

# Comparison of Adaptive Controllers for a Servomechanism

**Arnfinn Aas Eielsen**

Master of Science in Engineering Cybernetics

Submission date: July 2007

Supervisor: Jan Tommy Gravdahl, ITK

Co-supervisor: Bruce Woodley, NeoGuide Systems, Inc.



# Problem Description

## Background

This assignment concerns a servomechanism utilizing motors with metal graphite brushes. The servomechanism is used as a part of a robotic colonoscope, for which good position and force control are crucial.

Electric motors with metal graphite brushes experience a change in contact resistance depending on current load, which leads to varying gain in the motor. Using such a motor for force control with a discrete feedback controller is problematic, since current feedback will render the system unstable for any practical sampling time and controller gain. A possible solution to this problem is to use feedforward control, but this requires accurate parameter estimates. A least mean squares estimator is currently being used to learn the value of the resistance. A possible alternative to this is a gain-scheduler based on a model of the deterministic properties of the resistance attenuation phenomenon.

## Assignment

1. Describe and analyze the system.
2. Suggest models that can describe the resistance attenuation phenomenon.
3. Set up laboratory experiment(s) suitable for estimation of motor parameters.
4. Design and implement an adaptive scheme to estimate motor parameters through a series of experiments.
5. Use obtained experimental data to identify parameters in selected models.
6. Compare the different adaptive schemes through practical experiments.

Assignment given: 07. January 2007  
Supervisor: Jan Tommy Gravdahl, ITK



# Abstract

Electric motors with metal graphite brushes experience a change in contact resistance depending on current load. This varying resistance leads to varying gain in the motor, since the proportion of power dissipated due to the resistance changes. This investigation is concerned with a servomechanism, which uses such motors for position and force control. Force control with electric motors is typically accomplished by controlling the current. The servomechanism utilizes a discrete controller, and discretization renders the system unstable for any practical sampling times and controller gains when using current feedback. Feedforward is therefore used to control the current, and is thus sensitive to the variation in the motor's resistance. To counter the sensitivity, a parameter estimator using least mean squares is presently used to learn the value of the resistance.

An alternative to the parameter estimator might be a gain-scheduler based on a model of the resistance attenuation phenomenon. It should be possible to find such a model because of the deterministic quality of the resistance attenuation. In the course of this investigation, an extended Kalman filter has been developed to estimate current and resistance with accuracy. Estimated current and resistance data from a series of experiments has been fitted to a rational model using nonlinear regression. The obtained model was used as a gain-scheduler.

The estimated resistance from the extended Kalman filter, the least mean squares estimator, and the gain-scheduler were used in conjunction with the feedforward controller and compared for their ability to control the current. The extended Kalman filter provided the most accurate results, but at the expense of being more complex than the least mean squares estimator. The gain scheduler was the worst performer, most likely due to unmodeled effects. With some modifications, it was made to perform on par with the least mean squares estimator, but more work is required before it can be recommended for use.

# Acknowledgments

I would like to thank my supervisors Jan Tommy Gravdahl, Øyvind Stavdahl, and Bruce Woodley (from NeoGuide Systems, Inc.), for all of their help and valuable input, Terje Haugen for building the experiment rig, Stefano Bertelli, Per Inge Løvold and John Olav Horrigmo for helping me with the instrumentation, and Roxanne Daniels for proofreading.

Arnfinn Aas Eielsen, Trondheim, July 20, 2007

# Contents

|   |           |
|---|-----------|
| <b>Abstract</b>   | <b>i</b>  |
| <b>Acknowledgments</b>  | <b>ii</b> |
| <b>1 Introduction</b>   | <b>1</b>  |
| 1.1 Motivation . . . . .  | 1         |
| 1.2 Objectives . . . . .  | 2         |
| 1.3 Outline of the report . . . . .                                 | 3         |
| 1.4 Brief description of the servomechanism . . . . .               | 3         |
| <b>2 Modeling</b>   | <b>6</b>  |
| 2.1 Motor model . . . . .   | 6         |
| 2.1.1 Physical model . . . . .                                      | 6         |
| 2.1.2 State-space formulation . . . . .                             | 10        |
| 2.1.3 Transfer functions . . . . .                                  | 11        |
| 2.1.4 Stability . . . . .   | 12        |
| 2.1.5 Passivity . . . . .   | 12        |
| 2.1.6 Butterworth anti-aliasing filter . . . . .                    | 13        |
| 2.2 Thermal model . . . . .   | 14        |
| 2.2.1 Heat transfer . . . . .                                       | 14        |
| 2.2.2 Stability . . . . .   | 15        |
| 2.3 Resistance models . . . . .                                     | 15        |
| 2.3.1 Contact resistance due to current . . . . .                   | 17        |
| 2.3.2 Winding resistance due to temperature . . . . .               | 18        |
| 2.4 Wiener process as a model for time-varying quantities . . . . . | 18        |
| 2.5 Remarks . . . . .   | 19        |
| <b>3 Analysis</b>   | <b>20</b> |
| 3.1 Existing control structure . . . . .                            | 20        |
| 3.1.1 Controller . . . . .  | 20        |
| 3.1.2 Closed loop transfer functions . . . . .                      | 22        |
| 3.1.3 Stability . . . . .   | 23        |
| 3.2 The effect of parameter variation . . . . .                     | 23        |

|          |   |           |
|----------|---|-----------|
| 3.3      | Discrete analysis . . . . .   | 27        |
| 3.3.1    | Exact discretization . . . . .  | 27        |
| 3.3.2    | Discretization of motor model . . . . .                                       | 28        |
| 3.3.3    | Current feedback . . . . .  | 30        |
| 3.4      | Remarks . . . . .   | 34        |
| <b>4</b> | <b>Estimation</b>   | <b>35</b> |
| 4.1      | Introduction . . . . .  | 35        |
| 4.2      | Gradient method . . . . .   | 35        |
| 4.2.1    | Resistance estimation . . . . .   | 35        |
| 4.3      | Derivation, stability and convergence . . . . .                               | 37        |
| 4.4      | The effect of measurement bias . . . . .                                      | 38        |
| 4.5      | Kalman filters . . . . .  | 38        |
| 4.5.1    | Discrete Kalman filter . . . . .  | 39        |
| 4.5.2    | Extended Kalman filter . . . . .  | 40        |
| 4.5.3    | Estimation of unknown parameters, exogenous inputs, and random bias . . . . . | 41        |
| 4.5.4    | Kalman filter for thermal model . . . . .                                     | 43        |
| 4.5.5    | Motor model with Butterworth filter . . . . .                                 | 43        |
| 4.5.6    | Complete system model . . . . .   | 49        |
| 4.5.7    | Jacobians . . . . .   | 51        |
| 4.5.8    | Observability . . . . .   | 58        |
| 4.6      | Regression with least squares . . . . .                                       | 58        |
| 4.6.1    | Jacobians for resistance models . . . . .                                     | 60        |
| 4.6.2    | Implementation . . . . .  | 60        |
| <b>5</b> | <b>Experiments</b>  | <b>61</b> |
| 5.1      | Laboratory setup . . . . .  | 61        |
| 5.1.1    | Instrumentation . . . . .   | 61        |
| 5.1.2    | Experimental rig . . . . .  | 62        |
| 5.1.3    | Computer setup . . . . .  | 64        |
| 5.1.4    | Position measurement and velocity estimate . . . . .                          | 64        |
| 5.1.5    | Current measurement . . . . .   | 65        |
| 5.1.6    | Thermocouple . . . . .  | 66        |
| 5.2      | Implementation . . . . .  | 68        |
| 5.2.1    | Position control . . . . .  | 69        |
| 5.2.2    | Extended Kalman filter . . . . .  | 70        |
| 5.2.3    | Kalman filter for thermal model . . . . .                                     | 73        |
| 5.2.4    | Extended Kalman filter behavior . . . . .                                     | 73        |
| 5.3      | Performed experiments . . . . .   | 75        |
| 5.3.1    | Winding resistance and its temperature dependency . . . . .                   | 75        |
| 5.3.2    | Contact resistance and its current dependency . . . . .                       | 76        |
| 5.3.3    | Comparison of adaptive schemes . . . . .                                      | 78        |
| 5.4      | Remarks . . . . .   | 80        |

|          |   |            |
|----------|---|------------|
| <b>6</b> | <b>Results and conclusions</b>                              | <b>82</b>  |
| 6.1      | Winding resistance and its temperature dependency . . . . . | 82         |
| 6.2      | Contact resistance and its current dependency . . . . .     | 84         |
| 6.3      | Comparison of adaptive schemes . . . . .                    | 88         |
| 6.3.1    | Remarks . . . . .   | 94         |
| 6.4      | Conclusions . . . . .                                       | 96         |
| 6.5      | Recommendations . . . . .                                   | 96         |
|          | <b>References</b>   | <b>96</b>  |
|          | <b>A Additional figures</b>                                 | <b>99</b>  |
| A.1      | Open loop gain vs. frequency and resistance . . . . .       | 99         |
| A.2      | Closed loop gain vs. frequency and resistance . . . . .     | 100        |
|          | <b>B Contents of DVDs</b>                                   | <b>101</b> |
| B.1      | MATLAB scripts . . . . .                                    | 101        |
| B.1.1    | Analysis . . . . .  | 102        |
| B.1.2    | Experiments . . . . .                                       | 102        |
| B.1.3    | Experimental data processing . . . . .                      | 103        |
| B.1.4    | Nonlinear regression . . . . .                              | 104        |
| B.2      | Extended Kalman filter implementation . . . . .             | 104        |
|          | <b>C Experimental setups</b>                                | <b>107</b> |
|          | <b>D Simulink diagrams</b>                                  | <b>109</b> |

# Chapter 1

## Introduction

### 1.1 Motivation

Electric motors with metal graphite brushes experience a change in contact resistance which is dependent on current load. Figure 1.1 presents graphs of contact resistance vs. current for such brushes. The resistance for both brush polarities (anodic/cathodic) are shown separately and combined. As can be seen, the relationship is nonlinear, leading to a nonlinear  $i$ - $u$  characteristic. This varying resistance leads to varying gain in the motor, since the proportion of power dissipated due to the resistance changes.

If such a motor is going to be used for force/torque control, a current controller is typically used to generate the actuation voltage, since the torque output of the motor is proportional to the armature current. If a discrete controller is used with current feedback, the sampling time must match the electrical time constant of the motor, otherwise it will be unstable for high gains in the controller. Since the electrical time constant is typically very small, a discrete controller with current feedback is not practical. An alternative is to use feedforward to compute the actuation voltage, but this requires good estimates of the motor's parameters in order to be accurate. Hence some adaptive scheme is needed for feedforward to work well.

Presently, a least mean squares estimator is used to adapt to the changing resistance. A general impression is that estimators work better for slowly varying parameters than quickly varying ones, because of convergence rates and periods of transients that yield parameter estimation error.

The resistance attenuation phenomenon seems deterministic and can lead to large variations in short periods of time, especially at low currents, where the tangent to the graph in Figure 1.1 is steep. At low currents we have input/output values of small magnitude and relatively larger noise magnitude. Using an estimator under such conditions, one can typically expect inaccurate estimates and slow convergence rates.

A static model of the phenomenon yielding the parameter change could potentially give a better estimate, if the model and corresponding measurements are accurate enough. Both the estimator-based and the gain-scheduler-based scheme should benefit from measurement signal conditioning to remove noise and bias.

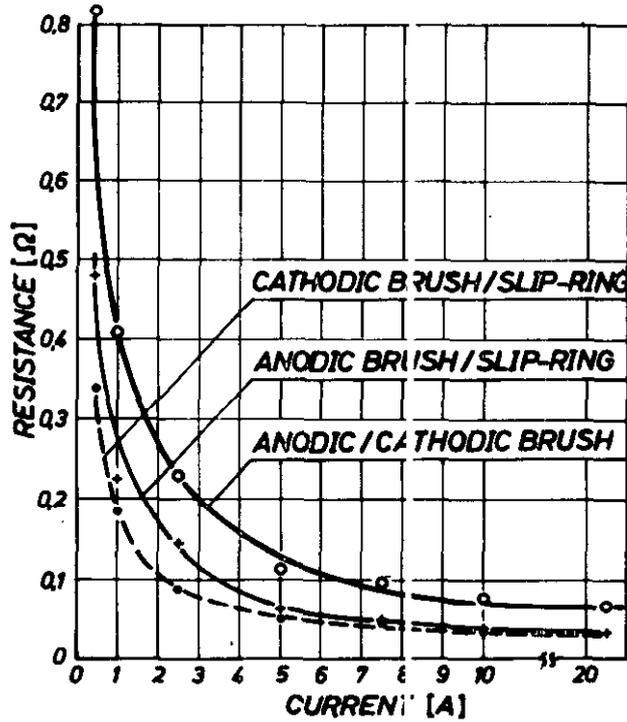


Figure 1.1: Contact resistance as a function of current [3].

## 1.2 Objectives

The main objective of this investigation is to find a model for the resistance attenuation phenomenon in electric motors with metal graphite brushes. Different adaptive schemes for discrete force/current control for such motors will also be compared. In pursuit of these goals, the following tasks were performed:

- Description and analysis of the existing system.
- Suggestion of models that can describe the resistance attenuation phenomenon.
- Construction of an experimental rig with necessary instrumentation.
- Development and implementation of a scheme to perform signal conditioning and obtain the resistance's current dependence from input/output data.
- Conduction of experiments with the developed scheme.
- Use of the experimental data to identify the parameters in the resistance attenuation models using nonlinear regression.
- Comparison of the existing adaptive scheme with the new scheme, and with a gain-scheduler based on the best-fit resistance attenuation model.

## 1.3 Outline of the report

This report is split into 6 chapters:

**Chapter 1** Introduction to the problem, outline of the tasks performed, the structure of the report, and a brief overview of the system.

**Chapter 2** Development and some analysis of significant physical models for the system. Models are found for the electric motor and the thermal dynamics of the motor. Models for the resistance attenuation phenomenon are proposed. The Wiener process as a model for time-varying quantities is briefly described.

**Chapter 3** In-depth description and analysis of the system, showing stability limits for the control structure in use, sensitivity to parameter variation, and why current feedback does not work well on this system.

**Chapter 4** Description and analysis of the gradient method-based/least mean squares parameter estimator that is currently in use, and development of an extended Kalman filter for signal conditioning and parameter estimation. Description and development of a nonlinear regression method for identifying resistance model parameters.

**Chapter 5** Description of the experimental setup, implementation of the estimators, and the performed experiments.

**Chapter 6** Discussion of the results from the experiments, and conclusions.

**Appendices** Additional figures and Simulink diagrams. Explanation of some of the experimental setups. Provides a description of the various MATLAB scripts used, the source code for the extended Kalman filter implementation, and the location of the different files on the accompanying digital versatile discs (DVDs.)

Portions of this report are based on work done on a project during the Fall of 2006. This includes some of the modeling and analysis, as well as the nonlinear regression method. The main output of the project was a simulator for the system. This simulator has been used during the work on this thesis to validate and debug the extended Kalman filter.

## 1.4 Brief description of the servomechanism

The servomechanism discussed in this report is going to be used in a robotic colonoscope. The colonoscope consists of several links connected by rotational joints that are actuated by cables that run from a rack of gearing mechanisms (called leadscrews), which are actuated by electric motors. The objective for the colonoscope is to shape itself and follow the path traced by the tip, which is controlled by a gastroenterologist during a procedure. The torque output from the motors, and thus the pull force from the

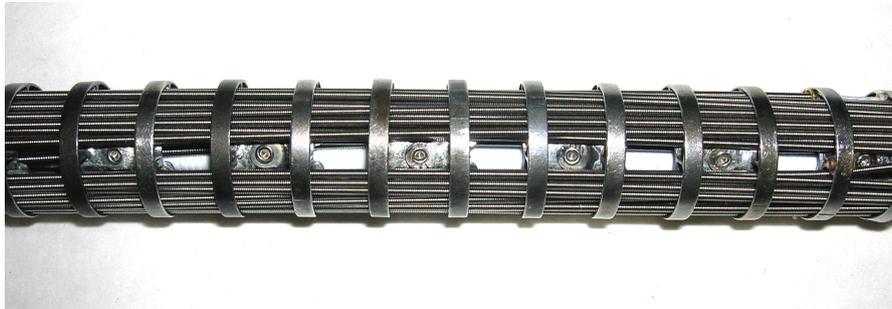
cables, is what causes the deflection of the colonoscope at the joints. The shape of the colonoscope is therefore determined by the torque output of the motors. There is also a safety aspect to consider. One wishes to limit the maximum torque output of the motors to prevent the colonoscope from exerting excessive force on the colon wall, in order to prevent injuries. The servomechanism therefore operates with a maximum current limit ( $i_{max} = 0.7$  [A].)

The colonoscope itself consists of 9 segments or links with rotational joints actuated by 36 cables (4 per segment) running through the colonoscope to a rack of direct current (DC) motors and gears, called the actuation control unit (ACU). Pictures of the colonoscope and the ACU are provided in Figures 1.3(a), 1.3(b) and 1.3(c).



Figure 1.2: Segment Control Unit (SCU).

The servomechanism is located on a segment control unit (SCU), shown in Figure 1.2, to which four DC motors are connected. The SCU is a circuit board with, among other things, a Texas Instruments C2000 digital signal processor (DSP), which implements the controller. Each of the DC motors is equipped with an encoder that measures the angular position of its rotor. Additionally, the angular position of a rotor determines the length of cable that has been pulled by the attached leadscrew. The SCU also contains H-bridge pulse width modulation (PWM) amplifiers to produce the voltage actuation signal, current sensors, and encoder decoders. The current sensors produce a voltage signal which is filtered by a Butterworth low-pass filter to avoid anti-aliasing when sampling.



(a) Colonoscope detail (without skin)



(b) Colonoscope (without skin)



(c) Actuation Control Unit (ACU)

Figure 1.3: Colonoscope and actuation control unit (ACU).

# Chapter 2

## Modeling

In order to perform analysis and to develop estimators and controllers for the servomechanism, models are needed for each of its components and for the physical phenomena which occur. This chapter presents the necessary models.

### 2.1 Motor model

#### 2.1.1 Physical model

To model an electric motor, some equations from elementary electromagnetism are needed. The magnetic force working in the motor is given by the Lorentz force [8, 18].

$$\vec{F} = q[\vec{E} + (\vec{v}_q \times \vec{B})]. \quad (2.1)$$

$\vec{F}$  is the force,  $q$  is charge,  $\vec{E}$  is the electric field strength vector,  $\vec{v}_q$  is the velocity of the charge and  $\vec{B}$  is the magnetic flux density vector.

Current intensity through a surface is defined as:

$$I \triangleq \frac{dq_S}{dt}$$

If a charge is moving in a segment of wire with a velocity  $\vec{v}_q$ , its displacement in time  $dt$  will be  $d\vec{\ell} = \vec{v}_q dt$ , thus:

$$d\vec{v}_q = \frac{d\vec{\ell}}{dt}$$

For the type of motor we are modeling, there should ideally only be a magnetic field in the space where the segment of wire is situated. Therefore, we can ignore the electric field,  $\vec{E} = 0$ . Finding the charge in the segment of wire to be  $dq_S = I dt$ , we can find the force on the segment of wire as

$$d\vec{F} = I dt \left( \frac{d\vec{\ell}}{dt} \times \vec{B} \right) \Rightarrow d\vec{F} = I(d\vec{\ell} \times \vec{B}).$$

We can now find the total force for a wire loop (which should sum to zero in a uniform magnetic field) as

$$\vec{F} = \oint_C I d\vec{\ell} \times \vec{B}$$

and the torque working on the wire loop will be

$$d\vec{T} = \vec{r} \times d\vec{F} \Rightarrow \vec{T} = \oint_C \vec{r} \times (I d\vec{\ell} \times \vec{B}).$$

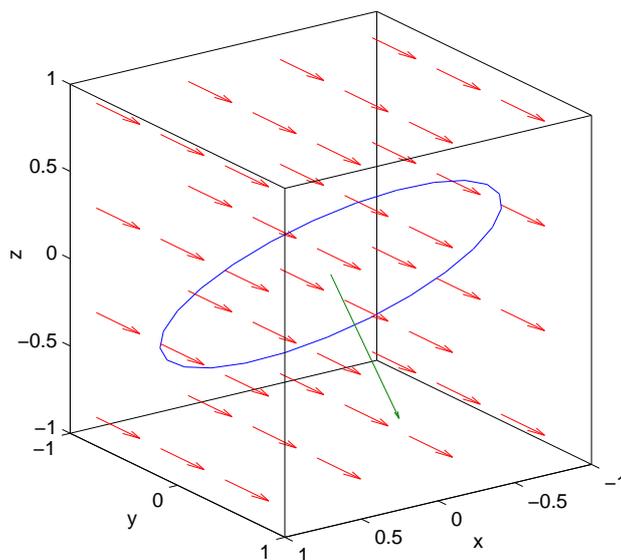


Figure 2.1: Circular loop, surface normal and magnetic field.

For a circular loop of radius  $a$  with center at the origin revolving around the  $x$ -axis (with an angle  $\phi$ ), we have the parameterization

$$\vec{r}_{circle} = (a \cos u, a \sin u \cos \phi, a \sin u \sin \phi)^\top.$$

The tangent (velocity) to this parameterization is

$$\frac{d\vec{\ell}}{du} = \frac{d\vec{r}_{circle}}{du} = (-a \sin u, a \cos u \cos \phi, a \cos u \sin \phi)^\top.$$

The moment arm for the force will in this case be

$$\vec{r}_{arm} = \vec{r}_{circle} \cdot \vec{j} = (0, a \sin u \cos \phi, 0)^\top,$$

assuming we have a uniform magnetic field  $\vec{B} = (0, B, 0)^\top$  ( $\vec{j}$  is the unit vector along the  $y$ -axis.) If we further assume uniform distribution of the current intensity  $I$ , we can find the total torque as:

$$\begin{aligned}\vec{T} &= \oint_C \vec{r}_{arm} \times (I d\vec{\ell} \times \vec{B}) = \\ &= \int_0^{2\pi} \vec{r}_{arm} \times (-IaB \cos u \sin \phi, 0, -IaB \sin u)^\top du = \\ &= \int_0^{2\pi} (-Ia^2B \sin^2 u \cos \phi, 0, -Ia^2B \sin u \cos u \sin \phi \cos \phi)^\top du = \\ &= (-Ia^2\pi B \cos \phi, 0, 0)^\top = (-ISB \cos \phi, 0, 0)^\top = I\vec{S} \times B \quad (2.2)\end{aligned}$$

where  $\vec{S} = (0, a^2\pi \sin \phi, -a^2\pi \cos \phi)^\top$  is the surface normal to the disc spanned by the circle (and  $|\vec{S}| = a^2\pi$ .)

As the loop turns in the magnetic field, the magnetic flux through the loop will vary, inducing a electromotive force defined by Faraday's Law, [8, 18]:

$$e = \oint_C \vec{E} \cdot d\vec{\ell} = -\frac{d\Phi}{dt} = -\frac{d}{dt} \int_S \vec{B} \cdot d\vec{S} \quad (2.3)$$

The disc spanned by the circle is parameterized by

$$\vec{r}_{disc} = (v \cos u, v \sin u \cos \phi, v \sin u \sin \phi)^\top$$

and its surface area element is

$$d\vec{S} = \frac{\partial \vec{r}_{disc}}{\partial u} \times \frac{\partial \vec{r}_{disc}}{\partial v} = (0, v \sin \phi, -v \cos \phi)^\top.$$

The magnetic flux through this surface is

$$\int_S \vec{B} \cdot d\vec{S} = \int_0^{2\pi} \int_0^a vB \sin \phi dv du = \int_0^{2\pi} \frac{1}{2} a^2 B \sin \phi du = a^2\pi B \sin \phi.$$

If the wire loop revolves around the  $x$ -axis as a function of time, we can use the following relation to express it with angular frequency:

$$\phi = \omega t$$

We now find the electromotive force induced in the loop as

$$e = -\frac{d}{dt}(a^2\pi B \sin \omega t) = -a^2\pi B \omega \cos \omega t = -SB\omega \cos \omega t. \quad (2.4)$$

This system is illustrated in Figure 2.1, where the circular loop, unit surface normal, and magnetic field are plotted.

If we add commutation to rectify the current in one direction (that is, we take the absolute value) and add loops spaced radially around the turning axis with an angle

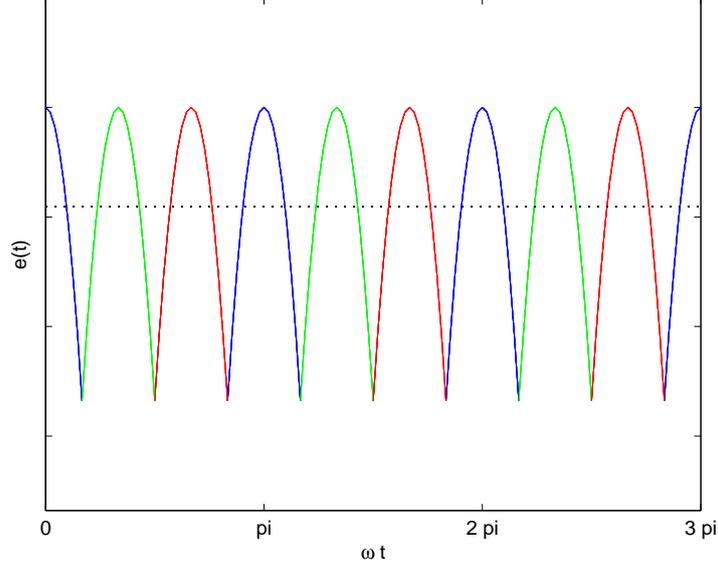


Figure 2.2: Electromotive force induced in loops.

of  $\frac{2}{3}\pi$ , we will get an induced electromotive force as shown in Figure 2.2. As can be seen, loops added with decreasing angle between them will result in an almost constant induced electromotive force. The result is that we can approximate both the torque and induced electromotive force as linear relations.

The torque is given by (changing notation for the current,  $I = i$ )

$$T = (| -SB \cos \omega t |) i$$

and the electromotive force is given by

$$e = (| -SB \cos \omega t |) \omega.$$

The terms in the parentheses are assumed to be constant due to commutation, and we get the linear relations:

$$T = K_T i \tag{2.5}$$

$$e = K_E \omega \tag{2.6}$$

As seen in the above equations, the two constants  $K_T$  and  $K_E$  are the same. This can also be shown as follows [7]: the electrical power is equal to  $P_E = ei = K_E \omega i$  and the mechanical power is  $P_T = \omega T = \omega K_T i$ . Assuming steady-state, it follows that

$$K_E \omega i = P_E = P_T = \omega K_T i \quad \Rightarrow \quad K_E = K_T.$$

To develop a full dynamic model, we can start by finding the governing equation for the current in the armature. Assuming there is a resistance  $R$ , a voltage source  $u$ , and a self-inductance  $L$ , we can use Faraday's law again [2] to obtain

$$\oint_C \vec{E} \cdot d\vec{l} = -u + e + Ri = -L \frac{di}{dt}. \quad (2.7)$$

The moment balance for the rotor can be expressed as [7]

$$J\dot{\omega} = T - T_L. \quad (2.8)$$

Here  $J$  is the moment of inertia of the rotor and  $T_L$  is the torque load (or disturbance.) Together with the phase variable  $\dot{\theta} = \omega$  we can set up a state-space representation of the system:

$$L \frac{di}{dt} = -Ri - K_E \omega + u \quad (2.9)$$

$$J\dot{\omega} = K_T i - T_L \quad (2.10)$$

$$\dot{\theta} = \omega \quad (2.11)$$

The system is also presented as a block diagram in Figure 2.3.

