

Implementation of Kalman Filtering for Differential-Algebraic Equations

by

Bogdan Sherbak

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Applied Mathematics

Waterloo, Ontario, Canada, 2020

© Bogdan Sherbak 2020

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis describes two Kalman filters which are usable on semi-explicit index-1 differential-algebraic equations, prior to which a discussion of linear and nonlinear Kalman filters is presented. Performance between differential-algebraic equation-compatible Kalman filters and their ordinary differential equation counterparts is compared in two examples. Basic existence and uniqueness theory of linear differential-algebraic equations is discussed along with the process of numerically approximating the solution. Desire to estimate the state of charge of a lithium ion cell is used as motivation. The electrochemical processes of a lithium ion cell are discussed. When discretized, the model of a lithium ion cell results in a differential-algebraic equation.

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Kirsten Morris, who guided me throughout this project. I would also like to thank my family and friends who supported me. I wish to acknowledge the help provided by the technical and support staff in the Department of Applied Mathematics and in the University of Waterloo as a whole. I would also like to show my appreciation to my committee members for their insight.

Table of Contents

| | |
|---|-----------|
| List of Figures | vii |
| List of Tables | x |
| 1 Introduction | 1 |
| 2 Model of a Lithium-Ion Cell | 4 |
| 2.1 Geometry and Electrochemical Processes of a Lithium-Ion Cell | 4 |
| 2.2 Full PDE Model of a Lithium-Ion Cell | 10 |
| 2.3 Chebyshev Collocation Method for Discretizing PDEs | 11 |
| 3 Differential Algebraic Equations | 20 |
| 3.1 Theory of Linear DAE's with Constant Coefficients | 20 |
| 3.2 Numerical Solution of DAE's using BDF | 28 |
| 4 Kalman Filtering of ODE's and DAE's | 33 |
| 4.1 Estimation of Linear Systems | 34 |
| 4.1.1 Least Squares Estimation | 34 |
| 4.1.2 Weighted Least Squares Estimation | 35 |
| 4.1.3 Recursive Estimation | 36 |
| 4.1.4 Propagating Mean and Error | 38 |
| 4.1.5 Kalman Filter for Linear Systems | 40 |
| 4.2 Kalman Filters for Nonlinear Systems | 42 |
| 4.2.1 Extended Kalman Filter | 42 |
| 4.2.2 Unscented Kalman Filter | 43 |
| 4.3 Kalman Filters for semi-explicit index-1 DAE | 49 |
| 4.3.1 EKF for semi-explicit index-1 DAE | 50 |
| 4.3.2 UKF for semi-explicit index-1 DAE | 52 |
| 4.4 Numerical Example - Chemical Reactor | 53 |
| 4.5 Numerical Example - Chemical Akzo Nobel Problem | 56 |
| 4.6 Numerical Example - Estimating Lithium-ion Cell SOC using a Kalman filter | 63 |
| 5 Conclusion | 68 |

List of Figures

| | | |
|-----|--|----|
| 2.1 | A picture of a lithium ion cell, [63]. The left- and right-most boundaries are made up of current collectors which serve as connection points between the cell and the external world. The left and right parts of the domain consist of a mixture of solid material and electrolyte. During discharging or charging, the solid materials consume (produce) lithium ions which condense from (dissolve into) the electrolyte. The middle consists of a porous separator which allows for the free movement of the electrolyte and lithium ions while ensuring that the solid material on either ends of the cell never come into contact with one another. | 6 |
| 2.2 | Runge Function and its interpolating polynomials with equidistant interpolating points. Runge's phenomenon is apparent for polynomials of order 12 and 14. | 12 |
| 2.3 | A plot of the first five Chebyshev polynomials of the first kind. | 14 |
| 2.4 | Runge Function and its interpolating polynomials with non-equidistant interpolating points. The interpolating points are located at extremas of Chebyshev polynomials of the first kind. Notice the lack of Runge's phenomenon. | 15 |
| 2.5 | Solution of Poisson's equation where the forcing function is $f(x) = x^3 + e^x$ and boundary conditions are homogeneous Dirichlet on the left and homogeneous Neumann on the right. The exact solution is $u(x) = (x^5 - 5x - 4)/20 - e(x + 1) + \exp(x) - e^{-1}$. Solution is approximated with 3 and 5 points. | 16 |
| 2.6 | Approximate solution of the heat equation $\partial u / \partial t = \partial^2 u / \partial x^2 + x^3 + e^x$ for $t \in [0, 10]$ with a homogeneous Dirichlet boundary condition on the left and a homogeneous Neumann condition on the right and the temperature is identically zero as the zero condition. The number of interpolation points used is 9, these points are located at the extremas of the 8th degree Chebyshev polynomial of the first kind. | 17 |
| 3.1 | Approximate solution of equation (3.8) and initial condition defined in Example 3.25. The equation is solved numerically in the time domain $t \in [0, 10]$ using <code>ode15s</code> in Matlab. | 32 |

| | | |
|-----|--|----|
| 4.1 | <p>The true state and its estimates of free falling mass. The initial position is $3 \times 10^5 m$ and the initial velocity is $-300 m/s$.</p> <p>Subplot 1: the true position of the falling mass (blue solid line), the EKF estimate of the position (red dashed line) and the UKF estimate of the estimate (yellow dash-dot line).</p> <p>Subplot 2: the true velocity of the falling mass (blue solid line), the EKF estimate of the velocity (red dashed line) and the UKF estimate of the velocity (yellow dash-dot line).</p> <p>After 250 seconds, the UKF estimate closely approximates the true position and velocity whereas there's a larger error in the EKF estimate of the position.</p> | 47 |
| 4.2 | <p>The covariances of the estimates of free falling mass.</p> <p>Subplot 1: the covariance of the EKF position estimate (blue dashed line) and the covariance of the UKF position estimate (red dash-dot line).</p> <p>Subplot 2: the covariance of the EKF velocity estimate (blue dashed line) and the covariance of the UKF velocity estimate (red dash-dot line).</p> <p>The UKF covariances are significantly lower than the EKF covariances by the end of the simulation, however the UKF velocity covariance has a hump before both the UKF position and velocity covariances drop significantly below the EKF covariances.</p> | 48 |
| 4.3 | <p>Plot of the chemical reactor in Example 4.2. The true state (black dotted line) and the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line). The temperature is measured with no noise. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 0.0336 s, 0.0513 s, 0.1820 s and 0.1814 s, respectively.</p> | 55 |
| 4.4 | <p>Plot of the chemical reactor in Example 4.2. The true state (black dotted line) and the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line). The concentration is measured with no noise. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 0.0295 s, 0.0477 s, 0.2151 s and 0.1962 s, respectively.</p> | 56 |
| 4.5 | <p>Plot of the chemical reactor in Example 4.2. The true state (black dotted line) and the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line). The reaction rate per unit volume is measured with no noise. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 0.0312 s, 0.0466 s, 0.1788 s and 0.1824 s, respectively.</p> | 57 |
| 4.6 | <p>Plot of the chemical reactor in Example 4.2. The measured state is temperature, the variance of the measurement noise is equal to $\sqrt{5}$. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 0.0342 s, 0.0562 s, 0.2153 s and 0.1853 s, respectively.</p> | 58 |

| | | |
|------|---|----|
| 4.7 | Plot of the chemical reactor in Example 4.3. The measured states are the third and the fifth, there is no measurement noise. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 6.6442 s, 15.5019 s, 184.2725 s and 138.3562 s, respectively. | 61 |
| 4.8 | Plot of the chemical reactor in Example 4.3. The measured states are the third and the fifth, the variances of the measurement noises are 9×10^{-4} and 10^{-8} , respectively. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 7.5082 s, 16.4353 s, 178.7342 s and 138.5590 s, respectively. | 62 |
| 4.9 | The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. There is no measurement noise in the system, the absolute error converges to about 7×10^{-4} . There is no current being applied on the cell. | 64 |
| 4.10 | The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. The measurement is noisy, the variance of the noise is 10^{-4} . The absolute error converges to about 9×10^{-3} . There is no current being applied on the cell. | 64 |
| 4.11 | The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. The measurement is noisy, the variance of the noise is 9×10^{-4} , the absolute error converges to about 2×10^{-2} . There is no current being applied on the cell. | 65 |
| 4.12 | The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. The estimate is given time to converge to the true state after which the cell is charged and discharged. There is no measurement noise. After the current is applied, the absolute error does not exceed 10^{-3} | 65 |
| 4.13 | The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. The cell is charged for a period of time, prior and after which the estimate is given time to converge to the true state. There is no measurement noise. After the current is applied, the absolute error does not exceed 10^{-3} | 66 |
| 4.14 | The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. The cell is charged for a period of time, prior and after which the estimate is given time to converge to the true state. Measurement noise has variance equal to 10^{-8} . There is a charging current being applied on the cell. After the current is applied, the absolute error does not exceed 10^{-3} | 67 |

List of Tables

| | | |
|-----|--|----|
| 2.1 | Parameter values for the model of a lithium-ion cell. | 19 |
| 3.1 | Values of α for k -step BDF methods where $1 \leq k \leq 6$ | 29 |
| 4.1 | Parameter values for falling mass example. | 46 |

Chapter 1

Introduction

Batteries are the basic energy storage unit for many devices in our everyday life, ranging from cell phones and laptops to medical equipment to hybrid and electric vehicles. They provide a source of energy for portable and self-propelled equipment that cannot always be connected to an electrical grid. An important point to consider when operating a self-powered electrical device is how long it can stay self-powered, that is how much energy is remaining in its battery.

Electrochemical cells are the elementary units of battery packs. An electricity storage unit in a hybrid/electric vehicle may be made up of a number of battery packs. State of charge (SOC) is the amount of energy left inside a cell compared to what is available when it is at its maximum capacity. It is given as a percentage between 0% and 100% where 0% is fully discharged and 100% is fully charged. Tracking the SOC is the main objective of a battery management system, this is done by measuring the voltage and, if available, current and temperature and using these values to update the SOC estimate. An accurate SOC estimate is important for the following reasons:

- Cells can be over-charged and over-discharged which can lead to damage. Access to an accurate SOC estimate can open the possibility of preventative measures which can prevent usage when the estimate is 0% and stop charging when the estimate is 100%.
- An operator can better plan the use of their equipment if they have access to an accurate SOC estimate. Conservative use for fear of running out of charge at an inconvenient time due to inaccurate SOC estimate can be eliminated. This can lead to more efficient usage and usage of lighter and cheaper batteries.
- During normal operation, battery packs may be discharged at different rates. This will lead to some packs depleting before others resulting in a drop of voltage and/or amperage until these batteries are recharged. A battery management system can balance battery packs so that they discharge at equivalent rates. This requires an accurate SOC estimate for each battery pack.
- Each time a battery is recharged, its life cycle decreases. Access to accurate SOC can lead to less premature charging.

Lithium-ion is a popular choice of battery material, other choices include nickel-cadmium and nickel-metal hydride. Lithium-ion batteries have some advantages over other materials: no memory effect, high life cycle, low self-discharge and high power.

The problem addressed in this thesis is that the SOC in a lithium-ion cell, or indeed in any widely used electrochemical cell, cannot be directly measured. Compare this to a fuel tank in which the remaining level of fuel can be measured by a simple float. Indirect methods are the remaining approach, this approach estimates the SOC via measurements of other values of a battery cell such as voltage. The reader might be familiar with using voltage to infer the SOC.

The approach taken in this thesis involves using mathematical models to simulate the battery processes. The processes of the physical cell are simulated with the goal of estimating the processes inside the physical cell by observing the simulated cell. If we can estimate the internal state of a cell we can then estimate the SOC. This is the purpose of the mathematical model.

A limitation of mathematical models is they can become very complicated even when modelling simple dynamics so simplifications and assumptions have to be made. This of course leads to errors in the model which can affect its accuracy. These errors compound with time leading to a bad estimate unless the model is re-calibrated at regular intervals.

One way to address this is to use feedback from the system being modelled as a corrective input into the model.

It should be noted here, the objective of this thesis is to present an implementation of a Kalman filter estimator that is applicable on the model of the lithium-ion cell. However, smartphones and on-board computers in common cars have limited CPU's which could not handle expensive on-line computations, this puts a constraint on the complexity of the estimator.

Before modeling the lithium-ion cell it is important to understand the origin and effect of the governing equations as well as the working domain. An analysis of this is done in Chapter 2 as well as a brief explanation of the spectral method of transforming the system partial differential equations into a system of differential-algebraic equations.

As is shown in Chapter 2, our model is a system of differential-algebraic equations (DAE's) and theory for these is not widely known. As such, existence-uniqueness theory of DAE's is presented in Chapter 3 for the linear case. Further, theory for the numerical solution of DAE's is also given.

Chapter 4 includes a discussion of Kalman filters for linear and non-linear ODE's. This leads into a discussion of non-linear Kalman filters that are applicable on DAE's. These Kalman filters are applied on example DAE's which includes the lithium-ion cell.

A DAE-compatible Kalman filter was first introduced by Becerra et al. [7], this is an extension of the extended Kalman filter modified to accept DAE's. Mandela et al. [39] improved upon this implementation of a DAE-compatible extended Kalman filter and introduced a DAE-compatible unscented Kalman filter. Purohit et al. [53] and Puranik et al. [54] developed a DAE-compatible Ensemble Kalman filter and a DAE-compatible iterative extended Kalman filter, respectively. An implementation of a DAE-compatible Kalman filter

has been applied on the model of a lithium-ion cell [12].

In this thesis, we examine the performance of DAE-compatible extended and unscented Kalman filters and their ODE counterparts. The DAE-compatible filters are applied on the original DAE models and the ODE-compatible filters are applied on the DAE models converted to ODEs. The performance of the ODE- and DAE-compatible state estimators is compared for several examples. The process for converting a DAE into an ODE is discussed in this thesis.

An easy but inaccurate estimate for the SOC can be obtained through measurement of the terminal voltage of the cell. Examples can be found here [74, 76]. Indeed, the terminal voltage is proportional to the SOC, however open circuit voltage plotted versus the SOC forms a hysteresis loop [2, 24, 65, 79] which makes using the terminal voltage as an estimate difficult. The terminal voltage depends on ambient temperature [24, 74].

Another common approach is known as coulomb counting. This method measures and integrates the current of the cell. Any errors caused by noise and rounding will grow due to the cumulative effect of integration. Examples of coulomb counting to produce an SOC estimate can be found here [50, 52, 70]. The accuracy of coulomb counting methods is affected by charging/discharging efficiency, self-discharging, capacity loss and memory effects as well as an accurate initial estimate of the SOC is required [10, 42, 50, 78]. These limitations can be overcome by book-keeping methods [10].

Model-based estimators are more accurate methods of estimating SOC. They can be divided into two categories, circuit and electrochemical. Circuit models are simple and have few parameters. Electrochemical models describe the physical and chemical behaviour inside the cell. They are complex and are limited in their online use due to computation requirements however the effects of ambient temperature and aging are easier to implement into the model. Difficulty in obtaining an accurate SOC measurement leads to conservative SOC measurement preventing the full cell capacity from being used. Accurate SOC measurement will allow the user to confidently use the cell to its full potential.

The SOC estimate is evaluated with the use of a Kalman filter where the measurements are voltage, current, the coulomb-counting estimate or a combination. An overview of electrochemical model-based methods can be found here [24, 57]. In this thesis, the SOC is estimated using the voltage as the only measurement. Other states of the lithium ion cell, such as the concentration of lithium ions are infeasible to measure.

The approach in this thesis is to model the lithium-ion cell using partial differential equations and apply a Kalman filter on it. Examples of a finite element simulation of the lithium-ion cell can be found here [63], finite difference [8], finite volume [77], spectral method [12]. Examples of applying a Kalman filter on a lithium-ion cell can be found here [3, 6, 12, 17, 34, 49, 51, 57, 64, 71, 75]. Other methods for estimating SOC include Monte Carlo [19] and neural networks [33].

There is interest in estimating other states in a cell, such as stress [25, 49], temperature [41, 44, 1], states of health and power [44] and lifetime [18].

Chapter 2

Model of a Lithium-Ion Cell

From the different methods of estimating SOC of a lithium-ion cell, model-based estimators are considered here. In this thesis, a model is constructed using equations which describe the physical and electrochemical processes in a lithium-ion cell. The states of this model are then used to estimate the state of charge of the electrochemical cell. Voltage measurements are derived from the model as well as measured from the physical cell and used to provide corrective forces on the model. An model of an electro-chemical cell is presented in this chapter. For more information on the model see [2, 66].

In everyday language, when we say battery, we sometimes mean cell. It's important to draw a distinction when looking at this on a technical level: a (electrochemical) *cell* is device that generates electrical energy from chemical reactions, a *battery* is a collection of cells arranged in series, parallel or a combination of the two. This chapter covers several topics: an explanation of the processes inside a lithium-ion cell, the derivation of the equations governing the processes inside a lithium ion cell and the process for simulating the resulting equations.

2.1 Geometry and Electrochemical Processes of a Lithium-Ion Cell

An electrochemical cell has three main components, a negative electrode, a positive electrode and an electrolyte. The electrodes are submerged in the electrolyte, have no contact with each other and are connected to current collectors. Everything is encased except for the current collectors, also known as terminals, which serve as the connection to the outside world, see Figure 2.1. The electrodes are made up of a slurry of conductive and porous material which we call here the solid particles or the solid phase.

In a lithium-ion cell, the positive electrode is a metal oxide, here lithium ferrophosphate (LFP, LiFePO_4), and the negative electrode is carbon. However, in practice, the electrodes are a mix of active materials and electrolyte. The electrodes undergo a **reduction-oxidation reaction** or **redox reaction** to generate electrical energy.

- Reduction: the gain of electrons.
- Oxidation: the loss of electrons.

In a fully charged cell the negative electrode has an abundance of electrons and (positively charged) lithium ions compared to the positive electrode. When the electrodes are connected by an external circuit the electrons flow to the positive electrode while the lithium ions dissolve at the negative electrode, travel through the electrolyte and are absorbed by the positive electrode. This proceeds until the connection is broken or the electric potential between the electrodes is zero and the cell is said to be *discharged*.

During discharge, the negative electrode undergoes *oxidation* and the positive electrode undergoes *reduction*. The electrode that undergoes *reduction* is called the **cathode** while the electrode that undergoes *oxidation* is called the **anode**.

When a cell is being charged, external energy is used to force oxidation at the positive electrode, force reduction at the negative electrode and move lithium ions from the positive to the negative electrode.

To construct a model of an electrochemical cell it's critical to understand its physical and electrochemical processes and how it affects the states of the system. Four states will be considered: the concentration of lithium-ions in the electrolyte, the concentration of lithium-ions in the solid, the electric potential of the electrolyte and the electric potential of the solid.

See figure 2.1 for a visual of the cell and [2, 66] for more information about the assumptions.

The mathematical model of a lithium-ion cell models four variables, the concentration of lithium ions in the electrolyte and the electrodes and the potential energy of the electrolyte and the electrodes. These are the assumptions for the model:

- The cell has one spatial dimension, ξ ,
- The negative electrode occupies $\xi \in [0, L_-]$, the positive occupies $\xi \in [L - L_+, L]$, and the electrolyte is everywhere, $\xi \in [0, L]$, thus the electrode regions are a superposition of electrode and electrolyte,
- The volume of the electrodes does not change as they produce/consume lithium ions or electrons and the solid particles release and absorb lithium ions through their boundaries,
- The electrodes are a collection of perfectly spherical solid particles,
- The concentration in a solid particles does not depend on angle, ie., for a given radius, r , the concentration is constant regardless of the angles θ and ϕ .

Butler-Volmer Equation and State of Charge

The Butler-Volmer equation (also known as the Erdey-Grúz-Volmer equation) is an important equation in model electrochemical cells. It describes how the difference in voltage

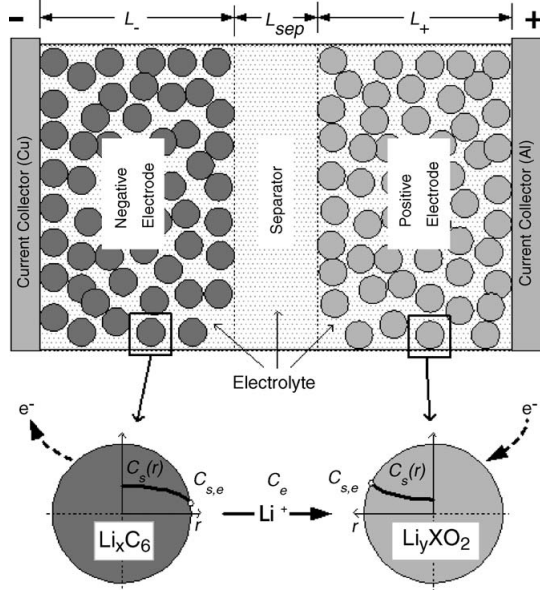


Figure 2.1: A picture of a lithium ion cell, [63]. The left- and right-most boundaries are made up of current collectors which serve as connection points between the cell and the external world. The left and right parts of the domain consist of a mixture of solid material and electrolyte. During discharging or charging, the solid materials consume (produce) lithium ions which condense from (dissolve into) the electrolyte. The middle consists of a porous separator which allows for the free movement of the electrolyte and lithium ions while ensuring that the solid material on either ends of the cell never come into contact with one another.

between the electrode and the electrolyte leads to a redox reaction. Given the difference in electric potentials between the electrolyte and solid material the Butler-Volmer equation predicts the dissolution/consumption of lithium ions. In other words, the Butler-volmer equation describes all interaction between the electrolyte and electrodes.

$$i^{li}(\xi, t) = a_s i_0 \left(\exp\left(\frac{\alpha F \eta}{RT}\right) - \exp\left(-\frac{\alpha F \eta}{RT}\right) \right) = 2a_s i_0 \sinh\left(\frac{\alpha F \eta}{RT}\right) \quad (2.1)$$

where $i^{li}(\xi, t)$ is the reaction rate of species k at location ξ and time t ,

- j_0 : exchange current density,
- $\eta(\xi, t) = \phi_s(\xi, t) - \phi_e(\xi, t) - U(\theta)$: activation overpotential - required potential difference to produce a current,
- $U(\theta)$: open circuit potential - potential of electrode as a function of concentration,
- θ_i : state of charge in electrode i .

and the rest of the parameters can be found in Table 2.1. The current exchange density is

$$j_0 = kF (c_s^{max} - c_s^{surf})^\alpha (c_s^{surf})^\alpha (c_e)^\alpha, \quad (2.2)$$

where

- k : reaction rate constant,
- c_s^{max} : the maximum concentration in the solid particle,
- c_s^{surf} : the surface concentration at the solid particle,
- c_e : the concentration in the electrolyte.

For compactness, let $i^{li}(\cdot) = i^{li}(\xi, t)$.

In the Butler-Volmer equation, we have $\sinh(\alpha F \eta / RT)$. At 20°C, $\frac{F}{RT} \approx 39.6$, since this is wrapped inside an exponential function, any change in η will result in a large change in $i^{li}(\cdot)$. This will likely lead to a computationally stiff dynamic.

The open circuit potential, $U(\theta_i)$ is described by the following equation according to [55]:

$$U(\theta_i) = 0.7222 + 0.1387\theta_i + 0.029\theta_i^{1/2} - \frac{0.0172}{\theta_i} + \frac{0.0019}{\theta_i^{1.5}} + 0.2808 \exp(0.9 - 15\theta_i) - 0.7984 \exp(0.4465\theta_i - 0.4108). \quad (2.3)$$

Define the average concentration across the electrode, ν_i^{avg} ,

$$\nu_a^{avg} = \frac{3}{L_- R^3} \int_0^{L_i} \int_0^R \frac{r^2 c_s}{c_s^{max}} dr d\xi \quad \nu_c^{avg} = \frac{3}{L_+ R^3} \int_{L_- + L_{sep}}^L \int_0^R \frac{r^2 c_s}{c_s^{max}} dr d\xi \quad (2.4)$$

for the anode and cathode, respectively, and ν_i^0 and ν_i^1 are the values when the concentration is at minimum and maximum values, respectively

Then the state of charge is [2, 12, 45]:

$$\theta_i = \frac{\nu_i^{avg} - \nu_i^0}{\nu_i^1 - \nu_i^0}. \quad (2.5)$$

Concentration of Lithium Ions in the Electrodes

The solid phase's mass balance of the lithium in a single particle is described by Fick's second law in spherical coordinates. $c_s(\xi, r, t) = c_s$ is the concentration of lithium ions at position ξ , radius r and time t , the D_s is the diffusion coefficient, s in the subscript indicates the solid phase and $r \in [0, R]$ where R is the radius of the particle. We assume that the concentration in the spherical particles is dependent only on the radius and not on the polar or azimuthal angles thus we have

$$\frac{\partial c_s(\xi, r, t)}{\partial t} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(D_s r^2 \frac{\partial}{\partial r} c_s(\xi, r, t) \right). \quad (2.6)$$

There is zero flux at $r = 0$ which leads to a homogeneous Neumann boundary. At the right boundary, $r = R$, the flux is dependent on the value of the Butler-Volmer equation at the spatial dimension ξ , this means we have a variable Neumann boundary. Thus the boundary conditions are:

$$\frac{\partial c_s}{\partial r}\Big|_{r=0} = 0, \quad \frac{\partial c_s}{\partial r}\Big|_{r=R} = \frac{i^{li}(\cdot)}{a_s F}. \quad (2.7)$$

This equation must be solved at every point along the radius of every particle along the spatial domain of both electrodes.

