of Energy and Process Engineering

January 26, 2021

NTNU

Preface

This master's thesis is a part of two-year master's degree in Industrial Cybernetics at the Department of Engineering Cybernetics at Norwegian University of Science and Technology (NTNU). The work presented in this report is a collaboration between the Department of Engineering Cybernetics and the Department of Energy and Process Engineering. The work presented in this thesis is a continuation of the previous study done in my preliminary project, *Review and Modeling of Battery Electrode Drying Process*[1].

The last five years have been educational with this thesis being one of many highlights. The thesis gave me the opportunity to apply the combined knowledge that I have acquired from my bachelor's degree in Renewable Energy Engineer and ongoing master's degree in Industrial Cybernetics. The period working on the thesis has been fun, challenging and sometimes frustrating, especially considering the unusual situation due to the Covid-19 pandemic.

First, I would like to thank my supervisors Odne S. Burheim and Jon Are Suul for valuable advice and guidance despite the challenges associated with the unusual situation due to Covid-19. I would also like to extend my gratitude to Asanthi Jinasena for invaluable help and advice in deriving, validating and solving the model. Without your contribution, this thesis would not have been the same. Furthermore, I would like to thank Håvard Karoliussen who first contacted me regarding the project, and have throughout my years as a student at NTNU guided me and given me opportunities within the academic community. Lastly, I would like to thank my partner, Mari R. Olsen, who has supported and motivated me through my last year of my education.

Abstract

The battery industry has experienced a significant increase in the last decades, mainly due to the growth in the electric vehicle (EV) industry and the applications of lithium ion batteries (LIBs). LIBs play a vital role in reducing the CO_2-emissions globally due to its ability to replace power supply based on fossil fuel with electricity. A key factor for maximizing the benefits of LIBs (and other batteries) is to reduce the carbon footprint of the battery itself. The majority of the carbon footprint of LIBs is related to the energy consumption during the drying process in the electrode production. The purpose of this thesis was to examine similar drying processes in various industries and derive a simple, yet accurate model which describes the physical phenomena of thin film drying which later can be used in optimization and control.

A one dimensional model was derived with the parameters; initial solvent content, initial thin film temperature, initial density, air temperature and air velocity. A numerical solution was obtained by applying the finite element method (FEM) and 4th order Runge-Kutta method to the model. The model showed promising behavior compared to similar models presented in different literature. A sensitivity analysis further indicated that air temperature and air velocity have a major impact on the evaporation rate, unlike the initial thin film temperature's influence which was rather small.

The model was linearized about the stationary conditions in an attempt to simplify the optimization and control problem. Although the linearized model shared the same transient behavior as the non-linear model, the dynamics of the linearized model was much faster. Along with large deviation in the stationary thickness, the fast dynamics rendered the linearized model unsuitable for optimization and control purposes. However, the linearized model was used to determine the controllability and observability of the system to assess whether or not a state estimator could be implemented.

With the air temperature and air velocity as control inputs and film temperature and thickness as measurement, the system was observable for small mesh resolutions. Thus, an extended Kalman filter (EKF) was implemented with success for estimating the states based on the measurements.

The process was optimized with respect to energy consumption with the air temperature and air velocity as optimization variables. Due to crack formation and adhesion forces which can be related to the evaporation, the optimization was carried out with upper bounds on the evaporation rate to avoid degeneration of the final performance of the cell. The results clearly indicated reduction in energy and drying time using a multi-stage drying unit compared to single-stage dryer. However, implementation of model predictive control (MPC) was unreasonable due to the computational time of the implemented optimization routine. As a result, simple PI-controllers were designed to track the optimal control input calculated by the optimization routine to optimize the overall process. The designed control systems successfully controlled the overall process to follow the optimal trajectory.

Sammendrag

Batteriindustrien har opplevd en betydelig økning de siste tiårene, hovedsakelig på grunn av veksten i elbilindustrien og anvendelsene av litiumionbatterier. Litiumionbatterier spiller en viktig rolle i å redusere CO_2-utslippene globalt på grunn av dets evne til å erstatte kraftkilder basert på fossilt drivstoff med elektrisitet. En nøkkelfaktor for å maksimere fordelene med litiumionbatterier (og andre batterier) er å redusere karbonavtrykket til selve batteriet. Hovedandelen av karbonavtrykket til litiumionbatterier er relatert til energibehovet under tørkeprosessen i elektrodeproduksjonen. Hensikten med denne oppgaven er å undersøke lignende tørkeprosesser i forskjellige industrier og utlede en enkel, men likevel nøyaktig modell som beskriver de fysiske fenomenene som oppstår under tørking av en tynnfilm og deretter benytte modellen til å optimalisere og kontrollere prosessen.

En endimensjonal modell ble utledet med initialbetingelser for parametrene; løsemiddel fraksjon, tynnfilmstemperatur og tetthet i tillegg til lufttemperatur og lufthastighet. En numerisk løsning ble funnet ved å anvende finite element metoden (FEM) og 4.ordens Runge-Kutta metode på modellen. Modellen viste lovende oppførsel sammenlignet med lignende modeller presentert i ulik litteratur. En sensitivitesanalyse indikerte videre at lufttemperaturen og lufthastigheten hadde stor påvirkning på fordampningshastigheten i motsetning til initialbetingelsen for tynnfilmstemperaturens som hadde liten påvirkning.

Modellen ble linearisert rundt de stasjonære forholdene i et forsøk på å forenkle optimaliseringen og kontrollen. Selv om den lineariserte modellen viste samme transient oppførsel som den ulineære modellen, var dynamikken i den lineariserte modellen mye raskere. I tillegg til et stort avvik i den stasjonære filmtykkelsen, gjorde den raske dynamikken den lineariserte modellen uegnet for optimalisering og kontrollering. Imidlertid ble den lineariserte modellen brukt til å evaluere systemets kontrollerbarhet og observerbarhet for å vurdere om en tilstandsestimator kunne implementeres.

Med lufttemperatur og lufthastighet som pådrag og filmtemperatur og tykkelse som måling, var systemet observerbart for et lavt antall elementer. Som et resultat ble et utvidet Kalman-filter implementert med suksess for å estimere tilstandene basert på målingene.

Prosessen ble optimalisert med hensyn til energiforbruk med lufttemperatur og lufthastighet som optimaliseringsvariabler. Da sprekkdannelse og vedheftningskrefter kan relateres til fordampningen, ble optimaliseringen utført med øvre grenser på fordampningshastigheten for å unngå degenerering av den endelige ytelsen til cellen. Resultatene indikerte tydelig redusert energibruk og tørketid for en flertrinns tørkeenhet sammenlignet med en enkeltrinns tørkeenhet. Implementering av modell prediktiv kontroll var imidlertid ikke hensiktsmessig på grunn av beregningstiden for den implementerte optimaliseringsrutinen. Som et resultat ble enkle PI-regulatorer designet for å følge de optimale pådragene beregnet av optimaliseringsrutinen for å optimalisere den totale prosessen. For det modellerte systemet ble prosessen kontrollert til å følge den optimale banen med suksess.

List of Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

The battery industry has experienced a significant increase in the last decades, mainly due to the growth in the electric vehicle (EV) industry and the applications of lithium ion batteries (LIB). As LIBs allows a large portion of the transportation sector to be shifted from fossil fuel to electricity and replace fossil fueled power generation in synergy with sustainable power generation, LIBs plays a vital role in reducing the CO_2-emissions globally.

A key factor for maximizing the benefits of LIBs (and other batteries) is to reduce the carbon footprint of the battery itself. The majority of the carbon footprint of LIBs is related to the consumed energy from the battery production[4]. Where as most of the energy is consumed during the drying process of the electrodes and maintaining operation conditions in the dry-room.

Although drying processes is one of the oldest processes applied and are found in various industries, most of the research done in the battery industry focus on improving performance by investigating new materials or technologies[5, 6]. The drying process is one of the key-steps in the battery process-chain, not only as reduced energy consumption and increased throughput yield economic benefits, but also due to its significant impact on the electrode structure and the final performance of the cells. A comprehensive understanding of how the drying parameters (such as temperature, air velocity, etc.) affects the performance and drying rate is essential for optimization.

Due to limited amount of adequate literature on the subject, the purpose of this thesis is to investigate the potential of different drying processes applied in industries such as: paper, food and pharmaceutical in order to derive a simple, yet accurate model for the electrode drying process. The model is then used to optimize and control the drying process.

This thesis is structured in a similar fashion as the majority of master's thesis with theory, experimental and results and discussion organized in separate chapters. The structure of chapters will share similar substructure with everything related to formulation of the model being presented first, then followed by topics related to the numerical method. Further, topics about linear system is presented, and lastly topics regarding optimization and control. This organizes the thesis such that results and discussion can be read independent of theory and implementation while at the same time maintaining the same flow at which the work was progressed. As an introduction, LIBs in general along with the process chain of LIBs production, relevant thin film process and models are briefly investigated.

1.1 Why CO_2 Free Battery Production in Norway?

The growing focus on climate-friendly solutions has resulted in a shift in the transport sector from fossil fuels to electricity. Although this transition is at an early stage, it is expected that the global EV market will grow in the years to come. Currently, China and Europe have the highest number of EV. Compared to other countries, Norway has already come a long way in the transition towards fully electric transport sector, as the share of EVs of new cars sold has exceeded 50%[7].

Provided the rest of the world will experience the same development, the demand for LIBs will increase significantly. With today's production mainly located in China, this can pose challenges in terms of security of supply. The Covid-19 pandemic is a good example as several of European EV manufacturers have had to stop production as the supply of LIBs from China disappeared[8]. In addition, according to Zeng and Li and Adams and Shachmurove, China will have increasing need of oil, coal and gas imports in the years to come as a result of rising energy consumption due to technological and economical growth[9, 10].

Establishing battery production in Norway would not only provide increased security of supply for the European market, but also reduce the increasing energy demand in China to some extent. Furthermore, the production of LIBs in China are mainly supplied by coal power, which negatively affects the greenhouse gas (GHG) emissions of LIBs. Since 95% of energy produced in Norway comes from hydropower plants, battery production can be supplied with clean, CO_2 free, energy (without taking into account the European energy mix), thus, reduce the GHG-emissions associated with the production of LIBs[11, 12]. In addition, Norway's relatively low ambient temperature and availability of cooling water will be advantageous as this may indicate lower humidity and energy requirements for industrial processes. Finally, establishing battery factories will also benefit the Norwegian economy, which as of today, is highly dependent on the oil market.

1.2 Process Chain

High throughput production of small LIBs for everyday electronics originated in the 90s[13]. Following recent growth and technological advance, the high throughput-process has been transferred to larger cell production, mainly due to the applications of LIBs in EVs. LIBs used in EVs place high demands on energy and power density, as well as lifetime and safety[13]. To achieve the desired specifications, cells are connected in series or/and parallel and safety systems are implemented to control voltage levels, currents and temperatures. As a result, the cell-to-pack production is a long process-chain of several individual steps. Figure 1.1 illustrates the complete process-chain with the intermediate products as electrode, cell and final package.

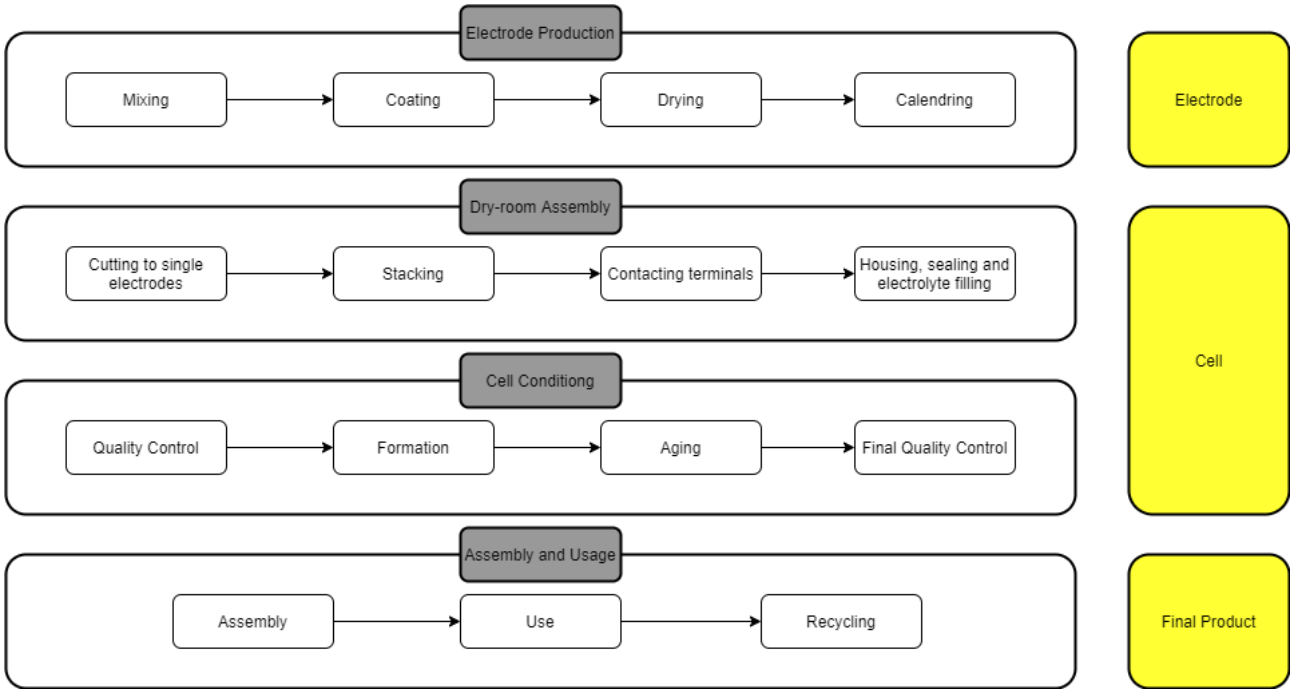


Figure 1.1: Generalized model of the process chain of battery production.

The process begins with the active materials (AMs) being dry- and wet-mixed together with carbon black (CB) additives and polymer binders in a solvent. The composition of the mixture differs from cathode and anode due to the different materials used in the respected electrodes. In EVs production, layered oxides ($LiNiCoMnO_2$ or $LiNiCoAlO_2$) or spinels ($LiMn_2O_4$) are often used as AMs for cathodes, where as for the anode graphite is normally used[13]. Recently it has become more common to add a small amount of silicon with the graphite to increase the energy density [13]. The polymer binders are added to improve mechanical properties such as stability, elasticity and adhesion, while the CB additives improves the electrical contact and reduces the overall impedance. The solvent for the mixtures changes with the different polymers as it needs to have the potential to dissolve the binder, which is the main reason why N-methyl-2-pyrrolidone (NMP) are used for cathode and water for anode. The usage of NMP introduces not only higher production costs compared to water, but also higher energy consumption due to high energy demand in the required NMP-recover process[14]. However, in recent years new polymer binders in aqueous solutions have been studied[15, 16, 17].

When the materials are uniformly mixed the resulting solution is coated onto both sides of the respected current collector (copper for anode, aluminium for cathode) through slot die coating. The coating process, which depends on various aspects of the process (drying time, current collector, etc.), is either done continuously or in patterns. The two side of the current collector is either coated at a separate stages of the liner or simultaneously. Two-sided coating machinery does exists, but in reference with Kwade et. al these are not frequently applied due to structural variations between the two sides[13]. The speed at which the mixture is coated onto the current collector later affects the drying time as higher speed imposes longer drying times[6].

The drying process is one of the major contributors to the high energy consumption in battery production due to the large amount of heat needed to evaporate the solvent. The heat transfer is mainly done by hot air being blow over the wet web coating (convection), but in recent years this has been combined with infrared radiation (IR) and separate drying stages to reduce the drying time and energy consumption[13]. The drying process also have a profound impact on the end performance of the battery, e.g drying temperature and air speed do affect the adhesion strength, resistance and crackling development[6].

With the coated current collector dried to a desired moisture content, it is compressed in the calendering process to a desired thickness. This process does not only reduces the thickness of the electrode, but also improves the energy density as well as conductivity and adhesion. The electrode film is later brought through a dry-room where it is first sliced to the desired dimensions. The operation conditions within the dry-room is carefully controlled to keep the moisture content in the air as low as possible. Thus, preventing damage caused by corrosion which can, among other things, lead to reduced lifetime. The sliced electrodes are then arranged with separators to form cells. There are three different method for arranging the electrodes and separators; winding, stacking and Z-folding. All three methods have their advantages and disadvantages regarding efficiency, mechanical load and energy density. The stacked electrodes and separators are then placed in housing and filled with electrolyte before the cells are being closed.

Before the cells are arranged into modules and later packs, the cells goes through a quality control followed by formation, aging and a final quality control. After the battery packs are assembled with the necessary monitor and cooling system, it is ready for use.

1.3 Industrial Thin Film Drying Process and Relevant Models

Drying is one of the oldest unit operation and are applied in various industries[18]. Although the processes are different for the distinct industries, they do operate on the same principal of heat and mass transfer. A common group of drying processes utilizes thin films where the wet material is formed as a thin film and dried continuously along an operating line. In this section thin film drying processes from different industries that resembles the one used in LIBs production is examined to investigate whether there are certain aspects that can be incorporated into a model for electrode-drying.

1.3.1 Food Industry

Drying processes are widely adopted in the food industry due to various reasons. Mainly, drying processes are used to preserve and extend the shelf life of the product, but also to achieve a desirable shape or structure. With vast quantities of different products, numerous different drying methods have been used depending on which methods are most appropriate. This depends on the quantity, size and shape of the product. Drying processes involving food products in the form of thin layers, shown in figure 1.2, are similar to the methods used in battery production. The product is dried in convection ovens where hot air heats and transports moisture away from the product. When drying food, physical and biochemical transformations often occur. Such transformations changes the characteristic of the drying process, which makes it very

complex. Erbay and Icier presents a detailed model based on convection ovens, while 22 different models for thin-layer drying processes are reviewed in the studies conducted by Onwude et. al[5, 18]. Furthermore, a mathematical model for drying was developed by Aboltins et. al in which IR was used as a heat source[19]. In addition to better preservation of nutrients, Aboltins et. al mentions that the drying process can be made more efficient by using radiation of a certain wavelength that is absorbed only by the water in the product[19].

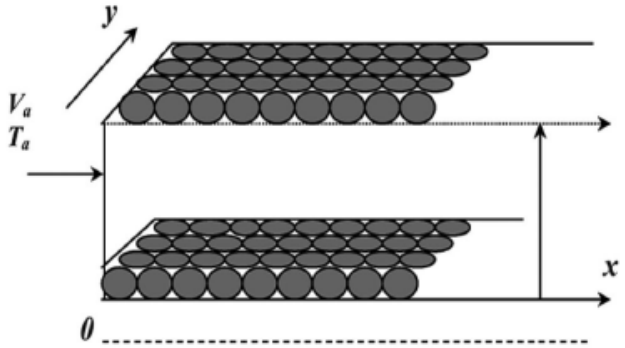
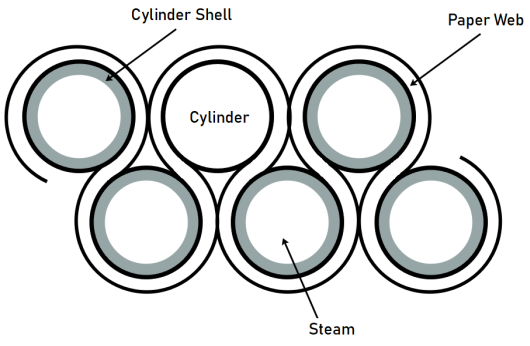


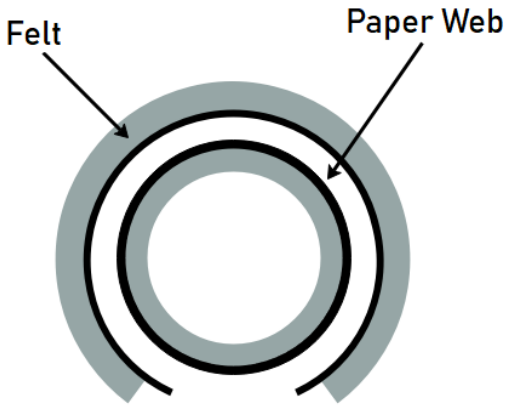
Figure 1.2: Illustration of the thin layer drying process[2].

1.3.2 Paper Industry

A common drying system applied in the paper industry is drying by heating-cylinders in series. Such a system typically consists of 20-120 cylinders with a diameter of 0.75-2 meters filled with steam[20]. Before entering the drying process, the paper web is coated onto a wire and wet-pressed to remove some of the moisture which results in the paper sheet entering the drying section with a moisture content of 55-60%[20]. The paper web is then thread around each cylinder and is heated by the steam through conduction as shown in figure 1.3a. The process is divided into several phases with different number of cylinders and temperature, with and without felts and with a final phase without cylinders so that the moisture can evaporate on both sides of the paper web. The process is controlled by the steam pressure, however, it is common to restrict the controllers to the last groups. The purpose of the felts (which are illustrated in figure 1.3b) is to hold the paper web in close contact to the drying-cylinder, thus further increasing the heat transfer to the web. It also prevents the paper from shrinking and deforming. Heo et. al presents a comprehensive model based on heat transfer through conduction using heating cylinders[20].



(a) Schematic for heating-cylinders.



(b) Heating-cylinder with felt.

Figure 1.3: Illustration of the drying process in the paper industry with heating-cylinders.

1.3.3 Pharmaceutical Industry

In Active Pharaceutical Ingridient (API) manufacture, drying processes are applied to reduce the moisture content to an acceptable level. Due to relative long process time, the drying process is often the bottle-neck in API production[21]. In addition, the drying process may have an impact on the properties of the product, such as API uniformity, which later affects unit operation downstream. API products formed as a thin film is more frequently applied due to enhanced properties, such as eliminating side effects and reduce dose frequency[22]. Thin film drier operation used in pharmaceutical production resembles the one found in electrode production as the slurry is coated onto a moving liner. The slurry then moves with the liner through drying compartment were the moisture is evaporated and carried away. Figure 1.4 illustrated the drying process used in pharmaceutical production. Mesbah et. al presents a mathematically model which describes the drying process and is later used in model predictive control (MPC)[23].

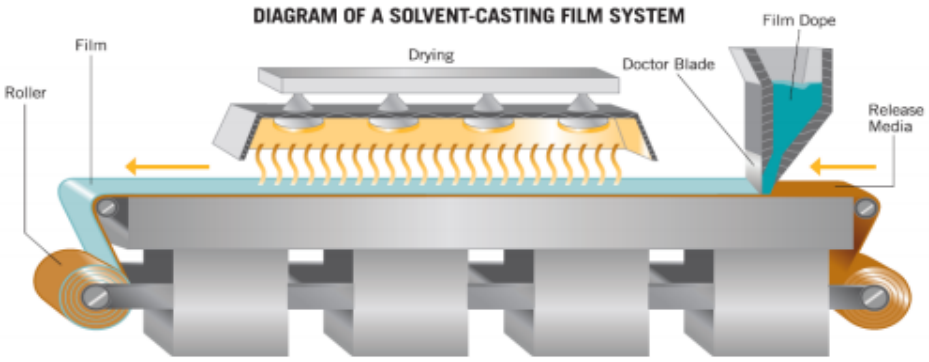


Figure 1.4: Illustration of a drying process in the pharmaceutical industry[3].

Chapter 2

Theory

The purpose of this thesis is to derive a simple model for a drying process which can later be used for optimization and control. As a consequence, theory from different fields will be applied; thermo- and fluid dynamics for the model derivation, numerical- integration and methods, linear system theory etc. To give an overview, the relevant theory applied will be presented in this chapter. The chapter is structured with the model derivation being presented first, then the numerical methods applied for solving the problem is presented in advance of the linear system theory, and lastly, the theory relevant for optimization and control.

2.1 Drying Process & Model Derivation

The drying process were briefly described in section 1.2 as one of the key steps in the process chain in battery production. As the process have a profound impact on the properties of the electrodes, it is important to understand how drying parameters such as temperature, velocities and moisture content affects the performance of the product as well as the drying time.

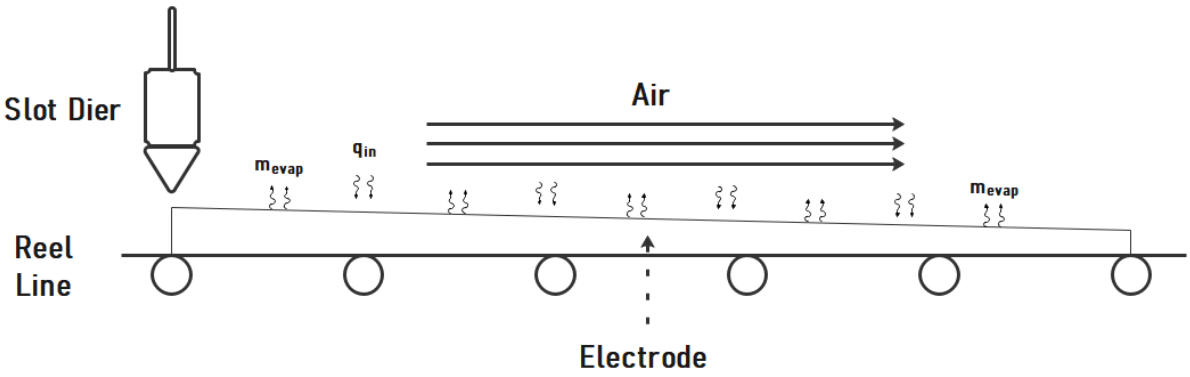


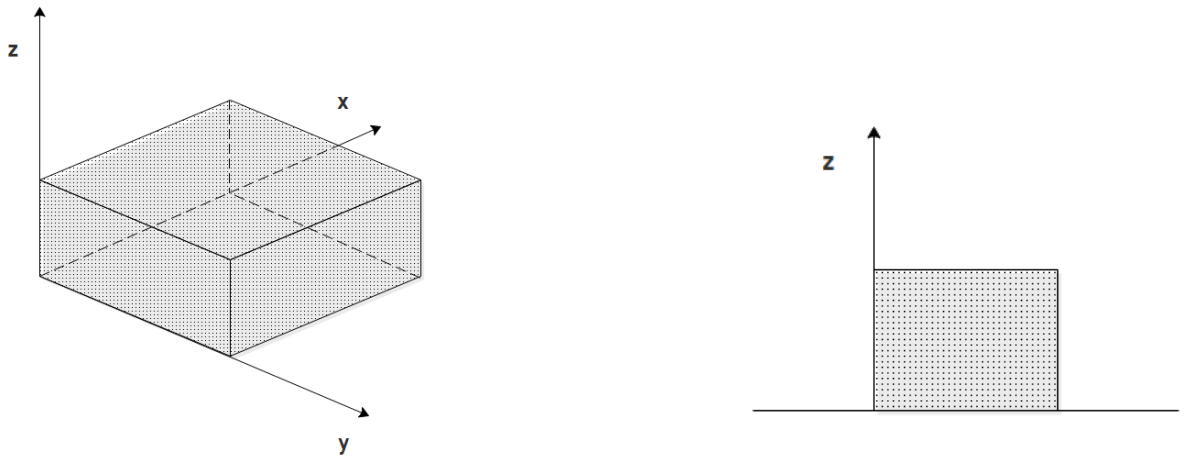
Figure 2.1: Illustration of the drying process.

The drying process can be illustrated as shown in figure 2.1 with the drying compartment consisting of a slot die, liner and fans which blow air along the surfaces of the thin film. The compartment is often divided into several section, each with their distinct fans, air velocities and temperatures. However, for simplistic reasons the drying process is viewed as a drying process with consisting of one compartment. As the thin film enters the drying compartment the solids are fully suspended in a pool of solvent. At this moment, the evaporation is only a surfaces phenomena. However, as the thin film shrinks due to evaporation, the solids will eventually form a porous media. At first, the porous media is completely saturated with solvent in the liquid phase. Further on, the thin film no longer shrinks and evaporation is no longer only a surface phenomenon. At this stage, the rate of evaporation is highly complex due to interplay of physical phenomenons within the thin film structure. Susarla et al. divides the drying process into two stages; the first stage includes evaporation as a surfaces phenomenon and shrinkage of the thin film, and the second stage with no shrinkage

and evaporation rate no longer being limited to the surface[24]. Consequently, the two distinct stages requires different mathematically descriptions. Therefore, this thesis will focus on the deriving a mathematically model for the first stage of the drying process. Given that the drying process had a significant impact on performance due to moisture content, adhesion and crack formation, the model has to be able to measure the performance of the cell. Compared to the moisture content, which can be expressed by weight or mole fractions, adhesion forces and crack formation are not easily expressed mathematically. However, studies conducted by Kumberg et al. reveals that adhesion forces and crack formation can be related to the rate of evaporation[25]. Thus, final moisture content and evaporation rate will be a measure of electrode performance in the model.

2.1.1 Model Derivation

In order to accurately express the physical phenomenon of drying mathematically, a framework has to be established. By considering the Cartesian coordinate system, as shown in figure 2.2a, with *x*, *y* and *z* representing the width, length and height of the film, respectively, the directional fluxes can be expressed in terms of the variables *x*, *y* and *z*. Furthermore, by assuming that the slurry and temperature distribution is uniform in the *x*-direction, the concentration and temperature gradients in the *x*-direction equals zero. These are well established assumptions as the width of the film is much greater than the thickness as well as the solution is uniformly mixed prior to being coated onto the current collector[23, 24, 26]. Movement within the film in the *y*-direction is neglected as these fluxes are assumed to be insignificant. This assumption is more applicable for slurries with lower weight fraction due to higher viscosity, thus higher resistance for movement. In addition, as the slurry is placed upon a liner with a given reel velocity, the *y*-position can be expressed by the drying time, *t*. Thus, limiting the model to only consider fluxes in the *z*-direction, thereby reducing the model from 3D to 1D, as illustrated in figure 2.2b. Resulting in a simpler model without reducing the accuracy of the model significantly.



(a) Cartesian coordinate system in 3-D.

(b) 1-D coordinate system.

Figure 2.2: Illustration of the different coordinate systems.

For the first stage of the drying process solvent transportation within the thin film are governed by diffusion, which is a term used for the movement of particles caused by concentration gradients within a region. As solvent evaporates at the surfaces, solvent particles at lower sections of the thin film will start to diffuse towards the surface to equalize the concentration level. Mathematically this can be expressed by the one dimensional diffusion equation[27]:

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial z}\left(D(C,T)\frac{\partial C}{\partial z}\right) \tag{2.1}$$

where C represents the weight fraction of the solvent (concentration), t (s) is the time, z (m) is the height and D ($\mathrm{m^2\,s^{-1}}$) is the diffusion coefficient. The diffusion coefficient is highly dependent on the concentration and temperature. There are several ways to approximate the diffusion coefficient, e.g the Stokes–Einstein equation. However, Mesbah et al. expresses the the diffusion coefficient as a function of temperature and polymer volume fraction[23]:

$$D = \mathrm{D_0} \cdot \left(\frac{1-\phi}{1+\phi}\right)^{\gamma} \cdot e^{(\frac{-\mathrm{E}}{\mathrm{RT}})} \tag{2.2}$$

$\mathrm{D_0}$ ($\mathrm{m^2\,s^{-1}}$) refers to the mutual diffusion coefficient and varies with different mixtures, ϕ is the polymer volume fraction, E ($\mathrm{J\,kmol^{-1}}$) is the activation energy and γ is a constant.

Equation (2.1) expresses the solvent mass transportation within the thin film. However, this does not apply at the boundaries of the thin film. Since the model is limited to one dimension, the only boundaries are at the top and bottom of the thin film as illustrated in figure 2.3. The liner itself is assumed to be impenetrable, thus, there is no loss of solvent through the bottom boundary. Mathematically this equivalent to applying Neumann boundary condition:

$$\begin{aligned} z = 0 : \\ \frac{\partial C}{\partial z} = 0 \end{aligned} \tag{2.3}$$

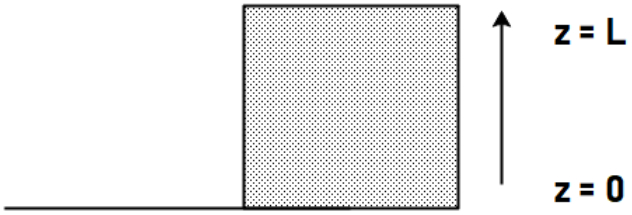


Figure 2.3: Illustration of the boundaries at $z = 0$ and $z = L$.

For the surface boundary one has to consider that the thin film is shrinking due to the evaporation. As a result, the boundary condition at the surface is not fixed at given height, z, because the boundary will move with the diminishing thin film thickness. Mesbah et al. expresses for the boundary condition as[23]:

$$\begin{aligned} z = L(t) : \\ D\frac{\partial C}{\partial z} + C\frac{\mathrm{d}L}{\mathrm{d}t} = -\frac{\mathrm{k_m M_s}}{\rho_\mathrm{s} \mathrm{R}} \cdot \left(\frac{P_s}{T} - \frac{\mathrm{P_a}}{\mathrm{T_a}} \right) \end{aligned} \tag{2.4}$$

Where, L (m) is the varying thickness of the film, $\mathrm{k_m}$ $(\mathrm{m\,s^{-1}})$ is the mass transfer coefficient, $\mathrm{M_s}$ $(\mathrm{kg\,kmol^{-1}})$ is the molecular weight of the solvent, ρ_s $(\mathrm{kg\,m^{-3}})$ is the density of the solvent, R $(\mathrm{J\,kmol^{-1}\,K^{-1}})$ is the universal gas constant, P_s (Pa) is the equilibrium partial pressure of the solvent at the thin film-air interface, T (K) is the varying film temperature, $\mathrm{P_a}$ (Pa) is the vapor pressure in the air and $\mathrm{T_a}$ (K) is the temperature of the air. The vapor pressure in the heated air, as well as the temperature of the air is considered to be constant throughout the drying process and are calculated using the Antoine equation:

$$\mathrm{P_a} = \varphi \cdot 133.332 \cdot 10^{\left(7.54826 - \frac{1979.68}{222.2 + (\mathrm{T_a} - 273.15)}\right)} \tag{2.5}$$

where φ is the relative humidity of the air and the constants in the equation is from [28]. In contrast to the vapor pressure, P_s varies with both solvent concentration and temperature. The theory of polymer solution are highly advanced and will not be studied in this thesis. However, Khansary expresses the solvent vapor pressure using the Flory Huggins theory as[29]:

$$P_s = \mathrm{P_0} \cdot e^{\left(\phi_b^2 \mathcal{X} + ln(\phi_a) + \left(1 - \frac{v_a}{v_b}\right)\phi_b\right)} \tag{2.6}$$

where $\mathrm{P_0}$ is the vapor pressure of the pure solvent and can be calculated using equation (2.5) with $\varphi = 1$ and replacing $\mathrm{T_a}$ with T. ϕ_i represents the volume fraction of component i, v_i is the number of molecules of component i and $\mathcal{X}$ is the Flory Huggins interaction parameter.

It is assumed that the density of the film is solely a function of moisture content and that the specific densities of the solvent and solids remains constant throughout the process. However, these assumptions are not accurate as the density depends on both pressure and temperature. These assumptions are made due to the inherent complexity of expressing the density as a function of porosity as the solid can create various of structures as the solvent evaporates. The density is assumed to follow the linear relation as a function of the solvent volume fraction, Φ:

$$\rho_f = \rho_\mathrm{s} \cdot \Phi + (1 - \Phi) \cdot \rho_\mathrm{solid} \tag{2.7}$$

where $\rho_{\mathrm{s}}(\mathrm{kg\,m^{-3}})$ is the density of the solvent and $\rho_{\mathrm{solid}}(\mathrm{kg\,m^{-3}})$ is the combined density of the solids in the thin film. Due to the changes in the combined film density, an expression for the rate of change of the density with respect to time is necessary (the reason will become obvious when the an expression for the temperature is derived). The solvent volume fraction is a function of the solvent concentration, which can be expressed as:

$$\Phi = \frac{C \cdot \rho_s}{C \cdot (\rho_s - \rho_{solid}) + \rho_{solid}}$$

Thus, the derivative of the thin film density with respect to time is proportional to the derivative of the derivative of the solvent concentration:

$$\frac{d\rho_f}{dt} = \frac{\rho_s \rho_{solid} \cdot (\rho_s - \rho_{solid})}{(C \cdot (\rho_s - \rho_{solid}) + \rho_{solid})^2} \cdot \dot{C} \tag{2.8}$$

The derivative of the solvent concentration is left undefined for now as it is dependent on the method chosen for the numerical approach.

To obtain an expression for the varying thickness of the film, we apply mass balance to the thin film. Along with solid densities of the respective components and the fact that there are no transportation of mass into the thin film such that the mass balance is solely a function of the evaporation that occurs at the thin-film-air interface, the mass balance can be expressed as follows:

$$\begin{gathered}\frac{d}{dt}(\rho_f V_f) = -\dot{m}A \\ \frac{d}{dt}(\rho_f AL) = \frac{dL}{dt}\rho_f A + \frac{d\rho_f}{dt}LA = -\dot{m}A \\ \frac{dL}{dt} = -\frac{\dot{m}}{\rho_f} - \frac{L}{\rho_f}\frac{d\rho_f}{dt}\end{gathered}$$

where $\dot{m}$ represents the mass rate out of the thin film due to evaporation. Thus, the thickness reduction of the thin film can be expressed as:

$$\frac{\mathrm{d}L}{\mathrm{d}t} = -\frac{\mathrm{k_m M_s}}{\rho_{\mathrm{f}} \mathrm{R}} \cdot \left(\frac{P_s}{T} - \frac{\mathrm{P_a}}{\mathrm{T_a}}\right) - \frac{L}{\rho_f}\frac{d\rho_f}{dt} \tag{2.9}$$

An expression for the thin film temperature is found by applying energy balance to the thin film. The heat transported out of the thin film equals the amount of heat needed to evaporate the solvent. The heat transferred into the thin film is expressed by heat transportation either by ordinary convection or radiation. To simplify the energy balance, the thickness of the film is assumed to be so small that the temperature development within the thin film is so fast that it can be considered to be uniform in the z-direction. Furthermore, the curvature of the thin film is so small that the contributions from surface tension and shear stress gradients are neglected. Thus, the energy balance can be expressed by the following formula:

$$\begin{gathered}\frac{d}{dt}(\rho_f C_p ALT) = -\dot{m}Ah_f g + \dot{Q}_{in}A \\ \rho_{\mathrm{f}}\mathrm{C_p}\mathrm{A}L\frac{\mathrm{d}T}{\mathrm{d}t} + \rho_{\mathrm{f}}\mathrm{C_p}\mathrm{A}T\frac{\mathrm{d}L}{\mathrm{d}t} + \mathrm{C_p}\mathrm{ATL}\frac{\mathrm{d}\rho_{\mathrm{f}}}{\mathrm{d}t} = -\frac{\mathrm{k_m M_s A}h_{fg}}{\mathrm{R}} \cdot \left(\frac{P_s}{T} - \frac{\mathrm{P_a}}{\mathrm{T_a}}\right) + \dot{q}_{in}\mathrm{A}\end{gathered}$$

Dividing by $\rho_{\mathrm{f}}\mathrm{C_p}\mathrm{AL}$ on both sides yields:

$$\frac{\mathrm{dT}}{\mathrm{dt}} = -\frac{\mathrm{T}}{\mathrm{L}}\frac{\mathrm{dL}}{\mathrm{dt}} - \frac{\mathrm{T}}{\rho_{\mathrm{f}}}\frac{\mathrm{d}\rho_{\mathrm{f}}}{\mathrm{dt}} - \frac{\mathrm{k_m M_s}h_{fg}}{\rho_{\mathrm{f}}\mathrm{C_p R}} \cdot \left(\frac{P_s}{T} - \frac{\mathrm{P_a}}{\mathrm{T_a}}\right) + \frac{\dot{q}_{in}}{\rho_{\mathrm{f}}\mathrm{C_p}} \tag{2.10}$$

Here $h_{fg}(\mathrm{J\,kg^{-1}})$ represents the latent heat of evaporation and $\mathrm{C_p}(\mathrm{J\,kg^{-1}\,K^{-1}})$ is the specific heat coefficient. h_{fg} is a function of the temperature and is given by:

$$h_{fg} = 6.991 \cdot T^2 - 6193 \cdot T + 1.848 \cdot 10^6 \tag{2.11}$$

Equation (2.11) was found using Matlab's "Curve fitting" tool with the values from [30] to form a second degree polynomial with $R^2 = 0.995$. The values from [30] is given in $\mathrm{J\,mol^{-1}}$ and are converted to $\mathrm{J\,kg^{-1}}$ in advance to get the correct units. $\mathrm{C_p}$ is assumed to be constant as it is a weak function of the temperature within the temperature range[23]. $\dot{\mathrm{q}}_{\mathrm{in}}$ $(\mathrm{W\,m^{-2}})$ represents the heat transfer into the thin film from the heat source. This can either be heat transferred due to

convection, radiation or a combination of both depending on which configuration used for the distinct drying processes. Heat transfer due to convection can be expressed as:

$$\dot{q}_{in} = k_c(\mathrm{T_a} - T) \tag{2.12}$$

where k_c is the heat transfer coefficient. Susarla et. al expresses the heat transfer coefficient as a function of the air velocity[24]:

$$k_c = 0.037 \cdot V_a^{0.8} \left(\frac{\mu_a}{\rho_a}\right)^{-0.8} Pr^{1/3} L^{-0.2} \tag{2.13}$$

Here, V_a is the velocity of the air, μ_a the dynamic viscosity of the air, Pr is the Prandtl number and L is the characteristic length. The total energy that is needed for the drying process can then be expressed as:

$$Q = \int_0^{t_f} \left(\dot{q}_{in} + \dot{m}_{evap} \cdot h_{fg}\right)\, dt \tag{2.14}$$

To finalize the model, initial conditions are needed to obtain an exact solution for the simulations. The initial conditions are given by:

$$\begin{aligned} t &= 0: \\ C(0,z) &= C_0 \\ T(0) &= T_0 \\ L(0) &= L_0 \\ \rho_f(0) &= \rho_0 \end{aligned} \tag{2.15}$$

This completes the drying model with parameters as air temperature, T_a, initial solvent weight fraction, C_0, initial thickness, L_0, initial film temperature, T_0, and wind velocity,V_a.

2.1.2 Dimensionless Model

A common approach when dealing with generalized system is to convert the system to only include dimensionless variables. The major advantage of implementing dimensionless variables is that it immobilizes the otherwise moving boundary condition at $\tilde{z} = 1$. To start off, the dimensionless variables are defined as:

$$\tilde{t} = \frac{t - \mathrm{t_r}}{\mathrm{t_s}}, \quad \tilde{\rho_f} = \frac{\rho_f - \rho_{\mathrm{f,r}}}{\rho_0}, \quad \tilde{L} = \frac{L - \mathrm{L_r}}{\mathrm{L_0}}, \quad \tilde{z} = \frac{z - \mathrm{z_r}}{\mathrm{L}}, \quad \tilde{T} = \frac{T - \mathrm{T_r}}{\mathrm{T_0}}, \quad \tilde{C} = \frac{C - \mathrm{C_r}}{\mathrm{C_0}}, \quad \tilde{D} = \frac{D - \mathrm{D_r}}{\mathrm{D_0}}$$

By setting all the reference values, $(\cdot)_r$, to equal zero, the variables can be expressed in terms of dimensionless variables:

$$t = \tilde{t}\mathrm{t_s} \tag{2.16a}$$

$$\rho_f = \tilde{\rho_f}\rho_0 \tag{2.16b}$$

$$L = \tilde{L}L_0 \tag{2.16c}$$

$$z = \tilde{z}L = \tilde{z}\tilde{L}L_0 \tag{2.16d}$$

$$T = \tilde{T}\mathrm{T_0} \tag{2.16e}$$

$$C = \tilde{C}\mathrm{C_0} \tag{2.16f}$$

$$D = \tilde{D}\mathrm{D_0} \tag{2.16g}$$

By substituting equation (2.16a) - equation (2.16g) into the continuous model from the previous section, we obtain the dimensionless model:

$$\frac{\mathrm{d}\rho_f}{\mathrm{d}\tilde{t}} = \frac{C_0 \cdot \rho_s \rho_{solid} \cdot (\rho_s - \rho_{solid})}{\rho_0 \cdot (\tilde{C}_N \cdot C_0 \cdot (\rho_s - \rho_{solid}) + \rho_{solid})^2} \cdot \dot{\tilde{C}}_N \tag{2.17a}$$

$$\frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} = -\frac{\mathrm{k_m M_s t_s}}{\tilde{\rho_\mathrm{f}}\rho_0 \mathrm{R L_0}} \cdot \left(\frac{P_s}{\tilde{T}\mathrm{T_0}} - \frac{\mathrm{P_a}}{\mathrm{T_a}} \right) - \frac{\tilde{L}}{\tilde{\rho_f}} \frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} \tag{2.17b}$$

$$\frac{\mathrm{d}\tilde{T}}{\mathrm{d}\tilde{t}} = -\frac{\mathrm{k_m M_s t_s} h_{fg}}{\rho_0 \mathrm{C_p R}\tilde{L} z_0} \cdot \left(\frac{P_s}{\tilde{T}{\mathrm{T_0}}^2} - \frac{\mathrm{P_a}}{\mathrm{T_a T_0}} \right) - \frac{\tilde{T}}{\tilde{L}} \frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} - \frac{\tilde{T}}{\tilde{\rho_f}} \frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} + \frac{\mathrm{t_s}}{\tilde{\rho_f}\rho_0 \mathrm{C_p}\tilde{z}\mathrm{L_0 T_0}} \dot{q}_{in} \tag{2.17c}$$

$$\frac{\partial \tilde{C}}{\partial \tilde{t}} = \frac{\mathrm{D_0 t_s}}{(\tilde{L}\mathrm{L_0})^2} \frac{\partial}{\partial \tilde{z}} \left(\tilde{D}(\tilde{C}C_0, \tilde{T}T_0) \frac{\partial \tilde{C}}{\partial \tilde{z}} \right) \tag{2.17d}$$

With the boundary conditions:

$$\tilde{z} = 0: \quad \frac{\partial \tilde{C}}{\partial \tilde{z}} = 0 \tag{2.18a}$$

$$\tilde{z} = 1: \quad \frac{\tilde{D}\mathrm{D_0 C_0}}{\tilde{\mathrm{L}}\mathrm{L_0}} \frac{\partial \tilde{C}}{\partial \tilde{z}} + \frac{\tilde{C}\mathrm{C_0 L_0}}{\mathrm{t_s}} \frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} = -\frac{\mathrm{k_m M_s}}{\tilde{\rho_\mathrm{f}}\rho_0 \mathrm{R}} \cdot \left(\frac{P_s}{\tilde{T}\mathrm{T_0}} - \frac{\mathrm{P_a}}{\mathrm{T_a}} \right) \tag{2.18b}$$

With the dimensionless initial values:

$$\begin{aligned} \tilde{t} &= 0: \\ \tilde{C}(0, \tilde{z}) &= 1 \\ \tilde{T}(0) &= 1 \\ \tilde{z}(0) &= 1 \\ \tilde{\rho_f} &= 1 \end{aligned} \tag{2.19}$$

The derivations of the dimensionless system can be found in appendix A. Note that all dimension factors, $(\cdot)_0$, are set to their respective initial value, except $\mathrm{t_s}$, which is set to unity for simplicity.