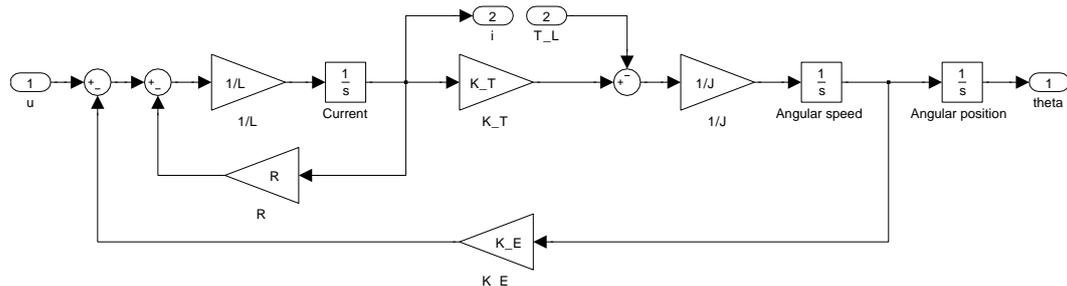


Figure 2.3: DC-motor block diagram.

The nominal electrical and mechanical parameter values stated by the motor manufacturer which are used in this investigation are given in Table 2.1, where they are converted to SI units.

### 2.1.2 State-space formulation

Equations (2.9), (2.10) and (2.11) can be put in a standard state-space formulation as follows:

$$\begin{aligned} \dot{x} &= Ax + Bu + B_d d \\ y &= Cx + Du \end{aligned} \quad (2.12)$$

$$\dot{x} = \begin{bmatrix} \frac{di}{dt} \\ \dot{\omega} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_E}{L} & 0 \\ \frac{K_T}{J} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} \cdot u + \begin{bmatrix} 0 \\ -\frac{1}{J} \\ 0 \end{bmatrix} \cdot T_L$$

Table 2.1: Nominal parameters for Maxon RE 36.118800

| Parameter | Value                  | Units                | Description         |
|-----------|------------------------|----------------------|---------------------|
| $L$       | $0.487 \times 10^{-3}$ | [H]                  | Terminal inductance |
| $R$       | 2.74                   | [ $\Omega$ ]         | Terminal resistance |
| $K_T$     | $56.6 \times 10^{-3}$  | [Nm/A]               | Torque constant     |
| $K_E$     | $K_T$                  | [V/rad·s]            | Speed constant      |
| $J$       | $6.78 \times 10^{-6}$  | [kg m <sup>2</sup> ] | Rotor inertia       |

$$y = \begin{bmatrix} i \\ \omega \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \cdot u$$

For the time being, we will assume that all the states are available for measurement.

### 2.1.3 Transfer functions

The Laplace transform of (2.12) can be found as:

$$\begin{aligned} s\hat{x} &= A\hat{x} + B\hat{u} + B_d\hat{d} \Rightarrow \\ (sI - A)\hat{x} &= B\hat{u} + B_d\hat{d} \Rightarrow \\ \hat{x} &= (sI - A)^{-1}B\hat{u} + (sI - A)^{-1}B_d\hat{d} \end{aligned}$$

Finding the Laplace transform of  $y$ , and inserting  $\hat{x}$  into the expression gives us:

$$\begin{aligned} \hat{y} &= C\hat{x} + D\hat{u} \Rightarrow \\ \hat{y} &= C[(sI - A)^{-1}B\hat{u} + (sI - A)^{-1}B_d\hat{d}] + D\hat{u} \Rightarrow \\ \hat{y} &= [C(sI - A)^{-1}B + D]\hat{u} + [C^\top(sI - A)^{-1}B_d]\hat{d} \end{aligned}$$

We now set  $\hat{y} = G(s)\hat{u}$  to be

$$G(s) = C(sI - A)^{-1}B + D \quad (2.13)$$

and  $\hat{y} = G_d(s)\hat{d}$  to be

$$G_d(s) = C(sI - A)^{-1}B_d. \quad (2.14)$$

This gives us all the transfer functions from the input  $u$  and the disturbance  $T_L$  to the states  $x = (i, \omega, \theta)^\top$ :

$$G(s) = \begin{bmatrix} \frac{sJ}{(s^2JL+sJR+K_TK_E)} \\ \frac{K_T}{(s^2JL+sJR+K_TK_E)} \\ \frac{K_T}{s(s^2JL+sJR+K_TK_E)} \end{bmatrix} \quad (2.15)$$

$$G_d(s) = \left[ \begin{array}{c} \frac{K_E}{(s^2 JL + sJR + K_T K_E)} \\ -\frac{sL + R}{(s^2 JL + sJR + K_T K_E)} \\ -\frac{sL + R}{s(s^2 JL + sJR + K_T K_E)} \end{array} \right] \quad (2.16)$$

### 2.1.4 Stability

Looking at Equations (2.15) and (2.16), we can see that the roots of the common denominator shared by all the transfer functions,

$$s^2 JL + sJR + K_T K_E = 0,$$

are given by

$$s = -\frac{JR \pm \sqrt{J^2 R^2 - 4JLK_T K_E}}{2JL}.$$

The real parts of the roots will always be negative if the parameters are positive and real, so the system is inherently stable.

$$-JR + \sqrt{J^2 R^2 - 4JLK_T K_E} > 0 \Rightarrow -4JLK_T K_E \not\geq 0$$

For the transfer functions to  $\theta$ , there is a root at zero because of the integrator from the angular velocity to the angular position. These transfer functions are thus marginally stable.

### 2.1.5 Passivity

The total energy for the motor is given by the energy stored in the inductor and the kinetic energy of the motor's shaft, so the storage function [13] will be:

$$V = \frac{1}{2}Li^2 + \frac{1}{2}J\omega^2 \geq 0 \quad (2.17)$$

The total derivative of (2.17) will then be, given Equations (2.9), (2.10) and  $K_E = K_T$ :

$$\dot{V} = iL \frac{di}{dt} + \omega J \dot{\omega} = i(-Ri - K_E \omega + u) + \omega(K_T i - T_L) = -Ri^2 + iu - \omega T_L$$

We know the resistance varies with current. If the load is a spring-damper system, it should be dependent on angular velocity and position. Then we get

$$\dot{V} = ui - R(i)i^2 - \omega T_L(\omega, \theta).$$

If  $u_R(i) = R(i)i$  and  $T_L(\omega, \theta)$  are passive, the system is passive. If  $R(i)$  is positive and real,  $R(i)i^2 \geq 0$  and is therefore passive.

A spring-damper system might be described as

$$T_L(\omega, \theta) = K_s \theta + K_d \omega,$$

with  $K_s$  and  $K_d$  being the spring constant and the damping constant respectively. Augmenting (2.17) with the energy stored in the spring, the storage function will be

$$V = \frac{1}{2}Li^2 + \frac{1}{2}J\omega^2 + \frac{1}{2}K_s\theta^2 \geq 0.$$

The total derivative will now be

$$\dot{V}_L = iL\frac{di}{dt} + \omega J\dot{\omega} + \theta K_s\omega = ui - R(i)i^2 - K_d\omega^2.$$

If  $K_d$  is positive and real,  $K_d\omega^2 \geq 0$  and is therefore passive, yielding an overall passive system.

We may therefore conclude that, given a passive load, as long as the value for the resistance is positive and real, the system will be passive. With estimation in mind, one must be sure that an estimated resistance will always satisfy these criteria.

### 2.1.6 Butterworth anti-aliasing filter

The current measurement on the SCU and in the experimental instrumentation is filtered by a second-order Butterworth filter with a cut-off frequency of  $\omega_c = 2\pi \cdot 100$  [rad/s] to avoid anti-aliasing.

A second-order, low-pass, unity-gain Butterworth filter with cut-off frequency,  $\omega_c$ , is described by the following transfer function [15]:

$$G(s) = \frac{\omega_c^2}{s^2 + \omega_c\sqrt{2}s + \omega_c^2} \quad (2.18)$$

The poles of the transfer function are:

$$s_{1,2} = -\omega_c \left( \frac{\sqrt{2}}{2} \pm i \frac{\sqrt{2}}{2} \right)$$

Both poles have negative real parts,  $\text{Re}[s_{1,2}] < 0$ , so the filter is stable.

A state-space realization of this transfer function can found as:

$$\begin{aligned} \dot{x}_f &= A_f x_f + B_f i \\ i_f &= C_f x_f \end{aligned} \quad (2.19)$$

$$\begin{aligned} \dot{x}_f = \begin{bmatrix} \frac{d}{dt} i'_f \\ \frac{d}{dt} i_f \end{bmatrix} &= \omega_c \cdot \begin{bmatrix} -\sqrt{2} & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} i'_f \\ i_f \end{bmatrix} + \omega_c \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot i \\ i_f &= \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i'_f \\ i_f \end{bmatrix} \end{aligned}$$

The Butterworth filter influences the observed current dynamics. Using the current measurement without accounting for the Butterworth filter renders both the extended Kalman filter and the gradient method-based/least mean squares parameter estimator unstable.

## 2.2 Thermal model

### 2.2.1 Heat transfer

A simple model for the heat transfer in the motor can be developed using the parameters from the datasheet provided by the manufacturer, given that the mounting of the motor is similar to that used to determine the parameters. It is stated that the parameters were found while the motors were end-mounted to a plastic plate, which should be similar to the conditions used during the experiments (see Figure 5.2(b).) The thermal data is presented in Table 2.2.

Table 2.2: Parameters for Maxon RE 36.118800

| Parameter | Value | Dimension | Description                         |
|-----------|-------|-----------|-------------------------------------|
| $\tau_1$  | 44.2  | [s]       | Thermal time constant, winding      |
| $\tau_2$  | 1120  | [s]       | Thermal time constant, motor        |
| $R_{th1}$ | 3.4   | [K/W]     | Thermal resistance, winding-housing |
| $R_{th2}$ | 6.4   | [K/W]     | Thermal resistance, housing-ambient |

The model is based on lumped thermal capacity and thermal resistance, and is very similar to an electric circuit with capacitors and resistors. Temperature will then be analogous to voltage, and heat will be analogous to current. Since the time constant of a RC-circuit will in either case be given by  $\tau = RC$ , we should be able to find the thermal capacities based on the given data as follows:

$$C_{th1} = \frac{\tau_1}{R_{th1}}, \quad C_{th2} = \frac{\tau_2}{R_{th2}}$$

The heat generated in the motor will be due to Joule losses [8] in the winding (ignoring losses due to friction and other dissipative effects.) The Joule losses should then be equal to the difference between the electrical power that is inputted to the motor, and the mechanical power outputted from the motor, or equal to power dissipated by the motor's electrical resistance. Thus

$$Q_J = R(i)i^2 = P_E - P_T = ui - T_L\omega,$$

where  $Q_J$  is heat, with dimensions [J/s] or [W].

The model below was adapted from [14]. It is a second-order system that gives us the transient behavior of the winding temperature,  $T_w$ , and the housing temperature,  $T_h$ , which we can measure.

$$\begin{aligned} C_{th1} \frac{dT_w}{dt} &= -\frac{1}{R_{th1}} (T_w - T_h) + Q_J \\ C_{th2} \frac{dT_h}{dt} &= -\frac{1}{R_{th2}} (T_h - T_\infty) + \frac{1}{R_{th1}} (T_w - T_h) \end{aligned}$$

We can put this in a state-space formulation, using Joule losses and the ambient temperature,  $T_\infty$ , as inputs:

$$\dot{T} = A_{th}T + B_{th}u \quad (2.20)$$

$$\begin{bmatrix} \dot{T}_w \\ \dot{T}_h \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_{th1}C_{th1}} & -\frac{1}{R_{th1}C_{th1}} \\ \frac{1}{R_{th1}C_{th2}} & -\left(\frac{1}{R_{th1}C_{th2}} + \frac{1}{R_{th2}C_{th2}}\right) \end{bmatrix} \cdot \begin{bmatrix} T_w \\ T_h \end{bmatrix} + \begin{bmatrix} \frac{1}{C_{th1}} & 0 \\ 0 & \frac{1}{R_{th2}C_{th2}} \end{bmatrix} \cdot \begin{bmatrix} Q_J \\ T_\infty \end{bmatrix}$$

## 2.2.2 Stability

To check the stability of the above system we can use Hurwitz's criterion [1]. We construct a matrix of the coefficients from the characteristic polynomial of the state matrix. The characteristic polynomial is given by:

$$|\lambda I - A_{th}| = \lambda^2 + \left( \frac{R_{th1}C_{th1} + R_{th2}C_{th1} + R_{th2}C_{th2}}{R_{th1}R_{th2}C_{th1}C_{th2}} \right) \lambda + \frac{1}{R_{th1}R_{th2}C_{th1}C_{th2}}$$

The constructed matrix becomes:

$$H = \begin{bmatrix} \frac{R_{th1}C_{th1} + R_{th2}C_{th1} + R_{th2}C_{th2}}{R_{th1}R_{th2}C_{th1}C_{th2}} & 0 \\ 1 & \frac{1}{R_{th1}R_{th2}C_{th1}C_{th2}} \end{bmatrix}$$

The system is stable if  $H$  is positive definite. A positive definite matrix has positive principal minors. All the parameter values are positive and real, so the following inequalities for the principal minors hold:

$$\begin{aligned} \frac{R_{th1}C_{th1} + R_{th2}C_{th1} + R_{th2}C_{th2}}{R_{th1}R_{th2}C_{th1}C_{th2}} &> 0 \\ \frac{R_{th1}C_{th1} + R_{th2}C_{th1} + R_{th2}C_{th2}}{(R_{th1}R_{th2}C_{th1}C_{th2})^2} &> 0 \end{aligned}$$

Thus, the system is stable, as would be expected.

## 2.3 Resistance models

From [3] we know that the contact resistance in an electrical motor with metal graphite brushes and slipping will vary with current, temperature, and brush wear,  $w$  [m]. Figure 1.1 shows the dependence on current, and Figure 2.4 presents the contact voltage vs. temperature, where the voltage drops with increasing temperature, indicating decreasing resistance. The manufacturer of the motors used during the experiments also states that the terminal resistance of the motor will vary with rotational position of the rotor, due to commutation.

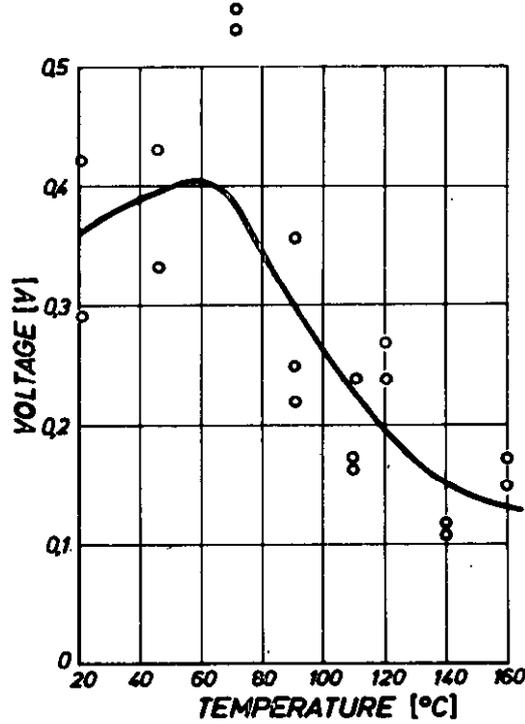


Figure 2.4: Contact voltage between brushes as a function of temperature [3].

From [8] we also know the resistivity in many conductors varies with temperature. Thus we know that the terminal resistance of the type of motor considered will vary with current, temperature, position, and wear.

The terminal resistance will most likely be a combination of the contact resistance and the resistance in the motor's windings:

$$R_{terminal}(i, T, \theta, w) = R_{contact}(i, T, \theta, w) + R_{winding}(T)$$

The contact resistance is formed by the system of metal graphite brushes and slipring.

We have chosen to account for the variation of contact resistance due to current. Our goal was to express the terminal resistance as a function of armature current,  $R(i)$ . Experience with the motor has shown that the variation of the terminal resistance will range between approximately 25 [ $\Omega$ ] and 2.5 [ $\Omega$ ], for low, 0.044 [A], and high, 2.3 [A], current respectively, when the motor is at room temperature. This variation can be attributed to armature current. The dominating behavior thus seems to be current dependent. We have not attempted to model the *contact resistance's* dependence on temperature. The contact resistance's dependence on position, wear, and other unknown effects was assumed to be negligible.

While performing the experiments, we have chosen to account for the *winding resistance's* dependence on temperature. At high currents, the heat produced in the motor by

Joule losses is significant, leading to a increase of up to about 1  $[\Omega]$ , which can be larger than the variation of the contact resistance at these currents. The terminal resistance has thus been assumed to be:

$$R_t(i, T) = R_c(i) + R_w(T)$$

Since no temperature measurement is available in the system where the controller is going to be used, one might try to account for the variation due to temperature for both the contact and winding resistance using some online adaptive scheme. This should also compensate for variation due to rotor position and brush wear.

### 2.3.1 Contact resistance due to current

Looking at Figure 1.1 and assuming the behavior is correct, we can postulate that the change in resistance vs. current can be described as a convex function that has asymptotic behavior as the current goes to infinity. The resistance should be bounded. To insure the passivity of the motor model, the output of the function should also always be positive and real. A function that has these characteristics is the exponential function,

$$S_1(x) = ae^{bx} + c \quad (2.21)$$

if  $b < 0$ ,  $a > 0$ ,  $c > 0$ ,  $x \in \mathbb{R}$ , and  $x \geq 0$ . This function will start at

$$\lim_{x \rightarrow 0} S(x) = a + c,$$

and end at

$$\lim_{x \rightarrow \infty} S(x) = c.$$

Exponential functions like this are often approximated using Padé approximants, which are rational expressions with polynomials in the numerator and the denominator. Using a rational function with polynomials of the same order, we should get similar behaviour as (2.21), but the shape will have different characteristics that might be desirable.

When taking into consideration the use of these models for finding a fit to a set of experimentally obtained data, as described in Section 4.6, it should be mentioned that exponential models like (2.21) are known to have poor statistical properties [20]. Rational functions should have better statistical properties.

The Padé approximant to (2.21) with order  $L = 1$  in the numerator and order  $M = 1$  in the denominator developed around the point  $x_0 = d$  is

$$S_2(x) = \frac{2(ae^{bd} + c) + b(ae^{bd} - c)(x - d)}{2 - b(x - d)} = \frac{\beta + \gamma x}{1 + \alpha x}, \quad (2.22)$$

and we must require that  $\beta > 0$ ,  $x \in \mathbb{R}$ , and  $x \geq 0$ . This function starts at,

$$\lim_{x \rightarrow 0} S_2 = \beta$$

and ends at:

$$\lim_{x \rightarrow \infty} S_2 = \frac{\gamma}{\alpha}$$

As can be seen, this model can have the same number of parameters as (2.21) when lumping the original parameters  $a, b, c, d$ , into  $\alpha, \beta, \gamma$ . It should be noted that the rational model will have different behavior when performing regression depending on the parameterization. It is also the latter parameterization that yield better statistical properties, and it is the parameterization that is used henceforth.

### 2.3.2 Winding resistance due to temperature

Resistivity in some materials has a temperature dependence [8]. Copper and graphite are two such materials. For these two materials this dependency is mostly linear, at least within the temperature ranges specified for the motors used during this work. It should be safe to assume that the electrical resistance will vary linearly with temperature according to the following model:

$$R = R_0 [1 + \alpha (T - T_0)], \quad (2.23)$$

where  $\alpha$  is the temperature resistance coefficient for the material and  $R_0$  is the resistance at temperature  $T_0$ . For copper  $\alpha_{Cu} = 0.0039$  [1/K], and for carbon (graphite)  $\alpha_C = -0.0005$  [1/K], [8]. For the purposes of this investigation, we have only considered the temperature dependency of copper, that is, the motor's windings. This is because we do not have an estimate of brush temperature, and the value of the temperature resistance coefficient for carbon is about an order of magnitude lower than that of copper. Additionally, it is not known if this value is directly applicable to a given metal graphite composite material.

## 2.4 Wiener process as a model for time-varying quantities

A Wiener, or Brownian-motion, process is often used to model quantities that change with time. In the system under consideration we have such quantities in the form of unknown random bias, unknown exogenous input, and varying model parameters. A Wiener process can briefly be described as being the output of an integrator driven with Gaussian white noise [4]. A state-space description will therefore be

$$\dot{x} = w,$$

where  $w$  is a Gaussian white noise process. In the discrete case, the above equation becomes

$$x_{k+1} = x_k + w_k,$$

where  $w_k$  is a Gaussian white noise sequence. The first and second order statistics are given by

$$E[x] = 0$$

and

$$E[x^2] = t$$

given that the process is ergodic. Since the variance increases with time, it is not a stationary process.

## 2.5 Remarks

The motor we used has a permanent magnet. The electromagnetic constant is determined by the winding geometry and the strength of the magnet. Ideally, these should be constant. The geometry can safely be assumed to be constant, but the magnetic field can be weakened by temperature and demagnetization. The field weakens with higher temperature. No significant variability of the electromagnetic constant was observed (see Section 5.2.2.)

Commutation yields ripple in the armature current. The ripple can be seen as noise, and increases with velocity.

# Chapter 3

## Analysis

### 3.1 Existing control structure

The structure of the existing controller is shown in Figure 3.1. The objective for the control loop is to track position and control the torque output. The proportional derivative (PD) controller produces a current reference, based on the feedback of an angular position measurement from an encoder, and an angular velocity estimate, produced as the first difference of the encoder signal. The current reference is ideally proportional to the desired torque, as in Equation (2.5).

The adaptive control law uses an estimate of the varying resistance to calculate the output voltage corresponding to the current reference. The calculated voltage is then converted to an equivalent pulse width modulated (PWM) signal which actuates the motor. Due to implementation aspects, the voltage output is delayed one sample instant. The estimator uses the velocity estimate and a current measurement, filtered by a Butterworth low-pass filter to avoid anti-aliasing, in addition to the output voltage.

The parameter estimator is described in Section 4.2. The reason current feedback and a simple proportional (P) regulator with high gain is not used to control the current (instead of the adaptive law), is that although such a controller for a DC motor in the continuous case has an infinite gain margin [7], the discretization makes this margin very small for this system, and it therefore does not give a small enough error. This is shown in Section 3.3.3. A proportional integral (PI) controller could be used to let the error go to zero asymptotically, even though the proportional term has a small gain. However, both because its performance is not on par with the feedforward controller, and in order to avoid any problems related to integral windup, it is not used (see Section 3.3.3 and 6.3.)

#### 3.1.1 Controller

The PD controller used to generate the current reference is given by

$$i_r = K_p e + K_d \dot{e}. \quad (3.1)$$

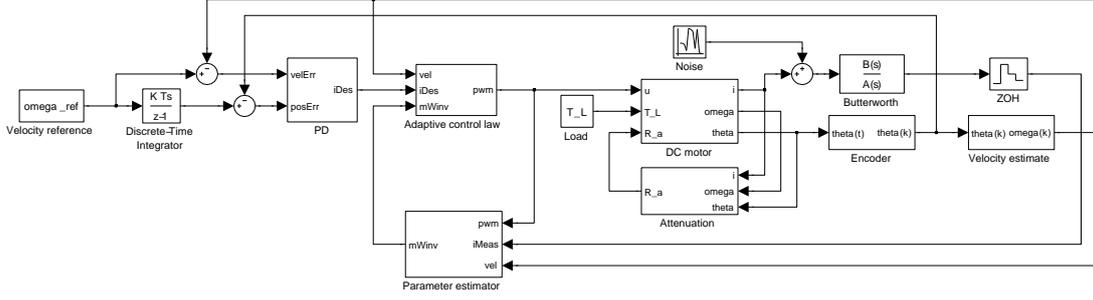


Figure 3.1: Existing control structure.

The current reference is then fed into the adaptive control law, which looks like this:

$$u = Ri_r + K_E\omega \quad (3.2)$$

The electromotive force generated in the motor,  $K_E\omega$ , is added to the voltage drop over the resistance. Given perfect estimates of  $K_E$  and  $R$ , this should yield the correct voltage to produce a current equal to the current reference. This is, in other words, feedforward control.

Combined, the control signal is

$$u = R(K_p e + K_d \dot{e}) + K_E\omega. \quad (3.3)$$

To find the gains of the PD controller, (3.1) is inserted into the relation  $T = K_T i$ , setting  $\theta_r = 0$ :

$$T = K_T i = K_T(-K_p\theta - K_d\dot{\theta}) \Rightarrow i = -K_p\theta - K_d\dot{\theta}$$

Using (2.10) and setting  $T_L = 0$ , this yields:

$$J\dot{\omega} = K_T i = -K_p\theta - K_d\dot{\theta}$$

Using (2.11), this is then equated to zero

$$\ddot{\theta} + \frac{K_d}{J}\dot{\theta} + \frac{K_p}{J}\theta = 0 \quad (3.4)$$

and is compared to the characteristic equation for a harmonic oscillator, typically expressed as:

$$\lambda^2 + 2\zeta\omega_0\lambda + \omega_0^2 = 0 \quad (3.5)$$

If we select a desired natural frequency,  $\omega_0$ , and damping factor,  $\zeta$ , we can compare (3.4) with (3.5) and get expressions for  $K_p$  and  $K_d$ :

$$K_p = \omega_0^2 J \quad (3.6)$$

$$K_d = 2\zeta\omega_0 J \quad (3.7)$$

### 3.1.2 Closed loop transfer functions

The Laplace transforms of Equations (2.9), (2.10) and (2.11) are:

$$sL\hat{i} = -R\hat{i} - K_E\hat{\omega} + \hat{u} \quad (3.8)$$

$$sJ\hat{\omega} = K_T\hat{i} - \hat{T}_L \quad (3.9)$$

$$s\hat{\theta} = \hat{\omega} \quad (3.10)$$

To find the closed loop transfer functions, we take the Laplace transform of, and insert Equations (3.1), (3.2) and (3.3) into the above equations.  $R^*$  is the actual value of the resistance,  $R$  is the estimate,  $w^*$  is the actual value of the angular velocity and  $w$  is the estimate. (The velocity is estimated in the implementation as a first difference, which is noisy.)