Concentration of Lithium Ions in Electrolyte

Consider the concentration of lithium ions in the electrolyte. When the cell is being charged, the lithium ions dissolve into the electrolyte at the positive electrode, travel through the electrolyte and are consumed by the negative electrode.

We assume here that we can ignore the convection of the lithium ions and the volume change of the electrodes as they gain or lose lithium ions and electrons [66]. We assume the lithium ions are only subject to the dynamics described by Fick's law and Butler-Volmer equation. The electrode regions are a superposition of electrolyte and solid material. Each lithium ion is assumed to be dissolved in one or the other, neither however, ever changes volume, by assumption.

For the lithium ions in the electrolyte phase, the lithium ion concentration in the electrolyte phase is defined by

$$\epsilon_e \frac{\partial c_e}{\partial t} = \frac{\partial}{\partial \xi} \left(D_e^{eff}(c_e) \frac{\partial c_e}{\partial \xi} \right) + \frac{1 - t_+^0}{F} i^{li}(\cdot) \quad (2.8)$$

where $D_e^{eff}(\cdot)$ is the effective electrolyte diffusion coefficient and t_+^0 is the transference number of the lithium ion with respect to the solvent velocity. In (2.8) the term $\frac{\partial}{\partial \xi} \left(D_e^{eff}(\cdot) \frac{\partial c_e}{\partial \xi} \right)$ is a diffusion term due to Fick's law, and $\frac{1 - t_+^0}{F} i^{li}(\cdot)$ is a reaction term due to the production and consumption of lithium ions at the electrodes as modelled by the Butler-Volmer equation across the electrodes. Note that the reaction term will always be zero in the separator region.

The effective diffusion coefficient is equal to the diffusion coefficient times electrolyte volume fraction taken to the power of the Bruggeman coefficient,

$$D_e^{eff} = D_e \epsilon_e^b.$$

The diffusion coefficient D_e has been shown in [69] to match the following equation:

$$D_e(c_e) = \exp \left[D_{00} + \frac{D_{01}}{T - T_{g0} - T_{g1} c_e} + D_{10} c_e \right] \quad (2.9)$$

where

$$D_{00} = -4.43, \quad D_{01} = -54, \quad D_{10} = -0.22, \quad T_{g0} = 229, \quad T_{g1} = 5,$$

and T is temperature in Kelvin.

The electrolyte exists in every part of the cell thus (2.8) must be solved in all regions of the cell, positive and negative electrodes and the separator region.

The electrolyte cannot escape beyond the bounds of the cell, in particular the lithium ions in the electrolyte cannot escape the bounds of the cell. This zero flux leads to the Neumann boundary conditions

$$\left. \frac{\partial c}{\partial \xi} \right|_{\xi=0} = \left. \frac{\partial c}{\partial \xi} \right|_{\xi=L} = 0 \quad (2.10)$$

where L is the length of the 1-D cell.

Electric Potential in Electrolyte

In this thesis, the equation describing the electric potential in the electrolyte, ϕ_e , is assumed to be [2, 12, 66]

$$0 = \frac{\partial}{\partial \xi} \left(k^{eff} \frac{\partial \phi_e}{\partial \xi} + k_D^{eff} \frac{\partial \ln(c_e)}{\partial \xi} \right) + a_s i^{li}(\cdot) \quad (2.11)$$

where the diffusional conductivity is $k_D^{eff} = k^{eff} \frac{RT(1-2t_+^0)}{F}$ and the effective conductivity k^{eff} is $k^{eff} = k^i \epsilon_e^b$ and the ionic conductivity k^i has been show in [69] to be equal to

$$k^i = \left(\sum_{i=1}^3 \sum_{j=1}^3 k_{ij} c_e^{i-1} T^{j-1} \right)^2 c_e,$$

where k_{ij} are the entries of

$$\begin{bmatrix} -10.5 & 0.0740 & -6.96e - 5 \\ 0.668 & -0.0178 & 2.80e - 5 \\ 0.494 & -8.86e - 4 & 0 \end{bmatrix}.$$

This means that the second derivative of electric potential of the electrolyte is assumed to be dependent on the concentration of lithium ions in the electrolyte as well as on the value of the Butler-Volmer equation.

The electric potential is time-independent. This is because it always reflects the present state its environment. This property results in the appearance of an algebraic constraint in the system of equations.

In order to ensure that there is a unique solution for the potential energy and since potential energy is a relative quantity we can assign it a value of zero at one point in the domain [2], we do this by enforcing a homogeneous Dirichlet boundary condition on the left boundary and since the potential energy cannot change beyond the boundary of the cell we have a homogeneous boundary condition on the right. Thus the boundary conditions are

$$\phi_e|_{\xi=0} = \left. \frac{\partial \phi_e}{\partial \xi} \right|_{\xi=L} = 0. \quad (2.12)$$

this completes the derivation of the equation describing the electric potential of the electrolyte.

Electric Potential in Electrodes

The electric potential in the solid material, ϕ_s , is described by the following equation, [2, 66]

$$\frac{\partial}{\partial \xi} \left(\sigma^{eff} \frac{\partial \phi_s}{\partial \xi} \right) = a_s i^{li} \quad (2.13)$$

The second derivative of the electric potential in the electrodes is dependent on the value of the Butler-Volmer equation. The boundary conditions are

$$\sigma^{eff} \frac{\partial \phi_s}{\partial \xi} \Big|_{\xi=0} = I(t), \quad \frac{\partial \phi_s}{\partial \xi} \Big|_{\xi=L_-} = 0 = \frac{\partial \phi_s}{\partial \xi} \Big|_{\xi=L_-+L_{sep}}, \quad \sigma^{eff} \frac{\partial \phi_s}{\partial \xi} \Big|_{\xi=L} = I(t). \quad (2.14)$$

There are no solids in the separator region ($\xi \in (L_-, L_{sep})$) so ϕ_s is not defined there, additionally, there is no flux into or out of that region.

In the above discussion, the four states of interest, concentration of lithium ions in the electrolyte, c_e , concentration of lithium ions in the solid material/electrodes, c_s , electric potential of the electrolyte, ϕ_e , electric potential of the electrode, ϕ_s were introduced along with the equations describing their behaviour.

2.2 Full PDE Model of a Lithium-Ion Cell

The model consists of four partial differential equations that were described in the previous section. Two of these equations, the electric potentials of the electrolyte and electrode, do not contain a temporal derivative which results in algebraic constraints within our model. This means that when the system of PDE's must be converted into a system of differential-algebraic equations not a system of ordinary differential equations.

Combining the preceding PDE's together in one place produces

$$\epsilon_e \frac{\partial c_e}{\partial t} = \frac{\partial}{\partial \xi} \left(D_e(\cdot) \frac{\partial c_e}{\partial \xi} \right) + \frac{1 - t_+^0}{F} i^{li}(\cdot) \quad (2.15a)$$

$$\frac{\partial c_s}{\partial t} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 D_s \frac{\partial c_s}{\partial r} \right) \quad (2.15b)$$

$$0 = \frac{\partial}{\partial \xi} \left(k^{eff} \frac{\partial \phi_e}{\partial \xi} + k_D^{eff} \frac{\partial \ln(c_e)}{\partial \xi} \right) + a_s i^{li}(\cdot) \quad (2.15c)$$

$$0 = \frac{\partial}{\partial \xi} \left(\sigma_s^{eff} \frac{\partial \phi_s}{\partial \xi} \right) - a_s i^{li}(\cdot) \quad (2.15d)$$

with the following boundary conditions

$$\frac{\partial c}{\partial \xi} \Big|_{\xi=0} = 0 \qquad \frac{\partial c}{\partial \xi} \Big|_{\xi=L} = 0 \qquad (2.16a)$$

$$\frac{\partial c_s}{\partial r} \Big|_{r=0} = 0 \qquad D_s \frac{\partial c_s}{\partial r} \Big|_{r=R} = \frac{-i^{li}(\cdot)}{a_s F} \qquad (2.16b)$$

$$\phi_e \Big|_{\xi=0} = 0 \qquad \frac{\partial \phi_e}{\partial \xi} \Big|_{\xi=L} = 0 \qquad (2.16c)$$

$$\sigma^{eff} \frac{\partial \phi_s}{\partial \xi} \Big|_{\xi=0} = I(t) \qquad \frac{\partial \phi_s}{\partial \xi} \Big|_{\xi=L_-} = 0 \qquad (2.16d)$$

$$\frac{\partial \phi_s}{\partial \xi} \Big|_{\xi=L_- + L_{sep}} = 0 \qquad \sigma^{eff} \frac{\partial \phi_s}{\partial \xi} \Big|_{\xi=L} = I(t). \qquad (2.16e)$$

The variable c_e is the concentration of lithium-ions in the electrolyte at time t and position ξ , c_s is the concentration of lithium-ions in the electrodes at time t , position ξ and radius r , ϕ_e is the electric potential of the electrolyte at time t and position ξ and ϕ_s is the electric potential of the electrodes at time t and position ξ .

The values of all constants and parameters used in the model are in Table 2.1.

The lithium ions are not able to travel through the ends of the cell hence the concentration of lithium ions in the electrolyte has homogeneous Neumann boundary conditions, (2.16a). The lithium ions in the spherical particles are not able to travel past zero radius leading to a homogeneous Neumann boundary condition on the left in (2.16b) but they are able to dissolve into the electrolyte and accumulate from the electrolyte leading to a non-homogeneous Neumann boundary condition on the right. Since potential energy is a relative quantity we can set the electric potential to be zero at a single point [2]. We do this on the left boundary by introducing a homogeneous Dirichlet boundary condition in (2.16c). On the right boundary, the potential energy is zero flux leading to a homogeneous Neumann boundary condition.

The domain for the electric potential in the solid phase is disjoint, it is present in the electrodes, from 0 to L_- and from $L_- + L_{sep}$ to L but not between L_- and $L_- + L_{sep}$, this is because there is no electrode in the separator region, $\xi \in (L_-, L_- + L_{sep})$. The electric potential of the electrodes is undefined outside of where the solid particles are present leading to homogeneous Neumann boundary conditions at the two inner boundaries, $x = L_-$ and $\xi = L_- + L_{sep}$ and non-homogeneous at the outer boundaries where the electrodes are connected to the current collectors (2.16d) (2.16e).

2.3 Chebyshev Collocation Method for Discretizing PDEs

Solving a partial differential equation using finite difference or finite element method involves approximating the solution in a local domain, the global solution is then obtained by combining local solutions

$$u(x) \approx u_N(x) = \sum_{i=1}^N \hat{u}_i \phi_i(x). \qquad (2.17)$$

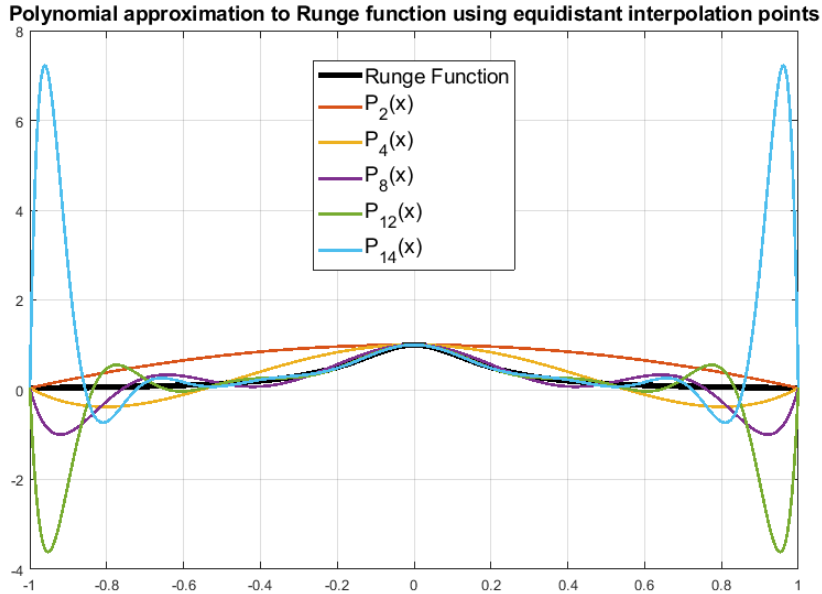


Figure 2.2: Runge Function and its interpolating polynomials with equidistant interpolating points. Runge’s phenomenon is apparent for polynomials of order 12 and 14.

Here \hat{u}_i are coefficients and $\phi_i(x)$ are polynomials defined on a subdomain, this forms a finite element approximation to $u(x)$.

Let the basis functions $\phi_i(x)$ in (2.17) instead be defined on the entire domain, this leads to the method for solving partial differential equations described in [13, 20, 38, 46, 68, 72]. For periodic domains, common choices of basis functions are $\cos(nx)$ or $\sin(nx)$. In our case the basis functions will be Chebyshev polynomials. In this method it is easy to find approximate solutions for partial differential equations.

Chebyshev Collocation

Let $P_N(x)$ be an N -degree polynomial approximation to a continuous function $f(x)$ where $x \in [a, b]$ and the interpolation points of $P_N(x)$ are equidistant. Then for a fixed $x^* \in [a, b]$, as $N \rightarrow \infty$, $P_N(x^*) \rightarrow f(x^*)$. However, for some continuous functions it may be the case that as $N \rightarrow \infty$, $\max_{x \in [a, b]} |P_N(x)| \rightarrow \infty$. In other words, the polynomial approximation converges piecewise but diverges uniformly, this is known as Runge’s phenomenon [46, 56, 21]. In Figure 2.2, Runge’s phenomenon can be seen for high order approximations to the Runge function ($f(x) = (1 + 25x^2)^{-1}$).

To address this issue in polynomial approximation, consider the Chebyshev polynomials of the first kind.

Definition 2.1. The n th-degree Chebyshev polynomial of the first kind, $T_n(x)$, is defined

recursively as

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_n(x) &= 2xT_{n-1}(x) - T_{n-2}(x). \end{aligned}$$

The Chebyshev polynomials of the first kind arise as the basis functions of the solution to the Chebyshev differential equation. The n th degree Chebyshev polynomial of the first kind is a solution to

$$(1 - x^2)y'' - xy' + n^2y = 0 \quad (2.18)$$

where n is a constant. In addition to being solutions to equation (2.18), they have found use as a good choice of basis functions in polynomial approximation.

There exist Chebyshev polynomial of the *second* kind, they are solutions to

$$(1 - x^2)y'' - xy' + n(n + 2)y = 0.$$

A discussion of these is beyond the scope of this thesis, see [13, 46] for more information.

Example 2.2. The first five Chebyshev polynomials of the first kind are:

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1. \end{aligned}$$

The first five Chebyshev polynomials of the first kind are plotted in Figure 2.3.

The Chebyshev polynomials of the first kind are orthogonal to each other with respect to the weight function $w(x) = (1 - x^2)^{-1/2}$, [46, Chapter 4]. In fact,

$$\int_{-1}^1 \frac{T_i(x)T_j(x)}{\sqrt{1 - x^2}} dx = \begin{cases} 0 & \text{if } i \neq j \\ \pi & \text{if } i = j = 0 \\ \frac{\pi}{2} & \text{if } i = j > 0 \end{cases}.$$

When choosing a set of basis functions, orthogonality between members of the set is a desired property.

Looking again at Runge's phenomenon, one way to eliminate it is to use interpolation points that are more densely distributed near the boundaries of the domain. A common way to do this is to use points that are the extremas of the Chebyshev polynomials of the first kind [67],

$$x_i = \cos\left(\frac{n}{N}\pi\right), \quad i = 0, \dots, N. \quad (2.19)$$

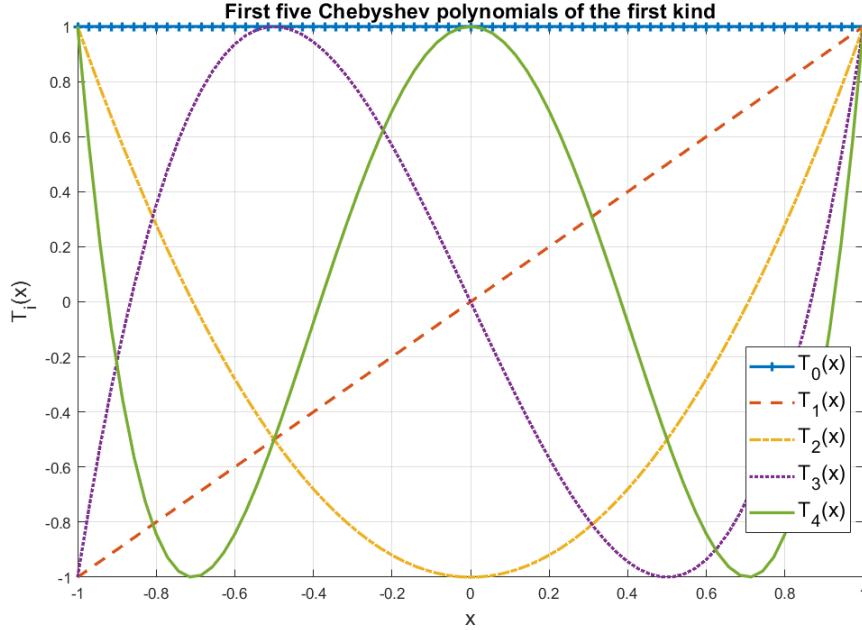


Figure 2.3: A plot of the first five Chebyshev polynomials of the first kind.

The points x_n are the extreme points of $T_N(x)$, the Chebyshev polynomial of the first kind of degree N . Compare Figures 2.2 and 2.4, the polynomial approximation in the latter is much better than the polynomial approximation in the former. This is because the Runge function is approximated with interpolation points located at extremas of Chebyshev polynomials of the first kind not with equidistant interpolation points.

Solving a PDE using Chebyshev Collocation Method

Let P_1 be an interpolating polynomial with interpolation points $x_0 = 1$, $x_1 = -1$ and values $f(x_0) = y_0$, $f(x_1) = y_1$:

$$P_1(x) = \frac{1}{2}(1+x)y_0 + \frac{1}{2}(1-x)y_1.$$

The derivative is:

$$P'_1(x) = \frac{1}{2}y_0 - \frac{1}{2}y_1.$$

Let U be a vector of values of a polynomial interpolant and consider the matrix

$$D = \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & -1/2 \end{bmatrix}. \quad (2.20)$$

If $U' = DU$, then U' is a vector of values of the derivative of a polynomial approximation.

Example 2.3. Let

$$U = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}$$

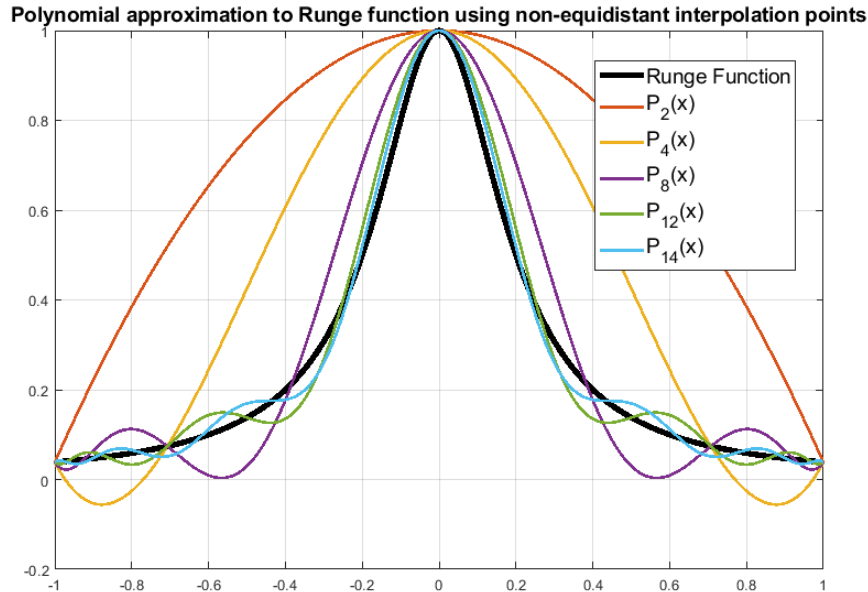


Figure 2.4: Runge Function and its interpolating polynomials with non-equidistant interpolating points. The interpolating points are located at extremas of Chebyshev polynomials of the first kind. Notice the lack of Runge's phenomenon.

be a vector of values at $x_0 = 1$ and $x_1 = -1$ of a polynomial approximation, the associated polynomial is

$$P_1(x) = \frac{1}{2}(1+x) \times 1 + \frac{1}{2}(1-x) \times \frac{1}{2}.$$

The derivative of the polynomial is

$$P_1'(x) = \frac{1}{4}.$$

We can obtain the values of the derivative at $x_0 = 1$ and $x_1 = -1$ by multiplying U by the differentiation matrix in equation (2.20)

$$U' = DU = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}.$$

Let U be a vector of coefficients of an interpolating polynomial $P_N(x)$. Then the vector U' such that the elements of U' are the coefficients of $dP_N(x)/dx$ can be easily calculated. This is done by obtaining the differentiation matrix D and using the formula $U' = DU$ [68]. A library of Matlab code for generating differentiation matrices can be found at <http://appliedmaths.sun.ac.za/~weideman/research/differ.html>. The theory for the functions in the library is provided in [72].

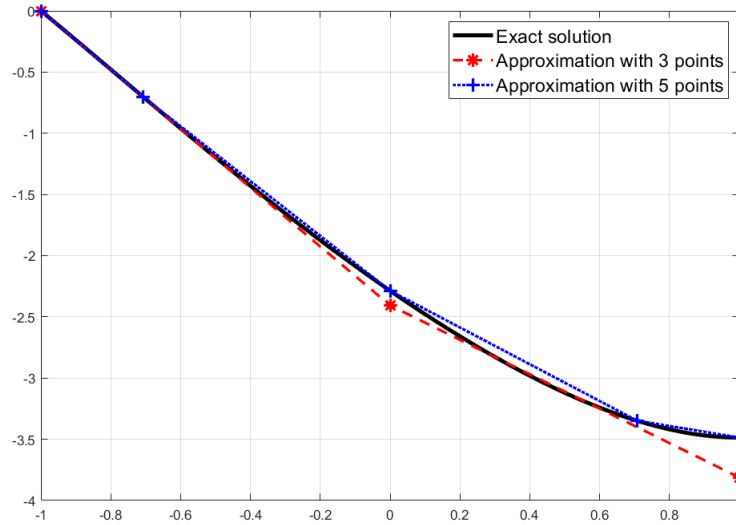


Figure 2.5: Solution of Poisson's equation where the forcing function is $f(x) = x^3 + e^x$ and boundary conditions are homogeneous Dirichlet on the left and homogeneous Neumann on the right. The exact solution is $u(x) = (x^5 - 5x - 4)/20 - e(x + 1) + \exp(x) - e^{-1}$. Solution is approximated with 3 and 5 points.

Example 2.4. For example, consider the Poisson equation

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= x^3 + e^x = f(x), \quad x \in [-1, 1], \\ u|_{x=-1} &= 0 = \frac{\partial u}{\partial x}|_{x=1}. \end{aligned}$$

An approximate solution U can be calculated

$$U = (D^2)^{-1} F$$

where F is the values of $f(x)$ at the Chebyshev interpolation points. Solutions with different amount of points can be seen in Figure 2.5.

The boundary conditions are incorporated into the differentiation matrices, this is done using the `cheb2dif` function found in the aforementioned Matlab library located at <http://appliedmaths.sun.ac.za/~weideman/research/differ.html>. The theory behind this approach is beyond the scope of this thesis but more information can be found in [72].