2.2 Numerical Method

Many problems in applied science and mathematics are described by differential equations. Thus, the the ability to solve such problems are an essential tool for many technological advances. For linear cases, an analytical solution can easily be obtained through calculus. However, for problems described by non-linear phenomenons, an analytical solution may nearly be impossible to obtain. A way around this problem is to rather than seeking an exact solution, one aim to find an approximation of the solution with sufficient accuracy. The fundamental idea of approximating solutions is to evaluate the functions at discrete points in space and/or time, converting continuous problems into a collection of discrete algebraic equations which are well suited for modern computers. Although numerical analysis are well suited for modern computers, numerical algorithms predates computers as e.g. the Newton-Raphson-method was published in 1968 [31]. However, the field of numerical analysis has seen significant growth in recent years due to the improvement made in computational power, extending the application of numerical methods to e.g. bio-mechanics, medicine, social life and economics.

There are a number of different numerical methods for solving non-linear differential equations, varying in terms of efficiency, accuracy and simplicity. For partial differential equations (PDEs), methods as finite volume (FVM), finite different (FDM) and finite element (FEM) is widely adopted for solving complex problems. Although they all represents the problem in mesh and sound some what similar, the approach for solving the differential equations of these methods are very different. The FDM converts the differential equations by approximations of the differentials them self, where as the FVM converts volume integrals containing divergence into surfaces integrals by the divergence theorem. The FEM approximates the unknown functions within the domain by using shape functions of a given degree.

Due to the non-linearities of the coupled differential equations presented in the previous section and the fact that equation (2.17d) is a non-linear PDE, an analytical solution would be challenging and in fact maybe impossible to obtain. Hence, a numerical approach is taken. In previous work, the FDM was applied which showed to be comparatively less efficient and inaccurate [32]. Furthermore, despite being suitable for fluid dynamics, the FEM is chosen rather than FVM due to its efficiency. Considering that the model is to be used for optimization and control, solving the problem in a respectable time-frame is essential.

2.2.1 Finite Element Method

The FEM was originally developed for solving problems for elasticity and structural loading in engineering. However, in recent years, the FEM are also widely applied to problems beyond structural analysis. The method has several variations, such as the meshfree method, Galerkin method, discontinuous Galerkin method and mixed FEM. Although each method differs from one another, they are all based on variational formulations. Due to the close relation between the numerical and variational formulation, the FEM gives useful bounds for the error of the numerical method. The approach taken in this thesis will be the one of a Galerkin formulation, which in essence means that in addition to the unknown functions, the weighting function are also described by the chosen shape functions.

The system consists of four equation, which of only one is a PDE. Thus, the FEM is only applied to the equation (2.17d) due to the changes in the spatial domain. Let Ω denote the spatial domain $(0,1)^{\mathbb{R}^1}$, the notation $d\Omega$ represent the boundaries on Ω, and Ω_t the time domain $(0, t_f)$ where t_f is the final time. The finite approximation problem is cast with a slightly different notation due to simplistic reason. The tilde notation in the previous chapter is omitted, thus, all variables are indeed the dimensionless variables. Furthermore, C is replaced by u just for the sake of generalization. The finite approximation problem in its strong form can then be cast as:

Find u such that:

$$\frac{du(z,t)}{dt} = \alpha \frac{\partial}{\partial z}\left(D(u,T)\frac{\partial u(z,t)}{\partial z}\right), \quad \forall z \in \Omega, \forall t \in \Omega_t \tag{2.20a}$$

$$\frac{\partial u}{\partial z} = 0, \quad z = 0 \quad \forall t \in \Omega_t \tag{2.20b}$$

$$D(u,T)\frac{\partial u}{\partial z} = \Gamma, \quad z = 1 \quad \forall t \in \Omega_t \tag{2.20c}$$

$$u(z,0) = C_0 \tag{2.20d}$$

where α is $\frac{D_0}{(\tilde{L}L_0)^2}$ and Γ is the boundary condition stated in equation (2.18b) rearranged with respect to $\tilde{D}\frac{\partial \tilde{C}}{\partial \tilde{z}}$. The finite approximation begins with deriving the variational and weak formulation from the strong formulation above.

Variational and Weak Formulation

The variatonal form is found by multiplying the strong formulation by a weighting function, $v(z)$, and integrating over the domain Ω. The weighting function is arbitrarily, however, it has to fulfill some criteria. Firstly, for the integral to exist it is required that both $v(z)$ and $v'(z)$ is bounded on Ω. Secondly, it has to make the boundary conditions well defined. Let $\mathcal{V}$ define the function space which contains all possible functions $v(z)$:

$$\mathcal{V} = \left\{ v : ||v'|| \leq \infty,\ ||v|| \leq \infty,\ v(0) = 0,\ v(1) = 1 \right\}$$

Clearly, $\mathcal{V}$ contains a infinite number of functions, thus, $\mathcal{V}$ has infinite dimensions. Rather than looking for an exact solution in an infinite function space, an approximation within a subspace $\mathcal{V}_h$ which contains all piecewise quadratics which ensures that the integral is well defined is considered:

$$\mathcal{V}_h \subset \mathcal{V}$$

The variational form can then be expressed as:

$$\int_\Omega v(z) \frac{du(z,t)}{dt} dz = \alpha \cdot \int_\Omega v(z) \frac{\partial}{\partial z} \left(D(u,T) \frac{\partial u(z,t)}{\partial z} \right) dz \tag{2.21}$$

To arrive at the weak formulation, the integral on the right hand side is resolved using integration by parts:

$$\int_\Omega v(z) \frac{\partial u(z,t)}{\partial t} dz = \alpha \cdot \left[D(u,T) v(z) \frac{\partial u(z,t)}{\partial z} \right]_{d\Omega} - \alpha \cdot \int_\Omega D(u,T) \frac{\partial u(z,t)}{\partial z} \frac{\partial v(z)}{\partial z} dz$$

$$\int_\Omega v(z) \frac{\partial u(z,t)}{\partial t} dz = \alpha v(z) \Gamma - \alpha \cdot \int_\Omega D(u,T) \frac{\partial u(z,t)}{\partial z} \frac{\partial v(z)}{\partial z} dz \tag{2.22}$$

where Γ represents the boundary condition at $z = 1$. Equation (2.22) is the weak formulation of equation (2.20a), which is later used as the basis for the formulation of the finite element approximation.

Finite Element Approximation

In the same manner as for the weighting function $v(z)$, the exact solution of $u(z,t)$ can be found in a function space $\mathcal{S}$ with infinite dimensions. The approximation of $u(z,t)$, from now on referred to as u_h, is approximated with peicewise quadratics, thus, the search for a solution is reduced to a finite function space $\mathcal{S}_h$ which contains all peicewise quadratics that ensures that the problem is well defined. Hence, the finite element approximation problem is defined as:

Find u_h such that:

$$u_h \in \mathcal{S}_h, \quad \mathcal{S}_h \subset \underbrace{\left\{ u : \frac{du(0,t)}{dz} = 0, D(u,T) \frac{du(1,t)}{dz} = \Gamma \right\}}_{= \mathcal{S}}$$

For the finite element approximation, the spatial domain is first subdivided into n_e number of subdomains referred to as elements. These elements now describes the problem on a local level and the unknown function u within each element. Figure 2.4 illustrates how the domain Ω is subdivided into n_e number of subdomains, and it is clear that $\Omega_e \subset \Omega$.

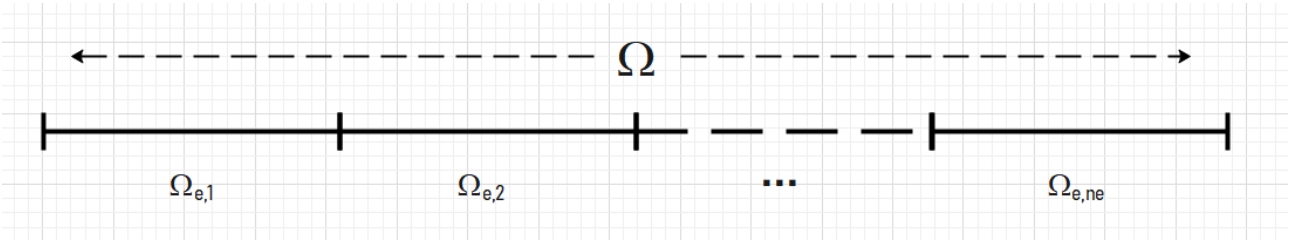


Figure 2.4: Illustration of the partition of the domain Ω in to n_e number of subdomains.

Within each element, the unknown function u is approximated by piecewise polynomials of n-th degree, referred to as shape functions, and the nodal values of u. The number of nodes in each element varies with the order of the piecewise polynomials as the sum of polynomials always equals one. Linear shape functions requires two nodes for each elements,

quadratic polynomials requires three and so forth. A common property of the shape functions is that it takes the value of unity at their corresponding spatial value, expressed mathematically as:

$$\phi_j(z_i) = \begin{cases} 1, & if\ i = j, \\ 0, & if\ i \neq j \end{cases} \quad , \quad i, j = 0, 1, ..., n$$

Higher order polynomials does not only implies higher accuracy of the approximated solution, but also higher complexity and computational power requirement for the method. In this thesis quadratic shape functions are used, as it assumed that polynomials of 2th order will resemble the unknown function u sufficiently. The domain Ω is subdivided into n_e number of elements, each with a length h. Now, each element is treated with local coordinates ranging from $z = 0$ to $z = h$. Three quadratic polynomials which satisfy the properties mentioned above is:

$$\phi_1(z) = 1 - \frac{3}{h}z + \frac{2}{h^2}z^2, \quad \phi_2(z) = \frac{4}{h}z - \frac{4}{h^2}z^2, \quad \phi_3(z) = \frac{2}{h^2}z^2 - \frac{1}{h}z \tag{2.23}$$

Figure 2.5 is a plot of the three quadratic shape functions for an arbitrary element using local coordinates. As previously

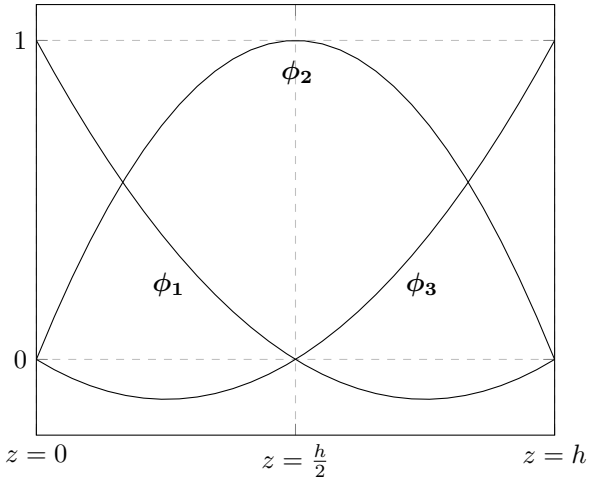


Figure 2.5: Plot of quadratic shape functions in local coordinates for a given element.

stated, the unknown function u is approximated by the shape functions along with nodal values. This can be represented by:

$$u_h = \sum_{i=1}^{3} \epsilon_i \phi_i \tag{2.24}$$

where ϵ_i represents the nodal value at the i'th node. Likewise, the weighting function, which also are approximated by its nodal values and the shape functions shares a similar definition:

$$v_h = \sum_{i=1}^{3} c_i \phi_i \tag{2.25}$$

where c_i represents the nodal value of the weighting function at the i'th node. Substituting equation (2.24) and equation (2.25) into equation (2.22) and restricting the domain to the one of a single element in the interior of Ω (this is for any element that is not on the boundary of Ω), the weak formulation can be expressed as:

$$\int_{\Omega_e} \sum_{i=1}^{3} c_i \phi_i \sum_{i=1}^{3} \dot{\epsilon}_i \phi_i \, dz = -\alpha \cdot \int_{\Omega_e} D(u_h, T) \frac{\partial \sum_{i=1}^{3} c_i \phi_i}{\partial z} \frac{\partial \sum_{i=1}^{3} \epsilon_i \phi_i}{\partial z} dz \tag{2.26}$$

To simplify, equation (2.24) and equation (2.25) is rewritten in matrix form as follows:

$$u_h = \sum_{i=1}^{3} \epsilon_i \phi_i = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \end{bmatrix} \cdot \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} = \boldsymbol{\Phi \epsilon} \tag{2.27}$$

$$v_h = \sum_{i=1}^{3} c_i \phi_i = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \boldsymbol{\Phi c} = \boldsymbol{c}^\top \boldsymbol{\Phi}^\top \tag{2.28}$$

where both ϵ_i and c_i are independent of z, while ϕ_i is independent of t. Thus, the derivatives can be written as:

$$\frac{\partial u_h}{\partial t} = \frac{\partial \boldsymbol{\Phi}\boldsymbol{\epsilon}}{\partial t} = \boldsymbol{\Phi}\dot{\boldsymbol{\epsilon}} \tag{2.29}$$

$$\frac{\partial u_h}{\partial z} = \frac{\partial \boldsymbol{\Phi}\boldsymbol{\epsilon}}{\partial z} = \frac{\partial \boldsymbol{\Phi}}{\partial z}\boldsymbol{\epsilon} = \boldsymbol{\Psi}\boldsymbol{\epsilon} \tag{2.30}$$

$$\frac{\partial \boldsymbol{c}^\top \boldsymbol{\Phi}}{\partial z} = \boldsymbol{c}^\top \frac{\partial \Phi}{\partial z} = \boldsymbol{c}^\top \boldsymbol{\Psi}^\top \tag{2.31}$$

Substituting equation (2.27)-equation (2.31) into equation (2.26) yields the weak formulation on matrix form:

$$\int_{\Omega_e} \boldsymbol{c}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\, dz \cdot \dot{\boldsymbol{\epsilon}} = -\alpha \cdot \int_{\Omega_e} D(u_h, T)\boldsymbol{c}^\top \boldsymbol{\Psi}^\top \boldsymbol{\Psi}\, dz \cdot \boldsymbol{\epsilon} \tag{2.32}$$

As c_i is just a scalar, hence, $\boldsymbol{c}^\top$ is a scalar vector, equation (2.32) is divided by $\boldsymbol{c}^\top$ on both sides.

$$\int_{\Omega_e} \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\, dz \cdot \dot{\boldsymbol{\epsilon}} = -\alpha \cdot \int_{\Omega_e} D(u_h, T)\boldsymbol{\Psi}^\top \boldsymbol{\Psi}\, dz \cdot \boldsymbol{\epsilon} \tag{2.33}$$

Written in a compact form, it becomes clearer that the PDE is discretizied in space and converted to ODE:

$$\boldsymbol{M_e} \cdot \dot{\boldsymbol{\epsilon}} = -\alpha \boldsymbol{K_e} \cdot \boldsymbol{\epsilon} \tag{2.34}$$

where the $\boldsymbol{M_e}$ and $\boldsymbol{K_e}$ is often referred to as the mass and stiffness matrices. The subscription denotes the local mass and stiffness matrices, and are defined as:

$$\boldsymbol{M_e} = \int_{\Omega_e} \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\, dz \tag{2.35}$$

$$\boldsymbol{K_e} = \int_{\Omega_e} D(u_h, T)\boldsymbol{\Psi}^\top \boldsymbol{\Psi}\, dz \tag{2.36}$$

When forming the global mass and stiffness matrices, the local matrices are appended to the global matrix along its diagonal. However, as each node at the boundary of an element interacts with two elements, the local matrices has to be appended accordingly. To avoid a large example, the procedure for assembling the global matrices are shown for linear shape function. However, the procedure is the same for matrices of higher order shape functions.

$$\boldsymbol{M} = \begin{bmatrix} M_{1,11} & M_{1,12} & 0 & 0 & \dots & 0 & 0 \\ M_{1,21} & M_{1,22} + M_{2,11} & M_{2,12} & 0 & \dots & 0 & 0 \\ 0 & M_{2,21} & M_{2,22} + M_{3,11} & M_{3,12} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & M_{n-1,22} + M_{n,11} & M_{n,12} \\ 0 & 0 & 0 & 0 & 0 & M_{n,21} & M_{n,22} \end{bmatrix} \tag{2.37}$$

$$\boldsymbol{K} = \begin{bmatrix} K_{1,11} & K_{1,12} & 0 & 0 & \dots & 0 & 0 \\ K_{1,21} & K_{1,22} + K_{2,11} & K_{2,12} & 0 & \dots & 0 & 0 \\ 0 & K_{2,21} & K_{2,22} + K_{3,11} & K_{3,12} & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & K_{n-1,22} + K_{n,11} & K_{n,12} \\ 0 & 0 & 0 & 0 & 0 & K_{n,21} & K_{n,22} \end{bmatrix} \tag{2.38}$$

where the notation $(\cdot)_{i,kj}$ refers to the kj'th element of the i'th local matrix with n being the final element. When no longer restricting the domain to a single element, the discretizied ODE for the entire domain Ω becomes:

$$\boldsymbol{M} \cdot \dot{\boldsymbol{\epsilon}} = \alpha \hat{\boldsymbol{\Gamma}} - \alpha \boldsymbol{K}(u_h) \cdot \boldsymbol{\epsilon} \tag{2.39}$$

where $\hat{\boldsymbol{\Gamma}}$ is:

$$\hat{\boldsymbol{\Gamma}} = \boldsymbol{\Phi}^\top \cdot D(u_h, T) \cdot \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \Gamma \end{bmatrix} \tag{2.40}$$

Discretization in Time

Equation (2.39) is discretizited in the spatial domain, however, the formulation needs to be extended with discretization in time to obtain an numerical formulation of the problem. For discretization of the time domain, a similar approach is taken as in [33], but with backwards Euler instead of forward Euler due to stability conditions. Consider a known solvent fraction nodal value at a discrete point in time ϵ_{n-1} and a unknown nodal value ϵ_n at at point Δt forward in time. The objective is to use the known nodal values at t_{n-1} to determine values at t_n. For the time discretization, linear polynomials are chosen to realize a backwards Euler time scheme. The shape functions are given by:

$$\hat{\phi_1} = 1 - \frac{\tau}{\Delta t}, \quad \hat{\phi_2} = \frac{\tau}{\Delta t} \tag{2.41}$$

where τ is the local coordinate of t within each increment. Thus, the nodal value as a function of τ can be expressed as:

$$\epsilon(\tau) = \hat{\phi_1}\epsilon_{n-1} + \hat{\phi_2}\epsilon_n \tag{2.42}$$

Differentiating equation (2.42) with respect to τ yields the following expression for $\dot{\epsilon}$:

$$\dot{\epsilon} = \frac{\epsilon_n - \epsilon_{n-1}}{\Delta t} \tag{2.43}$$

Substituting equation (2.42) and equation (2.43) into equation (2.39) yields the following expressions:

$$\boldsymbol{M} \cdot \frac{\epsilon_n - \epsilon_{n-1}}{\Delta t} = \alpha\hat{\boldsymbol{\Gamma}} - \alpha\boldsymbol{K}(u_{h,n}) \cdot \left(\left(1 - \frac{\tau}{\Delta t}\right)\epsilon_{n-1} + \frac{\tau}{\Delta t}\epsilon_n\right) \tag{2.44}$$

Furthermore, equation (2.44) is multiplied with a weighting function $w(\tau)$ and integrated over the time step. However, the weighting function $w(\tau)$, has to fulfill some criteria as $v(z)$ does. As for $v(z)$, $w(\tau)$ and $w'(\tau)$ has to be bounded on the interval $(0, \tau)$. In addition, $w(\tau)$ has to vanish at the boundaries of the interval.

$$\mathcal{W} = \left\{w : ||w'|| \leq \infty,\ ||w|| \leq \infty,\ w(0) = 0,\ w(1) = 0\right\}$$

$\mathcal{W}$ is a functions space of infinite dimensions, however, the approximated function lies within a subset of $\mathcal{W}$ which contains all the piecewise linears that ensures that the integral is well defined:

$$\mathcal{W}_h \subset \mathcal{W}$$

Multiplying equation (2.44) by the weighting functions $w(\tau)$ and integrating over the time increment yields:

$$\int_0^{\Delta t} w(\tau)\boldsymbol{M} \cdot \frac{\epsilon_n - \epsilon_{n-1}}{\Delta t}\, d\tau = \int_0^{\Delta t} w(\tau)\alpha\hat{\boldsymbol{\Gamma}}\, d\tau - \int_0^{\Delta t} w(\tau)\alpha\boldsymbol{K}(u_{h,n}) \cdot \left(\left(1 - \frac{\tau}{\Delta t}\right)\epsilon_{n-1} + \frac{\tau}{\Delta t}\epsilon_n\right) d\tau \tag{2.45}$$

Dividing by $\int_0^{\Delta t} w(\tau)d\tau$:

$$\boldsymbol{M} \cdot \frac{\epsilon_n - \epsilon_{n-1}}{\Delta t} = \alpha\hat{\boldsymbol{\Gamma}} - \frac{w(\tau)\alpha\boldsymbol{K}(u_{h,n}) \cdot \left(\left(1 - \frac{\tau}{\Delta t}\right)\epsilon_{n-1} + \frac{\tau}{\Delta t}\epsilon_n\right)\, d\tau}{\int_0^{\Delta t} w(\tau)d\tau} \tag{2.46}$$

Assuming that $w(\tau) = \frac{\tau}{\Delta t}$ and rearranging with respect to ϵ_n yields:

$$\boldsymbol{M}\boldsymbol{\epsilon_n} + \frac{2}{3}\alpha\Delta t\boldsymbol{K}(u_{h,n})\boldsymbol{\epsilon_n} = -\frac{1}{3}\alpha\Delta t\boldsymbol{K}(u_{h,n})\boldsymbol{\epsilon_{n-1}} + \alpha\Delta t\hat{\boldsymbol{\Gamma}} \tag{2.47}$$

Equation (2.47) is the finite element formulation of the problem stated in the introduction of this chapter. Clearly, the equation is non-linear due to backwards Euler scheme and the fact that the stiffness matrix $\boldsymbol{K}$ depends on the solution of the approximation. The non-linear equation is solved using an iterative algorithm presented in the next section.

Picard Iteration

To handle the non-linearities of equation (2.47) an iterative procedure has to be implemented which solves the non-linear equation iterative until a solution with a sufficient accuracy is found. There are several methods for solving non-linear equations e.g. the Newton-Raphson method and fixed-point iterations. Picard iteration may be the simplest method for

solving non-linear equations. However, its convergence is relative slow compared to other method, such as the Newton-Raphson method. Picard iteration is applicable to functions on the form:

$$x = g(x) \tag{2.48}$$

where $g(x)$ is an arbitrarily function. In discrete formulation:

$$x^{k+1} = g(x^k) \tag{2.49}$$

where the superscription denotes the current iteration. The iteration procedure is terminated as soon as the difference $\delta = |x^{k+1} - x^k|$ becomes lower than a desired threshold. Now, the focus is shifted to the Newton-Raphsons method. Consider the function:

$$g(x) = 0 \tag{2.50}$$

which assumed to be a non-linear function. The Taylor serie's expansions of $g(x)$ when higher order terms are neglected is as follows:

$$g(x) \approx g(x_0) + \boldsymbol{J}|_{x=x_0} \cdot (x - x_0) \tag{2.51}$$

where $\boldsymbol{J}$ is the Jacobian of $g(x)$. This can also be viewed as an iterative procedure where x_0 is the k'th iteration and x is the value at iteration k+1:

$$g(x_{k+1}) = g(x_k) + \boldsymbol{J}|_{x=x^k} \cdot (x_{k+1} - x_k) \tag{2.52}$$

Let now r and d define the residual in the solution and deviation in the x values, respectively. Then the linearization of the function $g(x)$ can be written in deviation variables from the solution $g(x) = 0$:

$$r_k = \boldsymbol{J}|_{x=x^k} \cdot d_k \tag{2.53}$$

The next iteration can then be determined by:

$$\begin{aligned} x^{k+1} &= x^k + d^k \\ x^{k+1} &= x^k + \boldsymbol{J}|^{-1}_{x=x^k} r^k \end{aligned} \tag{2.54}$$

Equation (2.54) is the Newton-Raphson algorithm used to solve a non-linear equations. It converges with a quadratic rate which is faster than for fixed-point method, such as Piccard iteration. However, the Newton-Raphson method requires that we calculate the Jacobian at each iteration, which can be computationally demanding. In addition, in situations where the Jacobian becomes singular or close to singular, the method fails to determine the root as the solution blows up due to division by zero. A solution to this problem is to use an approximation of the Jacobian. By linearizing the residual in terms of using the known values at the last iteration, the residual r can be expressed by the following:

$$r^k = \boldsymbol{b^{(k)}} - \boldsymbol{A^{(k)}} x^k \tag{2.55}$$

Furthermore, lets approximate the Jacobian using the system matrix $\boldsymbol{A}^{(k)}$. Substituting equation (2.55) and the approximation of $\boldsymbol{J}$ into equation (2.54) gives us:

$$\begin{aligned} x^{k+1} &= x^k + \boldsymbol{A^{(k)}}^{-1} \cdot \left(\boldsymbol{b^{(k)}} - \boldsymbol{A^{(k)}} x^k \right) \\ x^{k+1} &= \boldsymbol{A^{(k)}}^{-1} \boldsymbol{b^{(k)}} \end{aligned} \tag{2.56}$$

Comparing equation (2.49) and equation (2.56) it is clear that the latter is Picard iteration as a simplified Newton-Raphson method.

To apply the simplified Newton-Raphson method to the finite approximation equation in equation (2.47) every term in equation (2.47) is shifted to the left-hand side and set to equal zero:

$$\boldsymbol{r}^i = \boldsymbol{M}{\boldsymbol{\epsilon_n}}^i + \frac{2}{3}\alpha\Delta t \boldsymbol{K}(\boldsymbol{\epsilon_n}^i, T_n)\boldsymbol{\epsilon_n}^i + \frac{1}{3}\alpha\Delta t \boldsymbol{K}(\boldsymbol{\epsilon_n}^i, T_n)\boldsymbol{\epsilon_{n-1}} - \alpha\Delta t\hat{\boldsymbol{\Gamma}} = 0 \tag{2.57}$$

To get the residual on the form as in equation (2.55) the expression has to be linearizied. ϵ_{n-1} is the solvent fraction value at the previous node, thus, it can be treated as a constant. Furthermore, the linearization is rather simple by using the most recent approximation of $\boldsymbol{\epsilon}$. Prior to rearranging equation (2.57) to a form similar to the one in equation (2.55), the boundary condition is divided into two parts as one term contains the solvent fraction value at the boundary:

$$\alpha\hat{\boldsymbol{\Gamma}} = \underbrace{-\frac{1}{L_n} \cdot \frac{dL}{dt}}_{=\boldsymbol{B_c}} \cdot \epsilon_n^i - \underbrace{\frac{k_m M_s}{\rho_s R L_n L_0 C_0} \cdot \left(\frac{P_s(\epsilon_n^-)}{T_n T_0} - \frac{P_a}{T_a} \right)}_{=\boldsymbol{f}} \tag{2.58}$$

where ϵ_n^- denotes the most recent approximation. Thus, the residual can be written on the form as equation (2.55) with $\boldsymbol{A^{(k)}}$ and $\boldsymbol{B^{(k)}}$ being:

$$\boldsymbol{A^{(k)}} = -\left(\boldsymbol{M} + \frac{2}{3}\alpha\Delta t\boldsymbol{K}(\boldsymbol{\epsilon_n}^i, T_n) - \Delta t\boldsymbol{B}_c\right) \tag{2.59}$$

$$\boldsymbol{B^{(k)}} = \left(\frac{1}{3}\alpha\Delta t\boldsymbol{K}(\boldsymbol{\epsilon_n}^i, T_n) - \boldsymbol{M}\right)\cdot\boldsymbol{\epsilon}_{n-1} - \Delta t\boldsymbol{f} \tag{2.60}$$

Thus, the next iteration value of ϵ_n is expressed as:

$$\boldsymbol{\epsilon_n}^{i+1} = -\left(\boldsymbol{M} + \frac{2}{3}\alpha\Delta t\boldsymbol{K}(\boldsymbol{\epsilon_n}^i, T_n) - \Delta t\boldsymbol{B}_c\right)^{-1}\left(\left(\frac{1}{3}\alpha\Delta t\boldsymbol{K}(\boldsymbol{\epsilon_n}^i, T_n) - \boldsymbol{M}\right)\cdot\boldsymbol{\epsilon}_{n-1} - \Delta t\boldsymbol{f}\right) \tag{2.61}$$

Both $\boldsymbol{B_c}$ and $\boldsymbol{f}$ are extended with zeros such that it fits the dimensions of $\boldsymbol{M}$ and $\boldsymbol{K}$.

2.2.2 Time Discretization of the ODEs

As previously mentioned, the FEM is only applied to equation (2.17d) has all of the differential equation for the density, thin film thickness and thin film temperature does not involve differential with respect to the spatial domain. To solve equation (2.17a)-equation (2.17d) a simpler approach is taken as a forth order explicit Runge-Kutta method is applied. The Runge-Kutta method is a family of one-step methods for solving differential equations. Compared to the simpler Euler's method, the Runge-Kutta method demands more computation, but is in return more accurate and stable [34]. However, deriving the method is a tedious procedure as it involves applying the trapezoidal method and evaluating the differential at four different values. Thus, a general formulation of the fourth order explicit Runge-Kutta as stated by Egeland and Gravdalh in [35] is used with a slightly different notation. Figure 2.6a is an illustration of how the differential terms are evaluated at four different values to get a general idea of how the forth order Runge-Kutta approximates the next iteration value. In accordance to formulation by Egeland and Gravdalh in [35], consider the differential equation on the form:

$$\frac{du}{dt} = f(u, t) \tag{2.62}$$

Then the forth order explicit Runge-Kutta method is given by:

$$k_1 = f(u_n, t_n) \tag{2.63a}$$

$$k_2 = f(u_n + \frac{h}{2}k_1, t_n + \frac{h}{2}) \tag{2.63b}$$

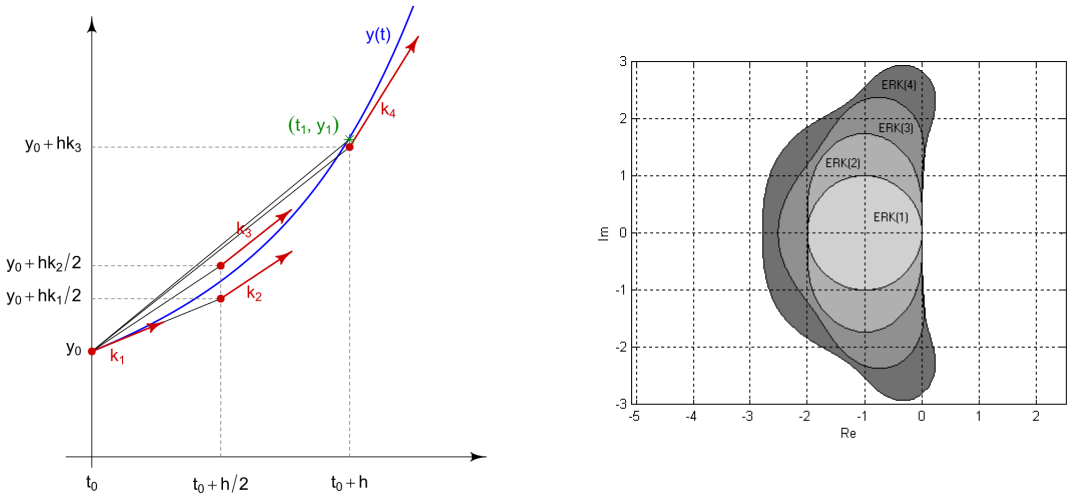
$$k_3 = f(u_n + \frac{h}{2}k_2, t_n + \frac{h}{2}) \tag{2.63c}$$

$$k_4 = f(u_n + hk_3, t_n + h) \tag{2.63d}$$

$$u_{n+1} = u_n + \frac{h}{6}\cdot(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.63e}$$

where $h = \Delta t$.

In contrast to implicit methods, the explicit methods are not unconditionally stable. As a result, depending on the chosen time (and spatial) increment, the approximated solution can grow unbounded and no longer give any meaningful information. To avoid instability, it is important to chose the parameters such that amplification factor is still within the stability region of the applied method. Figure 2.6b illustrated the stability region for the Runge-Kutta methods from order one to four. The forth order method has clearly larger stability region of that of an order one, which is identical to the stability region of the Euler's method. Stability estimates are often limited to linear systems as such estimates for non-linear systems are tedious to derive and involves to a large extent advanced mathematics. Thus, an estimate of the stability condition for this particular problem will not be derived. However, the concept of instability is important to be aware of as certain combination of parameter values can render the numerical method unstable.



(a) Illustration of the usage of four slope values for the forth order Runge-Kutta method [36].

(b) The stability regions of the Runge-Kutta method from order one to four [35].

Figure 2.6: Illustration of the approximation procedure and stability region of the Runge-Kutta method.

2.3 Linear System Theory

Linearization is in essence finding a linear approximation of a non-linear function around a specific point. Linear systems are easier to handle in terms of analysis, control and optimization as linear system theory can be applied which can not be extended to non-linear system. Thus, it is common in situations when dealing with non-linear system to derive a linear model which hopefully resembles the non-linear model with sufficiency. A linearized model is found by restricting the Taylor Serie's expansion to its linear terms. Consider the the non-linear differential function written in vector form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{2.64}$$

where $\boldsymbol{f}(\boldsymbol{x})$ is assumed to be infinite differentiable. Then it's Taylor Serie's expansion is:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x_0}) + \sum_{k=1}^{\infty} \frac{\boldsymbol{f^{(k)}}(\boldsymbol{x_0})}{\boldsymbol{k!}} \cdot (\boldsymbol{x} - \boldsymbol{x_0})^{\boldsymbol{k}} \tag{2.65}$$

Restricting the Taylor Serie's to its linear terms yields:

$$\dot{\boldsymbol{x}} \approx \boldsymbol{f}(\boldsymbol{x_0}) + \boldsymbol{J} \cdot (\boldsymbol{x} - \boldsymbol{x_0}) \tag{2.66}$$

A simple example of linearization of a function is illustrated in figure 2.7. The function $f(x) = x^2$ is linearized around $x = 1$. In a close neighbourhood of $x = 1$ the approximated linear model is accurate. However, when moving further away from the point $x = 1$ the linearized approximation becomes less accurate. For these reasons, a linear approximation of a non-linear model is only valid for small perturbations from the linearization point.

Usually, when controlling a process around the operating point, the model parameters are defined as deviations variables from the operating point. To convert the linearized model into a model expressed by deviation variables are done in a fairly straight forward manner. Again, consider the linear terms of the Taylor Serie's expansion as in equation (2.66). For simplicity $\dot{\boldsymbol{x}}$ is replaced by $\boldsymbol{f}(\boldsymbol{x})$ to make the notation easier to follow. A linear model expressed by deviation variables can then be derived as follows:

$$\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x_0}) \approx \boldsymbol{J} \cdot (\boldsymbol{x} - \boldsymbol{x_0})$$
$$\boldsymbol{\Delta f}(\boldsymbol{x}) = \boldsymbol{\Delta \dot{x}} \approx \boldsymbol{J} \cdot \boldsymbol{\Delta x}$$

The system is now a linear differential equation on the form:

$$\Delta \dot{\boldsymbol{x}} = \boldsymbol{A} \cdot \Delta \boldsymbol{x} \tag{2.67}$$

In a control problem, one or more states serves as manipulated variables used to drive the system in a desired direction. These states are removed form the state vector $\boldsymbol{x}$ and placed in a separate input vector $\boldsymbol{u}$. As a consequence, the linear

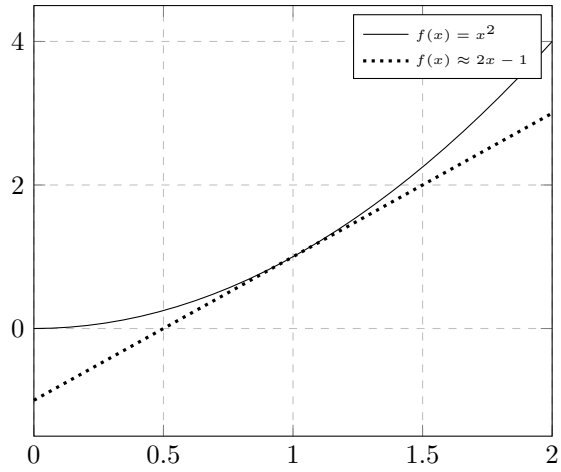


Figure 2.7: Linearization of the function $f(x) = x^2$ around $x = 1$.

system of differential equation can then be written in the form a of state space realization:

$$\begin{aligned} \Delta\dot{\boldsymbol{x}} &= \boldsymbol{A} \cdot \Delta\boldsymbol{x} + \boldsymbol{B} \cdot \Delta\boldsymbol{u} \\ \Delta\boldsymbol{y} &= \boldsymbol{C} \cdot \Delta\boldsymbol{x} + \boldsymbol{D} \cdot \Delta\boldsymbol{u} \end{aligned} \tag{2.68}$$

where $\Delta\boldsymbol{x}$ is the reduced state vector as one or more states is used as control inputs. The removed states from $\Delta\boldsymbol{x}$ are placed in $\Delta\boldsymbol{u}$. $\boldsymbol{B}$ is a matrix with the terms which contains $\Delta\boldsymbol{u}$, e.g. k_c if the air temperature T_a is used as a control input. $\Delta\boldsymbol{y}$ represents the measurements in the system which in most cases is limited to some of the states them self. However, the measurement can be a combination of several states and inputs as $\boldsymbol{C}$ and $\boldsymbol{D}$ is the matrices which contains the respective coefficients for $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{u}$ for the distinct measurement.

2.3.1 Stability

An important property of the continuous state-space model is the stability. An unstable system may be stabilized by feedback control. However, designing a controller is much easier when the system is already stable. The eigenvalues of the system matrix, $\boldsymbol{A}$, can be used as a measure of the system stability. To get an idea on how the eigenvalues can be a measurement of the system stability, consider the single variable differential equation:

$$\dot{y} = \lambda y \tag{2.69}$$

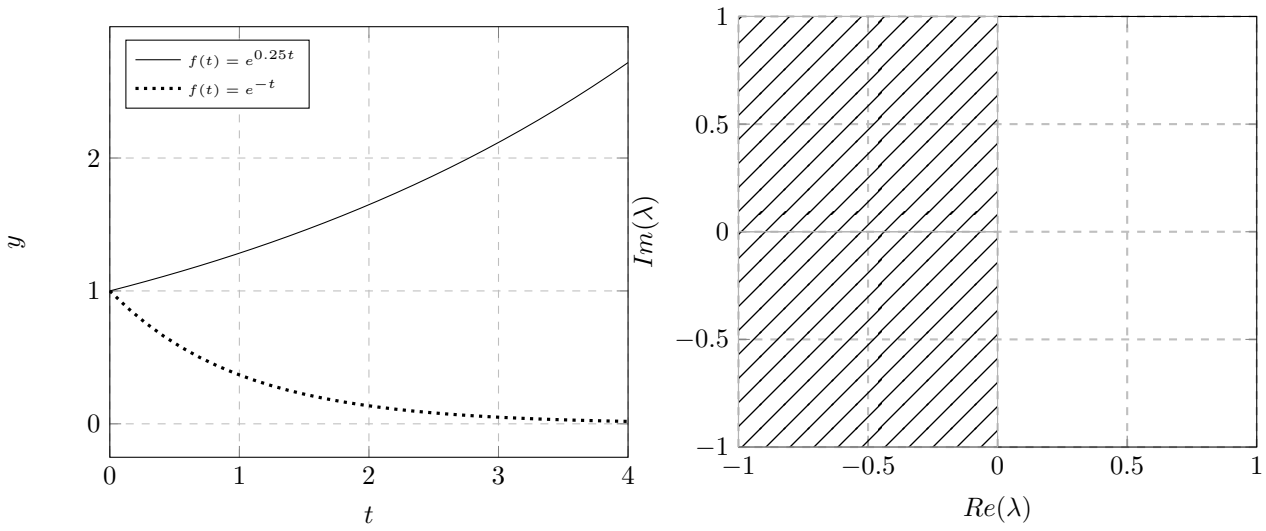
with the analytical solution:

$$y(t) = e^{\lambda t} \tag{2.70}$$

Figure 2.8a shows the plot of two distinct scenarios with the eigenvalues of the system being 0.25 and -1, respectively. As seen for the case with $\lambda = -1$, the function value is decreasing exponentially with time. For the second case $\lambda = 0.25$, the function value is increasing exponentially despite being lower than 1. Clearly, the effects the second case has on system is unwanted as state values will eventually grow unbounded with time as:

$$\lim_{t\to\infty} e^{\lambda t} = \infty, \quad \forall\lambda > 0$$

Thus, a system is stable when the real value of the eigenvalue are less or equal to zero. Figure 2.8b illustrates the region in which a continuous system remains stable. A special case occurs when an eigenvalue at zero is not a simple root. Where as a simple root with a real value of zero implies marginally stability, a root of higher multiplicity than 1 leaves the system unstable. However, Hovd mentions that in most practical situations system which are marginally stable will need to be stabilized by feedback and therefore classifies such system as unstable [37]. The eigenvalues effects on the system's stability can be extended to multivariable system as multivariable system can be solved in the same manner as monovariable system which is well explained by Chen in [38]. Thus, the stability region of a monovariable system also hold for a multivariable system with the exceptions of roots at zero with a multiplicity of two or higher.



(a) Illustration of the eigenvalues influence on the solution.

(b) The stability region of the a continuous system.

Figure 2.8: Plots for illustrating the effects of λ on the stability of a system.

2.3.2 Discretization of Continuous Model

In practice, modern computers and electronics does not have the ability to read and apply measurements and control inputs continuously. These actions are all executed at discrete points in time in intervals. As a consequence, a control system can not use a continuous model, as this requires a continuous flow of measurements readings and control input applications. The solution to this problem is to discretize the continuous model. There are several method to discretize continuous model, e.g. Euler's method and Zero-Oder-Hold method. The approach of the latter is taken. Assume zero order hold (which in essence means that each sample value is hold constant throughout the sampling interval) and a time step of length k, then integration over a time step yields:

$$\boldsymbol{x(n+nk) = e^{Ak}x(nk) + \int_{nk}^{nk+k} e^{A(nk+k-\tau)}B\tau d\tau} \tag{2.71}$$

which yields the following discrete system:

$$\boldsymbol{x_{n+1} = A_d x_k + B_d u_k} \tag{2.72}$$

where:

$$\boldsymbol{A_d = e^{Ak} = \sum_{j=0}^{\infty} \frac{A^j}{j!}} \tag{2.73}$$

$$\boldsymbol{B_d = \int_{nk}^{nk+k} e^{A(nk+k-\tau)}B\tau d\tau} \tag{2.74}$$

Equation (2.72) is the discrete state-space model. $\boldsymbol{C}$ and $\boldsymbol{D}$ remains the same as the measurements readings are not affected by the discretization. However, a property that is affected by the discretization is the stability region. As for the continuous system, it is desirable to avoid the discrete solution to grow unbound with increasing iterations. Because of the iterative procedure, the amplification factor can make the system grow unbounded in both positive and negative direction. Thus, the stability region for a discrete model is limited to the unit circle as illustrated in figure 2.9.

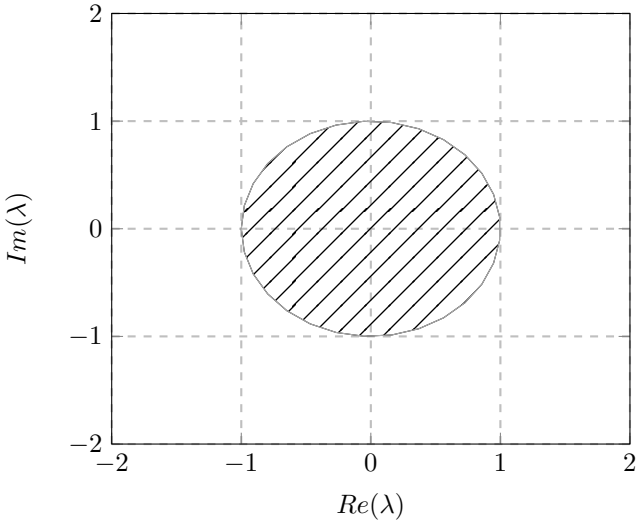


Figure 2.9: The stability region of a discrete linear system.

2.3.3 Controllability and Observability

For control purposes, information about whether or not the system is controllable based upon the chosen control inputs is of interest. With the term controllable, it means that the control input has the ability to move any state within the state space to an arbitrarily state within a finite time interval. If a system fulfills this criteria, the pair $(\boldsymbol{A}, \boldsymbol{B})$ is said to be controllable. For linear system, the controllability matrix is defined as:

$$\mathcal{C} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \boldsymbol{A^2B} & \dots & \boldsymbol{A^{n-1}B} \end{bmatrix} \tag{2.75}$$

where n is the dimension of the square matrix $\boldsymbol{A}$, or more convenient, the number of states. If the rank of $\mathcal{C}$ is of n, then the system controllable. For each linear dependent row vectors there is one state which is uncontrollable. Note that term controllability does not mention anything about which trajectory the states should take, nor the magnitude of the control input. Thus, in practice, controllability can be prevented by saturation limits. Furthermore, the observability matrix gives information whether or not the system is observable. Observability means that with information about the control input and measurement, any unknown initial states can be determined. If so, the pair $(\boldsymbol{A}, \boldsymbol{C})$ is said to be observable. The observability matrix is defined as:

$$\mathcal{O} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA^2} \\ \vdots \\ \boldsymbol{CA^{n-1}} \end{bmatrix} \tag{2.76}$$

In comparions to the controllability matrix, the system is observable if and only if the rank of $\mathcal{O}$ equals the number of states n.