Inserting (3.2) into (3.8) we get:

$$\begin{aligned} sL\hat{i} &= -R^*\hat{i} - K_E\hat{\omega}^* + R\hat{i}_r + K_E\hat{\omega} \Rightarrow \\ sL\hat{i} &= R\hat{i}_r - R^*\hat{i} + K_E(\hat{\omega} - \hat{\omega}^*), \quad \hat{d}_\omega = K_E(\hat{\omega} - \hat{\omega}^*) \Rightarrow \\ (sL + R^*)\hat{i} &= R\hat{i}_r + \hat{d}_\omega \end{aligned}$$

Any difference in actual and estimated angular velocity is here denoted  $\hat{d}_\omega$  and viewed as a disturbance. This results in:

$$\hat{i} = \frac{R\hat{i}_r + \hat{d}_\omega}{sL + R^*} \quad (3.11)$$

When using (2.5),  $T = K_T i$ , which gives the transfer function  $\hat{i}_r \rightarrow \hat{T}$ :

$$\hat{T} = \frac{K_T R}{sL + R^*} \hat{i}_r \quad (3.12)$$

Inserting (3.11) into (3.9) and using (3.10):

$$s^2 J \hat{\theta} = \frac{K_T R \hat{i}_r + K_T \hat{d}_\omega}{sL + R^*} - \hat{T}_L \quad (3.13)$$

Inserting (3.1) into (3.13):

$$\begin{aligned} s^2 J \hat{\theta} &= \frac{K_T R}{sL + R^*} \hat{i}_r + \frac{K_T}{sL + R^*} \hat{d}_\omega - \hat{T}_L, \quad \hat{T}'_L = \frac{K_T}{sL + R^*} \hat{d}_\omega - \hat{T}_L \Rightarrow \\ s^2 J \hat{\theta} &= \frac{K_T R}{sL + R^*} \hat{i}_r + \hat{T}'_L \end{aligned}$$

Here, the disturbance has been incorporated into  $\hat{T}'_L$ . Inserting Equation (3.1) into the above equation with  $e = \theta_r - \theta$  gives:

$$s^2 J \hat{\theta} = \frac{K_T R}{sL + R^*} (K_p(\theta_r - \theta) + K_d(\dot{\theta}_r - \dot{\theta})) + \hat{T}'_L \quad (3.14)$$

From (3.14) we get the complementary sensitivity function:

$$\hat{\theta} = \frac{sK_T R K_d + K_T R K_p}{s^3 L J + s^2 R^* J + s K_T R K_d + K_T R K_p} \hat{\theta}_r + \frac{sL + R^*}{s^3 L J + s^2 R^* J + s K_T R K_d + K_T R K_p} \hat{T}'_L \quad (3.15)$$

With the relation  $\theta = \theta_r - e$ , we get the sensitivity function:

$$\hat{e} = \frac{s^3 L J + s^2 R^* J}{s^3 L J + s^2 R^* J + s K_T R K_d + K_T R K_p} \hat{\theta}_r - \frac{sL + R^*}{s^3 L J + s^2 R^* J + s K_T R K_d + K_T R K_p} \hat{T}'_L \quad (3.16)$$

### 3.1.3 Stability

Using Hurwitz's criterion [1] for the closed loop system, we can construct a matrix using the coefficients in the denominator shared between the sensitivity functions above:

$$H = \begin{bmatrix} R^* J & K_T R K_p \\ L J & K_T R K_d \end{bmatrix}$$

If  $H$  is positive definite, the system is stable. The leading principal minors of  $H$  are:

$$\det \begin{bmatrix} R^* J & K_T R K_p \\ L J & K_T R K_d \end{bmatrix} = (R^* J)(K_T R K_d) - (K_T R K_p)(L J)$$

If these are positive [5], the matrix is positive definite, giving the following criteria for the stability of the closed loop system:

$$R^* J > 0 \quad (3.17)$$

$$R^* K_d > L K_p \quad (3.18)$$

The first criterion, (3.17), is always satisfied, since the parameters are positive and real. The second criterion, (3.18), is dependent on both the controller and the given system parameters. The inequality (3.18) can be expressed in terms of (3.6) and (3.7):

$$\zeta > \frac{L\omega_0}{R^* 2} \quad (3.19)$$

$$\omega_0 < \frac{R^* 2\zeta}{L} \quad (3.20)$$

## 3.2 The effect of parameter variation

As a measure of the effect a varying parameter has on a system, one can use relative sensitivity. In the frequency-domain this is called the Bode sensitivity function. For a function,  $F(s)$ , the Bode sensitivity function with respect to a parameter,  $\alpha$ , is defined as [11]

$$S_\alpha^{F(s)} \triangleq \frac{\partial F(s)/F(s)}{\partial \alpha/\alpha} = \frac{\alpha_0}{F(s)_0} \left. \frac{\partial F(s)}{\partial \alpha} \right|_{NOP}, \quad (3.21)$$

using a nominal operating point (NOP) for all the parameters.

The Laplace transform of (2.9), (2.10) and (2.11) are given by (2.15) and (2.16).

### Open loop

We can find the steady-state gain of  $\hat{u} \rightarrow \hat{\theta}$  as:

$$\lim_{s \rightarrow 0} s \frac{\hat{\theta}}{\hat{u}}(s) = \frac{1}{K_E}$$

and the steady-state gain of  $\hat{T}_L \rightarrow \hat{\theta}$  as:

$$\lim_{s \rightarrow 0} s \frac{\hat{\theta}}{\hat{T}_L}(s) = -\frac{R}{K_T K_E}$$

It is apparent that the resistance,  $R$ , has no influence on the open loop gain for  $\hat{u} \rightarrow \hat{\theta}$  in steady-state (because of the integration from angular velocity to angular position), and it is multiplicative with the load for  $\hat{T}_L \rightarrow \hat{\theta}$  in steady-state.

The Fourier transform of  $G_3$  is:

$$\frac{\hat{\theta}}{\hat{u}}(j\omega) = \frac{-\omega K_T J R + j(\omega^2 K_T J L - K_T^2 K_E)}{\omega^3 J^2 R^2 + \omega(\omega^2 J L - K_T K_E)^2}$$

From this we find the amplitude:

$$\left| \frac{\hat{\theta}}{\hat{u}}(j\omega) \right| = \sqrt{\frac{(\omega K_T J R)^2 + (\omega^2 K_T J L - K_T^2 K_E)^2}{(\omega^3 J^2 R^2 + \omega(\omega^2 J L - K_T K_E)^2)^2}}$$

As can be seen, the amplitude, or gain, of the motor at a given frequency,  $\omega_0$ , is approximately reciprocal to the resistance in the armature.

$$\left| \frac{\hat{\theta}}{\hat{u}}(j\omega_0, R) \right| \sim \frac{1}{R} \quad (3.22)$$

The relative sensitivities of the open loop system with respect to the resistance,  $R$ , are given by:

$$S_R^G(s) = \begin{bmatrix} -\frac{sJR}{(s^2 JL + sJR + K_T K_E)} \\ -\frac{sJR}{(s^2 JL + sJR + K_T K_E)} \\ -\frac{sJR}{(s^2 JL + sJR + K_T K_E)} \end{bmatrix} \quad (3.23)$$

$$S_R^{G_d}(s) = \begin{bmatrix} -\frac{sJR}{(s^2 JL + sJR + K_T K_E)} \\ \frac{RK_T K_E}{(s^2 JL + sJR + K_T K_E)(sL + R)} \\ \frac{RK_T K_E}{(s^2 JL + sJR + K_T K_E)(sL + R)} \end{bmatrix} \quad (3.24)$$

As can be seen,  $S_R^{G_1} = S_R^{G_2} = S_R^{G_3} = S_R^{G_{d1}}$  and  $S_R^{G_{d2}} = S_R^{G_{d3}}$ .

The magnitudes of  $S_R^{G_3}$  and  $S_R^{G_{d3}}$  are shown in Figure 3.2 using nominal parameters for a Maxon RE 36.118800. A plot of  $|G_3(j\omega, R)|$  is also presented in Figure A.1. The effect of parameter variation is most notable in an interval of  $(10^2, 10^4)$  [rad/s]  $\approx (16, 1600)$  [Hz]. This is where the proportionality of (3.22) is most pronounced.

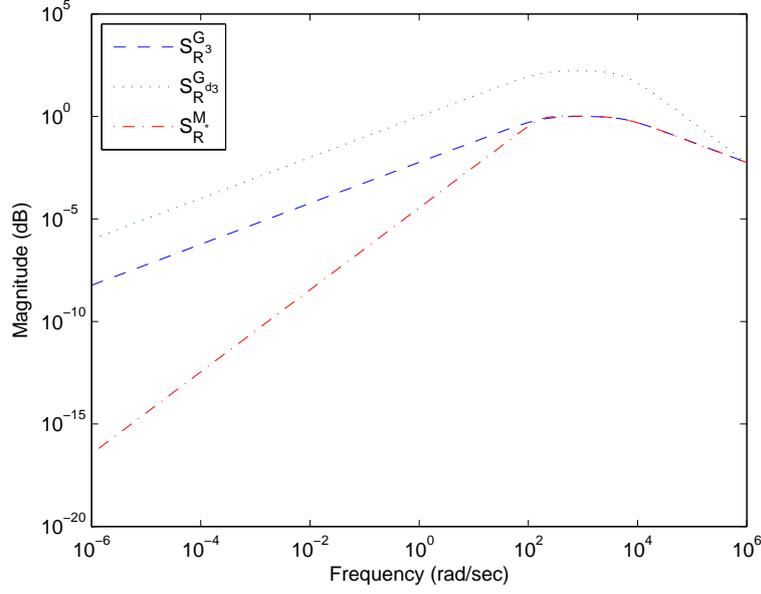


Figure 3.2: Open loop sensitivity to variation in  $R$  and closed loop sensitivity to variation in  $R^*$ .

### Closed loop example

Closing the loop with a serial-type compensation using some given controller,  $K(s)$ , the input to the motor is  $\hat{u} = K(s)\hat{e} = K(s)(\hat{\theta}_0 - \hat{\theta})$ . This gives us the following Laplace transform:

$$\hat{\theta} = \frac{K_T K(s)(\hat{\theta}_0 - \hat{\theta}) - (sL + R)\hat{T}_L}{s(s^2 J L + s J R + K_T K_E)}$$

Using the relation  $\hat{e} = \hat{\theta}_0 - \hat{\theta} \Rightarrow \hat{\theta} = \hat{\theta}_0 - \hat{e}$  we find an expression for  $\hat{e}$

$$\hat{e} = \frac{(sL + R)\hat{T}_L + s(s^2 J L + s J R + K_T K_E)\hat{\theta}_0}{K_T K(s) + s(s^2 J L + s J R + K_T K_E)}$$

and an expression for  $\hat{\theta}$

$$\hat{\theta} = \frac{K_T K(s)\hat{\theta}_0 - (sL + R)\hat{T}_L}{K_T K(s) + s(s^2 J L + s J R + K_T K_E)}.$$

If we put a step on each of the inputs  $\hat{\theta}_0 = \frac{a}{s}$  and  $\hat{T}_L = \frac{b}{s}$ , we get the following stationary values for  $\hat{e}$  and  $\hat{\theta}$ :

$$e_{ss} = \lim_{s \rightarrow 0} s \frac{1}{s} \left( \frac{(sL + R)b + s(s^2 J L + s J R + K_T K_E)a}{K_T K(s) + s(s^2 J L + s J R + K_T K_E)} \right) = \frac{Rb}{K_T K(s \rightarrow 0)}$$

$$\theta_{ss} = \lim_{s \rightarrow 0} s \frac{1}{s} \left( \frac{K_T K(s)a - (sL + R)b}{K_T K(s) + s(s^2 J L + s J R + K_T K_E)} \right) = \frac{K_T K(s \rightarrow 0)a - Rb}{K_T K(s \rightarrow 0)}$$

As can be seen, the error is proportional to the resistance in the armature. Depending on the behavior of the controller, the sensitivity (error) can go to zero and the complementary sensitivity (tracking) can go to  $a$ , as desired. For a proportional controller with very high gain, the error becomes very small, and for a proportional-integral controller,  $K_{PI}(s \rightarrow 0) = \infty$ , the error goes to zero in steady-state, as expected.

### Closed loop

From Equations (3.15) and (3.16) we get the complementary sensitivity function,

$$M(s) = \frac{sK_T R K_d + K_T R K_p}{s^3 L J + s^2 R^* J + sK_T R K_d + K_T R K_p} \quad (3.25)$$

sensitivity function,

$$N(s) = \frac{s^3 L J + s^2 R^* J}{s^3 L J + s^2 R^* J + sK_T R K_d + K_T R K_p} \quad (3.26)$$

and the sensitivities for the disturbances:

$$M_{d\omega}(s) = \frac{1}{s^3 L J + s^2 R^* J + sK_T R K_d + K_T R K_p} \quad (3.27)$$

$$M_{T_L}(s) = -\frac{sL + R^*}{s^3 L J + s^2 R^* J + sK_T R K_d + K_T R K_p} \quad (3.28)$$

$$N_{d\omega}(s) = -\frac{1}{s^3 L J + s^2 R^* J + sK_T R K_d + K_T R K_p} \quad (3.29)$$

$$N_{T_L}(s) = \frac{sL + R^*}{s^3 L J + s^2 R^* J + sK_T R K_d + K_T R K_p} \quad (3.30)$$

A step on each of the inputs  $\hat{\theta}_0 = \frac{a}{s}$ ,  $\hat{T}_L = \frac{b}{s}$  and  $\hat{d}_\omega = \frac{c}{s}$ , yields the following stationary values for  $\hat{\theta}$  and  $\hat{e}$ :

$$\theta_{ss} = \lim_{s \rightarrow 0} s \frac{1}{s} (aM(s) + bM_{T_L}(s) + cM_{d\omega}(s)) = a - b \frac{R^*}{K_T R K_p} + c \frac{1}{K_T R K_p}$$

$$e_{ss} = \lim_{s \rightarrow 0} s \frac{1}{s} (aN(s) + bN_{T_L}(s) + cN_{d\omega}(s)) = b \frac{R^*}{K_T R K_p} - c \frac{1}{K_T R K_p}$$

As in the previous example, the error approaches zero and the position approaches  $a$  with high proportional gain in the controller.

The relative sensitivity of (3.25) to change in  $R^*$  is given by

$$S_{R^*}^M = \frac{sR^* J}{s^3 L J + s^2 R^* J + sK_T R K_d + K_T R K_p}$$

and is plotted in Figure 3.2 as well. It can be shown for (3.26) that  $S_{R^*}^N = -S_{R^*}^M$ , and thus each gives the same response. As can be seen in the plot, the relative sensitivity is lower than in the open loop case, as would be expected.

If we look at the transfer function from the desired current to torque, (3.12), we can see that the effect of the varying resistance has a lot of influence on the torque output of the motor.

$$G_T(s) = \frac{K_T R}{sL + R^*} \quad (3.31)$$

A step input in the desired current with a torque setpoint  $i_0 = \frac{1}{K_T}T_0$ , yields the stationary value:

$$T_{ss} = \lim_{s \rightarrow 0} s \frac{1}{s} \frac{1}{K_T} T_0 (G_T(s)) = \frac{R}{R^*} T_0 \quad (3.32)$$

To match the torque setpoint perfectly, the estimated resistance (or the resistance assumed in the controller), must match the actual resistance,  $R = R^*$ .

The relative sensitivity of  $G_T(s)$  with respect to the estimated resistance,  $R$ , is very simple:

$$S_R^{G_T} = 1 \quad (3.33)$$

This means that varying the resistance assumed in the controller will influence the gain proportionally for all frequencies. A plot of  $|G_T(j\omega, R)|$  is presented in Figure A.2.

### 3.3 Discrete analysis

#### 3.3.1 Exact discretization

Given a continuous-time system of the form [5]

$$\begin{aligned} \dot{x} &= Ax(t) + Bu(t) \\ y &= Cx(t) + Du(t), \end{aligned} \quad (3.34)$$

we can find a discretization to put into the form

$$\begin{aligned} x_{k+1} &= A_\delta x_k + B_\delta u_k \\ y_k &= C_\delta x_k + D_\delta u_k. \end{aligned} \quad (3.35)$$

If we assume that the input to the system will be piecewise constant for the duration of the sampling time,  $T$ , the input can be described as a zero-order hold (ZOH) element:

$$u(t) = u(kT) = u_k \quad \text{for } kT \leq t < (k+1)T \quad (3.36)$$

Using this input, we can find an exact discretization, meaning the solution of (3.34) still equals the solution of (3.35).

For (3.34) we can find the solution as

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau \quad (3.37)$$

Using (3.37) with (3.36) we find

$$x_k = x(kT) = e^{AkT}x(0) + \int_0^{kT} e^{A(kT-\tau)}Bu(\tau)d\tau. \quad (3.38)$$

and

$$x_{k+1} = x((k+1)T) = e^{A(k+1)T}x(0) + \int_0^{(k+1)T} e^{A((k+1)T-\tau)}Bu(\tau)d\tau. \quad (3.39)$$

Rearranging (3.39) as

$$x_{k+1} = e^{AT} \left[ e^{AkT}x(0) + \int_0^{kT} e^{A(kT-\tau)}Bu(\tau)d\tau \right] + \int_{kT}^{(k+1)T} e^{A((kT+T-\tau)}Bu(\tau)d\tau$$

and substituting with (3.36), (3.38) and  $\alpha = kT + T - \tau$ , we get

$$x_{k+1} = e^{AT}x_k + \left( \int_0^T e^{A\alpha}d\alpha \right) Bu_k$$

The state, input, output and feedthrough matrices  $(A, B, C, D)$  of the discretized system will thus be given by:

$$A_\delta = e^{AT}, \quad B_\delta = \left( \int_0^T e^{A\tau}d\tau \right) B, \quad C_\delta = C, \quad D_\delta = D \quad (3.40)$$

If the state matrix of the continuous-time system is invertible, it can be shown that the discretized input matrix can be computed as

$$B_\delta = A^{-1}(e^{AT} - I)B = A^{-1}(A_\delta - I)B. \quad (3.41)$$

Otherwise it might also be computed as a power series:

$$B_\delta = \left( \int_0^T I + A\tau + A^2\frac{\tau^2}{2!} + \dots d\tau \right) B = \left( IT + A\frac{T^2}{2!} + A^2\frac{T^3}{3!} + \dots \right) B \quad (3.42)$$

### 3.3.2 Discretization of motor model

To do further work with the motor model in the context of discrete control, a discretization of (2.12) is needed. An exact discretization follows.

The state and input matrices are given by

$$A = \begin{bmatrix} -\frac{R}{L} & -\frac{K_E}{L} & 0 \\ \frac{K_T}{J} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \bar{B} = [B \quad B_d] = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{1}{J} \\ 0 & 0 \end{bmatrix},$$

where we have lumped both input matrices into one matrix,  $\bar{B}$ , for convenience. The eigenvalues of the system are given by:

$$\Delta(\lambda) = \det(A - \lambda I) = \lambda \frac{1}{JL} (JL\lambda^2 + JR\lambda + K_T K_E) = 0$$

$$\lambda_{1,2} = -\frac{JR \pm \sqrt{J^2 R^2 - 4JLK_T K_E}}{2JL}, \quad \lambda_3 = 0$$

Finding a set of accompanying eigenvectors,  $Aq_i = \lambda_i q_i$ , we can find the transition matrix for the system as a similarity transform:

$$e^{At} = Qe^{\hat{A}t}Q^{-1},$$

where  $\hat{A}$  is a diagonal matrix of distinct eigenvalues

$$\hat{A} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix},$$

and  $Q$  is a matrix composed of the eigenvectors

$$Q = [q_1 \quad q_2 \quad q_3] = \begin{bmatrix} \frac{J}{K_T} \lambda_1^2 & \frac{J}{K_T} \lambda_2^2 & 0 \\ \lambda_1 & \lambda_2 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

The inverse of the eigenvector matrix is given by

$$Q^{-1} = \frac{1}{\lambda_1 - \lambda_2} \begin{bmatrix} \frac{K_T}{J} \frac{1}{\lambda_1} & -\lambda_2 & 0 \\ -\frac{K_T}{J} \frac{1}{\lambda_2} & \lambda_1 & 0 \\ \frac{K_T}{J} \frac{\lambda_1 - \lambda_2}{\lambda_1 \lambda_2} & \frac{\lambda_2^2 - \lambda_1^2}{\lambda_1 \lambda_2} & \lambda_1 - \lambda_2 \end{bmatrix}.$$

We can now compute the transition matrix:

$$e^{At} = \frac{1}{\lambda_1 - \lambda_2} \begin{bmatrix} \lambda_1 e^{\lambda_1 t} - \lambda_2 e^{\lambda_2 t} & \frac{J}{K_T} \lambda_1 \lambda_2 (e^{\lambda_2 t} - e^{\lambda_1 t}) & 0 \\ \frac{K_T}{J} (e^{\lambda_1 t} - e^{\lambda_2 t}) & \lambda_1 e^{\lambda_2 t} - \lambda_2 e^{\lambda_1 t} & 0 \\ \frac{K_T}{J} \left( \frac{1}{\lambda_1} e^{\lambda_1 t} - \frac{1}{\lambda_2} e^{\lambda_2 t} + \frac{\lambda_1 - \lambda_2}{\lambda_1 \lambda_2} \right) & \frac{\lambda_1}{\lambda_2} e^{\lambda_2 t} - \frac{\lambda_2}{\lambda_1} e^{\lambda_1 t} + \frac{\lambda_2^2 - \lambda_1^2}{\lambda_1 \lambda_2} & \lambda_1 - \lambda_2 \end{bmatrix}$$

Inserting for the sampling time,  $T$ , and doing some algebraic simplification, yields the discrete state matrix.

$$A_\delta = e^{AT} = \Phi = \frac{1}{\lambda_1 - \lambda_2} \begin{bmatrix} \lambda_1 e^{\lambda_1 T} - \lambda_2 e^{\lambda_2 T} & \frac{K_E}{L} (e^{\lambda_2 T} - e^{\lambda_1 T}) & 0 \\ \frac{K_T}{J} (e^{\lambda_1 T} - e^{\lambda_2 T}) & \lambda_1 e^{\lambda_2 T} - \lambda_2 e^{\lambda_1 T} & 0 \\ \frac{K_T}{J} \left( \frac{1}{\lambda_1} (e^{\lambda_1 T} - 1) - \frac{1}{\lambda_2} (e^{\lambda_2 T} - 1) \right) & \frac{\lambda_1}{\lambda_2} (e^{\lambda_2 T} - 1) - \frac{\lambda_2}{\lambda_1} (e^{\lambda_1 T} - 1) & \lambda_1 - \lambda_2 \end{bmatrix} \quad (3.43)$$

Since  $A$  for this system is not invertible,  $B_\delta$  must be found by using a power series or the integral below. The latter was done.

$$B_\delta = \left( \int_0^T e^{A\tau} d\tau \right) \bar{B}$$

$$B_\delta = \Delta = \frac{1}{\lambda_1 - \lambda_2} \left[ \begin{array}{c} \frac{1}{L} (e^{\lambda_1 T} - e^{\lambda_2 T}) \\ \frac{K_T}{JL} \left( \frac{1}{\lambda_1} (e^{\lambda_1 T} - 1) - \frac{1}{\lambda_2} (e^{\lambda_2 T} - 1) \right) \quad \dots \\ \frac{K_T}{JL} \left( \frac{1}{\lambda_1^2} (e^{\lambda_1 T} - 1) - \frac{1}{\lambda_2^2} (e^{\lambda_2 T} - 1) + \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) T \right) \\ \frac{K_E}{JL} \left( \frac{1}{\lambda_1} (e^{\lambda_1 T} - 1) - \frac{1}{\lambda_2} (e^{\lambda_2 T} - 1) \right) \\ \frac{-1}{J} \left( \frac{\lambda_1}{\lambda_2} (e^{\lambda_2 T} - 1) - \frac{\lambda_2}{\lambda_1} (e^{\lambda_1 T} - 1) \right) \\ \frac{-1}{J} \left( \frac{\lambda_1}{\lambda_2^2} (e^{\lambda_2 T} - 1) - \frac{\lambda_2}{\lambda_1^2} (e^{\lambda_1 T} - 1) + \left( \frac{\lambda_2}{\lambda_1} - \frac{\lambda_1}{\lambda_2} \right) T \right) \end{array} \right] \quad (3.44)$$

It is worth noting, with implementation in mind, that some of the elements of  $\Phi$  and  $\Delta$  are very similar:

$$\delta_{11} = \frac{J}{K_T L} \phi_{21}, \quad \delta_{21} = \frac{1}{L} \phi_{31}, \quad \delta_{12} = \frac{K_E}{K_T L} \phi_{31} = \frac{K_E}{K_T} \delta_{21}, \quad \delta_{22} = \frac{-1}{J} \phi_{32}$$

### 3.3.3 Current feedback

In this section it is shown why simple current feedback does not work well on this system. This is also verified experimentally, as presented in Section 6.3.

Since the voltage input can be seen as having a zero-order hold (ZOH) element, we will use the exact discretization of the motor model to find the exact discrete stability behavior.

The angular position state is an integration of the angular velocity state, and discretization should move the continuous pole at  $s = 0$  to a pole at  $z = 1$ . It will not influence the stability. If we then consider the current and angular velocity states, and the controlled voltage input, we can find the stability margin for a proportional (P) regulator using current feedback.

Inserting the expressions for the eigenvalues in the requisite expressions from (3.43) and (3.44) gives us an explicit solution in terms of the motor's parameters.

$$\begin{aligned} \frac{1}{\lambda_1 - \lambda_2} &= \frac{JL}{\sqrt{J^2 R^2 - 4JLK_T K_E}} \\ \frac{1}{\lambda_1 - \lambda_2} \frac{J}{K_T} \lambda_1 \lambda_2 &= \frac{JK_E}{\sqrt{J^2 R^2 - 4JLK_T K_E}} \\ \frac{1}{\lambda_1 - \lambda_2} \frac{K_T}{J} &= \frac{LK_T}{\sqrt{J^2 R^2 - 4JLK_T K_E}} \\ \frac{\lambda_1}{\lambda_1 - \lambda_2} &= \frac{-JR + \sqrt{J^2 R^2 - 4JLK_T K_E}}{2\sqrt{J^2 R^2 - 4JLK_T K_E}} \end{aligned}$$

The discrete state matrix will now be

$$\Phi = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix},$$

where

$$\begin{aligned}\phi_{11} &= \frac{1}{2} \frac{(-JR + \sqrt{J^2R^2 - 4JLK_TK_E})e^{\lambda_1T} + (JR + \sqrt{J^2R^2 - 4JLK_TK_E})e^{\lambda_2T}}{\sqrt{J^2R^2 - 4JLK_TK_E}} \\ \phi_{12} &= \frac{LK_T(e^{\lambda_1T} - e^{\lambda_2T})}{\sqrt{J^2R^2 - 4JLK_TK_E}} \\ \phi_{21} &= \frac{JK_E(e^{\lambda_2T} - e^{\lambda_1T})}{\sqrt{J^2R^2 - 4JLK_TK_E}} \\ \phi_{22} &= \frac{1}{2} \frac{(-JR + \sqrt{J^2R^2 - 4JLK_TK_E})e^{\lambda_2T} + (JR + \sqrt{J^2R^2 - 4JLK_TK_E})e^{\lambda_1T}}{\sqrt{J^2R^2 - 4JLK_TK_E}}.\end{aligned}$$

The input matrix will be

$$\Delta = \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} = \left[ \frac{J(e^{\lambda_1T} - e^{\lambda_2T})}{\sqrt{J^2R^2 - 4JLK_TK_E}} \frac{1}{K_E} \left( 1 - \frac{(JR + \sqrt{J^2R^2 - 4JLK_TK_E})e^{\lambda_1T} - (JR - \sqrt{J^2R^2 - 4JLK_TK_E})e^{\lambda_2T}}{2\sqrt{J^2R^2 - 4JLK_TK_E}} \right) \right].$$

A proportional current controller is given by

$$u_k = K_i(i_r - i_k).$$

The state difference equations should be

$$x_{k+1} = \Phi x_k + \Delta u_k.$$

Adding the current controller we get the expression

$$\begin{aligned}x_{k+1} &= \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} K_i(i_r - i_k) = \\ &= \begin{bmatrix} \phi_{11} - \delta_1 K_i & \phi_{12} \\ \phi_{21} - \delta_2 K_i & \phi_{22} \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} \delta_1 K_i \\ \delta_2 K_i \end{bmatrix} i_r,\end{aligned}$$

or more briefly expressed as

$$x_{k+1} = \Phi' x_k + \Delta i_r.$$

Solving for the eigenvalues  $\det(\Phi' - \lambda I) = 0$  yields

$$\lambda_{1,2} = -\frac{K_i - \phi_{11} - \phi_{22}}{2} \pm \frac{(K_i - \phi_{11} - \phi_{22})^2 - 4(\phi_{11}\phi_{22} - \phi_{12}\phi_{21} + \phi_{12}K_i\delta_2 - \phi_{22}K_i\delta_1)}{2}.$$

Now solving  $\lambda_{1,2} = e^{i\theta}$ ,  $\theta \in (0, 2\pi)$ , with respect to  $K_i$ , yields one solution ( $\lambda_2 = -1$ ):

$$K'_i = \frac{(e^{\alpha_1} + e^{\alpha_2} + e^{\alpha_3} + e^{\alpha_4})(-JR^2 + 4LK_TK_E)}{(e^{\alpha_1} + e^{\alpha_2})(\sqrt{J^2R^2 - 4JLK_TK_E})},$$

where

$$\begin{aligned}\alpha_1 &= -\frac{JR + 2\sqrt{J^2R^2 - 4JLK_TK_E}}{2JL}T \\ \alpha_2 &= -\frac{R}{2L}T \\ \alpha_3 &= -\frac{2JR + \sqrt{J^2R^2 - 4JLK_TK_E}}{2JL}T \\ \alpha_4 &= -\frac{\sqrt{J^2R^2 - 4JLK_TK_E}}{2JL}T.\end{aligned}$$

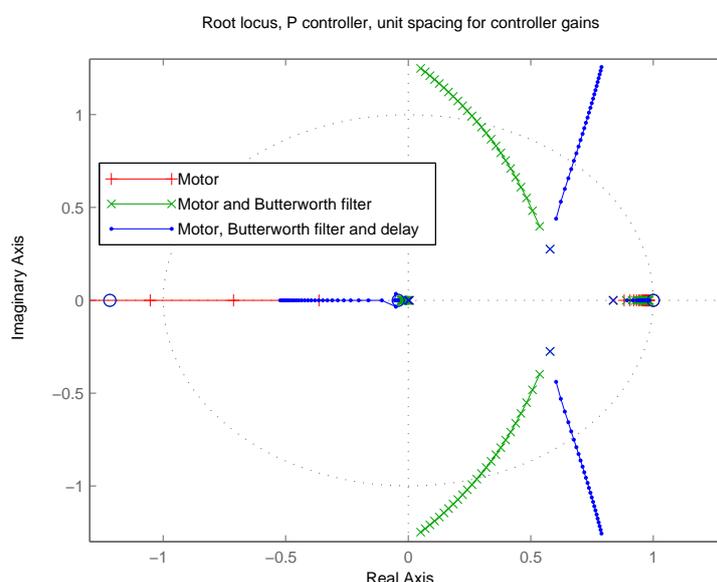


Figure 3.3: Root locus for discrete current feedback using a P controller.