Example 2.5. We seek an approximate solution to the heat equation

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} + x^3 + e^x, \quad x \in [-1, 1], \quad t \in [0, 10], \\ u|_{x=-1} &= 0 = \frac{\partial u}{\partial x}|_{x=1}, \\ u|_{t=0} &= 0. \end{aligned}$$

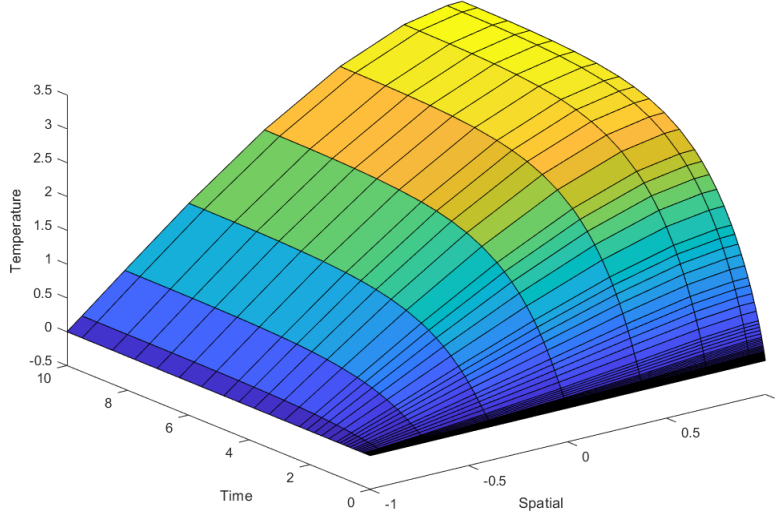


Figure 2.6: Approximate solution of the heat equation $\partial u/\partial t = \partial^2 u/\partial x^2 + x^3 + e^x$ for $t \in [0, 10]$ with a homogeneous Dirichlet boundary condition on the left and a homogeneous Neumann condition on the right and the temperature is identically zero as the zero condition. The number of interpolation points used is 9, these points are located at the extremas of the 8th degree Chebyshev polynomial of the first kind.

If U is a vector of coefficients of an interpolating polynomial then the ODE

$$\frac{\partial U}{\partial t} = D^2 U + F$$

produces an approximate solution at the Chebyshev interpolation points where D is the differentiation matrix and F is the values of $x^3 + e^x$ at the interpolation points. An approximate solution can be seen in Figure 2.6.

We can use the tools discussed in this subsection to discretize and approximately solve the system of PDE's which model the lithium ion cell discussed earlier in the chapter.

We approximate the solution of the lithium ion model defined by equations (2.15a)-(2.16e) using Chebyshev collocation outlined in this subsection. Discretize equations (2.15a)-(2.15d) and let C_e, C_s, Φ_e, Φ_s be the vectors of values of approximate solutions at the discrete points. We need to solve the following system of equations

$$\begin{bmatrix} \frac{d}{dt} C_e \\ \frac{d}{dt} C_s \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} D_{C_e}^2 & & & \\ & D_{C_s}^2 & & \\ & & D_{\Phi_e}^2 & \\ & & & D_{\Phi_s}^2 \end{bmatrix} \begin{bmatrix} C_e \\ C_s \\ \Phi_e \\ \Phi_s \end{bmatrix} + \begin{bmatrix} F_{C_e}(C_e, C_s, \Phi_e, \Phi_s) \\ F_{C_s}(C_e, C_s, \Phi_e, \Phi_s) \\ F_{\Phi_e}(C_e, C_s, \Phi_e, \Phi_s) \\ F_{\Phi_s}(C_e, C_s, \Phi_e, \Phi_s) \end{bmatrix}. \quad (2.21)$$

The matrices $D_{C_e}^2, D_{C_s}^2, D_{\Phi_e}^2, D_{\Phi_s}^2$ are appropriate differentiation matrices with boundary conditions incorporated as described earlier in this subsection (Matlab library located at

<http://appliedmaths.sun.ac.za/~weideman/research/differ.html>) and $F_{C_e}(\cdot)$, $F_{C_e}(\cdot)$, $F_{\Phi_e}(\cdot)$, $F_{\Phi_s}(\cdot)$ are forcing functions. An approximate solution of equation (2.21) can be found in Chapter 4.

Notice the absence of a time derivative for Φ_e and Φ_s , this means that the discretization is a system of differential-algebraic equations as opposed to the familiar system of ordinary differential equations. The contents of the next chapter include a discussion of existence and uniqueness theory for differential-algebraic equations as well as how one can numerically approximate a differential-algebraic equation.

In this chapter, the equations describing the electrochemical processes inside a lithium ion cell were presented and a method for simulating the equations was discussed.

| Symbol | Description | Value |
|------------------|--|--|
| Constants | | |
| R | Gas constant | $8.3145 \text{ JK}^{-1}\text{mol}^{-1}$ |
| F | Faraday's constant | $96485.33212 \text{ Cmol}^{-1}$ |
| b | Bruggeman coefficient | $\begin{cases} 4.1 & \text{in the anode} \\ 2.3 & \text{in the separator} \\ 1.5 & \text{in the cathode} \end{cases}$ |
| t_+^0 | Transference number | 0.435 |
| α | Charge transfer coefficient | 0.5 |
| k | Reaction rate constant | $\begin{cases} 1.764e - 11 \text{ m}^{2.5}\text{mol}^{-0.5}\text{s}^{-1} & \text{in the anode} \\ 6.667e - 11 \text{ m}^{2.5}\text{mol}^{-0.5}\text{s}^{-1} & \text{in the cathode} \end{cases}$ |
| Cell Geometry | | |
| L_- | Length of anode | $73.5e - 6 \text{ m}$ |
| L_{sep} | Length of separator region | $25.0e - 6 \text{ m}$ |
| L_+ | Length of cathode | $70.0e - 6 \text{ m}$ |
| Electrolyte | | |
| ϵ_e | Volume fraction of the electrolyte | $\begin{cases} 0.4382 & \text{in the anode} \\ 0.45 & \text{in the separator} \\ 0.3 & \text{in the cathode} \end{cases}$ |
| Solid Particles | | |
| R_s | Radius of solid particles | $\begin{cases} 12.5e - 6 \text{ m} & \text{in the anode} \\ 8.5e - 6 \text{ m} & \text{in the cathode} \end{cases}$ |
| D_s | Diffusivity in the solid particles | $\begin{cases} 5.5e - 14 \text{ m}^2\text{s}^{-1} & \text{in the anode} \\ 1.0e - 11 \text{ m}^2\text{s}^{-1} & \text{in the cathode} \end{cases}$ |
| ϵ_s | Volume fraction of solid particles | $\begin{cases} 0.0566 & \text{in the electrodes} \\ 0.5 & \text{in the separator} \end{cases}$ |
| a_s | Specific interfacial area of the electrode | $3\epsilon_s/R_s$ |
| c_s^{max} | Max concentration | $\begin{cases} 30555 \text{ mol m}^{-3} & \text{in the anode} \\ 51555 \text{ mol m}^{-3} & \text{in the cathode} \end{cases}$ |
| σ_s | Conductivity of solid particles | $\begin{cases} 100 \text{ Sm}^{-1} & \text{in the anode} \\ 10 \text{ Sm}^{-1} & \text{in the cathode} \end{cases}$ |
| σ_s^{eff} | Effective conductivity of solid particles | $\sigma_s \epsilon_s^b$ |

Table 2.1: Parameter values for the model of a lithium-ion cell.

Chapter 3

Differential Algebraic Equations

Differential-algebraic equations are a useful tool when it comes to modelling, they can arise when modelling, for example, conservation laws such as Kirchoff's law in electrical systems [40].

This chapter provides the basic theory of differential-algebraic equations (DAE's), only linear DAE's are considered in this chapter. At the end of the chapter a numerical method for solving DAE's is presented along with the associated theory.

A differential-algebraic equation combines differential equations with algebraic equations. The most general form of a differential-algebraic equation is

$$0 = F(t, x, \dot{x}) \tag{3.1}$$

where $F : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^m$.

Example 3.1. A simple example of a differential-algebraic equation is a description of a simplified Robertson problem

$$\begin{aligned} \dot{x}_1(t) &= -0.04x_1(t) + 10^4x_2(t), \\ 0 &= x_1(t) + x_2(t) - 1. \end{aligned}$$

where $x_1(t)$ and $x_2(t)$ are populations of species 1 and 2, respectively. This can be rewritten in linear form as:

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} -0.04 & 10^4 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix}.$$

3.1 Theory of Linear DAE's with Constant Coefficients

This subsection covers the basic existence-uniqueness theory for linear DAE's with constant coefficients. First some preliminaries are introduced then the main results are proven.

Definition 3.2. A *linear DAE with constant coefficients* is a differential-algebraic equation of the form

$$E\dot{x}(t) = Ax(t) + f(t) \tag{3.2}$$

where $E, A \in \mathbb{C}^{m \times n}$, $f : \mathbb{R} \rightarrow \mathbb{C}^m$ and E is singular. We also have an initial condition

$$x(t_0) = x_0. \quad (3.3)$$

The work in this subsection is mostly sourced from Differential-Algebraic Equations by Kunkel and Mehrmann [40, Section 2.1]. See there for more information.

Definition 3.3. Two pairs of matrices (E_1, A_1) , (E_2, A_2) are *strongly equivalent* if there exist nonsingular matrices $P \in \mathbb{C}^{m \times m}$, $Q \in \mathbb{C}^{n \times n}$ such that

$$E_2 = PE_1Q \quad A_2 = PA_1Q. \quad (3.4)$$

We can write $(E_1, A_1) \sim (E_2, A_2)$.

Example 3.4. Given square matrices A and B of equal dimensions, there exists a QZ decomposition that produces two unitary matrices Q and Z such that

$$\hat{A} = QAZ \quad \hat{B} = QBZ$$

where \hat{A} and \hat{B} are upper-triangular [22, Theorem 7.7.1]. Thus $(A, B) \sim (\hat{A}, \hat{B})$ and the QZ decomposition produces a strong equivalence.

Example 3.5. Consider the matrices

$$E_1 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \quad A_1 = \begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}.$$

Calculating the QZ decomposition of these matrices provides the following left and right non-singular transform matrices

$$Q = \begin{bmatrix} 0.1412 & 0.3441 & 0.5470 & 0.7499 \\ -0.8247 & -0.4261 & -0.0276 & 0.3710 \\ -0.5430 & 0.6708 & 0.2875 & -0.4153 \\ -0.0714 & 0.5000 & -0.7857 & 0.3571 \end{bmatrix} \quad Z = \begin{bmatrix} 0.3191 & -0.7734 & -0.5477 & 0 \\ 0.4294 & -0.3400 & 0.7303 & 0.4082 \\ 0.5397 & 0.0934 & 0.1826 & -0.8165 \\ 0.6500 & 0.5268 & -0.3651 & 0.4082 \end{bmatrix}$$

along with $E_2 = QE_1Z$ and $A_2 = QA_1Z$

$$E_2 = \begin{bmatrix} 38.2113 & -5.6140 & 8.6964e-15 & 1.5320e-14 \\ 0 & -2.0936 & 1.7946e-15 & 3.2593e-15 \\ 0 & 0 & 1.3190e-16 & -8.4178e-16 \\ 0 & 0 & 0 & 1.1598e-15 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 2.1749 & 1.3190 & -1.6392e-15 & -1.5331e-15 \\ 0 & 3.6783 & 6.3862e-16 & 1.3240e-15 \\ 0 & 0 & 5.7155 & -1.3041 \\ 0 & 0 & 0 & 4.1991 \end{bmatrix}$$

The purpose of strong equivalency is that it allows the transformation of the given system into a different (and simpler) form without distorting the original problem. The simpler form that we will be looking for is called a *Weierstrass canonical form* which will be introduced in Definition 3.9.

Lemma 3.6. [40, Lemma 2.2] *Strong equivalence is an equivalence relation, i.e. strong equivalence satisfies the following three properties:*

- $(E, A) \sim (E, A)$ (*reflexivity*),
- if $(E_1, A_1) \sim (E_2, A_2)$ then $(E_2, A_2) \sim (E_1, A_1)$ (*symmetry*),
- if $(E_1, A_1) \sim (E_2, A_2)$ and $(E_2, A_2) \sim (E_3, A_3)$ then $(E_1, A_1) \sim (E_3, A_3)$ (*transitivity*).

The relation introduced in Definition 3.3 is an equivalence relation.

Proof. • Reflexivity: Let P and Q in Definition 3.3 be equal to the identity matrices of appropriate sizes.

- Symmetry: Given $E_2 = PE_1Q$ and $A_2 = PA_1Q$ with non-singular P and Q then $E_1 = P^{-1}E_2Q^{-1}$ and $A_1 = P^{-1}A_2Q^{-1}$ implies $(E_2, A_2) \sim (E_1, A_1)$.
- Transitivity: Given $E_2 = P_1E_1Q_1$, $A_2 = P_1A_1Q_1$ and $E_3 = P_2E_2Q_2$, $A_3 = P_2A_2Q_2$ then $E_3 = P_2P_1E_1Q_1Q_2$ and $A_3 = P_2P_1A_1Q_1Q_2$ where P_2P_1 and Q_2Q_1 are non-singular implies $(E_3, A_3) \sim (E_1, A_1)$. □

Definition 3.7. Let $E, A \in \mathbb{C}^{m \times n}$. The matrix pair (E, A) is regular if $m = n$ and the characteristic polynomial $p(\lambda) = \det(\lambda E - A)$ is not the zero polynomial.

Lemma 3.8. [40, Lemma 2.6] *Every matrix pair that is strongly equivalent to a regular matrix pair is regular.*

Proof. Let $E_2 = PE_1Q$ and $A_2 = PA_1Q$ with non-singular P and Q , we can assume the matrices are square. Then consider the characteristic polynomial of (E_2, A_2)

$$\begin{aligned}
 p_2(\lambda) &= \det(\lambda E_2 - A_2) \\
 &= \det(\lambda PE_1Q - PA_1Q) \\
 &= \det P \det(\lambda E_1 - A_1) \det Q \\
 &= \det P p_1(\lambda) \det Q
 \end{aligned}$$

Since $\det P \neq 0$ and $\det Q \neq 0$ then $p_2(\lambda)$ is not the zero polynomial if and only if $p_1(\lambda)$ is not the zero polynomial. □

Definition 3.9. Let $E, A \in \mathbb{C}^{n \times n}$. If

$$E = \begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix}.$$

where J is a matrix in Jordan normal form and N is a nilpotent matrix then we say that (E, A) is in *Weierstrass canonical form*. It is allowed that either J or N is equal to zero.

Theorem 3.10. [40, Theorem 2.7] Let $E, A \in \mathbb{C}^{n \times n}$ and (E, A) be regular. Then

$$(E, A) \sim \left(\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix}, \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix} \right) \quad (3.5)$$

where J is a matrix in Jordan normal form and N is a nilpotent matrix also in Jordan normal form.

Proof. Since (E, A) is regular, there exists a $\lambda_0 \in \mathbb{C}$ with $\det(\lambda_0 E - A) \neq 0$ implying that $\lambda_0 E - A$ is nonsingular. Hence,

$$\begin{aligned} (E, A) &\sim (E, A - \lambda_0 E + \lambda_0 E) \\ &\sim ((A - \lambda_0 E)^{-1} E, I + \lambda_0 (A - \lambda_0 E)^{-1} E) \end{aligned}$$

Definition 3.3 with $P = (A - \lambda_0 E)$, $Q = I$. Next, there exists a non-singular matrix S such that $S^{-1}[(A - \lambda_0 E)^{-1} E]S = \tilde{J} = \text{diag}(J', N')$ where \tilde{J} is the Jordan normal form of $(A - \lambda_0 E)^{-1} E$, J' consists of Jordan blocks of nonzero eigenvalues and is non-singular and N' consists of Jordan blocks of zero eigenvalues and is nilpotent and strictly upper triangular. Furthermore,

$$S^{-1}[I + \lambda_0 (A - \lambda_0 E)^{-1} E]S = S^{-1}IS + \lambda_0 S^{-1}(A - \lambda_0 E)^{-1}ES = I + \lambda_0 \text{diag}(J', N').$$

Hence,

$$(E, A) \sim \left(\begin{bmatrix} J' & 0 \\ 0 & N' \end{bmatrix}, \begin{bmatrix} I + \lambda_0 J' & 0 \\ 0 & I + \lambda_0 N' \end{bmatrix} \right).$$

Next, left multiply by $\text{diag}(J'^{-1}, (I + \lambda_0 N')^{-1})$

$$(E, A) \sim \left(\begin{bmatrix} I & 0 \\ 0 & (I + \lambda_0 N')^{-1} N' \end{bmatrix}, \begin{bmatrix} J'^{-1} + \lambda_0 I & 0 \\ 0 & I \end{bmatrix} \right).$$

The inverse of an upper triangular matrix is upper triangular so $\tilde{J}^{-1} (I + \lambda_0 N')^{-1}$ is upper triangular. Upper triangular matrix times strictly upper triangular matrix is strictly upper triangular so $(I + \lambda_0 N')^{-1} N'$ is strictly upper triangular. Perform Jordan decomposition on both $J'^{-1} + \lambda_0 I$ and $(I + \lambda_0 N')^{-1} N'$ to produce their Jordan normal forms, J and N , respectively. J is non-singular and N is strictly upper triangular hence nilpotent. Finally,

$$(E, A) \sim \left(\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix}, \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix} \right),$$

as required. □

Definition 3.11. Consider a regular matrix pair (E, A) with a corresponding Weierstrass canonical form as in Definition 3.9. Let ν be the index of nilpotency of N , i.e. $N^\nu = 0$ and $N^{\nu-1} \neq 0$ or if $N = 0$ let $\nu = 1$. The *index* of (E, A) is ν , denoted by $\text{ind}(E, A) = \nu$.

Lemma 3.12. [40, Lemma 2.10] Suppose that the pair (E, A) has two Weierstrass canonical forms

$$(E, A) \sim \left(\begin{bmatrix} I & 0 \\ 0 & N_i \end{bmatrix}, \begin{bmatrix} J_i & 0 \\ 0 & I \end{bmatrix} \right), \quad i = 1, 2,$$

where d_i is the size of block J_i . Then $d_1 = d_2$ and, $N_1^\nu = 0$ and $N_1^{\nu-1} \neq 0$ if and only if $N_2^\nu = 0$ and $N_2^{\nu-1} \neq 0$.

Lemma 3.12 shows that the index of (E, A) does not depend on the transformation matrices so long as the transformation matrices are non-singular and do result in a Weierstrass canonical form.

Now, given (3.2) where (E, A) is regular there exist P and Q such that $(E, A) \sim (\text{diag}(I, N), \text{diag}(J, I))$. Let $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = Pf$ where f_1 has as many rows as J has columns. Then (3.2) is equivalent to

$$\dot{x}_1(t) = Jx_1(t) + f_1(t) \tag{3.6a}$$

$$N\dot{x}_2(t) = x_2(t) + f_2(t) \tag{3.6b}$$

where $Q \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x$ and x_1 has as many rows as J has columns.

Initial value problems such as (3.6a) are solvable for continuous f_1 and uniquely solvable if f_1 is Lipschitz. Let's consider (3.6b).

Lemma 3.13. Consider (3.6b) with $f_2 \in C^\nu(\mathbb{R}, \mathbb{C}^n)$, that is, f is ν times differentiable and maps a value in \mathbb{R} to a value \mathbb{C}^n . The index of nilpotency of N is ν . Then (3.6b) has the unique solution

$$x_2 = - \sum_{i=0}^{\nu-1} N^i f_2^{(i)} \tag{3.7}$$

Proof. Rewrite (3.6b) as $(I - Nd/dt)x_2 + f_2 = 0$ then

$$x_2 = - \left(I - N \frac{d}{dt} \right)^{-1} f_2 = - \sum_{i=0}^{\infty} \left(N \frac{d}{dt} \right)^i f_2 = - \sum_{i=0}^{\nu-1} N^i f_2^{(i)}$$

by the Maclaurin series of $(1 - x)^{-1}$. Substituting this into (3.6b)

$$\begin{aligned}
N\dot{x}_2 - x_2 - f_2 &= -\sum_{i=0}^{\nu-1} N^{i+1} f^{(i+1)} + \sum_{i=0}^{\nu-1} N^i f_2^{(i)} - f_2 \\
&= -\sum_{i=1}^{\nu-1} N^i f^{(i)} + N^0 f_2 + \sum_{i=1}^{\nu-1} N^i f_2^{(i)} - f_2 \\
&= -\sum_{i=1}^{\nu-1} N^i f^{(i)} + \sum_{i=1}^{\nu-1} N^i f_2^{(i)} + I f_2 - f_2 \\
&= 0 + 0 = 0.
\end{aligned}$$

Thus (3.7) is a solution of (3.6b). \square

This provides us with a complete solution, however notice that from Lemma 3.13 the solution x_2 is determined without an initial condition. This tells us that unlike linear ODEs, linear DAE's do not guarantee a solution for all initial conditions, the initial conditions need to satisfy the algebraic constraint. Also, x_2 is only continuously differentiable if $f^{\nu-1}$ is continuously differentiable.

Definition 3.14. Given a linear DAE

$$E\dot{x} = Ax + f(t)$$

and an initial condition x_0 . The initial condition is *consistent* if the initial value problem given by the linear DAE and initial condition has at least one solution.

Putting together the results leads to the following theorem.

Theorem 3.15. [40, Theorem 2.12] *Let the pair (E, A) of square matrices be regular and let P and Q be non-singular matrices that transform (3.2) and (3.3) to Weierstrass canonical form so that*

$$PEQ = \begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix}, \quad PAQ = \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix}, \quad Pf = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

and define

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = Q^{-1}x, \quad \begin{bmatrix} x_{0,1} \\ x_{0,2} \end{bmatrix} = Q^{-1}x_0.$$

Furthermore, assume $f \in C^\nu(\mathbb{R}, \mathbb{C}^n)$ where $\nu = \text{ind}(E, A)$. Then:

1. The differential-algebraic equation (3.2) is solvable.
2. An initial condition (3.3) is consistent if and only if

$$x_{0,2} = -\sum_{i=0}^{\nu-1} N^i f_2^{(i)}(t_0).$$

3. The set of consistent initial values x_0 is non-empty.

4. Every initial value problem with a consistent initial condition is uniquely solvable.

Theorem 3.16. [40, Theorem 2.14] Let $E, A \in \mathbb{C}^{n \times n}$, if (E, A) is not a regular matrix pair then the homogeneous initial value problem

$$E\dot{x} = Ax, \quad x(t_0) = 0$$

has a nontrivial solution.

Proof. Since (E, A) is not regular, $\text{rank}(\lambda E - A) < n$ for all $\lambda \in \mathbb{C}$. Let $\lambda_i, i = 1, \dots, n+1$ be pairwise different complex numbers. For every λ_i , there is a $v_i \in \mathbb{C}^n \setminus \{0\}$ where $(\lambda_i E - A)v_i = 0$ and v_i 's are linearly dependent. Hence, there exist complex numbers $\alpha_i, i = 1, \dots, n+1$ such that

$$\sum_{i=1}^{n+1} \alpha_i v_i = 0$$

where not all α_i 's are zero. Let

$$x(t) = \sum_{i=1}^{n+1} \alpha_i v_i \exp(\lambda_i(t - t_0))$$

then $x(t_0) = 0$ and

$$E\dot{x}(t) = \sum_{i=1}^{n+1} \alpha_i \lambda_i E v_i \exp(\lambda_i(t - t_0)) = \sum_{i=1}^{n+1} \alpha_i A v_i \exp(\lambda_i(t - t_0)) = Ax(t).$$

Since $x(t)$ is not the zero function we have a nonzero solution of the homogeneous value problem. \square

What if the initial condition is $x_0 \neq 0$ but consistent with the algebraic constraint? Then define $y = x - x_0$ and consider $E\dot{y} = Ay + Ax_0$ with $y(t_0) = 0$ which is equivalent to $E\dot{x} = Ax, x(t_0) = x_0$. If we linearize then we obtain $E\dot{y} = Ay$ with $y(t_0) = 0$ to which the above theorem applies.

Example 3.17. [40] Consider the differential-algebraic equation of the form $E\dot{x}(t) = Ax(t) + f(t)$ where

$$E = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad f = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}. \quad (3.8)$$

The characteristic polynomial $p(\lambda) = \det(\lambda E - A) = \lambda - \lambda^2$ is not the zero polynomial so (E, A) is regular. Additionally, assume that $f \in C^1(\mathbb{R}, \mathbb{R}^3)$. To put into Weierstrass canonical form use

$$P = I \quad Q = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and obtain

$$PEQ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad PAQ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad Pf = \begin{bmatrix} \tilde{f}_1 \\ \tilde{f}_2 \end{bmatrix}$$

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = Q^{-1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ x_1 \end{bmatrix}$$

where $\tilde{f}_1 = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$, $\tilde{f}_2 = f_3$, $\tilde{x}_1 = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}$ and $\tilde{x}_2 = x_1$. By Theorem 3.15, this DAE is solvable. The nilpotent matrix, N , has nilpotency index $\nu = 1$ and is equal to 0, so we have a consistent solution if $\tilde{x}_2 = -N^0 \tilde{f}_2$ (i.e. $x_1 = -f_3$). Thus if we have a consistent initial condition ($x_{0,1} = -f_3(t_0)$) then we have a unique solution.

Write equation (3.8) as a system of equations

$$\begin{aligned} \dot{x}_2 &= x_2 + f_1, \\ \dot{x}_3 &= f_2, \\ 0 &= x_1 + f_3. \end{aligned} \tag{3.9}$$

Clearly equation (3.9) has a unique solution if $x_1 = -f_3$, this is consistent with the statement provided by theorem 3.15.