2.4 State Estimation

In control problems, such as in stabilization using state-feedback, knowing the state values is a necessity. However, measurement of every state is rarely available in practice and for those that are, the signals will be contaminated by noise. A solution to this problem is to use state estimation which uses the available measurements and the covariance of the process and measurement noise to calculate a more accurate estimation of the states. A widely adopted method for doing so is by implementing a Kalman filter. The Kalman filter comes in several different variations and can be applied to linear and non-linear systems. The unscented Kalman filter (UKF) and extended Kalman filter (EKF) is methods which can be applied to non-linear problems, with the latter being the one applied in this thesis [39]. The EKF is in essence the original linear Kalman filter, but for each iteration it is linearized about the estimate of the current mean and covariance. Consider the non-linear discrete system:

$$\begin{aligned} \boldsymbol{x_k} &= \boldsymbol{f}(\boldsymbol{x_{k-1}}, \boldsymbol{u_{k-1}}, \boldsymbol{w_{k-1}}) \\ \boldsymbol{y_k} &= \boldsymbol{h}(\boldsymbol{x_k}, \boldsymbol{v_k}) \end{aligned} \tag{2.77}$$

where $\boldsymbol{f}$ is the non-linear state function, $\boldsymbol{w}$ is the process noise, $\boldsymbol{h}$ is the measurement function and $\boldsymbol{v}$ is the measurement noise. The process and measurement noise is assumed to be normally distributed Gaussian white noise, such that:

$$\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{Q}), \quad \boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{R})$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ represents the covariance matrices for $\boldsymbol{w}$ and $\boldsymbol{v}$, respectively. Linearizing the non-linear state function about the expected values yields:

$$\boldsymbol{x_k} \approx \boldsymbol{f}(\boldsymbol{x_{k-1,a}}, \boldsymbol{u_{k-1,a}}, \boldsymbol{w_{k-1,a}}) + \boldsymbol{J_x} \cdot (\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1,a}}) + \boldsymbol{J_u} \cdot (\boldsymbol{u_{k-1}} - \boldsymbol{u_{k-1,a}}) + \boldsymbol{J_w}(\boldsymbol{w_{k-1}} - \boldsymbol{w_{k-1,a}}) \tag{2.78}$$

$$\boldsymbol{y_k} \approx \boldsymbol{h}(\boldsymbol{x_{k,a}}, \boldsymbol{v_{k,a}}) + \boldsymbol{H_x} \cdot (\boldsymbol{x_k} - \boldsymbol{x_{k,a}}) + \boldsymbol{J_v} \cdot (\boldsymbol{v_k} - \boldsymbol{v_{k,a}}) \tag{2.79}$$

where the subscription $(\cdot)_{k-1,a}$ denotes the expected value of the previous iteration. Depending on the information that is available, the most accurate estimation of the states is always desired. In the situations where all of the measurement is available, an estimate based upon the expected values of all the measurement can be calculated, often referred to as a posteriori estimate. However, there is situation where the last measurement has not yet been processed, a priori estimate can be calculated based upon the expected values of the previous measurement. The priori estimate can then later be smoothed when the last measurement has been acquired, producing a more accurate estimate. As the priori estimate is determined by the expected values of the previous measurements, it can be calculated based upon the expected value of the previous posteriori estimate. Using the linearized model in equation (2.78) while keeping in mind that the function is linearized about the expected values, the priori estimate can be expressed by:

$$\boldsymbol{x_k^-} = E[\boldsymbol{x_{k-1}^+}] \approx \boldsymbol{f}(\boldsymbol{x_{k-1}^+}, \boldsymbol{u_{k-1}}, \boldsymbol{0}) \tag{2.80}$$

where the superscription $^-$ and $^+$ denotes the priori and posteriori estimates, respectively. Following the same for the priori estimate of the covariance along with the assumption of the process noise enters the system directly. As a consequence, the Jacobian of the non-linear function with respect to w becomes one. In addition, as the process noise is assumed to be normally distributed Gaussian white noise with a constant covariance, the $\boldsymbol{Q}$ matrix does not change for each iteration. The priori estimate of the covariance matrix is expressed as (derivation of the expression can be found in appendix B):

$$\boldsymbol{P_k^-} = \boldsymbol{J_x} \boldsymbol{P_{k-1}^+} \boldsymbol{J_x^\top} + \boldsymbol{Q} \tag{2.81}$$

With the $\boldsymbol{Q}$ being unknown in practice, it becomes a tuning parameter for the EKF. Choosing small values for the diagonal elements of $\boldsymbol{Q}$ implies that the estimated is of high accuracy and small variations are expected. In contrast, choosing large values implies that large variations is expected, and that the estimate is of poor accuracy.

For the posteriori estimate, the idea is to produce a more accurate estimate using the last available measurement. As the priori estimate is purely based on the previous control inputs and estimated states, it does not take into account unmodelled dynamics of the actual plant. As a consequence, there will be an error between the actual measurement of the system and the predicted measurement. To solve this problem the error in the actual and predicted measurement is multiplied with a corrective factor and added to the priori state estimation. Using the linearized model in equation (2.79) for the measurement about the current priori estimate, the posteriori estimate can be expressed as:

$$\boldsymbol{x_k^+} = \boldsymbol{x_k^-} + \boldsymbol{K_k} \left(\boldsymbol{y_k} - \boldsymbol{h}(\boldsymbol{x_k^-}) \right) \tag{2.82}$$

where $\boldsymbol{K_k}$ is the corrective factor matrix also known as the Kalman gain. In terms of least square error minimization, the Kalman gain is the matrix that minimizes the sum of the squared errors. For linear cases the Kalman gain is optimal[39].

As for non-linear cases the Kalman gain is only suboptimal. There are iterative methods for reducing the linearization error, however, such method will not be implemented in this thesis. The derivation of the Kalman gain is cumbersome and yields little to no additional intuition on how the Kalman gain minimizes the error. Therefore, the Kalman gain is expressed in its simplest form (for derivation see appendix B):

$$\boldsymbol{K_k} = \boldsymbol{P_k^-}\,\boldsymbol{J_h^\top}\left(\boldsymbol{J_h}(\boldsymbol{x_k^-})\boldsymbol{P_k^-}\,\boldsymbol{J_h^\top}(\boldsymbol{x_k^-}) + \boldsymbol{R}\right)^{-1} \tag{2.83}$$

In the same manner as for $\boldsymbol{Q}$, the covariance matrix of the measurement $\boldsymbol{R}$ can be used as a tuning parameter. However, the covariance of the measurement can be obtained by measuring the process about a stationary operating point. The same rules applies to $\boldsymbol{R}$ as for $\boldsymbol{Q}$, with large elements implies large expected variations and vice versa. Finally, the posteriori estimate of the covariance is needed. As for the Kalman gain, the derivation of the posteriori estimate of the covariance matrix can be found in appendix B. The posteriori estimate of covariance matrix can be found by:

$$\boldsymbol{P_k^+} = \left(\boldsymbol{I} - \boldsymbol{K_k}\boldsymbol{J_h}(\boldsymbol{x_k^-})\right)\boldsymbol{P_k^-} \tag{2.84}$$

Figure 2.10 is an illustration of how the EKF estimates the states in parallel with the physical plant. The plant block should not be confused by the non-linear model f, as the plant represents the exact dynamics of the process which is unknown, whereas f is only an approximation. The computation of the Kalman gain and the posteriori estimate of the covariance matrix is omitted as the purpose of the figure is to illustrate how the EKF computes the estimates based upon the control inputs, previous estimates and available measurements in its simplicity. For a more detailed block diagram, see the work by Terejanu et al. in [40]. However, the figure shows the very essence of the Kalaman filter, which is that the state estimation based upon the approximated model is smoothed by using the available measurement of the physical plant.

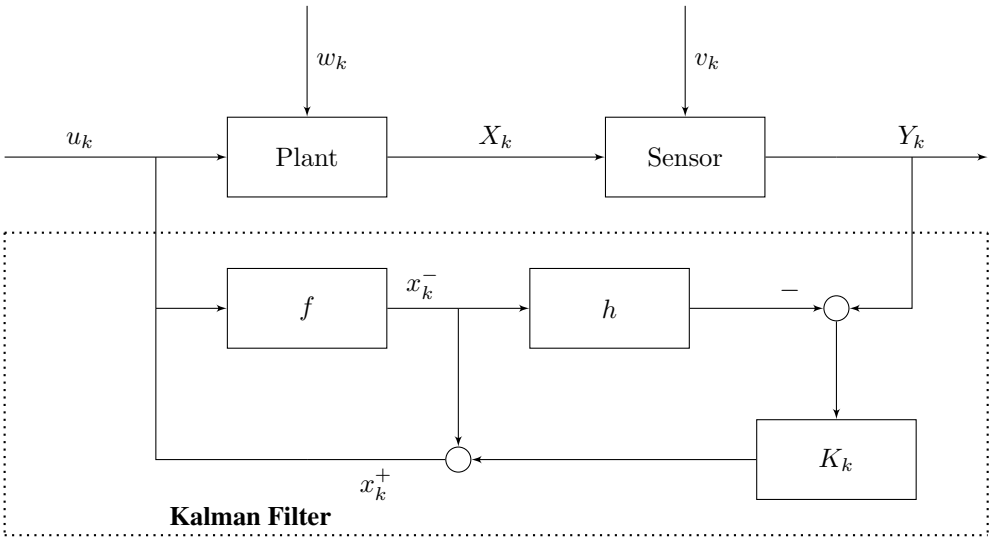


Figure 2.10: Simplified block diagram of the plant with the extended Kalman filter implemented.

2.4.1 EKF - Algorithm

Based upon the previous elaboration, the EKF can summarized as a iterative algorithm as followed: Initialize:

$$\begin{aligned} \boldsymbol{x_0^+} &= E[\boldsymbol{x_0}] = \boldsymbol{x_0} \\ \boldsymbol{P}_0 &= E[(\boldsymbol{x_0} - \boldsymbol{x_0^+})(\boldsymbol{x_0} - \boldsymbol{x_0^+})^\top] \end{aligned} \tag{2.85a}$$

Priori estimates:

$$\boldsymbol{x_k^-} \approx \boldsymbol{f}(\boldsymbol{x_{k-1}^+}, \boldsymbol{u_{k-1}}) \tag{2.85b}$$

$$\boldsymbol{P}_k^- = \boldsymbol{J_f}(\boldsymbol{x_{k-1}^+})\boldsymbol{P_{k-1}}\boldsymbol{J_f^\top}(\boldsymbol{x_{k-1}^+}) + \boldsymbol{Q} \tag{2.85c}$$

Posteriori estimates:

$$\boldsymbol{K_k} = \boldsymbol{P_k^-}\boldsymbol{J_h^\top}\left(\boldsymbol{J_h}(\boldsymbol{x_k^-})\boldsymbol{P_k^-}\boldsymbol{J_h^\top}(\boldsymbol{x_k^-}) + \boldsymbol{R}\right)^{-1} \tag{2.85d}$$

$$\boldsymbol{P_k^+} = \left(\boldsymbol{I} - \boldsymbol{K_k}\boldsymbol{J_h}(\boldsymbol{x_k^-})\right)\boldsymbol{P_k^-} \tag{2.85e}$$

$$\boldsymbol{x_k^+} \approx \boldsymbol{x_k^-} + \boldsymbol{K_k}\left(\boldsymbol{y_k} - \boldsymbol{h}(\boldsymbol{x_k^-})\right) \tag{2.85f}$$

2.5 Optimization

In everyday practice, achieving the best possible outcome is always desirable, whether it is maximizing the total income of a business, saving as much fuel as possible for a space rocket or controlling a process in the best possible way. As a result, optimization theory is widely adopted in a broad collection of industries, such as; economics, science and medicine to determine optimal conditions for specific problems. Although optimization can be seen as a relatively young in terms of technology, optimization has and always will occur in nature through evolution. As for example, the shape of the body of fish has throughout history evolved to reduce the aerodynamic forces which allow the fish to swim more efficiently, thus reduces the energy needed for moving.

At the very heart of every optimization problem is the objective at which the engineer want to optimize. The objective can be minimizing the time it takes to drive through a city, reducing the heat needed for a process or simply finding a minima of a polynomial. Mathematically, the objective can take many forms and in reality two problem will never be the same. However, problems are classified as either linear, quadratic or non-linear. In addition, the objective are in most cases limited by given conditions, referred to as constraints. As a consequence, one distinguish between unconstrained and constrained optimization problem. In practice, all optimization problem is limited by some constraints to some degree. The constraints restricts the optimization at reaching the same optimal point as for an unconstrained problem. Figure 2.11 is an illustration on how the minima of the objective function $f(x) = x^2$ is limited by the constraint $c(x) \geq 10 - 5x$. Here, $c(x)$ is an inequality constraint, which in essence means that the optimal point can be anywhere as long as it is greater than $c(x)$. However, it could be a equality constraint, which again implies that the optimal optimal point has to equal $c(x)$. In this particular case, it does not matter whether $c(x)$ is an inequality or equality constraint as the minima lies at the interception of $f(x)$ and $c(x)$. Naturally, an objective can be restricted by a combination of several constraints, which can be both inequality and equality constraints. In addition, constraints can be constants, linear or non linear.

As the optimization problem comes in varies forms, naturally, there exists numerous different methods for solving optimization problems. Popular method such as the simplex method is well suited for solving linear optimization problems. For non linear problems, a class of methods known interior-points methods can be used. However, in this thesis, a sequential quadratic program (SQP) method is applied for solving the non-linear constrained optimization problem.

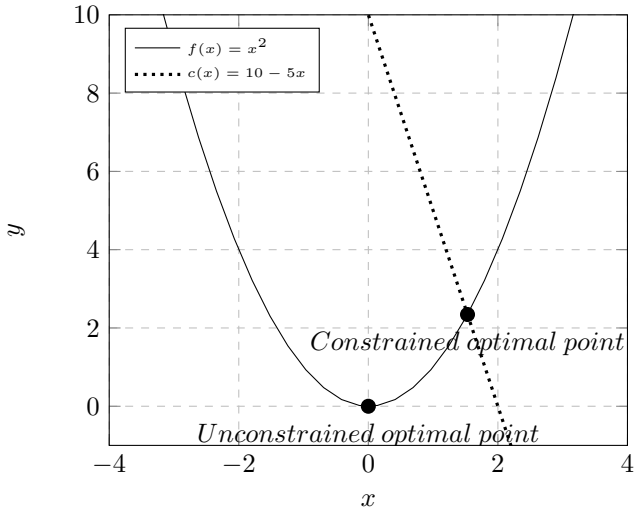


Figure 2.11: Illustration of how constraints limits the optimal (minima) conditions.

The SQP method for solving non-linear optimization problems is an iterative method at which the objective and constraint function are twice continuously differential. It involves using Lagrangian multipliers to design a quadratic subproblem and then using the respective minimizer of the subproblem to define the next iteration. The derivation of the method is to a large extent highly theoretical and is not emphasized in this thesis. For a comprehensive elaboration of the SQP method, see the work by Nocedal and Wright in [41]. Instead, figure 2.12 shows a simplistic flowchart of the SQP algorithm. The algorithm starts by initializing the optimization vectors, as well as other variables before it progresses by solving the subproblem. If the constraints are violated, the optimization variable is updates and the subproblem is recomputed. If the solution to the subproblem satisfy the constraints, the algorithm progresses by checking whether or not the solution has converged. This is mainly done by checking if the 2-norm of previous and current iteration has decreased below a certain threshold. If so, the current iteration is accepted as the solution to the optimization problem.

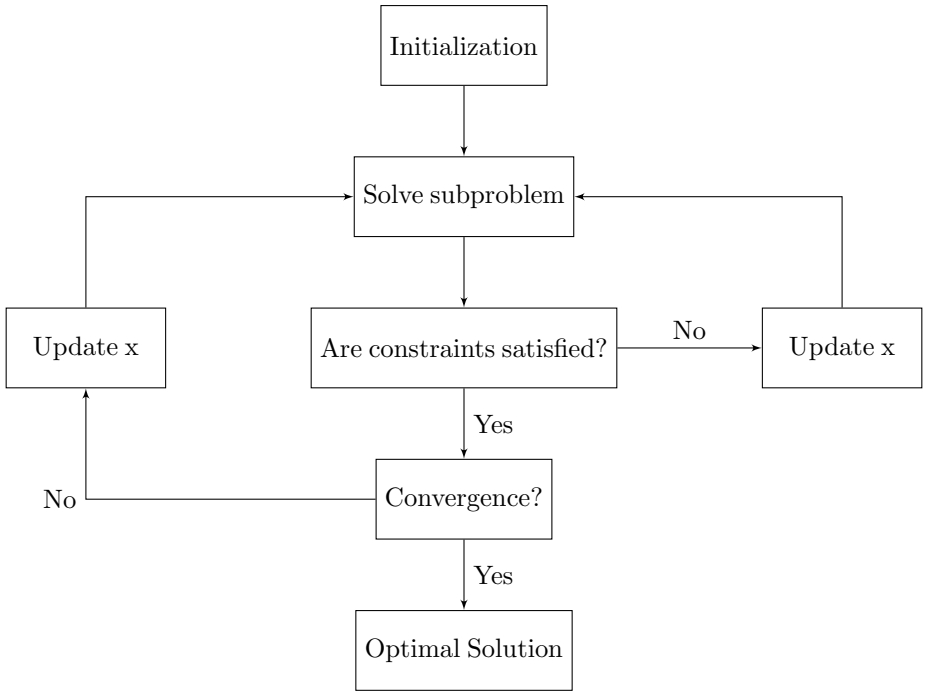


Figure 2.12: Simplistic flowchart of the SQP-method.

As the derivation of the SQP method is omitted in the thesis, the general formulation of an optimization problem as stated by Jongen et al. in [42]:

$$\arg\min_{x \in \mathbb{R}^n} f(x) \tag{2.86a}$$

such that

$$c_i(x) = 0, \quad i \in \mathcal{E} \tag{2.86b}$$

$$c_i(x) \geq 0, \quad i \in \mathcal{I} \tag{2.86c}$$

where $f(x)$ is the objective function, x is the optimization variable and $c_i(x)$ is the constraint functions. $\mathcal{E}$ and $\mathcal{I}$ denotes the set which contains the equality and inequality constraints, respectively.

2.6 Control Theory

In many situations it is desirable to keep process variables within a close proximity of a reference value. The reasons are many, some examples may be to increase the throughput of a process, increase the performance of a product or keep a machine from operating at condition which may cause rapid degeneration. Process control is found in one form or another in every system, and with the increasing development in computational power, control systems becomes more sophisticated and advanced for each day.

2.6.1 Laplace Transform and Scaling

To avoid repeating already presented theory, a dynamic system used for control purposes are generally linearized and expressed using a state-space model as in equation (2.68). Usually, a process is affected by some disturbance, d, of one or another form. Disturbances are added to the state-space model either by adding a separate term or by treating it as an input. In this thesis both approaches is taken depending on which is more convenient for the given situation. It is important to note that the state-space model is expressed in deviation variables from the point at which the dynamic model is linearized about. Hence, the model is limited to operating conditions in close proximity of the linearization point. As the perturbations may be large during start up of a process, a common approach in control engineering is set the controllers to manual until the process reaches the operating point. A soon as the process reaches its operating point, the controllers are set to automatic and the process is controlled based upon a linearized model [37]. Signals that enters the system are rarely constant (with reference signals being an exception) and varies over time. As the signal may be changing slowly or fast with time, it is difficult to asses how the system response to signals of every form. Rather than analysing the systems response in the time domain, the system is converted to the s-domain (Laplace-domain), where s is the complex variable $j\omega$. Consider the linearized state-space model:

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{E}\boldsymbol{d} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} + \boldsymbol{F}\boldsymbol{d}\end{aligned}$$

Using the rules of typical Laplace transforms as found in the table in [43] by Murray, the state-space model can be converted to the s-domain as follows:

$$\begin{aligned}s\boldsymbol{x}(\boldsymbol{s}) + \boldsymbol{x}(\boldsymbol{0}) &= \boldsymbol{A}\boldsymbol{x}(\boldsymbol{s}) + \boldsymbol{B}\boldsymbol{u}(\boldsymbol{s}) + \boldsymbol{E}\boldsymbol{d}(\boldsymbol{s}) \\ \boldsymbol{y}(\boldsymbol{s}) &= \boldsymbol{C}\boldsymbol{x}(\boldsymbol{s}) + \boldsymbol{D}\boldsymbol{u}(\boldsymbol{s}) + \boldsymbol{F}\boldsymbol{d}(\boldsymbol{s}) \\ \boldsymbol{y}(\boldsymbol{s}) &= \left(\boldsymbol{C}\left(s\boldsymbol{I} - \boldsymbol{A}\right)^{-1}\boldsymbol{B} + \boldsymbol{D}\right)\boldsymbol{u}(\boldsymbol{s}) + \left(\boldsymbol{C}\left(s\boldsymbol{I} - \boldsymbol{A}\right)^{-1}\boldsymbol{E} + \boldsymbol{F}\right)\boldsymbol{d}\end{aligned}$$

which is later reduced to:

$$\boldsymbol{y}(\boldsymbol{s}) = \boldsymbol{G}(\boldsymbol{s})\boldsymbol{u}(\boldsymbol{s}) + \boldsymbol{G_d}(\boldsymbol{s})\boldsymbol{d}(\boldsymbol{s}) \tag{2.87}$$

Figure 2.13 is a block diagram of the Laplace transformed system. This block diagram is later extended with controller and feedback.

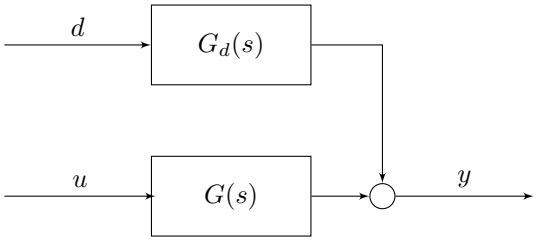


Figure 2.13: Block diagram of the Laplace transformed system.

With the magnitude of the transfer functions varying from system to system, it is difficult to apply general measures of performance of the control system. Rather than representing the system with its original values, it is common to scale the system such that the maximum acceptable deviation from the reference value, available control input as well as expected disturbances is all set to 1. This does not only makes the control system easier to understand for someone without any knowledge of the system, but also enables the application of general measurement of performance and robustness. The scaled variables can be defined as:

$$\hat{u} = \frac{u}{u_{max}}, \quad \hat{y} = \frac{y}{y_{max}}, \quad \hat{d} = \frac{d}{d_{max}} \tag{2.88}$$

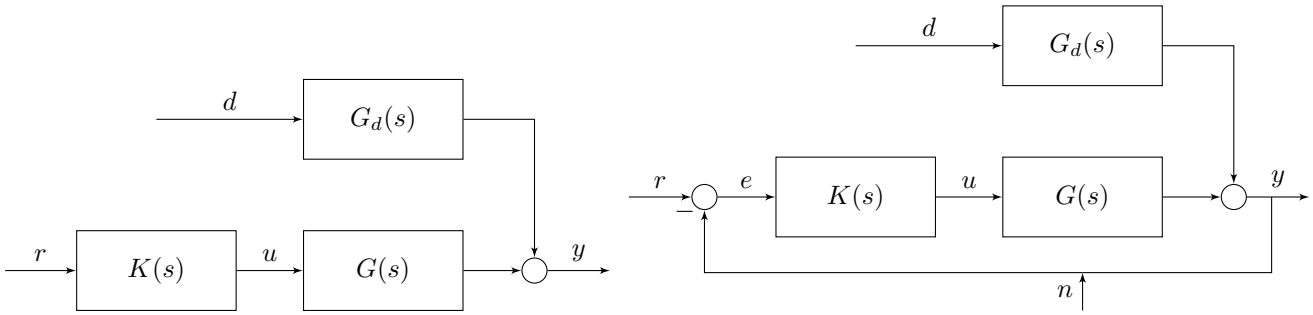
For multivariable systems the scaling is achieved by diagonal scaling matrices, with each diagonal element being the appropriate scaling constant for the corresponding input/output/disturbance. On matrix form, the scaled system can be expressed as:

$$\begin{aligned} \boldsymbol{W_y}\hat{\boldsymbol{y}}(\boldsymbol{s}) &= \boldsymbol{G}(\boldsymbol{s})\boldsymbol{W_u}\hat{\boldsymbol{u}}(\boldsymbol{s}) + \boldsymbol{G_d}(\boldsymbol{s})\boldsymbol{W_d}\hat{\boldsymbol{d}}(\boldsymbol{s}) \\ \hat{\boldsymbol{y}}(\boldsymbol{s}) &= \hat{\boldsymbol{G}}(\boldsymbol{s})\hat{\boldsymbol{u}}(\boldsymbol{s}) + \hat{\boldsymbol{G}_d}(\boldsymbol{s})\hat{\boldsymbol{d}}(\boldsymbol{s}) \end{aligned} \tag{2.89}$$

where $\boldsymbol{W_y}$, $\boldsymbol{W_u}$ and $\boldsymbol{W_d}$ is the scaling matrices for the output, input and disturbances, respectively. The hat notation is from here omitted for simplicity, hence, when presenting transfer function later on it is assumed that it is scaled.

2.6.2 Feedback Control

In general, it is distinguished between two types of control loops; open and closed loop control. In open loop control structures, the corrective input signal is calculated by a controller independently of the process output. As a consequence, the controller has no information about how the output is behaving. Regardless of the output growing out of bounds or oscillating, the controller can not adjust the control input to correct the behavior of the output. What causes the unwanted behavior of the output can be for example uncertainties in the process model, changes in the disturbance or the process being unstable. Thus, the applicability of an open loop control structure is limited. Figure 2.14a shows a block diagram of an open loop control system with $K(s)$ being the controller. It is clear from the illustration that the controller has no knowledge of the output and calculates what it "thinks" is the best control input based purely on the reference input. In contrast, in a closed loop control structure, the output is fed back to the controller which allows it to use the information of the output to calculate a corrective control input. This enables the controller to adjust for changes in the output caused by various reason. It can correct for model uncertainties and changes in the disturbance. It also have the ability to stabilize an unstable process. These are some of the advantages of feedback control (closed loop) and makes it, by far, the most applied control structure. Figure 2.14b is a block diagram with feedback implemented. Here r, e and n represents the reference value, error between the reference and the actual output and the measurement noise from the sensor used to obtain the measurement, respectively.



(a) Open loop control structure.

(b) Closed loop control structure.

Figure 2.14: Block diagram of open loop and closed loop control structure.

The idea of feedback control is to minimize the error e. There are many different approaches for minimizing the error, e.g. using P/PI/PID-controllers, H_2 - optimal control or MPC. Which controller that fits best depends on the process and its complexity. However, P/PI/PID-controllers are in most situations sufficient for achieving desirable control performance [44]. Naturally, testing and tuning the controller whilst the process is operating is not reasonable as it may cause damage to components. Thus, it is desirable to assess how different signal affects the error prior to implementing the control system on the actual process. Using the block diagram of a feedback control structure in figure 2.14b to derive an expression for the error e yields:

$$e = \big(I + G(s)K(s)\big)^{-1} G_d(s)d(s) - \big(I + G(s)K(s)\big)^{-1} r(s) - \big(I + G(s)K(s)\big)^{-1} G(s)K(s)n(s)$$

This can be further reduced to:

$$e = S(s)G_d(s)d(s) - S(s)r(s) - T(s)n(s) \tag{2.90}$$

$S(s)$ and $T(s)$ is referred to as the sensitivity function and the complementary sensitivity function. By intuition, the magnitude of S and T amplifies the disturbance and noise signals influence on the error. To be precise, the sensitivity and complementary sensitivity function tells how much the disturbance and noise signal is affected by the feedback, respectively. Naturally, the best option would be to make both S and T as low as possible. However, there is a trade off as S and T is related by:

$$S + T = 1 \tag{2.91}$$

This implies that by reducing either one results in an increase in the other. The term "waterbed effect" is often used when explaining the relation of S and T. As a consequence, rejecting both disturbances and measurement noise with the same frequency is impossible. However, as most disturbances is signals of relative low frequency compared to the frequency of measurement noise, controllers are designed such that the magnitude of S is small at low frequencies and larger for higher frequencies. However, large values in both S and T implies poor control at the respective frequency. Thus, it is of interest to keep the values of S and T as low as possible. In fact, the peak values of S and T is used as a measure of control performance. The peak values are defined as:

$$M_S = \max_{\omega} \left|S(j\omega)\right| \tag{2.92}$$

$$M_T = \max_{\omega} \left|T(j\omega)\right| \tag{2.93}$$

A similar definition using the largest singular value is applicable for multi-input-multi-output system (MIMO). However, MIMO systems is not considered in this thesis. For strictly proper systems, M_S will always be great or equal to 1 with $M_S \leq 2$ being considered acceptable[37]. Similarly, M_T will always be greater or equal to one with $M_T \leq 1.3$ being considered acceptable[37].

Designing the controller such that S is small for low frequencies and larger at higher frequencies implies that the controllers ability to reject disturbances drops off as the frequency increases. The frequency range at which the controller is effective is referred to as the bandwidth frequency. In most situations the lower limit of the bandwidth is at zero, resulting in the bandwidth being referred to the frequency at which the controllers stops being effective. Considering that S tells how much the disturbances is amplified by the feedback, it indicates when the effect of the controller start to drop off. Hovd defines in [37] the bandwidth frequency for single-input-single-output system (SISO) as:

$$\omega_b \rightarrow \left|S(j\omega)\right| = \frac{1}{\sqrt{2}} \tag{2.94}$$

Ideally, one want to the bandwidth to be as large as possible to increase the range at which the controller is effective. However, there are several factor which restricts the bandwidth. One factor is time delays which puts a major limitation on the bandwidth. As a consequence, processes with large internal/input/output delay is inherently difficult to control. In practice, every system has a time delay to some extent.

2.6.3 Loop Shaping

There are several ways for tuning controllers, e.g. Ziegler-Nichols method, step-response fitting or IMC-tuning. Loop shaping is a technique for tuning the controller in which the magnitude of a Bode-plot is shaped such that desirable closed loop performances is achieved. A Bode-plot is a representation of the magnitude and phase as a function of frequency of a transfer function. The magnitude is either represented in dB or absolute values. In this thesis the latter representation is used. As Bode-plot is mainly a graphical representation of the transfer function, it will not be further explained.

A prerequisite for applying the rules of loop shaping is that the system is appropriately scaled as previously mentioned. The idea is to plot the frequency response of GK, G_d, S and T to assess control performance and stability. Within the bandwidth, it is desirable that the controller has the ability cancel out any disturbances and track the reference. Furthermore, to minimize the influence of the feedback on the measurement noise the magnitude of the open loop gain should be as low as possible for frequencies above the bandwidth. In summary, it is required that:

- For disturbance rejection: $||GK|| \gg ||G_d||$ for frequencies below the bandwidth.
- For reference tracking: $||GK|| \gg ||r||$ for frequencies below the bandwidth.
- For low the sensitivity to measurement noise: $||GK|| \ll 1$ for frequencies above the bandwidth.

In addition, M_T and M_S should be as low as possible and at least as small such that the inequalities previously mentioned is fulfilled. The stability of a system can be determined in the Bode-plot by looking at the frequency at which $|GK| = 1$, known as the crossover-frequency, ω_c. The system is stable as long as $|GK|$ remains less than 1 for all frequencies above ω_c provided that the phase at ω_c is larger than -180°. Furthermore, the gain and phase margin are terms which indicates how far a system are from being unstable. The gain and phase margin are defined as:

$$GM = \frac{1}{|GK(j\omega_{180}|)} \tag{2.95}$$

$$PM = 180^\circ + \angle GK(j\omega_c) \tag{2.96}$$

where ω_{180} is the frequency at which the phase for GK equals $-180°$. To maintain robustness form model uncertainties, it is recommended that $GM \geq 2$ and $PM \geq 60°$ [44]. However, GM and PM does not guarantee good control performance.

Figure 2.15 shows a Bode-plot of an arbitrary system with the terms mentioned above pointed out. When tuning, a initial guess for a controller is used when plotting the Bode-plot. If the requirement mentioned above is not satisfied, the controller is tuned and the requirement is checked again. This process is repeated until the system shows satisfying performance.

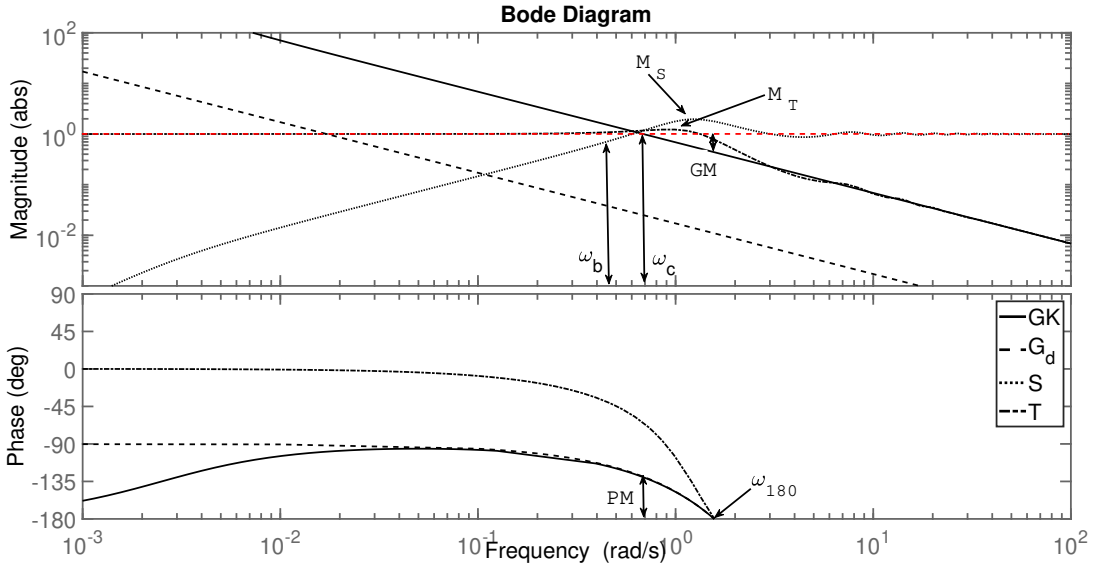


Figure 2.15: Example of a Bode-plot with performance indications marked.

2.6.4 Anti-Windup

In addition to the requirements mentioned in the previous section, the usage of the control input also has to be considered. For an ideal controller, it can stabilize and control a system as it can produce control inputs of unlimited magnitude. In practice, control input is bounded and large inputs may cause saturation. For controllers consisting purely of a proportional term this does not cause any problem except limiting the input implemented in the system. However, for controllers with a integrator, saturation causes an additional unwanted effect. In situations when the input is saturated, the integral term in the controller keeps growing and "winds up" in an attempt to compensate for the error in the output. When the output has been brought back near the reference value, the input value is still affected by the integrator adding to the control input during the saturation. To prevent wind up effects from happening, anti-windup schemes are implemented in the controller. For a PI/PID controller this is achieved by having the implemented control input fed back through a low pass filter. Figure 2.16 shows the block diagram of a simple anti-windup implementation for a PI controller as stated in [37]. Here the integration term is implemented in the low pass filter.

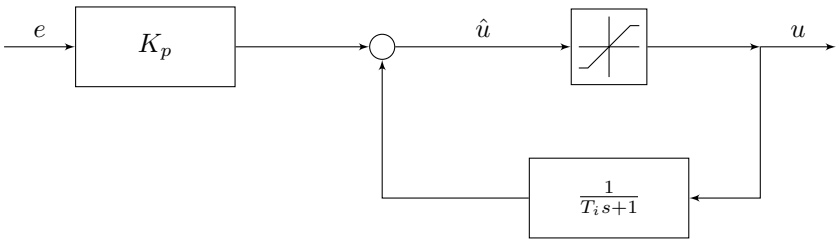


Figure 2.16: Implementation of anti-windup for a PI-controller.

Chapter 3

Model Development and Implementation

The work done in this thesis was carried out in Matlab. Thus, a majority of the implementation of the previous discussed topics was necessary. This chapter is structured with the same substructure as the previous chapter. The different implementations and simulations are explained, and important code snippets are presented in the chapter.

3.1 Implementation of the Numerical Method

Due to equation (2.17a) - equation (2.17d) being implicit, the equations has to be solved simultaneously. As a consequence, prior to implementing the numerical method in Matlab, the equations are arranged in a matrix from with $\boldsymbol{\Theta}$ and $\frac{\boldsymbol{dX}}{\boldsymbol{dt}}$ being the vector with the thickness, temperature and density states and differential terms, respectively:

$$\boldsymbol{\Theta} = \begin{bmatrix} L \\ T \\ \rho_f \end{bmatrix}, \quad \frac{\boldsymbol{dX}}{\boldsymbol{dt}} = \begin{bmatrix} \frac{dL}{dt} \\ \frac{dT}{dt} \\ \frac{d\rho_f}{dt} \end{bmatrix}$$

In matrix form, the system of equations can be represented as:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \frac{\boldsymbol{dX}}{\boldsymbol{dt}} = \begin{bmatrix} 0 & 0 & -\frac{L}{\rho_f} \\ -\frac{T}{L} & 0 & -\frac{T}{\rho_f} \\ w & 0 & 0 \end{bmatrix} \cdot \frac{\boldsymbol{dX}}{\boldsymbol{dt}} + \begin{bmatrix} -h \\ -g + \dot{q} \\ v \end{bmatrix}$$

$$\frac{\boldsymbol{dX}}{\boldsymbol{dt}} = \begin{bmatrix} 1 & 0 & \frac{L}{\rho_f} \\ \frac{T}{L} & 1 & \frac{T}{\rho_f} \\ -w & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -h \\ -g + \dot{q} \\ v \end{bmatrix} \tag{3.1}$$

where h, g, $\dot{q}$ and v are the terms in equation (2.17a)-equation (2.17d) which do not contain any of the derivatives. The derivation of equation (3.1) and its terms can be found in appendix C. Furthermore, $\frac{\boldsymbol{dX}}{\boldsymbol{dt}}$ is a function of $\boldsymbol{\Theta}$ and applying the Runge-Kutta method as stated in equation (2.63) with the system written on the form as in equation (3.1) yields the following iterative algorithm:

$$\boldsymbol{k_1} = \frac{\boldsymbol{dX}}{\boldsymbol{dt}}(\boldsymbol{\Theta_n}) \tag{3.2a}$$

$$\boldsymbol{k_2} = \frac{\boldsymbol{dX}}{\boldsymbol{dt}}\left(\boldsymbol{\Theta_n} + \frac{\Delta t}{2}\boldsymbol{k_1}\right) \tag{3.2b}$$

$$\boldsymbol{k_3} = \frac{\boldsymbol{dX}}{\boldsymbol{dt}}\left(\boldsymbol{\Theta_n} + \frac{\Delta t}{2}\boldsymbol{k_2}\right) \tag{3.2c}$$

$$\boldsymbol{k_4} = \frac{\boldsymbol{dX}}{\boldsymbol{dt}}\left(\boldsymbol{\Theta_n} + \Delta t\boldsymbol{k_3}\right) \tag{3.2d}$$

$$\boldsymbol{\Theta_{n+1}} = \boldsymbol{\Theta_n} + \frac{1}{6} \cdot (\boldsymbol{k_1} + 2\boldsymbol{k_2} + 2\boldsymbol{k_3} + \boldsymbol{k_4}) \tag{3.2e}$$

The function listed in listing 3.1 finds the gradient k_i at each iteration for the Runge-Kutta method. Furthermore, listing 3.2 lists the overall implementation of the numerical method with the combination of the Runge-Kutta method and the simplified Newton-Raphsons method derived from the FEM method in section 2.2. The main script can be found in appendix D.1 in appendix D.

```
function k = RungeKutta(p,t,v,w,K)
    %-------------------------------------------------------------------------%
    %------------Calculates the slopes for Runge-Kutta Method-----------------%
    %-------------------------------------------------------------------------%

    %% Initialization
    qin = p.k_c*(p.Ta(t)-v(2)*p.T0);
    h_fg = 6.991*(v(2)*p.T0)^2-6193*(v(2)*p.T0)+1.985e06;
    D = D_Coefficient(w.*p.C0,v(2)*p.T0,p);
    Ps = p_sol(w(end)*p.C0, v(2)*p.T0,p);

    %% Solves the Implicit Function dXdt(Theta + delta t*k_n)
    Q = D.*K; b = p.D0/((v(1)*p.L0)^2);
    q = (p.rho_s*p.rho_p*(p.rho_p-p.rho_s)*p.C0)/(p.rho0*(w(end)*p.C0*(p.rho_s-p.rho_p)+p.rho_p)^2);
    B = [-(p.k_m*p.Ms)/(v(3)*p.rho0*p.R*p.L0)*(Ps/(v(2)*p.T0)-p.Pa(p.Ta(t))/p.Ta(t));
        -(p.k_m*p.Ms*h_fg)/(v(3)*p.rho0*p.Cp*p.R*v(1)*p.L0*p.T0)*(Ps/(v(2)*p.T0)-p.Pa(p.Ta(t))/p.Ta(t))+qin/(v(3)*p.rho0*p.Cp*p.T0*v(1)*p.L0);
        -q*(p.k_m*p.Ms)/(p.R*p.rho0*v(1)*p.L0)*(Ps/(v(2)*p.T0)-p.Pa(p.Ta(t))/p.Ta(t))-q*b*Q(end,end-2:end)*w(end-2:end)];
    A = [1 0 v(1)/v(3); v(2)/v(1) 1 v(2)/v(3); q*v(3)/v(1) 0 1];

    k = A\B;
end
```

Listing 3.1: Implementation of the Runge-Kutta gradient function in Matlab

```
function [x, DD] = solveDryingSystem(p,t,x0,M,K)

%-------------------------------------------------------------------------%
%-------------------Solves the drying system with FEM---------------------%
%--------------------------Dimensionless Model----------------------------%
%-------------------------------------------------------------------------%

x = zeros(length(p.z)+3,length(t)); x(:,1) = x0; eps = 1e-04; max_iter = 20; DD = zeros(length(p.z),length(t));
for i = 1:length(t)-1

    %% The Runge-Kutta Method for Solving the ODEs
    k1 = RungeKutta(p,i*p.dt,x(1:3,i),x(4:end,i),K);
    k2 = RungeKutta(p,i*p.dt,x(1:3,i)+p.dt*k1/2,x(4:end,i),K);
    k3 = RungeKutta(p,i*p.dt,x(1:3,i)+p.dt*k2/2,x(4:end,i),K);
    k4 = RungeKutta(p,i*p.dt,x(1:3,i)+p.dt*k3,x(4:end,i),K);

    x(1:3,i+1) = x(1:3,i)+1/6*p.dt*(k1+2*k2+2*k3+k4);

    %% The Simplified Newton-Raphson's Method for Solving the Finite Element Approximation
    j = 1; c_next = x(4:end,i); dc = 1; b = p.D0/((x(1,i)*p.L0)^2);
    R = zeros(size(M)); F = zeros(length(p.z),1);
    while max(dc) > eps && max_iter > j
        D = D_Coefficient(c_next.*p.C0,x(2,i)*p.T0,p); Q = D.*K;
        Ps = p_sol(c_next(end)*p.C0, x(2,i)*p.T0,p);
        R(end) = k1(1)/x(1,i);K_star = -(b*Q+R); F(end) = -(p.k_m*p.Ms)/(x(3,i)*p.rho0*p.R*x(1,i)*p.L0*p.C0)*(Ps/(x(2,i)*p.T0)-p.Pa(p.Ta(0))/p.Ta(0));
        A = -(M-p.dt*K_star); B = -M*x(4:end,i)-p.dt*F;
        c_prev = c_next; c_next = A\B; dc = norm(c_prev-c_next); j = j+1;
    end
    x(4:end,i+1) = c_next; DD(:,i) = D;
end
end
```

Listing 3.2: Implementation of the numerical method in Matlab

3.2 Linearization and Linear Analysis

The linearization is, as mentioned, an attempt to simplify the control and optimization problems by converting the non-linear system into a linear system. The implementation in Matlab is done in the order which the linear system theory is presented. In addition, step responses are included in the implementation to assess whether or not the linear system reacts to excitations in the control inputs in the same way as the non-linear model.

3.2.1 Linearization

To begin with, linearization done analytically would be time consuming and mistakes would easily been made due to the complexity of the system. To simplify the work and to prevent any human mistakes, the model is implemented in Matlab using the Symbolic Toolbox. Thus, the model is represented by symbolic variables which can take any value. However, the symbolic variables are restricted to real values as complex values does not occur in practice for the chosen variables. A snip out of the linearization function in Matlab which illustrates the declaration of the symbolic variables and the variable dependent parameters is shown in listing 3.3. The complete `Linearization.m` is found in appendix D.2.

```
%% Declaring the Symbolic Variables
c = sym('c', [length(p.z) 1], 'real');
syms L T rho Ta V real;

%% Declaring the Parameters in Terms of the Symbolic Variables
Ps = p_sol(c(end)*p.C0,T*p.T0,p);
D = D_Coefficient(c*p.C0,T*p.T0,p);
h_fg = 6.991*(T*p.T0)^2-6193*T*p.T0+1.848e06;
Pa = p.rh*133.332*10.^(7.54826-1979.68./(222.2+(Ta-273)));
[k_c , k_m ] = TransferCoefficients(V);
qin = k_c*(Ta-T*p.T0);
```

Listing 3.3: Declaration of the symbolic variables

Furthermore, the Jacobian is found using Matlabs built in function `jacobian`. The Jacobian is first found for the density, thickness and temperature, and then the solvent fractions. This is purely a result of the procedure of the numerical approach, and could, in hindsight, be made more efficient by only calculating the Jacobian of the whole system. Anyway, the Jacobians are evaluated at the final values of the non-linear model using the `subs` function which substitutes the chosen symbolic variables with predefined values. This is realized by solving the non-linear system prior to the linearization procedure. However, this does require that the drying time is sufficiently long such that stationary conditions for the non-linear system is achieved. The state-space matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are assembled by adding the respective elements from the Jacobians in the correct order of the state vector. The state vector is always defined as in the numerical method with T_a and V being added at the end depending on if they are chosen as control inputs or not. In appendix D.2, both T_a and V is used as control inputs, thus the elements of the Jacobians which contains the terms with T_a and V is placed in matrix $\boldsymbol{B}$. As for now, there is no automatic procedure to form the matrices based upon the chosen control inputs, nor for the measurement matrix $\boldsymbol{C}$, so this has to be done manually. The system is then converted to a state-space system using Matlab's built in function `ss`, where $\boldsymbol{D}$ is set to zero as the measurement of interest is the states them self. The main reason for using the `ss` function is because it allows the use for the function `c2d` which converts the continuous state-space model into a discrete state-space model with a given time sample. By default, the `c2d` function uses the Zero-Order-Hold method to discretize the model, which is explained in equation (2.71).