This yields the stability margin,  $K_i < K'_i$ . Numerically, with data from Table 2.1,  $K'_i \approx 2.8$ . This value will, of course, change if the motor's parameters change.

As this method of performing stability analysis is very cumbersome, MATLAB was used to generate a root locus plot of the poles when adding a Butterworth filter and output delay for the voltage.

Using the discretized transfer function  $u \rightarrow i$ ,  $G_1$ , from (2.15) together with the discretized transfer function for the Butterworth filter, (2.18), Figure 3.3 is produced, which shows the root locus for the motor with a discrete current controller, as well as the system with a Butterworth filter connected in series with the current output, and with a Butterworth filter together with a unit delay,  $z^{-1}$ , on the voltage input. The delay is present due to an implementation aspect on the SCU. The Butterworth filter makes the stability margin wider, and the delay narrows it.

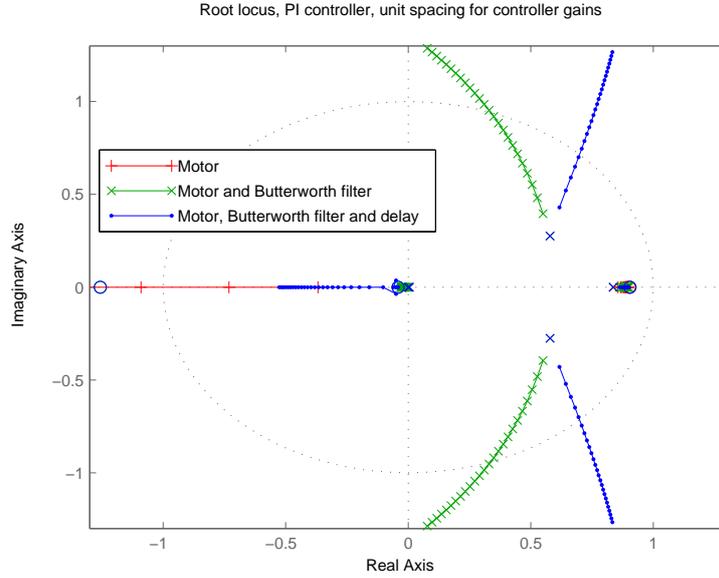


Figure 3.4: Root locus for discrete current feedback using a PI controller.

The last plot is representative of the real system, and yields a maximum gain of  $K'_i \approx 5$ , which was deemed to be too low, i.e., not enough to give a sufficiently small error.

A proportional integral (PI) regulator could be used to let the current error go to zero asymptotically. A PI regulator is given by:

$$G_{PI}(s) = K_P \frac{1 + T_I s}{T_I s} = K_P \frac{1 + 1/K_I s}{1/K_I s}$$

Exact discretization yields the the  $z$ -transform as:

$$G_{PI}(z) = \frac{K_P z + T K_I - K_P}{z - 1} \quad (3.45)$$

Adding this to the system yields the root locus plot in Figure 3.4, here using an integral gain of value  $K_I = 100$ . The location of the poles does not change much for integral gains below this value, but will start to move out of the unit circle if the integral gain is set higher. So, up to about  $K_I = 100$ , the stability margin for the proportional term is about the same.

A PI regulator is not used, however, because it does not provide good performance at these gains, as shown in Section 6.3, and also in order to avoid problems that could occur with integral windup. The system will sometimes saturate at the current limit ( $i_{max} = 0.7$  [A]). Saturation typically causes integral windup.

The MATLAB script used to produce Figures 3.3 and 3.4 is described in Appendix B.1.1.

### 3.4 Remarks

Generally, numerical integration using exact discretization will be stable for any sampling time. For the system under consideration, any other type of fixed-step explicit numerical integration scheme was found to be mostly unstable, since it is not practical or possible to sample at a rate that matches the time constant of the electrical dynamics,  $\tau = L/R$ .

Euler integration and Runge-Kutta (RK) integration up to the 5th order were tried for solving the motor's differential equations. Varying the resistance in the motor model, the RK5 scheme was stable at low resistance, since lower resistances produce larger time constants, which match the sampling time used.

The electrical dynamics might be ignored in some cases, avoiding these stability issues altogether. As a rule of thumb, the sampling time should then be longer than five time constants.

# Chapter 4

## Estimation

### 4.1 Introduction

In this chapter we will present two estimation schemes to be able to learn the values of varying parameters in the motor. The first scheme is the gradient method, which is the one used now. The other is the extended Kalman filter. We will also present a method for performing nonlinear regression.

The gradient method is used to estimate the motor's terminal resistance. It uses a noisy first difference estimate of the velocity, and a current measurement that often has a bias component which will cause an asymptotic error in the estimate. For the real system, this bias component is slowly varying, and a static correction is used, where the mean value of the current measurement is found before any voltage is applied, and that value is henceforth subtracted from the measurement when voltage is applied. The bias component of the instrumentation used in the experiments was constantly changing and had a hysteresis effect, so static correction was not applicable.

The extended Kalman filter was developed to in order to estimate any of the motor's parameters. It should also provide better velocity estimates, and be able to track unknown measurement bias. It was also necessary to track an unknown exogenous input, the torque load disturbance, since the filter was developed using all the state-space equations for the motor. In this manner, it was hoped that more accurate estimates could be obtained.

An observer for the motor's winding temperature was also developed using a steady-state Kalman filter. This was done to be able to compensate for the temperature dependency in the motor's winding.

### 4.2 Gradient method

#### 4.2.1 Resistance estimation

Since the current is available for measurement, and an estimate of the angular velocity can be produced from the encoder signal, the equation describing the electrical dynamics

can be used to estimate the resistance:

$$L \frac{di}{dt} = -Ri - K_E \omega + u \quad (4.1)$$

The electrical time constant for the equation,  $\tau = L/R$ , is very small for the motors used in this system. Using the data from Table 2.1, the value of the time constant is  $\tau \approx 0.2 \times 10^{-3}$  [s]. Comparing it to the sampling period used for the current measurement on the SCU,  $T = 1 \times 10^{-3}$  [s], it can be neglected ( $\tau \ll T$  or  $T \approx 5\tau$ ), giving the equation:

$$u_R = Ri = u - K_E \omega \quad (4.2)$$

Angular velocity,  $\omega$ , is available as the first difference of the measurement from the encoders, and can in turn be used to estimate the electromotive force,  $e = K_E \omega$ , in (4.2), giving the simple relation:

$$i = \frac{1}{R} u_R = G u_R \quad (4.3)$$

The control signal,  $u$ , is, of course, also known. This is the model used: a scalar equation with one unknown parameter,  $G$ , the conductance [S].

To estimate the varying resistance, the gradient method [10], or least mean squares (LMS) algorithm [23], is used in the existing system. The method is summarized in Table 4.1 [10].

Table 4.1: Gradient method

|                  |                                    |
|------------------|------------------------------------|
| Parametric model | $y = \theta^* u$                   |
| Estimation model | $\hat{y} = \theta u$               |
| Estimation error | $\epsilon = y - \hat{y}$           |
| Adaptive law     | $\dot{\theta} = \gamma \epsilon u$ |

In our case, the adaptive law is:

$$\dot{G} = \gamma (i - G u_R) u_R \quad (4.4)$$

A discretization of (4.4) can be found using Euler integration [7] (this gives the LMS algorithm as given in [23].) Given a differential equation of the following form:

$$\dot{x} = f(x(t), v(t), t)$$

we can find the solution numerically from the following equation, where  $t = hk$ , and  $h$  is the time step:

$$x(k+1) = x(k) + hf(x(k), v(k), k)$$

For Equation (4.4), this gives us:

$$G(k+1) = G(k) + h\gamma [i(k) - G(k)u_R(k)] u_R(k) \quad (4.5)$$

A block diagram of this discrete adaptive law is presented in Figure D.1.

### 4.3 Derivation, stability and convergence

In [10], it is shown how to derive the method summarized in Table 4.1. We start out by using the difference between a measurement and the corresponding model response. Let  $\theta^*$  be the true value of a given parameter, and  $\theta$  be the estimate of it. The model response in our case is given by

$$\hat{y}(t) = \theta(t)u(t),$$

where the measurement from the real system is denoted as  $y(t)$ , and  $u(t)$  is the known forcing signal (the circumflex on  $\hat{y}(t)$  here means the estimate.) Looking at the difference between the two values gives

$$\epsilon = y - \hat{y} = y - \theta u. \quad (4.6)$$

Defining the parameter error as  $\tilde{\theta} \triangleq \theta - \theta^*$ , we can express Equation (4.6) as

$$\epsilon = \theta^* u - \theta u = -\tilde{\theta} u.$$

When the parameter error goes to zero, the estimate has converged to the true value of the parameter, given the chosen model. We thus want to minimize (4.6) with respect to  $\theta$ , for example by using some iterative technique, such as the gradient method. The gradient method above is derived using an instantaneous cost criterion

$$J(\theta) = \frac{\epsilon^2}{2} = \frac{(y - \theta u)^2}{2}.$$

This is a convex function for all  $t$  and therefore has a global minimum because it satisfies  $\nabla J(\theta) = -(y - \theta^* u)u = 0$  at the minimum, and has a positive definite Hessian  $\nabla^2 J(\theta) = u^2 > 0$  for all  $\theta$ , if  $u \neq 0$ .

The gradient method for this case is given by

$$\dot{\theta} = -\gamma \nabla J(\theta) = \gamma(y - \theta u)u = \gamma \epsilon u, \quad \theta(0) = \theta_0, \quad (4.7)$$

where  $\gamma > 0$  is a scaling constant that influences the convergence of the method. The choice of  $\gamma$  is usually a trade-off between fast convergence and noise suppression. (For a discrete implementation one would also like to reduce any oscillations occurring due to overshooting the minimum value when the step size is large.) Comparing Equation (4.7) with the summary in Table 4.1, this is the adaptive law.

To show the stability of the adaptive law, we can rewrite it using the parameter error,  $\dot{\tilde{\theta}} = \dot{\theta} - \dot{\theta}^* = \gamma \epsilon u - \dot{\theta}^*$ , where  $\dot{\theta}^* = 0$  (assuming it is a constant), giving us

$$\dot{\tilde{\theta}} = \gamma \epsilon u = -\gamma \tilde{\theta} u^2, \quad \tilde{\theta}(0) = \theta_0 - \theta^*. \quad (4.8)$$

We can analyze this function using a Lyapunov function in quadratic form

$$V(\tilde{\theta}) = \frac{\tilde{\theta}^2}{2\gamma}.$$

The total derivative of this is

$$\dot{V}(\tilde{\theta}) = \frac{\tilde{\theta}\dot{\tilde{\theta}}}{\gamma} = -\tilde{\theta}^2 u^2 = -\epsilon^2 \leq 0, \quad (4.9)$$

which tells us that the equilibrium  $\tilde{\theta}_e = 0$  is uniformly stable. The properties of the forcing signal  $u$  determine any further stability characteristics and convergence of  $\tilde{\theta}$  to zero.

For the parameter estimate  $\theta$  to converge to  $\theta^*$  the signal  $u$  needs to be persistently exciting, which means that it must satisfy

$$\int_t^{t+T_0} u^2(\tau) d\tau \geq \alpha_0 T_0, \quad \forall t \geq 0$$

where  $\alpha_0, T_0 > 0$ . The closed form solution of (4.8) is given by

$$\tilde{\theta} = e^{-\gamma \int_0^t u^2(\tau) d\tau} \tilde{\theta}(0),$$

so it can be seen that any signal  $u$  satisfying the persistent excitation criterion will lead to an exponential convergence in the solution, assuming  $\theta^*$  is constant.

#### 4.4 The effect of measurement bias

The asymptotic behavior, when  $t \rightarrow \infty$ , of the gradient method, (4.7), gives us the final value of the estimated parameter as

$$\dot{\theta} = \gamma(y - \theta u)u = 0 \quad \Rightarrow \quad \theta = \frac{y}{u},$$

as should be expected. If we have a bias in the measurement,  $y = y_{actual} + y_{bias}$ , this will affect the final value of the estimate. The asymptotic value of (4.4) in the presence of the current bias,  $i_b$ , will be:

$$G_\infty = \frac{i + i_b}{u_R} \quad \Rightarrow \quad R_\infty = \frac{u_R}{i + i_b} = \frac{R^*}{1 + i_b/i}$$

To get an impression of the effect the bias will have, we can compute  $e_R = R^* - \frac{R^*}{1+i_b/i}$  for various values of the current,  $i$ . It is apparent that the estimation error will decrease with increasing current, as can be seen in Figure 4.1, where we have used  $R^* = 10$  [ $\Omega$ ],  $i_b = 1$  [mA]. To get accurate estimates at low currents, it is therefore desirable to remove any bias in the current measurements.

#### 4.5 Kalman filters

In this section, we will briefly present the discrete Kalman filter, the extended Kalman filter (EKF), and how to use the EKF to estimate parameters, unknown exogenous inputs, and unknown random bias. When an extended Kalman filter is used in this manner, augmenting it with additional states to be estimated, it allows us to estimate both states and parameters, and is sometimes called the augmented state Kalman filter (ASKF), [12], or joint extended Kalman filter (joint-EKF), [22].

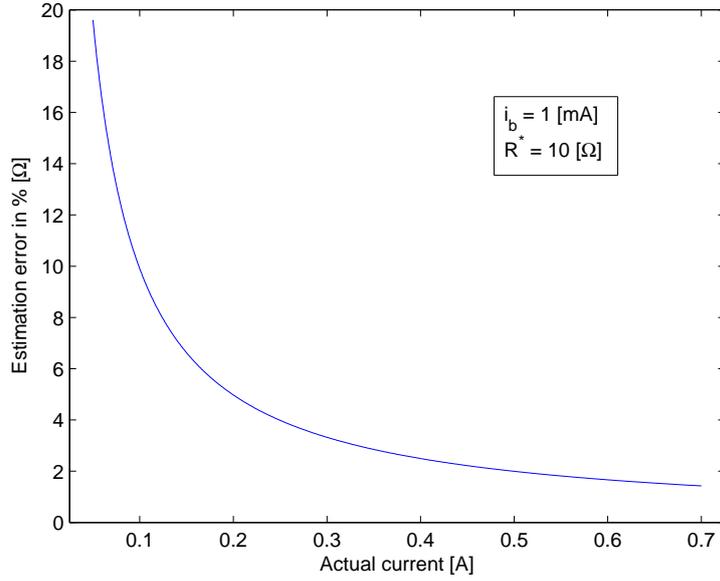


Figure 4.1: Asymptotic estimation error in the presence of current bias in the measurement.

#### 4.5.1 Discrete Kalman filter

A set of difference equations can be described as

$$x_{k+1} = F_k x_k + G_k u_k + w_k \quad (4.10)$$

$$y_k = H_k x_k + v_k, \quad (4.11)$$

where  $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^m$ ,  $u \in \mathbb{R}^l$ .  $w_k$  and  $v_k$  are stationary Gaussian white noise sequences. The covariance matrices for  $w_k$  and  $v_k$  are given by:

$$E[w_k w_i^\top] = Q_k \delta_{ki}$$

$$E[v_k v_i^\top] = R_k \delta_{ki}$$

$$E[w_k v_i^\top] = 0, \quad \forall k, i$$

Here  $\delta_{ki}$  denotes the Kronecker delta function, defined as:

$$\delta_{ki} = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$$

$Q_k$  and  $R_k$  are the process noise covariance and measurement noise covariance, respectively. The discrete Kalman filter for this system is summarized below [21].

## Filtering

Entering the sequence with the prior estimate  $\hat{x}_0^-$  and its error covariance  $P_0^-$ , the Kalman gain is computed as

$$K_k = P_k^- H_k^\top \left[ H_k P_k^- H_k^\top + R_k \right]^{-1}, \quad (4.12)$$

the estimate is updated with the measurement  $y_k$  as

$$\hat{x}_k = \hat{x}_k^- + K_k \left[ y_k - H_k \hat{x}_k^- \right], \quad (4.13)$$

and the error covariance for the updated estimate is computed as

$$P_k = [I - K_k H_k] P_k^-. \quad (4.14)$$

## Prediction

The next estimate is computed from the state equations

$$\hat{x}_{k+1}^- = F_k \hat{x}_k + G_k u_k, \quad (4.15)$$

and the error covariance is projected ahead as

$$P_{k+1}^- = F_k P_k F_k^\top + Q_k. \quad (4.16)$$

If the system is linear time invariant (LTI), and the noise has stationary statistics, the steady-state Kalman filter can be used. This means that  $P_{k+1}^- = P_k^-$  for large  $k$ . If we insert (4.12) into (4.14), and then the result into (4.16), we get the discrete algebraic Riccati equation (dropping the  $k$  subscripts since the matrices are constant):

$$F P^- F^\top - P^- - F P^- H^\top \left[ H P^- H^\top + R \right]^{-1} H P^- F^\top + Q = 0$$

Solving the Riccati equation with respect to  $P^-$  will give us a stationary Kalman gain when inserted into (4.12). Inserting (4.13) into (4.15), we can now update the state estimates with one equation:

$$\hat{x}_{k+1} = F(I - KH)\hat{x}_k + Gu_k + FK y_k \quad (4.17)$$

### 4.5.2 Extended Kalman filter

A set of nonlinear difference equations can be described as

$$x_{k+1} = f_k(x_k, u_k) + w_k \quad (4.18)$$

$$y_k = g_k(x_k) + v_k, \quad (4.19)$$

where we have the same conditions for  $x$ ,  $y$ ,  $u$ ,  $w$ , and  $v$  as for the discrete Kalman filter.

Estimating the states can now be done by linearizing about the filter's estimated trajectory, yielding the extended Kalman filter (EKF). The EKF for this system is summarized below [9].

## Filtering

Entering the sequence with the prior estimate  $\hat{\chi}_0^-$  and its error covariance  $P_0^-$ , the Kalman gain is computed as

$$K_k = P_k^- \bar{H}_k^\top \left[ \bar{H}_k P_k^- \bar{H}_k^\top + R_k \right]^{-1}, \quad (4.20)$$

the estimate is updated with the measurement  $y_k$  as

$$\hat{x}_k = \hat{x}_k^- + K_k \left[ y_k - g_k(\hat{x}_k^-) \right], \quad (4.21)$$

and the error covariance for the updated estimate is computed as

$$P_k = \left[ I - K_k \bar{H}_k \right] P_k^-. \quad (4.22)$$

## Prediction

The next estimate is computed from the state equations

$$\hat{x}_{k+1}^- = f_k(\hat{x}_k, u_k), \quad (4.23)$$

and the error covariance is projected ahead as

$$P_{k+1}^- = \bar{F}_k P_k \bar{F}_k^\top + Q_k. \quad (4.24)$$

The linearization about the estimated trajectory is given by the Jacobian matrices for the state and measurement equations:

$$\bar{F}_k = \frac{\partial f_k}{\partial x_k^\top}(\hat{x}_k, u_k), \quad \bar{H}_k = \frac{\partial g_k}{\partial x_k^\top}(\hat{x}_k^-) \quad (4.25)$$

### 4.5.3 Estimation of unknown parameters, exogenous inputs, and random bias

In this section we will here explain how to augment a state-space description in order to estimate unknown parameters, unknown exogenous inputs, and unknown random bias, using the Wiener process model. This is primarily based on [12]. The parameter estimation makes this a nonlinear problem, to which the EKF should be amenable [9, 4].

Given the following discrete equations

$$x_{k+1} = A(\theta_k)x_k + B_u(\theta_k)u_k + B_d(\theta_k)d_k + B_b(\theta_k)b_k + w_k^x \quad (4.26)$$

$$d_{k+1} = d_k + w_k^d \quad (4.27)$$

$$b_{k+1} = b_k + w_k^b \quad (4.28)$$

$$\theta_{k+1} = \theta_k + w_k^\theta \quad (4.29)$$

$$y_k = C(\theta_k)x_k + C_b(\theta_k)b_k + v_k \quad (4.30)$$

where  $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^m$ , and  $u \in \mathbb{R}^l$ .  $d \in \mathbb{R}^r$  is the vector of unknown exogenous disturbances,  $b \in \mathbb{R}^q$  is the unknown bias vector, and  $\theta \in \mathbb{R}^p$  is the vector of unknown parameters.  $d$ ,  $b$ , and  $\theta$  are all described as Wiener processes.  $w_k^x$ ,  $w_k^d$ ,  $w_k^b$ ,  $w_k^\theta$ , and  $v_k$  are Gaussian white noise sequences.

Defining an augmented state vector and noise sequence vector as

$$\chi_k = \begin{bmatrix} x_k \\ d_k \\ b_k \\ \theta_k \end{bmatrix}, \quad w_k = \begin{bmatrix} w_k^x \\ w_k^d \\ w_k^b \\ w_k^\theta \end{bmatrix},$$

we can further define an augmented state, input and output matrix as

$$F(\theta_k) = \begin{bmatrix} A(\theta_k) & B_d(\theta_k) & B_b(\theta_k) & 0 \\ 0 & I_r & 0 & 0 \\ 0 & 0 & I_q & 0 \\ 0 & 0 & 0 & I_p \end{bmatrix}$$

$$G(\theta_k) = \begin{bmatrix} B_u(\theta_k) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$H(\theta_k) = [ C(\theta_k) \quad 0 \quad C_b(\theta_k) \quad 0 ].$$

The process noise covariance matrix and the measurement noise covariance matrix are given by:

$$\begin{aligned} \mathbb{E}[w_k w_i^\top] &= Q_k \delta_{ki} \\ \mathbb{E}[v_k v_i^\top] &= R_k \delta_{ki} \\ \mathbb{E}[w_k v_i^\top] &= 0, \quad \forall k, i \end{aligned}$$

Because of the unknown parameters, this is a set of nonlinear equations:

$$f(\chi_k, u_k) = F(\theta_k)\chi_k + G(\theta_k)u_k = \begin{bmatrix} A(\theta_k)x_k + B_u(\theta_k)u_k + B_d(\theta_k)d_k + B_b(\theta_k)b_k \\ d_k \\ b_k \\ \theta_k \end{bmatrix} \quad (4.31)$$

$$g(\chi_k) = H(\theta_k)\chi_k = C(\theta_k)x_k + C_b(\theta_k)b_k. \quad (4.32)$$

Linearizing the state and measurement equations about the estimated trajectory, we find the linearized state matrix as

$$\bar{F}_k = \frac{\partial f(\chi_k, u_k)}{\partial \chi_k^\top} = \begin{bmatrix} F(\theta_k) & \frac{\partial}{\partial \theta_k^\top} [F(\theta_k)\chi_k + G(\theta_k)u_k] \\ 0 & I_{r+q+p} \end{bmatrix}_{\chi_k = \hat{\chi}_k, \theta_k = \hat{\theta}_k}, \quad (4.33)$$

and the linearized output matrix as

$$\bar{H}_k = \frac{\partial g(\chi_k^-)}{\partial \chi_k^\top} = \left[ H(\theta_k) \quad \frac{\partial}{\partial \theta_k^\top} [H(\theta_k)\chi_k] \right]_{\chi_k = \hat{\chi}_k^-, \theta_k = \hat{\theta}_k^-}. \quad (4.34)$$

Given that the complete, linearized system is observable, the parameters, exogenous inputs, and random bias can now be estimated using the EKF.

If the model of the system is deterministic and can be assumed to be accurate, the process noise covariance for the states in  $x$  are typically set to zero.

$$\mathbb{E}[w_k^x w_i^{x\top}] = Q_k^x = 0, \quad \forall k, i$$

The process noise variances for the unknown sizes in  $d$ ,  $b$  and  $\theta$ ,

$$\begin{aligned} \mathbb{E}[w_k^d w_i^{d\top}] &= Q_k^d \delta_{ki} \\ \mathbb{E}[w_k^b w_i^{b\top}] &= Q_k^b \delta_{ki} \\ \mathbb{E}[w_k^\theta w_i^{\theta\top}] &= Q_k^\theta \delta_{ki}, \end{aligned}$$

should roughly correspond to the possible range of variation. If it is known that, for example,  $\theta_i$  is likely to change by an amount of  $\Delta\theta_i$  over the interval of interest  $\Delta t$ , we can require

$$i^{\text{th}} \text{ diagonal element of } Q_k^\theta = \frac{\Delta\theta_i^2}{\Delta t}. \quad (4.35)$$

This should provide, at minimum, a starting point for tuning the process noise covariance matrix elements [9].

#### 4.5.4 Kalman filter for thermal model

Since the housing temperature is made available for measurement in the experimental instrumentation, the winding temperature could be observed using a state estimator and the thermal model from Section 2.2. The observer used was the stationary Kalman filter (4.17). The filter was found in a straightforward manner using MATLAB's `kalman` function. The MATLAB script is described in Appendix B.1.2.

#### 4.5.5 Motor model with Butterworth filter

To avoid anti-aliasing, the current measurement is filtered by a second-order low-pass Butterworth filter. The Butterworth filter alters the system dynamics to such a degree that a Kalman filter will not be stable when using only the motor model. Adding the Butterworth filter from (2.19) to the system yields the following state matrix

$$A = \begin{bmatrix} -\frac{R}{L} & 0 & 0 & -\frac{K_E}{L} & 0 \\ \omega_c & -\omega_c\sqrt{2} & -\omega_c & 0 & 0 \\ 0 & \omega_c & 0 & 0 & 0 \\ \frac{K_T}{J} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (4.36)$$

where we define the new state vector to be:

$$x_k = \begin{bmatrix} i_k \\ i'_{fk} \\ i_{fk} \\ \omega_k \\ \theta_k \end{bmatrix}$$

Descriptions of the denotations are given in Table 4.2.