In this thesis, we are interested in semi-explicit index-1 DAE's

Definition 3.18. A differential-algebraic equation is called *semi-explicit index-1* if it can be written in the form

$$\dot{x}_D = f_D(t, x_D, x_A), \tag{3.10a}$$

$$0 = f_A(t, x_D, x_A). \tag{3.10b}$$

where $f_D : \mathbb{R} \times \mathbb{R}^{n_D} \times \mathbb{R}^{n_A} \rightarrow \mathbb{R}^{n_D}$, $f_A : \mathbb{R} \times \mathbb{R}^{n_D} \times \mathbb{R}^{n_A} \rightarrow \mathbb{R}^{n_A}$. In this form, the derivative, \dot{x}_D , is isolated and the equations describing the differential and algebraic processes are separated. The subscripts D and A indicate differential and algebraic, respectively. x_D and x_A are referred to as the differential and algebraic states, respectively. Additionally, the Jacobian of the algebraic constraint evaluated with respect to the algebraic state is non-singular,

$$\det \left(\frac{\partial f_A}{\partial x_A} \right) \neq 0.$$

Semi-explicit index-1 DAE's are also known as Hessenberg index-1 DAE's.

Consider a semi-explicit index-1 DAE, since $\det(\partial f_A / \partial x_A) \neq 0$ then by the Implicit Function Theorem, given time t and differential state x_D , we can solve for the algebraic state, x_A .

In this section, we discussed existence-uniqueness theory for linear differential-algebraic equations and used Theorem 3.15 to show that a linear DAE has a unique solution. We also defined semi-explicit index-1 differential equations. Theory of nonlinear DAE's is beyond the scope of this thesis but can be found in [40].

3.2 Numerical Solution of DAE's using BDF

This section covers one numerical method for solving DAE's - backward differential formulae method. This method is the focus since it is suitable for *stiff* differential equations, indeed it's A-stable for orders 1 and 2 [29, Theorem 4.10], also it's available for use in Matlab as `ode15s`. A-stability means that no matter the step size that a method uses, the numerical solution of $\dot{x}(t) = kx(t)$ approaches 0 for $k < 0$. This is a desirable property for solving stiff equations.

Definition 3.19. A k -step linear multi-step method for the numerical solution of an ordinary differential equation $\dot{x} = f(t, x)$ is defined by

$$\sum_{l=0}^k \alpha_{k-l} x_{i-l} = h \sum_{l=0}^k \beta_{k-l} f(t_{i-l}, x_{i-l}) \quad (3.11)$$

where h is the step size, $\alpha_i, \beta_i \in \mathbb{R}$ are coefficients. We also assume that $\alpha_k \neq 0$.

Definition 3.20. Let the true solution of a system at time t_n be $x(t_n)$ and the numerical solution at the same time be x_n where h is the time step used by the numerical method. Suppose the initial conditions satisfy

$$\|x(t_0) - x_0\| \leq \tilde{C}h^p$$

for some positive integer p where \tilde{C} is a constant independent of h , $x(t_0)$ is the initial condition and x_0 is the initial state of the numerical solution. Then a numerical method is said to be *convergent of order p* if the error between the exact solution, $x(t_n)$, and the numerical solution, x_n , at time t_n can be bounded in the following way

$$\|x(t_n) - x_n\| \leq Ch^p$$

where C is a constant independent of h .

Definition 3.21. A k -step linear multi-step method is a *backward differentiation formulae (BDF) method* if

$$\beta_0 = \dots = \beta_{k-1} = 0, \quad \beta_k = 1$$

and α 's are found in Table 3.1.

Note: The 1-step BDF method is the backward Euler method.

Theorem 3.22. [23] A k -step BDF method is convergent if and only if $1 \leq k \leq 6$.

We now apply the BDF on a linear DAE.

Theorem 3.23. [40, Theorem 5.24] If (E, A) is regular with $\text{ind}(E, A) = 1$ then the k -step BDF methods with $1 \leq k \leq 6$ applied to the system $E\dot{x}(t) = Ax(t) + f(t)$, $x(t_0) = x_0$ are convergent of order k .

| α_{k-l} | $l = 0$ | $l = 1$ | $l = 2$ | $l = 3$ | $l = 4$ | $l = 5$ | $l = 6$ |
|----------------|---------|---------|---------|---------|---------|---------|---------|
| $k = 1$ | 1 | -1 | | | | | |
| $k = 2$ | 3/2 | -2 | 1/2 | | | | |
| $k = 3$ | 11/6 | -3 | 3/2 | -1/3 | | | |
| $k = 4$ | 25/12 | -4 | 3 | -4/3 | 1/4 | | |
| $k = 5$ | 137/60 | -5 | 5 | -10/3 | 5/4 | -1/5 | |
| $k = 6$ | 147/60 | -6 | 15/2 | -20/3 | 15/4 | -6/5 | 1/6 |

Table 3.1: Values of α for k -step BDF methods where $1 \leq k \leq 6$
[40, Table 5.3]

Proof. Applying the BDF method to $E\dot{x} = Ax + f(t)$ produces

$$E \frac{1}{h} \sum_{l=0}^k \alpha_{k-l} x_{i-l} = Ax_i + f(t_i).$$

We know that the system decouples into two parts, given by the Weierstrass canonical form.

$$\frac{1}{h} \sum_{l=0}^k \alpha_{k-l} x_{1,i-l} = Jx_{1,i} + f_1(t_i) \quad (3.12a)$$

$$N \frac{1}{h} \sum_{l=0}^k \alpha_{k-l} x_{2,i-l} = x_{2,i} + f_2(t_i) \quad (3.12b)$$

The theory for the differential part (3.12a) is beyond the scope of this thesis but can be found in numerical methods textbooks such as [29], it is convergent of order k . We focus on the algebraic part (3.12b). By assumption, the matrix N has degree of nilpotency equal to 1 so N is the zero matrix. Thus the numerical solution to the algebraic part at time t_i is

$$x_{2,i} = -f_2(t_i).$$

Since (3.12a) is convergent of order k and (3.12b) can be solved exactly then the BDF applied to $E\dot{x}(t) = Ax(t) + f(t)$ is convergent of order k . □

This thesis provides a proof only for the case that the matrix pair (E, A) is index 1, although a proof is possible for higher index however that is beyond the scope of this thesis, find the proof in [40].

Given a semi-explicit index-1 DAE

$$\begin{aligned} \dot{x}_D &= f_D(t, x_D, x_A), \\ 0 &= f_A(t, x_D, x_A). \end{aligned}$$

Introduce a small value ϵ

$$\begin{aligned}\dot{x}_D &= f_D(t, x_D, x_A), \\ \epsilon \dot{x}_A &= f_A(t, x_D, x_A).\end{aligned}$$

Apply a k -step linear multi-step method (3.11) to obtain

$$\begin{aligned}\sum_{l=0}^k a_{k-l} x_{D,i-l} &= h \sum_{l=0}^k b_{k-l} f_D(t_{i-l}, x_{D,i-l}, x_{A,i-l}), \\ \epsilon \sum_{l=0}^k a_{k-l} x_{A,i-l} &= h \sum_{l=0}^k b_{k-l} f_A(t_{i-l}, x_{D,i-l}, x_{A,i-l}).\end{aligned}$$

Now set $\epsilon = 0$

$$\sum_{l=0}^k a_{k-l} x_{D,i-l} = h \sum_{l=0}^k b_{k-l} f_D(t_{i-l}, x_{D,i-l}, x_{A,i-l}), \quad (3.16a)$$

$$0 = h \sum_{l=0}^k b_{k-l} f_A(t_{i-l}, x_{D,i-l}, x_{A,i-l}). \quad (3.16b)$$

Equation (3.16) defines the *direct approach* for solving semi-explicit index-1 DAE using a k -step linear multi-step method. If we replace equation (3.16b) with the algebraic constraint

$$\sum_{l=0}^k a_{k-l} x_{D,i-l} = h \sum_{l=0}^k b_{k-l} f_D(t_{i-l}, x_{D,i-l}, x_{A,i-l}), \quad (3.17a)$$

$$0 = f_A(t_i, x_{D,i}, x_{A,i}). \quad (3.17b)$$

then equation (3.17) defines the *indirect approach*. When solving DAE's Matlab uses the direct approach except when solving semi-explicit problems in Simulink [59].

We now have a method for numerically solving differential-algebraic equations. To use a BDF method in MATLAB choose `ode15s` as your ODE/DAE solver and set "BDF" to "on" in the options structure. Maximum order can be controlled with "MaxOrder", acceptable values are 1 through 5. Other solver choices are `ode23t` and `ode15i`, `ode15s` and `ode23t` can be used only for semilinear DAE's (DAE's of the form $E\dot{x} = f(t, x)$ where E may be singular) and `ode15i` can be used to solve DAE's of the general form $F(t, x, \dot{x}) = 0$.

For more information on solving index-1 DAE's in Matlab see [59], [Solve Differential Algebraic Equations \(DAEs\)](#), [Solve Semilinear DAE System](#) and [Solve DAEs Using Mass Matrix Solvers](#).

Example 3.24. Consider a semi-explicit index-1 DAE

$$\begin{aligned}\dot{x}_D &= f_D(t, x_D, x_A), \\ 0 &= f_A(t, x_D, x_A).\end{aligned}$$

where $x_D \in \mathbb{R}^{n_D}$ and $x_A \in \mathbb{R}^{n_A}$. If we want to approximately solve this equation in Matlab for $t \in [0, 1]$ given a consistent initial condition $x_0 = [x_{D,0} \ x_{A,0}]^T$, then define a mass matrix E

$$E = \begin{bmatrix} I_{n_D \times n_D} & 0_{n_D \times n_A} \\ 0_{n_A \times n_D} & 0_{n_A \times n_A} \end{bmatrix}$$

and let

$$\text{opts} = \text{odeset}('Mass', E)$$

and combine the differential and algebraic equations into one function

$$F(t, x) = \begin{bmatrix} f_D(t, x_D, x_A) \\ f_A(t, x_D, x_A) \end{bmatrix} \text{ where } x = \begin{bmatrix} x_D \\ x_A \end{bmatrix}.$$

Use `ode15s` to numerically solve the initial value problem

$$[\mathbf{t}, \mathbf{x}] = \text{ode15s}(@(\mathbf{t}, \mathbf{x}) F(\mathbf{t}, \mathbf{x}), [0 \ 1], \mathbf{x}_0, \text{opts}).$$

The i th row in \mathbf{x} contains approximate solution at time equal to the i th element in \mathbf{t} .

Example 3.25. Consider equation (3.8) in Example 3.17, let

$$f(t) = \begin{bmatrix} 0 \\ \frac{1}{2}t \sin(t) \\ \cos(t) \end{bmatrix}, \quad x_0 = \begin{bmatrix} -1 \\ 0.0001 \\ 1 \end{bmatrix}.$$

Notice that $x_{0,1} = -f_3(0)$, so we have a consistent initial condition. We can use Matlab's `ode15s` function to numerically solve this DAE. An approximate solution is plotted in Figure 3.1.

In this chapter, theory and numerics for linear DAE's were discussed. The theory provided a way to check for existence and uniqueness of linear differential-algebraic equations and introduced the concept of consistent initial conditions. An example of approximate solution to a simple DAE was presented. Numerics for the lithium ion model discussed in the previous chapter are in the next chapter.

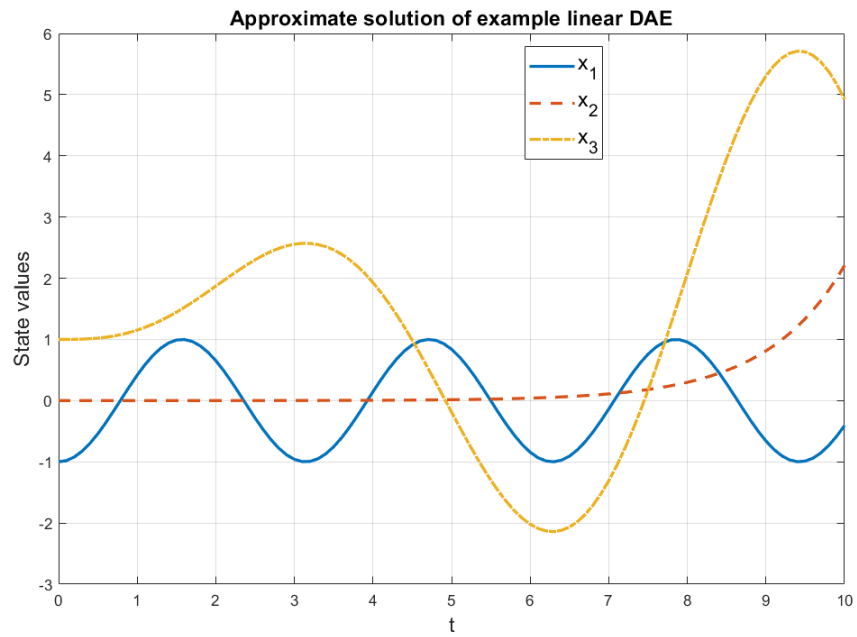


Figure 3.1: Approximate solution of equation (3.8) and initial condition defined in Example 3.25. The equation is solved numerically in the time domain $t \in [0, 10]$ using `ode15s` in Matlab.

Chapter 4

Kalman Filtering of ODE's and DAE's

State estimation methods for ordinary differential equations have been well studied. The Kalman filter (KF), first introduced in [37], is widely used and is an optimal estimator for linear systems under the assumption that we seek to minimize the mean square error and that the system noise is Gaussian. Extended Kalman filter (EKF) and the Unscented Kalman filter (UKF) are common filtering techniques for nonlinear systems. EKF uses the linearization of the system whereas UKF uses sample points to predict the mean and covariance. The Kalman filter is named after Rudolf Emil Kálmán.

The Kalman filter works in two phases, a predict phase in which the KF produces an estimate of the state variables along with their uncertainty matrix. In the next phase, the update phase, the estimate is corrected using a weighted average with more weight given to estimates and measurements with lower uncertainty.

To apply KF, which has been developed for linear systems, on nonlinear systems one can linearize the state transition and measurement functions around the current state estimate. This is the approach used in EKF. EKF only works well for nonlinear systems that are 'almost' linear [31].

Instead of linearizing the system, UKF estimates the state mean and covariance using sample points. The sample points, called sigma points, are systematically generated around the mean with deviation based on the covariance and propagated through the nonlinear functions from which new mean and covariance is achieved. UKF does not discard the nonlinearities of the system. Instead of propagating a single estimate like EKF, UKF propagates all sigma points, the number of which grows linearly with the size of the system. This can become very computationally expensive.

The Kalman filter has applications in biology [28, 43, 47], chemistry [14], GPS [4, 35], meteorology [27, 26, 58], neurology [73], stock market [11] and others [5, 48].

The next section describes the Kalman filter algorithm for a linear system, the section after provides two Kalman filter algorithms for nonlinear systems. The third section provides an extension of the two Kalman filter algorithms to semi-explicit index-1 DAE systems.

4.1 Estimation of Linear Systems

4.1.1 Least Squares Estimation

Before considering the Kalman filter it is useful to discuss how to estimate a constant vector given linear but noisy measurements. A portion of the contents of this subsection are referenced from [62, Section 3.1].

Suppose x is a constant but unknown vector of length n and y is a noisy measurement vector of length m where each element of y is a linear combination of the elements of x plus noise. Defining H to be the measurement matrix, we have:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} H_{11}x_1 + \dots + H_{1n}x_n \\ \vdots \\ H_{m1}x_1 + \dots + H_{mn}x_n \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix},$$
$$y = Hx + v.$$

Define \hat{x} to be the estimate of x and let ϵ_y be the measurement residual between the measurement and $H\hat{x}$ vectors:

$$\epsilon_y = y - H\hat{x}.$$

It has been shown that the most probable value of x is the value of \hat{x} that minimizes the sum of squares between y and $H\hat{x}$ [62]. Equivalently, the best estimate \hat{x} minimizes

$$\begin{aligned} J &= \epsilon_{y1}^2 + \dots + \epsilon_{ym}^2 \\ &= \epsilon_y^T \epsilon_y. \end{aligned} \tag{4.1}$$

Since $\epsilon_y = y - H\hat{x}$,

$$\begin{aligned} J &= (y - H\hat{x})^T (y - H\hat{x}) \\ &= y^T y - \hat{x}^T H^T y - y^T H \hat{x} + \hat{x}^T H^T H \hat{x}. \end{aligned}$$

To find the minimum calculate the derivative of J with respect to \hat{x} and set it to zero,

$$\frac{\partial J}{\partial \hat{x}} = -y^T H - y^T H + 2\hat{x}^T H^T H = 0.$$

If we assume that the nullspace of H is trivial then $H^T H$ is positive definite. This leads to

$$\begin{aligned} H^T y &= H^T H \hat{x} \\ \hat{x} &= (H^T H)^{-1} H^T y. \end{aligned} \tag{4.2}$$

The second derivative of J is

$$\frac{\partial^2 J}{\partial \hat{x}^2} = 2H^T H$$

which is positive definite thus (4.2) is the unique minimum of (4.1).

In this subsection, it is shown how to calculate the best estimate given a noisy measurement.

4.1.2 Weighted Least Squares Estimation

Suppose we again have the problem of finding an estimate given a noisy measurement but the covariance of the noise may be different for each element of y . A portion of the contents of this subsection are referenced from [62, Section 3.2]. Given

$$y = Hx + \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix},$$

$$\mathbb{E}(v_i^2) = \sigma_i^2, \quad i = 1, \dots, m.$$

Assume that the noises are zero-mean and independent. The covariance matrix is

$$R = \mathbb{E}(vv^T) = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_m \end{bmatrix}.$$

The measurement covariance matrix R is assumed to be positive definite. Enforce that by requiring that R is a diagonal matrix with positive diagonal entries.

Now we seek to minimize the weighted sum of squares of the elements of ϵ_y

$$J = \frac{\epsilon_{y1}^2}{\sigma_1^2} + \dots + \frac{\epsilon_{ym}^2}{\sigma_m^2}. \quad (4.3)$$

The reason for minimizing the weighted sum of squares is that it allows us to put more trust into the less noisy variables. If, for example, σ_1 is relatively large then y_1 is a relatively noisy measurement thus minimizing ϵ_{y1} is not as valuable as minimizing the other elements of ϵ_y . Rewriting equation (4.3),

$$\begin{aligned} J &= \epsilon_y^T R^{-1} \epsilon_y \\ &= (y - H\hat{x})^T R^{-1} (y - H\hat{x}) \\ &= y^T R^{-1} y - \hat{x}^T H^T R^{-1} y - y^T R^{-1} \hat{H} x + \hat{x}^T H^T R^{-1} H \hat{x}. \end{aligned} \quad (4.4)$$

If we assume that the nullspace of H is the trivial nullspace and combine this with the assumption that R is a diagonal matrix with positive diagonal entries then $H^T R^{-1} H$ is positive definite. Let a be a vector and $b = Ha$, consider

$$\begin{aligned} a^T H^T R^{-1} H a &= (Ha)^T R^{-1} H a \\ &= b^T R^{-1} b. \end{aligned}$$

Since the nullspace of H is trivial then $a = 0 \iff b = 0$. Since R is positive definite then R^{-1} is positive definite, so if $b \neq 0$ then $b^T R^{-1} b > 0$. Thus if $a \neq 0$ then $a^T H^T R^{-1} H a > 0$ which is the definition of positive definiteness.

Take the derivative of equation (4.4), set it equal to zero:

$$\begin{aligned}\frac{\partial J}{\partial \hat{x}} &= -2y^T R^{-1}H + 2\hat{x}^T H^T R^{-1}H = 0, \\ y^T R^{-1}H &= \hat{x}^T H^T R^{-1}H, \\ H^T R^{-1}y &= H^T R^{-1}H\hat{x}, \\ \hat{x} &= (H^T R^{-1}H)^{-1} H^T R^{-1}y.\end{aligned}$$

The second derivative of J is

$$\frac{\partial^2 J}{\partial \hat{x}^2} = 2H^T R^{-1}H. \quad (4.5)$$

This is positive definite thus $\hat{x} = (H^T R^{-1}H)^{-1} H^T R^{-1}y$ is the unique minimum of (4.3).

4.1.3 Recursive Estimation

Consider the situation where we are provided measurements y_k in discrete-time and we update the estimate whenever a new measurement arrives. The vector which we are trying to estimate is constant; an evolving estimation target is considered in the next subsection. We can use (4.5) to calculate the new estimate, however there is a more efficient to obtain a new estimate. When a new measurement arrives we obtain a new estimate by updating the latest estimate. This is recursive estimation. A portion of the contents of this subsection are referenced from [62, Section 3.3].

A linear recursive estimator has the form

$$y_k = H_k x + v_k, \quad (4.6a)$$

$$\hat{x}_k = \hat{x}_{k-1} + K_k (y_k - H_k \hat{x}_{k-1}). \quad (4.6b)$$

The measurement y_k is provided according to (4.6a) where H_k is known but x_k and v_k are unknown. A new estimate \hat{x}_k is calculated by (4.6b). The matrix K_k is to be determined, it is called the estimator gain matrix or the Kalman gain matrix and $(y_k - H_k \hat{x}_{k-1})$ is called the correction term. We assume that $\mathbb{E}(v_k) = 0 \forall k$ where $\mathbb{E}(\cdot)$ is the expected value.

Before considering the gain matrix let us examine the mean of the estimation error of the linear recursive estimator. The estimation error is $\epsilon_{x,k} = x - \hat{x}_k$, the mean of which is:

$$\begin{aligned}\mathbb{E}(\epsilon_{x,k}) &= \mathbb{E}(x - \hat{x}_k) \\ &= \mathbb{E}(x - \hat{x}_{k-1} + K_k (y_k - H_k \hat{x}_{k-1})) \\ &= \mathbb{E}(\epsilon_{x,k-1} + K_k (H_k x + v_k - H_k \hat{x}_{k-1})) \\ &= \mathbb{E}(\epsilon_{x,k-1} + K_k H_k (x - \hat{x}_{k-1}) + K_k v_k) \\ &= \mathbb{E}(\epsilon_{x,k-1} + K_k H_k \epsilon_{x,k-1} + K_k v_k) \\ &= \mathbb{E}(\epsilon_{x,k-1}) + K_k H_k \mathbb{E}(\epsilon_{x,k-1}) + K_k \mathbb{E}(v_k) \\ &= (I - K_k H_k) \mathbb{E}(\epsilon_{x,k-1}) + K_k \mathbb{E}(v_k)\end{aligned}$$

If $\mathbb{E}(\epsilon_{x,k-1}) = 0$ and $\mathbb{E}(v_k) = 0$ then $\mathbb{E}(\epsilon_{x,k}) = 0$, that is if the initial estimate is equal to the expected value of x ($\hat{x}_0 = \mathbb{E}(x)$) then the expected value of the estimate will always be equal to the expected value of true value. If this is satisfied then (4.6) is called an unbiased estimator since the expected estimate will be equal to the true value. Notice that an unbiased estimator will be unbiased for any gain matrix K_k .

Let us turn our attention to the cost function. We will seek to minimize the expected value of the square of the Euclidean norm between the estimate and the true value, if R is a diagonal matrix then this is the same cost function as used in the previous subsection,

$$\begin{aligned}
J_k &= \mathbb{E}(\|x - \hat{x}_k\|_2^2) \\
&= \mathbb{E}((x_1 - \hat{x}_{k,1})^2 + \dots + (x_n - \hat{x}_{k,n})^2) \\
&= \mathbb{E}(\epsilon_{x_1,k}^2 + \dots + \epsilon_{x_n,k}^2) \\
&= \mathbb{E}(\epsilon_{x,k}^T \epsilon_{x,k}) \\
&= \mathbb{E}(\text{Tr}(\epsilon_{x,k} \epsilon_{x,k}^T)) \\
&= \text{Tr}(\mathbb{E}(\epsilon_{x,k} \epsilon_{x,k}^T)) \\
&= \text{Tr}(P_k)
\end{aligned} \tag{4.7}$$

where $P_k = \mathbb{E}(\epsilon_{x,k}^T \epsilon_{x,k})$ is the covariance of the estimation error. We desire a recursive formula for calculating P_k :

$$\begin{aligned}
P_k &= \mathbb{E}(\epsilon_{k,n} \epsilon_{k,n}^T) \\
&= \mathbb{E}\left(\left((I - K_k H_k) \epsilon_{x,k-1} - K_k v_k\right) \left((I - K_k H_k) \epsilon_{x,k-1} - K_k v_k\right)^T\right) \\
&= (I - K_k H_k) \mathbb{E}(\epsilon_{x,k-1} \epsilon_{x,k-1}^T) (I - K_k H_k)^T \\
&\quad - K_k \mathbb{E}(v_k \epsilon_{x,k-1}^T) (I - K_k H_k)^T - (I - K_k H_k) \mathbb{E}(\epsilon_{x,k-1} v_k^T) K_k^T \\
&\quad + K_k \mathbb{E}(v_k v_k^T) K_k^T
\end{aligned} \tag{4.8}$$

The measurement at time k is not incorporated into the estimate at time $k - 1$ so the estimation error at time $k - 1$ is independent of the measurement noise at time k . Thus

$$\begin{aligned}
\mathbb{E}(v_k \epsilon_{x,k-1}^T) &= \mathbb{E}(v_k) \mathbb{E}(\epsilon_{x,k-1}^T) \\
&= 0,
\end{aligned}$$

since both are zero-mean, the second and thirds terms in (4.8) are zero. Thus,

$$P_k = (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T \tag{4.9}$$

where $R_k = \mathbb{E}(v_k v_k^T)$ is the covariance of the measurement noise at time k . This establishes the recursive formula for the covariance of the error, notice that if P_{k-1} and R_k are symmetric and positive definite (as is required) then P_k will be positive definite.