The system is then solved for the predefined simulation time with the discrete model using a simple for-loop as shown in the snip out of `Linearization.m` in listing 3.4. Here, the control input vector remains constant throughout the simulation time. This is the case as the non-linear system is solved with constant control inputs. If one of the control inputs is changed during the simulation of the non-linear system, $\boldsymbol{u}$ as to be change accordingly. This has not been implemented in the current code, but can easily be added by storing the values of the control inputs for the non-linear system and sending the control input vector as an argument for the `Linearization.m` function. Note that the linearized model is expressed in terms of deviation variables, so the control input data for the non-linear model has to be converted to deviation variables from the operating point in order to get the correct values.

```
%% Solving the Linear System
Ad = sys_d.A; Bd = sys_d.B;
a = zeros(length(p.z)+3,length(t));
a(:,1) = y0;
u = u0;
for i = 1:length(t)-1
   a(:,i+1) = Ad*a(:,i)+Bd*u;
end
```

Listing 3.4: Solving the linear system for a predefined simulation time.

To account for the model being expressed by deviation variables, the solution is converted back into their original form prior to being used in post-processing. Listing 3.5 shows a snip out of the main script which converts the solution from the linearization model into its original values. $(v.\cdot)$ is the linearization point and $(p.\cdot)$ is the intial values of the problem.

```
[y_l,sys_c,sys] = Linearization(p,v,M,K,y0,u0,t);

L_l = (y_l(1,:)+v.L)*p.L0;
T_l = (y_l(2,:)+v.T)*p.T0;
rho_l = (y_l(3,:)+v.rho)*p.rho0;
C_l = (y_l(4:end,:)+v.C)*p.C0;
```

Listing 3.5: Converting from deviation variables to original variables.

The stability of both the continuous and discrete linear system is easily assessed using the function `eig(A)` which calculates the eigenvalues of the system matrix $\boldsymbol{A}$ along with the stability criteria mentioned in section 2.3.

The controllability and observability matrices is found using Matlab's built in functions `ctrb` and `obsv`. The controllability and observability is then determined accordingly to as stated in section 2.3. Listing 3.6 is a code snippet of the `Linearization` function which determines the controllability and observability.

```
%% Determining the Controllability and Observability of the Linear System
OB = obsv(Ad,C); CB = ctrb(Ad,Bd);
OX = ['Number of unobservable states: ', num2str(rank(A)-rank(OB))];
CX = ['Number of uncontrollable states: ',num2str(rank(A)-rank(CB))];
disp(OX)
disp(CX)
```

Listing 3.6: Determining the controllability and observability of the discrete system.

3.2.2 Linear Analysis

To test the linearized model, a series of test will be carried out. Firstly, the linearized model is compared to the non-linear model. This is done in straight forward manner by plotting the solution of the non-linear model versus the linearized model with the same initial values. Furthermore, the validity range of the linearized model is tested as well as how it responds to excitations in the control inputs.

For all the test, the Matlab built in function `lsim` is used. The function simulates the time response of a dynamic system with arbitrary control inputs. Thus, a control input vector must be defined. For the different tests, the control input vector takes the different values due to the purpose of the test. Listing 3.7 shows a code snippet of the main script for which the test are implemented. Note that in this particular code snippet the air temperature, T_a and air velocity,V, is chosen as control inputs.

The linearized model is tested with different initial perturbations from the operating point to assess at which point the model fails to converged to the stationary values. For each simulation the values of $y(0)_{OP}$ is changed with further increasing perturbations. The second test is simply a simulation with no excitations in the control input with the model being initialized at the linearization point. For a stable system the model should not experience any changes at all. However, in case the system is unstable, the solution will eventually grow unbounded. Finally, the linearized model is tested when the control inputs are excited. Rather than constructing the increase/decrease in control inputs as step function (heaviside funtion), the change in control inputs is constructed as a ramp function as this resembles an actual change more accurately.

```
%% Linear Simulation - Small Deviation From OP
u = zeros(2,length(t));
y0_OP = [0.05 -0.1 -0.1 ones(1,length(p.z))*0.1];
y_no = lsim(sys,u,t, y0_OP);
%% Linear Simulations - No Excitation
y_ne = lsim(sys,u,t);
%% Linear Simulations - With Excitation
t_hta = 20; step_Ta = 5;                                    % Steptime in minutes
 and stepsize for Ta
t_hv = 25; step_V = 2;                                      % Steptime in minutes
 and stepsize for V
t_hta = (t_hta*60)/p.dt;
t_hv = (t_hv*60)/p.dt;

t_ramp = 5;                                                 % Duration of the
ramp
t_ramp = (t_ramp*60)/p.dt;                                  % Converting the
duration of the ramp to fit t-vector
for i = 1:t_ramp                                            % Constructing ramp
functions for the steps in u
   u(1,t_hta+1+i) = step_Ta/t_ramp*i;
   u(2,t_hv+1+i) = step_V/t_ramp*i;                         % Linear increas in
control input
end
u(1,(t_hta+1+t_ramp):end) = step_Ta;                        % Setting the step in
 the control input for the rest of the simulation
u(2,(t_hv+1+t_ramp):end) = step_V;
y_e = lsim(sys,u,t);
```

Listing 3.7: Implmentation of the different tests of the linearized model in Matlab

3.3 State Estimation

For the state estimation, the EKF algorithm is implemented in the same manner as stated in equation (2.85). Due to lack of actual measurements, the solution of the non-linear model will be used as measurements for estimation. Thus, the EKF routine runs in parallel with the non-linear model. Listing 3.8 is a code snippet with the EKF-algorithm implemented in parallel with the numerical solution of the non-linear problem. The `SolveIter` function is the same routine as found in listing 3.2. The process and measurement noise is generated using the `wgn` function with the noise magnitude given in dBW. However, as the estimation and the measurements from the non-linear model is represented using its dimensionless variables choosing appropriate magnitudes for the noise is of importance. Normally, signals is transmitted in terms of voltage or ampere and later converted to its appropriate units through transfer functions[45].

```
%% Declartion of Non-Linear Solution Vector and Estimation Vector
x = zeros(length(p.z)+3,length(t)); a = ones(length(p.z)+3,1); x(:,1) = a; v = x;

%% Generating Measurement and Process Noise in dBW
wnm = wgn(2, length(t),-24); wnp = wgn(length(p.z)+3, length(t),-96);
cm = cov(wgn(2,1,-24)); cp = cov(wgn(length(p.z)+3,1,-96));

%% Assigning the Q, R and P_0 Matrices
Q = diag(ones(1,length(p.z)+3)); R = diag([1 1]); P_k = Q;

%% Measurement Function h
C = zeros(2,length(p.z)+3); C(1,1) = 1; C(2,2) = 1;

%% Analytical Expression of the Jacobian and Converting to Matlab Function to Avoid Symbolic
Toolbox in for-loop
[J_ff, J_h, Y] = Jacobians(p,M,K);
g = matlabFunction(J_ff, 'vars', {Y});
tic
for i = 1:length(t)-1

    %% Solution of the Non-Linear Function f
    v(:,i+1) = SolveIter(p,M,K,v(:,i),i*p.dt);
    a = SolveIter(p,M,K,a,i*p.dt) + wnp(:,i);

    %% Model Forecast Step/Predictors
    z = v(1:2,i+1)+wnm(:,i);
    J_f = g([x(:,i)' p.Ta(i*p.dt) p.V]);
    P_kf = J_f*P_k*J_f'+Q;
    %% Data Assimiliation Step/Corrector
    KK = P_kf*J_h'/(J_h*P_kf*J_h'+R);
    P_k = (eye(size(P_k))-KK*J_h)*P_kf;
    x(:,i+1) = a + KK*(z-C*a); a = x(:,i+1);
end
```

Listing 3.8: Implementation of the EKF-algorithm in parallel with the non-linear model.

3.4 Optimization

When optimizing the drying process, designing a routine which solves the optimization problem from scratch is a project by itself. Thus, one of Matlab's built-in function is rather used. The `fmincon` function is a routine which solves non-linear minimization problems. By default, the `fmincon` function uses the interior-point method, but may be configured to use other methods such as SQP. Since the SQP method is applied, the function is configured to do so. Listing 3.9 shows a code-snippet with the `fmincon`-function being reconfigured to use the SQP as well as setting maximum allowed iterations and constraint tolerance. To view the full documentation of the `fmincon`-function, see [46].

```
%% Optimization
disp('Optimizing!')
tic
options = optimoptions('fmincon', 'Algorithm','sqp','MaxIterations',3000,'ConstraintTolerance',1
e-06);
[u,fval] = fmincon(@(u)OFun(u,x,p),u0,[],[],[],[],vl,vu,@(u)nlCon(u,x,p),options);
toc
```

Listing 3.9: Code-snippet of the configuration and implementation of the `fmincon`-function.

To begin with, the `fmincon` function assumes that the problem is continuous, so the discrete model can not be implemented directly. A solution to this problem is rather simple by augmenting the optimization vector with the number of discrete optimization variables. Hence, the `fmincon` function will solve for N number of variables. The choice of optimization variables depends on the objective that is to be optimized. For the objective function, the choice is to either optimize with respect to performance or energy demand. As stated in section 4.1, the model is not suited for control and optimization with respect to performance, which leaves the choice of optimization to energy consumption. Minimizing the heat needed for the process implies minimizing equation (2.14) with respect to the control inputs. Furthermore, having the control inputs continuously changing does not fit with reality. Thus, the drying process is divided into three separated zones, each with a distinct air temperature and air velocity. This reduces the number of optimal control inputs needed to be determined to only six. The constraints for the optimization problem have to ensure that the process reaches a desirable final solvent fraction as well as having upper and lower bound on the air temperature and air velocity to prevent unrealistic control inputs. In addition, as stated in section 2.1, the evaporation rate has a profound impact on the end performance, which implies that constraint on the maximum evaporation rate allowed should also be considered. With the general formulation of an optimization problem presented in equation (2.86), the optimization problem with respect to heat consumption for the drying process can be stated as follows:

$$\arg\min_{\boldsymbol{u}} \ \boldsymbol{f}(\boldsymbol{u},\boldsymbol{x}) = \sum_{i=1}^{N} k_c \cdot (u_i(1) - x(2)_i \cdot T_0) + \dot{m}_i(\boldsymbol{x}, u_i(2)) \cdot h_{fg,i}(x(2)) \tag{3.3a}$$

such that:

$$T_0 \leq u_i(1) \leq 370 \tag{3.3b}$$

$$5 \leq u_i(2) \leq 15 \tag{3.3c}$$

$$C_N \leq 0.1 \tag{3.3d}$$

$$\dot{m}_i \leq a \tag{3.3e}$$

where a is set to an arbitrary constant (as several values will be tested), N refers to the value at the end of the drying-time, $u(1)$ is the air temperature and $u(2)$ is the air velocity. As both C_N and $\dot{m}_i$ is a function of $\boldsymbol{x}$ and $u(2)$, the non-linear model has to be solved in order to determine the final solvent fraction as well as the evaporation rate at each discrete iteration. The objective function is implemented as shown in listing D.4 in appendix D.3. Furthermore, the `fmincon` function requires the non-linear constraints to be declared as a separate function which is called on during the routine. Thus, the non-linear constraints function is implemented in somewhat same fashion as the objective function by solving the non-linear model to compute the $\boldsymbol{c}(\boldsymbol{u},\boldsymbol{x})$ and $\boldsymbol{c_{eq}}(\boldsymbol{u},\boldsymbol{x})$. Listing D.5 in appendix D.3 shows the function which calculates the non-linear constraints. As `fmincon` requires both $\boldsymbol{c}(\boldsymbol{u},\boldsymbol{x})$ and $\boldsymbol{c_{eq}}(\boldsymbol{u},\boldsymbol{x})$ to be defined, $\boldsymbol{c_{eq}}(\boldsymbol{u},\boldsymbol{x})$ is simply set to being empty as there are no non-linear equality constraints. Furthermore, the main script, which defines all variables and parameters, as well as does the post-processing is listed in listing D.3 in appendix D.3.

3.5 Control

The overall system with control features is implemented in Simulink rather than directly in Matlab due to the convenience of working with block diagrams. In section 2.5 it was concluded that the computational time of the optimization problem was too high for the implementation of an MPC to be reasonable. Thus, a decentralized control structure (multiple single loops) is implemented for controlling the air temperature and air velocity to track the calculated optimal control inputs. In an attempt to replicate a realistic control scenario, the heating of the inlet air as well as the acceleration of the air velocity is modelled with disturbances. Prior to describing the implementation of the controller and the overall system in Simulink, the controllers are tuned in accordance to the theory presented in section 2.6. However, the performance of the controllers are presented in section 4.5. Simple PI-controllers are used as these are the type of controllers that are most applied in industrial processes [37].

3.5.1 Air Temperature Controller

For the heat exchanger, the process is modelled using the heat equation as stated in the book [47] by Widder:

$$\frac{dQ}{dt} = c\rho V \frac{dT}{dt} \tag{3.4}$$

where c is the specific heat of the air, ρ is the density and V is the volume. The heat exchanger is modelled such that energy needed for heating the air is provided by a heating element. The disturbances are also assumed to be heat loss to the surroundings which affects the process in the same manner as the heating elements, thus, the heat transferred to the air is expressed as:

$$\frac{dQ}{dt} = \dot{Q} = u + d \tag{3.5}$$

where d denotes the disturbance. Substituting equation (3.5) into equation (3.4) yields:

$$\dot{T} = \frac{1}{c\rho V} \cdot u + \frac{1}{c\rho V} \cdot d \tag{3.6}$$

Here positive magnitudes of d represents heat transferred into of the system. Following the rules of Laplace transform, linearization and appropriate scaling as presented in section 2.6, the system can be expressed as:

$$y(s) = G(s) \cdot u(s) + G_d(s) \cdot d(s) \tag{3.7}$$

where $y(s)$ represents $T(s)$ and $G(s) = G_d(s) = \frac{1}{c\rho V s}$.

Considering the linearized transfer function for the heating process in equation (3.7), the power needed to increase the temperature of the air by one Kelvin can be determined. The expression is linearized about $T = 350\ K$ which is the average of the upper and lower bounds set for the air temperature in the optimization problem. Having $c = 0.7214 \frac{kJ}{kg}$, $\rho = 1.009 \frac{kg}{m^3}$ and $V = 1\ m^3$, the power needed to change the temperature is approximately $8.84 \frac{kW}{T}$ [48, 49]. Thus, the maximum allowed magnitude of the control input and disturbance is in the same scale. Assuming that the heating element has the ability to change the temperature by half a Kelvin at any point, the maximum control input is set to $5\ kW$, whereas the maximum disturbance the system may experience is set to $1\ kW$. The output T is expected to remain within the boundaries set by the optimization problem, which implies no larger deviation than $20\ K$ from the linearization point. With this in mind the scaling parameters are as follows:

$$W_y = 20, \quad W_u = 5, \quad W_d = 1 \tag{3.8}$$

Heating a medium does not happen instantaneously which can be translated into a delay in the output. Assuming that the heating process is relative fast, the output delay is set to 1 second. Furthermore, the process is assumed to have no input delay. This is not realistic as there will be delays related to the heating element. However, this assumption is made to avoid to large restriction on the bandwidth frequency. With the assumption and scaling mention above the transfer functions for the temperature process is:

$$G = \frac{0.25}{0.7279s} \cdot e^{-s}, \quad G_d = \frac{0.0125}{0.7279s} \cdot e^{-s} \tag{3.9}$$

The controller was tuned in accordance to the procedure described in section 2.6.3.

3.5.2 Air Velocity Controller

For the acceleration of the wind, the system is modelled using a fan to increase the velocity of the air. To calculate the energy needed for accelerating the wind, the Bernoulli equation has stated in the book by Cengel and Cimbala is used [50]:

$$\frac{dQ}{dt} = \frac{V_{out}^2 - V_{in}^2}{2} \cdot \dot{m} \tag{3.10}$$

Rearranging with respect to V_{out} and setting $\frac{dQ}{dt} = u$ yields:

$$V_{out} = \sqrt{\frac{2u}{\dot{m}} + V_{in}^2} \tag{3.11}$$

The disturbance is assumed here to affect the air velocity directly, e.g. measurement noise and so on. Applying the same rules of Laplace transform, linearization and scaling as for the heat exchanger, the air acceleration process can be expressed by:

$$V(s) = G(s) \cdot u(s) + G_d(s) \cdot d(s) \tag{3.12}$$

where $G_d(s) = 1$ as the disturbance enters the process directly and $G(s)$ is:

$$G(s) = \frac{1}{\dot{m}} \cdot \frac{1}{\sqrt{\frac{2u^*}{\dot{m}} + V_{in}^2}}$$

where u^* is the energy needed to accelerate the air to the velocity at which the model is linearized about.

For scaling the air velocity process, the same procedure as for the temperature process is followed. The process is linearized about $15\ \frac{m}{s}$ with a inlet velocity of $1\ \frac{m}{s}$ and mass rate of $1\ \frac{kg}{s}$. This results in the power needed to change the air velocity to be $7.5\ \frac{kW}{\frac{m}{s}}$. Assuming that the fan has the ability to change the air velocity by slightly more than one meter per second at any point, the maximum control input is set to $10\ kW$. For the air velocity system, the disturbances are assumed to affect the process directly and assuming that affects the process of much lower magnitude compared to u, the maximum disturbance is set to $0.5\ kW$. Furthermore, the output is expected to not deviate more than $5\ \frac{m}{s}$ from the linearization point. The scaling parameters then becomes:

$$W_y = 5, \quad W_u = 10, \quad W_d = 0.5 \tag{3.13}$$

Assuming that fan has the ability to change the air velocity instantaneously once the current control action is implemented the process has no time delays. However, in contrast to the air temperature process, it is here assumed that the control input is delay by 1 second which translates into an input delay (the fan takes 1 second to produce the correct input). This yields the following transfer functions:

$$G = 0.133 \cdot e^{-s}, \quad G_d = 0.1 \tag{3.14}$$

The controller was tuned in accordance to the procedure described in section 2.6.3.

3.5.3 Implementation

In Matlab, the heating and air velocity process was converted into a single state-space model with four inputs; two for the control inputs and two for the disturbance inputs. Figure 3.2 shows how the decentralized controller with two PI-controllers (one for each control input) was implemented in Simulink. In Simulink, the references signals are converted and scaled appropriated prior to entering the control loop to fit the model which is expressed in deviation variables. The control inputs are later converted back into their original values before used in solving the drying process. Anti-windup schemes as explained in section 2.6.4 was implemented to prevent windup effects caused by any saturation in the control inputs, while random disturbances with a given frequency is generated using a signal generator.

The decentralized control is a subsystem of the overall drying process. The implementation of the overall system in Simulink is illustrated in the simplified block diagram shown infigure 3.1 which consists of the decentralized controller K, the control plant (air temperature and velocity process), the plant, sensors and the extended Kalman Filter. The actual implementation can be found in figure F.1 in appendix F. Due to the lack of actual measurements from a physical plant, the proposed model is used as the "unknown"-plant model to provide measurements for the state estimation. The senors subsystem extracts the measurement variables from the "unknown" state variable X and adds the appropriate noise to the measurement. The Kalman filter block is the same procedure as presented in listing 3.8. Both the plant and Kalman filter block is realized using the Matlab Function Block for Simulink, with the respective functions being the same a previously presented. The only change made to the functions is that every variable that is used and calculated during the routine is determined within the respective function as the Matlab Function Block does not support nested functions.

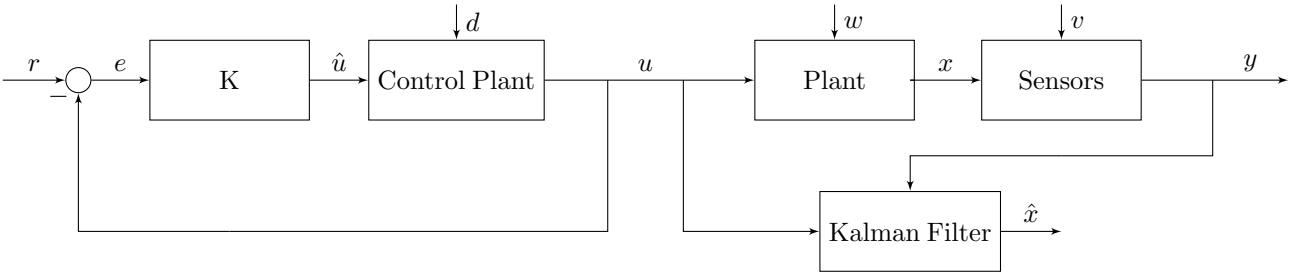


Figure 3.1: Simplified block diagram of the overall system.

In practice, each drying stage is controlled by distinct controllers, which implies that each controller is unaffected by the process in other drying stages. Implementing the process as in figure F.1 implies that the distinct stages is controlled by the same controller. This will cause effects when the reference in control inputs changes as the thin-film enters a new drying stage which wont occur in practice. To avoid this, and at the same time avoid implementing more controllers with delayed start time which has to be selected based on a time criteria, the system is solved sequentially for each drying stage. This can be done as each drying stage is in essence the same process. Therefore, the implementation in figure F.1 is used for each stage but initialized with the different parameter values and starting conditions. The script for initializing, simulation and processing the solution is listed in listing D.6 and can be found in appendix D.4.

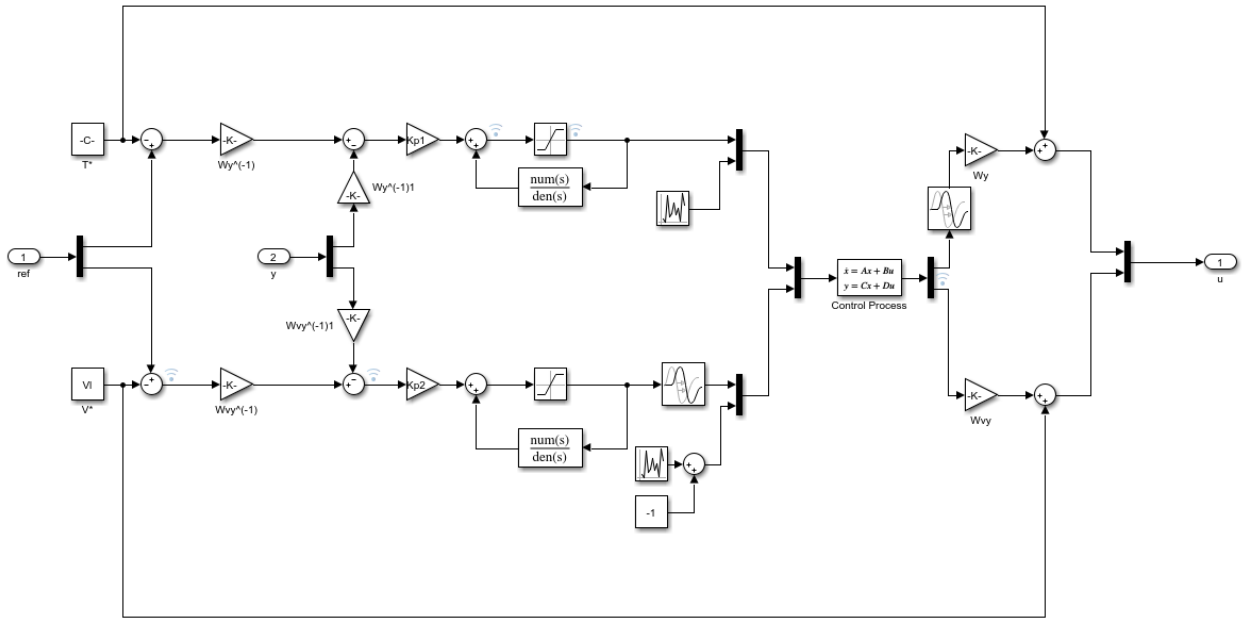


Figure 3.2: Implementation of the decentralized controller with saturation and anti-windup scheme.

Chapter 4

Results and Discussion

In this chapter the results from various simulations and experiments are presented and discussed. The chapter is structured in the most reasonable way with validation of the model and its numerical solution being first. The behavior of the model will be compared with results from different studies to ensure the validity of the model and pin point any limitations or errors. Furthermore, the results of the linear analysis is carried out to assess whether or not a linearized model can be used rather than the non-linear model for control purposes which will greatly reduce the complexity of the optimization and control problem. Lastly, the sections with the optimization and control results are presented and evaluated to assess the performance and applicability of such a control structure in the battery industry.

4.1 Model Validation and Numerical Method

To achieve acceptable performance of any given control structure, the accuracy of the model that describes the dynamics has to be sufficient. Therefore, it is important that the model derived in section 2.1 resembles the general behavior of an electrode drying process as much as possible in terms of transient behavior and general characteristics. Equally important is the accuracy of the numerical method applied for solving the model. Thus, several simulations are carried out to provide information of the proposed model which is used to assess its validity. The simulations are carried out with, to some extent, a random battery composition as such information turned out to be difficult to obtain. However, the parameters are either obtained or chosen based on the work done by Sursala et al. and Mesbah et al. in [24, 23] such that the composition resembles the one of an actual battery. Table 4.1 lists the parameters used in the simulations.

Table 4.1: Chosen parameters for simulations.

Parameter	**Value**	**Unit**
ρ_s	838	$\frac{\mathrm{kg}}{\mathrm{m}^3}$
ρ_{solids}	1200	$\frac{\mathrm{kg}}{\mathrm{m}^3}$
C_p	1900	$\frac{\mathrm{J}}{\mathrm{kg \cdot K}}$
R	8314	$\frac{\mathrm{J}}{\mathrm{kmol \cdot K}}$
M_s	99.133	$\frac{\mathrm{kg}}{\mathrm{kmol}}$
D_0	$9{\cdot}10^{-9}$	$\frac{\mathrm{m}^2}{\mathrm{s}}$
L_0	$0.165{\cdot}10^{-3}$	m
T_0	330.5	K
C_0	0.5	$\frac{\mathrm{kg_s}}{\mathrm{kg_f}}$
ϕ	0.25	-
$\mathcal{X}$	0.45	-
γ	1.15	-
E	$7.70{\cdot}10^{6}$	$\frac{\mathrm{J}}{\mathrm{kmol}}$
V	15	$\frac{\mathrm{m}}{\mathrm{s}}$
T_a	350	K

4.1.1 Model Validation

Figure 4.1 shows the development of the film thickness, film temperature, solvent fraction and film density using the parameters listed in table 4.1. The plots clearly shows the behavior of the system and how it reaches stationary conditions after approximately 150 minutes. The thickness decreases linearly for the majority of the deformation of the thin film until the deformation slows down and decreases exponentially. This is in close relation with the reduction of the solvent fraction as it seems to follow the same pattern. A constant drying rate followed by a falling drying rate is a common behavior of drying processes which is well documented in published literature, such as the work published by Ludovico-Marques et al. and King and King in [51, 52]. The evaporation seems to continue even though the concentration level at the top layer has reached a stationary value. This is because of the transportation of the solvent from the lower layers being limited by a further decreasing diffusion coefficient. This is common trait for diffusion limited drying process and is well explained and studied by van Ballegooijen et al. in [53]. Their results also shows the same transient behavior as in figure 4.1. A plot of the decreasing diffusion coefficient can be found in figure E.1 in appendix E. Furthermore, the temperature reaches the air temperature within approximately 45 seconds and is very fast relative to the solvent fraction and thickness development. An explanation for the huge differences in settling time is the heat transfer coefficient, k_c, being more than 10^4 greater in magnitude than the mass transfer coefficient, k_m. This result further justifies the simplification of neglecting temperature gradients within the thin film as such gradients would have insignificant impact on the drying behavior of the thin film. The density seems simply to be a reflection of the thickness as it increases with decreasing film thickness. This makes sense as the density is assumed to be a linear function of the solvent fraction, thus, making the film thickness and density both related to the solvent fraction.

Comparing these results with the work presented by Sursala et al. and Mesbah et al. in [24, 23] indicates that the model successfully resembles the general behavior and characteristics of that of an electrode drying process. However, the model does not account for the forming of a porous media during the deformation of the thin film which is a major limitation of the model. As a result, the final thickness of an actual film would be greater due to formation of void spaces within the film, leaving the model unfit for control purposes in cases in which the final thickness of the film has to accurate. Thus, prior to be used for performance related control, the model should be extended to account for the forming of a porous media. Nevertheless, the model can still be applicable for energy minimization problems in which high accuracy of the film thickness is not needed.

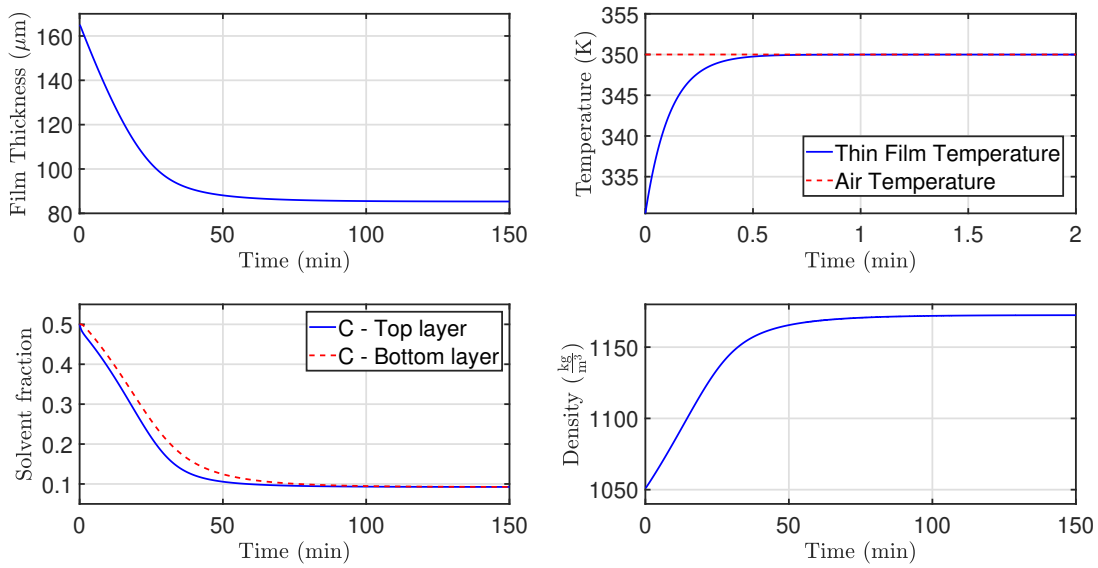


Figure 4.1: Simulation of the model with the parameters listed in table 4.1.

4.1.2 Mesh Independency and Accuracy

The accuracy of the approximated solution is important when the system is evaluated and used for control purposes. Naturally, as the numerical method finds an approximation of the exact solution, error due to round-off, discretization and truncating infinite sums will always be present. However, it is of interest to keep the error at a minimum without compromising performance as higher accuracy invokes higher computational power. Insight in the numerical method's ability to approximate accurate solution is key for determining the models limitation. Thus, various simulation are carried out to assess the accuracy of the model in terms of mesh independency, time-step dependency and mass conservation.

Mesh Independency

Mesh independency is simply a term used for the number of elements that is required for the solution to no longer change when the spatial increment is decreased. Figure 4.2 shows results of the simulation with an increasing number of elements. The approximated solution seems close to independent of the number of elements chosen for the spatial domain. However, the thickness do seems to have some small variations as it approaches the final drying time. A closer look at the reveals that there is indeed some variations in the final values of the parameters with respect to the number of elements. Figure 4.3 plots the normalized final values for each parameter with an increasing number of elements for three different time increments. For the temperature, the variations are limited to the 10^{-9} which are insignificant and the temperature is said to be mesh independent. The final value of the evaporation rate experiences variations of approximately 0.6 $\frac{\mu g}{s}$ which are relative low when the scale of the evaporation rate (as in figure 4.2) is taken into consideration. In addition, the solvent fraction also experience a relative small variation, of only 10^{-6}. Considering the relative low scale of the variations caused by spatial and time discretization, these three variables is considered to be mesh independent. However, the thickness experiences a variation of $5 \cdot 10^{-3}$ due to the mesh resolution which is thought to cause significant errors. The variation is reduced to $1 \cdot 10^{-4}$ from 100 to 200 elements. Therefore, the thickness is considered to become mesh independent in the spatial domain at around 200 elements. As a result, the complete system becomes mesh independent at approximately 200 elements.

Considering the argumentation in the previous section, the most reasonable choice in terms of accuracy is to use 200 elements or more. However, as figure 4.4 shows, the computational time for solving the system increases exponentially with the number of elements. This implies that a combination of low steps in both time and space implies higher computational time for solving the system. If the model is to be used for a MPC, a optimization problem based upon the model has to be solved for each time step. A large computational time will then enforce a large sample time for the control structure. Large sample times will miss important dynamics in the system and the worst-case scenario occurs when the sample time is brought lower than the computational time for the numerical method. This will cause an internal delay and render the overall system unstable. As a consequence, there is a trade off in terms of accuracy and computational time when choosing the steps in space and time. Based upon the observations done, a combination of either a time increment of one or two seconds with 20 elements seems like a overall good compromise between accuracy and computational demand.

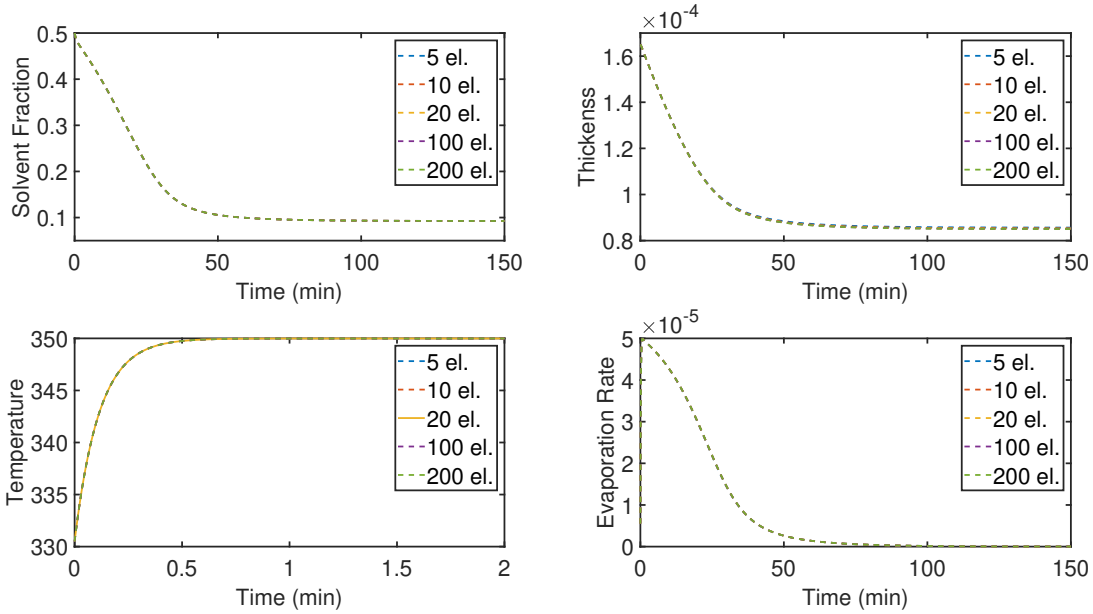


Figure 4.2: Simulation with different numbers of elements for the discretization of the spatial domain.

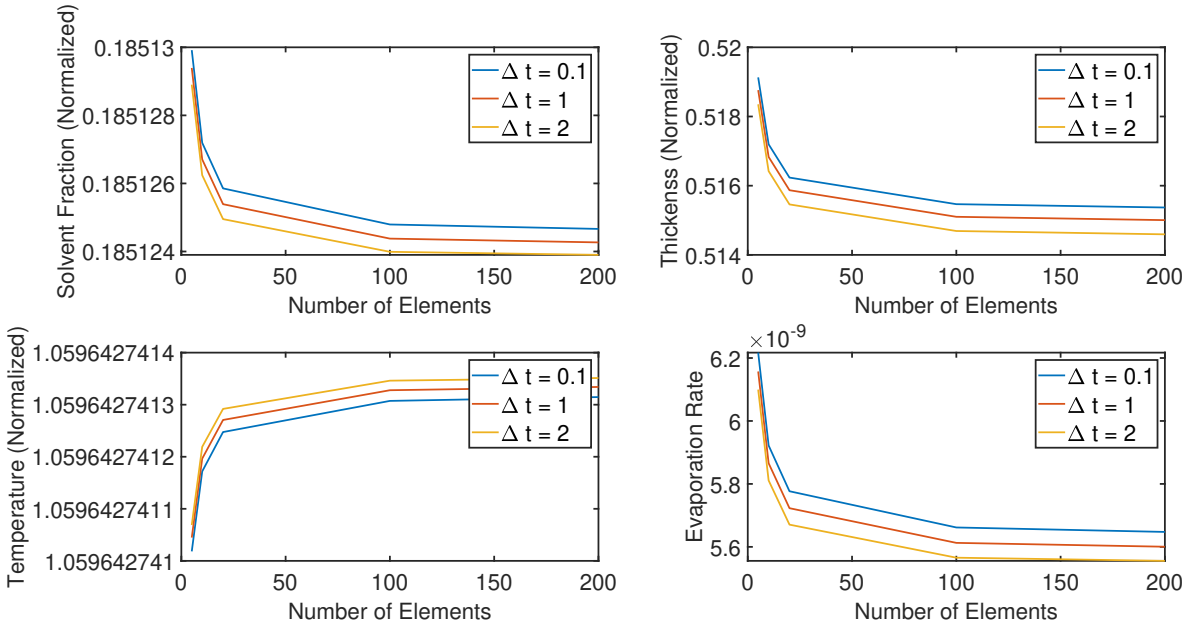


Figure 4.3: Variation of the final values of each parameter with respect to the number of elements.

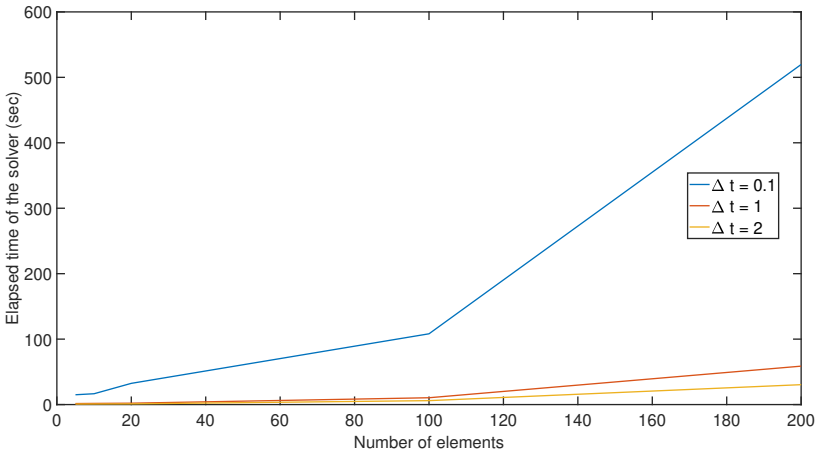


Figure 4.4: Computational time as a function of number of elements.

Comparisons of Quadratic and Linear Shape Functions

In the previous sections quadratic shape functions was used to approximate the unknown solution. It also is of interest to see how well linear shape functions approximates the unknown solution compared to quadratics. Figure 4.5 shows a plot of the normalized error between the solution using quadratic and linear shape functions (keep in mind that it is assumed that the solution with quadratic shape function is more accurate). Initially the error is of about 0.68 % and decreases with increasing number of elements. The figure also illustrates the time ratio of the elapsed time for the solver between quadratic and linear shape functions. For the ratio of the elapsed time for the solver, the ratio increases exponentially with increasing elements. This makes sense as quadratic elements requires three nodes for each element whereas linear only requires two. Thus, the total number of nodes needed to be computed is $2 \cdot n_e + 1$ and $n_e + 1$ for quadratic and linear elements, respectively. This results indicates that elapsed time of the numerical solver can be greatly reduce by allowing a small increase error. From a control perspective, using linear shape function instead of quadratic may increase the controllablity and observability of the system as using linear shape functions implies less states compared to quadratic shape functions.

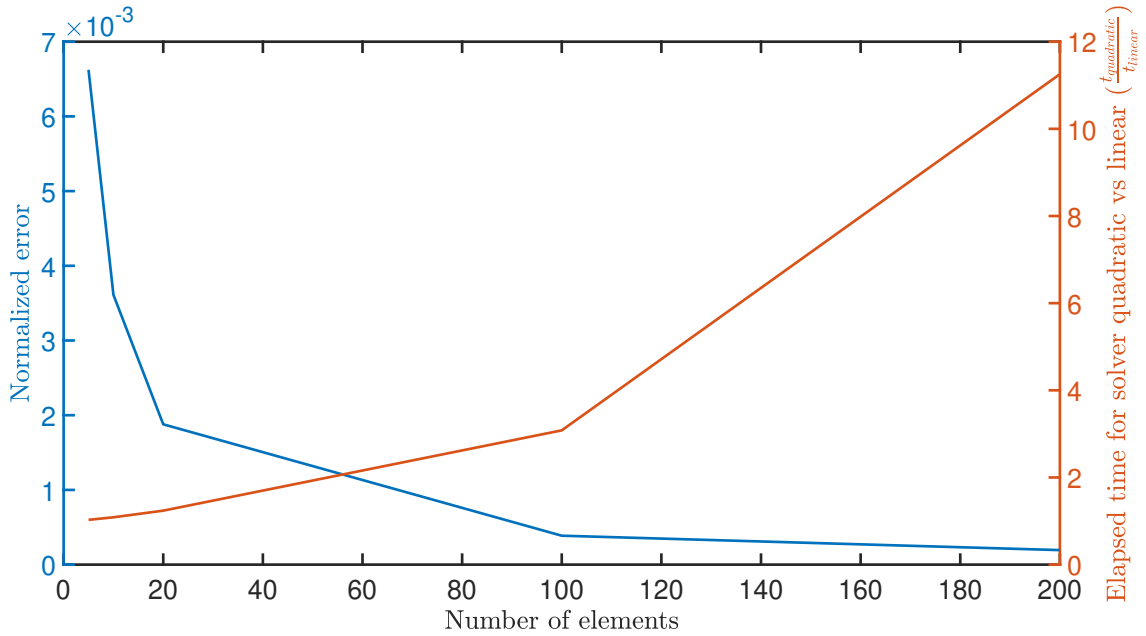


Figure 4.5: Comparison of linear and quadratic shape functions.

Mass Conservation

It was previously stated that the model is not suited in control structures for performance purposes due to the lack of porosity. However, the model can still be applicable for problems related to energy minimization. If so, the model has to conserve energy and mass. In essence, conservation of mass implies conservation of energy and vice versa, which means that by looking at the mass conservation based upon the amount of mass evaporated compared to the difference in initial and final film volume, it should equal. For simulations with varying number of elements and a time increment, the error in mass conservation is $0.16\% \pm 0.04\%$. Some errors are to be expected as the solution is approximated and errors due occurs for the same reasons previously mentioned.

4.1.3 Sensitivity Analysis

Information about how different parameters affects the drying process in terms of drying time, energy consumption and final solvent content is not only interesting from a control perspective, but also for general knowledge of drying processes. To gain this information, several simulation are carried out to assess each of the individual parameters influence on the drying process. For each of the tests, the selected parameter is increased or decreased with a certain percentage while all other parameters are kept constant. The base values for every parameters are the ones found table 4.1, however, these base values can be arbitrarily chosen.

Energy

Figure 4.6 shows a sensitivity plot with respect to the energy consumption of the drying process. The initial film temperature has the largest effect on the energy consumption of all the parameters. The main reason is that the energy needed for evaporating the solvent is in the same order of magnitude as the energy needed for the heating of the film. Thus, regardless of the rapid temperature development, reducing the energy needed for heating the film affects the overall energy requirement significantly. However, this does not take into account that energy saved or added when changing the initial film temperature zeroes out when considering that the energy is either added or saved prior to the drying process. Hence, the initial film temperature will have no impact on the overall energy requirement of the system unless there are more/less efficient heating methods prior to the drying process. Furthermore, consumed energy decreases with increasing initial film temperature. This can be deducted straight from equation (2.12) as larger temperature differences implies larger heat transportation. Secondly, the air temperature affects the drying process in the same fashion as the initial film temperature with decreasing energy requirement with increasing air temperature. The same arguments regarding preheating can be used for the air temperature. However, the air temperature also affects the vapor pressure and the maximum temperature the film can achieve. As the latent heat of evaporation is a decreasing function of the film temperature, the energy needed for evaporation reduces with increasing film temperature (a plot of the latent heat of evaporation as a function of T can be found in figure E.2 in appendix E). Considering this, the air temperature does shows good potential for reducing the energy requirement for the process.

The initial solvent fraction and thickness also has a large impact on the energy consumed for the process, however, in the reverse direction compared to the initial film and air temperature. Intuitively, this makes sense as increasing either of them implies a large amount of solvent has to evaporate during the drying process. Hence, from an energy perspective, a slurry composition with as little solvent as possible will yield the lowest energy consumption for the drying process. Furthermore, the air velocity seems to have little impact on the consumed heat. Considering that both the heat and mass transfer coefficient is a function of the velocity one might except a larger influence. However, the changes in both coefficient is too small for the air velocity to have a large impact on the energy consumption. The initial diffusion coefficient seems to affect the heat consumed with the same magnitude but in the reverse direction as the air velocity. Intuitively, the diffusion coefficient should not have any impact on the consumed heat as it mainly limits movement of the solvent within the film. Lastly, the relative humidity of the air has the least influence on the consumed heat during the process.

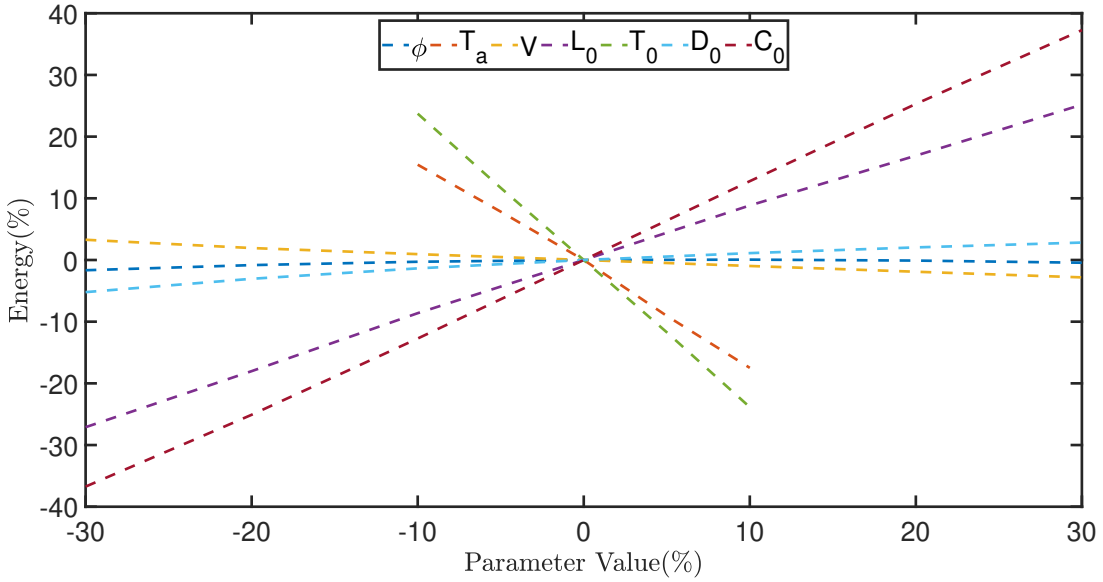


Figure 4.6: Sensitivity plot for energy consumption.