Table 4.2: Descriptions of denotations.

| Denotation | Description  |
|------------|--|
| $i$        | Armature current.  |
| $i'_f$     | Internal state of the Butterworth filter.                        |
| $i_f$      | Output of the Butterworth filter, the filtered armature current. |
| $\omega$   | Angular velocity of the motor's rotor.                           |
| $\theta$   | Angular position of the motor's rotor.                           |

The eigenvalues of this system are given by:

$$\det(A - \lambda I) = \lambda \frac{1}{JL} (JL\lambda^2 + JR\lambda + K_E K_T) (\lambda^2 + \omega_c \sqrt{2}\lambda + \omega_c^2)$$

$$\lambda_{1,2} = -\frac{JR \pm \sqrt{J^2 R^2 - 4JL K_T K_E}}{2JL}, \quad \lambda_3 = 0, \quad \lambda_{4,5} = -\omega_c \left( \frac{\sqrt{2}}{2} \pm i \frac{\sqrt{2}}{2} \right)$$

The eigenvalues of the motor and the Butterworth filter are not changed by a serial connection.

For simplicity, we can define a damping factor,  $\mu$ , and frequency,  $\nu$ , from the eigenvalues corresponding to the Butterworth filter as:

$$\mu \triangleq -\omega_c \frac{\sqrt{2}}{2}, \quad \nu \triangleq \omega_c \frac{\sqrt{2}}{2}, \quad \mu = -\nu$$

Further, we can define the factors of the above determinant to be:

$$\Delta_1 \triangleq \frac{1}{JL} (JLs^2 + JRs + K_E K_T)$$

$$\Delta_2 \triangleq s^2 + \omega_c \sqrt{2}s + \omega_c^2 = \left( s + \omega_c \frac{\sqrt{2}}{2} \right)^2 + \left( \omega_c \frac{\sqrt{2}}{2} \right)^2$$

We can then find the transfer matrix as follows:

$$G(s) = (sI - A)^{-1} = \begin{bmatrix} \frac{s}{\Delta_1} & 0 & 0 & -\frac{K_E}{L\Delta_1} & 0 \\ \frac{s^2\omega_c}{\Delta_1\Delta_2} & \frac{s}{\Delta_2} & -\frac{\omega_c}{\Delta_2} & -\frac{s\omega_c K_E}{L\Delta_1\Delta_2} & 0 \\ \frac{s\omega_c^2}{\Delta_1\Delta_2} & \frac{\omega_c}{\Delta_2} & \frac{s+\omega_c\sqrt{2}}{\Delta_2} & -\frac{\omega_c^2 K_E}{L\Delta_1\Delta_2} & 0 \\ \frac{K_T}{J\Delta_1} & 0 & 0 & \frac{s+R/L}{\Delta_1} & 0 \\ \frac{K_T}{sJ\Delta_1} & 0 & 0 & \frac{s+R/L}{s\Delta_1} & \frac{1}{s} \end{bmatrix} \quad (4.37)$$

For a system of this size, it was more straightforward to find the transition matrix using the transfer matrix than using, e.g., similarity transforms or the Cayley-Hamilton theorem. The transition matrix is given by the inverse Laplace transform:

$$e^{At} = \mathcal{L}^{-1} [(sI - A)^{-1}]$$

### Laplace transform pairs

To find the time domain solutions for the terms introduced by the Butterworth filter, we need the following standard Laplace transform pairs:

$$\frac{1}{s-a} \stackrel{\mathcal{L}}{=} e^{at} \quad (4.38)$$

$$\frac{s-\alpha}{(s-a)(s-b)} \stackrel{\mathcal{L}}{=} \frac{1}{a-b} (ae^{at} - be^{bt}) \quad (4.39)$$

$$\frac{1}{(s-\alpha)^2 + \beta^2} \stackrel{\mathcal{L}}{=} \frac{1}{\beta} e^{\alpha t} \sin \beta t \quad (4.40)$$

$$\frac{1-\alpha}{(s-\alpha)^2 + \beta^2} \stackrel{\mathcal{L}}{=} e^{\alpha t} \cos \beta t \quad (4.41)$$

$$\frac{s}{(s-\alpha)^2 + \beta^2} \stackrel{\mathcal{L}}{=} e^{\alpha t} \left( \frac{\alpha}{\beta} \sin \beta t + \cos \beta t \right) \quad (4.42)$$

Using these, we find the inverse transform for the terms with both factors  $\Delta_1$  and  $\Delta_2$  in the denominator, starting with a general Laplace transform expressed with partial fractions as:

$$\begin{aligned} g(s) = \frac{k}{(s-a)(s-b)((s-\alpha)^2 + \beta^2)} &= \frac{A_1}{s-a} + \frac{A_2}{s-b} + \frac{Ms+N}{(s-\alpha)^2 + \beta^2} \Rightarrow \\ &A_1(s^3 - 2\alpha s^2 + \alpha^2 s + \beta s - bs^2 + 2abs - \alpha^2 b - \beta^2 b) + \\ &+ A_2(s^3 - 2\alpha s^2 + \alpha^2 s + \beta s - as^2 + 2\alpha as - \alpha^2 a - \beta^2 a) + \\ &+ M(s^3 - (a+b)s^2 + abs) + N(s^2 - (a+b)s + ab) = k \end{aligned} \quad (4.43)$$

Gathering the coefficients for each power of  $s$  gives the following system of equations:

$$\begin{aligned} A_1 + A_2 + M &= 0 \\ -2A_1\alpha - A_1b - 2A_2\alpha - A_2a - M(a+b) + N &= 0 \\ A_1\alpha^2 + A_1\beta^2 + 2A_1\alpha b + A_2\alpha^2 + A_2\beta^2 + 2A_2\alpha a + Mab - N(a+b) &= 0 \\ -A_1\alpha^2 b - A_1\beta^2 b - A_2\alpha^2 a - A_2\beta^2 a + Nab &= k \end{aligned}$$

Solving with respect to the coefficients in the partial fractions gives us:

$$\begin{aligned}
\delta_1 &\triangleq 2\alpha a - a^2 - \alpha^2 - \beta^2 \\
\delta_2 &\triangleq 2\alpha b - b^2 - \alpha^2 - \beta^2 \\
A_1(k_{ij}) &= \frac{k_{ij}}{\delta_1(b-a)} \\
A_2(k_{ij}) &= \frac{k_{ij}}{\delta_2(a-b)} \\
M(k_{ij}) &= \frac{k_{ij}(a+b-2\alpha)}{\delta_1\delta_2} \\
N(k_{ij}) &= \frac{k_{ij}(3\alpha^2 - \beta^2 - 2\alpha(a+b) + ab)}{\delta_1\delta_2}
\end{aligned}$$

The inverse transform of (4.43) and corresponding time derivatives (assuming a relaxed system) are:

$$\begin{aligned}
u &= \frac{N(k_{ij}) + \alpha M(k_{ij})}{\beta} \\
v &= M(k_{ij}) \\
f_1(k_{ij}, a, b, \alpha, \beta, t) &= \mathcal{L}^{-1}[g(s)] = A_1 e^{at} + A_2 e^{bt} + (u \sin \beta t + v \cos \beta t) e^{\alpha t} \quad (4.44)
\end{aligned}$$

$$\begin{aligned}
u &= \frac{\alpha N(k_{ij}) + \alpha^2 M(k_{ij}) - \beta^2 M(k_{ij})}{\beta} \\
v &= N(k_{ij}) + 2\alpha M(k_{ij}) \\
f_2(k_{ij}, a, b, \alpha, \beta, t) &= \mathcal{L}^{-1}[s \cdot g(s)] = aA_1 e^{at} + bA_2 e^{bt} + (u \sin \beta t + v \cos \beta t) e^{\alpha t} \quad (4.45)
\end{aligned}$$

$$\begin{aligned}
u &= \frac{\alpha^2 N(k_{ij}) + \alpha^3 M(k_{ij}) - \beta^2 N(k_{ij}) - 3\alpha\beta^2 M(k_{ij})}{\beta} \\
v &= 2\alpha N(k_{ij}) + 3\alpha^2 M(k_{ij}) - \beta^2 M(k_{ij}) \\
f_3(k_{ij}, a, b, \alpha, \beta, t) &= \mathcal{L}^{-1}[s^2 \cdot g(s)] = a^2 A_1 e^{at} + b^2 A_2 e^{bt} + (u \sin \beta t + v \cos \beta t) e^{\alpha t} \quad (4.46)
\end{aligned}$$

### Transition matrix

We can now set up, inserting for the sampling time,  $T$ , the discrete state matrix for the system, using the elements already found for the discretized model of the motor in Equation (3.43).

$$e^{AT} = \Phi = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} & \phi_{35} \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} & \phi_{45} \\ \phi_{51} & \phi_{52} & \phi_{53} & \phi_{44} & \phi_{55} \end{bmatrix} \quad (4.47)$$

$$\phi_{12} = \phi_{13} = \phi_{15} = \phi_{25} = \phi_{35} = \phi_{42} = \phi_{43} = \phi_{45} = \phi_{52} = \phi_{53} = 0$$

$$\begin{aligned} \phi_{11} &= \frac{1}{\lambda_1 - \lambda_2} \left( \lambda_1 e^{\lambda_1 T} - \lambda_2 e^{\lambda_2 T} \right) \\ \phi_{14} &= \frac{1}{\lambda_1 - \lambda_2} \frac{K_E}{L} \left( e^{\lambda_2 T} - e^{\lambda_1 T} \right) \\ \phi_{21} &= f_3(\omega_c, \lambda_1, \lambda_2, \mu, \nu, T) \\ \phi_{22} &= e^{\mu T} (\cos \nu T - \sin \nu T) \\ \phi_{23} &= -\omega_c \frac{1}{\nu} e^{\mu T} \sin \nu T \\ \phi_{24} &= f_2\left(-\omega_c \frac{K_E}{L}, \lambda_1, \lambda_2, \mu, \nu, T\right) \\ \phi_{31} &= f_2(\omega_c^2, \lambda_1, \lambda_2, \mu, \nu, T) \\ \phi_{32} &= \omega_c \frac{1}{\nu} e^{\mu T} \sin \nu T = -\phi_{23} \\ \phi_{33} &= e^{\mu T} (\sin \nu T + \cos \nu T) \\ \phi_{34} &= f_1\left(-\omega_c^2 \frac{K_E}{L}, \lambda_1, \lambda_2, \mu, \nu, T\right) \\ \phi_{41} &= \frac{1}{\lambda_1 - \lambda_2} \frac{K_T}{J} \left( e^{\lambda_1 T} - e^{\lambda_2 T} \right) \\ \phi_{44} &= \frac{1}{\lambda_1 - \lambda_2} \left( \lambda_1 e^{\lambda_2 T} - \lambda_2 e^{\lambda_1 T} \right) \\ \phi_{51} &= \frac{1}{\lambda_1 - \lambda_2} \frac{K_T}{J} \left( \frac{1}{\lambda_1} \left( e^{\lambda_1 T} - 1 \right) - \frac{1}{\lambda_2} \left( e^{\lambda_2 T} - 1 \right) \right) \\ \phi_{54} &= \frac{1}{\lambda_1 - \lambda_2} \left( \frac{\lambda_1}{\lambda_2} \left( e^{\lambda_2 T} - 1 \right) - \frac{\lambda_2}{\lambda_1} \left( e^{\lambda_1 T} - 1 \right) \right) \\ \phi_{55} &= 1 \end{aligned}$$

The input matrix for the system will now be given as:

$$B = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -\frac{1}{J} \\ 0 & 0 \end{bmatrix}$$

To find the exact discretization easily, we use the elements already found in Equation (3.44), and for the rest, we can find the terms introduced by the Butterworth filter using the relation of time integration in the  $s$ -domain:

$$\mathcal{L} \left[ \int_0^t f(\tau) d\tau \right] = \frac{1}{s} \mathcal{L} [f]$$

For the integration of the transition matrix, this yields:

$$\mathcal{L} \left[ \int_0^T e^{A\tau} d\tau \right] = \frac{1}{s} (sI - A)^{-1}$$

It is now straightforward to find two of the needed expressions, since they will be the same as Equations (4.45) and (4.44):

$$\begin{aligned} \mathcal{L}^{-1} \left[ \frac{1}{s^2} \cdot g(s) \right] &= f_2(k_{ij}, a, b, \alpha, \beta, t) \\ \mathcal{L}^{-1} \left[ \frac{1}{s} \cdot g(s) \right] &= f_1(k_{ij}, a, b, \alpha, \beta, t) \end{aligned}$$

The last one will be given by:

$$\begin{aligned} u &= \frac{N(k_{ij}) + \alpha M(k_{ij})}{\beta} \\ v &= M(k_{ij}) \end{aligned}$$

$$\begin{aligned} f_0(k_{ij}, a, b, \alpha, \beta, t) &= \int_0^t f_1(k_{ij}, a, b, \alpha, \beta, \tau) d\tau = \\ &= A_1 \frac{1}{a} (e^{at} - 1) + A_2 \frac{1}{b} (e^{bt} - 1) + \\ &+ (u ((\alpha \sin \beta t - \beta \cos \beta t) e^{\alpha t} + \beta) + v ((\alpha \cos \beta t + \beta \sin \beta t) e^{\alpha t} - \alpha)) \frac{1}{\alpha^2 + \beta^2} \end{aligned} \quad (4.48)$$

$$\left( \int_0^T e^{A\tau} d\tau \right) B = \Delta = \begin{bmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \\ \delta_{31} & \delta_{32} \\ \delta_{41} & \delta_{42} \\ \delta_{51} & \delta_{52} \end{bmatrix} \quad (4.49)$$

$$\begin{aligned}
\delta_{11} &= \frac{1}{\lambda_1 - \lambda_2} \frac{1}{L} \left( e^{\lambda_1 T} - e^{\lambda_2 T} \right) = \frac{J}{K_T L} \phi_{41} \\
\delta_{21} &= f_2 \left( \omega_c \frac{1}{L}, \lambda_1, \lambda_2, \mu, \nu, T \right) \\
\delta_{31} &= f_1 \left( \omega_c^2 \frac{1}{L}, \lambda_1, \lambda_2, \mu, \nu, T \right) \\
\delta_{41} &= \frac{1}{\lambda_1 - \lambda_2} \frac{K_T}{JL} \left( \frac{1}{\lambda_1} \left( e^{\lambda_1 T} - 1 \right) - \frac{1}{\lambda_2} \left( e^{\lambda_2 T} - 1 \right) \right) = \frac{1}{L} \phi_{51} \\
\delta_{51} &= \frac{1}{\lambda_1 - \lambda_2} \frac{K_T}{JL} \left( \frac{1}{\lambda_1^2} \left( e^{\lambda_1 T} - 1 \right) - \frac{1}{\lambda_2^2} \left( e^{\lambda_2 T} - 1 \right) + \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) T \right) \\
\delta_{12} &= \frac{1}{\lambda_1 - \lambda_2} \frac{K_E}{JL} \left( \frac{1}{\lambda_1} \left( e^{\lambda_1 T} - 1 \right) - \frac{1}{\lambda_2} \left( e^{\lambda_2 T} - 1 \right) \right) = \frac{K_E}{K_T L} \phi_{51} \\
\delta_{22} &= f_1 \left( \omega_c \frac{K_E}{JL}, \lambda_1, \lambda_2, \mu, \nu, T \right) \\
\delta_{32} &= f_0 \left( \omega_c^2 \frac{K_E}{JL}, \lambda_1, \lambda_2, \mu, \nu, T \right) \\
\delta_{42} &= \frac{1}{\lambda_1 - \lambda_2} \frac{-1}{J} \left( \frac{\lambda_1}{\lambda_2} \left( e^{\lambda_2 T} - 1 \right) - \frac{\lambda_2}{\lambda_1} \left( e^{\lambda_1 T} - 1 \right) \right) = -\frac{1}{J} \phi_{54} \\
\delta_{52} &= \frac{1}{\lambda_1 - \lambda_2} \frac{-1}{J} \left( \frac{\lambda_1}{\lambda_2^2} \left( e^{\lambda_2 T} - 1 \right) - \frac{\lambda_2}{\lambda_1^2} \left( e^{\lambda_1 T} - 1 \right) + \left( \frac{\lambda_2}{\lambda_1} - \frac{\lambda_1}{\lambda_2} \right) T \right)
\end{aligned}$$

#### 4.5.6 Complete system model

Since the full state-space description is used, the load  $T_L$ , must be estimated as an unknown exogeneous input. The current measurement is known to have an unknown varying bias component,  $i_b$ , which needs to be estimated in order to improve the accuracy of all the estimated quantities. Setting up the system as in (4.26) yields the following description (interchanging  $\theta$  with  $\alpha$  so as not to confuse angular velocity with unknown parameters)

$$x_{k+1} = \Phi(\alpha)x_k + \Delta_u(\alpha)u_k + \Delta_{T_L}(\alpha)T_{Lk} + \Delta_{i_b}(\alpha)i_{bk+1} + w_k^x \quad (4.50)$$

$$T_{Lk+1} = T_{Lk} + w_k^{T_L} \quad (4.51)$$

$$i_{bk+1} = i_{bk} + w_k^{i_b} \quad (4.52)$$

$$\alpha_{k+1} = \alpha_k + w_k^\alpha \quad (4.53)$$

$$y_k = C(\alpha)x_k + C_{i_b}(\alpha)i_{bk} + v_k, \quad (4.54)$$

where  $\Phi$  is as defined in (4.47), and we split (4.49) into

$$\Delta_u(\alpha) = \begin{bmatrix} \delta_{11} \\ \delta_{21} \\ \delta_{31} \\ \delta_{41} \\ \delta_{51} \end{bmatrix}, \quad \Delta_{T_L}(\alpha) = \begin{bmatrix} \delta_{12} \\ \delta_{22} \\ \delta_{32} \\ \delta_{42} \\ \delta_{52} \end{bmatrix}.$$

The current bias should be present only in the measurement, therefore  $\Delta_{i_b} = 0$ . The augmented state and noise sequence vectors will be

$$\chi_k = \begin{bmatrix} i_k \\ i'_{fk} \\ i_{fk} \\ \omega_k \\ \theta_k \\ T_{Lk} \\ i_{bk} \\ \alpha_k \end{bmatrix}, \quad w_k = \begin{bmatrix} w_k^x \\ w_k^{T_L} \\ w_k^{i_b} \\ w_k^\alpha \end{bmatrix},$$

and the state and input matrices will be

$$F(\alpha) = \begin{bmatrix} \Phi(\alpha) & \Delta_{T_L}(\alpha) & 0 & 0 \\ 0 & I_r & 0 & 0 \\ 0 & 0 & I_q & 0 \\ 0 & 0 & 0 & I_p \end{bmatrix}$$

$$G(\alpha) = \begin{bmatrix} \Delta_u(\alpha) \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The physical measurements available in this context will be the current and the angular position, thus

$$y_k = \begin{bmatrix} i_{fk} + i_{bk} \\ \theta_k \end{bmatrix}, \quad v_k = \begin{bmatrix} v_k^{i_f} \\ v_k^\theta \end{bmatrix}.$$

This gives us the output matrices

$$C(\alpha) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad C_{i_b}(\alpha) = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

which in the augmented system will be

$$H(\alpha) = [ C(\alpha) \quad 0 \quad C_{i_b}(\alpha) \quad 0 ].$$

If we choose to estimate one parameter from the model, the state, input and output matrices will then be:

$$F(\alpha) = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} & \delta_{12} & 0 & 0 \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} & \delta_{22} & 0 & 0 \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} & \phi_{35} & \delta_{32} & 0 & 0 \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} & \phi_{45} & \delta_{42} & 0 & 0 \\ \phi_{51} & \phi_{52} & \phi_{53} & \phi_{44} & \phi_{55} & \delta_{52} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad G(\alpha) = \begin{bmatrix} \delta_{11} \\ \delta_{21} \\ \delta_{31} \\ \delta_{41} \\ \delta_{51} \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

#### 4.5.7 Jacobians

In order to estimate model parameters with the EKF, we need to find the linearization of the system's equations about the filter's estimated trajectory. Thus, we need the Jacobians as defined in (4.33) and (4.34)

$$\frac{\partial}{\partial \theta_k^T} [F(\theta_k) \chi_k + G(\theta_k) u_k], \quad \frac{\partial}{\partial \theta_k^T} [H(\theta_k) \chi_k].$$

For the rest of this section, we change  $\alpha$  back to  $\theta$  to denote parameters. We can immediately see that

$$\frac{\partial}{\partial \theta_k^T} [H(\theta_k) \chi_k] = 0,$$

since no parameters appear in the output matrix. For the motor, the parameters we can linearize for are:

$$\theta \in \{J, K_E, K_T, L, R\}$$

For convenience, we define some symbols to save some writing:

$$\begin{aligned} l_1 &= \lambda_1(\theta) \\ l_2 &= \lambda_2(\theta) \\ e_1 &= e^{\lambda_1(\theta)T} = e^{l_1 T} \\ e_2 &= e^{\lambda_2(\theta)T} = e^{l_2 T} \\ w &= \lambda_1(\theta) - \lambda_2(\theta) = l_1 - l_2 \end{aligned}$$

The partial derivatives of these expressions with respect to some parameter  $\theta$  are:

$$l'_1 = \frac{\partial l_1}{\partial \theta}, \quad l'_2 = \frac{\partial l_2}{\partial \theta}, \quad e'_1 = \frac{\partial e_1}{\partial \theta} = l'_1 T e_1, \quad e'_2 = \frac{\partial e_2}{\partial \theta} = l'_2 T e_2, \quad w' = \frac{\partial w}{\partial \theta} = l'_1 - l'_2$$

From the discretization of the motor with Butterworth filter model, we know that the following elements must be zero:

$$\begin{aligned} \frac{\partial}{\partial \theta} \phi_{12} = \frac{\partial}{\partial \theta} \phi_{13} = \frac{\partial}{\partial \theta} \phi_{15} = \frac{\partial}{\partial \theta} \phi_{25} = \frac{\partial}{\partial \theta} \phi_{35} = \\ = \frac{\partial}{\partial \theta} \phi_{42} = \frac{\partial}{\partial \theta} \phi_{43} = \frac{\partial}{\partial \theta} \phi_{45} = \frac{\partial}{\partial \theta} \phi_{52} = \frac{\partial}{\partial \theta} \phi_{53} = 0 \end{aligned}$$

Also, if we restrict ourselves to parameters for the motor, elements with parameters due solely to the Butterworth filter will be zero when linearizing with respect to motor parameters:

$$\frac{\partial}{\partial \theta} \phi_{22} = \frac{\partial}{\partial \theta} \phi_{23} = \frac{\partial}{\partial \theta} \phi_{32} = \frac{\partial}{\partial \theta} \phi_{33} = 0$$

In addition to these,  $\frac{\partial}{\partial \theta} \phi_{55} = \frac{\partial}{\partial \theta} (1) = 0$ .

### Pattern for elements related to the motor alone

All the elements related to the motor alone will have the following pattern:

$$\begin{aligned} \phi_{ij} &= k_{ij}(\theta) \cdot \frac{1}{w} \cdot g_{ij}(l_1, l_2, e_1, e_2) \\ k_{ij}' &= \frac{\partial}{\partial \theta} k_{ij}(\theta), \quad g_{ij}' = \frac{\partial}{\partial \theta} g_{ij}(l_1, l_2, e_1, e_2) \end{aligned}$$

All of the derivatives with respect to a parameter,  $\theta$ , will thus have the form:

$$\frac{\partial}{\partial \theta} \phi_{ij} = k_{ij} \frac{(g_{ij}' w - g_{ij} w')}{w^2} + k_{ij}' \frac{1}{w} g_{ij}$$

We can therefore calculate all these derivatives finding  $g_{ij}'$  and  $k_{ij}'$  for the corresponding elements in the discrete state and input matrices.