Let us now find a gain matrix K_k that minimizes the trace of the covariance (4.7). Substitute (4.9) into (4.7) and take the derivative, note that $\partial \text{Tr}(ABA^T) / \partial A = 2AB$ if B is symmetric:

$$\frac{\partial J_k}{\partial K_k} = 2(I - K_k H_k) P_{k-1} (-H_k^T) + 2K_k R_k. \tag{4.10}$$

Set (4.10) to zero:

$$\begin{aligned}
K_k R_k &= (I - K_k R_k) P_{k-1} H_k^T \\
K_k R_k &= P_{k-1} H_k^T - K_k R_k P_{k-1} H_k^T \\
K_k R_k + K_k R_k P_{k-1} H_k^T &= P_{k-1} H_k^T \\
K_K (R_k + H_k P_{k-1} H_k^T) &= P_{k-1} H_k^T \\
K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1}
\end{aligned} \tag{4.11}$$

The second derivative is

$$\begin{aligned}
\frac{\partial^2 J_k}{\partial K_K^2} &= 2(-H_k^T) P_{k-1}^T (-H_k^T) + 2R_k \\
&= 2H_k^T P_{k-1}^T H_k^T + 2R_k.
\end{aligned}$$

With the assumptions that P_{k-1} and R_k are positive definite the second derivative is positive definite thus (4.11) is the unique minimum.

Let $S_k = (H_k P_{k-1} H_k^T + R_k)$ and substitute (4.11) into (4.9),

$$\begin{aligned}
P_k &= (I - P_{k-1} H_k^T S_k^{-1} H_k) P_{k-1} (I - P_{k-1} H_k^T S_k^{-1} H_k)^T + K_k R_k K_k^T, \\
&= P_{k-1} - P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} - P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} \\
&\quad + P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1} H_k^T S_k^{-1} R_k S_k^{-1} H_k P_{k-1}.
\end{aligned}$$

Notice the middle of the fourth term contains $H_k P_{k-1} H_k^T$ and the middle of the fifth term contains R_k , the other variables in the last two terms are the same.

$$\begin{aligned}
P_k &= P_{k-1} - 2P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1} H_k^T S_k^{-1} R_k S_k^{-1} H_k P_{k-1}, \\
&= P_{k-1} - 2P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1} H_k^T S_k^{-1} H_k P_{k-1}, \\
&= P_{k-1} - P_{k-1} H_k^T S_k^{-1} H_k P_{k-1}, \\
&= P_{k-1} - K_k H_k P_{k-1} \text{ by (4.11)}, \\
&= (I - K_k H_k) P_{k-1}.
\end{aligned} \tag{4.12}$$

It is common to see the correction of the covariance written as (4.12).

4.1.4 Propagating Mean and Error

Before discussing the Kalman filter, there is still one more matter to address. Previously, we discussed how to filter noise given noisy measurements assuming the true state is constant, in this subsection, we assume that the true state is subject to a noisy process. A portion of the contents of this subsection are referenced from [62, Section 4.1].

Given a linear discrete-time system:

$$x_k = A_k x_{k-1} + w_k \tag{4.13}$$

where w_k is zero-mean Gaussian noise with covariance matrix Q_k .

The expected value of (4.13) is

$$\begin{aligned}\mathbb{E}(x_k) &= \hat{x}_k \\ &= A_k \hat{x}_{k-1}.\end{aligned}\tag{4.14}$$

So the expected value is propagated by applying the linear transformation.

To obtain the formula for propagating the covariance, take the outer product of $x_k - \hat{x}_k$ with itself:

$$\begin{aligned}(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T &= (A_k x_{k-1} + w_k - \hat{x}_k)(A_k x_{k-1} + w_k - \hat{x}_k)^T \\ &= (A_k(x_{k-1} - \hat{x}_{k-1}) + w_k)(A_k(x_{k-1} - \hat{x}_{k-1}) + w_k)^T \\ &= A_k(x_{k-1} - \hat{x}_{k-1})(x_{k-1} - \hat{x}_{k-1})^T A_k^T \\ &\quad + A_k(x_{k-1} - \hat{x}_{k-1})w_k^T + w_k(x_{k-1} - \hat{x}_{k-1})^T A_k^T + w_k w_k^T\end{aligned}$$

Now take the expected value of both sides

$$\begin{aligned}\mathbb{E}((x_k - \hat{x}_k)(x_k - \hat{x}_k)^T) &= \mathbb{E}(A_k(x_{k-1} - \hat{x}_{k-1})(x_{k-1} - \hat{x}_{k-1})^T A_k^T) \\ &\quad + \mathbb{E}(A_k(x_{k-1} - \hat{x}_{k-1})w_k^T) + \mathbb{E}(w_k(x_{k-1} - \hat{x}_{k-1})^T A_k^T) + \mathbb{E}(w_k w_k^T)\end{aligned}\tag{4.15}$$

As in the previous subsection, the measurement at time k is not incorporated into the estimate at time $k-1$ so the estimation error at time $k-1$ is independent of the measurement noise at time k , so

$$\begin{aligned}\mathbb{E}((x_{k-1} - \hat{x}_{k-1})w_k^T) &= \mathbb{E}(x_{k-1} - \hat{x}_{k-1})\mathbb{E}(w_k^T) \\ &= 0,\end{aligned}$$

Define P_k as in the previous subsection, the covariance of the estimation error. Thus (4.15) is equivalent to

$$\begin{aligned}P_k &= A_k \mathbb{E}((x_{k-1} - \hat{x}_{k-1})(x_{k-1} - \hat{x}_{k-1})^T) A_k^T + Q_k \\ P_k &= A_k P_{k-1} A_k^T + Q_k.\end{aligned}\tag{4.16}$$

In the case that A and Q are constant, the steady state covariance of the error, if it exists, may be of interest, the formula for which is

$$P = A P A^T + Q.\tag{4.17}$$

This is known as the discrete Lyapunov equation. If all eigenvalues of A are less than 1 in magnitude and Q is symmetric and positive (semi-)definite then the solution to (4.17) exists and is unique, symmetric and positive (semi-)definite [62, Theorem 21]. To solve the discrete Lyapunov equation (4.17) in Matlab use [dlyap](#).

4.1.5 Kalman Filter for Linear Systems

The linear discrete-time Kalman filter combines the material in the above two sections to produce an algorithm that is optimal for filtering noise. This subsection presents the Kalman filter algorithm for a linear discrete-time system. There exists a Kalman filter which can be applied on linear continuous-time systems and hybrid systems. A system is called hybrid if the process is continuous and the measurements are available at discrete times, these systems are also known as continuous-discrete. It should be noted that a system with a continuous process may be transformed into a discrete system. Given

$$\dot{x} = Ax, \quad x(0) = x_{initial}$$

then a discrete-time system is defined as

$$x_k = \exp(h_k A) x_{k-1}, \quad x_0 = x_{initial},$$

where the time between x_k and x_{k-1} is $h = t_k - t_{k-1}$.

Given a linear discrete-time system:

$$\begin{aligned} x_k &= A_k x_{k-1} + w_k \\ y_k &= H_k x_k + v_k \end{aligned}$$

where $x_k \in \mathbb{R}^n$ is the state at time k , $A_k \in \mathbb{R}^{n \times n}$ is the process matrix at time k , $y_k \in \mathbb{R}^m$ is the measurement at time k , $H_k \in \mathbb{R}^{m \times n}$ is the measurement matrix at time k . The noises, w_k and v_k , are white, zero-mean, uncorrelated and have known covariance matrices Q_k and R_k

$$\begin{aligned} w_k &\sim N(0, Q_k) \\ v_k &\sim N(0, R_k) \\ \mathbb{E}[w_k w_j^T] &= \begin{cases} Q_k & k = j \\ 0 & k \neq j \end{cases} \\ \mathbb{E}[v_k v_j^T] &= \begin{cases} R_k & k = j \\ 0 & k \neq j \end{cases} \\ \mathbb{E}[w_k v_j^T] &= 0. \end{aligned}$$

The discrete-time Kalman filter can be applied on the above system to filter out the noise and provide an optimal estimate of the true state. After initialization, the algorithm progresses one time step at a time, using only the updated estimates from the previous step, along with the given information about the system. Each time step has two phases, the predict phase and the update phase.

The predicted (pre-updated) estimates are indicated by a minus sign in the superscript and the updated estimates are indicated by a plus sign in the superscript.

Below is the Kalman filter algorithm:

Initialize:

$$\begin{aligned}\hat{x}_0^+ &= \mathbb{E}(x_0) \\ P_0^+ &= \mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T].\end{aligned}$$

1. Predict phase:

(a) Predict state estimate, (4.14),

$$\hat{x}_k^- = A_k \hat{x}_{k-1}^+$$

(b) Predict estimate covariance, (4.16),

$$P_k^- = A_k P_{k-1}^+ A_k^T + Q_k$$

2. Update phase:

(a) Obtain the optimal Kalman gain, (4.11),

$$K_k = P_k^- H_{k-1}^T (H_{k-1} P_k^- H_{k-1}^T + R_k)^{-1}$$

(b) Update state estimate, (4.6b),

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-)$$

(c) Update estimate covariance, (4.12),

$$P_k^+ = (I - K_k H_k) P_k^-$$

The Kalman filter is applied recursively to produce state estimates x_1^+ , x_2^+ , \dots at times $t = 1, 2, \dots$. The outputs of the Kalman filter can be tuned by changing the covariance matrices Q and R . If the measurement is to be trusted more than the model then the norm $\|R\|$ should be lower than $\|Q\|$ and vice versa, the greater the trust in the measurement over the model the greater the ratio $\|Q\|/\|R\|$.

The formula for P_k^+ may be replaced by the so-called Joseph stabilized version

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T.$$

This is the original equation for covariance derived in (4.9). The Joseph stabilized version is more robust and stable [62, Section 5.1] and guarantees that despite numerical inaccuracies P_k^+ will symmetric and positive-definite as long as P_k^- and R_k are. For more information see [62, Sections 5.1, 5.4, 9.1].

4.2 Kalman Filters for Nonlinear Systems

The algorithm presented in the previous section is only applicable on linear systems. However, many systems are nonlinear. This section discusses two generalizations of the standard Kalman filter: the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). Other such generalizations exist such as Particle Filter/Sequential Monte Carlo, Ensemble Kalman Filter and Iterative Extended Kalman Filter.

4.2.1 Extended Kalman Filter

The Extended Kalman filter is a generalization of the KF to nonlinear systems. The EKF requires the Jacobian matrices of the nonlinear process and measurement functions with respect to the state and noise to be available. The idea of the EKF is to linearize at each time step and treat the system as a time-varying linear system.

Suppose we have a non-linear discrete time system:

$$\begin{aligned}x_k &= f(t_k, x_{k-1}, w_k), \\y_k &= h(t_k, x_k, v_k), \\w_k &\sim N(0, Q_k), \\v_k &\sim N(0, R_k).\end{aligned}$$

The noises, w_k and v_k , are allowed to act nonlinearly on the process and measurement. Below is the Extended Kalman filter algorithm:

Initialize:

$$\begin{aligned}\hat{x}_0^+ &= \mathbb{E}(x_0) \\P_0^+ &= \mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T].\end{aligned}$$

1. Predict phase:

(a) Predict state estimate,

$$\hat{x}_k^- = f(t_k, \hat{x}_{k-1}^+, 0).$$

(b) Compute the Jacobians to linearize the process,

$$A_k = \left. \frac{\partial f}{\partial x} \right|_{(t_k, x_{k-1}^+, 0)} \quad \text{and} \quad L_k = \left. \frac{\partial f}{\partial w} \right|_{(t_k, x_{k-1}^+, 0)}.$$

(c) Predict estimate covariance,

$$P_k^- = A_k P_{k-1}^+ A_k^T + L_k Q_k L_k^T.$$

2. Update phase:

(a) Compute the Jacobians to linearize the measurement,

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{(t_k, x_k^-, 0)} \quad \text{and} \quad M_k = \left. \frac{\partial h}{\partial w} \right|_{(t_k, x_k^-, 0)}.$$

(b) Obtain the optimal Kalman gain,

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1}.$$

(c) Update state estimate,

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - h(t_k, \hat{x}_k^-, 0)).$$

(d) Update estimate covariance,

$$P_k^+ = (I - K_k H_k) P_k^-.$$

The EKF uses a first-order approximation to obtain a linear estimate to the system. This is a clear limitation and as it may fail to provide a good estimate when the system is highly nonlinear. Higher order approximations exist and can be used to obtain better estimates. This is the idea behind the Iterated EKF and the Second-Order EKF, a discussion of these is beyond the scope of this thesis but for more information see [62, Section 13.3].

4.2.2 Unscented Kalman Filter

This subsection discusses a different Kalman filter-based approach to estimating nonlinear systems. The Unscented Kalman Filter [31] does not linearize or approximate the nonlinear system, instead it aims to capture the mean and covariance of the nonlinear system through the use of unscented transformations. An unscented transform is a function that estimates the result of applying a (non-linear) function on a probability density function given a finite set of samples [31, 62], for more information see [62, Section 14.2]. The UKF was intended to be more accurate than the EKF and indeed the use of UKF can provide improvement over the EKF [62]. The cost of using UKF over EKF is greater computation time.

Unscented transformation works on two principles: (1) it is easy to carry out a nonlinear transformation on a single point and (2) it is easy to generate a set of sample points in state space whose sample probability distribution function approximates the true probability distribution function.

Given the mean \bar{x} and covariance P of a vector x , generate a set of sample points called sigma points whose mean and covariance equals mean \bar{x} and covariance P , respectively. Next, apply the nonlinear function $y = f(x)$ to each of the sigma points to obtain the transformed vectors. The effectiveness of the unscented transformation relies on the idea that the mean and covariance of the transformed vectors will be a good approximation of the true mean and covariance. The increased computational cost of the UKF comes from computing many points through the nonlinear transformation.

Suppose we have n -state nonlinear discrete time system

$$\begin{aligned}x_k &= f(t_k, x_{k-1}) + w_k, \\y_k &= h(t_k, x_k) + v_k, \\w_k &\sim N(0, Q_k), \\v_k &\sim N(0, R_k).\end{aligned}$$

Notice that here we assume the noise acts linearly on the system. Below is the Unscented Kalman filter algorithm:

Initialize:

$$\begin{aligned}\hat{x}_0^+ &= \mathbb{E}(x_0) \\P_0^+ &= \mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]\end{aligned}$$

1. Predict phase:

(a) Generate $2n + 1$ sigma points, $\sigma_{k-1,i}$ around \hat{x}_{k-1}^+ ,

$$\begin{aligned}\sigma_{k,0}^- &= \hat{x}_{k-1}^+, \\ \sigma_{k,i}^- &= \hat{x}_{k-1}^+ + \left(\sqrt{(n + \kappa)P_{k-1}^+} \right)_i, \quad i = 1, \dots, n, \\ \sigma_{k,n+i}^- &= \hat{x}_{k-1}^+ - \left(\sqrt{(n + \kappa)P_{k-1}^+} \right)_i, \quad i = 1, \dots, n,\end{aligned}$$

and $2n + 1$ weights,

$$W_0 = \frac{\kappa}{n + \kappa}, \quad W_i = \frac{1}{2(n + \kappa)}, \quad i = 1, \dots, 2n + 1,$$

where $\left(\sqrt{(n + \kappa)P_{k-1}^+} \right)_i$ is the i -th column of $\sqrt{(n + \kappa)P_{k-1}^+}$ and κ is a tuning parameter.

(b) Propagate the sigma points,

$$\sigma_{k,i}^+ = f(t_k, \sigma_{k,i}^-), \quad i = 0, \dots, 2n.$$

(c) Combine the sigma points in a weighted average to predict state estimate,

$$\hat{x}_k^- = \sum_{i=0}^{2n} W_i \sigma_{k,i}^+.$$

(d) Predict estimate covariance,

$$P_k^- = Q_k + \sum_{i=0}^{2n} W_i (\sigma_{k,i}^+ - \hat{x}_k^-) (\sigma_{k,i}^+ - \hat{x}_k^-)^T.$$

2. Update phase:

(a) Generate $2n + 1$ sigma points, $\tau_{k,i}^-$ around \hat{x}_k^- ,

$$\begin{aligned}\tau_{k,0}^- &= \hat{x}_k^-, \\ \tau_{k,i}^- &= \hat{x}_k^- + \left(\sqrt{(n + \kappa)P_{k-1}^+} \right)_i, \quad i = 1, \dots, n \\ \tau_{k,n+i}^- &= \hat{x}_k^- - \left(\sqrt{(n + \kappa)P_{k-1}^+} \right)_i, \quad i = 1, \dots, n\end{aligned}$$

(b) Transform the sigma points into predicted estimates,

$$\tau_{k,i}^+ = h(t_k, \tau_{k,i}^-), \quad i = 0, \dots, 2n.$$

(c) Predict measurement at time t_k ,

$$\hat{y}_k = \sum_{i=0}^{2n} W_i \tau_{k,i}^+.$$

(d) Estimate the covariance of the predicted measurement,

$$P_y = R_k + \sum_{i=0}^{2n} W_i (\tau_{k,i}^+ - \hat{y}_k) (\tau_{k,i}^+ - \hat{y}_k)^T.$$

(e) Estimate the cross covariance between \hat{x}_k^- and \hat{y}_k ,

$$P_{xy} = \sum_{i=0}^{2n} W_i (\tau_{k,i}^- - \hat{x}_k^-) (\tau_{k,i}^+ - \hat{y}_k).$$

(f) Obtain the optimal Kalman gain,

$$K_k = P_{xy} P_y^{-1}.$$

(g) Update state estimate,

$$x_k^+ = x_k^- + K_k (y_k - y_k^-).$$

(h) Update estimate covariance,

$$P_k^+ = P_k^- - K_k P_y K_k^T.$$

The tuning parameter κ can be used to vary the estimates, any value of κ is allowable as long as $n + \kappa \neq 0$. It is possible reduce higher-order errors in the mean and covariance approximation by adjusting κ . For example, if x is Gaussian then $\kappa = 3 - n$ will reduce fourth-order error terms in the approximation of the mean and covariance [30, 32].

| Symbol | Description | Value |
|------------------------|---|-------------------------|
| ρ_0 | Air density at sea level | 1.225 kg/m ³ |
| g | Acceleration due to gravity | 9.8 m/s ² |
| k | Constant describing the relation between altitude and air density | 20 000 m |
| b | Ballistic coefficient | 0.1 m ² /kg |
| $\mathbb{E}[w_1^2(t)]$ | Process covariance of position | 0 |
| $\mathbb{E}[w_2^2(t)]$ | Process covariance of velocity | 0 |
| a | Altitude of angle-measuring device | 200 000 m |
| M | Horizontal distance between angle-measuring device and falling mass | 50 000 m |

Table 4.1: Parameter values for falling mass example.

Example 4.1. Consider a mass falling towards the surface of the Earth and suppose we are trying to estimate its vertical position and velocity. Suppose we have a device which can measure the angle between the mass and the horizontal. The device is at altitude a and the horizontal distance between the device and the falling mass is M . The equations and parameters used here are:

$$\begin{aligned}\dot{x}_1 &= x_2 + w_1 \\ \dot{x}_2 &= -\frac{\rho_0 b}{2} \exp(-x_1/k) x_2 |x_2| - g + w_2 \\ y(t_k) &= \arctan\left(\frac{x_1 - a}{M}\right) + v_k\end{aligned}$$

The parameter values can be found in Table 4.1. The measurement is obtained every 0.5 seconds. The term $-\frac{\rho_0 b}{2} \exp(-x_1/k) x_2 |x_2|$ accounts for drag due to air resistance.

The initial conditions of the true state are

$$x_0 = [300\ 000 \quad -300]^T$$

and the Kalman filters are initialized with

$$\hat{x}_0^+ = [0 \quad 0]^T, \quad P_0^+ = \begin{bmatrix} 500\ 000 & 0 \\ 0 & 0.02 \end{bmatrix}, \quad \kappa = 0.5.$$

The results are displayed in Figures 4.1 and 4.2. Figure 4.1 shows the true state along with its estimates and Figure 4.2 shows the covariances. The UKF performs better, this is clear in both figures. It is closer to the true states and its covariance for position and velocity are smaller at the end of the simulation. At the end of the simulation, the position covariance of UKF is 5.75×10^3 , which is large but the true position at that time is 6.96×10^4 . The position covariance of EKF is 1.28×10^5 , significantly larger. However, the computation time of EKF was 0.3366 seconds and 1.1252 seconds for UKF, this is 3.3 times longer.

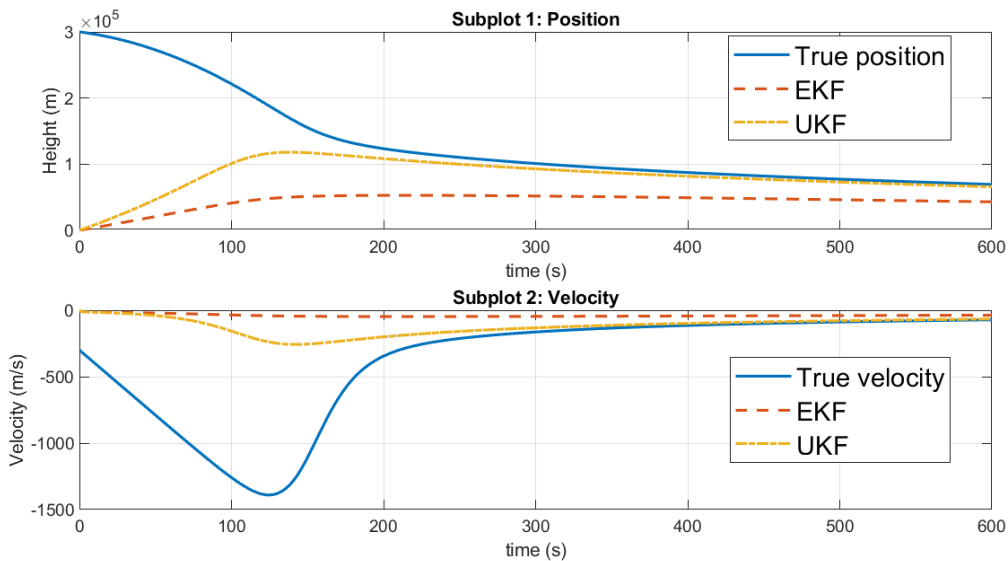


Figure 4.1: The true state and its estimates of free falling mass. The initial position is 3×10^5 m and the initial velocity is -300 m/s.

Subplot 1: the true position of the falling mass (blue solid line), the EKF estimate of the position (red dashed line) and the UKF estimate of the estimate (yellow dash-dot line).

Subplot 2: the true velocity of the falling mass (blue solid line), the EKF estimate of the velocity (red dashed line) and the UKF estimate of the velocity (yellow dash-dot line).

After 250 seconds, the UKF estimate closely approximates the true position and velocity whereas there's a larger error in the EKF estimate of the position.

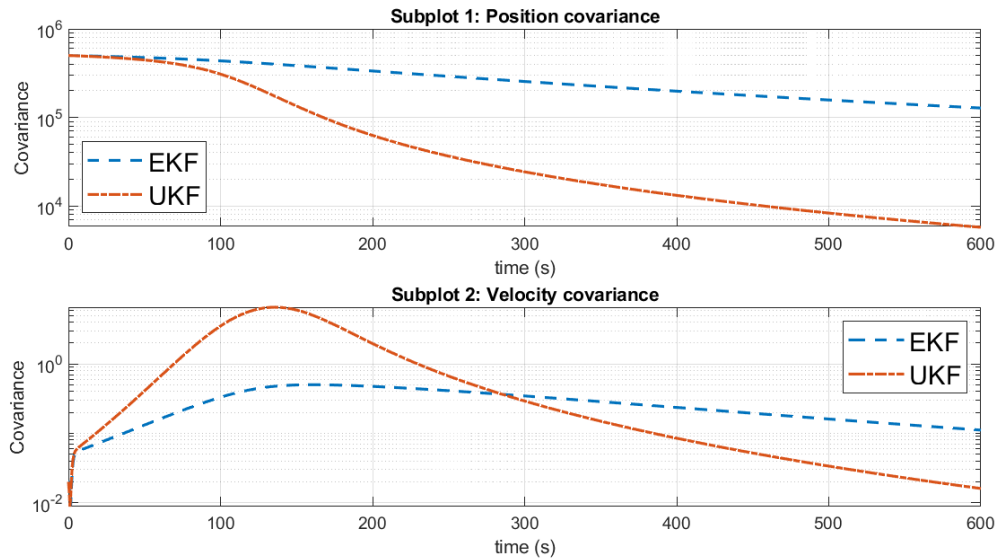


Figure 4.2: The covariances of the estimates of free falling mass.