Solvent Fraction

Figure 4.7 shows the sensitivity plot with respect to the final solvent fraction. In addition to T_a having a small influence, the relative humidity affects the final solvent fraction the most. This can be explained by the vapor pressure being a function of both the relative humidity and air temperature. However, the vapor pressure is a much weaker function with respect to the air temperature compared to the relative humidity. Increasing vapor pressure increases the equilibrium pressure which again allows for more solvent to evaporate. Further, neither of the initial parameters values does affect the final moisture content. Considering the vapor pressure being the only limiting factor of the final solvent fraction, it has already been stated that the vapor pressure is function of air temperature and relative humidity, only. Thus, neither of the initial parameters value will have any influence on the final solvent fraction. The same argumentation can be used for the air velocity.

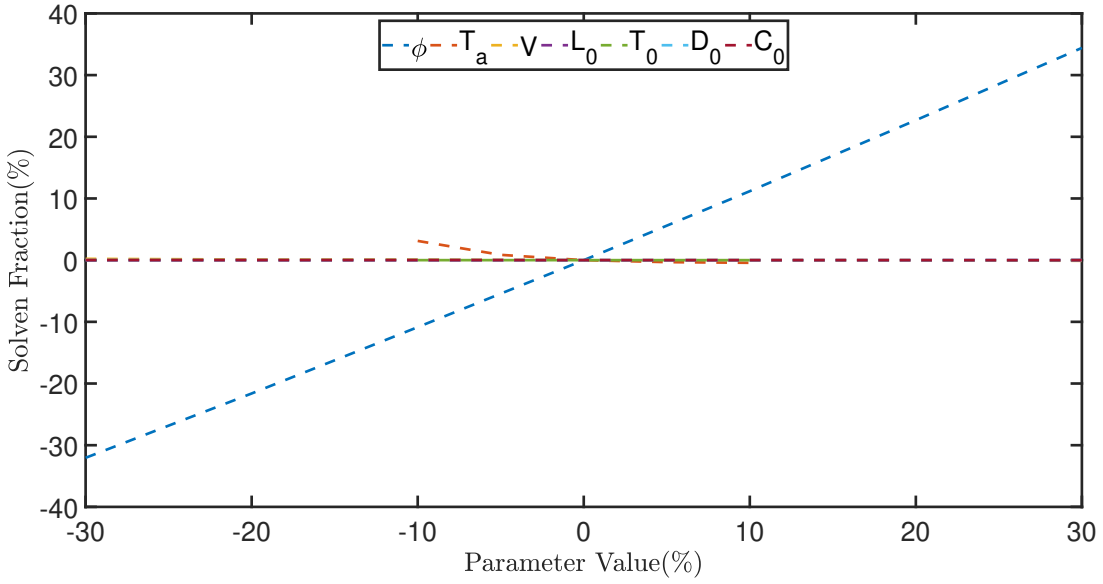
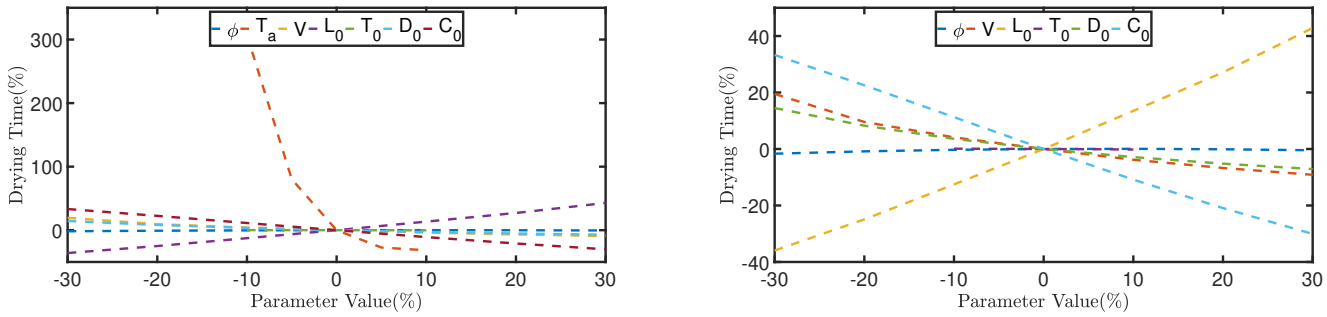


Figure 4.7: Sensitivity plot for solvent fraction.

Drying Time

Figure 4.8 shows the sensitivity plots for the drying time. The drying time is set equal to the time when the evaporation and the change in evaporation falls below a given threshold. From figure 4.8a shows the sensitivity plot with the air temperature included. Clearly, the air temperature has an huge influence on the drying time compared to the parameters and to better see how the drying time is affected by the other parameters, figure 4.8b shows the same plot but with the air temperature excluded. The reason for the air temperatures large influence can be explained with the pressure difference being the driving force of the evaporation. As the vapor pressure decreases with air temperature it limits the amount of solvent that can evaporate and vice versa. However, figure 4.8a also shows that the drying time decreases exponentially with decreasing air temperature rather than linearly. The main reason for the exponentially reduction in drying time is diffusion limiting behavior. The drying time is prolonged as a consequence of the restricted movement of the solvent within the film. As the evaporation is assumed to be limited to the surface only, the solvent fraction at the surface layer quickly reaches the final solvent content, and from this point the evaporation is limited to the rate at which the solvent reaches the surface layer. Thus, increasing the diffusion coefficient along with air temperature would yield even lower drying times.

For the initial film temperature there are no significant changes in the drying time. This is due to the rapid temperature development compared to solvent removal. Likewise, the relative humidity as minor influence on the drying time. However, by changing the relative humidity it was observed from figure 4.7 that the final solvent content also changed. Thus, in practice, changing the relative humidity implies that more or less solvent evaporates within the same time frame, however, the threshold at which the drying time is determined is reached approximately at the same time. Furthermore, the air velocity and diffusion coefficient seems to have the same influence on the drying time. Changing the air velocity affects the mass transfer coefficient which allows for more/less mass to evaporate at any given time. As previously mentioned, the diffusion coefficient limits the movement of the solvent within the film, thus, restricting the amount of solvent that is at the surface layer of the film. Naturally, increasing initial thickness yields longer drying time as it implies more solvent has to evaporate. Likewise, one might assume that the initial solvent fraction would influence the drying time in the same manner. However, this is not the chase. It would seem that increasing initial solvent fraction yields lower drying times. The main reason for this is the way the partial pressure has been modelled using the Flory Huggin's theory in equation (2.6). Figure E.3 in appendix E shows how the evaporation rate increases with the increasing initial solvent fraction and that it also that the evaporation is brought to a stop faster for increasing initial solvent fractions. This implies that there is a trade-off between trying time and heat consumed when considering the initial solvent fraction.



(a) Sensitivity plot for drying time.

(b) Sensitivity plot for drying time without T_a.

Figure 4.8: Sensitivity plots for the drying time.

4.1.4 Control Inputs

From a control perspective, it is reasonable to choose the parameters that affects the process the most as control inputs to achieve effective control. In the previous sections the influence of each parameter regarding heat consumed, solvent fraction and drying time was presented which gave good indication of which parameters that is suited as a manipulated parameter. However, there are limitations. Although some of the initial parameters values had a large influence over the drying process, they are not suited as control inputs. This is because that these parameters describes the mixture prior to drying process and can not be manipulated throughout the process. As a consequence, these parameters are much more interesting consider optimization of the composition with respect to either heat consumed, solvent fraction or drying time. For the process, there are three parameters that can be manipulated during the process; air temperature, air velocity and the relative humidity. The latter does affect the final solvent fraction which implies that it can be used to control the final thickness of the film. However, for the end product, the least amount of moisture is always desired. Thus, choosing the relative humidity will always result in control input that ensures the lowest relative humidity as possible. One could argue that by choosing the relative humidity of the air, one could control the evaporation rate as the vapor pressure is function of the relative humidity. However, it is believed that control of the relative humidity is much slower than the control of the air temperature. Hence, the same affects on the evaporation rate can be achieved faster by using the air temperature as a control input instead. In addition, as previously mentioned, the model fails to represent the porosity of the film which renders it unsuitable for performance control and optimization in which thickness is of major concern. As a consequence, choosing the relative humidity as a control input for optimization and control with respect to energy and drying time minimization would been ineffective as it as little to no impact on either as shown in the previous sections. Therefore, the most reasonable control inputs is the air temperature supplemented by the air velocity due to their influence on both heat requirement and drying times.

4.1.5 Summary

From the results of the simulation is was concluded that the derived model resembles the general transient behavior and characteristics of that of an electrode drying process. However, the model does not account for the forming of a porous media which renders the model unsuitable for performance control and optimization in which accurate solution of thickness is important. Nevertheless, the model and its numerical solution as a low error in mass conservation which makes the model suitable for energy optimization and control. To further increase the applicability of the model, it was recommend that the model was extended to account for porosity.

The numerical solution of the model showed promising results with low error in mass conservation. However, the solution requires a large amount of elements to become mesh independent, which implies high computational time. This is unwanted as high computational time may invoke instability when the system is to be controlled using the model. To avoid high computational time, it is suggested that the elapsed time of the solver was decreased in favor of increased error in the solution by reducing the number of the elements for the finite element approximation. Furthermore, the elapsed time of the solver could be further decreased by using linear shape functions instead of quadratic shape functions. Assuming that quadratic shape function is more accurate, using linear shape functions instead of quadratics further increases the error of the numerical solution. Thus, there is an important trade-off between accuracy and computational time that has to be taken into consideration when the model is to be used for control purposes.

From the sensitivity analysis, the behavior of the model was further validated. The model seemed to react reasonable to changed in the different parameters when considering the energy consumption, final solvent fraction and drying time. Furthermore, the results gave good indication of which parameters that is suitable as control inputs. As many of the parameters are initial values, only the air velocity, air temperature and relative humidity turned out the be promising candidates as control inputs. The latter was later dismissed as the lowest final solvent fraction is always desired, which would yield the lowest possible control input for the relative humidity.

4.2 Linear Analysis

Linear system are less complicated to deal with for system analysis, optimization and control. For these reasons, a linearized system is derived and tested to determine whether or not a linear approximation of the non-linear model can used for control purposes. The linear model is compared with the non-linear in terms of transient behavior and characteristics, as well as the controllability and observability is checked based upon the derived state-space realization. Lastly, a comparisons between the transient behavior of the linear and non-linear model when the manipulated variables are excited is also presented.

4.2.1 Comparison of the Linear Model With the Non-Linear Model

For this simulation, both the linear and non-linear models are simply simulated using the values found in table 4.1 and the linear system is linearized around the stationary values of the non-linear model. Figure 4.9 shows the simulation with both the linear and non-linear system plotted. Clearly, the linear model fails to resemble the non-linear model sufficiently as the solvent concentration and evaporation rate decreases rapidly. In addition there is a huge stationary deviation in the final thickness of the thin film. A reason for this is the rapid decrease in the solvent fraction which implies a rapid decrease in the partial pressure of the solvent. Thus, the difference between the partial pressure of the solvent and the vapor pressure of the air converges rapidly to zero. As the difference in pressure is the driving force of the evaporation, the amount of mass that evaporates is being restricted which naturally yields a higher final thickness due to less solvent being evaporated. Although the temperature development is much more accurate relative to the other variables, it too is also faster for the linear model compared to the non-linear model. These results all point in the direction of the initial perturbations being too large, and that these initial deviations are outside the validity range of the model. This can be compared to the example in figure 2.7 where x-values of 0 or 2 yields relative large errors between the linear and non-linear function. By looking closer at the plots for the drying rate, the linear model does seem to follow the non-linear model from approximately 45 min. A further investigation of the validity range of the linear model is presented in upcoming section. However, with the information provided by the particular simulation, the conclusion is that the initial perturbations are too large for the linear system to be used for the entire drying process.

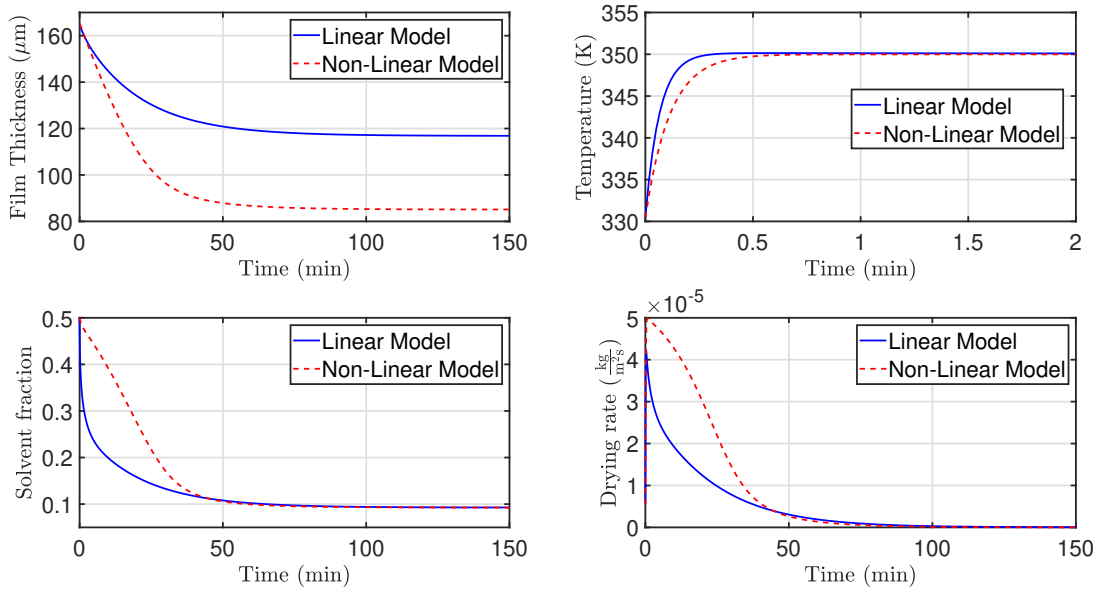


Figure 4.9: Simulation of the linear and non-linear model using the parameters found in table 4.1.

The stability of the system is assessed by looking at the eigenvalues of the system matrix. Despite the number of elements chosen for the finite element approximation, the linearized system always becomes marginally stable as the both the continuous and discrete system have either a eigenvalue of 0 or 1, respectively. Figure 4.10 plots the eigenvalues of the continuous and discrete linear system with 1, 10 and 20 elements. It is clear that for every simulation both system is marginally stable. If we are to use the classification of Hovd in [37], these system would been defined as unstable. However, form the simulations it can be observed that the solutions does not grow unbound, thus, a conclusion is drawn for the system to be marginally stable regardless of the discretization of the spatial domain.

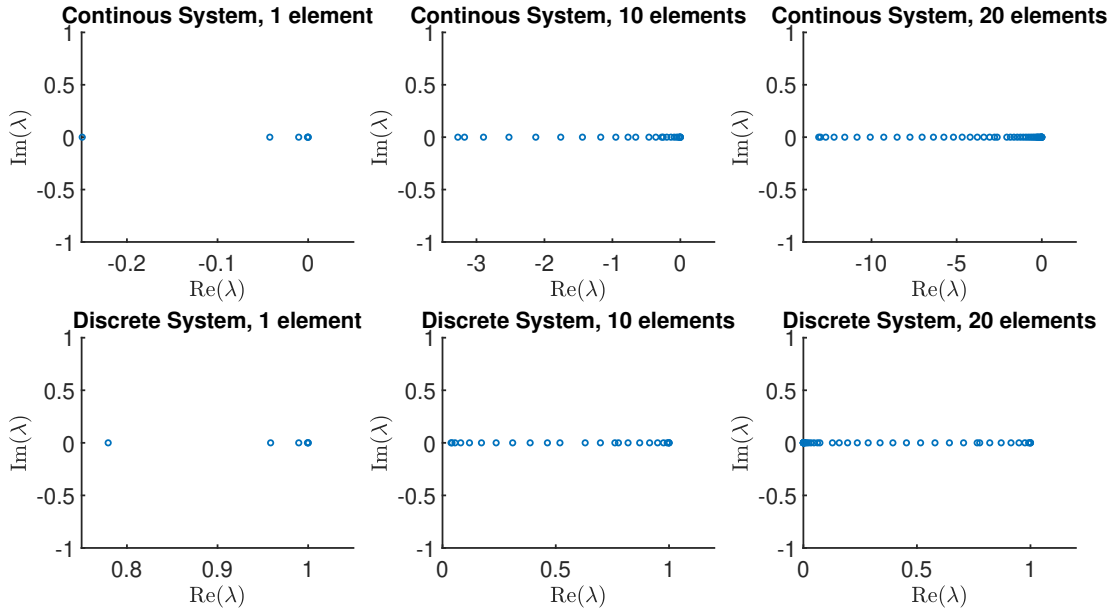


Figure 4.10: The eigenvalues with increasing number of elements

4.2.2 Initial Perturbations

It is interesting to see whether or not the linearized model has the ability to converge to the linearization point when initialized with a given perturbation. The simulation are carried out with increasing initial perturbations of the variables, and for each simulation the initial perturbation are identical for every variable. Figure 4.11 shows the results from these simulations. The results indicates the same issue as in the previous section with a stationary deviation in the thickness. These deviation still occurs with relative small initial perturbation, even though it is on a small scale. The reason is the same as for the previous scenario with the reduction of the solvent fraction being so fast that evaporation stops to occur before the thin film has reached its final thickness. Thus, an identical initial offset of the variables (the thickness and solvent fraction) will always yield a stationary deviation in the thickness. A stationary deviation can also been seen for the density, which makes sense as it is a function of both the solvent fraction and thickness. Due to these errors, the linearized approximation of the non-linear model does not seem to be a promising alternative for control and optimization purposes. Figure 4.12 illustrates how the stationary deviation in the thickness changes with varying initial perturbations in the solvent fraction. In this case, the initial perturbations for the rest of the variables are set to 0.05. Although different values for the initial perturbations for the solvent fraction and the thickness does not make much sense for a physical perspective, the plot shows how the models fails to arrive at the correct final thickness.

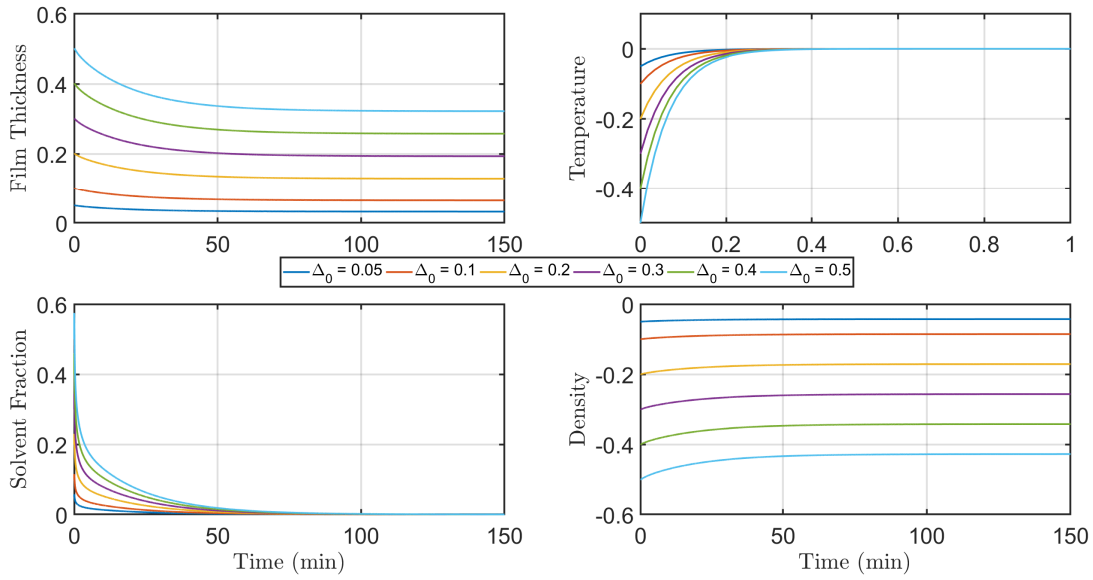


Figure 4.11: Simulations with different initial perturbations for the linear system.

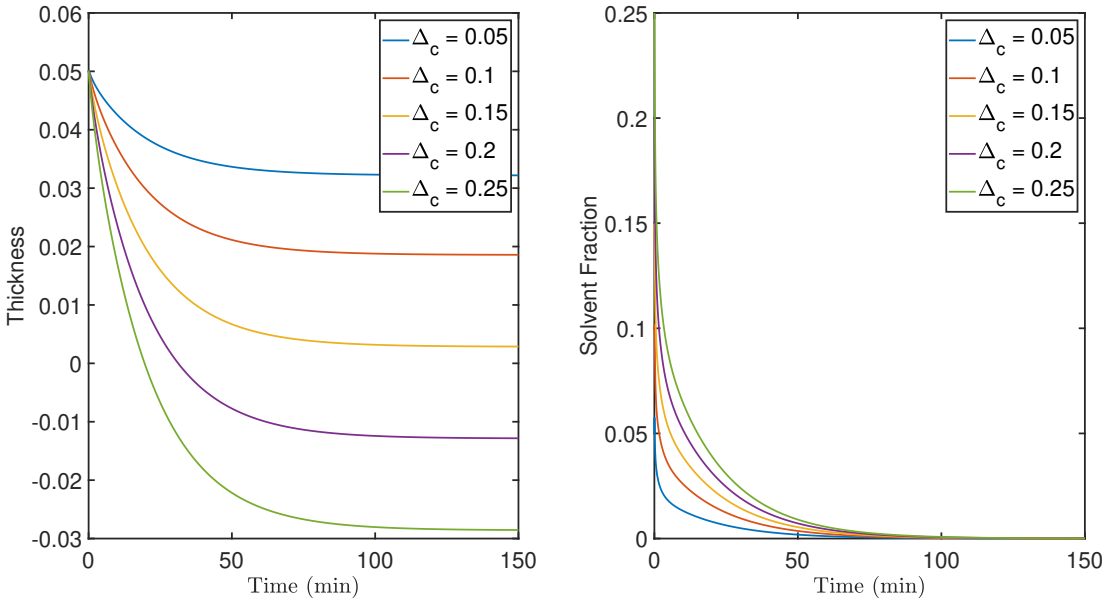


Figure 4.12: Stationary deviation in thickness with varying initial perturbations in solvent fraction.

4.2.3 Controllability and Observability

Table 4.2 list the number of uncontrollable and unobservable states with T_a and V being chosen as control inputs for two scenarios: (i) Only measurement for the thickness and temperature is available. (ii) Measurement for every state is available. The latter is an unrealistic scenario, as measurements of the solvent fraction in the layers of the film is not possible. However, it is of interest to see whether or not such a setup can improve the properties of the system.

For the parameters chosen in table 4.1, the system is observable for a small number of elements, but fails to be controllable despite the number of elements. The number of uncontrollable and unobservable states increases with the number of elements chosen for the spatial discretization and a large increase in uncontrollable and unobservable states occurs from 10 to 20 elements. The increase in the number of uncontrollable and observable states are expected as this introduces more solvent fraction states. A reason for the large increase in both uncontrollable and unobservable state might be that the control inputs T_a and V only affects the solvent fraction at the top layer through the boundary condition. As the the number of elements increases, the solvent fraction at the boundary node interacts with a lower percentage of the total number of solvent fraction node. Thus, limiting the influence of T_a and V on the system. This reasoning fits with the explanation of controllability by Micu and Zuazua in [54]. In the seconds scenario with every state being measured, naturally the system is observable regardless of the number of elements. Considering the definition of the controllability matrix in equation (2.75), changing the number or setup of the measurements does not change the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Hence, the controllability is unaffected by the number of measurements.

Table 4.2: Number of uncontrollable and unobservable state with T_a and V as control inputs.

Number of Elements	**2 Measurements**		**Every State Measured**	
	Nr. Uncontrollable States	**Nr. Unobservable States**	**Nr. Uncontrollable States**	**Nr. Unobservable States**
1	1	0	1	0
5	2	0	2	0
10	2	2	2	0
20	17	18	17	0
30	37	38	37	0
40	56	58	56	0

One might assume that changing the selected control inputs might improve the controllablility and observability of the system. However, as discussed in previous sections, T_a and V is the most appropriate variables to be used as control inputs due to their influence on the system and that they may be changed during the drying process. As both T_a and V interacts with the same states, the controllability should not improve by only using one of them instead of both. In fact, removing either T_a or V as a control input and instead treat it as a state should introduce additional uncontrollable states. Table 4.3 shows the number of uncontrollable and unobservable states with either T_a or V used as control inputs while the control input that is not selected is instead treated as a state. In addition to measurements for the thickness and temperature, measurement of either T_a and V is used for this particular test. Compared to the results form table 4.2, these two control setups only introduce more uncontrollable and unobservable states which fits well with the assumption above. The reason for the small variation of uncontrollable states are not known. Intuitively, it should be the same for both cases for the same reasons previously mention. An explanation could be round-off done by Matlab which in some occasions leaves some of the rows of the controllability matrix linearly dependent. Treating the non-selected control input as a constant instead of a state removes one uncontrollable state for each scenario in table 4.3.

Table 4.3: Number of uncontrollable and unobservable state with either T_a and V as control input.

Number of Elements	**T_a as Control Input**		**V as Control Input**	
	Nr. Uncontrollable States	**Nr. Unobservable States**	**Nr. Uncontrollable States**	**Nr. Unobservable States**
1	3	0	2	0
5	3	1	3	1
10	4	3	4	3
20	21	19	22	19
30	40	39	40	39
40	60	59	59	59

In addition, the solvent fraction node at the boundary is also used as control input togheter with T_a and V. Table 4.4 shows the number of the uncontrollable and unobservable states with C_n, T_a and V as control inputs. The measurements are limited to the thickness and temperature only. The results indicates that this setup reduces the number of uncontrollable states by four compared to the setup in table 4.2. However, the number of unobservable states increases by two. The reason for the decrease in uncontrollable state could be that by using C_n as a control input the control actions now acts on a large percentage of the solvent states compared to the case in table 4.2. T_a and V only affects the solvent fractions

indirectly through C_n, but as for this case, control actions with C_n directly acts on nearby solvent fraction nodes, thus expanding the influence of the control actions on the system. The reason for the increase in unobservable states is most likely due to measurements of L and T no longer can be used to determine the initial conditions of the solvent fraction node close to the boundary node as this node is now used as a control input.

Table 4.4: Number of uncontrollable and unobservable state with the boundary node, T_a and V as control inputs.

Number of Elements	Nr. Uncontrollable States	Nr. Unobservable States
1	0	0
5	0	0
10	0	4
20	13	21
30	33	40
40	52	60

In summary, the controllability of the linear system is dependent on the spatial discretization which is reasonable as an increasing number of elements implies more states. Despite being (in most cases) uncontrollable, this does not mean that system cannot be controlled at all. The controllablity has to be treated as an indication of the possibility of controlling the system as a whole. Thus, regardless of the system having uncontrollable states, the controllable states can still be forced to a desired value by control actions. Determining which of the states that is controllable is not straight forward and cannot be deducted from the controllablility matrix. However, with the observation made of the results from the different simulations carried out above, it is most likely that the states of interest (this is the thickness, temperature and solvent fraction at surface) is controllable. Thus, control of the system could still be applied to the system with success.

Comparison of Linear and Quadratic Shape Functions

Table 4.5 shows the number of the uncontrollable and unobservable states using linear shape functions instead of quadratics. Compared to table 4.2, there is no change in the number of uncontrollable states for up to 10 elements. However, from 20 and higher number of elements, there is a large difference in both number of uncontrollable and unobservable states. The main reason for the reduced number of uncontrollable and unobservable states is the reduced number of elements when using linear shape functions instead of quadratics. However, as the system becomes unobservable somewhere between 10 and 20 elements there is not much to gain in sense of accuracy using linear shape functions instead of quadratics for an estimator considering the additional error introduced by linear shape functions as shown in figure 4.5.

Table 4.5: Number of uncontrollable and unobservable state with T_a and V as control inputs for linear shape functions.

Number of Elements	2 Measurements		Every State Measured	
	Nr. Uncontrollable States	Nr. Unobservable States	Nr. Uncontrollable States	Nr. Unobservable States
1	1	0	1	0
5	2	0	2	0
10	2	0	2	0
20	2	3	2	0
30	8	9	8	0
40	17	18	17	0

4.2.4 Step Response

Figure 4.13 shows the plot of the measurements of the thickness and temperature of the linear system initialized at the stationary point when the control inputs are being excited. The thickness experiences a inverse response in which the thickness increases when the air temperature is excited positively. When the temperature reaches a stationary point, the thickness reduces to its initial value. The cause of the inverse response is not known. Most likely, it is caused by inaccurate modelling of the process, but this can not be concluded. The air velocity does not seem to have any affect on the thickness, nor the temperature. Intuitively, the air velocity should have some effect on the temperature development as it affects the heat transfer coefficient. However, a step size of 2 $\frac{m}{s}$ does not have any significant impact on the temperature gradient of the film. Furthermore, for the linear system, neither the air temperature and air velocity has any affect on the final thickness, which also implies no effect on the final solvent fraction. This fits well fit the observation made in the sensitivity analysis. As a consequence, control using the air temperature and air velocity at a stationary point does not make any sense. Instead, it is more reasonable to apply control using the air temperature and air velocity for controlling

the rate at which the solvent evaporates. As a further verification, figure 4.14 shows a plot with both the linear and non-linear system with the control inputs being excited. Clearly, both the linear and non-linear system has the same transient behavior in the thickness before settling at the operating point (there is a small error in the initial value of the non-linear model from the operating point in the thickness due to the initialization of the linear and non-linear model not being exact).

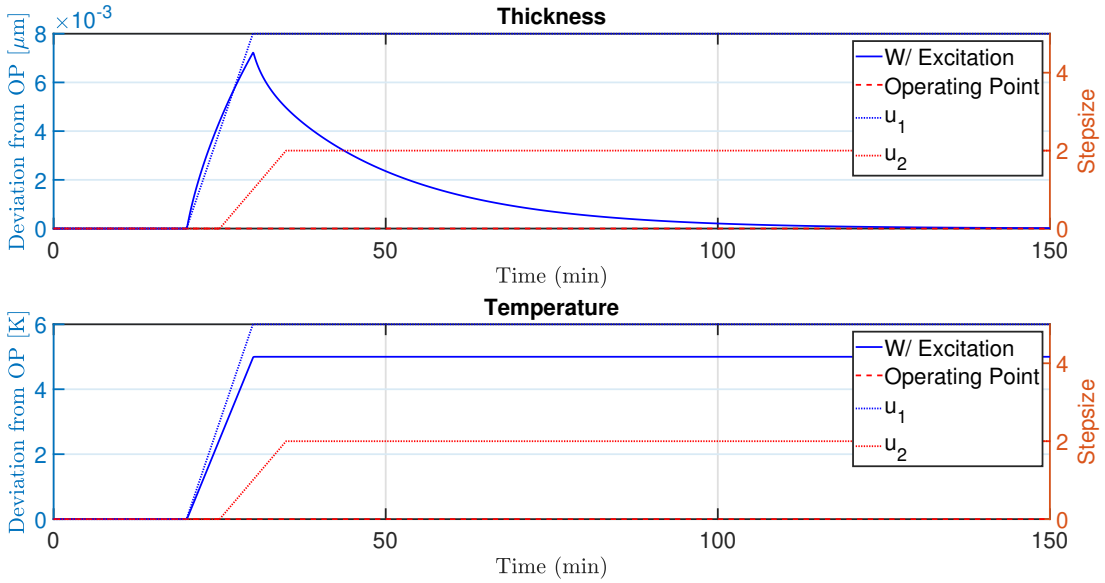


Figure 4.13: Step response of the linear system.

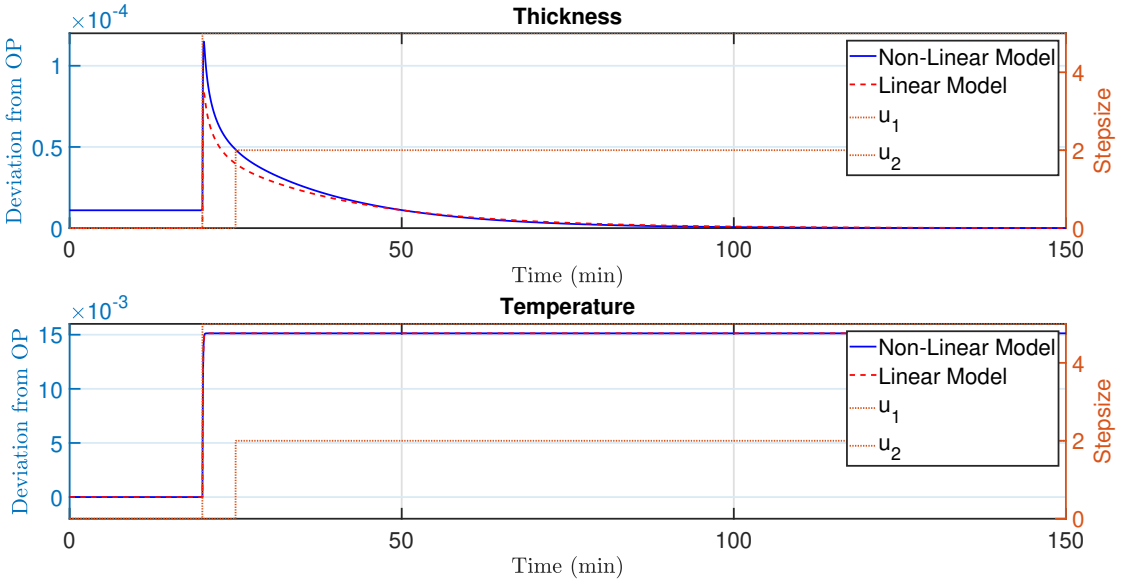


Figure 4.14: Comparison of step response of the linear and non-linear model.

4.2.5 Summary

The results above indicates that the linear model does experience stationary deviations for the thickness and density, as well as the reduction in solvent is very fast compared to that of the non-linear model. Also for small initial perturbations the linear model fails to converge to the linearization point for the same reason as the stationary deviations. Both the continuous and discrete linear model is marginally stable, and thus needs to be stabilized by feedback control in accordance with Hovd[37]. The controllability and observability of the system is dependent on the spatial discretization as increasing the mesh resolution implies more solvent fraction nodes. For a reasonable mesh resultion in time ans space, in terms of computational time and accuracy, the linear system is neither controllable nor observable. However, it is believed that the states of interest (thickness, temperature and solvent fraction at the boundary) is controllable, thus control by using T_a and V as control inputs can still be applied with success. The linear model behaves relative similar to the non-linear model in terms of step responses when both system is initialized at the stationary conditions. However, as neither T_a nor V has an influence on the final solvent moisture content, control at the stationary conditions is pointless considering that the drying process is over when these conditions are reached. As consequence, a conclusion is drawn that the linearized model is not an alternative for optimization and control purposes.

Considering using the model for an EKF, using linear shape functions instead of quadratics yields no particular benefits in terms of accuracy as the system becomes unobservable with approximately the same number of elements. As the system becomes unobservable for relatively small number of elements despite choosing linear shape functions, the computational time for an EKF algorithm would not improve significantly.

4.3 State Estimation

Due to exact measurement of every state can not be achieved in practice state estimation using a EKF is tested to asses whether or not such an estimator scheme can be applied for optimization and control purposes. From the linear analysis, a system configuration with T_a and V as inputs was observable for 5 elements for the spatial domain. However, it turns out that the non-linear model with similar system configuration is not. Thus, the number of elements used for the spatial domain is set to 2 to maintain an observable system.

4.3.1 Noise Magnitude

Prior to testing the EKF, reasonable noise magnitudes has to be determined. It is assumed that sensors for the temperature and film thickness has relative high accuracy, thus, the maximum off-sets in the thickness measurement should not exceed more than one percentage and even lower for the temperature measurement. Figure 4.15 shows the plots of the maximum offset in the measured values with increasing measurement noises. Note that the x-axis is logarithmic and given in watts. This is purely a result of the `wgn` function used to generate the noise as explained in section 3.3. The units of the noise is of no particular interest in practice, but it is used to generate noise which affects the system in such a way that can be expected in practice. From figure 4.15, it seems that the temperature is slightly more prone to measurement noise. Allowing up to one percent offset in the thickness measurement the maximum magnitude of the measurement is set to approximately -48 dBW. For the temperature a maximum offset of one percent translates into variations in the temperature measurement of +- 3.5-3.8 degrees Kelvin. An offset of 0.1 percent is much more realistic, thus, the temperature measurement noise is limited to approximately -72 dBW. Figure 4.16 shows a plot of the thickness and temperature measurement with sensor noise of -48 dBW and -72 dBW, respectively. It is assumed that the magnitude of the variations caused by the generated noise in figure 4.16 resembles realistic measurement data.

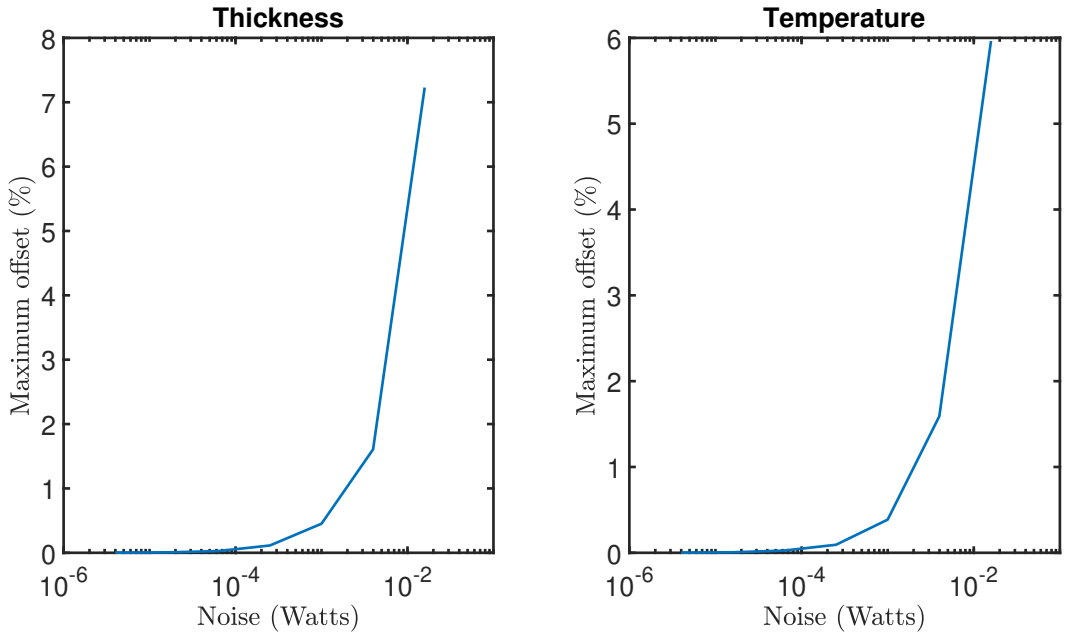


Figure 4.15: Maximum off-set due to measurement noise.

For the process noise it is harder to make a decent guess of the magnitude of the noise due to the uncertainties of the process. However, the model itself is a source of error which can be traced back to round-off errors, unmodelled disturbances etc. In comparison to the measurements noise, the process noise will also affect the states differently. Thus, different magnitudes of the process noise might be needed for the distinct states. Figure 4.17 shows the plot of the thickness, temperature, solvent fraction at the boundary and density with added process noise of -48 dBW. Both the thickness and solvent fraction is relatively unaffected by the process noise compared to the temperature and the density. As the temperature and density is much more affected by the process noise, the magnitude of the noise is reduced to -72 dBW for both states. As there is a large uncertainties related to the process noise, as well as larger emphasize on the measurement noise, the magnitude of the process noise for the temperature and density is reduced to -72 dBW for both states.

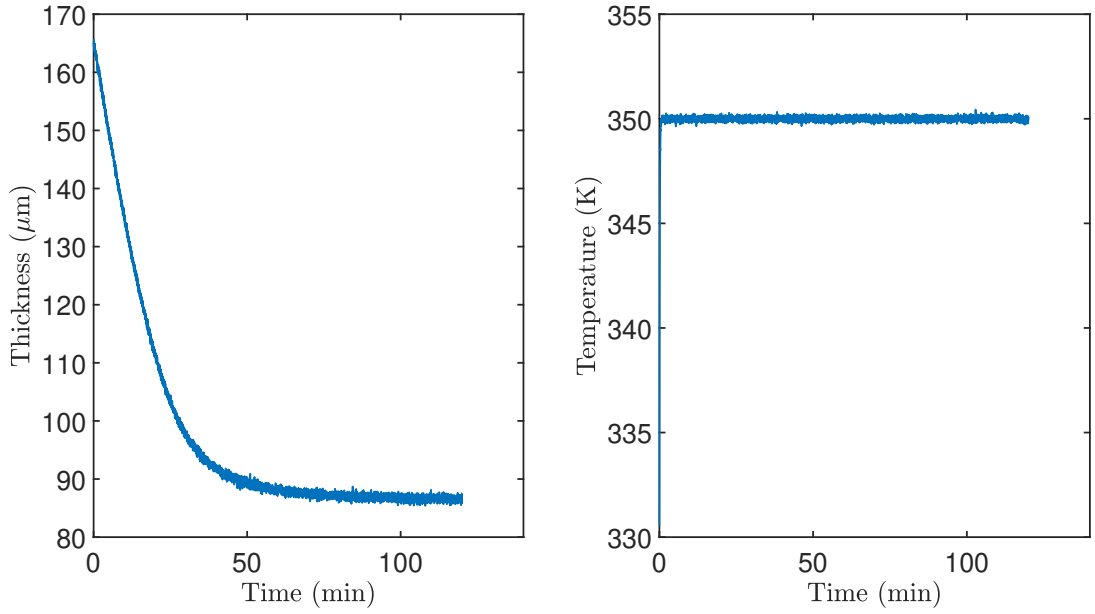


Figure 4.16: Measurement with -48 dBW magnitude noise for the thickness sensor and -72 dBW magnitude noise for the temperature sensor.

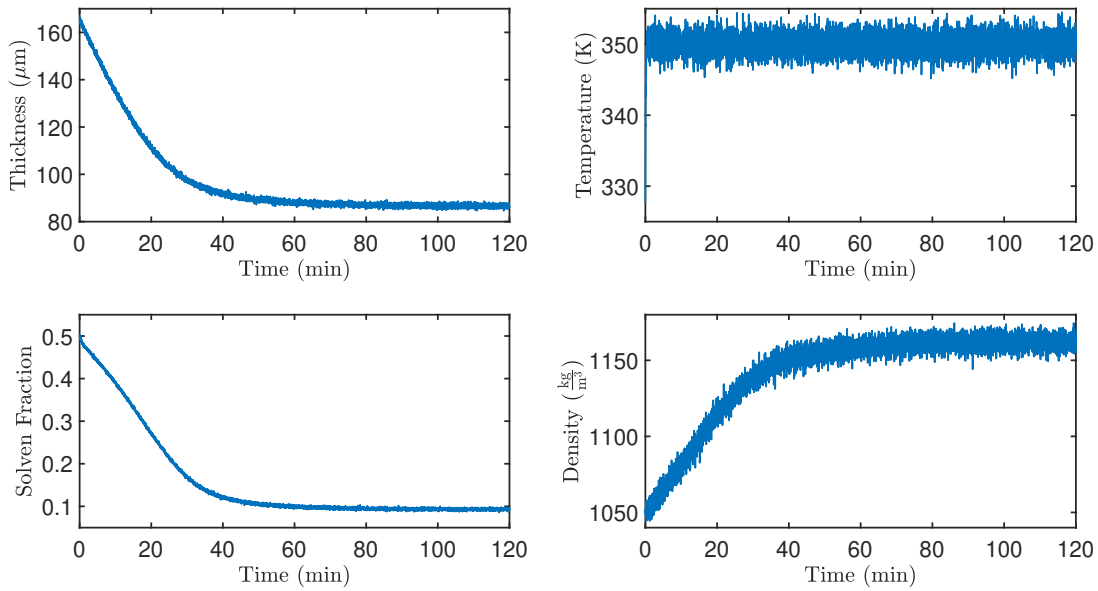


Figure 4.17: Solution of the non-linear model with added process noise.

4.3.2 Tuning And State Estimation

Initially the EKF was tested with both $\boldsymbol{Q}$ and $\boldsymbol{R}$ being diagonal matrices with ones at the diagonal and $\boldsymbol{P}_0$ set to zero. Figure 4.18 shows the estimated states with the initial tuning of the EKF. The estimates of the solvent fraction and density is good. The main reason for the well estimations of the density is the low process noise applied to the density. However, increasing either the process or measurement noise, the density estimation becomes more inaccurate. One can argue that the density is of no particular interest, hence, as long as the thickness, temperature and solvent fraction is well estimated inaccurate estimation of the density could be disregarded. Rather than disregarding the inaccurate estimation of the density, one could remove it completely from the estimation by implementing a reduced order EKF instead of that of a full order. In some situations, a reduced order observer/EKF improves the properties of the observer/EKF. Furthermore, the thickness and temperature estimation has unacceptable variations. Thus, the initial setup of $\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{P}_0$ does not suffice for accurate estimates. The EKF was further tuned by trail and error. Figure 4.19 shows the plots of the tuned EKF. Here $\boldsymbol{Q}$ is set to a diagonal matrix with 10^{-5} on its diagonals except for the thickness, temperature and density, which was set to 0.1, 0.1 and 10, respectively. $\boldsymbol{R}$ is set to $diag(10, 2)$ and $\boldsymbol{P}_0$ to $\mathbf{0}$. As the initial setup estimated both the density and the solvent fraction well, there is no significant improvement with the tuned EKF. However, there is a clear improvement in the estimation of the thickness and the temperature as the variations in the estimate is significantly reduced. The reasoning for the given tuning values is due to how the different parameters are affected by process and measurement noise. The solvent fraction is relatively unaffected compared to the other parameters, thus, the solution of the non-linear model can be ”trusted” and have a low covariance. For the thickness and temperature we expect a larger variation than for the solvent fraction, thus the covariance is increased compared to the one of the solvent fraction. Lastly, for the density it is excepted an even larger variation and the covariance for the density is further increased. For the $\boldsymbol{R}$ matrix, it is assumed that there is larger expected variations in the thickness measurement due to the small scale. Hence, the element of the thickness is set higher compared to the one of the temperature. Furthermore, the initial covariance matrix $\boldsymbol{P}_0$ is set to zero as it is assumed that the film is initialized with perfect accuracy.