### Pattern for elements related to the motor and the Butterworth filter

Looking at the functions (4.48), (4.44), (4.45) and (4.46), the arguments involving motor parameters will lead to expressions in terms of the partial fractions as:

$$a(\theta), \quad b(\theta), \quad k_{ij}(\theta) \quad \Rightarrow \quad \delta_1(\theta), \quad \delta_2(\theta), \quad A_1(\theta), \quad A_2(\theta), \quad M(\theta), \quad N(\theta)$$

Finding the derivatives of  $a(\theta)$ ,  $b(\theta)$ ,  $k_{ij}(\theta)$

$$\frac{\partial}{\partial \theta} a = a', \quad \frac{\partial}{\partial \theta} b = b', \quad \frac{\partial}{\partial \theta} k_{ij} = k_{ij}'$$

we can now find the partial derivatives for these functions with respect to some parameter  $\theta$ , using the following expressions.

$$\begin{aligned} \delta_1 &= 2\alpha a - a^2 - \alpha^2 - \beta^2 \\ \delta_1' &= \frac{\partial}{\partial \theta} \delta_1 = 2\alpha a' - 2aa' \end{aligned}$$

$$\delta_2 = 2\alpha b - b^2 - \alpha^2 - \beta^2$$

$$\delta_2' = \frac{\partial}{\partial \theta} \delta_2 = 2\alpha b' - 2bb'$$

$$A_1 = \frac{k_{ij}}{\delta_1(b-a)}$$

$$A_1' = \frac{\partial}{\partial \theta} A_1 = \frac{k_{ij}' \delta_1(b-a) - k_{ij} (\delta_1'(b-a) + \delta_1(b'-a))}{(\delta_1(b-a))^2}$$

$$A_2 = \frac{k_{ij}}{\delta_2(a-b)}$$

$$A_2' = \frac{\partial}{\partial \theta} A_2 = \frac{k_{ij}' \delta_2(a-b) - k_{ij} (\delta_2'(a-b) + \delta_2(a'-b))}{(\delta_2(a-b))^2}$$

$$M = \frac{k_{ij}(a+b-2\alpha)}{\delta_1 \delta_2}$$

$$M' = \frac{\partial}{\partial \theta} M = \frac{(k_{ij}'(a+b-2\alpha) + k_{ij}(a'+b')) \delta_1 \delta_2 - k_{ij}(a+b-2\alpha)(\delta_1' \delta_2 + \delta_1 \delta_2')}{(\delta_1 \delta_2)^2}$$

$$N = \frac{k_{ij}(3\alpha^2 - \beta^2 - 2\alpha(a+b) + ab)}{\delta_1 \delta_2}$$

$$p = (k_{ij}'(3\alpha^2 - \beta^2 - 2\alpha(a+b) + ab) + k_{ij}(-2\alpha(a'+b') + a'b + ab')) \delta_1 \delta_2 - k_{ij}(3\alpha^2 - \beta^2 - 2\alpha(a+b) + ab)(\delta_1' \delta_2 + \delta_1 \delta_2')$$

$$N' = \frac{\partial}{\partial \theta} N = \frac{p}{(\delta_1 \delta_2)^2}$$

The partial derivatives of the functions are then given by:

$$\begin{aligned} \frac{\partial}{\partial \theta} f_0(k_{ij}, a, b, \alpha, \beta, t) &= f_0'(k_{ij}, k_{ij}', a, b, a', b', \alpha, \beta, t) = \\ &A_1' \frac{1}{a} (e^{at} - 1) + A_1 \frac{1}{a^2} (a'te^{at}a - (e^{at} - 1)a') + \\ &+ A_2' \frac{1}{b} (e^{bt} - 1) + A_2 \frac{1}{b^2} (b'te^{bt}b - (e^{bt} - 1)b') + \\ &+ (u'((\alpha \sin \beta t - \beta \cos \beta t)e^{\alpha t} + \beta) + v'((\alpha \cos \beta t + \beta \sin \beta t)e^{\alpha t} - \alpha)) \frac{1}{\alpha^2 + \beta^2} \end{aligned}$$

where

$$u' = \frac{\partial}{\partial \theta} u(\theta) = \frac{N' + \alpha M'}{\beta}$$

$$v' = \frac{\partial}{\partial \theta} v(\theta) = M'$$

$$\frac{\partial}{\partial \theta} f_1(k_{ij}, a, b, \alpha, \beta, t) = f'_1(k_{ij}, k'_{ij}, a, b, a', b', \alpha, \beta, t) =$$

$$A_1' e^{at} + A_1 a' t e^{at} + A_2' e^{bt} + A_2 b' t e^{bt} + (u' \sin \beta t + v' \cos \beta t) e^{\alpha t}$$

where

$$u' = \frac{\partial}{\partial \theta} u(\theta) = \frac{N' + \alpha M'}{\beta}$$

$$v' = \frac{\partial}{\partial \theta} v(\theta) = M'$$

$$\frac{\partial}{\partial \theta} f_2(k_{ij}, a, b, \alpha, \beta, t) = f'_2(k_{ij}, k'_{ij}, a, b, a', b', \alpha, \beta, t) =$$

$$a' A_1 e^{at} + a A_1' e^{at} + a A_1 a' t e^{at} + b' A_2 e^{bt} + b A_2' e^{bt} + b A_2 b' t e^{bt} +$$

$$+ (u' \sin \beta t + v' \cos \beta t) e^{\alpha t}$$

where

$$u' = \frac{\partial}{\partial \theta} u(\theta) = \frac{\alpha N' + \alpha^2 M' - \beta^2 M'}{\beta}$$

$$v' = \frac{\partial}{\partial \theta} v(\theta) = N' + 2\alpha M'$$

$$\frac{\partial}{\partial \theta} f_3(k_{ij}, a, b, \alpha, \beta, t) = f'_3(k_{ij}, k'_{ij}, a, b, a', b', \alpha, \beta, t) =$$

$$2aa' A_1 e^{at} + a^2 A_1' e^{at} + a^2 A_1 a' t e^{at} + 2bb' A_2 e^{bt} + b^2 A_2' e^{bt} + b^2 A_2 b' t e^{bt} +$$

$$+ (u' \sin \beta t + v' \cos \beta t) e^{\alpha t}$$

where

$$u' = \frac{\partial}{\partial \theta} u(\theta) = \frac{\alpha^2 N' + \alpha^3 M' - \beta^2 N' - 3\alpha \beta^2 M'}{\beta}$$

$$v' = \frac{\partial}{\partial \theta} v(\theta) = 2\alpha N' + 3\alpha^2 M' - \beta^2 M'$$

## Matrix elements

$$\frac{\partial}{\partial \theta} \phi_{11}$$

$$k_{11}^{\phi} = 1, \quad g_{11} = l_1 e_1 - l_2 e_2, \quad g_{11}' = l_1' e_1 + l_1 e_1' - l_2' e_2 - l_2 e_2', \quad k_{11}^{\phi'} = \frac{\partial}{\partial \theta} (1) = 0$$

$$\frac{\partial}{\partial \theta} \phi_{14}$$

$$k_{14}^{\phi} = \frac{K_E}{L}, \quad g_{14} = e_2 - e_1, \quad g_{14}' = e_2' - e_1', \quad k_{14}^{\phi'} = \frac{\partial}{\partial \theta} \left( \frac{K_E}{L} \right)$$

$$\frac{\partial}{\partial \theta} \phi_{21}$$

$$k_{21}^{\phi} = \omega_c, \quad k_{21}^{\phi'} = \frac{\partial}{\partial \theta} (\omega_c), \quad f_3'(k_{21}^{\phi}, k_{21}^{\phi'}, l_1, l_2, l_1', l_2', \mu, \nu, T)$$

$$\frac{\partial}{\partial \theta} \phi_{24}$$

$$k_{24}^{\phi} = -\omega_c \frac{K_E}{L}, \quad k_{24}^{\phi'} = \frac{\partial}{\partial \theta} \left( -\omega_c \frac{K_E}{L} \right), \quad f_2'(k_{24}^{\phi}, k_{24}^{\phi'}, l_1, l_2, l_1', l_2', \mu, \nu, T)$$

$$\frac{\partial}{\partial \theta} \phi_{31}$$

$$k_{31}^{\phi} = \omega_c^2, \quad k_{31}^{\phi'} = \frac{\partial}{\partial \theta} (\omega_c^2), \quad f_2'(k_{31}^{\phi}, k_{31}^{\phi'}, l_1, l_2, l_1', l_2', \mu, \nu, T)$$

$$\frac{\partial}{\partial \theta} \phi_{34}$$

$$k_{34}^{\phi} = -\omega_c^2 \frac{K_E}{L}, \quad k_{34}^{\phi'} = \frac{\partial}{\partial \theta} \left( -\omega_c^2 \frac{K_E}{L} \right), \quad f_1'(k_{34}^{\phi}, k_{34}^{\phi'}, l_1, l_2, l_1', l_2', \mu, \nu, T)$$

$$\frac{\partial}{\partial \theta} \phi_{41}$$

$$k_{41}^{\phi} = \frac{K_T}{J}, \quad g_{44} = e_1 - e_2, \quad g_{41}' = e_1' - e_2', \quad k_{41}^{\phi'} = \frac{\partial}{\partial \theta} \left( \frac{K_T}{J} \right)$$

$$\frac{\partial}{\partial \theta} \phi_{44}$$

$$k_{44}^{\phi} = 1, \quad g_{44} = l_1 e_2 - l_2 e_1, \quad g_{44}' = l_1' e_2 + l_1 e_2' - l_2' e_1 - l_2 e_1', \quad k_{44}^{\phi'} = \frac{\partial}{\partial \theta} (1) = 0$$

$$\frac{\partial}{\partial \theta} \phi_{51}$$

$$k_{51}^{\phi} = \frac{K_T}{J}, \quad g_{51} = \frac{1}{l_1} (e_1 - 1) - \frac{1}{l_2} (e_2 - 1),$$

$$g_{51}' = -\frac{l_1'}{l_1^2} (e_1 - 1) + \frac{1}{l_1} e_1' + \frac{l_2'}{l_2^2} (e_2 - 1) - \frac{1}{l_2} e_2',$$

$$k_{51}^{\phi'} = \frac{\partial}{\partial \theta} \left( \frac{K_T}{J} \right)$$

$$\frac{\partial}{\partial \theta} \phi_{54}$$

$$\begin{aligned} k_{54}^{\phi} &= 1, \quad g_{54} = \frac{l_1}{l_2} (e_2 - 1) - \frac{l_2}{l_1} (e_1 - 1), \\ g_{54}' &= \frac{l_1' l_2 - l_1 l_2'}{l_2^2} (e_2 - 1) + \frac{l_1}{l_2} e_2' - \frac{l_2' l_1 - l_2 l_1'}{l_1^2} (e_1 - 1) - \frac{l_2}{l_1} e_1', \\ k_{54}^{\phi'} &= \frac{\partial}{\partial \theta} (1) = 0 \end{aligned}$$

$$\frac{\partial}{\partial \theta} \delta_{11}$$

$$k_{11}^{\delta} = \frac{J}{K_T L}, \quad \frac{\partial}{\partial \theta} \delta_{11} = \frac{\partial}{\partial \theta} (k_{11}^{\delta} \cdot \phi_{41}) = k_{11}^{\delta'} \phi_{41} + k_{11}^{\delta} \frac{\partial}{\partial \theta} \phi_{41}, \quad k_{11}^{\delta'} = \frac{\partial}{\partial \theta} \left( \frac{J}{K_T L} \right)$$

$$\frac{\partial}{\partial \theta} \delta_{21}$$

$$k_{21}^{\delta} = \omega_c \frac{1}{L}, \quad k_{21}^{\delta'} = \frac{\partial}{\partial \theta} \left( \omega_c \frac{1}{L} \right), \quad f_2'(k_{21}^{\delta}, k_{21}^{\delta'}, l_1, l_2, l_1', l_2', \mu, \nu, T)$$

$$\frac{\partial}{\partial \theta} \delta_{31}$$

$$k_{31}^{\delta} = \omega_c^2 \frac{1}{L}, \quad k_{31}^{\delta'} = \frac{\partial}{\partial \theta} \left( \omega_c^2 \frac{1}{L} \right), \quad f_1'(k_{31}^{\delta}, k_{31}^{\delta'}, l_1, l_2, l_1', l_2', \mu, \nu, T)$$

$$\frac{\partial}{\partial \theta} \delta_{41}$$

$$k_{41}^{\delta} = \frac{1}{L}, \quad \frac{\partial}{\partial \theta} \delta_{41} = \frac{\partial}{\partial \theta} (k_{41}^{\delta} \cdot \phi_{51}) = k_{41}^{\delta'} \phi_{51} + k_{41}^{\delta} \frac{\partial}{\partial \theta} \phi_{51}, \quad k_{41}^{\delta'} = \frac{\partial}{\partial \theta} \left( \frac{1}{L} \right)$$

$$\frac{\partial}{\partial \theta} \delta_{51}$$

$$\begin{aligned} k_{51}^{\delta} &= \frac{K_T}{JL}, \quad h_{51} = \frac{1}{l_1^2} (e_1 - 1) - \frac{1}{l_2^2} (e_2 - 1) + \left( \frac{1}{l_2} - \frac{1}{l_1} \right) T, \\ h_{51}' &= -\frac{2l_1 l_1'}{l_1^4} (e_1 - 1) + \frac{1}{l_1^2} e_1' + \frac{2l_2 l_2'}{l_2^4} (e_2 - 1) - \frac{1}{l_2^2} e_2' + \left( \frac{l_1'}{l_1^2} - \frac{l_2'}{l_2^2} \right) T, \\ k_{51}^{\delta'} &= \frac{\partial}{\partial \theta} \left( \frac{K_T}{JL} \right) \end{aligned}$$

$$\frac{\partial}{\partial \theta} \delta_{12}$$

$$k_{12}^{\delta} = \frac{K_E}{K_T L}, \quad \frac{\partial}{\partial \theta} \delta_{12} = \frac{\partial}{\partial \theta} (k_{12}^{\delta} \cdot \phi_{51}) = k_{12}^{\delta'} \phi_{51} + k_{12}^{\delta} \frac{\partial}{\partial \theta} \phi_{51}, \quad k_{12}^{\delta'} = \frac{\partial}{\partial \theta} \left( \frac{K_E}{K_T L} \right)$$

$$\frac{\partial}{\partial \theta} \delta_{22}$$

$$k_{22}^{\delta} = \omega_c \frac{K_E}{JL}, \quad k_{22}^{\delta'} = \frac{\partial}{\partial \theta} \left( \omega_c \frac{K_E}{JL} \right), \quad f_1'(k_{22}^{\delta}, k_{22}^{\delta'}, l_1, l_2, l_1', l_2', \mu, \nu, T)$$

$$\frac{\partial}{\partial \theta} \delta_{32}$$

$$k_{32}^\delta = \omega_c^2 \frac{K_E}{JL}, \quad k_{32}^{\delta'} = \frac{\partial}{\partial \theta} \left( \omega_c^2 \frac{K_E}{JL} \right), \quad f'_0(k_{32}^\delta, k_{32}^{\delta'}, l_1, l_2, l'_1, l'_2, \mu, \nu, T)$$

$$\frac{\partial}{\partial \theta} \delta_{42}$$

$$k_{42}^\delta = \frac{-1}{J}, \quad \frac{\partial}{\partial \theta} \delta_{42} = \frac{\partial}{\partial \theta} (k_{42}^\delta \cdot \phi_{54}) = k_{42}^{\delta'} \phi_{54} + k_{42}^\delta \frac{\partial}{\partial \theta} \phi_{54}, \quad k_{42}^{\delta'} = \frac{\partial}{\partial \theta} \left( \frac{-1}{J} \right)$$

$$\frac{\partial}{\partial \theta} \delta_{52}$$

$$k_{52}^\delta = \frac{-1}{J}, \quad h_{52} = \frac{l_1}{l_2^2} (e_2 - 1) - \frac{l_2}{l_1^2} (e_1 - 1) + \left( \frac{l_2}{l_1} - \frac{l_1}{l_2} \right) T,$$

$$h_{52}' = \frac{l_1' l_2^2 - l_1 2 l_2 l_2'}{l_2^4} (e_2 - 1) + \frac{l_1}{l_2^2} e_2' - \frac{l_2' l_1^2 - l_2 2 l_1 l_1'}{l_1^4} (e_1 - 1) - \frac{l_2}{l_1^2} e_1' +$$

$$+ \left( \frac{l_2' l_1 - l_2 l_1'}{l_1^2} - \frac{l_1' l_2 - l_1 l_2'}{l_2^2} \right) T,$$

$$k_{52}^{\delta'} = \frac{\partial}{\partial \theta} \left( \frac{-1}{J} \right)$$

Calculating the partial derivatives (and thus the Jacobians) this way, we need only specify the partial derivatives for the eigenvalues,  $\lambda_{1,2} = l_{1,2}$ , and the constants appearing in some of the expressions,  $k_{ij}$ , with respect to the chosen parameter.

### The partial derivatives of the eigenvalues for some motor parameters

The partial derivatives with respect to resistance are:

$$l'_{1R} = \frac{\partial}{\partial R} \lambda_1 = -\frac{1}{2L} \left( 1 + \frac{JR}{\sqrt{J^2 R^2 - 4JLK_T K_E}} \right)$$

$$l'_{2R} = \frac{\partial}{\partial R} \lambda_2 = -\frac{1}{2L} \left( 1 - \frac{JR}{\sqrt{J^2 R^2 - 4JLK_T K_E}} \right)$$

The partial derivatives with respect to the electromagnetic constant are, assuming  $K_T = K_E = K$ :

$$l'_{1K} = \frac{\partial}{\partial K} \lambda_1 = \frac{2K}{\sqrt{J^2 R^2 - 4JLK^2}}$$

$$l'_{2K} = \frac{\partial}{\partial K} \lambda_2 = -\frac{2K}{\sqrt{J^2 R^2 - 4JLK^2}}$$

### 4.5.8 Observability

To be able to estimate states in a given system using a Kalman filter, the states of the system must be observable. In the case where we have augmented the state vector to include unknown inputs, biases and parameters, the new states must also be observable, which is a stricter requirement. The system must also be driven by a persistently exciting signal of sufficient order, to be sure that the parameter estimates converge. Discretization and nonlinearities might remedy the situation somewhat.

The observability of a system can be determined by computing the rank of the observability matrix

$$Ob = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}.$$

If  $\text{rank}(Ob) = n$ , the system is observable. Finding the rank numerically with finite precision arithmetic is unreliable. Finding the singular value decomposition of  $Ob$  and looking at the singular values should tell us about the effective rank of the system [17]. Small singular values mean that the corresponding states might be marginally observable. For the Kalman filter, this means that estimates of any states which yield small singular values from the observability matrix will either converge slowly or diverge.

Observability analysis for the linearized filter is provided in the MATLAB script described in Appendix B.1.1. Here, we have calculated the singular values for various combinations of unknown current measurement bias, unknown exogenous input (torque load) and different motor parameters, with and without a velocity measurement. The system seems to have a limit of seven well-conditioned observable states without a velocity measurement, and eight well-conditioned observable states when adding a velocity measurement. Adding additional states yielded small singular values, meaning that they might be marginally observable. In practice, the extended Kalman filter implementation performed well when the process noise variances were tuned properly.

## 4.6 Regression with least squares

To find the parameters in the models given in Section 2.3.1, one can use regression on datasets that show the relationship between the current and the resistance. The models above are nonlinear with respect to both the free variable  $x$ , and the parameters,  $\theta_{S_1} = (a, b, c)^\top$ ,  $\theta_{S_2} = (\alpha, \beta, \gamma)^\top$ . This can be solved as a nonlinear least squares problem.

In [16], the least squares problem is formulated using an objective function of the form

$$f(\theta) = \frac{1}{2} \sum_{j=0}^m r_j^2(\theta), \quad (4.55)$$

where  $r_j$  is called a residual and is a smooth function with mapping  $\mathbb{R}^n \rightarrow \mathbb{R}$ . That is, there are  $n$  parameters and  $m$  measurements. If we define the residual as

$$r_j(\theta) = y_j - S_p(\theta; x_j), \quad (4.56)$$

taking  $y$  to be a measurement and  $S_p(\theta; x_j)$  to be the value of the chosen model,  $p \in \{1, 2\}$ , at the point where the measurement was taken, we can subject Equation (4.55) to

$$\min_{\theta} f(\theta) = \min_{\theta} \frac{1}{2} \sum_{j=0}^m (y_j - S_k(\theta; x_j))^2 \quad (4.57)$$

and find a fit for the parameters for the chosen model.

Because of the way the objective function is formulated, the gradient and the Hessian will have a special structure. With the Jacobian of  $r(\theta) = (r_1(\theta), r_2(\theta), \dots, r_m(\theta))^{\top}$  defined as

$$J(\theta) \triangleq \begin{bmatrix} \partial r_j \\ \partial \theta_i \end{bmatrix} \quad \begin{matrix} j = 1, 2, \dots, m \\ i = 1, 2, \dots, n \end{matrix}$$

(here  $J(\theta)$  denotes the Jacobian, not the cost criterion) we get the gradient using the chain rule as

$$\nabla f(\theta) = \sum_{j=0}^m r_j(\theta) \nabla r_j(\theta) = J(\theta)^{\top} r(\theta), \quad (4.58)$$

and taking the Jacobian of the gradient we get the Hessian as

$$\nabla^2 f(\theta) = \sum_{j=0}^m \nabla r_j(\theta) \nabla r_j(\theta)^{\top} + \sum_{j=0}^m r_j(\theta) \nabla^2 r_j(\theta) = J(\theta)^{\top} J(\theta) + \sum_{j=0}^m r_j(\theta) \nabla^2 r_j(\theta). \quad (4.59)$$

Finding (4.57) can be done using a line search method, where one iteratively searches along a chosen direction  $p_k$  from the current iterate  $\theta_k$  for a new iterate  $\theta_{k+1}$ , which should yield a lower value for the objective function (4.55):

$$\theta_{k+1} = \theta_k + \mu_k p_k$$

$\mu_k$  is the step length along the chosen direction. The search direction typically has the form

$$p_k = -B_k^{-1} \nabla f_k(\theta),$$

where  $B_k$  is a symmetric and nonsingular matrix. The steepest descent direction is given by the identity matrix  $B_k = I$ . Newton's method uses the Hessian  $B_k = \nabla^2 f(\theta_k)$ . In the least squares problem, one often uses an approximation to the Hessian (4.59),  $\nabla^2 f(\theta_k) \approx J(\theta_k)^{\top} J(\theta_k)$ , ignoring the last term. This is called the Gauss-Newton method. The search direction is then found solving the resulting Newton equations, using (4.58), for  $p_k^{GN}$ :

$$\nabla^2 f(\theta_k) p_k = -\nabla f(\theta_k) \quad \Rightarrow \quad J(\theta_k)^{\top} J(\theta_k) p_k^{GN} = -J(\theta_k)^{\top} r(\theta)$$

Solving for  $p_k^{GN}$  numerically, directly using the inverse of  $J(\theta_k)^\top J(\theta_k)$ , is not generally recommended because of round-off errors. If the Jacobian  $J(\theta_k)$  is rank-deficient, or close to rank-deficient, the Levenberg-Marquardt method is recommended. The Optimization Toolbox in MATLAB provides a function `lsqnonlin` for the nonlinear least squares problem that can be set to use either method.

`lsqnonlin` also implements a large-scale algorithm, stated to be using a subspace trust region method based on an interior-reflective Newton method. This method can use numerical approximations to the Jacobians. Using the large-scale algorithm provided a better fit than the other methods when the dataset was large. The differences in parameters found for smaller datasets were very small, method-to-method. The large-scale algorithm was selected for finding the various model parameters, as seen in Chapter 6.

### 4.6.1 Jacobians for resistance models

The Jacobian of (4.56) with  $\theta_{S_1} = (a, b, c)^\top$ , using the model (2.21), is given by:

$$J_1(\theta_{S_1}) = \begin{bmatrix} e^{bx_j} & ax_j e^{bx_j} & (1, \dots, 1)_{1 \times m}^\top \end{bmatrix}_{m \times 3},$$

The Jacobian of the model (2.22) with  $\theta_{S_2} = (\alpha, \beta, \gamma)^\top$  is given by:

$$J_2(\theta_{S_2}) = \begin{bmatrix} -\frac{(\beta + \gamma x_j)x_j}{(1 + \alpha x_j)^2} & \frac{1}{1 + \alpha x_j} & \frac{x_j}{1 + \alpha x_j} \end{bmatrix}_{m \times 3},$$

### 4.6.2 Implementation

The Gauss-Newton method was implemented as a MATLAB script. MATLAB functions intended for use with `lsqnonlin` were written to implement the two models presented in Section 2.3.1 (as well as other models, which performed rather badly.) The scripts used are described in Appendix B.1.4.

# Chapter 5

## Experiments

### 5.1 Laboratory setup

#### 5.1.1 Instrumentation

The instrumentation setup used in the experiments is shown in Figure 5.1. A computer was equipped with a data acquisition (DAQ) card, with possibilities for digital to analog (D/A) conversion, analog to digital (A/D) conversion and decoding of position encoders. The voltage actuation signal was generated by software and outputted to the D/A converter before it was amplified using a linear amplifier. The amplified signal was connected to the motor. A Hall effect current transducer was connected in series with the motor. The output signal from the transducer was then filtered through a low-pass Butterworth filter with Sallen-Key topology, before being inputted to the A/D converter. The position was measured using an encoder, which was connected to the DAQ card's decoder. A type K (Chromel-Alumel) thermocouple was fixed on the motor's housing, and the signal was amplified by a custom thermocouple amplifier before being inputted to the A/D converter. The parts used for the measurement instrumentation are shown in Table 5.1.

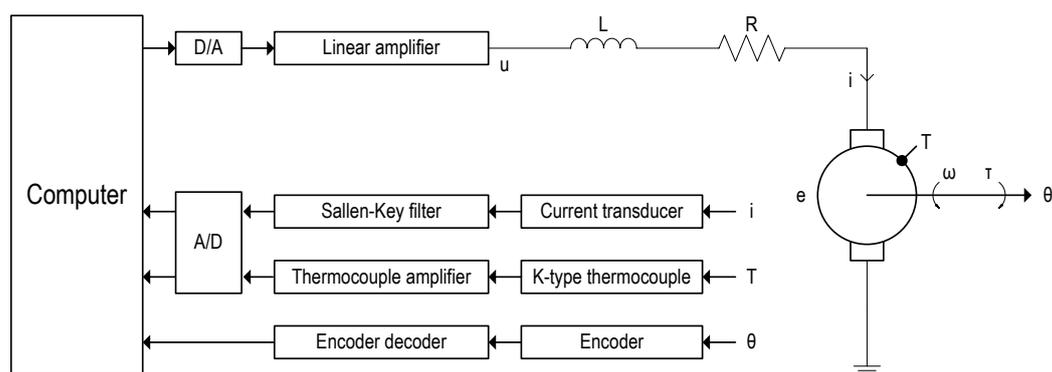


Figure 5.1: Laboratory setup.

Table 5.1: Parts list

| Manufacturer      | Part No. | Description                    |
|-------------------|----------|--------------------------------|
| LEM               | HX 05-NP | Hall effect current transducer |
| N/A               | N/A      | Position encoder               |
| Analog Devices    | AD595AQ  | Thermocouple amplifier         |
| Texas Instruments | TL082CP  | Operational amplifier          |

### 5.1.2 Experimental rig

An experimental rig was built to be able to apply load to motors in a convenient manner. It is shown in Figure 5.2(a). It consists of a wooden vertical plate mounted on a wooden horizontal base. Motors are mounted in a rack made out of a plastic plate on top of the vertical plate, shown in Figure 5.2(b).

Spindles with an attached thread provide the ability to add weights or a spring as load, and run these up and down, coiling the thread around the spindles. A hole was drilled in the base to be able to run weights for a longer distance than would otherwise be possible.

A tray was used to hold several weights at the same time, yielding more resolution in the load. The tray was weighed using a scale, and the mass was found to be  $m_{tray} = 0.087$  [kg].

Three different sizes of spindle were used. Measuring their geometries and finding their masses, the moment of inertia for each spindle was found using

$$J_z = \frac{1}{2}m(r_1^2 + r_2^2),$$

which is the formula for a hollow cylinder with inner radius  $r_1$ , outer radius  $r_2$ , and mass  $m$ , about the symmetry axis  $z$ . The details are in the MATLAB script described in Appendix B.1.2. The values found are listed in Table 5.2. Each spindle had an outer radius of  $r_2 = r_{spindle} = 7.5 \times 10^{-3}$  [m].

Table 5.2: Moment of inertia for each spindle.

|         |                       |                      |
|---------|-----------------------|----------------------|
| $J_z^s$ | $0.85 \times 10^{-6}$ | [kg m <sup>2</sup> ] |
| $J_z^m$ | $1.1 \times 10^{-6}$  | [kg m <sup>2</sup> ] |
| $J_z^l$ | $1.6 \times 10^{-6}$  | [kg m <sup>2</sup> ] |



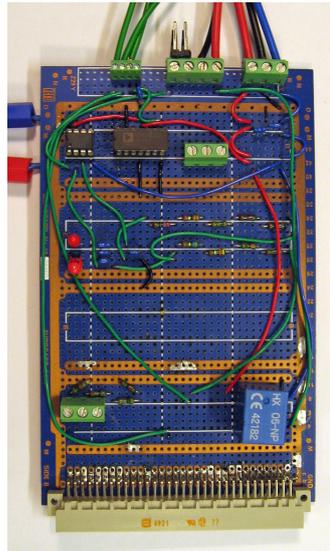
(a) Experimental rig.



(b) Motor mounting.



(c) Power supply, linear amplifier and DAQ interface.



(d) Instrumentation.

Figure 5.2: Laboratory setup.

### 5.1.3 Computer setup

The hardware consisted of a workstation with a DAQ card, and a combined power supply and linear amplifier. The hardware components are listed in Table 5.3.

The software used was MATLAB with Simulink and Real-Time Workshop, together with WinCon, which is software that aids with rapid prototyping and hardware-in-the-loop simulations. This ran on a workstation with the Windows operating system and VenturCom RTX, which is an extension to Windows that makes it more suitable for real-time applications. The various software components are listed in Table 5.4.

Table 5.3: Hardware

|  |
|--|
| Workstation:<br>Intel Pentium 4 CPU @2.80 GHz, 512 MB RAM (DataRespons Norge AS) |
| Data acquisition card:<br>MultiQ-3 (Quanser Consulting, Inc.)                    |
| Power supply and linear amplifier:<br>UPM-2405 (Quanser Consulting, Inc.)        |

Table 5.4: Software

|                                 |                   |                          |
|---------------------------------|-------------------|--------------------------|
| MATLAB                          | 6.1.0.450 (R12.1) | MathWorks, Inc.          |
| Simulink and Real-Time Workshop | 4.1 (R12.1)       | MathWorks, Inc.          |
| WinCon Server and Client        | 3.3               | Quanser Consulting, Inc. |
| WinCon Toolbox                  | 3.2.0             | Quanser Consulting, Inc. |
| Quanser Toolbox                 | 1.2.0             | Quanser Consulting, Inc. |
| Windows XP Professional         | SP1               | Microsoft Corp.          |
| VenturCom RTX                   | 5.1               | VenturCom                |

### 5.1.4 Position measurement and velocity estimate

Encoders were used to measure the angular positions of the motor rotors. The encoders that were used give out a quantized signal with 2000 counts per revolution. The quantization interval for these is therefore  $\Delta_\theta = \frac{2\pi}{2000}$  [rad/count]. The quantized signal is sampled after being decoded.