Subplot 1: the covariance of the EKF position estimate (blue dashed line) and the covariance of the UKF position estimate (red dash-dot line).

Subplot 2: the covariance of the EKF velocity estimate (blue dashed line) and the covariance of the UKF velocity estimate (red dash-dot line).

The UKF covariances are significantly lower than the EKF covariances by the end of the simulation, however the UKF velocity covariance has a hump before both the UKF position and velocity covariances drop significantly below the EKF covariances.

4.3 Kalman Filters for semi-explicit index-1 DAE

It is possible to convert a DAE into an ODE and apply a well known state estimation method. There are three problems with this approach [39]:

1. Converting a DAE into an ODE and simulating it using an ODE solver can introduce significant errors.
2. The updated estimates produced by the state estimator will not necessarily satisfy the algebraic constraints.
3. If the DAE is reduced to a model containing only the differential states then measurements which are dependent on algebraic states may not be easily used.

This section explores semi-explicit index-1 DAE-compatible Kalman filters. A reminder of the of definition of semi-explicit index-1 DAE's is presented followed by a discussion on the filters presented in [39] then some examples.

A semi-explicit index-1 DAE-compatible EKF was introduced in [7] but this filter is limited in that it can only process systems where no algebraic states are measured. This filter was expanded upon in [39], allowing measurement of algebraic variables. Also introduced in [39] is a formulation for semi-explicit index-1 DAE-compatible UKF. The two filters introduced in [39] are the focus of this section.

Recall: a semi-explicit index-1 DAE has the form

$$\dot{x}_D = f_D(t, x_D, x_A), \quad (4.18a)$$

$$0 = f_A(t, x_D, x_A), \quad (4.18b)$$

where

$$\det \left(\frac{\partial f_A}{\partial x_A} \right) \neq 0.$$

Differentiate equation (4.18b) with respect to t

$$\begin{aligned} 0 &= \frac{\partial f_A}{\partial t} + \frac{\partial f_A}{\partial x_D} \dot{x}_D + \frac{\partial f_A}{\partial x_A} \dot{x}_A \\ \dot{x}_A &= - \left(\frac{\partial f_A}{\partial x_A} \right)^{-1} \left(\frac{\partial f_A}{\partial t} + \frac{\partial f_A}{\partial x_D} \dot{x}_D \right). \end{aligned} \quad (4.19)$$

This is the formula for transforming a semi-explicit index-1 DAE into an ODE. This allows numerical methods for ODE's to be applied on a DAE, however, reasons for not doing this can be found above.

Some define the index of a DAE as the number of differential operations needed to transform a DAE into an ODE, specifically this is referred to as the differential index, see [15, 16, 40]. This is consistent with the index-1 in semi-explicit index-1 DAE as it can be transformed into an ODE with one differential operation as done in equation (4.19).

4.3.1 EKF for semi-explicit index-1 DAE

The EKF presented here was first introduced in [39] which is a modification of the EKF introduced in [7]. The EKF in [7] is semi-explicit index-1 DAE-compatible however it is not compatible with systems where the measurement function is dependent on an algebraic state.

This EKF propagates the state estimate using a DAE solver and propagates the covariance by converting the DAE into an ODE and linearizing about the state estimate, then the matrix exponential of the Jacobian is obtained and the formula used is

$$P_{k+1|k} = AP_{k|k}A^T + Q$$

where P is the covariance, A is the matrix exponential of the Jacobian and Q is the process noise. The covariance has dimensions equal to the amount of the differential states.

The EKF introduced in [39] is able to handle semi-explicit index-1 DAE's where the measurement depends on algebraic states. The covariance has dimensions equal to the amount of differential states plus the amount of algebraic states.

Consider a semi-explicit index-1 DAE with noise and measurements taken at discrete intervals

$$\dot{x}_D = f_D(t, x_D, x_A) + w \quad (4.20a)$$

$$0 = f_A(t, x_D, x_A) \quad (4.20b)$$

$$y_k = h(t_k, x_{D,k}, x_{A,k}) + v_k \quad (4.20c)$$

where w and v_k are Gaussian noise with covariance matrices Q and R respectively. Notice that here we assume that noise is injected linearly into the system as opposed to the system used in the EKF.

Linearize (4.20a) and (4.20b) to obtain

$$\dot{x}_D = \begin{bmatrix} A_k & B_k \end{bmatrix} \begin{bmatrix} x_D \\ x_A \end{bmatrix} = A_k x_D + B_k x_A \quad (4.21a)$$

$$0 = \begin{bmatrix} C_k & D_k \end{bmatrix} \begin{bmatrix} x_D \\ x_A \end{bmatrix} = C_k x_D + D_k x_A \quad (4.21b)$$

where

$$\begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} = \begin{bmatrix} \frac{\partial f_D}{\partial x_A}(t_k, x_{D,k}, x_{A,k}) & \frac{\partial f_D}{\partial x_A}(t_k, x_{D,k}, x_{A,k}) \\ \frac{\partial f_A}{\partial x_D}(t_k, x_{D,k}, x_{A,k}) & \frac{\partial f_A}{\partial x_A}(t_k, x_{D,k}, x_{A,k}) \end{bmatrix}.$$

Differentiate the linearized algebraic constraint (4.21b)

$$\begin{aligned} 0 &= C_k \dot{x}_D + D_k \dot{x}_A \\ \dot{x}_A &= -D_k^{-1} C_k \dot{x}_D \\ \dot{x}_A &= -D_k^{-1} C_k (A_k x_D + B_k x_A) \\ \dot{x}_A &= -D_k^{-1} C_k A_k x_D - D_k^{-1} C_k B_k x_A \end{aligned}$$

Writing this together with the linearized differential part (4.21a)

$$\begin{bmatrix} \dot{x}_D \\ \dot{x}_A \end{bmatrix} = \begin{bmatrix} A_k & B_k \\ -D_k^{-1}C_k A_k & -D_k^{-1}C_k B_k \end{bmatrix} \begin{bmatrix} x_D \\ x_A \end{bmatrix}.$$

Let

$$L_k = \begin{bmatrix} A_k & B_k \\ -D_k^{-1}C_k A_k & -D_k^{-1}C_k B_k \end{bmatrix}$$

and define the transition matrix as

$$\phi_k = \exp(\Delta t_k L_k)$$

where Δt_k is the size of a time step from t_{k-1} to t_k , i.e. $\Delta t_k = t_k - t_{k-1}$.

Given state estimates and covariance $\hat{x}_{D,k-1|k-1}$, $\hat{x}_{A,k-1|k-1}$, $P_{k-1|k-1} \in \mathbb{R}^{(n_D+n_A) \times (n_D+n_A)}$, we wish to progress the estimate from time t_{k-1} to t_k . At time t_k , measurement y_k is available. Below is the process for progressing according to EKF method introduced in [39, Section 4].

1. Predict phase:

- (a) Propagate the state estimate from time t_{k-1} to t_k with $\hat{x}_{D,k-1|k-1}$ and $\hat{x}_{A,k-1|k-1}$ as the initial conditions using a DAE solver such as Matlab's ode15s. Obtain $\hat{x}_{D,k|k-1}$ and $\hat{x}_{A,k|k-1}$.
- (b) Obtain new covariance,

$$P_{k|k-1} = \phi_k P_{k-1|k-1} \phi_k^T + \begin{bmatrix} I_{n_D \times n_D} \\ -D_k^{-1}C_k \end{bmatrix} Q \begin{bmatrix} I_{n_D \times n_D} \\ -D_k^{-1}C_k \end{bmatrix}^T$$

where I is an identity matrix and an element of $\mathbb{R}^{n_D \times n_D}$.

2. Update phase:

- (a) Compute the Kalman gain,

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R)^{-1}$$

where $H_k = \begin{bmatrix} \frac{\partial h}{\partial x_D}(t_k, x_{D,k|k-1}, x_{A,k|k-1}) & \frac{\partial h}{\partial x_A}(t_k, x_{D,k|k-1}, x_{A,k|k-1}) \end{bmatrix}$ is the linearized measurement at time t_k .

- (b) Update the state estimate,

$$\begin{bmatrix} \hat{x}_{D,k|k} \\ \hat{x}_{A,k|k} \end{bmatrix} = \begin{bmatrix} \hat{x}_{D,k|k-1} \\ \hat{x}_{A,k|k-1} \end{bmatrix} + K_k (y_k - h(t_k, \hat{x}_{D,k|k-1}, \hat{x}_{A,k|k-1})).$$

- (c) Solve

$$0 = f_A(t_k, \hat{x}_{D,k|k}, \hat{x}_{A,k|k})$$

for $\hat{x}_{A,k|k}$ to obtain a consistent updated algebraic state.

- (d) Calculate the updated covariance,

$$P_{k|k} = (I_{(n_D+n_A) \times (n_D+n_A)} - K_k H_k) P_{k|k-1}.$$

We now have the next state estimates $x_{D,k|k}$ and $x_{A,k|k}$ and covariance matrix $P_{k|k}$.

4.3.2 UKF for semi-explicit index-1 DAE

Given state estimates and covariance $\hat{x}_{D,k-1|k-1}$, $\hat{x}_{A,k-1|k-1}$, $P_{k-1|k-1} \in \mathbb{R}^{n_D \times n_D}$, we wish to progress the estimate from time t_{k-1} to t_k . At time t_k , measurement y_k is available. Below is the process for progressing according to UKF method introduced in [39, Section 5]. Notice that here the covariance matrix is of dimension $n_D \times n_D$ whereas it was $(n_D + n_A) \times (n_D + n_A)$ for EKF.

1. Predict phase:

- (a) Choose $2n_D + 1$ sigma points, $\sigma_{k-1|k-1,i}^D$ around $\hat{x}_{D,k-1|k-1}$

$$\begin{aligned}\sigma_{k-1|k-1,0}^D &= \hat{x}_{D,k-1|k-1}, \\ \sigma_{k-1|k-1,i}^D &= \hat{x}_{D,k-1|k-1} + \left(\sqrt{(n + \kappa)P_{k-1|k-1}} \right)_i, \quad i = 1, \dots, n_D \\ \sigma_{k-1|k-1,n_D+i}^D &= \hat{x}_{D,k-1|k-1} - \left(\sqrt{(n + \kappa)P_{k-1|k-1}} \right)_i, \quad i = 1, \dots, n_D\end{aligned}$$

with associated weights

$$W_0 = \frac{\kappa}{n + \kappa}, \quad W_i = \frac{1}{2(n + \kappa)}, \quad i = 1, \dots, 2n_D + 1$$

and where $\left(\sqrt{(n + \kappa)P_{k-1|k-1}} \right)_i$ is the i -th column of $\sqrt{(n + \kappa)P_{k-1|k-1}}$ and κ is a tuning parameter.

- (b) Solve $0 = f_A(t_{k-1}, \sigma_{k-1|k-1,i}^D, \sigma_{k-1|k-1,i}^A)$ for $\sigma_{k-1|k-1,i}^A$, $i = 0, \dots, 2n_D$. Obtain the sigma points for the algebraic state.
- (c) Propagate the sigma points from time t_{k-1} to t_k using a DAE solver such as Matlab's ode15s. Obtain $\sigma_{k|k-1,i}^D$ and $\sigma_{k|k-1,i}^A$ for $i = 0, \dots, 2n_D$.
- (d) Predict the differential state,

$$\hat{x}_{k|k-1}^D = \sum_{i=0}^{2n_D} W_i \sigma_{k|k-1,i}^D.$$

- (e) Get covariance,

$$P_{k|k-1} = Q + \sum_{i=0}^{2n_D} W_i \left(\sigma_{k|k-1,i}^D - \hat{x}_{k|k-1} \right) \left(\sigma_{k|k-1,i}^D - \hat{x}_{k|k-1} \right)^T.$$

- (f) Repeat step 1a with $\hat{x}_{k|k-1}^D$ and $P_{k|k-1}$ instead of $\hat{x}_{D,k-1|k-1}$ and $P_{k-1|k-1}$. Obtain $\sigma_{k-1|k-1,i}^D$ for $i = 0, \dots, 2n_D$.
- (g) Re-solve $0 = f_A(t_{k-1}, \sigma_{k-1|k-1,i}^D, \sigma_{k-1|k-1,i}^A)$ for $\sigma_{k-1|k-1,i}^A$, $i = 0, \dots, 2n_D$. Obtain the sigma points for the algebraic state.

2. Update phase:

(a) Predict differential and algebraic state states,

$$\begin{bmatrix} \hat{x}_{D,k|k-1} \\ \hat{x}_{A,k|k-1} \end{bmatrix} = \sum_{i=0}^{2n_D} W_i \begin{bmatrix} \sigma_{k|k-1,i}^D \\ \sigma_{k|k-1,i}^A \end{bmatrix}.$$

(b) Calculate the observation of the sigma points, the mean, the covariance and the cross-covariance with the state sigma points.

$$\begin{aligned} \sigma_{k,i}^Y &= h(t_k, \sigma_{k|k-1,i}^D, \sigma_{k|k-1,i}^A) \quad i = 0, \dots, 2n_D \\ \hat{y}_k &= \sum_{i=0}^{2n_D} W_i \sigma_{k,i}^Y \\ S_k &= R + \sum_{i=0}^{2n_D} W_i (\sigma_{k,i}^Y - \hat{y}_k) (\sigma_{k,i}^Y - \hat{y}_k)^T \\ C_k &= \sum_{i=0}^{2n_D} W_i (\sigma_{k|k-1,i}^D - \hat{x}_{k|k-1}) (\sigma_{k|k-1,i}^Y - \hat{y}_k)^T \end{aligned}$$

(c) Compute the Kalman gain,

$$K_k = C_k S_k^{-1}$$

and let K_k^D be the first n_D rows of K_k .

(d) Compute the corrected differential states,

$$\hat{x}_{D,k|k} = \hat{x}_{D,k|k-1} + K_k (y_k - \hat{y}_k).$$

(e) Compute the corrected algebraic states by solving $0 = f_A(t_k, \hat{x}_{D,k|k}, \hat{x}_{A,k|k})$ for $\hat{x}_{A,k|k}$.

(f) Correct the covariance matrix of the differential states,

$$P_{k|k} = P_{k|k-1} - K_k^D S_k (K_k^D)^T.$$

We now have the next state estimates $x_{D,k|k}$ and $x_{A,k|k}$ and covariance matrix $P_{k|k}$. An DAE-compatible Ensemble Kalman filter can be found here [53] and a DAE-compatible Iterative Extended Kalman filter can be found here [54].

4.4 Numerical Example - Chemical Reactor

This example taken from [40, Example 1.4] with a modification on the dynamics of the temperature, add periodic heating. Consider a model of a chemical reactor

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{c} \\ \dot{T} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} k_1(c_a - c) - r \\ k_1(T_a - T) + k_2r - k_3(T - T_C) + 10(\sin(0.1\pi t) + 1) \\ r - k_3 \exp(-k_4/T)c \end{bmatrix}$$

where c is concentration, T is temperature and r is reaction rate per unit volume and $c_0, T_0, T_C, k_1, k_2, k_3, k_4$ are constants.

Rewrite the model to split the differential and algebraic parts,

$$\begin{aligned} \begin{bmatrix} \dot{c} \\ \dot{T} \end{bmatrix} &= \begin{bmatrix} k_1(c_a - c) - r \\ k_1(T_a - T) + k_2r - k_3(T - T_C) + 10(\sin(0.1\pi t) + 1) \end{bmatrix} = f_D(t, [c, T]^T, r) \\ 0 &= r - k_3 \exp(-k_4/T)c = f_A(t, [c, T]^T, r). \end{aligned} \quad (4.22)$$

Here the differential states are c and T and the only algebraic state is r . To determine if this DAE is index-1 check if the Jacobian of the algebraic constraint with respect to the algebraic state is non-singular. The Jacobian $\partial f_A / \partial r = 1$, so this DAE is index-1 and DAE-compatible Kalman filters are applicable.

Example 4.2. Given the DAE (4.22), with constants

$$\begin{aligned} c_a &= 100, \\ T_a &= 2, \quad T_C = 5, \\ k_1 &= 0.2, \quad k_2 = 1, \quad k_3 = 0.25, \quad k_4 = 10, \end{aligned}$$

and initial conditions

$$\begin{aligned} c_0 &= 200 \\ T_0 &= 10 \\ r_0 &= k_3 \exp(-k_4/T_0)c_0 \approx 18.39 \end{aligned}$$

notice this initial condition is consistent with the algebraic constraint. Suppose we can only measure the temperature T , set the measurement covariance to 0.1 and let the UKF tuning parameter, κ , be equal to 1. Thus,

$$\begin{aligned} C &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \\ R &= 0.1, \\ \kappa &= 1. \end{aligned}$$

Measurements are available every 5 seconds. The initial state will be the zero vector:

$$\hat{x}_{0|0} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \text{ (consistent with algebraic constraint).}$$

In addition to using the index-1 DAE-compatible EKF and UKF, we convert this DAE to an ODE by changing the algebraic constraint into a differential equation as in equation (4.19). Apply the common ODE-compatible EKF and UKF as defined in Section 4.2 on this ODE. The four filters used here require the process covariances and initial covariances to be defined with different dimensions. For the filters applied on ODE we have:

$$Q = \text{diag}(0.01, 0.01, 0.01), \quad P_{0|0} = \text{diag}(0.01, 0.01, 0.01).$$

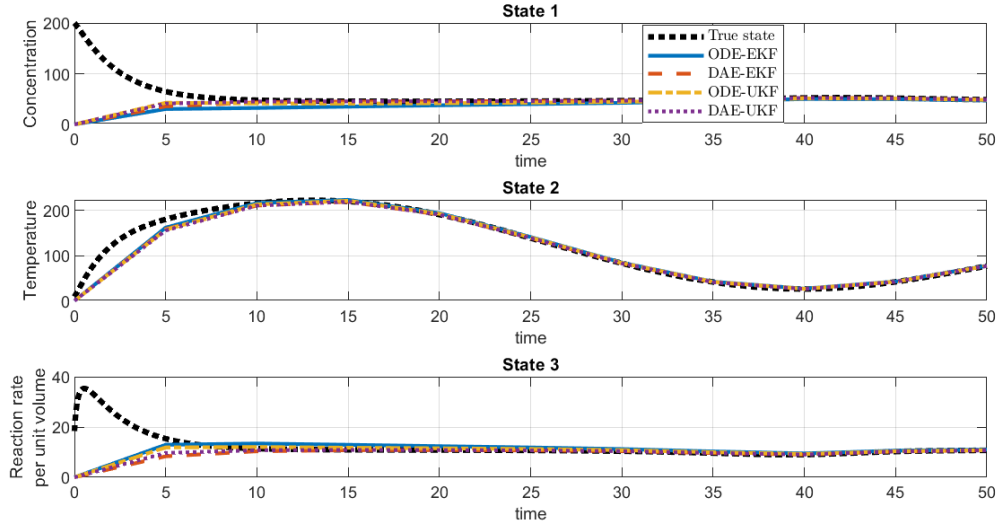


Figure 4.3: Plot of the chemical reactor in Example 4.2. The true state (black dotted line) and the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line). The temperature is measured with no noise. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 0.0336 s, 0.0513 s, 0.1820 s and 0.1814 s, respectively.

For the EKF applied on the DAE we have

$$Q = \text{diag}(0.01, 0.01), \quad P_{0|0} = \text{diag}(0.01, 0.01, 0.01).$$

For the UKF applied on the DAE we have

$$Q = \text{diag}(0.01, 0.01), \quad P_{0|0} = \text{diag}(0.01, 0.01).$$

The estimate provided by the EKF applied on the ODE is labelled ODE-EKF, the estimate provided by EKF applied on the DAE is labelled DAE-EKF, similarly for UKF.

Executing a simulation of this model from time equals 0 to 50 we can see in Figure 4.3 that all filters follow the true state of the temperature (black dotted line) almost exactly. The estimates for the other variables track the true state well. The computations are done in Matlab 2019b using an Intel i5-8265 CPU at 1.60 GHz.

Let's change the measurement matrix C from $[0 \ 1 \ 0]$ to $[1 \ 0 \ 0]$, now measure the concentration not the temperature. See Figure 4.4 for plot. Now the estimates for the concentration follows the true state almost exactly and the other estimates take time to converge to the true state.

Let's again change the measurement matrix C from $[1 \ 0 \ 0]$ to $[0 \ 0 \ 1]$, now measure the reaction rate. Contrary to what was seen in the previous two figures, here the estimate of the measured variable does not jump to the true estimate after one iteration but like the

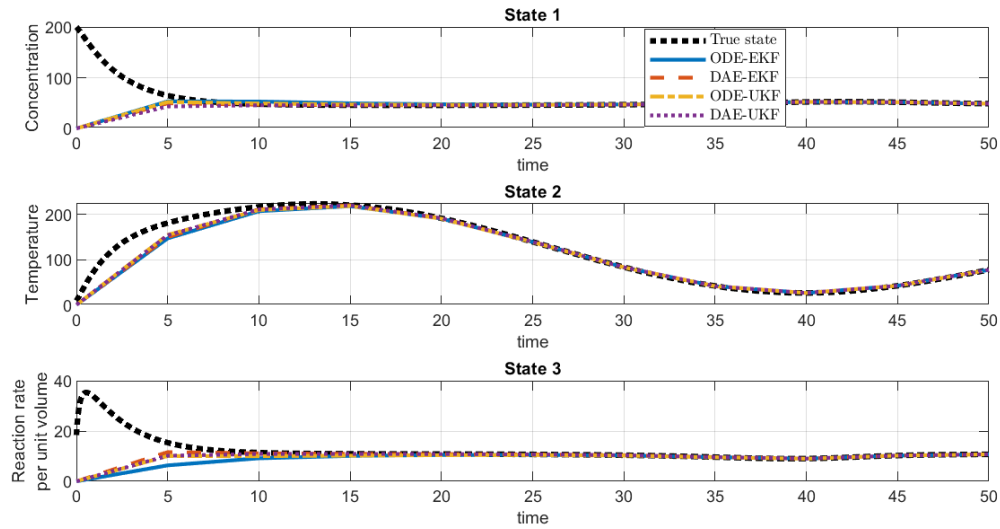


Figure 4.4: Plot of the chemical reactor in Example 4.2. The true state (black dotted line) and the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line). The concentration is measured with no noise. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 0.0295 s, 0.0477 s, 0.2151 s and 0.1962 s, respectively.

rest of the estimates takes a few iterations to converge. See Figure 4.5. This suggests that it might be best to rely on the measurement of the differential values as opposed to algebraic ones.

Let's switch C back to $[0 \ 1 \ 0]$ and introduce measurement noise into the system, the noise is Gaussian with variance equal to $\sqrt{5}$. In Figure 4.6, there is a plot of the state estimates and the absolute error between the estimates and the true state. It is clear from both plots that applying a DAE-compatible filter on the original DAE provides a better estimate than transforming the DAE into an ODE and applying a standard ODE filter. The output covariance is not discussed here because the filters produce covariance matrices of different sizes thus a comparison between them may not be fair.

4.5 Numerical Example - Chemical Akzo Nobel Problem

Example 4.3. Consider the model of the chemical Akzo Nobel problem [36]. This is a model of a chemical process in which two species, FLB and ZHU, are mixed while carbon dioxide is continuously added. Use a Kalman filter to track the states of this model.