Figure 4.20 shows the error between the estimated states and the solution of the non-linear model. With the relative small error, the figure reflects the high accuracy of the estimated states as observed above. However, the error in the density increases as time goes on, implying that the error in the density estimated by the EKF increases with increased density. For the rest of the states, the error seems to be uniformly distributed.

The control inputs are excited during the simulations to ensure that the estimation behaves well during input changes. The air velocity is excited with a negative change of $\Delta 5$ at approximately 13 minutes, and the air temperature is excited with a positive change of $\Delta 10$ at 20 minutes. From both figure 4.18 and figure 4.19 it is clear that the EKF behave well even with input changes. This suggest that the EKF can be implemented in control structures without rendering the overall system unstable. However, large process noise do make the EKF unstable as singularities occurs when inverting a close to zero Jacobian. As a consequence, the EKF can not be implemented in situations with large process noise. This is also the case with a large resolution of the spatial domain. An alternative to EKF is the UKF which do not use an inverted Jacobian. Thus, singularities will no longer be of concern. As a suggestion for further work, an UKF could be implemented instead of an EKF to ensure stability.

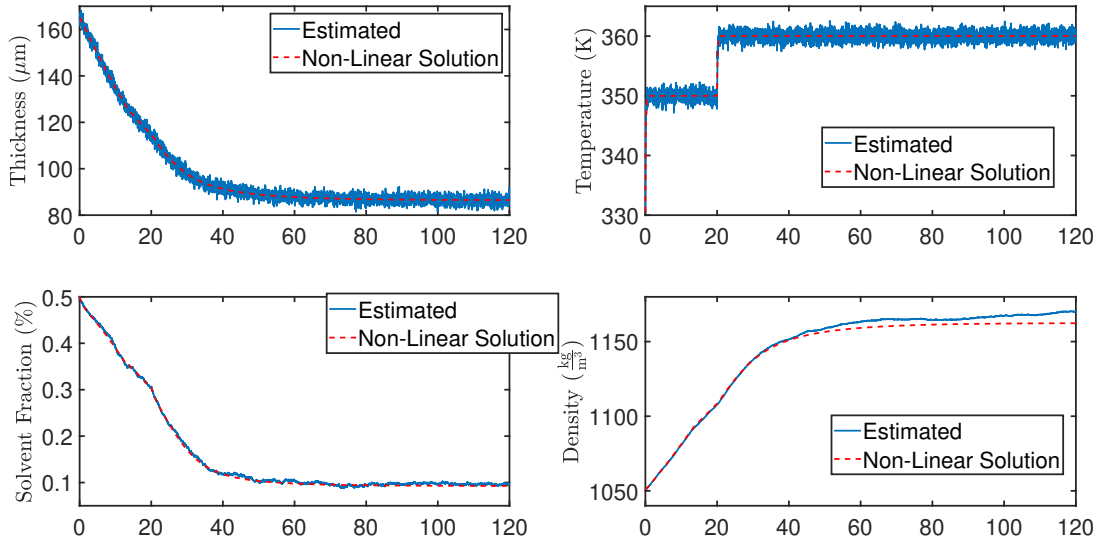


Figure 4.18: The estimated states prior to tuning $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{P_0}$.

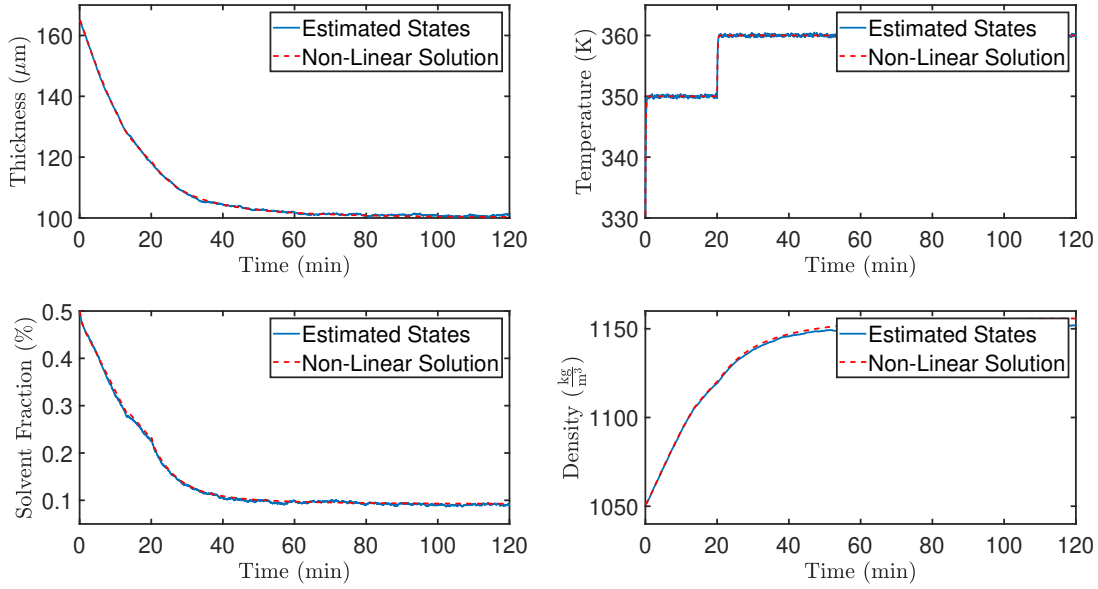


Figure 4.19: The estimated states with $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{P_0}$ tuned.

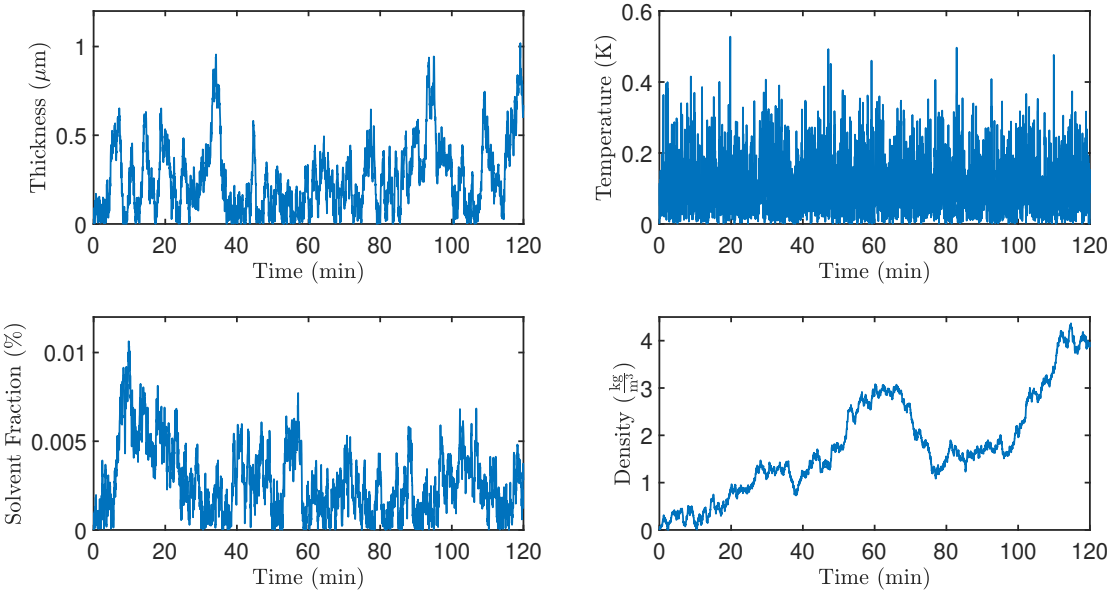


Figure 4.20: The error between the estimated states and the non-linear solution.

4.4 Optimization

For the optimization of the process, the model is implemented as described in section 3.4 and tested using the parameter values in table 4.1. The tests is carried out with different constraint with respect to maximum evaporation rate to assess how much of a reduction in heat consumption that can be achieved. The main reason for using the evaporation rate as the limiting factor of the process is its close relation to crack formation in the material as mentioned in section 2.1. As described in section 3.4, the drying process is divided into three separate stages, each with its distinct temperature. Different length of the three zones is also tested to see whether or not this has any effect on the reduction in heat consumption. At first, the process is optimized with respect to the air temperature only by treating the air velocity as a constant. Later, the process is optimized with respect to both air temperature and air velocity to assess whether or not this is more beneficial in terms of energy reduction.

4.4.1 Optimization With Respect to Air Temperature

Figure 4.21 shows the plot of the thickness, temperature, solvent fraction and evaporation rate for both simulation with and without optimal control inputs. In this particular case the evaporation limit is set to $5 \cdot 10^{-5}\ \frac{kg}{s}$ (this is approximately the maxima of the evaporation rate of the non-optimized solution), and the transition between the different drying zones are set to 6 and 20 minutes with the total drying time being 80 minutes. For the thickness and solvent fraction there is not much difference between the optimized and non-optimized trajectory. This can be explained by that there is no particular difference in the evaporation rate. However, when the thin film enters the third and final drying section, the air temperature as well as the evaporation rate drops which is reflected by a slower thickness and solvent fraction development. The reduction in heat needed for the drying process is reduced by 7.75%. However, the drying time is increased by 19.4%. Furthermore, figure 4.22 shows the comparison of the non-optimized and optimized trajectory with the drying zone transition set at $t_1 = 20$ and $t_2 = 40$ minutes, while the evaporation constraint is unchanged. For the first drying section the optimized and non-optimized remains the same as the air temperature is unchanged. However, as the thin film enters the second section, the air temperature increases to about 364 K which increases the evaporation of the solvent. A closer look reveals that the evaporation rate exceeds the upper limit set for the evaporation rate. This is due to the constraint tolerance set for the `fmincon`-function and the relative small scale of the evaporation. Decreasing the tolerance will keep the evaporation rate from exceeding the constraint at the expanse of the air temperature. Nevertheless, reducing the tolerance will most likely yield a small reduction in the air temperature. Compared to the first configuration of the drying sections, the heat consumption is now reduced by 25.32%. In addition, the drying time is reduced by 33.67%. The reduction in drying time should not be taken too literally, as by changing the threshold at which the drying is considered to be over does changed the overall reduction. However, it gives some indication on whether the drying time decreases or increases.

Table 4.6 is a comparison of configurations with different upper evaporation rate bound and drying section separations. Similar plots for the different configurations as figure 4.21 and figure 4.22 can be found in figure E.6-figure E.5 in appendix E. The table show that separating the the drying section more evenly yields higher reduction in both drying time and heat consumption. A three-zoned drying unit clearly reduces the heat consumption by utilizing the temperature related latent heat of evaporation which decreases with increased film temperature. By trail and error, the optimal separation of the drying section turned out to be approximately around $t_1 = 20$ and $t_2 = 40$ minutes. By extending the optimization problem to include for variable section separation this can be determined more accurately. Extending the optimization problem can further optimize the drying process and can be a part of future work. Furthermore, increasing the upper evaporation limit speeds up the drying process as well as reducing the heat needed even more as it allows for utilization of even higher temperatures. However, these results are representative for this particular battery composition only, hence, other composition will experience different reduction in heat consumption and drying time. Nevertheless, it is reasonable to assume that any composition will behave in a similar manner. Of course, the upper bounds on the evaporation rate has to be determined experimentally for any battery composition. But the results clearly indicates that a multi-section drying unit is more efficient in terms of heat consumption and drying time compared to a single stage drying unit. The benefits of multi-stage drying units are also documented in other published literature which further verifies the results[55, 56, 23].

Table 4.6: Listing of reduction in heat consumption and drying time and optimal control inputs based on evaporation rate constraint and drying section separation.

E_{bound} $(\frac{kg}{s})$	t_1 (min)	t_2 (min)	$Q_{red}(\%)$	$t_{red}(\%)$	u_1 (K)	u_2 (K)	u_3 (K)
$4 \cdot 10^{-5}$	6	20	-9.1	-6	330.5	346.1	351.2
	20	40	12.1	-11.9	345.5	353.1	335.6
$5 \cdot 10^{-5}$	6	20	7.8	-19.39	350.2	352.1	340.1
	20	40	25.3	33.67	350.2	363.9	330.5
$7 \cdot 10^{-5}$	6	20	25.72	-3.9	357.7	361.1	339.2
	20	40	31.7	41.4	357.6	370	330.5

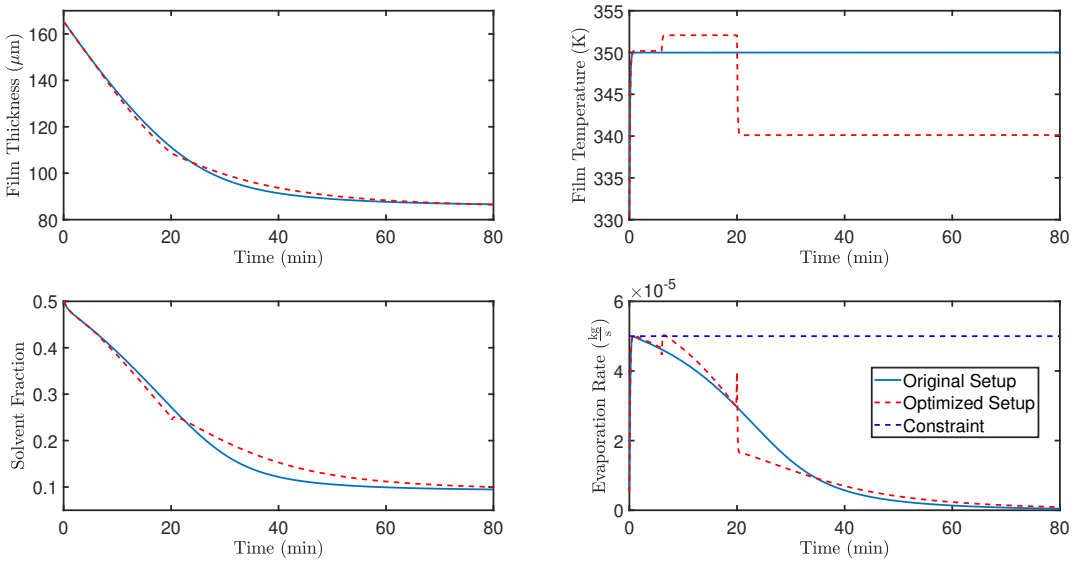


Figure 4.21: Comparison of the numerical solution with and without optimized air temperature control input with the constrain $e_i \leq 5 \cdot 10^{-5}$ $\frac{kg}{s}$. The drying zone transition is set at $t_1 = 6$ and $t_2 = 20$ min.

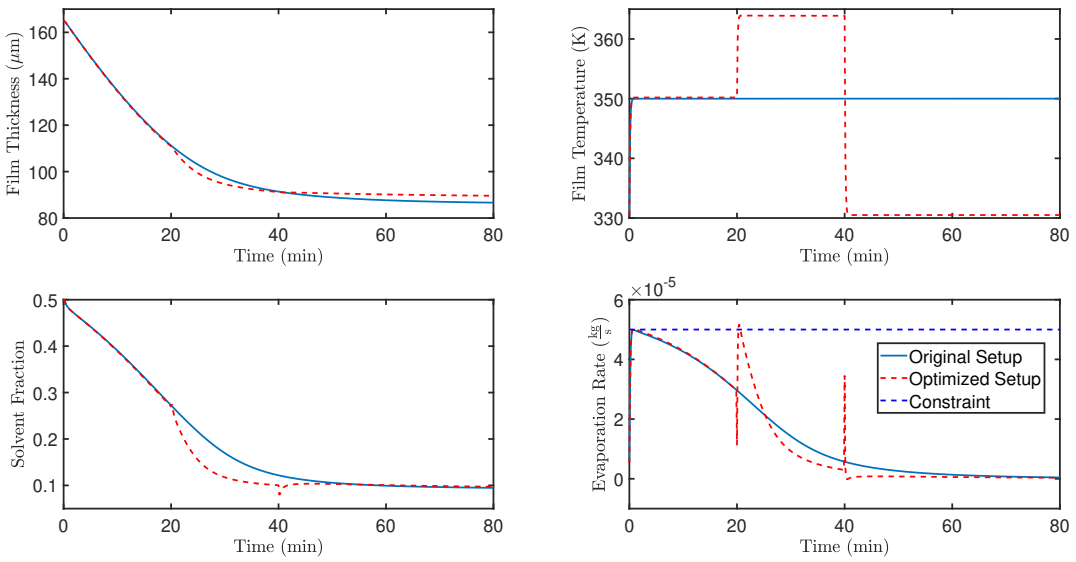


Figure 4.22: Comparison of the numerical solution with and without optimized air temperature control input with the constrain $e_i \leq 5 \cdot 10^{-5}$ $\frac{kg}{s}$. The drying zone transition is set at $t_1 = 20$ and $t_2 = 40$ min.

4.4.2 Optimization With Respect To Air Temperature And Air Velocity

Figure 4.23 shows the the comparison of the non-optimized trajectory and the optimized trajectory with respect to the air temperature and air velocity. The upper bound on the evaporation rate is set to $5 \cdot 10^{-5}$ $\frac{kg}{s}$. The different drying stages were separated at $t_1 = 20$ and $t_2 = 40$ minutes. The reduction in energy consumption is 14.9%, while the reduction in drying time is found to be 33.7%. Compared to the optimal trajectory with the same setup when only the air temperature is treated as an optimization variable, the reduction in the energy consumption is approximately 10% lower. However, the reduction in drying time is the same. Furthermore, table 4.7 lists the optimal control inputs calculated for two different setups along with the reduction in energy needed for the process and reduction in drying time.

Table 4.7: Listing of reduction in heat consumption and drying time and optimal control inputs based on evaporation rate constraint and drying section separation.

E_{bound} $(\frac{kg}{s})$	t_1 (min)	t_2 (min)	$Q_{red}(\%)$	$t_{red}(\%)$	u_1 (K)	u_2 (K)	u_3 $(\frac{m}{s})$	v_1 $(\frac{m}{s})$)	v_2 (K)	v_3 $(\frac{m}{s})$)
$5 \cdot 10^{-5}$	20	40	14.9	33.7	369.3	370.0	330.5	5.0	9.7	15.0
$7 \cdot 10^{-5}$	20	40	16.0	39.7	370.0	370.0	330.5	5.0	15.0	15.0

Compared to the results from the previous section, the reduction in both energy consumption and drying time is lower. The reason for the decreased optimal point when both the air temperature and air velocity are treated as optimization variables is not known. However, by looking at figure 4.24, the evaporation rate peaks when the thin film enters the second drying stage which could be sign of the air velocity's relation to the mass transfer coefficient causing the weird results. Despite the higher air temperature for the optimized trajectory, the evaporation rate follows the same trajectory as the non-optimized solution within the first drying stage due to the much lower air velocity. However, as the thin film enters the second drying stage, only the air velocity changes from 5 $\frac{m}{s}$ to 15 $\frac{m}{s}$ which causes a large spike in the evaporation rate. The spikes in the evaporation are mainly caused as the changes in the control inputs happens instantaneously. In practice, the thin film would experience a smoother change with respect to the control input as it is transferred from one drying stage to another. However, a similar observation can be made for the optimized trajectories when the air velocity is treated as a constant in the previous section. Thus, it cannot be concluded that air velocity's affect on the mass transfer function is the cause of the unexpected results. However, it suggested as future work to further investigate the cause.

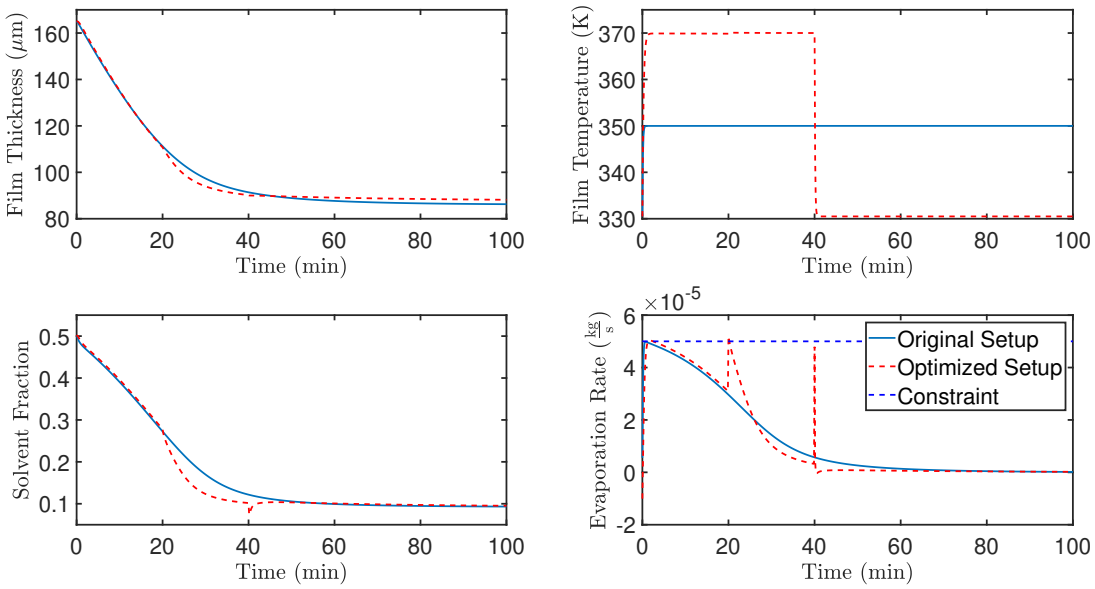


Figure 4.23: Comparison of the numerical solution with and without optimized control inputs with the constrain $e_i \leq 5 \cdot 10^{-5}$ $\frac{kg}{s}$. The drying zone transition was set at $t_1 = 6$ and $t_2 = 20$ min.

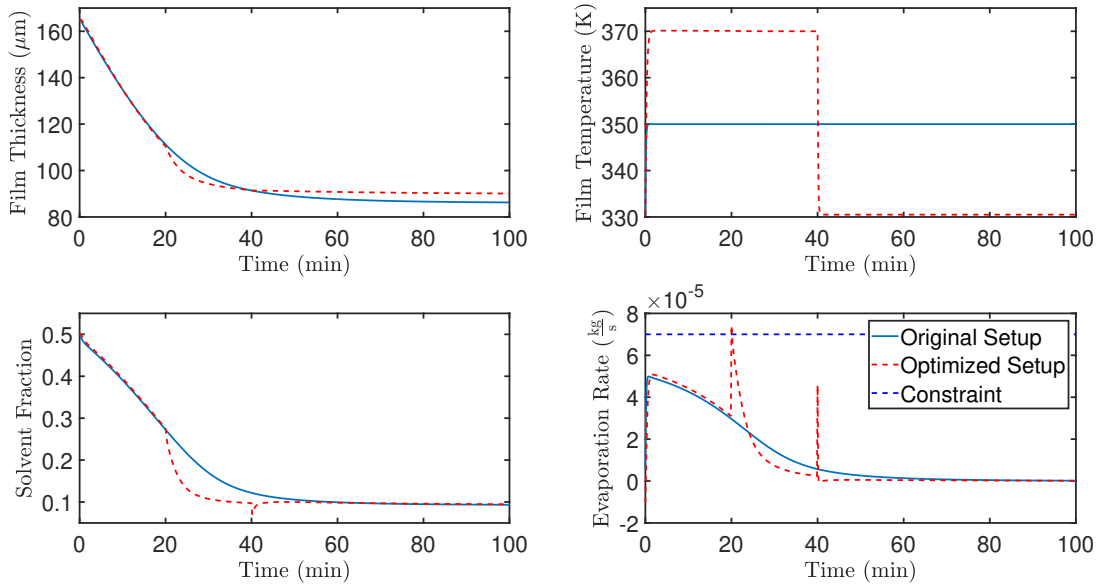


Figure 4.24: Comparison of the numerical solution with and without optimized control inputs with the constrain $e_i \leq 7 \cdot 10^{-5} \frac{kg}{s}$. The drying zone transition was set at $t_1 = 20$ and $t_2 = 40$ min.

4.4.3 Further Notes

The optimization problem is formulated such that it does not take into account the energy needed to heat up the air temperature nor energy needed for the fans. One can argue that this is the case for both the optimized and non-optimized solution so that it evens out. However, higher air temperatures and velocities would require more energy to produce. An interesting future development could be to investigate the total energy reduction of the overall system in terms of heat and exergy analysis in which heat recovery method is utilized. An exergy analysis can provide useful information of the quality of the energy and to what extent heat recovery method is useful. From a control perspective, the elapsed time for solving the optimization problem is approximately half a minute. Considering implementing a MPC, the current computational time is too high as an MPC solves the optimization problem at each iteration which in turn will invoke large internal delays. Having larger computational power at hand, which could solve the optimization problem in a reasonable time frame, would enable the implementation of an MPC. However, as this is not the case, controlling the process using P/PI/PID controllers with feedback is more reasonable.

For the optimal trajectories with a upper bound on the evaporation rate at $7 \cdot 10^{-5} \frac{kg}{s}$, the final thickness deviates from that of the non-optimized trajectory. A similar observation was made in the linear analysis section, although on a larger scale. However, this indicates that there is some faults in the relation between the evaporation rate and the thickness of the film as it seems that the solvent fraction is not affected in the same way. This can be related to the way the partial pressure of the solvent is calculated by using Flory Huggin's theory or how the partial pressure is coupled with the thickness. This should be investigated for further verification. However, the relatively small deviation along with the model already discarded from performance optimization and control with respect to the thickness should not have significant impact on the results presented above.

4.5 Control

In this section, the results from the controlled system are presented and discussed. However, prior to controlling the overall system, the performance of each controller is assessed. The controllers are tested with disturbances of different frequency to assess whether or not the control system is exposed for disturbances of a particular frequency and the performance in general.

4.5.1 Controller Performance

To begin with, the processes that the proposed controllers are to control are modelled in an attempt to replicate an actual process that occur in the battery industry. The actual models of the heating and acceleration systems are not assumed to be of high accuracy. However, the main focus is to replicate behaviors of the control inputs that one might find in the industry. Therefore, as long as the control system is tuned properly and the output signals behave some what realistically, the accuracy of the actual control process models are of no particular concern.

The proposed controllers are:

$$K_T = \frac{1.5s + 0.1}{s}, \quad K_V = \frac{3s + 5}{s}$$

Air Temperature Controller

For the air temperature controller the gain and phase margin are found to be 3 and $59°$. The gain margin is well above the recommended value of $GM \geq 2$, while the phase margin is slightly below the recommended value of $60°$. With the definition of the bandwidth frequency as stated in section 2.6, it is found to be approximately at $\omega_b = 0.36\ \frac{rad}{s}$. Within the bandwidth region $||GK|| \gg ||G_d||$ which indicates good disturbance rejection within the bandwidth. $||G_d||$ falls well below 1 around the bandwidth. Thus, disturbances of frequency larger than ω_b is dampen by the process it self. As a consequence, control is not needed at these frequencies. In addition, with the reference being constant at zero, $||GK|| \gg ||r||$ within the bandwidth which indicates good reference tracking. $||GK||$ also drops well below 1 for frequencies above ω_b implying low sensitivity to measurement noises. Finally, the peak values M_S and M_T is found to be at 1.62 and 1.01, respectively. This indicates good control performance within the bandwidth as well as the control system is robust to model uncertainties. Figure 4.25 shows the Bode-plot of the system with the proposed controller for the air temperature process.

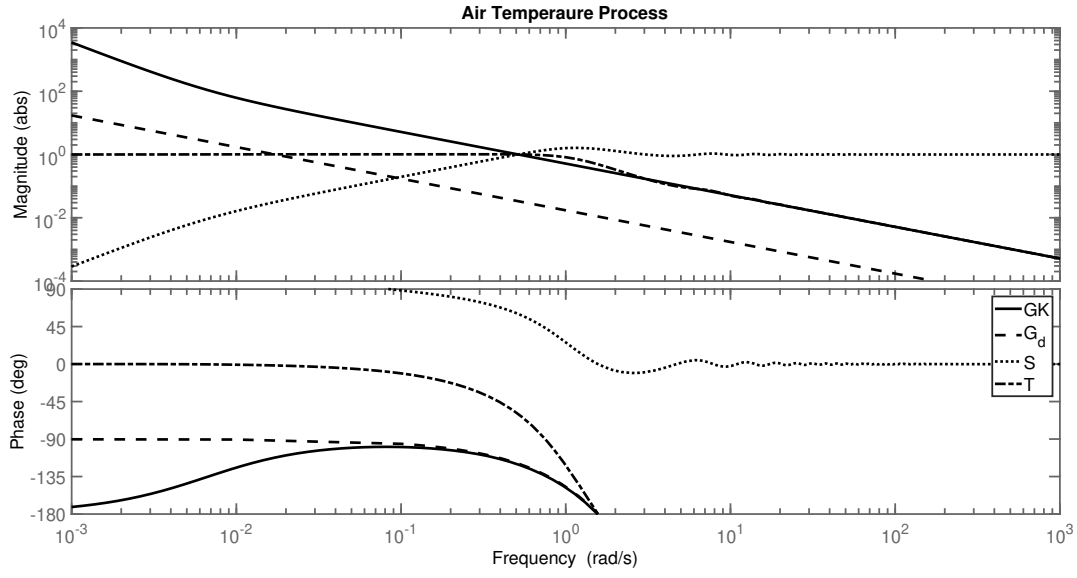


Figure 4.25: Frequency response for the air temperature heating process.

Figure 4.26 shows simulations of the temperature control system with the proposed PI controller with disturbances of increasing frequency. As the figure shows the system becomes less affected disturbances of increasing frequency. This fits well with the information obtained from the Bode-plot in figure 4.25 with a decreasing magnitude of G_d. This shows that even though the disturbance signal is of a frequency higher than the bandwidth, the process remains close to its reference

value. This is due to the disturbance been dampen by the process it self rather than the controller efficiently correcting the deviation in the error. However, at frequencies within the bandwidth it is clear that the controller is performing well as the uncontrolled output deviates with about 50 Kelvin while the controlled output stays close to the reference value. For the controlled system, the temperature deviates at most by 1 Kelvin, which is considered to be good when compared to the accurate temperature control for industrial processes developed by Li and Yu in [57] (which deviates with 0.5 Kelvin). Furthermore, figure 4.27 shows the saturated and unsaturated control inputs during the simulation. The scaled control input is peaking at approximately 0.05. With a saturation limit of $|u_{max}| = 1$, the control input is far from being saturated and the process uses little of the available control input. This implies that scaling with respect to u could be better. Nevertheless, the controller manages to keep the output value within a close proximity of the reference value despite disturbances.

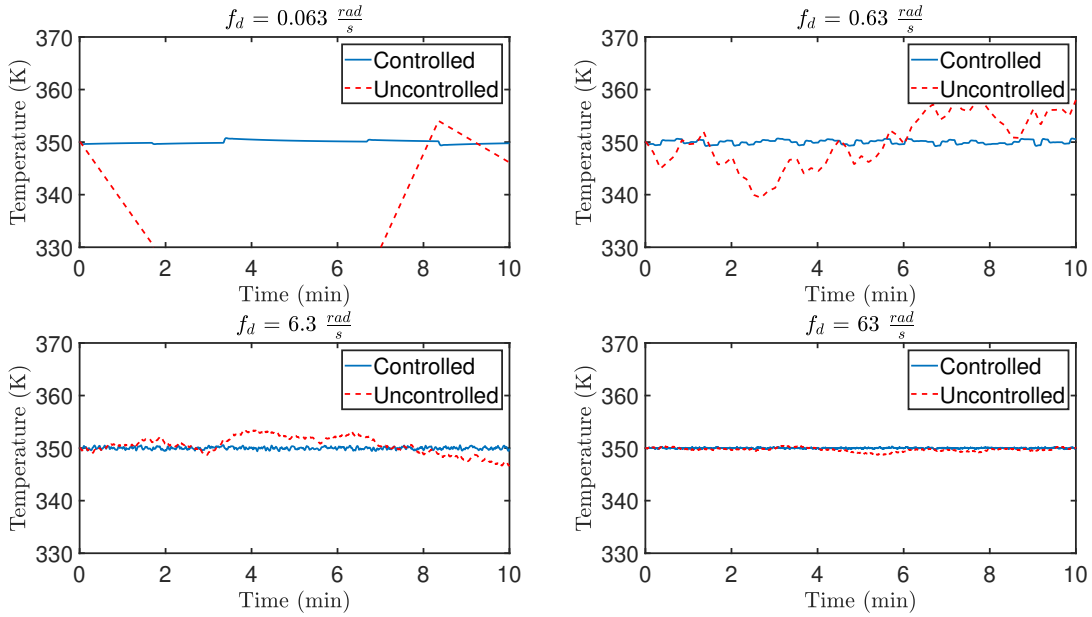


Figure 4.26: Simulation of the temperature control with disturbances of different frequency.

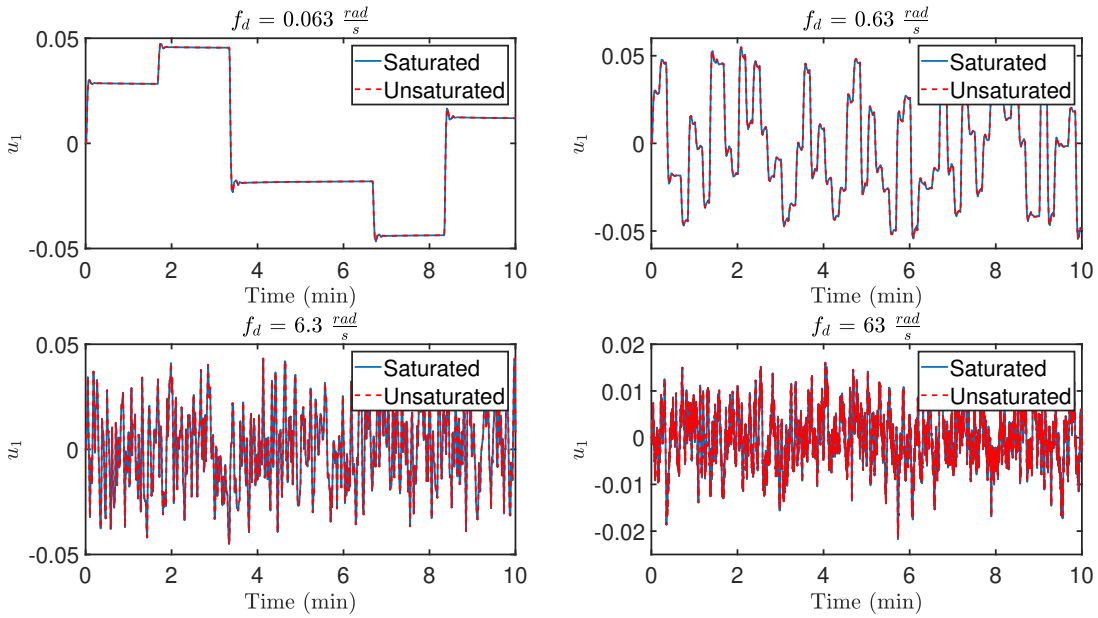


Figure 4.27: Saturated and unsaturated control input for the temperature process with disturbances of different frequency.

Air Velocity Controller

The Bode-plot of the air velocity process with the proposed controller is illustrated in figure 4.28. The gain and phase margin are found to be approximately 2.09 and $72°$. This is slightly above recommend considering the gain margin. However, the phase margin is well above the recommended value of $60°$. Thus, the system is far from being unstable. The bandwidth frequency is approximately $0.56\ \frac{rad}{s}$, limiting the controller to frequencies up to roughly $1\ \frac{rad}{s}$. Within the bandwidth, $||GK|| \gg ||G_d||$ which implies good disturbance rejection. In addition, the magnitude of GK is much lager than $||r||$. Thus, good reference tracking can also be expected. However, both $||GK||$ and $||G_d||$ remains close to 1 for all frequencies above ω_b. As a consequence, disturbances with frequency in this range wont be rejected and thus, affecting the process with a larger magnitude than for lower frequencies. Furthermore, M_S and M_T are found to be 1.91 and 0.91. Although this is below recommended, both S and T oscillates peaking at M_S and M_T for frequencies above ω_b, further indicating that control at these frequencies is poor. Nevertheless, due to the input delay, the control performance of the system is difficult to improve. Thus, the controller is left unchanged.

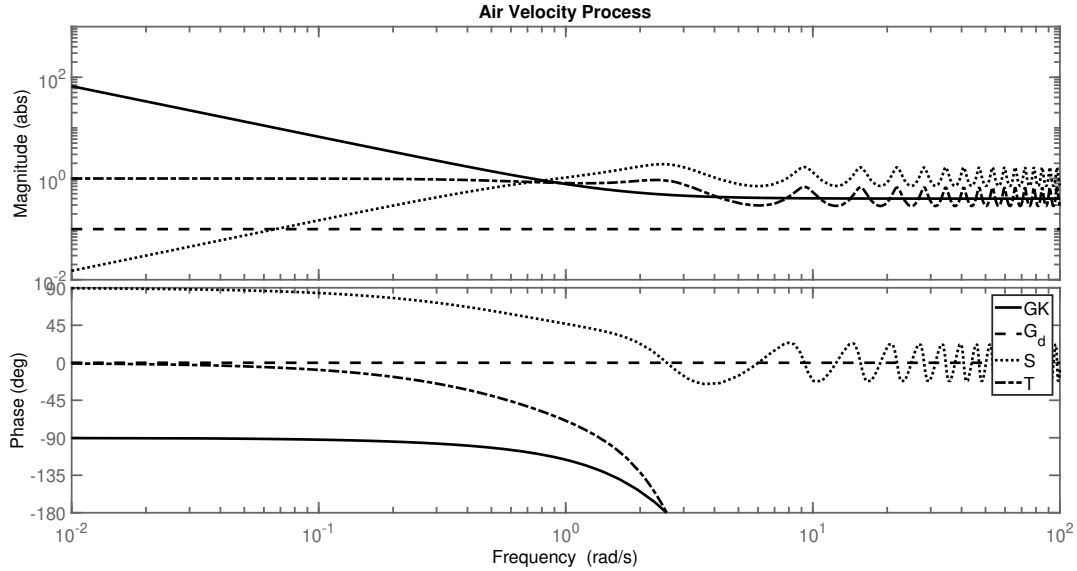


Figure 4.28: Frequency response for the air velocity process.

Figure 4.29 shows the simulations of the air velocity control system with disturbance of increasing frequency. As the process is scalar function, the disturbance is implemented with a bias to represent forces which works in the opposite direction of the control input. This is mainly done to show that the process is in need of a integrator to correct for the stationary offset that is created by the disturbance. As the figure shows, for every simulation the controller manage to correct for the stationary deviation. However, as the frequency of the disturbance increases, the output value becomes increasingly oscillatory. Considering the observation made when studying the Bode-plot, this behavior is reasonable. Furthermore, figure 4.30 shows the saturated and unsaturated control inputs during the simulation. With a saturation limit at $|u_{max}| = 1$, the control input remains within the range of the actuator. Compared to the magnitude of the control inputs for the temperature controller, the scaling of the air velocity process with respect to the control input is more appropriate. However, in events with changes in the reference, the control input is likely to become saturated. Nevertheless, as reference changes only occurs at start up of the drying process this is of little concern. In summary, the controller performs well within the bandwidth with decent disturbance rejection. However, as the frequency increases the control system fails to reject disturbances.

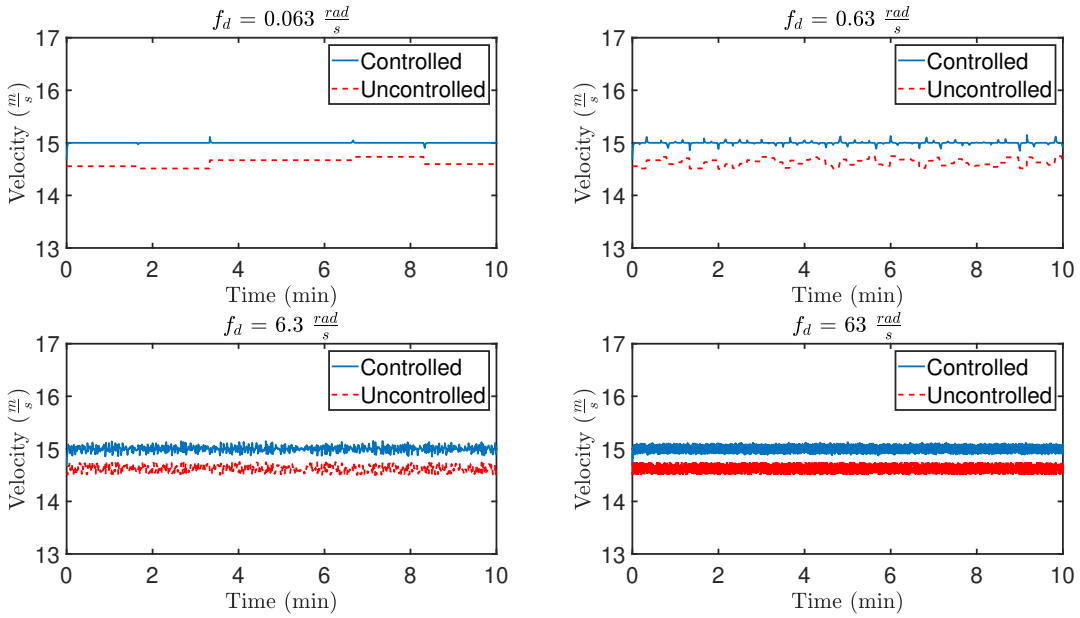


Figure 4.29: Simulation of the air velocity control with disturbances of different frequency.

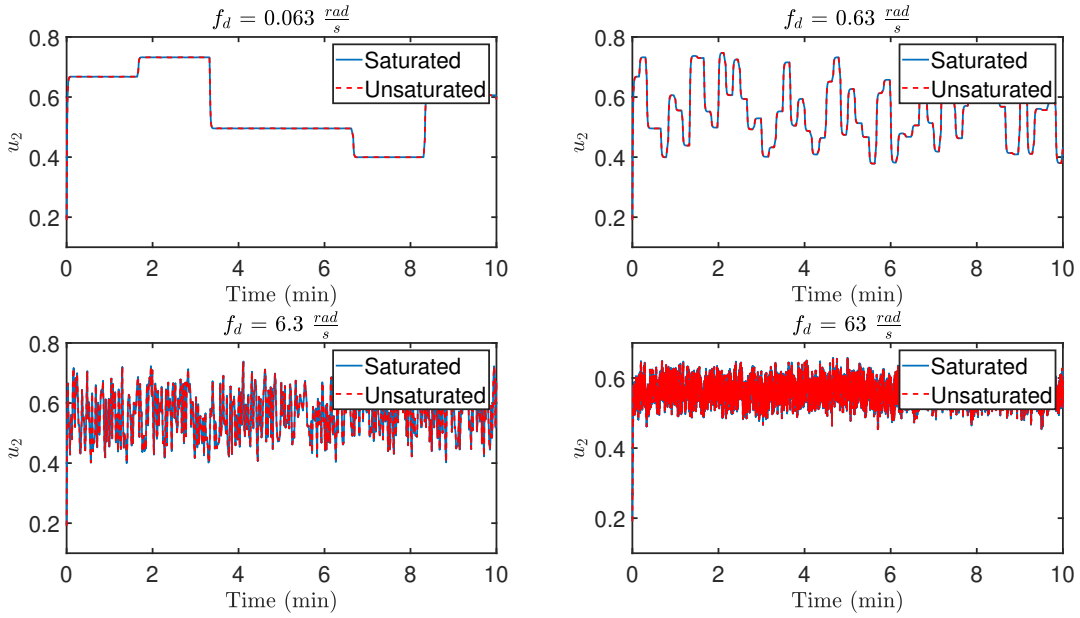


Figure 4.30: Saturated and unsaturated control input for the air velocity process with disturbances of different frequency.

4.5.2 Control of the Overall System

For controlling the complete system, the controllers are set to follow the optimal trajectory for the control inputs obtained from the optimization. Table 4.8 lists the parameters from the selected configuration from section 4.4.

Table 4.8: List of parameter selected for the simulation.

Parameter	$t_1\ (min)$	$t_2\ (min)$	$T_{a,1}\ (K)$	$T_{a,2}\ (K)$	$T_{a,3}\ (K)$	$V\ (\frac{m}{s})$	$E_{bound}(\frac{kg}{s})$
Value	20	40	350.2	363.9	330.5	15	$5 \cdot 10^{-5}$

Figure 4.31 shows a comparison of the optimal trajectory and the estimated states from the simulation of the complete system. In this particular case, the control processes are excited by disturbances at $0.63\ \frac{rad}{s}$ which are within the bandwidth of both controllers. The system manages to follow the optimal trajectory with high precision despite the disturbances in the control process and the measurement and process noises for the sensor and the plant. However, such accuracy can not be expected in reality as there are several factors which is not considered when designing the control systems. A significant reason for the high accuracy is that the "unknown" plant is modelled using the proposed model. As both the plant and the EKF uses the same model to produce the output and estimates, the only thing causing variations between the two are the measurements and process noises and control inputs. However, as figure 4.32 shows, the control inputs implemented on the plant is far from constant which implies that the output of the system is not significantly affected by variations in the control input. Nevertheless, a conclusion whether or not the overall system is fit for control purposes can not be drawn without testing the model on a physical plant.

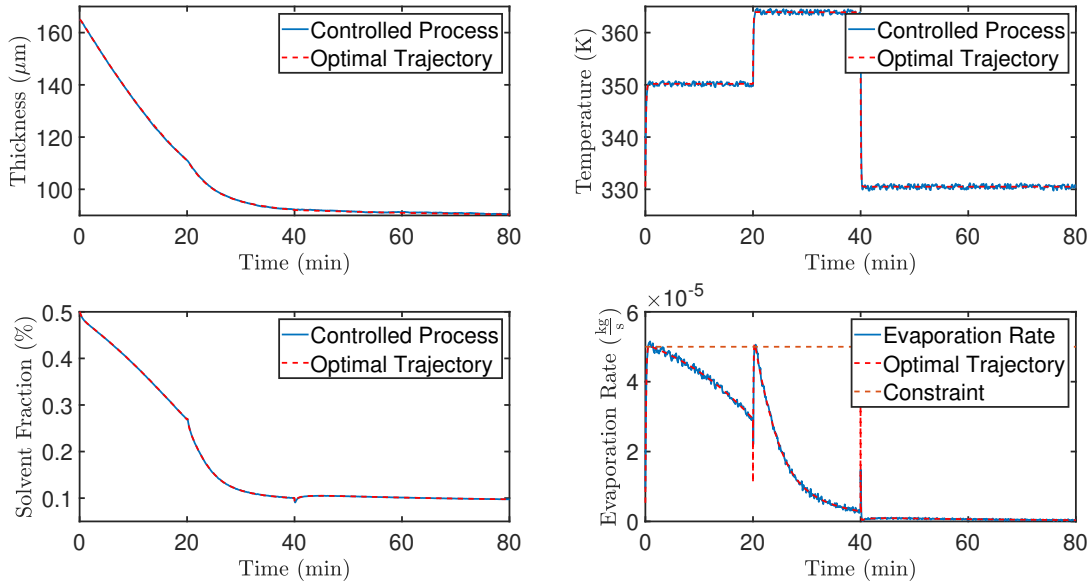


Figure 4.31: Comparison of the optimal trajectory and estimated states from the simulation when the control process is excited by disturbances at $0.63\ \frac{rad}{s}$.