The error of the position measurement is within  $e_\theta \in (-\frac{\Delta_\theta}{2}, \frac{\Delta_\theta}{2})$ . If the quantizer noise is assumed to be uniformly distributed, the variance of the quantization noise should be given by

$$\sigma_e^2 = \text{E}[e^2] = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e^2 de = \frac{\Delta^2}{12}$$

yielding the variance of the noise on the position measurement as

$$\sigma_\theta^2 = \left( \frac{2\pi}{2000} \right)^2 \frac{1}{12}. \quad (5.1)$$

An estimate of the angular velocity is needed in the adaptive control law and the gradient method estimator. This is found using the first difference

$$\omega_k = \frac{\theta_k - \theta_{k-1}}{T},$$

where  $T$  is the sampling time used. The velocity estimate will then have a quantization interval of  $\Delta_\omega = \Delta_\theta f_s$ , where  $f_s = \frac{1}{T}$  is the sampling frequency. The velocity error will be within  $e_\omega \in (-\frac{\Delta_\omega}{2}, \frac{\Delta_\omega}{2})$ , yielding the variance of the noise on the velocity estimate as

$$\sigma_\omega^2 = \left( \frac{2\pi}{2000} \right)^2 \frac{1}{12} f_s^2. \quad (5.2)$$

The noise will increase with increasing sampling frequency.

### 5.1.5 Current measurement

A Hall effect current transducer was used to measure the current. This was put in series with the connection to the motor, and outputted a voltage,  $V_{out}$ , suitable for the A/D converter. The configuration of the transducer yielded a transfer function for the measured size as  $i = 5/4V_{out}$ . The output of the current transducer was filtered by a Sallen-Key filter before being read by the A/D converter. The filter is equivalent to the filter present on the SCU. Some output bias was experienced during the use of the transducer. The bias exhibited a hysteresis effect, and was to some degree proportional to the magnitude of the motor current. Higher motor current yielded a larger bias with the same polarity as the motor current. The bias was never observed to be outside of the transducer's specified limits,  $\pm 40$  [mV], or  $\pm 50$  [mA].

The Sallen-Key filter is shown in Figure 5.3, [15]. It is an active filter, producing a two-pole low-pass response using two resistors, two capacitors and a unity-gain buffer amplifier. It has a transfer function given by

$$G(s) = \frac{1}{R_1 R_2 C_1 C_2 s^2 + C_2 (R_1 + R_2) s + 1}. \quad (5.3)$$

To make this filter equivalent to a Butterworth filter, one can compare the coefficients in Equation (2.18) with those in (5.3) to compute the values for the resistors and capacitors. The two relations become:

$$\frac{1}{\omega_c^2} = R_1 R_2 C_1 C_2, \quad \frac{\sqrt{2}}{\omega_c} = C_2 (R_1 + R_2)$$

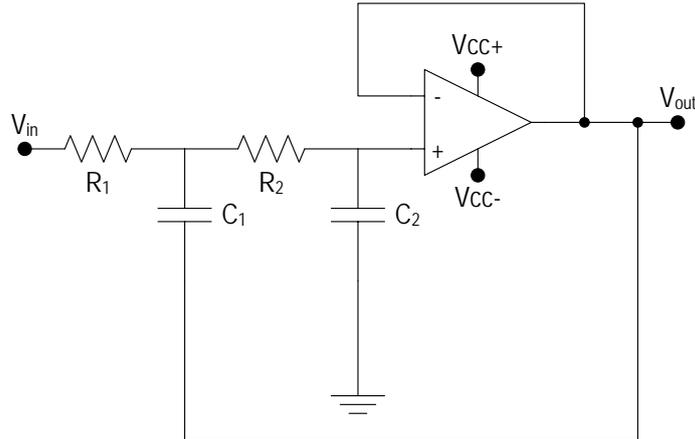


Figure 5.3: Sallen-Key filter.

The Sallen-Key filter was set up with  $R_1 = 8.06$  [k $\Omega$ ],  $R_2 = 95.3$  [k $\Omega$ ],  $C_1 = 0.15$  [ $\mu$ F] and  $C_2 = 0.022$  [ $\mu$ F], the same as the filter present on the SCU. This should give a cut-off frequency of

$$f_c = \frac{\omega_c}{2\pi} = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \approx 100 \text{ [Hz]}.$$

The operational amplifier listed in Table 5.1 was used as the buffer amplifier.

Because of commutation, the noise in the current measurement tends to grow with angular velocity. Figure 5.4 presents a graph of the standard deviation of the current measurement found while running the motor at various speeds. For each of the points produced, the motor was at some steady-state operating point, so the standard deviation found should be representative of the noise. A linear fit was found, and the noise variance was hence found and updated according to the following expression:

$$\sigma_{i_f}^2(\omega) = (0.00025|\omega| + 0.0020)^2 \quad (5.4)$$

### 5.1.6 Thermocouple

Thermocouples are a type of temperature sensor that utilise the Seebeck effect, which dictates that when two dissimilar conductors forming a loop are subjected to a temperature gradient, an electromotive force,  $e$ , is produced. This electromotive force can be measured and will give a relationship between the measured voltage and the temperature gradient. This relationship is typically nonlinear, and can be approximated by a polynomial:

$$\Delta T(e) = \sum_{n=0}^N a_n e^n \quad (5.5)$$

If the coefficients,  $a_n$ , and the temperature at one end of the loop are known, the temperature at the other end can be calculated. The coefficients are standardized for

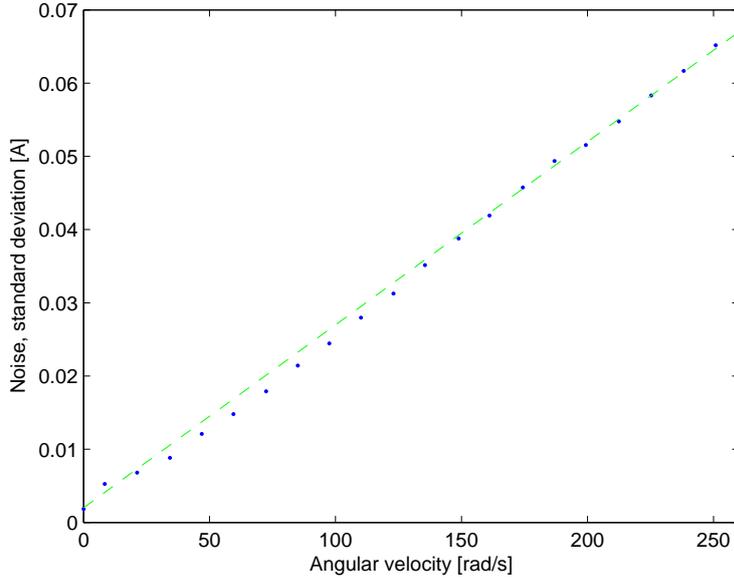


Figure 5.4: Current measurement noise as a function of angular velocity.

various combinations of conductors. A type K thermocouple, consisting of conductors made of the metal alloys Chromel (Nickel-Chromium) and Alumel (Nickel-Aluminium), was used during the experiments. The coefficients needed for this type of thermocouple were obtained from [6], and are shown in Table 5.5. They are valid on a domain of  $e \in (0.000, 20.644)$  [mV], yielding a range of  $T \in (0, 500)$  [°C]. The error, if the voltage measurement is correct, should be within  $(-0.05, 0.04)$  [°C].

The thermocouple voltage was amplified by a custom thermocouple amplifier, listed in Table 5.1. The output voltage,  $V_{out}$ , of this amplifier was then converted back to the thermocouple voltage using the following expression:

$$e = \frac{1}{247.3} V_{out} - 11 \times 10^{-6} \quad (5.6)$$

The thermocouple amplifier had built-in cold junction compensation, so the thermocouple voltage should be referenced to 0 [°C].

In practice, the above ideal conditions did not apply. The measurements were very noisy, and deviated from what would be expected in a few ways.

There was a coupling from the motor current, causing some voltage to be superimposed onto the signal from the AD595 measurement amplifier. This was not a magnetic coupling, since a steady-state current induced the same amount of voltage as a time-varying current. The source of this disturbance was not found, but the search suggested that the disturbance might have been picked up after the terminals connecting the voltage output signals to the A/D converter. The voltage was compensated for in software, where the following relation would yield the same mean value for the measurement before

Table 5.5: Type K thermocouple  $e \rightarrow T$  coefficients,  $e \in (0.000, 20.644)$  [mV].

|       |                            |
|-------|----------------------------|
| $a_0$ | $0.000000 \times 10^0$     |
| $a_1$ | $2.508355 \times 10^1$     |
| $a_2$ | $7.860106 \times 10^{-2}$  |
| $a_3$ | $-2.503131 \times 10^{-1}$ |
| $a_4$ | $8.315270 \times 10^{-2}$  |
| $a_5$ | $-1.228034 \times 10^{-2}$ |
| $a_6$ | $9.804036 \times 10^{-4}$  |
| $a_7$ | $-4.413030 \times 10^{-5}$ |
| $a_8$ | $1.057734 \times 10^{-6}$  |
| $a_9$ | $-1.052755 \times 10^{-8}$ |

and after some voltage with resulting current  $i$ , would be applied to the motor:

$$V'_{out} = V_{out} - 0.0032i$$

The noise from the measurement amplifier was very low when the motor was quiescent. The noise increased with increasing current and velocity. The highest variance found was about  $\sigma_{T_h}^2 \approx 1.0$ .

Figure 5.5 shows the measurements obtained from the AD595 measurement amplifier vs. measurements made with a Fluke 54II thermometer. The Fluke thermometer was used as a reference, and the temperature calculated using the measurement amplifier was then calibrated using the linear fit

$$T_{calibrated} = 1.35T_{AD595} - 9.75,$$

also shown in Figure 5.5.

The reason the Fluke 54II thermometer was not used for all of the experiments was that it could only hold 500 datapoints, had to be started and stopped manually, and transferring the collected data was cumbersome, making it impossible to automate the gathering of experimental data.

## 5.2 Implementation

The estimators were implemented in Simulink, and built for the WinCon target using Real-Time Workshop. The extended Kalman filter was implemented as a Simulink S-Function in C, the gradient method/least mean squares filter was implemented as a regular block diagram, and the Kalman filter for the thermal model was implemented using MATLAB's `kalman` function and added as a discrete state-space block. The resulting Simulink diagram is shown in Figure 5.6.

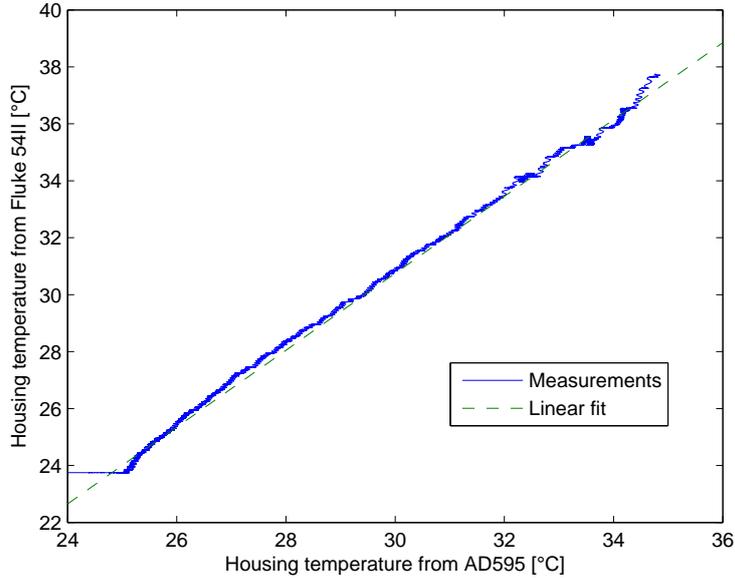


Figure 5.5: Temperature measurement calibration.

### 5.2.1 Position control

A proportional angular position controller, shown in Figure D.5, was used to have control over how to move the attached weights. A gain of  $K_P = 10$  was used, as lower values yielded an unpractical steady-state error for heavy weights. A trajectory generator was used to generate smooth references for positioning the weight vertically. The MATLAB script is described in Appendix B.1.2.

A sinusoidal position reference was also used. This was added to the generated trajectory at some specified instant in time, and ramped up to the desired amplitude. For reference, we will list the function generating the angular position reference, and the resulting angular velocity, angular acceleration, and angular jerk.

$$\begin{aligned}
 \theta'_r &= A_r \sin(\omega_r t) \\
 \omega'_r &= A_r \omega_r \cos(\omega_r t) \\
 \alpha'_r &= -A_r \omega_r^2 \sin(\omega_r t) \\
 \varphi'_r &= -A_r \omega_r^3 \cos(\omega_r t)
 \end{aligned}$$

For all of the experiments with weights,  $A_r = 3\pi$  was used to ensure that all of the coils in the winding would be excited. It was found that using an angular frequency of  $\omega_r = 3\pi$  was practical. For higher angular frequency, the vertical speed of the weights could not keep up with the tangential speed of the spindle, and they would be suspended in air for about half a period. Other frequencies that were tried were close to some resonant mode of the system of thread and tray (effectively a double pendulum), and

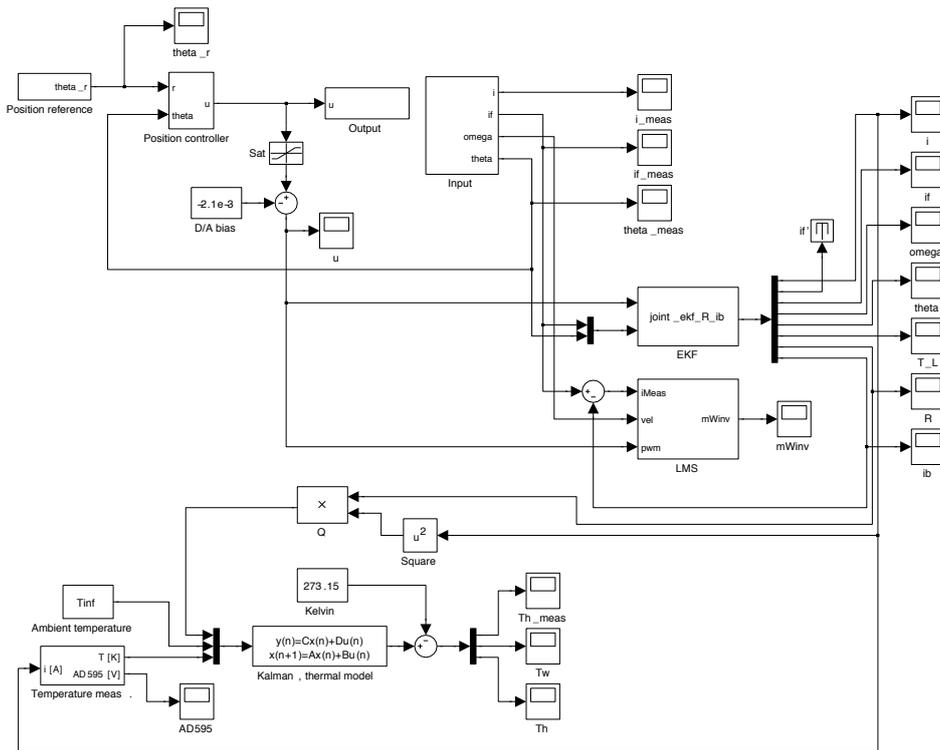


Figure 5.6: Simulink implementation.

would make it oscillate excessively.

### 5.2.2 Extended Kalman filter

The extended Kalman filter from Section 4.5.3 was implemented in C. C was used so that the source code would be portable to other platforms, and because it was the only supported format for implementing S-Functions in the version of Simulink/Real-Time Workshop used. Functions to do basic matrix operations like addition and multiplication were written. The Kalman filter requires a matrix inversion, so a numerically reliable method was needed to do this. A Gauss-Jordan elimination algorithm was implemented for this purpose. It was compared to the implementation of the same algorithm from [19]. The implementation from [19] was more general, well-tested, and simpler to use, so this was the one used in the implementation of the filter. Gauss-Jordan elimination was used because the algorithm is fairly easy to understand and implement, and yields good numerical precision if the matrix to be inverted is well-conditioned, which always seemed to be the case for this particular filter.

Because the full state-space description of the motor was used, it was necessary to model the torque load disturbance,  $T_L$ , as an unknown exogenous input. Because of the time-varying property of the bias in the current measurement,  $i_b$ , estimation of this

bias was added of this as well. This gave us seven states to estimate. To estimate model parameters, we thus needed to estimate at least eight states, meaning that we might be close to a marginally observable system. In practice, the filter performed well, with no diverging states, when the process noise variances were properly tuned.

The output signal for the D/A converter was used as the voltage input, and the filtered current and the position encoder signal were used as measurements. Using the noisy first difference estimate of the angular velocity as a measurement, in the hopes of improving observability, did not seem to have any noticeable effect on the various estimates. It was therefore not used, simplifying the filter.

We chose to estimate only the motor’s terminal resistance,  $R$ , for the experiments. There was no reason to believe that the other motor parameters would change significantly while running the motor. The inductance and the moment of inertia for the rotor and spindle could confidently be assumed to be constant. The motor’s electromagnetic constant was estimated with the extended Kalman filter (in addition to the terminal resistance), using a sinusoidal position reference,  $A_r = 3\pi$ ,  $\omega_r = 3\pi$ , while the motor was loaded with a weight of  $m = 0.500$  [kg]. The time-series of the estimated value is shown in Figure 5.7. As can be seen, the estimated values are a little higher than what the manufacturer has stated (Table 2.1.) Other loads were also tried, with similar results. The value provided by the manufacturer was used.

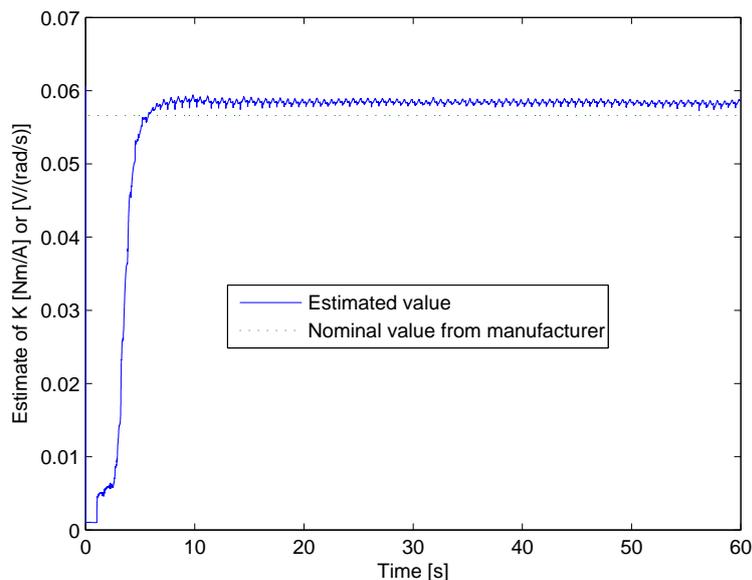


Figure 5.7: Estimated electromagnetic constant,  $K = K_T = K_E$ .

The minimum sampling time that could reliably be used with WinCon was  $T = 5 \times 10^{-4}$ . This was the value used for most of the experiments, with the hopes of increasing accuracy. For the comparisons of the adaptive schemes, the same sampling

time as the SCU,  $T' = 1 \times 10^{-3}$ , was used.

For the measurement noise covariance matrix  $R$ , the value for position noise variance found in (5.1) was used, and for the filtered current, we used the expression found in (5.4) to update the variance using the estimated angular velocity.

$$R_k = \begin{bmatrix} \sigma_{i_f}^2(\omega_k) & 0 \\ 0 & \sigma_{\theta_k}^2 \end{bmatrix}$$

For deterministic processes, the elements of the process noise covariance matrix,  $Q$ , are typically set to zero. However, to ensure numerical stability of the error covariance matrix,  $P$ , [4] recommends using small, positive values in the major-diagonal elements of the process noise covariance matrix. Otherwise, the error covariance matrix might become non-positive-definite when the number of steps become large, which will lead to divergence problems. The diagonal elements of  $Q_k^x$  were set to  $10^{-21}$ .

Taking into consideration (4.35), we could get some idea of the variation of the torque load as follows: taking the time derivative of (2.10), ignoring the current dynamics, assuming the torque load would otherwise be constant, and solving for  $\dot{T}_L$ , yields:

$$\dot{T}_L = K_T \frac{di}{dt} - J\ddot{\omega} \approx -J\varphi$$

Using Euler integration,

$$\dot{x} = au \quad \Rightarrow \quad x_{k+1} = x_k + \Delta t au_k \quad \Rightarrow \quad \frac{\Delta x}{\Delta t} = au_k.$$

One would expect the maximum variance of this quantity to be

$$\sigma_{T_L}^2 \approx \frac{\Delta T_L^2}{\Delta t} = (-J_{max} \varphi_{max})^2 T \approx 4.4 \times 10^{-6},$$

using  $T = 5 \times 10^{-4}$ ,  $\varphi_{max} = A_r \omega_r^3$  (from Section 5.2.1), and  $J_{max} = J + J_z^l$  (from Tables 2.1 and 5.2.) Since the maximum jerk will only occur briefly at the inflection points of the acceleration, and because of friction and noise effects, the variance should be somewhat lower than this. A value of

$$\sigma_{T_L}^2 = 1 \times 10^{-6}$$

was chosen when we were interested in tracking the load as fast as possible. The estimates were typically very close to  $T_L \approx r_{spindle} m_{weight} g$ , using  $g = 9.81$  [m/s<sup>2</sup>]. One would expect friction and rotor/spindle acceleration to account for any discrepancy.

Trial and error were used to find variances for the current measurement bias and the resistance that yielded estimates for these quantities which seemed reasonable. The current measurement bias was expected to vary slowly, so a very small variance should be used. Estimates for both bias and resistance diverged if the process variance for the measurement bias was set too low. Using

$$\sigma_{i_b}^2 = 1 \times 10^{-15},$$

at  $T = 5 \times 10^{-4}$ , gave an estimate that corresponded well to multimeter measurements (the current transducer's output voltage was measured when the motor was quiescent.)

The quantity with the most variability was the resistance. When estimating with the purpose of finding a dataset for fitting a model with nonlinear regression, it was preferable to find a value close to the optimal mean and averaging away noise than to have fast convergence. When using a sampling time of  $T = 5 \times 10^{-4}$ , the resistance's variance was selected to be

$$\sigma_R^2 = 1 \times 10^{-7}.$$

The variance for the torque estimate was also reduced for these experiments, and a value of

$$\sigma_{T_L}^2 = 1 \times 10^{-9}$$

was chosen. When interested in tracking the terminal resistance as quickly as possible, a variance of

$$\sigma_R^2 = 1$$

was used. A higher value did not seem to result in faster tracking, just noisier estimates.

The resulting process covariance matrix, using  $Q_k^d = \sigma_{T_L}^2$ ,  $Q_k^\theta = \sigma_R^2$ , and  $Q_k^b = \sigma_{i_b}^2$  became:

$$Q_k = \begin{bmatrix} Q_k^x & 0 & 0 & 0 \\ 0 & Q_k^d & 0 & 0 \\ 0 & 0 & Q_k^\theta & 0 \\ 0 & 0 & 0 & Q_k^b \end{bmatrix}$$

The different variances were scaled if other sampling times were used. Changing from  $T = 5 \times 10^{-4}$  to  $T' = 1 \times 10^{-3}$ ,  $Q_k$  was multiplied with  $T'/T = 2$ .

### 5.2.3 Kalman filter for thermal model

For the steady-state Kalman filter used for the thermal model, the measurement noise variance found in Section 5.1.6 was used,  $R = \sigma_{T_h}^2 = 1.0$ . This is a rather high value, but it did not seem to have any adverse effects on the filter's ability to track the housing temperature, since the thermal dynamic for the motor housing is very slow. The process noise covariance matrix was set to

$$Q_k = \begin{bmatrix} 0.10 & 0 \\ 0 & 0.25 \end{bmatrix},$$

so as to trust the model for the winding temperature more than the model for the housing temperature, and each of those more than the measurement.

### 5.2.4 Extended Kalman filter behavior

Figure 5.8 presents plots of the voltage input and various states for the EKF when loaded with a spring and given a sinusoidal position reference. It should be noted that the resistance estimate tends to be higher when the magnitude of the current is low,

and that the torque load estimate is higher when the spring is more stretched (at higher values of angular position.) This experiment also utilized a different motor than was used in all other experiments. The resistance estimates might not be directly comparable.

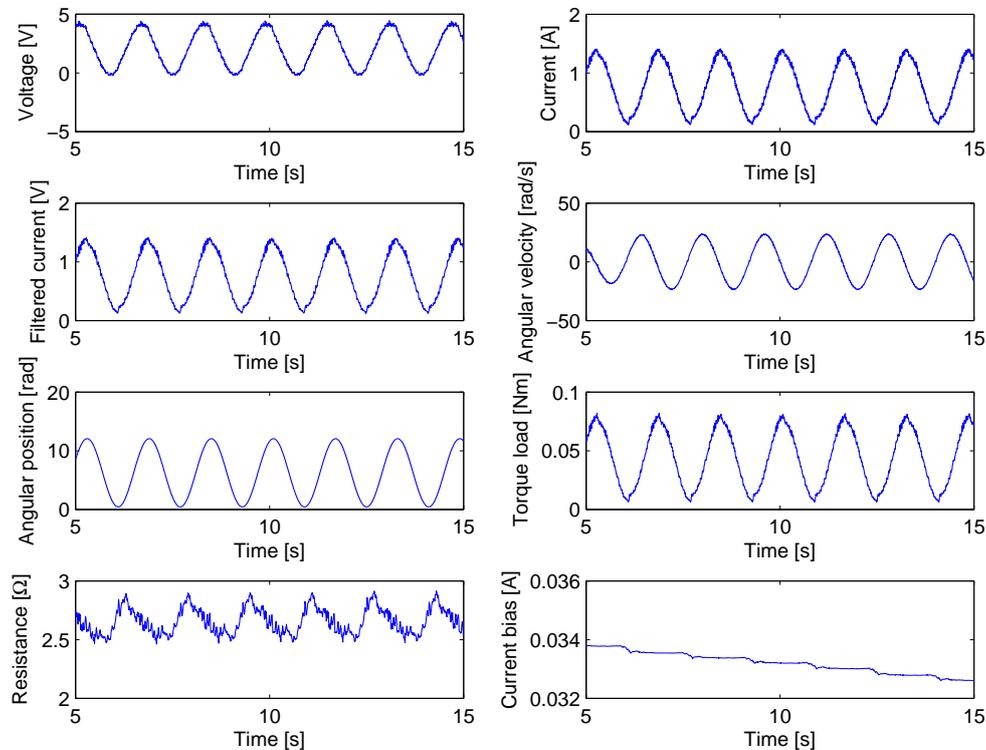


Figure 5.8: Kalman filter input and states, motor loaded with a spring.