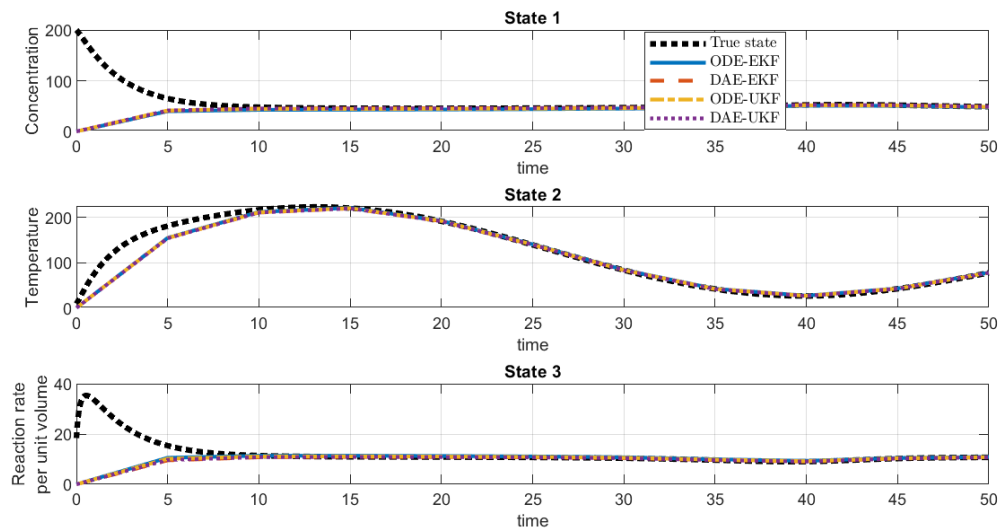
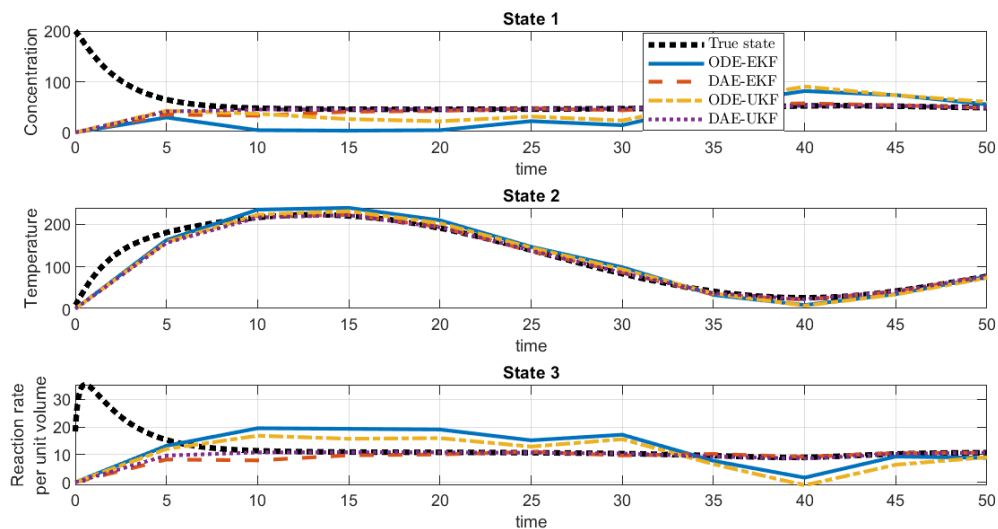
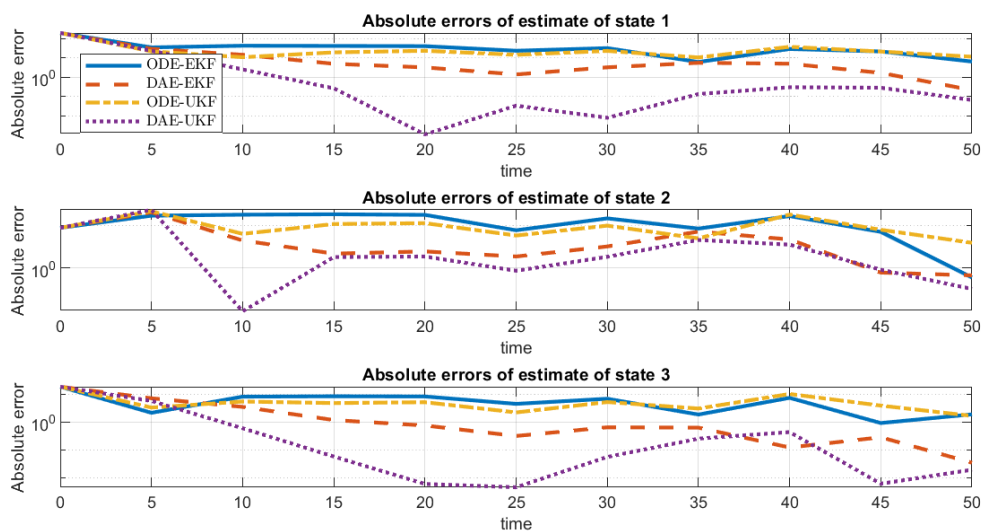


Figure 4.5: Plot of the chemical reactor in Example 4.2. The true state (black dotted line) and the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line). The reaction rate per unit volume is measured with no noise. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 0.0312 s, 0.0466 s, 0.1788 s and 0.1824 s, respectively.



(a) The true state (black dotted line) and the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line).



(b) The absolute error of the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line).

Figure 4.6: Plot of the chemical reactor in Example 4.2. The measured state is temperature, the variance of the measurement noise is equal to $\sqrt{5}$. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 0.0342 s, 0.0562 s, 0.2153 s and 0.1853 s, respectively.

The differential and algebraic states are

$$x_D = \begin{bmatrix} [FLB] \\ [CO_2] \\ [FLBT] \\ [ZHU] \\ [ZLA] \end{bmatrix} \quad x_A = [FLB.ZHU]$$

where the square brackets denote concentration. For simplicity, define the following

$$\begin{aligned} r_1 &= k_1 x_{D,1}^4 x_{D,2}^{0.5}, \\ r_2 &= k_2 x_{D,3} x_{D,4}, \\ r_3 &= \frac{k_2}{K} x_{D,1} x_{D,5}, \\ r_4 &= k_3 x_{D,1} x_{D,4}^2, \\ r_5 &= k_4 x_A^2 x_{D,2}^{0.5}, \\ F_{in} &= klA \left(\frac{p(CO_2)}{H} - x_{D,2} \right). \end{aligned}$$

The forcing function is

$$f_D(t, x_D, x_A) = \begin{bmatrix} -2r_1 + r_2 - r_3 - r_4 \\ -0.5r_1 - r_4 - 0.5r_5 + F_{in} \\ r_1 - r_2 + r_3 \\ -r_2 + r_3 - 2r_4 \\ r_2 - r_3 + r_5 \end{bmatrix},$$

and the algebraic constraint is

$$f_A(t, x_D, x_A) = K x_{D,1} x_{D,4} - x_A.$$

The Jacobian of the constraint with respect to the algebraic state is equal to -1 thus this system is index 1. The constants are

$$\begin{aligned} k &= [18.7 \quad 0.58 \quad 0.09 \quad 0.42]^T, \\ K &= 34.4, \quad klA = 3.3, \quad p(CO_2) = 0.9, \quad H = 737. \end{aligned}$$

and the initial conditions are

$$x_0 = [0.444 \quad 0.00123 \quad 0 \quad 0.007 \quad 0 \quad 0.003108K]^T.$$

Here we measure two states $x_{D,3} = [FLBT]$ and $x_{D,5} = [ZHU]$, the measurement covariance matrix is $diag(5e - 3, 1e - 6)$ and the UKF tuning parameter, κ , is equal to 1. Thus,

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$R = diag(5e - 3, 1e - 6),$$

$$\kappa = 1.$$

The measurements are obtained every 20 seconds. The initial state of the estimator is

$$\hat{x}_{0|0} = [0.5 \quad 0.001 \quad 0.8 \quad 0.001 \quad 0.001 \quad 0.0005K]^T \text{ (consistent with algebraic constraint)}$$

In addition to using the index-1 DAE-compatible EKF and UKF, we convert this DAE to an ODE by changing the algebraic constraint into a differential equation as in equation (4.19). Apply the common ODE-compatible EKF and UKF as defined in Section 4.2 on this ODE. The four filters used here require the process covariances and initial covariances to be defined with different dimensions. For the filters applied on ODE we have:

$$Q = 10^{-7}I_{6 \times 6}, \quad P_{0|0} = 10^{-7}I_{6 \times 6}.$$

For the EKF applied on the DAE we have

$$Q = 10^{-7}I_{5 \times 5}, \quad P_{0|0} = 10^{-7}I_{6 \times 6}.$$

For the UKF applied on the DAE we have

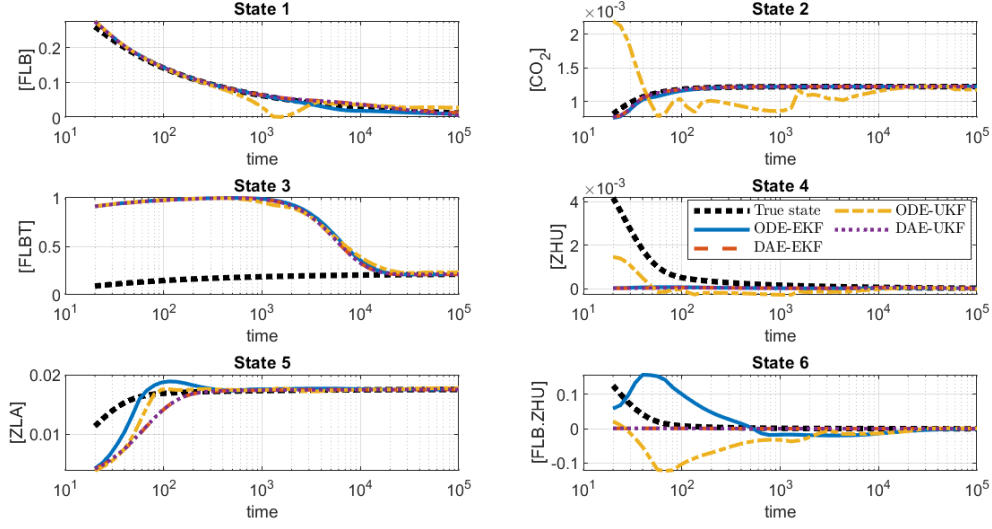
$$Q = 10^{-7}I_{5 \times 5}, \quad P_{0|0} = 10^{-7}I_{5 \times 5}.$$

Executing a simulation of this model from time equals 0 to 100 000. The results are in Figure 4.7. We can see that all filters converge to the true state, however, for state 3 and 6 the filters which use the DAE converge to a much smaller error.

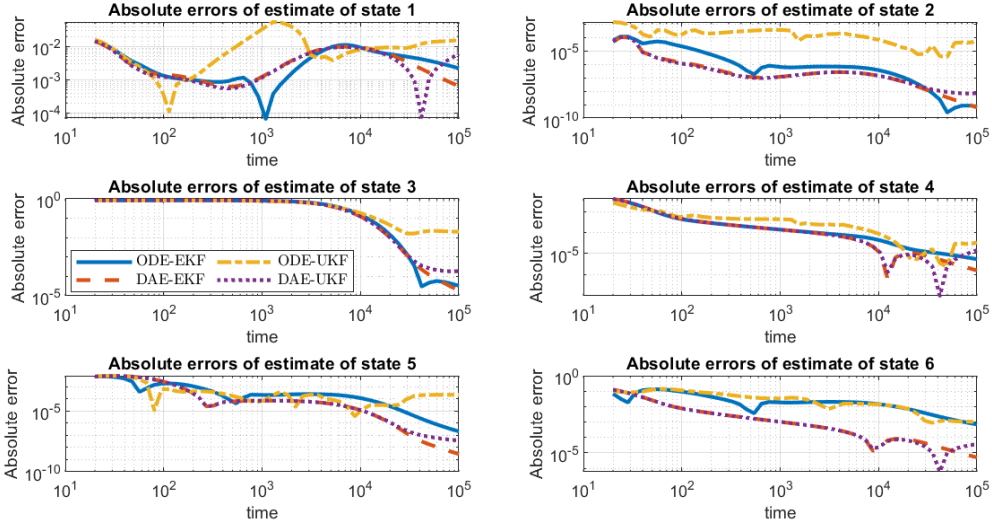
Let's add measurement noise, the measurement of the third state will have white noise with a variance of 9×10^{-4} and the measurement of the fifth state will have white noise with a variance of 10^{-8} . The plot is in Figure 4.8, the estimate of the algebraic state provided by the DAE-compatible filters is much better compared to the estimate provided by the ODE filters.

In this section, we introduced a pair of Kalman filters, EKF and UKF, which can be applied on semi-explicit index-1 differential-algebraic equations. Then, two examples were considered, in both we attempted to track the states of a given DAE. In both examples, this was done in four ways, applying the DAE-compatible filters, DAE-EKF and DAE-UKF, on the original DAE model and transforming the DAE into an ODE using equation (4.19) then applying well known ODE-compatible filters, ODE-EKF and ODE-UKF.

The DAE-compatible filters performed as good or better than the ODE-compatible filters. In noisy and noise-less cases, they performed noticeably better in estimating the value of the algebraic state. In estimating the measured states, all filters performed about the same. The run-time of the EKF filters is a fraction of the run-time of the UKF filters. Despite this, the performances of DAE-EKF and DAE-UKF was about the same. To the author's knowledge, comparisons between DAE- and ODE-compatible filters have not been previously made.

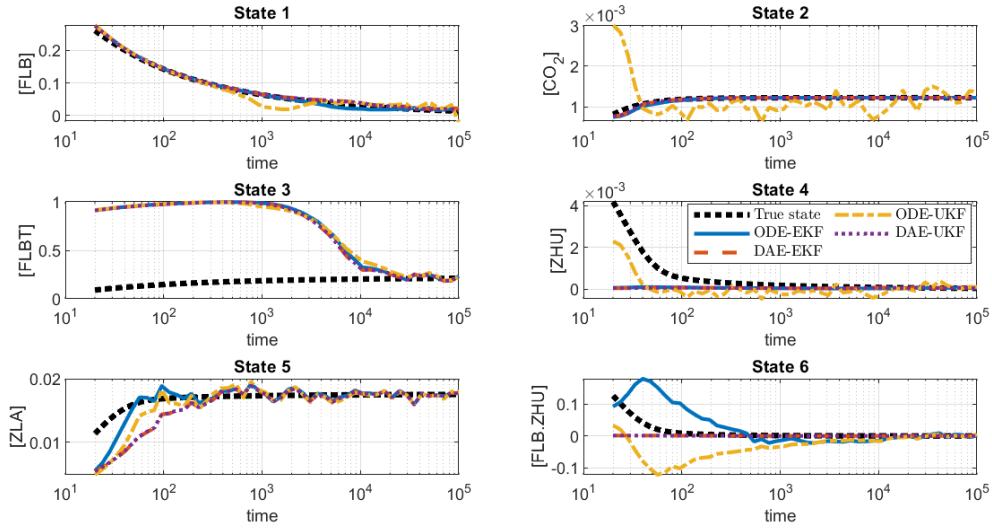


(a) The true state (black dotted line) and the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line).

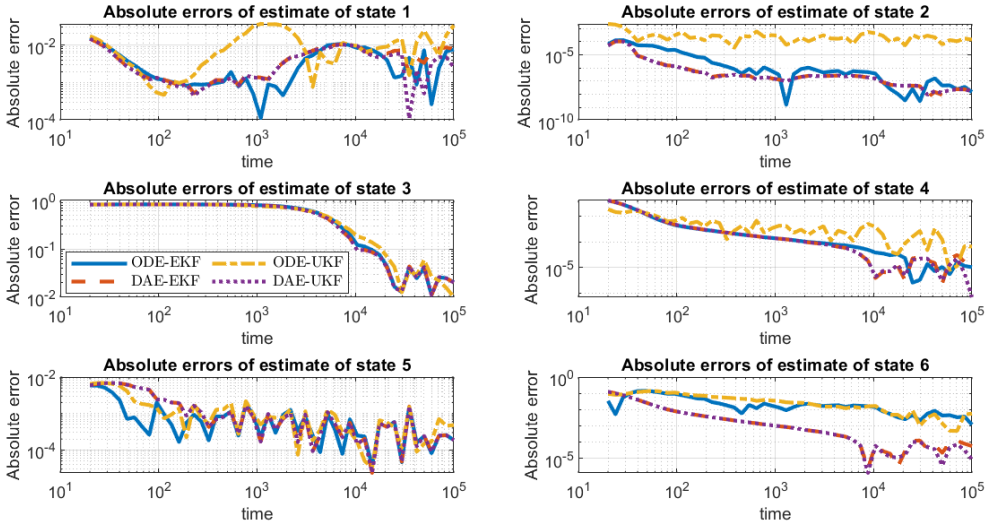


(b) The absolute error of the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line).

Figure 4.7: Plot of the chemical reactor in Example 4.3. The measured states are the third and the fifth, there is no measurement noise. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 6.6442 s, 15.5019 s, 184.2725 s and 138.3562 s, respectively.



(a) The true state (black dotted line) and the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line).



(b) The absolute error of the state estimates produced by EKF on the ODE (blue solid line), EKF on DAE (red dashed line), UKF on ODE (yellow dash-dot line) and UKF on DAE (purple dotted line).

Figure 4.8: Plot of the chemical reactor in Example 4.3. The measured states are the third and the fifth, the variances of the measurement noises are 9×10^{-4} and 10^{-8} , respectively. The computation times of ODE-EKF, DAE-EKF, ODE-UKF, DAE-UKF are 7.5082 s, 16.4353 s, 178.7342 s and 138.5590 s, respectively.

4.6 Numerical Example - Estimating Lithium-ion Cell SOC using a Kalman filter

In this section, the state of charge (SOC) of the lithium ion cell model is estimated using the DAE-compatible EKF with voltage as the only measurement. The DAE is semi-explicit index-1 and is a discretization of a system of partial differential equations. The value of the SOC is a value derived from the concentration of lithium ions in the anode. Measuring other states, such as the concentration of the lithium ions is infeasible in practice. Indeed, the terminal voltage is proportional to the SOC, however open circuit voltage plotted versus the SOC forms a hysteresis loop [2, 24, 65, 79] which makes using the terminal voltage as an estimate difficult.

The DAE modelling the lithium-ion cell is high dimensional, thus neither the ODE- or DAE- UKF filter is used, the computation time would be too great. As already stated, converting a semi-explicit index-1 DAE into an ODE requires the derivation of the Jacobian of the algebraic constraint. For this model, the computation time of computing the inverse Jacobian is too great and thus the ODE-EKF is not presented either.

The estimation is tested under different current inputs and different variance values for measurement noise, in all cases the estimate performs well. In Figure 4.9, the input current is equal to zero and the measurement noise is equal to zero, the estimate converges to the true state. At convergence, the absolute value of the difference between the true state and the estimate is about 10^{-3} . In each figure, the absolute error is shown, this is the absolute value of the difference between the true SOC and the estimated SOC.

In Figures 4.10 and 4.11, the input current is equal to zero but the measurement noise has variance equal to 10^{-4} and 9×10^{-4} , respectively. In both cases the estimate converges. At convergence the absolute values of the difference between the true state and the estimate is about 10^{-2} and 10^{-1} , respectively.

In Figure 4.12, a charging current and a discharging current is applied and the measurement noise is equal to zero, the estimate converges to the true state. At convergence, the absolute value of the difference between the true state and the estimate is about 10^{-3} .

In Figure 4.13, a charging current is applied and the measurement noise is equal to zero, the estimate converges to the true state. At convergence, the absolute value of the difference between the true state and the estimate is about 10^{-3} .

In Figure 4.14, a charging current is applied and the measurement noise has variance equal 10^{-8} , the estimate converges to the true state. At convergence, the absolute value of the difference between the true state and the estimate is about 10^{-3} .

In this section, simulation results are presented when the model is subjected to different current inputs and different standard deviation values of measurement noise. In all cases the estimator performs well. When the noise is small the absolute value of the difference between the estimate and the true state settles below 0.01. In the cases where the variance of the noise is large the estimator performs well also.

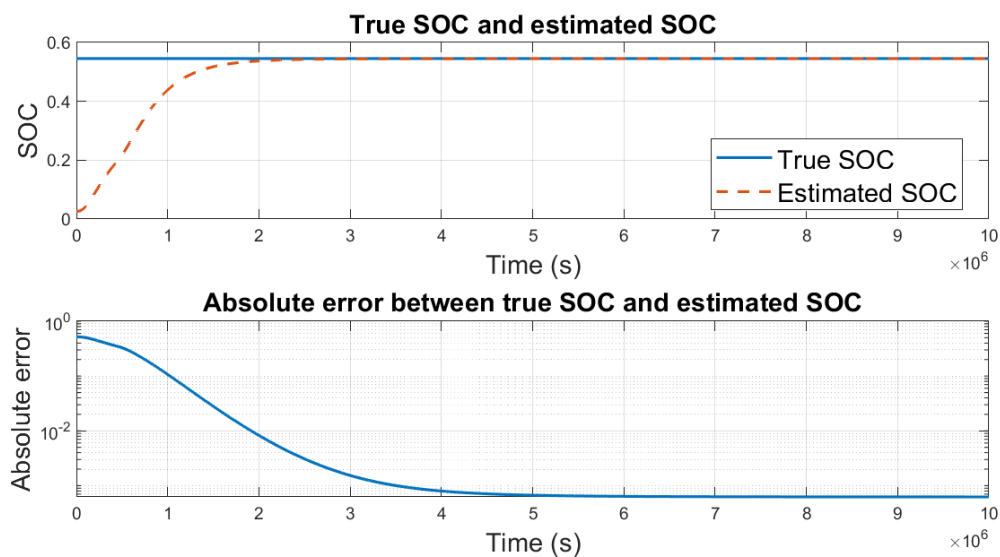


Figure 4.9: The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. There is no measurement noise in the system, the absolute error converges to about 7×10^{-4} . There is no current being applied on the cell.

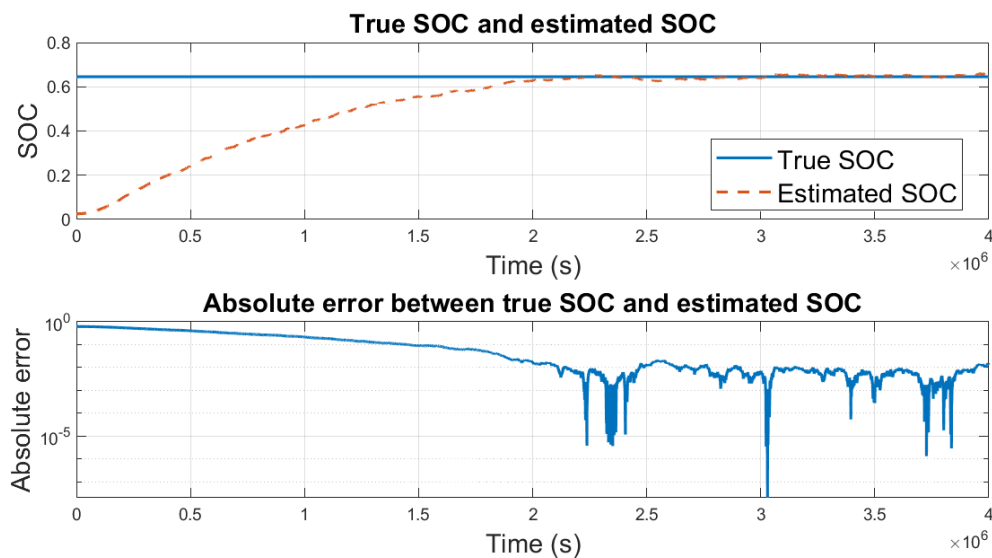


Figure 4.10: The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. The measurement is noisy, the variance of the noise is 10^{-4} . The absolute error converges to about 9×10^{-3} . There is no current being applied on the cell.

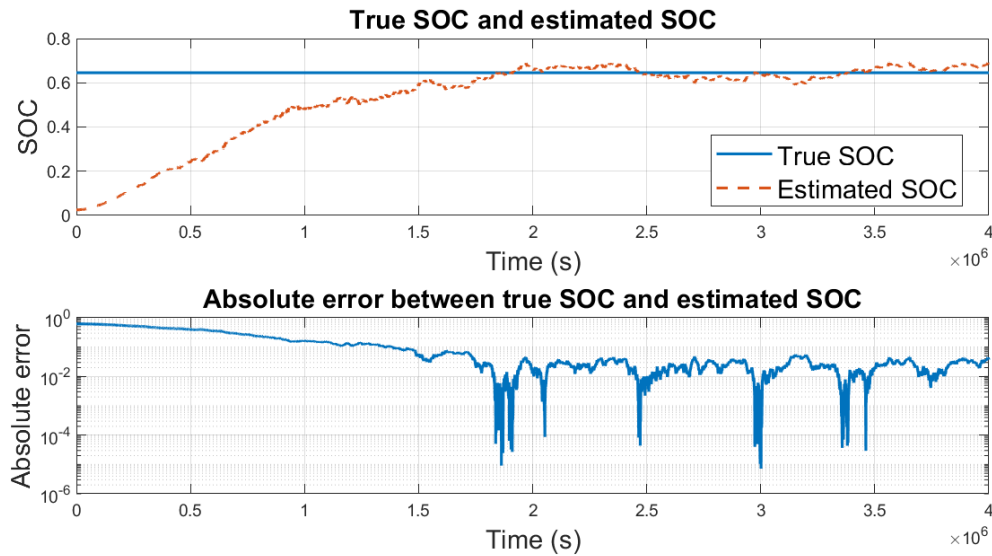


Figure 4.11: The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. The measurement is noisy, the variance of the noise is 9×10^{-4} , the absolute error converges to about 2×10^{-2} . There is no current being applied on the cell.