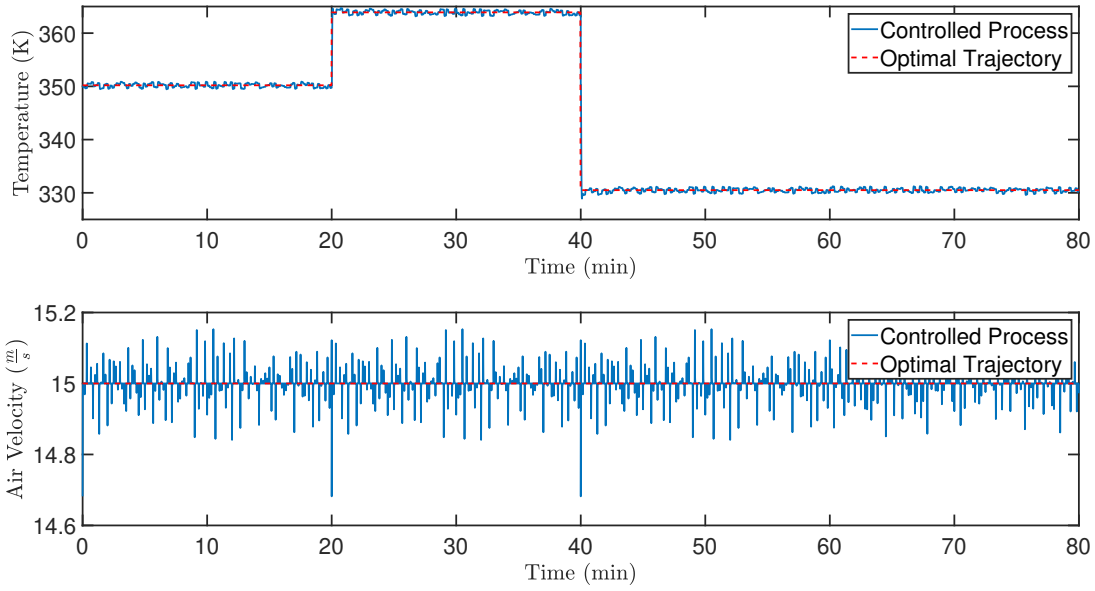


Figure 4.32: Comparison of the optimal trajectory and simulated inputs when the control process is excited by disturbances at $0.63\ \frac{rad}{s}$.

Chapter 5

Conclusion

The purpose of this thesis was to derive a model which describes the drying process for electrodes used in battery production which could be used in optimization and control.

A one dimensional model was derived using theorems from fluid- and thermo-dynamics with the parameters initial solvent content, initial thin film temperature, heat source temperature and air velocity. The system were solved by applying the FEM in combination with 4th order Runge-Kutta method to the model in Matlab®. The numerical method became mesh independent at roughly 200 elements. However, as the model was to be used for control and optimization purposes, a resolution in time and space of approximately 1-2 seconds and 20 elements was found to be a good compromise between accuracy and computational time. Furthermore, comparison of the solution using linear and quadratic shape functions showed that with increasing number of elements the difference in the solution decreased exponentially. In addition, the difference in computational time increased exponentially, indicating that linear shape functions was most reasonable for a large number of elements. Finally, simulations showed that the numerical method conserves mass and energy.

The model showed promising behavior when comparing it with other published literature on the similar topics. However, the model does not account for the formation of the porous media during the drying process. As a consequence, the model was unfit for optimization and control with respect to performance in which the film thickness was of importance. Despite this, the model was fit for optimization and control with respect to energy as the model conserves mass and energy. Furthermore, a sensitivity analysis indicated that all of the parameters had varying effects on the drying process. The air temperature and air velocity had the greatest influence, while the drying process was least affected by changes in the initial thin film temperature. This was mainly due to the parameters influence on the vapor pressure and the mass transport coefficient which were the limiting factors for the evaporation rate. However, none of the parameters affected the final moisture content, which was solely dependent on the relative humidity of the air.

In attempt to simplify the task at hand, the model was linearized about the stationary conditions. The linear model shared similar behavior to the non-linear model. However, the dynamics of the linear model was much faster compared to the non-linear model. In addition, the linear model experienced a large deviation in the film thickness. The main reason for the large difference was large initial perturbation from the linearization point and that the process was highly non-linear. As a result, the linearized model was not a suitable alternative to the non-linear model for optimization and control purposes. Nevertheless, the linear model was used to assess the controllability and observability of the system which indicated that the model is uncontrollable despite the number of elements used for the finite element approximation. However, the analysis indicated that the system were observable for a small number of elements, indicating that a Kalman filter could be used for estimating the states.

An EKF was derived for estimating the states. Due to lack of a physical plant or data, measurements were produced using the derived model and appropriate measurement and process noise. The different states were affected differently by both process and measurement noise, with the density being the state that was affected the most. The initial configuration of unity for the EKF already showed promising performance. However, by tuning the accuracy of the estimated states were further improved. Despite the promising performance showed by the EKF algorithm, the EKF was prone to singularities when the Jacobian became close to zero. Thus, it was recommended to implement an UKF in future work.

The process were optimized with respect to the chosen control inputs (air temperature and air velocity) in order to reduce the energy needed for the process. In addition, the drying process was divided into three separate stages in an attempt to

utilize higher air temperatures for a lower amount of time. The results clearly indicated that a multi-stage dryer reduced both energy and time for the drying process. This was found to be in accordance with other published work. However, unexpected results was obtained when the process was optimized using both the air temperature and air velocity as optimization variables. The air velocity's influence of the mass transfer coefficient was pointed out as a possible reason. However, further investigation was suggested.

Due to lack of a physical plant, simple models for heating the air temperature and acceleration was derived in an attempt to recreate an actual drying process. Simple PI-controller were designed through the loop shaping technique to control the systems. The control system managed to replicate inputs which can be expected in practice. However, despite a successfully implementation and simulation of the overall system, a conclusion other than that the model showed promising performance for optimization and control for energy minimization purpose could not been drawn.

Chapter 6

Outlook and Further Work

In this thesis, several topics related to the drying process of a battery electrode has been visited. Although the thesis has arrived at some conclusion based on the results presented, further investigated should be carried to further verify discussed subject. In addition, the model and the numerical method could be improved upon in terms of accuracy and efficiency. As a brief summarization, the list below contains suggested topics for future work:

- Further development of the model to account for the porosity of the material, as well as looking into the relation between the thickness and the evaporation rate.
- Further validation of the model by obtain accurate values for parameters and properties for a distinct composition.
- Further improving the numerical method in terms of stability and efficiency.
- Implementing a UKF for state estimation to avoid problems related to singularities.
- Further investigation of the problem related to optimization with both the air temperature and air velocity as optimization variables.
- Comparing the overall energy reduction for multi-stage drying units with heat recovery methods implemented by applying exergy analysis.
- Further verifying the accuracy of the overall system by implementing the model on a physical plant.

Bibliography

[1] Emil Oppegård. Review and modeling of battery electrode drying process, 2020.

[2] Salah [Ben Mabrouk], Elaid Benali, and Hatem Oueslati. Experimental study and numerical modelling of drying characteristics of apple slices. *Food and Bioproducts Processing*, 90(4):719 – 728, 2012.

[3] M Mohammadi Amin, Avinash Bhaskar Gangurde, and V. Alai. Oral film technology: Challenges and future scope for pharmaceutical industry. 2015.

[4] Chris Yuan, Yelin Deng, Tonghui Li, and Fan Yang. Manufacturing energy analysis of lithium ion battery pack for electric vehicles. *CIRP Annals*, 66(1):53–56, January 2017.

[5] Zafer Erbay and Filiz Icier. A review of thin layer drying of foods: Theory, modeling, and experimental results. *Critical reviews in food science and nutrition*, 50:441–64, May 2010.

[6] Bastian Westphal, Henrike Bockholt, T. Günther, W. Haselrieder, and Arno Kwade. Influence of convective drying parameters on electrode performance and physical electrode properties. *ECS Transactions*, 64(22):57–68, April 2015.

[7] Norsk elbilforening statistikk elbil. `https://elbil.no/elbilstatistikk/`. Accessed: 08-06-2020.

[8] Autovista Group covid-19 could derail ev uptake, increasing oem emissions and fines. `https://autovistagroup.com/news-and-insights/covid-19-could-derail-ev-uptake-increasing-oem-emissions-and-fines`. Accessed: 08-06-2020.

[9] Bo Zeng and Chuan Li. Forecasting the natural gas demand in china using a self-adapting intelligent grey model. *Energy*, 112:810 – 825, 2016.

[10] F. Gerard Adams and Yochanan Shachmurove. Modeling and forecasting energy consumption in china: Implications for chinese energy demand and imports in 2020. *Energy Economics*, 30(3):1263 – 1278, 2008.

[11] K.E. Stensby G. Holm Midttømme A.M. Ruud L. Lia, T. Jensen. The current status of hydropower development and dam construction in norway. 2015.

[12] Electricity - annually- ssb. `https://www.ssb.no/en/energi-og-industri/statistikker/elektrisitet/aar`. Accessed: 2020-05-26.

[13] Arno Kwade, Wolfgang Haselrieder, Ruben Leithoff, Armin Modlinger, Franz Dietrich, and Klaus Droeder. Current status and challenges for automotive battery production technologies. *Nature Energy*, 3(4):290–300, April 2018.

[14] Shabbir Ahmed, Paul A. Nelson, Kevin G. Gallagher, and Dennis W. Dees. Energy impact of cathode drying and solvent recovery during lithium-ion battery manufacturing. *Journal of Power Sources*, 322:169–178, August 2016.

[15] Nam-Soon Choi, Se-Young Ha, Yongwon Lee, Jun Yeong Jang, Myung-Hwan Jeong, Woo Cheol Shin, and Makoto Ue. Recent progress on polymeric binders for silicon anodes in lithium-ion batteries. 6, June 2015.

[16] Samira Maazi, Amir H. Navarchian, Mohsen Khosravi, and Pu Chen. Effect of poly (vinylidene fluoride)/poly (vinyl acetate) blend composition as cathode binder on electrochemical performances of aqueous li-ion battery. *Solid State Ionics*, 320:84–91, July 2018.

[17] Z. P. Cai, Y. Liang, W. S. Li, L. D. Xing, and Y. H. Liao. Preparation and performances of lifepo4 cathode in aqueous solvent with polyacrylic acid as a binder. *Journal of Power Sources*, 189(1):547–551, April 2009.

[18] Daniel I. Onwude, Norhashila Hashim, Rimfiel B. Janius, Nazmi Mat Nawi, and Khalina Abdan. Modeling the thin-layer drying of fruits and vegetables: A review. *COMPREHENSIVE REVIEWS IN FOOD SCIENCE AND FOOD SAFETY*, 15(3):599–618, February 2020.

[19] A Aboltins, J Palabinskis, et al. Fruit drying process investigation in infrared film dryer. *Agronomy research*, 14(1):5–13, 2016.

[20] Chang Heo, Hyunjun Cho, and Yeong-Koo Yeo. Dynamic modeling of paper drying processes. *Korean Journal of Chemical Engineering*, 28:1651–1657, 08 2011.

[21] Marcello Murru, Giovanni Giorgio, Sara Montomoli, Francois Ricard, and Frantisek Stepanek. Model-based scale-up of vacuum contact drying of pharmaceutical compounds. *Chemical Engineering Science*, 66(21):5045 – 5054, 2011.

[22] Sandeep Karki, Hyeongmin Kim, Seon-Jeong Na, Dohyun Shin, Kanghee Jo, and Jaehwi Lee. Thin films as an emerging platform for drug delivery. *Asian Journal of Pharmaceutical Sciences*, 11(5):559 – 574, 2016.

[23] Ali Mesbah, Ashlee N. Ford Versypt, Xiaoxiang Zhu, and Richard D. Braatz. Nonlinear model-based control of thin-film drying for continuous pharmaceutical manufacturing. *Ind. Eng. Chem. Res.*, 53(18):7447–7460, May 2014.

[24] Naresh Susarla, Shabbir Ahmed, and Dennis W. Dees. Modeling and analysis of solvent removal during li-ion battery electrode drying. *Journal of Power Sources*, 378:660–670, February 2018.

[25] Jana Kumberg, Marcus Müller, Ralf Diehm, Sandro Spiegel, Christian Wachsmann, Werner Bauer, Philip Scharfer, and Wilhelm Schabel. Drying of lithium-ion battery anodes for use in high-energy cells: Influence of electrode thickness on drying time, adhesion, and crack formation. *Energy Technology*, 7(11):1900722, 2019.

[26] A Mathematical, Model of the, Drying Process, and A. K. Haghi. Acta polytechnica vol. 41 no. Technical Report http://ctn.cvut.cz/ap/, 2001.

[27] M. H. Jacobs. *Diffusion Processes*, pages 1–145. Springer Berlin Heidelberg, Berlin, Heidelberg, 1935.

[28] Saturated vapor pressure. `http://ddbonline.ddbst.com/AntoineCalculation/AntoineCalculationCGI.exe?component=N-Methyl-2-pyrrolidone`. Accessed: 28-05-2020.

[29] Milad Asgarpour Khansary. Vapor pressure and flory-huggins interaction parameters in binary polymeric solutions. *Korean Journal of Chemical Engineering*, 33(4):1402–1407, 2016.

[30] Dortmund Data Bank heat of vaporization of n-methyl-2-pyrrolidone. `http://www.ddbst.com/en/EED/PCP/HVP_C284.php`. Accessed: 15-05-2020.

[31] John Fauvel. Algorithms in the pre-calculus classroom: Who was newton-raphson? *Mathematics in School*, 27(4):45–47, 1998.

[32] Emil Oppegård, Asanthi Jinasena, Anders Hammer Strømman, Jon Are Suul, and Odne Stokke Burheim. Study of an industrial electrode dryer of a lithium-ion battery manufacturing plant: Dynamic modelling. EasyChair Preprint no. 4314, EasyChair, 2020.

[33] Whokko Schirén. Finite element method for 1d transient convective heat transfer problems. 2018.

[34] James F Epperson. *An introduction to numerical methods and analysis*. John Wiley & Sons, 2013.

[35] Olav Egeland and Jan Tommy Gravdahl. *Modeling and simulation for automatic control*, volume 76. Marine Cybernetics Trondheim, Norway, 2002.

[36] Wikimedia Commons. File:runge-kutta slopes.svg — wikimedia commons, the free media repository, 2020. [Online; accessed 6-December-2020].

[37] Morten Hovd. Lecture notes for the course advanced control of industrial processes. *Department of Engineering Cybernetics*, 2009.

[38] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, Inc., USA, 3rd edition, 1998.

[39] *Nonlinear Kalman filtering*, chapter 13, pages 393–431. John Wiley Sons, Ltd, 2006.

[40] Gabriel A Terejanu et al. Extended kalman filter tutorial. *University at Buffalo*, 2008.

[41] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[42] Hubertus Th Jongen, Klaus Meer, and Eberhard Triesch. *Optimization theory*. Springer Science & Business Media, 2007.

[43] Murray R Spiegel. *Laplace transforms*. McGraw-Hill New York, 1965.

[44] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley Sons, Inc., Hoboken, NJ, USA, 2005.

[45] OA Olsen. Industrielle målemetoder. *Odd Arild Olsen forlag*, 2015.

[46] fun.

[47] David Vernon Widder. *The heat equation*, volume 67. Academic Press, 1976.

[48] Air - density, specific weight and thermal expansion coefficient at varying temperature and constant pressures.

[49] Air - density, specific weight and thermal expansion coefficient at varying temperature and constant pressures. `https://www.engineeringtoolbox.com/air-density-specific-weight-d_600.html`.

[50] Y. Cengel and J. Cimbala. *Fluid Mechanics Fundamentals and Applications: Third Edition*. McGraw-Hill Higher Education, 2013.

[51] Marco Ludovico-Marques, Carlos Chastre, Abdel Salam Hamdy Makhlouf, and Mahmood Aliofkhazraei. Chapter 10 - effect of artificial accelerated salt weathering on physical and mechanical behavior of sandstone samples from surface reservoirs. In *Handbook of Materials Failure Analysis with Case Studies from the Oil and Gas Industry*, pages 215–233. Butterworth-Heinemann, 2016.

[52] R. P. King and R. P. King. 6 - sedimentation and thickening. In *Introduction to Practical Fluid Flow*, pages 159–194. Butterworth-Heinemann, Oxford, 2002.

[53] W. G. E. van Ballegooijen, A. M. van Loon, and A. J. J. van der Zanden 1. Modelling diffusion-limited drying behaviour in a batch fluidized bed dryer. *null*, 15(3-4):837–855, January 1997.

[54] Sorin Micu and Enrique Zuazua. An introduction to the controllability of partial differential equations. *Quelques questions de théorie du contrôle. Sari, T., ed., Collection Travaux en Cours Hermann, to appear*, 2004.

[55] M. Djaeni, P. Bartels, J. Sanders, G. van Straten, and A. J. B. van Boxtel. Multistage zeolite drying for energy-efficient drying. *Drying Technology*, 25(6):1053–1067, 2007.

[56] NA Artyukhova, AB Shandyba, and A Ye Artyukhov. Energy efficiency assessment of multi-stage convective drying of concentrates and mineral raw materials. *Scientific Bulletin of National Mining University*, (1), 2014.

[57] X. Li and Y. Yu. A high accuracy temperature control system based on arm9. In *2011 International Conference on Electrical and Control Engineering*, pages 23–26, 2011.

Appendices

Appendix A

Derivation of Dimensionless Model

Substituting equation (2.16a) and equation (2.16b) into equation (2.8) we get:

$$\frac{\mathrm{d}(\tilde{\rho_f}\rho_0)}{\mathrm{d}(\tilde{t}\mathrm{t_s})} = \frac{\rho_s\rho_{solid}\cdot(\rho_s-\rho_{solid})}{(\tilde{C}C_0\cdot\cdot(\rho_s-\rho_{solid})+\rho_{solid})^2}\cdot\dot{\tilde{C}}C_0 \tag{A.1a}$$

$$\frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} = \frac{C_0\rho_s\rho_{solid}\cdot(\rho_s-\rho_{solid})\cdot\mathrm{t_s}}{\rho_0\cdot(\tilde{C}C_0\cdot\cdot(\rho_s-\rho_{solid})+\rho_{solid})^2}\cdot\dot{\tilde{C}} \tag{A.1b}$$

By substituting equation (2.16a) and equation (2.16c) into equation (2.9) we get:

$$\frac{\mathrm{d}(\tilde{L}\mathrm{L_0})}{\mathrm{d}(\tilde{t}\mathrm{t_s})} = -\frac{\mathrm{k_m M_s}}{\rho_s\mathrm{R}}\cdot\left(\frac{P_s}{\tilde{T}T_0}-\frac{\mathrm{P_a}}{\mathrm{T_a}}\right) \tag{A.2a}$$

$$\frac{\mathrm{L_0}}{\mathrm{t_s}}\frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} = -\frac{\mathrm{k_m M_s}}{\rho_s\mathrm{R}}\cdot\left(\frac{P_s}{\tilde{T}T_0}-\frac{\mathrm{P_a}}{\mathrm{T_a}}\right) \tag{A.2b}$$

$$\frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} = -\frac{\mathrm{k_m M_s t_s}}{\rho_s\mathrm{R L_0}}\cdot\left(\frac{P_s}{\tilde{T}T_0}-\frac{\mathrm{P_a}}{\mathrm{T_a}}\right) \tag{A.2c}$$

Furthermore, by substituting equation (2.16a), equation (2.16e) and equation (2.16c) into equation (2.10) we get:

$$\tilde{\rho_f}\rho_0\tilde{L}\mathrm{L_0}\frac{\mathrm{d}(\tilde{T}\mathrm{T_0})}{\mathrm{d}(\tilde{t}\mathrm{t_s})}+\tilde{\rho_f}\rho_0\tilde{T}\mathrm{T_0}\frac{\mathrm{d}(\tilde{L}\mathrm{L_0})}{\mathrm{d}(\tilde{t}\mathrm{t_s})}+\tilde{L}\mathrm{L_0}\tilde{T}\mathrm{T_0}\frac{\mathrm{d}(\tilde{\rho_f}\rho_0)}{\mathrm{d}(\tilde{t}\mathrm{t_s})} = -\frac{\mathrm{k_m M_s}h_{fg}}{\mathrm{C_p R}}\cdot\left(\frac{P_s}{\tilde{T}\mathrm{T_0}}-\frac{\mathrm{P_a}}{\mathrm{T_a}}\right)+\frac{\dot{q}_{in}}{\mathrm{C_p}} \tag{A.3a}$$

$$\tilde{\rho_f}\rho_0\tilde{L}\mathrm{L_0}\frac{\mathrm{T_0}}{\mathrm{t_s}}\frac{\mathrm{d}\tilde{T}}{\mathrm{d}\tilde{t}}+\tilde{\rho_f}\rho_0\tilde{T}\mathrm{T_0}\frac{\mathrm{L_0}}{\mathrm{t_s}}\frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}}+\tilde{L}\mathrm{L_0}\tilde{T}\mathrm{T_0}\frac{\rho_0}{t_s}\frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} = -\frac{\mathrm{k_m M_s}h_{fg}}{\mathrm{C_p R}}\cdot\left(\frac{P_s}{\tilde{T}\mathrm{T_0}}-\frac{\mathrm{P_a}}{\mathrm{T_a}}\right)+\frac{\dot{q}_{in}}{\mathrm{C_p}} \tag{A.3b}$$

$$\frac{\mathrm{d}\tilde{T}}{\mathrm{d}\tilde{t}} = -\frac{\mathrm{k_m M_s t_s}h_{fg}}{\tilde{\rho_f}\rho_0\mathrm{C_p R}\tilde{L}\mathrm{L_0}}\cdot\left(\frac{P_s}{\tilde{T}{\mathrm{T_0}}^2}-\frac{\mathrm{P_a}}{\mathrm{T_a T_0}}\right)+\frac{\mathrm{t_s}}{\tilde{\rho_f}\rho_0\mathrm{C_p}\tilde{L}\mathrm{L_0 T_0}}\cdot\dot{q}_{in}-\frac{\tilde{T}}{\tilde{L}}\frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}}-\frac{\tilde{T}}{\tilde{\rho_f}}\frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} \tag{A.3c}$$

Finally, by substituting equation (2.16a), equation (2.16f), equation (2.16g) and equation (2.16d) into equation (2.1) we get:

$$\frac{\partial(\tilde{C}\mathrm{C_0})}{\partial(\tilde{t}\mathrm{t_s})} = \frac{\partial}{\partial(\tilde{z}L)}\left(\tilde{D}(\tilde{C}C_0,\tilde{T}T_0)\mathrm{D_0}\frac{\partial(\tilde{C}\mathrm{C_0})}{\partial\,(\tilde{z}L)}\right) \tag{A.4a}$$

$$\frac{\mathrm{C_0}}{\mathrm{t_s}}\frac{\partial\tilde{C}}{\partial\tilde{t}} = \frac{\mathrm{D_0 C_0}}{\left(\tilde{L}\mathrm{L_0}\right)^2}\frac{\partial}{\partial\tilde{z}}\left(\tilde{D}(\tilde{C}C_0,\tilde{T}T_0)\frac{\partial\tilde{C}}{\partial\tilde{z}}\right) \tag{A.4b}$$

$$\frac{\partial\tilde{C}}{\partial\tilde{t}} = \frac{\mathrm{D_0 t_s}}{\left(\tilde{L}\mathrm{L_0}\right)^2}\frac{\partial}{\partial\tilde{z}}\left(\tilde{D}(\tilde{C}C_0,\tilde{T}T_0)\frac{\partial\tilde{C}}{\partial\tilde{z}}\right) \tag{A.4c}$$

For the boundary condition at $\tilde{z} = 1$:

$$\tilde{D_0}\mathrm{D_0}\frac{\partial(\tilde{C}\mathrm{C_0})}{\partial(\tilde{z}L)} + \tilde{C}\mathrm{C_0}\frac{\mathrm{d}(\tilde{L}\mathrm{L_0})}{\mathrm{d}(\tilde{t}\mathrm{t_s})} = -\frac{\mathrm{k_m M_s}}{\rho_\mathrm{s}\mathrm{R}} \cdot \left(\frac{P_s}{\tilde{T}\mathrm{T_0}} - \frac{\mathrm{P_a}}{\mathrm{T_a}}\right) \tag{A.5a}$$

$$\frac{\tilde{D_0}\mathrm{D_0C_0}}{\tilde{L}\mathrm{L_0}}\frac{\partial\tilde{C}}{\partial\tilde{z}} + \frac{\tilde{C}\mathrm{C_0}\tilde{L}\mathrm{L_0}}{\mathrm{t_s}}\frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} = -\frac{\mathrm{k_m M_s}}{\rho_\mathrm{s}\mathrm{R}} \cdot \left(\frac{P_s}{\tilde{T}\mathrm{T_0}} - \frac{\mathrm{P_a}}{\mathrm{T_a}}\right) \tag{A.5b}$$

Appendix B

Derivation of Extended Kalman Filter Equations

Using the linearized model equation (2.78) linearized about the priori estimate yields the following:

$$\begin{aligned}\boldsymbol{P_k^-} &= E[(\boldsymbol{x_k} - \boldsymbol{x_k^-})(\boldsymbol{x_k} - \boldsymbol{x_k^-})^\top] \\ &= E[(\boldsymbol{J_x}(\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1}^+}) + \boldsymbol{J_w}\boldsymbol{w_{k-1}})(\boldsymbol{J_x}(\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1}^+}) + \boldsymbol{J_w}\boldsymbol{w_{k-1}})^\top] \\ &= E[(\boldsymbol{J_x}(\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1}^+})(\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1}^+})^\top \boldsymbol{J_x}^\top + \boldsymbol{J_x}(\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1}^+})\boldsymbol{w_{k-1}}^\top \boldsymbol{J_w}^\top \\ &+ \boldsymbol{J_w}\boldsymbol{w_{k-1}}(\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1}^+})^\top + \boldsymbol{J_w}\boldsymbol{w_{k-1}}\boldsymbol{w_{k-1}}^\top \boldsymbol{J_w}^\top]\end{aligned}$$

Noting that the process noise has no correlation with the states as well as recalling that $\boldsymbol{R}$ represents the covariance matrix of the process noise, the priori estimate of the covariance matrix, $\boldsymbol{P_k^-}$ is reduced to:

$$\begin{aligned}\boldsymbol{P_k^-} &= E[\boldsymbol{J_x}(\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1}^+})(\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1}^+})^\top \boldsymbol{J_x}^\top + \boldsymbol{J_w}\boldsymbol{w_{k-1}}\boldsymbol{w_{k-1}}^\top \boldsymbol{J_w}^\top] \\ \boldsymbol{P_k^-} &= \boldsymbol{J_x} E[(\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1}^+})(\boldsymbol{x_{k-1}} - \boldsymbol{x_{k-1}^+})^\top]\boldsymbol{J_x} + \boldsymbol{J_w} E[\boldsymbol{w_{k-1}}\boldsymbol{w_{k-1}}^\top]\boldsymbol{J_w}^\top\end{aligned}$$

Recalling that $\boldsymbol{P_k^+} = E[(\boldsymbol{x_k} - \boldsymbol{x_k^+})(\boldsymbol{x_k} - \boldsymbol{x_k^+})^\top]$ and assuming that the process noise enters the system directly yields:

$$\boldsymbol{P_k^-} = \boldsymbol{J_x}\boldsymbol{P_{k-1}^+}\boldsymbol{J_x} + \boldsymbol{Q} \tag{B.1}$$

Prior to the posteriori estimate, let us first define the error in the state estimation as:

$$\begin{aligned}\boldsymbol{e_k} &= \boldsymbol{x_k} - \boldsymbol{x_k^+} \\ &= \boldsymbol{f}(\boldsymbol{x_{k-1}}, \boldsymbol{u_{k-1}}, \boldsymbol{w_{k-1}}) - \boldsymbol{x_k^-} - \boldsymbol{K_k}(\boldsymbol{h}(\boldsymbol{x_k}, \boldsymbol{v_k}) - \boldsymbol{h}(\boldsymbol{x_k^-}, \boldsymbol{v_k})) \\ &\approx \boldsymbol{f}(\boldsymbol{x_{k-1}}, \boldsymbol{u_{k-1}}, \boldsymbol{w_{k-1}}) - \boldsymbol{f}(\boldsymbol{x_{k-1}^+}, \boldsymbol{u_{k-1}}, \boldsymbol{w_{k-1}}) - \boldsymbol{K_k}(\boldsymbol{h}(\boldsymbol{x_k}, \boldsymbol{v_k}) - \boldsymbol{h}(\boldsymbol{x_k^-}, \boldsymbol{v_k})) \\ &\approx \boldsymbol{J_{f,x}}(\boldsymbol{x_{k-1}^+})\boldsymbol{e_{k-1}} + \boldsymbol{J_{f,w}}\boldsymbol{w_{k-1}} - \boldsymbol{K_k}(\boldsymbol{J_{h,x}}(\boldsymbol{x_k^-})\boldsymbol{e_k^-} + \boldsymbol{J_{h,v}}\boldsymbol{v_k}) \\ &\approx \boldsymbol{J_{f,x}}(\boldsymbol{x_{k-1}^+})\boldsymbol{e_{k-1}} + \boldsymbol{J_{f,w}}\boldsymbol{w_{k-1}} - \boldsymbol{K_k}\boldsymbol{J_{h,x}}(\boldsymbol{x_k^-})(\boldsymbol{J_{f,x}}(\boldsymbol{x_{k-1}^+})\boldsymbol{e_{k-1}} + \boldsymbol{J_{f,w}}\boldsymbol{w_{k-1}}) - \boldsymbol{K_k}\boldsymbol{J_{h,v}}\boldsymbol{v_k} \\ &\approx (\boldsymbol{I} - \boldsymbol{K_k}\boldsymbol{J_{h,x}}(\boldsymbol{x_k^-}))\boldsymbol{J_{f,x}}(\boldsymbol{x_{k-1}^+})\boldsymbol{e_{k-1}} + (\boldsymbol{I} - \boldsymbol{K_k}\boldsymbol{J_{h,x}}(\boldsymbol{x_k^-}))\boldsymbol{J_{f,w}}\boldsymbol{w_{k-1}} - \boldsymbol{K_k}\boldsymbol{J_{h,v}}\boldsymbol{v_k}\end{aligned}$$

Now, the posteriori estimate of the covariance matrix can be found by:

$$\begin{aligned}\boldsymbol{P_k^+} &= E[\boldsymbol{e_k}\boldsymbol{e_k}^\top] \\ &= \left(\boldsymbol{I} - \boldsymbol{K_k}\boldsymbol{J_{h,x}}(\boldsymbol{x_k^-})\right)\boldsymbol{J_{f,x}}(\boldsymbol{x_{k-1}^+})\boldsymbol{P_{k-1}^+}\boldsymbol{J_{f,x}}(\boldsymbol{x_{k-1}^+})^\top \left(\boldsymbol{I} - \boldsymbol{K_k}\boldsymbol{J_{h,x}}(\boldsymbol{x_k^-})\right)^\top \\ &+ \left(\boldsymbol{I} - \boldsymbol{K_k}\boldsymbol{J_{h,x}}(\boldsymbol{x_k^-})\right)\boldsymbol{Q}\left(\boldsymbol{I} - \boldsymbol{K_k}\boldsymbol{J_{h,x}}(\boldsymbol{x_k^-})\right)^\top + \boldsymbol{K_k}\boldsymbol{R}\boldsymbol{K_k}^\top \\ &= \left(\boldsymbol{I} - \boldsymbol{K_k}\boldsymbol{J_{h,x}}(\boldsymbol{x_k^-})\right)\boldsymbol{P_k^-}\left(\boldsymbol{I} - \boldsymbol{K_k}\boldsymbol{J_{h,x}}(\boldsymbol{x_k^-})\right)^\top + \boldsymbol{K_k}\boldsymbol{R}\boldsymbol{K_k}^\top\end{aligned} \tag{B.2}$$

Now, we like to find the Kalman gain that minimizes the $tr(\boldsymbol{P_k^+})$. This is found by setting the differential of $tr(\boldsymbol{P_k^+})$ with respect to K_k to 0:

$$\frac{\partial tr(\boldsymbol{P_k^+})}{\partial \boldsymbol{K_k}} = 0$$

$$0 = -\left(\boldsymbol{J_{h,x}(x_k^-)P_k^-}\right)^\top - \boldsymbol{P_k^- J_{h,x}(x_k^-)^\top} - \boldsymbol{2K_k J_{h,x}(x_k^-)P_k^- J_{h,x}(x_k^-)^\top} + \boldsymbol{2K_k R}$$

$$\boldsymbol{K_k} = \boldsymbol{P_k^- J_{h,x}(x_k^-)^\top}\left(\boldsymbol{J_{h,x}(x_k^-)P_k^- J_{h,x}(x_k^-) + R}\right)^{\boldsymbol{-1}} \tag{B.3}$$

Substituting equation (B.3) back into equation (B.2) yields:

$$\begin{aligned}
\boldsymbol{P_k^+} &= \left(\boldsymbol{I - K_k J_{h,x}(x_k^-)}\right)\boldsymbol{P_k^-} - \left(\boldsymbol{I - K_k J_{h,x}(x_k^-)}\right)\boldsymbol{P_k^- J_{h,x}(x_k^-)^\top K_k^\top} + \boldsymbol{K_k R K_k^\top} \\
&= \left(\boldsymbol{I - K_k J_{h,x}(x_k^-)}\right)\boldsymbol{P_k^-} - \left(\boldsymbol{P_k^- J_{h,x}(x_k^-)^\top - K_k J_{h,x}(x_k^-)P_k^- J_{h,x}(x_k^-)^\top - K_k R}\right)\boldsymbol{K_k^\top} \\
&= \left(\boldsymbol{I - K_k J_{h,x}(x_k^-)}\right)\boldsymbol{P_k^-} - \left(\boldsymbol{P_k^- J_{h,x}(x_k^-)^\top - P_k^- J_{h,x}(x_k^-)^\top}\right)\boldsymbol{K_k^\top} \\
\boldsymbol{P_k^+} &= \left(\boldsymbol{I - K_k J_{h,x}(x_k^-)}\right)\boldsymbol{P_k^-}
\end{aligned}$$

Appendix C

Derivation of System Equations in Matrix Form

We start by rewriting equation (2.8) in dimensionless form:

$$\frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} = q \cdot \dot{\tilde{C}} \tag{C.1}$$

where q are:

$$q = \frac{C_0 \rho_s \rho_{solid} \cdot (\rho_s - \rho_{solid})}{(\tilde{C} C_0 \cdot \cdot (\rho_s - \rho_{solid}) + \rho_{solid})^2} \tag{C.2}$$

Substituting equation (2.39) in to equation (C.1) and we get:

$$\frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} = q \cdot \left(\alpha \hat{\mathbf{\Gamma}} - \alpha \boldsymbol{K}(\boldsymbol{\Phi\epsilon}) \cdot \boldsymbol{\epsilon} \right) \tag{C.3a}$$

$$\frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} = \underbrace{-\frac{q}{L_n} \cdot \epsilon_n^N}_{=w} \cdot \frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} \underbrace{-q \cdot \frac{k_m M_s}{\tilde{\rho_f} \rho_0 R \tilde{L}_n L_0 C_0} \cdot \left(\frac{P_s}{T_n T_0} - \frac{P_a}{T_a} \right) - q\alpha \boldsymbol{K_{e,N}} \cdot \begin{bmatrix} \epsilon_n^{N-2} \\ \epsilon_n^{N-1} \\ \epsilon_n^N \end{bmatrix}}_{=v} \tag{C.3b}$$

$$\frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} = w \cdot \frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} + v \tag{C.3c}$$

where $\boldsymbol{K_{e,N}}$ refers to the last row of the local stiffness matrix at the boundary element. The superscription $(\cdot)^N$ refers to the node at the boundary at any specific time step. For the thickness equations we have:

$$\frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} = \underbrace{\frac{k_m M_s}{\tilde{\rho_f} \rho_0 R L_0} \cdot \left(\frac{P_s}{\tilde{T} T_0} - \frac{P_a}{T_a} \right)}_{=-h} - \frac{\tilde{L}}{\tilde{\rho_f}} \frac{d\rho_f}{dt} \tag{C.4}$$

For the temperature equation we got:

$$\frac{\mathrm{d}\tilde{T}}{\mathrm{d}\tilde{t}} = -\frac{\tilde{T}}{\tilde{L}} \frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} - \frac{\tilde{T}}{\tilde{\rho_f}} \frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} \underbrace{-\frac{\mathrm{k_m M_s t_s} h_{fg}}{\rho_f \mathrm{C_p R} \tilde{L} \mathrm{L_0}} \cdot \left(\frac{P_s}{\tilde{T} {\mathrm{T_0}}^2} - \frac{\mathrm{P_a}}{\mathrm{T_a T_0}} \right)}_{=-g} + \underbrace{\frac{\mathrm{t_s}}{\rho_f \mathrm{C_p} \tilde{L} \mathrm{L_0 T_0}} \dot{q}_{in}}_{=\dot{q}} \tag{C.5}$$

Together they form the matrix equation:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} \\ \frac{\mathrm{d}\tilde{T}}{\mathrm{d}\tilde{t}} \\ \frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{\tilde{L}}{\tilde{\rho_f}} \\ -\frac{\tilde{T}}{\tilde{L}} & 0 & -\frac{\tilde{T}}{\tilde{\rho_f}} \\ w & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} \\ \frac{\mathrm{d}\tilde{T}}{\mathrm{d}\tilde{t}} \\ \frac{\mathrm{d}\tilde{\rho_f}}{\mathrm{d}\tilde{t}} \end{bmatrix} + \begin{bmatrix} -h \\ \dot{q} - g \\ v \end{bmatrix} \tag{C.6}$$

Appendix D

Matlab-Scripts

D.1 Main-Script

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%% Electrode Drying Model w/FEM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;

%-------------------------------------------------------------------------%
%------------------------Declaring Variables------------------------------%
%-------------------------------------------------------------------------%

p.rho_s = 838; %@(T) -0.8833*T+1292;              % Density of solvent [kg/m ^3]
p.rho_p = 1200;              % Density of the solids [kg/m^3] 4000 without air
p.Cp = 1900;                % Specific heat coefficient [J/ kgK ]
p.Ms = 99.133;                  % Molecular weight of solvent[kg/ kmol ]
p.R = 8314;                 % Universal gas constant [J/kmolK ]
p.Ta = @(t) 350+5*(t>1200);% + 20*(t>3000)); % Ambient air temperature [K] (80-130)C
p.rh = 0.25;                 % Relativ humidity of ambient air [%]
p.Pa = @(Ta) p.rh*133.332*10.^(7.54826-1979.68./(222.2+(Ta-273) )); % Vapor pressure in the ambient
    air [Pa]
p.V = @(t) 15+2*(t>1210);                     % Air velocity [m/s] (4-25)
%[p.k_c , p.k_m ] = TransferCoefficients(V); % Heat and mass transfer coefficients [W/m^2K] [m/s]

%% Initial Values
p.L0 = 0.165e-03;         % Initial thin film thickness (75-240)mum
p.T0 = 330.5;             % Initial thin film temperature
p.D0 = 9e-9;              % Initial diffusion coefficient
p.C0 = 0.5;               % Initial solvent concentration .5455 or .3243 (solvent mass/total mass)
[p.rho0 ,p.V_s,p.V_so] = GetDensity(p); % Intial density, solvent volume and solid volume

lambda = 30e-03; %
rho_a = 1.009; % Denisty of the air
mu_a = 20.74e-06; % Viscocity of the iar
Cp = 1000; % Specific heat of the air
D = 0.282e-06; % Diffusion coefficient
LL = 1; % Charactheristic length
Pr = Cp* mu_a / lambda ; % Prandtl number
Sc = mu_a /( rho_a *D); % Schmidt number
Le = Sc/Pr; % Lewis number
p.k_c = @(V) 0.037*V.^(0.8).*( mu_a / rho_a )^( -0.8)*Pr^(1/3)*lambda*LL^( -0.2) ;
p.k_m = @(k_c) k_c /( rho_a *Cp)*Le ^( -2/3) ;

%-------------------------------------------------------------------------%
%------------------------Finite Element Method----------------------------%
%-------------------------------------------------------------------------%

%% Number of elements and matrix assembling
noe = 10;                           % Number of elemetns
non = 2*noe+1;                      % Number of nodes (assuming quadratic elements)
p.z = linspace(0,1, non);           % Spacestep (assuming uniform element size)
[M,K] = AssembleMatriceQuad(p.z);   % Assembling the dampning and stiffness matrices
```

```
x0 = ones(1,length(p.z)+3);         % Initial values vector, x0(1:2) initial thickness and
    temperatur
x0(1:3) = [0.516693157188501 1.059001512161357 1.115504352140297
]; x0(4:end) = 0.184988256905711;
SimTime = 150*60;                   % Simulation time in minutes
p.dt = 1;                           % Timestep
t = 0:p.dt:SimTime;                 % Time vector

%% Solving the system
tic
% [y, p.rho, ConD] = SolveSystem(t,x0,p,M,K);          % Solving the system
[y D] = solveDryingSystem(p,t,x0,M,K);
toc
L = y(1,:)*p.L0;                        % Solution vector for thickness
T = y(2,:)*p.T0;                        % Solution vector for temperature
rho = y(3,:)*p.rho0;
C = y(4:end,:)*p.C0;                    % Solution matrix for solvent concentration


%-------------------------------------------------------------------------%
%--------------------------- Linear System -------------------------------%
%-------------------------------------------------------------------------%
v.T = p.Ta(0)/p.T0;
v.C = C(end)/p.C0;
v.rho = rho(end)/p.rho0;
v.L = L(end)/p.L0;
v.Ta = p.Ta(0);
v.V = p.V(0);

y0 = zeros(size(y,1),1);
% [L T C]
y0(1:3) = [1-v.L 1-v.T 1-v.rho]; y0(4:end) = 1-v.C;
% y0 = 0;
u0 = [0 0]';
tic
[y_l,sys_c,sys] = Linearization(p,v,M,K,y0,u0,t);
toc
L_l = (y_l(1,:)+v.L)*p.L0;
T_l = (y_l(2,:)+v.T)*p.T0;
rho_l = (y_l(3,:)+v.rho)*p.rho0;
C_l = (y_l(4:end,:)+v.C)*p.C0;

%% Linear Simulation - Small Deviation From OP
u = zeros(2,length(t));
y0_OP = [0.05 -0.05 -0.05 ones(1,length(p.z))*0.25];
y_no = lsim(sys,u,t, y0_OP);
%% Linear Simulations - No Excitation
y_ne = lsim(sys,u,t,'zoh');
%% Linear Simulations - With Excitation
t_hta = 20; step_Ta = 5;                                        % Steptime in minutes and
     stepsize for Ta
t_hv = 25; step_V = 2;                                          % Steptime in minutes and
     stepsize for V
t_hta = (t_hta*60)/p.dt;
t_hv = (t_hv*60)/p.dt;

t_ramp = 10;                                                     % Duration of the ramp
t_ramp = (t_ramp*60)/p.dt;                                      % Converting the duration
     of the ramp to fit t-vector
for i = 1:t_ramp                                                % Constructing ramp
    functions for the steps in u
   u(1,t_hta+1+i) = step_Ta/t_ramp*i;
   u(2,t_hv+1+i) = step_V/t_ramp*i;                             % Linear increas in
    control input
end
u(1,(t_hta+1+t_ramp):end) = step_Ta;                            % Setting the step in the
     control input for the rest of the simulation
u(2,(t_hv+1+t_ramp):end) = step_V;
y_e = lsim(sys,u,t);

%% Linear Vs Non-Linear
t_hh = 1201/p.dt; t_vh = 1211/p.dt;                             % Steptime of the non-
    linear model
```

```
u_h = zeros(2,length(t));
u_h(1,t_hh:end) = step_Ta; u_h(2,t_vh:end) = step_V;
y_lvn = lsim(sys,u_h,t,'zoh');
T_lvn = T/p.T0-v.T; L_lvn = L/p.L0-v.L;
%--------------------------------------------------------------------------%
%----------------------------Post Processing-------------------------------%
%--------------------------------------------------------------------------%

% T = correctTemperature(p,T);                                              % Correction for small overshoot in temperature
Ps = p_sol(C(end,:),T,p);                                                   % Vector for solvent partial pressure at thin film - air interface
Ps_l = p_sol(C_l(end,:),T_l,p);                                             % Vector for solvent partial pressure for linear model
Ta = p.Ta(t(:))'; Pa = p.Pa(p.Ta(t(:)))';
Evap_rate = (p.k_m(p.k_c(p.V(t(:))))'.*p.Ms)./p.R.*(Ps./(T)-Pa./Ta);        % Vector for evaporation [kg/m^2]
E_l = (p.k_m(p.k_c(p.V(t(:))))'.*p.Ms)./p.R.*(Ps_l./(T_l)-Pa./Ta);          % Vector for evaporation for linear model
% m_error_p = 100*MassConservationCheck(t,C(end),p,L(end),Evap_rate);   % Error in mass [%]
Q = HeatConsumed(p,T,Evap_rate);                                            % Total heat needed [kJ/m^2]


for i=2:length(t)                                                           % Assessing drying time based on evaporation criterias
    check = abs(Evap_rate(i)-Evap_rate(i-1));
    E_constraint = 1e-07;                                                   % Constraint for the evaporation rate
    dE_constraint = 1e-09;                                                  % Constraint for the change in evaporation rate

    if abs(Evap_rate(i)) <= E_constraint && check <= dE_constraint
        drying_time = i*p.dt/60                                             % Drying time in minutes
        C(end,i)
        break
    end
end

error = (1-((p.L0*p.rho0-L(end)*rho(end))/trapz(p.dt,Evap_rate)))*100
%--------------------------------------------------------------------------%
%-------------------------------Plots--------------------------------------%
%--------------------------------------------------------------------------%
t = t/60;
figure
%% Thickness Plot
subplot (221)
plot (t,L*10^6, 'b')
grid
xlabel('Time (min)','interpreter','latex ')
ylabel (' Film Thickness ($\mathrm{\mu m}$)', 'interpreter','latex ')

%% Temperature Plot
subplot (222)
plot (t,T,'b-')
hold on
plot([0 t(end)],[p.Ta(0) p.Ta(0)],'r--')
hold off
grid
xlabel (' Time (min)', 'Interpreter','latex ')
ylabel (' Temperature (K)', 'Interpreter','latex ')
legend('Thin Film Temperature','Air Temperature')
xlim([0 2]); ylim([p.T0 p.Ta(0)+5])
%% Evaporation Plot
subplot (224)
plot (t,rho, 'b')
grid
xlabel(' Time (min)', 'Interpreter','latex ')
ylabel(' Density ($\mathrm{\frac{kg}{m^3}}$)', 'Interpreter','latex ')

%% Solvent Fraction Plot
subplot (223)
```

```
plot (t,C(end,:),'b',t,C(1,:),'r--')
grid
xlabel(' Time (min)', 'Interpreter','latex ')
ylabel(' Solvent fraction', 'Interpreter','latex ')
legend('C_{top layer}','C_{bottom layer}')
%-------------------------------------------------------------------------%
%------------------------- Linear Plots ----------------------------------%
%-------------------------------------------------------------------------%
figure
%% Thickness Plot
subplot (221)
plot (t,L_l*10^6,'b', t, L*10^6,'r--')
grid
xlabel('Time (min)','interpreter','latex ')
ylabel (' Film Thickness ($\mathrm{\mu m}$)', 'interpreter','latex ')
legend('Linear Model', 'Non-Linear Model')

%% Temperature Plot
subplot (222)
plot (t,T_l,'b-',t,T,'r--')
grid
xlabel (' Time (min)', 'Interpreter','latex ')
ylabel (' Temperature (K)', 'Interpreter','latex ')
legend('Linear Model', 'Non-Linear Model')
xlim([0 2])
%% Evaporation Plot
subplot (224)
plot (t,E_l,'b',t, Evap_rate,'r--')
grid
xlabel(' Time (min)', 'Interpreter','latex ')
ylabel(' Drying rate ($\mathrm{\frac{kg}{m^2s}}$)', 'Interpreter','latex ')
legend('Linear Model', 'Non-Linear Model')
%% Solvent Fraction Plot
subplot (223)
plot (t,C_l(end,:),'b',t,C(end,:),'r--')
grid
xlabel(' Time (min)', 'Interpreter','latex ')
ylabel(' Solvent fraction', 'Interpreter','latex ')
legend('Linear Model', 'Non-Linear Model')

%-------------------------------------------------------------------------%
%---------------------- Linear Simulation Plots --------------------------%
%-------------------------------------------------------------------------%
figure
%% Thickness Plot
subplot (211)
yyaxis left
plot (t,y_e(:,1)'*p.L0*10^6,'b', t, y_ne(:,1)'*p.L0*10^6,'r')
ylabel ('Deviation from OP [$\mathrm{\mu m}$]', 'interpreter','latex ')
yyaxis right
h = plot(t,u(1,:),'b:',t,u(2,:),'r:');
set(h,{'LineWidth'},{2})
grid
xlabel('Time (min)','interpreter','latex ')
ylabel('Stepsize')
legend('W/ Excitation', 'Operating Point', 'u_1','u_2')

%% Temperature Plot
subplot (212)
yyaxis left
plot (t,y_e(:,2)'*p.T0,'b', t,y_ne(:,2)'*p.T0,'r');
ylabel ('Deviation from OP [K]', 'Interpreter','latex ')
yyaxis right
h = plot(t,u(1,:),'b:',t,u(2,:),'r:');
set(h,{'LineWidth'},{2})
grid
xlabel (' Time (min)', 'Interpreter','latex ')
ylabel('Stepsize')
legend('W/ Excitation', 'Operating Point', 'u_1','u_2')

figure
%% Thickness Plot
subplot (211)
```

```
yyaxis left
plot (t,L_lvn,'b', t, y_lvn(:,1)','r')
ylabel ('Deviation from OP', 'interpreter','latex ')
yyaxis right
plot(t,u_h(1,:),':',t,u_h(2,:),':');
grid
xlabel('Time (min)','interpreter','latex ')
ylabel('Stepsize')
legend('Non-Linear Model', 'Linear Model', 'u_1','u_2')

%% Temperature Plot
subplot (212)
yyaxis left
plot (t,T_lvn,'b', t,y_lvn(:,2)','r');
ylabel ('Deviation from OP', 'Interpreter','latex ')
yyaxis right
plot(t,u_h(1,:),':',t,u_h(2,:),':')
grid
xlabel (' Time (min)', 'Interpreter','latex ')
ylabel('Stepsize')
legend('Non-Linear Model', 'Linear Model', 'u_1','u_2')
```

Listing D.1: Main Matlab script.

D.2 Linearization Implmentation

```
function [a,sys,sys_d] = Linearization(p,v,M,K,y0,u0,t)
    %-------------------------------------------------------------------------%
    %----------------Linearization of the Non-Linear Model--------------------%
    %---------------------Using the Symbolic Toolbo---------------------------%
    %-------------------------------------------------------------------------%

    %% Declaring the Symbolic Variables
    c = sym('c', [length(p.z) 1], 'real');
    syms L T rho Ta V real;

    %% Declaring the Parameters in Terms of the Symbolic Variables
    Ps = p_sol(c(end)*p.C0,T*p.T0,p);
    D = D_Coefficient(c*p.C0,T*p.T0,p);
    h_fg = 6.991*(T*p.T0)^2-6193*T*p.T0+1.848e06;
    Pa = p.rh*133.332*10.^(7.54826-1979.68./(222.2+(Ta-273)));
    [k_c , k_m ] = TransferCoefficients(V);
    qin = k_c*(Ta-T*p.T0);


    %% Determining the Jacobian for the Density, Thickness and Temperature Based on the Startionary
    Conditions for the Given Problem
    Q = D.*K; b = p.D0/((L*p.L0)^2);
    q = (p.rho_s*p.rho_p*(p.rho_p-p.rho_s)*p.C0)/(p.rho0*(c(end)*p.C0*(p.rho_s-p.rho_p)+p.rho_p)^2);
    B = [-(k_m*p.Ms)/(rho*p.rho0*p.R*p.L0)*(Ps/(T*p.T0)-Pa/Ta);
        -(k_m*p.Ms*h_fg)/(rho*p.rho0*p.Cp*p.R*L*p.L0*p.T0)*(Ps/(T*p.T0)-Pa/Ta)+qin/(rho*p.rho0*p.Cp*
    p.T0*L*p.L0);
        -q*(k_m*p.Ms)/(p.R*p.rho0*L*p.L0)*(Ps/(T*p.T0)-Pa/Ta)-q*b*Q(end,end-2:end)*c(end-2:end)];
    A = [1 0 L/rho; T/L 1 T/rho; q*rho/L 0 1]; dXdt = A\B;
    Y = [L T rho c' Ta V]; W = [v.L v.T v.rho ones(1,length(p.z))*v.C v.Ta v.V];
    J = double(subs(jacobian(dXdt,Y),Y,W));

    %% Determining the Jacobian for the Solvent Fraction Based on the Startionary Conditions for the
     Given Problem
    b = p.D0/((L*p.L0)^2); R = Q; R(:,:) = 0; B = c; B(:) = 0;
    R(end) = dXdt(1)/L; K_star = -(b*Q+R);
    B(end) = -(k_m*p.Ms)/(rho*p.rho0*p.R*L*p.L0*p.C0)*(Ps/(T*p.T0)-Pa/Ta);
    dCdt = M\K_star*c+M\B;
    J_C = double(subs(jacobian(dCdt,Y),Y,W));

    %% Defining the Matrices for the Linear State-Space Realization
    A = [J(:,1:end-2); J_C(:,1:end-2)];
    B = [J(:,end-1:end); J_C(:,end-1:end)];
    C = zeros(2,length(p.z)+3); C(1,1) = 1; C(2,2) = 1;
%     C = diag(ones(1,length(p.z)+2));                        % If every
%     state is measured

```

```
    %% Converting the System Into a Continuous and Discrete State-Space Realization
    sys = ss(A,B,C,0);
    sys_d = c2d(sys,p.dt);

    %% Solving the Linear System
    Ad = sys_d.A; Bd = sys_d.B;
    a = zeros(length(p.z)+3,length(t));
    a(:,1) = y0;
    u = u0;
    for i = 1:length(t)-1
       a(:,i+1) = Ad*a(:,i)+Bd*u;
    end

    %% Determining the Controllability and Observability of the Linear System
    OB = obsv(Ad,C); CB = ctrb(Ad,Bd);
    OX = ['Number of unobservable states: ', num2str(length(A)-rank(OB))];
    CX = ['Number of uncontrollable states: ',num2str(length(A)-rank(CB))];
    disp(OX)
    disp(CX)

end
```

Listing D.2: Linearization script.

D.3 Optimization

```
clear;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%% Electrode Drying Model w/FEM %%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%% Optimization %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------------------------------------------------------------------%
%-------------------------Declaring Variables-----------------------------%
%-------------------------------------------------------------------------%

p.rho_s = 838;                                                  % Density of solvent [kg/m ^3]
p.rho_p = 1200;                                                 % Density of the solids [kg/m^3]
    4000 without air
p.Cp = 1900;                                                    % Specific heat coefficient [J/ kgK
    ]
p.Ms = 99.133;                                                  % Molecular weight of solvent[kg/
    kmol ]
p.R = 8314;                                                     % Universal gas constant [J/kmolK ]
p.Ta = 350;                                                     % Ambient air temperature [K]
    (80-130)C
p.rh = 0.25;                                                    % Relativ humidity of ambient air
    [%]
p.Pa = p.rh*133.332*10.^(7.54826-1979.68./(222.2+(p.Ta-273)));  % Vapor pressure in the ambient air
    [Pa]
p.V = 15;                                                       % Air velocity [m/s] (4-25)
[p.k_c , p.k_m ] = TransferCoefficients(p.V);                   % Heat and mass transfer
    coefficients [W/m^2K] [m/s]

%% Initial Values
p.L0 = 0.165e-03;                               % Initial thin film thickness (75-240)mum
p.T0 = 330.5;                                   % Initial thin film temperature
p.D0 = 9e-9;                                    % Initial diffusion coefficient
p.C0 = 0.5;                                     % Initial solvent concentration .5455 or .3243 (
    solvent mass/total mass)
[p.rho0 ,p.V_s,p.V_so] = GetDensity(p);         % Intial density, solvent volume and solid volume

%% Number of elements and matrix assembling
noe = 2;                                        % Number of elemetns
non = 2*noe+1;                                  % Number of nodes (assuming quadratic elements)
p.z = linspace(0,1, non);                       % Spacestep (assuming uniform element size)
[p.M,p.K] = AssembleMatriceQuad(p.z);           % Assembling the dampning and stiffness matrices
x0 = ones(1,length(p.z)+3);                     % Initial values vector, x0(1:2) initial thickness
    and temperatur
SimTime = 80*60;                                % Simulation time in minutes
p.dt = 8;                                       % Timestep
p.t = 0:p.dt:SimTime;                           % Time vector
p.t_1 = 6; p.t_2 = 20;
```

```
%-------------------------------------------------------------------------%
%----------------------------Optimization---------------------------------%
%-------------------------------------------------------------------------%

p.Noi = 1; p.Nos = length(p.z)+3;               % Length of control input vector and state vector
p.N = length(p.t);                              % Number of discrete points
u = zeros(3,1); u0 = u; u0(:) = p.Ta;           % Declaration of u vector
x = ones(p.Nos,1);                              % Declaration of initial z vector

%% Bounds
ul = p.T0; uu = 370;
ul = repmat(ul,3,1); uu = repmat(uu,3,1);
vl = ul; vu = uu;                               % Upper and lower bounds on optimization variable
p.C_limit = 0.2;                                % Upper bounds for final solvent concentration
p.E_limit = 5e-05;                              % Upper bounds on evaporation rate

%% Optimization
disp('Optimizing!')
tic
options = optimoptions('fmincon','Algorithm','sqp','MaxIterations',3000,'ConstraintTolerance',1e-06)
    ;
[u,fval] = fmincon(@(u)OFun(u,x,p),u0,[],[],[],[],vl,vu,@(u)nlCon(u,x,p),options);
toc

%% Solving the Non-Linear System, With/Without Optimal Control Inputs
u_0 = p.Ta*ones(length(p.t),1);
u_o = zeros(length(p.t),1); u_o(1:p.t_1*60/p.dt) = u(1); u_o(p.t_1*60/p.dt+1:p.t_2*60/p.dt) = u(2);
    u_o(p.t_2*60/p.dt+1:end) = u(3);

y_0 = solveSystemOP(u_0,x,p); y_o = solveSystemOP(u_o,x,p);


%% Comparison of Original and Optimal setup
Ps_0 = p_sol(y_0(end,:).*p.C0,y_0(2,:).*p.T0,p); Ps_o = p_sol(y_o(end,:).*p.C0,y_o(2,:).*p.T0,p);
Pa_0 = p.rh*133.332*10.^(7.54826-1979.68./(222.2+(u_0-273)))'; Pa_o = p.rh
    *133.332*10.^(7.54826-1979.68./(222.2+(u_o-273)))';
E_0 = (p.k_m*p.Ms)./p.R.*(Ps_0./(y_0(2,:).*p.T0)-Pa_0./u_0'); E_o = (p.k_m*p.Ms)./p.R.*(Ps_o./(y_o
    (2,:).*p.T0)-Pa_o./u_o');
Q_0 = HeatConsumedOP(p,y_0(2,:)*p.T0,E_0,u_0'); Q_o = HeatConsumedOP(p,y_o(2,:)*p.T0,E_o,u_o');

Q_reduction = (1-Q_o/Q_0)*100

t = p.t/60;
%% Plots
figure
subplot(221)
plot(t,y_0(1,:)*p.L0*10^6, t, y_o(1,:)*p.L0*10^6,'r--')
legend('Original Setup','Optimized Setup')
xlabel('Time (min)','interpreter','latex ')
ylabel (' Film Thickness ($\mathrm{\mu m}$)', 'interpreter','latex ')
subplot(222)
plot(t,y_0(2,:)*p.T0, t, y_o(2,:)*p.T0,'r--')
legend('Original Setup','Optimized Setup')
xlabel('Time (min)','interpreter','latex ')
ylabel (' Film Temperature (K)', 'interpreter','latex ')
subplot(223)
plot(t,y_0(end,:)*p.C0, t, y_o(end,:)*p.C0,'r--')
legend('Original Setup','Optimized Setup')
xlabel('Time (min)','interpreter','latex ')
ylabel (' Solvent Fraction', 'interpreter','latex ')
subplot(224)
plot(t,E_0, t, E_o,'r--')
legend('Original Setup','Optimized Setup')
xlabel('Time (min)','interpreter','latex ')
ylabel (' Evaporation Rate ($\mathrm{\frac{kg}{s}}$)', 'interpreter','latex ')
```

Listing D.3: Optimization script.

```
function ObjFun = OFun(u_i,z,p)
    ObjFun = 0; x = z; a = x;
    uu = zeros(length(p.t),1); uu(1:(p.t_1*60/p.dt)) = u_i(1); uu((p.t_1*60/p.dt)+1:(p.t_2*60/p.dt))
     = u_i(2); uu((p.t_2*60/p.dt)+1:end) = u_i(3);
    uv = zeros(length(p.t),1); uv(1:(p.t_1*60/p.dt)) = u_i(4); uv((p.t_1*60/p.dt)+1:(p.t_2*60/p.dt))
     = u_i(5); uv((p.t_2*60/p.dt)+1:end) = u_i(6);
    [k_c,k_m] = TransferCoefficients(uv);
    for i = 1:length(p.t)-1
        u = uu(i);
        %% The Runge-Kutta Method for Solving the ODEs
        k1 = RungeKuttaCon(p,x(1:3),x,u,k_c(i),k_m(i));
        k2 = RungeKuttaCon(p,x(1:3)+p.dt*k1/2,x,u,k_c(i),k_m(i));
        k3 = RungeKuttaCon(p,x(1:3)+p.dt*k2/2,x,u,k_c(i),k_m(i));
        k4 = RungeKuttaCon(p,x(1:3)+p.dt*k3,x,u,k_c(i),k_m(i));

        a(1:3) = x(1:3)+1/6*p.dt*(k1+2*k2+2*k3+k4);

        %% The Simplified Newton-Raphson's Method for Solving the Finite Element Approximation
        j = 1; c_next = x(4:end); dc = 1; b = p.D0/((x(1)*p.L0)^2); max_iter = 20;
        R = zeros(size(p.M)); F = zeros(length(p.z),1);
        Pa =  p.rh*133.332*10^(7.54826-1979.68/(222.2+(u-273)));
        while max(dc) > eps && max_iter > j
            D = D_Coefficient(c_next.*p.C0,x(2)*p.T0,p); Q = D.*p.K;
            Ps = p_sol(c_next(end)*p.C0, x(2)*p.T0,p);
            R(end) = k1(1)/x(1);K_star = -(b*Q+R); F(end) = -(k_m(i)*p.Ms)/(x(3)*p.rho0*p.R*x(1)*p.
    L0*p.C0)*(Ps/(x(2)*p.T0)-Pa/u);
            A = -(p.M-p.dt*K_star); B = -p.M*x(4:end)-p.dt*F;
            c_prev = c_next; c_next = A\B; dc = norm(c_prev-c_next); j = j+1;
        end
        a(4:end) = c_next;
        h_fg = 6.991*(x(2)*p.T0)^2-6193*(x(2)*p.T0)+1.985e06;
        ObjFun = [ObjFun k_c(i)*(u-x(2)*p.T0)+((h_fg*k_m(i)*p.Ms)/p.R)*(Ps/(p.T0*x(2))-Pa/u)];
        x = a;
    end
    ObjFun = trapz(p.dt,ObjFun);
end
```

Listing D.4: Implementation of objective function.

```
function [c, ceq] = nlCon(u_i,z,p)
    x = z; a = x; check = true; ceq = [];
    uu = zeros(length(p.t),1); uu(1:(p.t_1*60/p.dt)) = u_i(1); uu((p.t_1*60/p.dt)+1:(p.t_2*60/p.dt))
     = u_i(2); uu((p.t_2*60/p.dt)+1:end) = u_i(3);
    uv = zeros(length(p.t),1); uv(1:(p.t_1*60/p.dt)) = u_i(4); uv((p.t_1*60/p.dt)+1:(p.t_2*60/p.dt))
     = u_i(5); uv((p.t_2*60/p.dt)+1:end) = u_i(6);
    [k_c,k_m] = TransferCoefficients(uv);
    for i = 1:length(p.t)-1
        u = uu(i);
        %% The Runge-Kutta Method for Solving the ODEs
        k1 = RungeKuttaCon(p,x(1:3),x,u,k_c(i),k_m(i));
        k2 = RungeKuttaCon(p,x(1:3)+p.dt*k1/2,x,u,k_c(i),k_m(i));
        k3 = RungeKuttaCon(p,x(1:3)+p.dt*k2/2,x,u,k_c(i),k_m(i));
        k4 = RungeKuttaCon(p,x(1:3)+p.dt*k3,x,u,k_c(i),k_m(i));

        a(1:3) = x(1:3)+1/6*p.dt*(k1+2*k2+2*k3+k4);

        %% The Simplified Newton-Raphson's Method for Solving the Finite Element Approximation
        j = 1; c_next = x(4:end); dc = 1; b = p.D0/((x(1)*p.L0)^2); max_iter = 20;
        R = zeros(size(p.M)); F = zeros(length(p.z),1);
        Pa =  p.rh*133.332*10^(7.54826-1979.68/(222.2+(u-273)));
        while max(dc) > eps && max_iter > j
            D = D_Coefficient(c_next.*p.C0,x(2)*p.T0,p); Q = D.*p.K;
            Ps = p_sol(c_next(end)*p.C0, x(2)*p.T0,p);
            R(end) = k1(1)/x(1);K_star = -(b*Q+R); F(end) = -(k_m(i)*p.Ms)/(x(3)*p.rho0*p.R*x(1)*p.
    L0*p.C0)*(Ps/(x(2)*p.T0)-Pa/u);
            A = -(p.M-p.dt*K_star); B = -p.M*x(4:end)-p.dt*F;
            c_prev = c_next; c_next = A\B; dc = norm(c_prev-c_next); j = j+1;
        end
        a(4:end) = c_next;
        if check
           c = (k_m(i)*p.Ms)/p.R*(Ps/(x(2)*p.T0)-Pa/u)-p.E_limit;
           check = false;
        else
```

```
            c = [c;  (k_m(i)*p.Ms)/p.R*(Ps/(x(2)*p.T0)-Pa/u)-p.E_limit];
        end
        x = a;
    end
    Pa =  p.rh*133.332*10^(7.54826-1979.68/(222.2+(u_i(3)-273)));
    Ps = p_sol(x(end)*p.C0, x(2)*p.T0,p);
    c = [x(end)-p.C_limit; c;  (k_m(end)*p.Ms)/p.R*(Ps/(x(2)*p.T0)-Pa/u)-p.E_limit];
end
```

Listing D.5: Implementation of non-linear constraint function.

D.4 Control

```
clear;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%% Electrode Drying Model w/FEM %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Control %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------------------------------------------------------------------%
%-------------------------Declaring Variables-----------------------------%
%-------------------------------------------------------------------------%

p.rho_s = 838;                                  % Density of solvent [kg/m ^3]
p.rho_p = 1200;                                 % Density of the solids [kg/m^3] 4000 without air
p.Cp = 1900;                                    % Specific heat coefficient [J/ kgK ]
p.Ms = 99.133;                                  % Molecular weight of solvent[kg/ kmol ]
p.R = 8314;                                     % Universal gas constant [J/kmolK ]
p.Ta = 350;                                     % Ambient air temperature [K]
p.rh = 0.25;                                    % Relativ humidity of ambient air [%]
p.V = 15;                                       % Air velocity [m/s]
[p.k_c , p.k_m ] = TransferCoefficients(p.V);
p.k = 0.024;

%% Initial Values
p.L0 = 0.165e-03;                       % Initial thin film thickness (75-240)mum
p.T0 = 330.5;                           % Initial thin film temperature
p.D0 = 9e-9;                            % Initial diffusion coefficient
p.C0 = 0.5;                             % Initial solvent concentration
[p.rho0 ,p.V_s,p.V_so] = GetDensity(p); % Intial density, solvent volume and solid volume


%% Number of elements and matrix assembling
noe = 2;                                    % Number of elemetns
non = 2*noe+1;                              % Number of nodes (assuming quadratic elements)
p.z = linspace(0,1, non);                   % Spacestep (assuming uniform element size)
[p.M,p.K] = AssembleMatriceQuad(p.z);       % Assembling the dampning and stiffness matrices
x0 = ones(1,length(p.z)+3);                 % Initial values vector, x0(1:2) initial thickness and
    temperatur
SimTime = 80*60;                            % Simulation time in minutes
p.dt = 1;
E_limit = 5e-5;
%% Optimal Control Inputs
uref1 = [350.2; 15]; uref2 = [363.9; 15]; uref3 = [330.5;15];
t_1 = 20; t_2 = 40;
%% Kalman Filter Parameters
H = zeros(2, length(p.z)+3); H(1,1) = 1; H(2,2) = 1;
[J_ff, p.J_h, Y] = Jacobians(p,p.M,p.K);
g = matlabFunction(J_ff, 'vars', {Y});
p.Q = 10^(-5)*diag(ones(1,length(p.z)+3)); p.RR = diag([10 2]); p.P_k = zeros(size(p.Q));
p.Q(1,1) = 0.1; p.Q(2,2) = 0.1; p.Q(3,3) = 100; p.P_k = diag(ones(1,length(p.z)+3));
%% Heat Exchanger
c = 0.7214;
pp = 1.009;
G = tf(1,[c*pp 0],'outputdelay',1);
Gd = tf(1,[c*pp 0],'outputdelay',1);
W_y = 20; W_u = 5; W_d = .25;
G = W_y^(-1)*G*W_u; Gd = W_y^(-1)*Gd*W_d;
K = tf([1.5 .01],[1 0]); [num, den] = tfdata(K);
Kp1 = num{1}(1); Ti1 = 1/num{1}(2);

%% Wind Fan
Vl = 15; Vin = 1; ql = (Vl^2-Vin^2)/2;
```

```
k = 1/(sqrt(2*ql+Vin^2));
Gv = tf(k,1,'inputdelay',1); Gvd = tf(1,1);
Wv_y = 5; Wv_u = 10; Wv_d = .25;
Gv = Wv_y^(-1)*Gv*Wv_u; Gvd = Wv_y^(-1)*Gvd*Wv_d;
Kv = tf([3 5],[1 0]); [numv,denv] = tfdata(Kv);
Kp2 = numv{1}(1); Ti2 = 1/numv{1}(2);
%% Overall Control Process
sys = [G Gd 0 0; 0 0 Gv Gvd];
sys = ss(sys);

% hws = get_param(bdroot,'modelworkspace');
% hws.assignin('p',p);

%% Simulation
sim('PIControl.slx');
x0 = x_hat1.Data(end,:);
sim('PIControl2.slx');
x0 = x_hat2.Data(end,:);
sim('PIControl3.slx');

%% Variables
L = [x_measure1.Data(:,1)' x_measure2.Data(:,1)' x_measure3.Data(:,1)']*p.L0;
Le = [x_hat1.Data(:,1)' x_hat2.Data(:,1)' x_hat3.Data(:,1)']*p.L0;
T = [x_measure1.Data(:,2)' x_measure2.Data(:,2)' x_measure3.Data(:,2)']*p.T0;
Te = [x_hat1.Data(:,2)' x_hat2.Data(:,2)' x_hat3.Data(:,2)']*p.T0;
rho = [x_hat1.Data(:,3)' x_hat2.Data(:,3)' x_hat3.Data(:,3)']*p.rho0;
C = [x_hat1.Data(:,4:end)' x_hat2.Data(:,4:end)' x_hat3.Data(:,4:end)']*p.C0;
t = [x_hat1.Time' (x_hat2.Time')+x_hat1.Time(end) (x_hat3.Time')+(x_hat1.Time(end)+x_hat2.Time(end))
    ]/60;
uT = [u_a1.Data(:,1)' u_a2.Data(:,1)' u_a3.Data(:,1)'];
uV = [u_a1.Data(:,2)' u_a2.Data(:,2)' u_a3.Data(:,2)'];
p.t = t;
%% Evaporation Rate
Ps = p_sol(C(end,:),Te,p);
Pa = p.rh*133.332*10.^(7.54826-1979.68./(222.2+(uT-273)));
E = (p.k_m.*p.Ms)./p.R.*(Ps./(Te)-Pa./uT);

%% Optimized Trajectory
u_o = zeros(length(t),1); u_o(1:t_1*60/p.dt) = uref1(1); u_o(t_1*60/p.dt+1:t_2*60/p.dt) = uref2(1);
    u_o(t_2*60/p.dt+1:end) = uref3(1);
v_o = zeros(length(t),1); v_o(1:t_1*60/p.dt) = uref1(2); v_o(t_1*60/p.dt+1:t_2*60/p.dt) = uref2(2);
    v_o(t_2*60/p.dt+1:end) = uref3(2);
x0 = ones(1,length(p.z)+3);
y_o = solveSystemOP(u_o,v_o,x0',p);

L_o = y_o(1,:)*p.L0;
T_o = y_o(2,:)*p.T0;
C_o = y_o(4:end,:)*p.C0;
[k_c,k_m] = TransferCoefficients(v_o);

Ps = p_sol(C_o(end,:),T_o,p);
Pa = p.rh*133.332*10.^(7.54826-1979.68./(222.2+(u_o'-273)));
E_o = (k_m'.*p.Ms)./p.R.*(Ps./(T_o)-Pa./u_o');
%% Plots
figure
subplot(221)
plot(t,Le*10^6,t,L_o*10^6,'r--')
ylabel('Thickness ($\mathrm{\mu m}$)','interpreter','latex')
xlabel('Time (min)','interpreter','latex')
legend('Controlled Process', 'Optimal Trajectory')
subplot(222)
plot(t,Te,t,T_o,'r--')
ylabel('Temperature (K)','interpreter','latex')
xlabel('Time (min)','interpreter','latex')
legend('Controlled Process', 'Optimal Trajectory')
subplot(223)
plot(t,C(end,:),t,C_o(end,:),'r--')
ylabel('Solvent Fraction (\%)','interpreter','latex')
xlabel('Time (min)','interpreter','latex')
legend('Controlled Process', 'Optimal Trajectory')
subplot(224)
plot(t,E,t,E_o,'r--')
hold on
```

```
plot([0 t(end)],[E_limit E_limit],'--')
ylabel('Evaporation Rate ($\mathrm{\frac{kg}{s}}$)','interpreter','latex')
xlabel('Time (min)','interpreter','latex')
legend('Evaporation Rate','Optimal Trajectory','Constraint')
figure
subplot(211)
plot(t,uT,t,u_o,'r--')
ylabel('Temperature (K)','interpreter','latex')
xlabel('Time (min)','interpreter','latex')
legend('Controlled Process', 'Optimal Trajectory')
subplot(212)
plot(t,uV,t,v_o,'r--')
ylabel('Air Velocity ($\frac{m}{s}$)','interpreter','latex')
xlabel('Time (min)','interpreter','latex')
legend('Controlled Process', 'Optimal Trajectory')
```

Listing D.6: Main Matlab script for initialization, simulation and processing the control system

Appendix E

Additional Figures

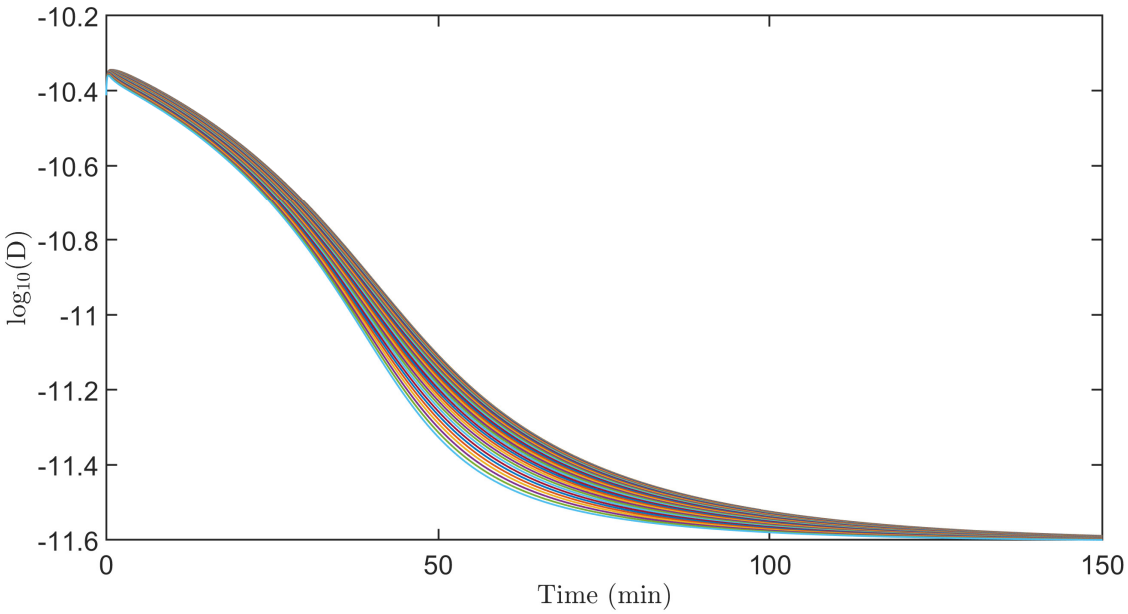


Figure E.1: The diffusion coefficient in the different layers throughout the drying time.

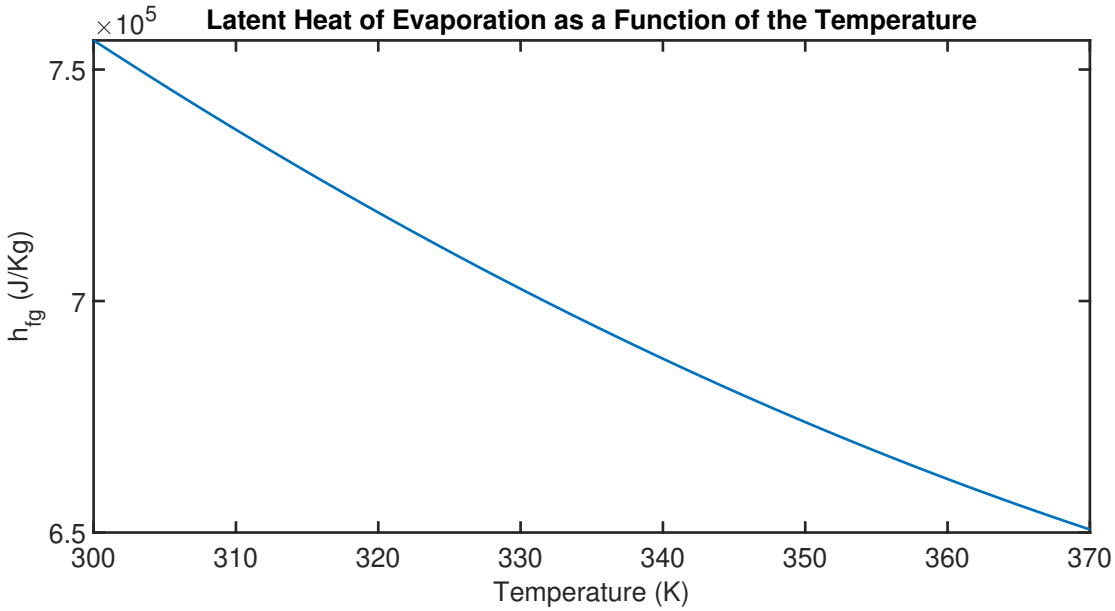


Figure E.2: The latent heat of evaporation as a function of the temperature.

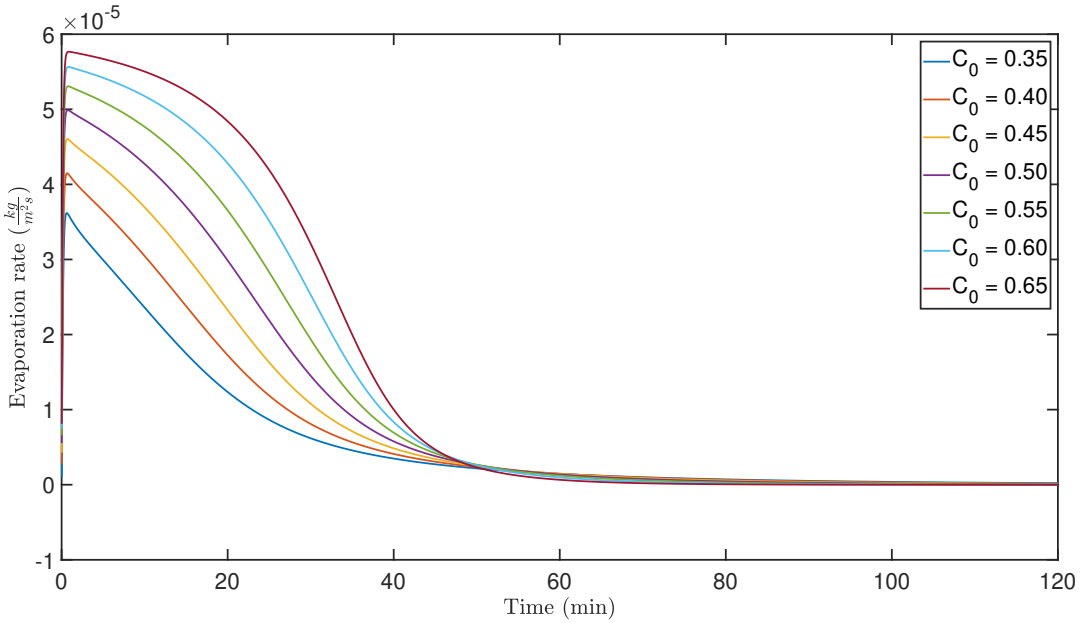


Figure E.3: Evaporation rate with different initial solvent fractions.

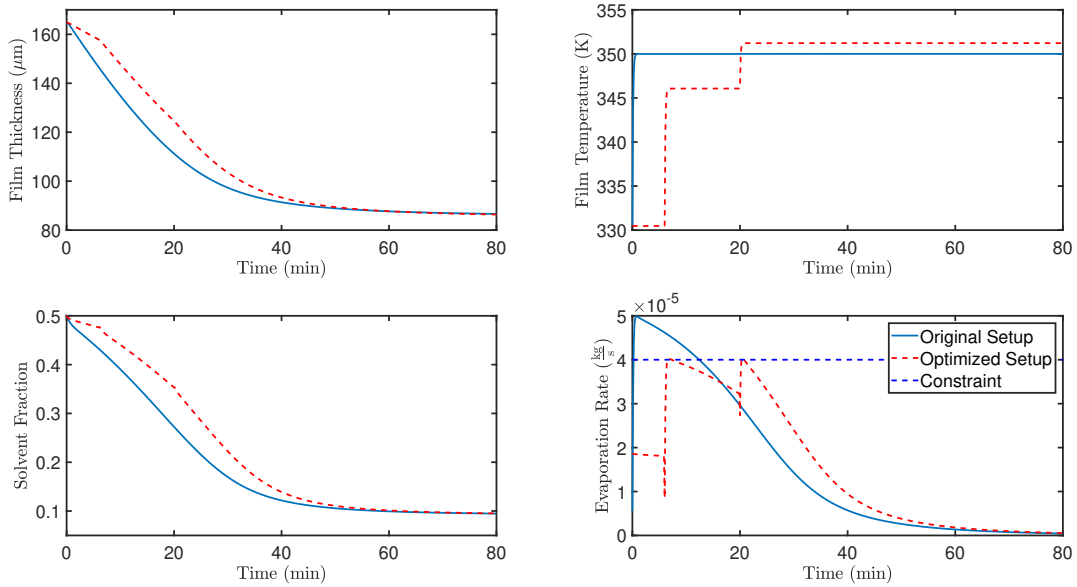


Figure E.4: Comparison of the numerical solution with and without optimized control inputs with the constrain $e_i \leq 4 \cdot 10^{-5}$. The drying zone transition is set at $t_1 = 6$ and $t_2 = 29$ min.

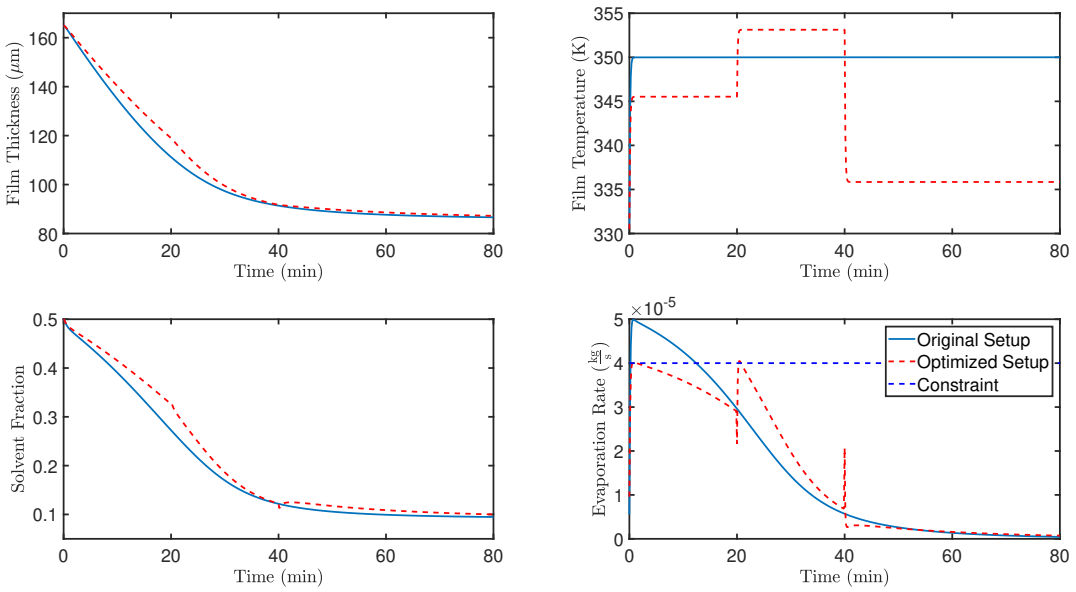


Figure E.5: Comparison of the numerical solution with and without optimized control inputs with the constrain $e_i \leq 4 \cdot 10^{-5}$. The drying zone transition is set at $t_1 = 20$ and $t_2 = 40$ min.

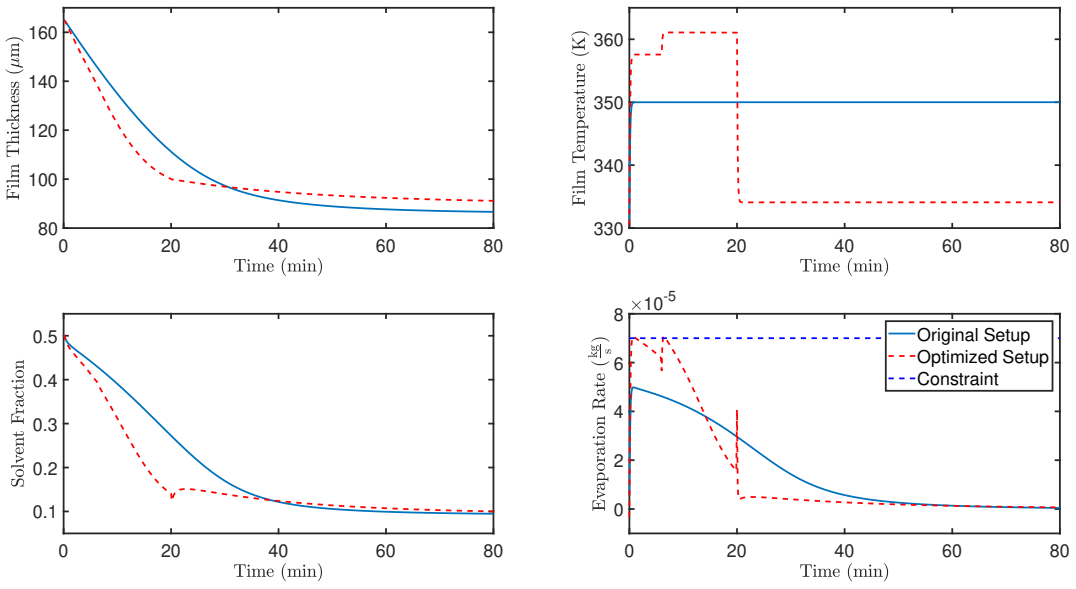


Figure E.6: Comparison of the numerical solution with and without optimized control inputs with the constrain $e_i \leq 7 \cdot 10^{-5}$. The drying zone transition is set at $t_1 = 6$ and $t_2 = 20$ min.

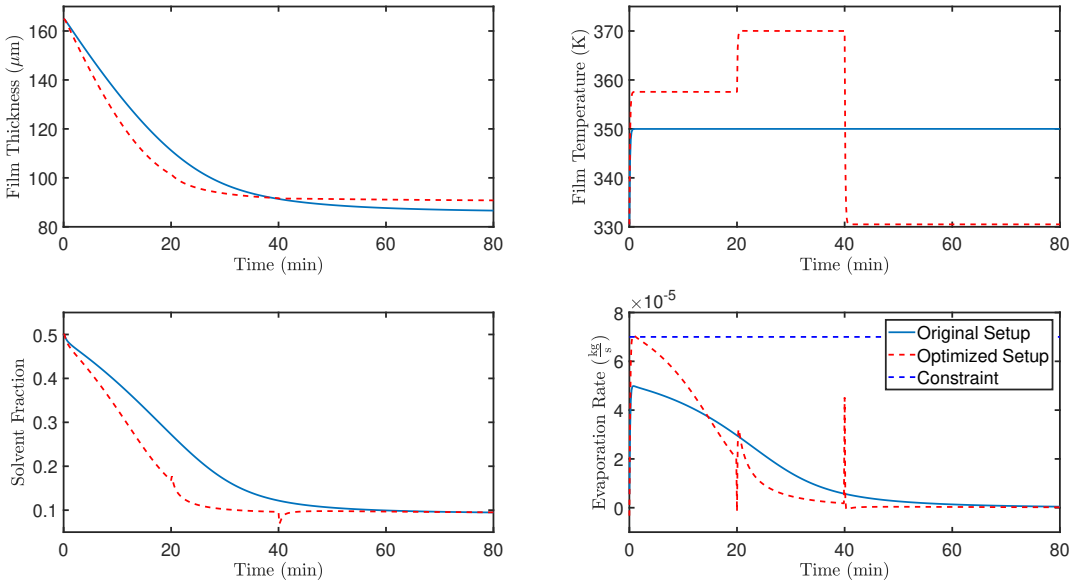


Figure E.7: Comparison of the numerical solution with and without optimized control inputs with the constrain $e_i \leq 7 \cdot 10^{-5}$. The drying zone transition is set at $t_1 = 20$ and $t_2 = 40$ min.

Appendix F

Simulink Implementations

F.1 Implementation of the Overall System

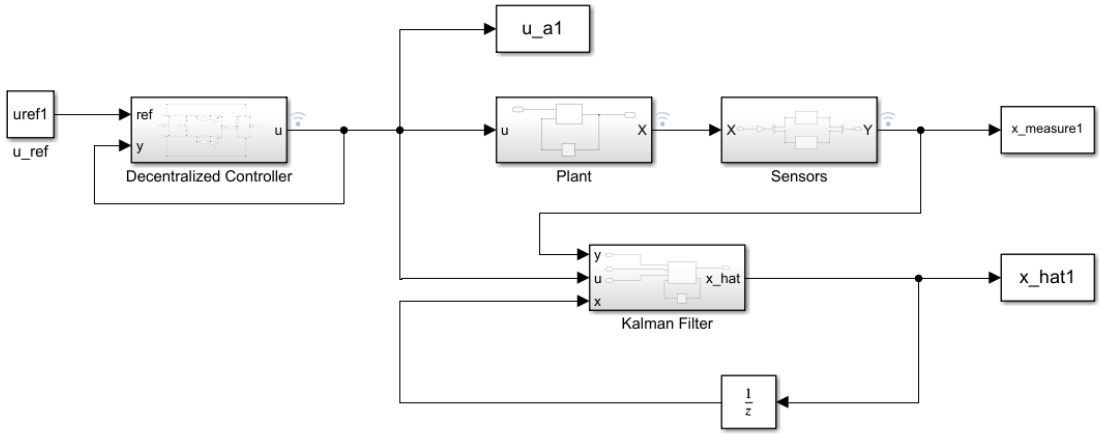


Figure F.1: Implementation of overall system in Simulink.

Emil Oppegård

Optimization and Control of Battery Electrode Drying Process

NTNU
Norwegian University of
Science and Technology