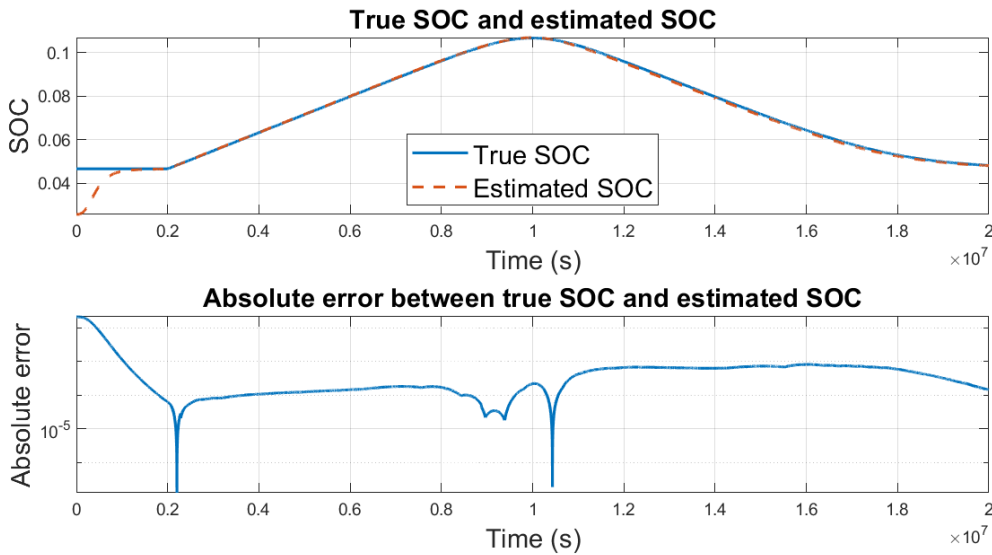


Figure 4.12: The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. The estimate is given time to converge to the true state after which the cell is charged and discharged. There is no measurement noise. After the current is applied, the absolute error does not exceed 10^{-3} .

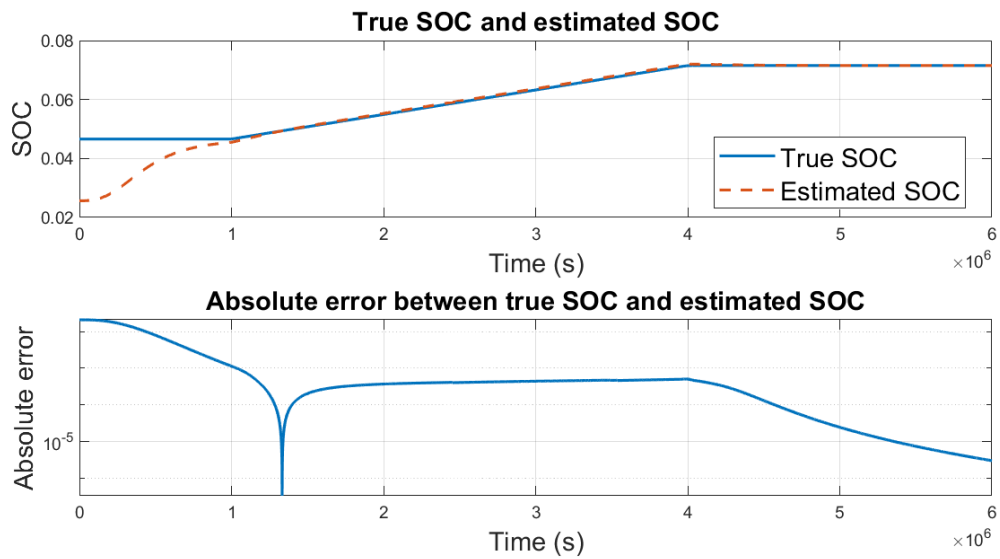


Figure 4.13: The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. The cell is charged for a period of time, prior and after which the estimate is given time to converge to the true state. There is no measurement noise. After the current is applied, the absolute error does not exceed 10^{-3} .

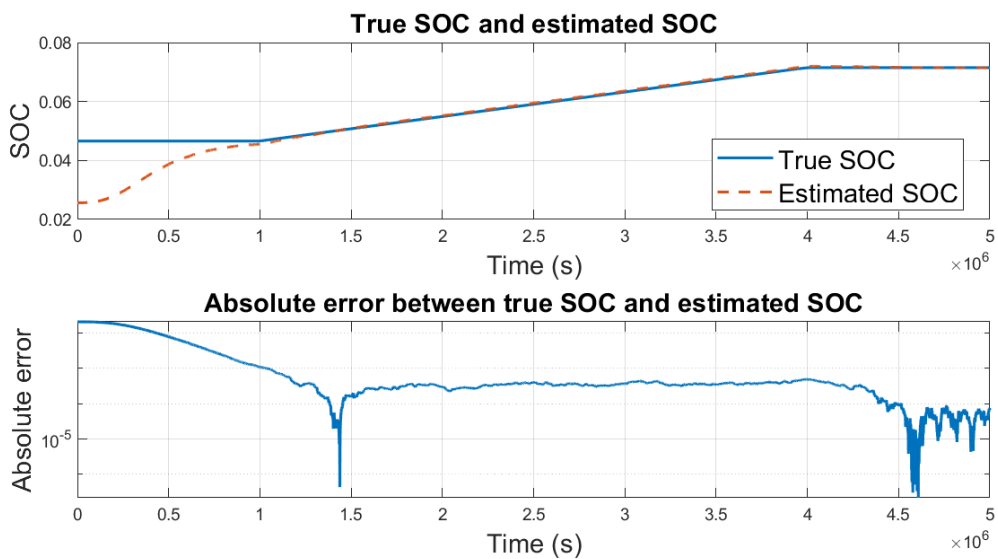


Figure 4.14: The top plot displays the true SOC (blue solid line) and the estimated SOC (dashed red line). The bottom plot displays the absolute error between the true and estimated SOC. The cell is charged for a period of time, prior and after which the estimate is given time to converge to the true state. Measurement noise has variance equal to 10^{-8} . There is a charging current being applied on the cell. After the current is applied, the absolute error does not exceed 10^{-3}

Chapter 5

Conclusion

In the second chapter, a lithium ion cell consisting of four coupled partial differential equations was presented, two equations modelled the lithium ion concentration in the electrolyte and electrodes and two modelled the electric potential in the electrolyte and electrodes. The electric potential equations did not contain a temporal derivative. When discretized, this produced algebraic constraints as opposed to the usual differential equation, thus the overall discretization generated a DAE.

Existence-uniqueness theory of DAE's was discussed for a linear DAE. The theory provided a way to check if a unique solution to a DAE is available. The concept of consistent initial conditions was introduced, this requires that for a DAE to be solvable the initial conditions have to satisfy the algebraic constraint, a concept not present in ordinary differential equations.

Numerical theory of solving differential-algebraic equations was presented in Chapter 3. This theory showed that it is possible to solve DAE's using backward differentiation formulae. Steps for approximately solving index-1 DAE's in Matlab were presented. An example was solved numerically to demonstrate the capability of the numerical method.

Chapter 4 was centred on Kalman filters. Basic theory of recursive least-squares estimator was discussed then the Kalman filter was presented. Two non-linear Kalman filters, the extended Kalman filter and unscented Kalman filter, were discussed and implemented in an example. Then their DAE-compatible counterparts were examined and applied on two examples, along with the ODE-compatible filters. The performance of the different filters was compared by studying the computation times and the absolute errors of the estimates. The DAE-compatible filters estimated the true state more accurately. In the presence of measurement noise, the accuracy of the DAE-estimates over the ODE-estimates is magnified. To the author's knowledge, comparisons between DAE- and ODE-compatible filters have not been made.

The semi-explicit index-one differential-algebraic equation-compatible extended Kalman filter was used on the discretized model of the lithium ion cell with satisfactory results with voltage as the only measurement. The filter was able to track the state of charge in the presence of measurement noise.

An implementation of a conservation law into the lithium-ion model could be a way to

improve to model. The lithium ions are restricted to the inside of the electrochemical cell, either as particles dissolved in the electrolyte or as part of the electrolyte. The total amount of lithium ions inside the cell remains constant thus an improvement one could make to the model would be to enforce this with an algebraic constraint.

Some attempts at estimating the state of charge of a lithium-ion cell using a Kalman filter can be found here [2, 3, 6, 57]. A DAE-compatible Kalman filter has been applied on a model of a lithium ion cell [12] with the goal of estimating the SOC.

Future work can include:

- comparing the already developed DAE-compatible ensemble Kalman filter [53] and DAE-compatible iterative extended Kalman filter [54] with their ODE counterparts,
- developing a DAE-compatible H_∞ filter or developing DAE-compatible counterparts of other Kalman filters such as second-order EKF,
- developing a filter which is compatible with higher order DAE's, higher order DAE's appear in constrained multi-body systems [9, 60] and vehicle dynamics [61].

Developing Kalman filters for DAE's allows us to perform estimation directly on the given system without introducing errors by converting it into an ODE.

Bibliography

- [1] Nassim Abdul Samad, Jason Siegel, Anna Stefanopoulou, and Aaron Knobloch. Observability analysis for surface sensor location in encased battery cells. *Proceedings of the American Control Conference*, 2015:299–304, 07 2015.
- [2] Sepideh Afshar. *Lithium-ion battery SOC estimation*. PhD thesis, University of Waterloo, 2017.
- [3] Sepideh Afshar, Kirsten Morris, and Amir Khajepour. State-of-charge estimation using an EKF-based adaptive observer. *IEEE Transactions on Control Systems Technology*, pages 1–17, 2018.
- [4] Ali Almagbile, Jinling Wang, and Weidong Ding. Evaluating the performances of adaptive Kalman filter methods in GPS/INS integration. *Journal of Global Positioning Systems*, 9(1):33–40, jun 2010.
- [5] Francois Auger, Mickael Hilairret, Josep M. Guerrero, Eric Monmasson, Teresa Orłowska-Kowalska, and Seiichiro Katsura. Industrial applications of the Kalman filter: a review. *IEEE Transactions on Industrial Electronics*, 60(12):5458–5471, dec 2013.
- [6] Faten Ayadi, Mongi Lahiani, and Nabil Derbel. Kalman filter Observer for SoC prediction of lithium cells. *Advances in Science, Technology and Engineering Systems Journal*, 2(4):180–188, sep 2017.
- [7] V.M. Becerra, P.D. Roberts, and G.W. Griffiths. Applying the extended Kalman filter to systems described by nonlinear differential-algebraic equations. *Control Engineering Practice*, 9(3):267–281, mar 2001.
- [8] Michal Beneš, Radek Fučík, Vladimír Havlena, Vladimír Klement, Miroslav Kolář, Ondřej Polívka, Jakub Solovský, and Pavel Strachota. An efficient and robust numerical solution of the full-order multiscale model of lithium-ion battery. *Mathematical Problems in Engineering*, 2018:1–12, 2018.
- [9] Brahim Benhammouda and Hector Vazquez-Leal. Analytical solution of a nonlinear index-three DAEs system modelling a slider-crank mechanism. *Discrete Dynamics in Nature and Society*, 2015:1–14, 2015.

- [10] Hendrik Johannes Bergveld. *Battery management systems : design by modelling*. PhD thesis, Netherlands, 6 2001.
- [11] Ranjeeta Bisoi and P.K. Dash. A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter. *Applied Soft Computing*, 19:41–56, jun 2014.
- [12] A.M. Bizeray, S. Zhao, S.R. Duncan, and D.A. Howey. Lithium-ion battery thermal-electrochemical model-based state estimation using orthogonal collocation and a modified extended Kalman filter. *Journal of Power Sources*, 296:400–412, nov 2015.
- [13] John P Boyd. *Chebyshev and Fourier spectral methods*. Lecture notes in engineering ; 49. Springer-Verlag, Berlin, 2000.
- [14] Steven D. Brown. The Kalman filter in analytical chemistry. *Analytica Chimica Acta*, 181:1–26, 1986.
- [15] Stephen L. Campbell and C. William Gear. The index of general nonlinear DAEs. *Numerische Mathematik*, 72(2):173–196, dec 1995.
- [16] Stephen L. Campbell and E. Griepentrog. Solvability of general differential algebraic equations. *SIAM Journal on Scientific Computing*, 16(2):257–270, mar 1995.
- [17] M Corno, N Bhatt, S M Savaresi, and M Verhaegen. Electrochemical model-based state of charge estimation for Li-Ion cells. *IEEE Transactions on Control Systems Technology*, 23(1):117–127, jan 2015.
- [18] Marcia de Fatima Brondani, Airam Teresa Zago Romcy Sausen, Paulo Sérgio Sausen, and Manuel Osório Binelo. Parameter estimation of lithium ion polymer battery mathematical model using genetic algorithm. *Computational and Applied Mathematics*, 37(1):296–313, dec 2018.
- [19] Guangzhong Dong, Zonghai Chen, and Jingwen Wei. Sequential Monte Carlo filter for state-of-charge estimation of lithium-ion batteries based on auto regressive exogenous model. *IEEE Transactions on Industrial Electronics*, 66(11):8533–8544, nov 2019.
- [20] Denys Dutykh. A brief introduction to pseudo-spectral methods: application to diffusion problems. jun 2016.
- [21] James F. Epperson. On the Runge example. *The American Mathematical Monthly*, 94(4):329, apr 1987.
- [22] Gene H. Golub and Charles H. Van Loan. *Matrix computations*. Johns Hopkins studies in the mathematical sciences. Johns Hopkins University Press, Baltimore, third edition, 1996.

- [23] E. Hairer and G. Wanner. On the Instability of the BDF Formulas. *SIAM Journal on Numerical Analysis*, 20(6):1206–1209, dec 1983.
- [24] M.A. Hannan, M.S.H. Lipu, A. Hussain, and A. Mohamed. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. *Renewable and Sustainable Energy Reviews*, 78:834–854, oct 2017.
- [25] Sung-Kwon Hong, Bogdan I. Epureanu, and Matthew P. Castanier. Parametric reduced-order models of battery pack vibration including structural variation and prestress effects. *Journal of Power Sources*, 261:101–111, sep 2014.
- [26] P. L. Houtekamer, Herschel L. Mitchell, and Xingxiu Deng. Model error representation in an operational ensemble Kalman filter. *Monthly Weather Review*, 137(7):2126–2143, jul 2009.
- [27] P. L. Houtekamer, Herschel L. Mitchell, Gérard Pellerin, Mark Buehner, Martin Charon, Lubos Spacek, and Bjarne Hansen. Atmospheric data assimilation with an ensemble Kalman filter: results with real observations. *Monthly Weather Review*, 133(3):604–620, mar 2005.
- [28] Jiatang Hu, Katja Fennel, Jann Paul Mattern, and John Wilkin. Data assimilation with a local ensemble Kalman filter applied to a three-dimensional biological model of the Middle Atlantic Bight. *Journal of Marine Systems*, 94:145–156, jun 2012.
- [29] Arieh Iserles. *A first course in the numerical analysis of differential equations*, volume 65. 01 2008.
- [30] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, mar 2000.
- [31] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the Kalman filter to nonlinear systems. page 182, jul 1997.
- [32] S.J. Julier and J.K. Uhlmann. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3):401–422, mar 2004.
- [33] Bi Jun, Sai Shao, Guan Wei, and Wang Lu. State of charge estimation of li-ion batteries in an electric vehicle based on a radial-basis-function neural network. *Chinese Physics B*, 21:118801, 11 2012.
- [34] Wang Junping, Guo Jingang, and Ding Lei. An adaptive kalman filtering based state of charge combined estimator for electric vehicle battery pack. *Energy Conversion and Management - ENERG CONV MANAGE*, 50:3182–3186, 12 2009.

- [35] Dah-Jing Jwo and Sheng-Hung Wang. Adaptive fuzzy strong tracking extended Kalman filtering for GPS navigation. *IEEE Sensors Journal*, 7(5):778–789, may 2007.
- [36] B Kafash, A Kaffash, and N Lalehzari. Chemical Akzo Nobel problem; mathematical description and its solution via model maker software. *MATCH-COMMUNICATIONS IN MATHEMATICAL AND IN COMPUTER CHEMISTRY*, 71(2):279–286, 2014.
- [37] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, mar 1960.
- [38] David A. Kopriva. *Implementing spectral methods for partial differential equations*, volume 53 of *Scientific Computation*. Springer Netherlands, Dordrecht, 1st ed. 20 edition, 2009.
- [39] Ravi Kumar Mandela, Raghunathan Rengaswamy, Shankar Narasimhan, and Lakshmi N. Sridhar. Recursive state estimation techniques for nonlinear differential algebraic systems. *Chemical Engineering Science*, 65(16):4548–4556, aug 2010.
- [40] Peter Kunkel and Volker Mehrmann. *Differential-algebraic equations*. European Mathematical Society Publishing House, Zuerich, Switzerland, feb 2006.
- [41] Chi-Yuan Lee, Shuo-Jen Lee, Yi-Ming Hung, Chien-Te Hsieh, Yu-Ming Chang, Yen-Ting Huang, and Jyun-Ting Lin. Integrated microsensor for real-time microscopic monitoring of local temperature, voltage and current inside lithium ion battery. *Sensors and Actuators A: Physical*, 253, 10 2016.
- [42] Feng Leng, Cher Ming Tan, Rachid Yazami, and Minh Duc Le. A practical framework of electrical based online state-of-charge estimation of lithium ion batteries. *Journal of Power Sources*, 255:423–430, jun 2014.
- [43] Lijun Qian, Haixin Wang, and E.R. Dougherty. Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering. *IEEE Transactions on Signal Processing*, 56(7):3327–3339, jul 2008.
- [44] Xinfan Lin, Youngki Kim, Shankar Mohan, Jason B. Siegel, and Anna G. Stefanopoulou. Modeling and estimation for advanced battery management. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):393–426, may 2019.
- [45] Languang Lu, Xuebing Han, Jianqiu Li, Jianfeng Hua, and Minggao Ouyang. A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of Power Sources*, 226:272–288, mar 2013.
- [46] John C Mason and David C Handscomb. *Chebyshev polynomials*. CRC press, 2002.
- [47] N. Meskin, H. Nounou, M. Nounou, and A. Datta. Parameter estimation of biological phenomena: an unscented Kalman filter approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(2):537–543, mar 2013.

- [48] Howard Musoff and Paul Zarchan. *Fundamentals of Kalman filtering: a practical approach, third edition*. American Institute of Aeronautics and Astronautics, Reston, VA, jan 2009.
- [49] Woochul Nam, Ji-Young Kim, and Ki-Yong Oh. The characterization of dynamic behavior of Li-ion battery packs for enhanced design and states identification. *Energy Conversion and Management*, 162:264–275, apr 2018.
- [50] Kong Soon Ng, Chin-Sien Moo, Yi-Ping Chen, and Yao-Ching Hsieh. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied Energy*, 86(9):1506–1511, sep 2009.
- [51] Gregory Plett. Extended kalman filtering for battery management systems of LiPB-based HEV battery packs. *Journal of Power Sources*, 134:262–276, 08 2004.
- [52] V. Pop, H. J. Bergveld, J. H G Op Het Veld, P. P L Regtien, D. Danilov, and P. H L Notten. Modeling battery behavior for accurate state-of-charge indication. *Journal of the Electrochemical Society*, 153(11):A2013, 2006.
- [53] Yash Puranik, Vinay A. Bavdekar, Sachin C. Patwardhan, and Sirish L. Shah. An ensemble Kalman filter for systems governed by differential algebraic equations (DAEs). *IFAC Proceedings Volumes*, 45(15):531–536, 2012.
- [54] Jalesh L. Purohit and Sachin C. Patwardhan. Development of iterative extended Kalman filter for DAE System. *IFAC-PapersOnLine*, 51(1):691–696, 2018.
- [55] P. Ramadass, Bala Haran, Parthasarathy M. Gomadam, Ralph White, and Branko N. Popov. Development of first principles capacity fade model for Li-Ion cells. *Journal of The Electrochemical Society*, 151(2):A196, 2004.
- [56] Carl Runge. Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten. *Zeitschrift für Mathematik und Physik*, 46(224-243):20, 1901.
- [57] Shriram Santhanagopalan and Ralph E. White. Online estimation of the state of charge of a lithium ion cell. *Journal of Power Sources*, 161(2):1346–1355, oct 2006.
- [58] A. J. Segers, H. J. Eskes, R. J. van der A, R. F. van Oss, and P. F. J. van Velthoven. Assimilation of GOME ozone profiles and a global chemistry–transport model using a Kalman filter with anisotropic covariance. *Quarterly Journal of the Royal Meteorological Society*, 131(606):477–502, jan 2005.
- [59] Lawrence F. Shampine, Mark W. Reichelt, and Jacek A. Kierzenka. Solving index-I DAEs in MATLAB and Simulink. *SIAM Review*, 41(3):538–552, jan 1999.
- [60] B. Simeon. An extended descriptor form for the numerical integration of multibody systems. *Applied Numerical Mathematics*, 13(1-3):209–220, sep 1993.

- [61] B. Simeon, F. Grupp, C. Führer, and P. Rentrop. A nonlinear truck model and its treatment as a multibody system. *Journal of Computational and Applied Mathematics*, 50(1-3):523–532, may 1994.
- [62] Dan Simon. *Optimal state estimation: Kalman, H_∞ , and nonlinear approaches*. John Wiley & Sons, Inc., Hoboken, NJ, USA, may 2006.
- [63] Kandler Smith and Chao Yang Wang. Solid-state diffusion limitations on pulse operation of a lithium ion cell for hybrid electric vehicles. *Journal of Power Sources*, 161(1):628–639, oct 2006.
- [64] Kandler A. Smith, Chris Rahn, and Chao-Yang Wang. Model-based electrochemical estimation and constraint management for pulse operation of lithium ion batteries. *Control Systems Technology, IEEE Transactions on*, 18:654 – 663, 06 2010.
- [65] Venkat Srinivasan and John Newman. Existence of path-dependence in the LiFePO₄ electrode. *Electrochemical and Solid State Letters - ELECTROCHEM SOLID STATE LETT*, 9, 01 2006.
- [66] Karen E. Thomas, John Newman, and Robert M. Darling. *Mathematical modeling of lithium batteries*, pages 345–392. Springer US, Boston, MA, 2002.
- [67] Lloyd Trefethen. *Approximation theory and approximation practice*. 01 2013.
- [68] Lloyd N. Trefethen. *Spectral methods in MATLAB*. Society for Industrial and Applied Mathematics, jan 2000.
- [69] Lars Ole Valøen and Jan N. Reimers. Transport properties of LiPF₆-based Li-Ion battery electrolytes. *Journal of The Electrochemical Society*, 152(5):A882, 2005.
- [70] Mark Verbrugge and Edward Tate. Adaptive state of charge algorithm for nickel metal hydride batteries including hysteresis phenomena. *Journal of Power Sources*, 126(1-2):236–249, feb 2004.
- [71] Dinh Vinh Do, Christophe Forgez, Khadija El Kadri Benkara, and Guy Friedrich. Impedance observer for a li-ion battery using kalman filter. *Vehicular Technology, IEEE Transactions on*, 58:3930 – 3937, 11 2009.
- [72] J. A. Weideman and S. C. Reddy. A MATLAB differentiation matrix suite. *ACM Transactions on Mathematical Software (TOMS)*, 26(4):465–519, dec 2000.
- [73] Daniel M. Wolpert and Zoubin Ghahramani. Computational principles of movement neuroscience. *Nature Neuroscience*, 3(S11):1212–1217, nov 2000.
- [74] Yinjiao Xing, Wei He, Michael Pecht, and Kwok Leung Tsui. State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures. *Applied Energy*, 113:106–115, jan 2014.

- [75] Long Xu, Junping Wang, and Quanshi Chen. Kalman filtering state of charge estimation for battery management system based on a stochastic fuzzy neural network battery model. *Energy Conversion and Management*, 53(1):33–39, jan 2012.
- [76] Yee-Pien Yang, Jieng-Jang Liu, and Chia-Hao Tsai. Improved estimation of residual capacity of batteries for electric vehicles. *Journal of the Chinese Institute of Engineers*, 31(2):313–322, mar 2008.
- [77] Yi Zeng, Paul Albertus, Reinhardt Klein, Nalin Chaturvedi, Aleksandar Kojic, Martin Z. Bazant, and Jake Christensen. Efficient conservative numerical schemes for 1d nonlinear spherical diffusion equations with applications in battery modeling. *Journal of the Electrochemical Society*, 160(9):A1565–A1571, jul 2013.
- [78] Yanhui Zhang, Wenji Song, Shili Lin, and Ziping Feng. A novel model of the initial state of charge estimation for LiFePO₄ batteries. *Journal of Power Sources*, 248:1028–1033, feb 2014.
- [79] Letao Zhu, Zechang Sun, Haifeng Dai, and Xuezhe Wei. A novel modeling methodology of open circuit voltage hysteresis for LiFePO₄ batteries based on an adaptive discrete Preisach model. *Applied Energy*, 155, 10 2015.