### Bias rejection

Figure 5.9 compares the measured, filtered current and the corresponding Kalman filter estimate. This dataset was collected while running the motor against its friction only, with a sinusoidal position reference of  $A_r = 3\pi$  and  $\omega_r = 3\pi$ . Under these conditions, one should expect the friction to produce about the same load in both the positive and the negative direction, and therefore the current to have a mean value of zero. The arithmetic mean of the measured, filtered current was  $\langle i_f \rangle = 0.028$ , and the corresponding estimate had an arithmetic mean of  $\langle \hat{i}_f \rangle = -0.0019$ . The arithmetic mean of the estimated bias was  $\langle \hat{i}_b \rangle = 0.030$ , indicating that the filter can track and reject measurement bias.

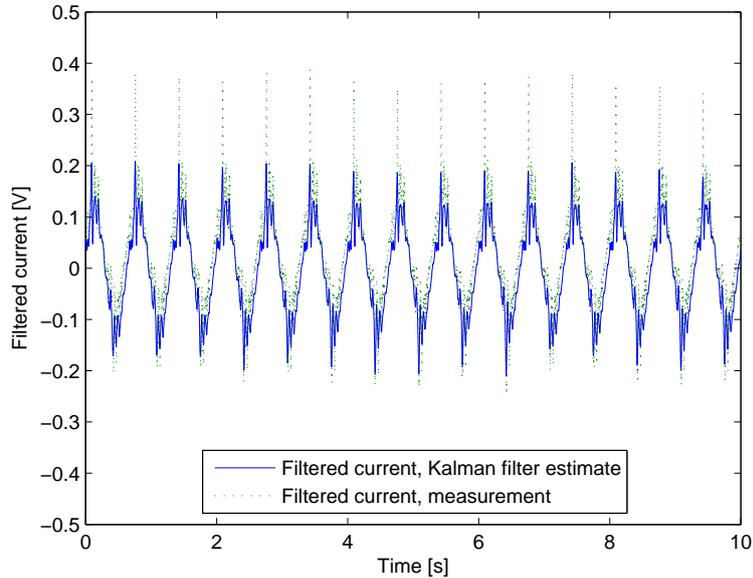


Figure 5.9: Filtered current measurement compared to the Kalman filter estimate.

## 5.3 Performed experiments

### 5.3.1 Winding resistance and its temperature dependency

#### Rationale

This experiment was performed in order to achieve three things: firstly, it was desired to check the extended Kalman filter’s ability to estimate the resistance, compared with the model for temperature dependent resistance in (2.23).

Secondly, it was necessary to calibrate the measurements from the AD595 thermocouple measurement amplifier. A Fluke 54II thermometer was used as a reference (the results of this calibration are presented in Section 5.1.6.) Also, using the measurements from the Fluke 54II thermometer when estimating the winding temperature should provide more accurate estimates.

Thirdly, an estimate of the lower bound on the terminal resistance was desired. From what we know about the contact resistance, it should be expected that at high currents, it will be very small and have little variability because of its asymptotic behavior. What we are left with then, is the winding resistance, in addition to a small, almost constant, contact resistance. This should form a lower bound for the terminal resistance. It provides a sanity check for any adaptive scheme for estimating resistance: the actual resistance in the motor will never be lower than this value.

Using the lower bound as an estimate of the winding resistance at a reference temperature,  $R_{w0}$ , in addition to the estimated winding temperature, we should find the

approximate proportion of the estimated terminal resistance due to the winding. Applying the inverse of (2.23) to the approximate proportion and referencing to a given temperature, we can remove the temperature dependency of the winding resistance in the terminal resistance estimates.

## Procedure

At higher currents, the Joule losses are also high. This results in rising winding temperature and higher winding resistance, as predicted by (2.23). The motor was therefore run with a high current load. It was loaded with a weight of  $m = 1.495$  [kg], and run up and down with a sinusoidal position reference, with  $A_r = 3\pi$  and  $\omega_r = 3\pi$ , for about 5 minutes. The root mean square (RMS) value of the current was  $i_{rms} \approx 2.3$ . WinCon was set to use a sampling time of  $T = 5 \times 10^{-4}$  [s] during the course of this experiment.

In addition to the estimates from the extended Kalman filter and the Kalman filter for the thermal model, the housing and ambient temperature were recorded with a Fluke 54II thermometer, with a sampling time of 1 [s].

## Additional data processing

The data recorded with the Fluke 54II thermometer were used to reconstruct an estimate of the winding temperature in Simulink, using the Kalman filter for the thermal model. The diagram is shown in Figure D.8.

### 5.3.2 Contact resistance and its current dependency

#### Rationale

The intent of this experiment was to identify the relationship between the terminal resistance and the armature current. The chosen approach was to run the motor with different loads, producing different armature currents and corresponding terminal resistances. Collecting a large dataset of estimates for the armature current and terminal resistance should provide a good basis for finding a fit to the models in Section 2.3.1, using nonlinear regression.

A sinusoidal position reference, as described in Section 5.2.1, was used when estimating with the purpose of finding a dataset for fitting a model with nonlinear regression. This was because of the limited vertical distance we could run with weights attached. This required a rather large process noise variance for the resistance to ensure fast convergence, so we feared that transients and noise might provide poorer estimates than what could be achieved by using average values (the arithmetic mean of the Joule losses and the root mean square of armature current were used to find average values for the armature current and terminal resistance.) It also proved to be more practical and time-efficient, since the motor could be left running for a long period of time, instead of recording many short time-series.

## Procedure

The motor was loaded in two different ways. To get estimates at low currents, the motor was run against its own friction and inertia. To produce higher currents, the motor was loaded with weights.

Running against friction, either a fixed amplitude of  $A_r = 3\pi$  was used and the frequency was varied, or the motor was run with a fixed frequency of  $\omega_r = 3\pi$  and the amplitude was varied. This way we control the load because viscous friction will increase with increasing velocity, and the inertia will contribute to higher current magnitudes with increasing acceleration. The details are given in Appendix C.

Since nothing needed to be done with the experimental rig between each run when loading the motor using it's own friction, the gathering of data was automated and typically, the system was run overnight.

When loading the motor with different weights, the motor was run using combinations of weights to obtain the following set of masses:

$$m_{weight} \in \{0.050, 0.100, 0.150, 0.200, 0.250, 0.300, 0.350, 0.400, 0.450, 0.500, \dots \\ 0.600, 0.700, 0.800, 0.900, 1.000, 1.495\}$$

The system was set to run using the sinusoidal position reference with  $A_r = 3\pi$  and  $\omega_r = 3\pi$ , after lifting the weight off the floor using a generated trajectory reference. Details are given in Appendix C.

Before running the system to collect a set of time series, the initial housing temperature and ambient temperature were measured using the Fluke 54II thermometer. The ambient temperature was assumed to be close to constant for the duration of the run.

WinCon was set to use a sampling time of  $T = 5 \times 10^{-4}$  [s].

## Additional data processing

The estimate of the terminal resistance,  $R$ , should be a combination of the contact resistance and the winding resistance:

$$R = R_t = R_c(i) + R_w(T_w)$$

To remove the temperature dependency of the winding resistance, we can start by writing Equation (2.23) as:

$$R_{w0} = R_w - \alpha(T_w - T_{w0})R_{w0} \quad (5.7)$$

What we would like to obtain is the terminal resistance referencing the winding resistance to a given temperature:

$$R_0 = R_{t0} = R_c + R_{w0} \quad (5.8)$$

Inserting (5.7) into (5.8) we get

$$R_0 = R_{t0} = R_c + R_w - \alpha(T_w - T_{w0})R_{w0} = R_t - \alpha(T_w - T_{w0})R_{w0}.$$

If we have an estimate of  $R_{w0}$ , the temperature dependency of the winding resistance can be removed by subtracting  $\alpha(T_w - T_{w0})R_{w0}$  from the estimated terminal resistance.

The average power dissipated by the terminal resistance when the current and resistance are time-varying should be

$$\langle Q_J \rangle = \langle R(i)i^2 \rangle,$$

while the average power for time-varying current and a constant resistance is

$$\langle Q_J \rangle = \langle \bar{R}i^2 \rangle = \bar{R}\langle i^2 \rangle = \bar{R}i_{rms}^2.$$

The equivalent constant terminal resistance given time-series of the estimated armature current and terminal resistance should then be:

$$\bar{R} = \frac{\langle R(i)i^2 \rangle}{i_{rms}^2}$$

The set of numbers constituting the RMS value of the armature current and the equivalent constant terminal resistance is what was used to find a fit for the resistance models in Section 2.3.1, using nonlinear regression.

For reference, the arithmetic mean of a set of numbers is defined as

$$\langle x \rangle \triangleq \frac{1}{n} \sum_{i=1}^n x_i,$$

and the root mean square of a set of numbers is defined as

$$x_{rms} \triangleq \sqrt{\langle x^2 \rangle} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}.$$

### 5.3.3 Comparison of adaptive schemes

#### Rationale

It was desirable to get an indication of how well the different adaptive schemes performed. To this end, we compared the measured current with current reference produced by the PD regulator, (3.1), while using the estimated resistance from the different adaptive schemes and the adaptive control law from Equation (3.2). The error should ideally be zero, so the RMS value of the time-series of the error should be a measure of the performance. Additionally a test was conducted using a PI regulator for the current.

#### PD regulator and adaptive control law setup

For this experiment, the control structure shown in Figure 3.1 was implemented. This control structure was already implemented in the Simulink simulator of the system developed in the preceding project during the Fall of 2006. It was copied and re-implemented for use with WinCon.

The adaptive control law/feedforward controller from Equation (3.2) was used to control the motor's armature current. The resistance in this equation is updated with the most recent resistance estimate,  $R_k$ , and the velocity estimate,  $\omega_k$ , when running. The resulting control signal is therefore given as:

$$u_k = R_k i_{rk} + K_E \omega_k \quad (5.9)$$

We also added the one-sample time delay that is present in the SCU implementation. The corresponding Simulink diagram is shown in Figure D.10.

The PD regulator gains, from Equation (3.1), were chosen as specified in Equations (3.6) and (3.7), that is

$$\begin{aligned} K_p &= \omega_0^2 J \\ K_d &= 2\zeta \omega_0 J \end{aligned}$$

where  $\omega_0$  (the bandwidth) was set to  $\omega_0 = 2\pi \cdot 27$ , and  $\zeta$  (the damping) was set to  $\zeta = \sqrt{2}$ . The value of the moment of inertia  $J$  was the sum of the rotor inertia from Table 2.1 and the inertia for the chosen spindle, from Table 5.2. When using the EKF, its angular position and velocity estimates were used as the feedback to the PD regulator. The Simulink implementation is shown in Figure D.10.

We also ran a test with a PI regulator for the armature current. The expression from (3.45) was implemented, using a proportional gain of  $K_P = 5$ , and an integral gain of  $K_I = 100$ . The Simulink diagram is shown in Figure D.11.

### LMS parameter estimator setup

Figure D.1 shows the block diagrams of the parameter estimator from Equation (4.5). The Butterworth filter present after the summation of the induced voltage and the output voltage is added so as to match the Butterworth filter on the current measurement in the top level diagram.

The scaling constant  $\mu = h\gamma$  (see (4.5)) was chosen to be the same as in the SCU implementation,  $\mu = 0.02$ . The initial value of the estimated conductance was set to  $G_0 = 1/2.74$ , the reciprocal of the nominal value from Table 2.1.

### Gain-scheduler setup

As will be shown in Section 6.2, the model that best fit the experimental data, was the rational model from Equation (2.22). The obtained parameter values were used as presented in Table 6.1. The computed resistance inputted to the adaptive control law in Equation (5.9) was therefore:

$$R_k = \frac{\beta + \gamma|i_k|}{1 + \alpha|i_k|} \quad (5.10)$$

Equation (5.10) is thus the gain-scheduler (GS), and its Simulink diagram is shown in Figure D.13. The absolute value of the current was used, since we wanted the resistance attenuation to be symmetric and bounded.

As will be shown in Section 6.3, the above computation did not immediately yield a very good result. It was therefore attempted to adapt one of the parameters in (5.10) in the hopes of adjusting the shape of the model somewhat, to give a better fit. The shape is controlled mostly by the parameters  $\alpha$  and  $\gamma$ . They also determine the asymptotic behavior. To find an estimator similar to the one found in Section 4.2, we chose to estimate  $\alpha$ . The derivation of the estimator follows.

$$\begin{aligned} \frac{1}{R(i)} = G(i) &= \frac{1 + \alpha|i|}{\beta + \gamma|i|} \\ \varphi(i, u_R) &\triangleq \frac{|u_R|}{\beta + \gamma|i|} \\ |\hat{i}| = (1 + \alpha|i|)\varphi &\Rightarrow \epsilon = |i| - |\hat{i}| = |i| - (1 + \alpha|i|)\varphi \\ J(\alpha) = \frac{\epsilon^2}{2} &\Rightarrow \nabla J(\alpha) = -\epsilon|i|\varphi \\ \dot{\alpha} = -\gamma\nabla J(\alpha) &= \gamma\epsilon|i|\varphi \end{aligned} \tag{5.11}$$

This was implemented as a discrete estimator in the same fashion as (4.5), using Euler integration. The resulting Simulink diagram is shown in Figure D.14.

We will denote the gain-scheduler using the estimated parameter  $\alpha$  as  $\text{GS}^\alpha$ . The scaling constant was chosen to be  $\mu_{GS} = 100$ . It can be noted that this estimator is estimating a slowly varying parameter.

## Procedure

The motor was either run using a sinusoidal reference with  $A_r = 3\pi$  and  $\omega_r = 3\pi$ , or with a generated trajectory specified to lift the tray of weights from the floor to the highest possible vertical position physically allowed by the rig, and down again. When using the sinusoidal reference, the motor was run for 75 [s], of which the last 60 [s] were recorded. The generated trajectory is shown in Figure 5.10. A load of  $m_w = 0.250$  [kg] was used. The estimated current measurement bias from the EKF was subtracted from the current measurement in all the cases.

The sinusoidal reference was used for the EKF, LMS, GS, and  $\text{GS}^\alpha$ . Tests were also run using a constant resistance in the adaptive law, chosen to be the nominal resistance provided by the motor's manufacturer, and using the PI regulator. The generated trajectory reference was used for the EKF, LMS, and  $\text{GS}^\alpha$ .

## 5.4 Remarks

The D/A converter produced a bias in the voltage actuation signal. The bias seemed to be constant. It was corrected for in software, subtracting as much as possible allowed by the D/A converter's resolution from the signal outputted to the D/A converter in Simulink. The remaining bias was subtracted from the voltage actuation signal inputted to the estimators.

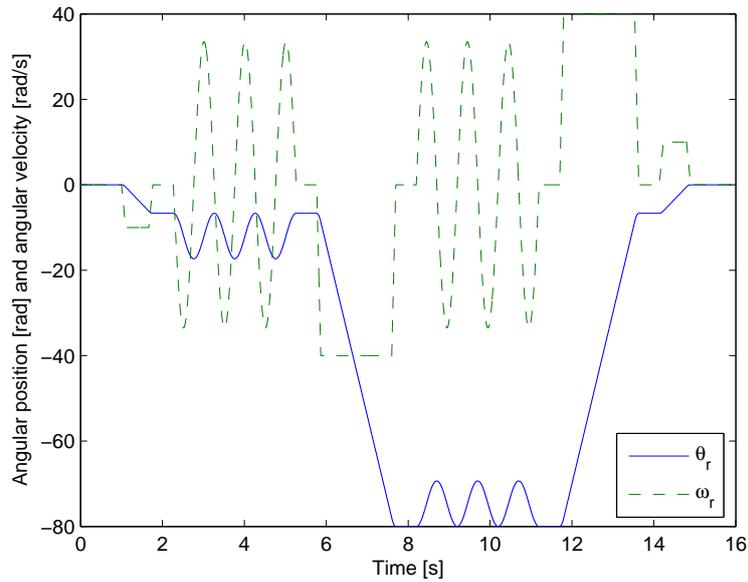


Figure 5.10: Reference trajectory.

All of the experiments, except for the experiment in Section 5.2.4, were done using one motor. One might expect that other motors will result in slightly different estimated values, and that the values obtained during these experiments will not necessarily translate directly to other motors of the same type.

# Chapter 6

## Results and conclusions

### 6.1 Winding resistance and its temperature dependency

To find an estimate of the lower bound for the terminal resistance, one could assume that the contact resistance is negligible at a high current, meaning the lower bound will also be an estimate of the winding resistance.

Figure 6.1 shows the thermal behaviour of the motor, when loaded at  $i_{rms} \approx 2.3$  [A]. The Joule losses are high, leading to a rapid heating of the motor's windings, from about 25 to 90 [°C] in less than 5 minutes. The temperature increase will lead to a resistance increase since the temperature resistance coefficient for copper is a positive value.

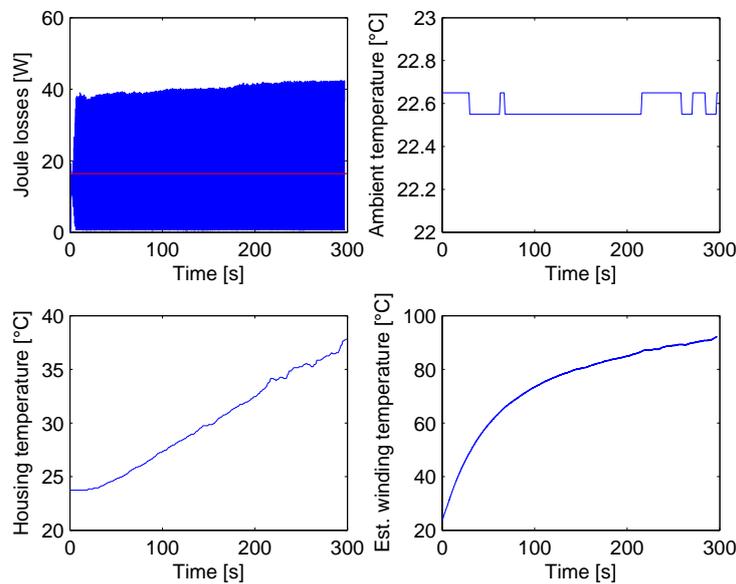


Figure 6.1: Thermal behaviour.

Figure 6.2 shows the estimated resistance of the motor as a function of the estimated winding temperature for two different runs, starting a room temperature. The estimated resistances were filtered with a first-order low-pass Butterworth filter with a cut-off frequency at  $f_c = 1/\tau_1$  (the fastest thermal time constant), since we were only interested in the thermal dynamics. The first 15 seconds of data, while the amplitude of the reference was ramped up to  $A_r$ , were skipped, along with an additional few seconds, to make sure the Kalman filter estimating the resistance had settled.

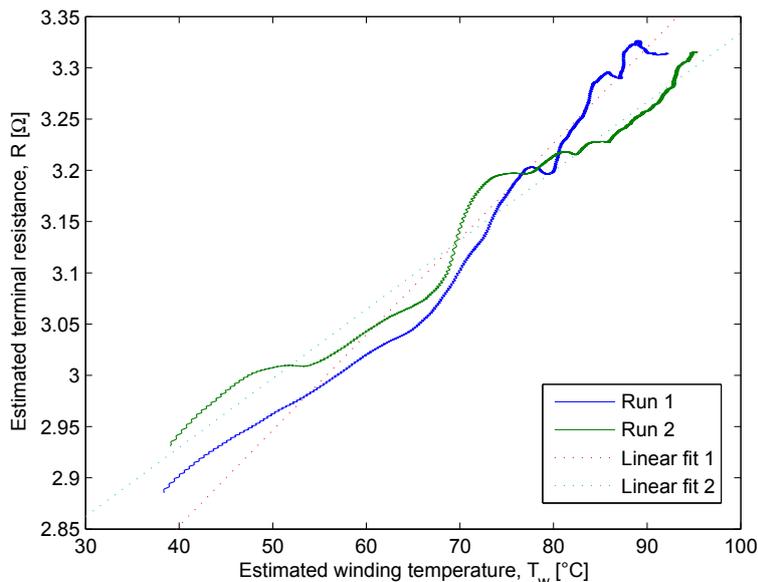


Figure 6.2: Resistance as a function of temperature.

A linear fit was found using linear regression in MATLAB. The fit gives us the two coefficients of the expression:

$$y = ax + b \quad (6.1)$$

They were found to be  $a \in \{0.00933, 0.00674\}$  and  $b \in \{2.48, 2.66\}$ . Comparing Equation (2.23) with (6.1), and solving for the values of  $\alpha$  and  $R_0$  for a given  $T_0$ , gives us

$$\alpha = \frac{a}{aT_0 + b}, \quad R_0 = aT_0 + b. \quad (6.2)$$

Using a reference temperature of  $T_0 = 25$  [°C], the temperature resistance coefficient found using (6.2) was  $\alpha \in \{0.00344, 0.00238\}$  [1/K], both of which are lower than the coefficient listed for copper in [8] (and Section 2.3.2.) The resistance at  $T_0$  becomes  $R_0 \in \{2.71, 2.82\}$  [Ω].

This experiment was done only twice, as the motor needed about  $5\tau_2 \approx 1.5$  [hours] to cool down to room temperature. Thus, it did not give a good statistical basis to provide point estimates for  $\alpha$  and  $R_0$ , and since we were estimating the terminal resistance, we

might expect that any temperature effects in the contact resistance influenced the result. The thermal data cited by the manufacturer might also be somewhat inaccurate for our particular motor mounting.

The experiment did, however, give a good indication that both the extended Kalman filter and the Kalman filter for the thermal dynamics works as expected. The values obtained for the temperature resistance coefficient are close to the listed value for copper,  $\alpha_{Cu} = 0.0039$  [1/K], and the values for the terminal resistance were also close to the value cited by the manufacturer (2.74 [ $\Omega$ ] at a nominal current of 1.45 [A] and a temperature of 25 [ $^{\circ}$ C].)

We would henceforth use a temperature resistance coefficient of  $\alpha = 0.0039$  [1/K], a reference temperature of  $T_0 = 25$  [ $^{\circ}$ C], and set our winding resistance to be  $R_{w0} = 2.74$  [ $\Omega$ ]. The lower bound was set to 2 [ $\Omega$ ], to ensure that the extended Kalman filter would always provide resistance estimates with a higher value than this.

## 6.2 Contact resistance and its current dependency

There was a large variability in the estimated terminal resistance. Figure 6.3 presents four different datasets that were collected using friction as load, as described in Appendix C. It clearly displays the attenuation phenomenon.

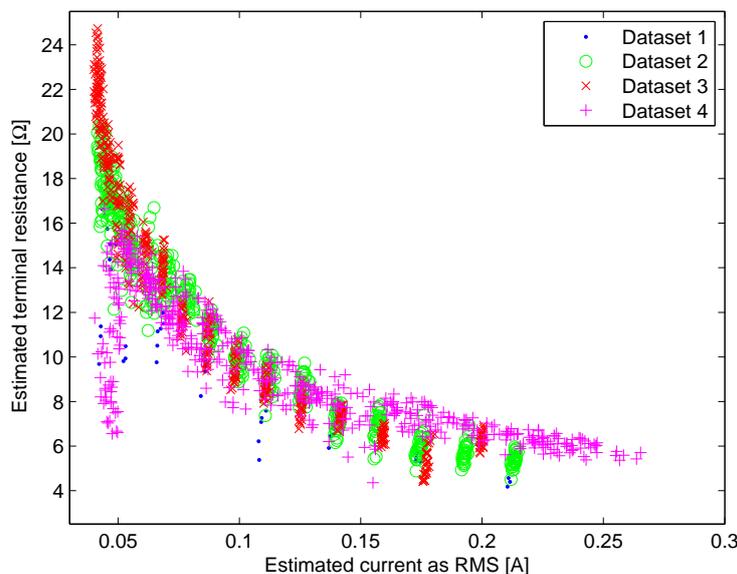


Figure 6.3: Terminal resistance as a function of current.

The most interesting dataset is perhaps No. 4. It is plotted again in Figure 6.4 with the standard error for each point estimate of the resistance. Before this dataset was collected, the motor had been loaded with weights up to 1.000 [kg] before going

back to only friction. It suggests that there is some sort of hysteresis. After loading the motor with a high current, the motor needs to be run for some time at low current before the resistance grows back to a high value. The standard deviation is also very large, especially at lower currents, with a tendency to diminish as the current grows; fewer time-series were therefore gathered when the load was higher. For this dataset, we found the maximum standard deviation to be  $\sigma_{max} = 1.67 \text{ } [\Omega]$ .

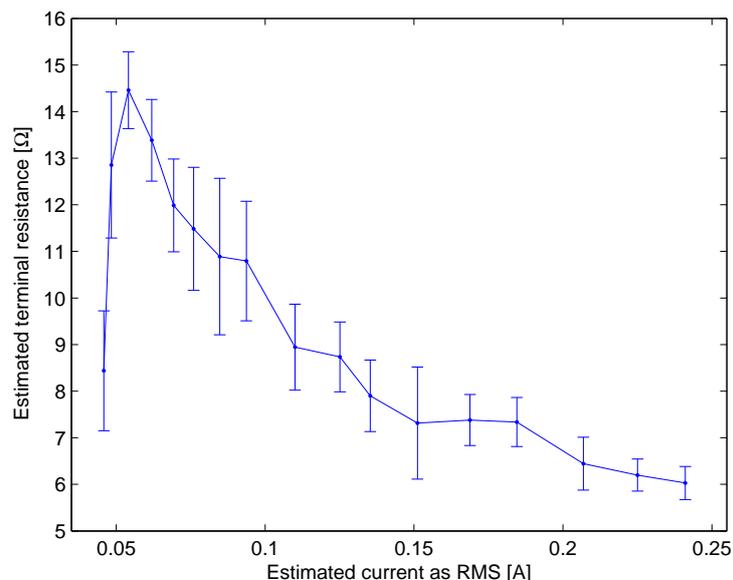


Figure 6.4: Terminal resistance as a function of current with standard error.

The motor always seemed to need some time to “recover” a high resistance when switching from a high to a low current, but not the other way around. The pattern seemed to be that the motor needed more time to “recover” the longer it was run at a high current, and the higher the current was.

A procedure where one successively increases the load, after running the motor with no load for some time, should therefore ensure that the resistance will have stabilized at a high value and successively decrease as the load increases.

The highest resistance will probably be dependent on the state of repair of the system of brushes and slipping at a given point in time. It can be conjectured that a given state will persist for some time, producing repeatable results, but will eventually change. How long a given state will persist does not seem deterministic.

The dataset shown in Figure 6.5 was found using the procedure described above. The maximum resistance was found to be around  $10 \text{ } [\Omega]$ , and this was achieved after running the motor for some time with only friction as a load. The resistance can be seen to be monotonously decreasing as the load increases.

Figure 6.6 displays the fits obtained for the models from Section 2.3.1, using all

available data. It is evident that the rational model provides the best fit. The obtained model parameters are presented in Table 6.1.

Table 6.1: Parameter values from nonlinear regression.

| Rational model, equation (2.22).    |          |
|-------------------------------------|----------|
| $\alpha$                            | 142.256  |
| $\beta$                             | 102.330  |
| $\gamma$                            | 334.304  |
| Exponential model, equation (2.21). |          |
| $a$                                 | 21.3489  |
| $b$                                 | -13.8737 |
| $c$                                 | 3.9170   |

Since there is such a large variance in the obtained resistance estimates, a static model will not provide a perfect estimate at a given armature current. The best-fit curve can, however, be assumed to be that which minimizes the sum of the squared residuals. If one has obtained a large dataset which includes all, or most of, the variation possible, the obtained static model should be representative of the mean. It is unlikely that all of the possible variation will occur during a period of use of the motor, so the mean value given by the model fit will likely be consistently higher or lower than the true value of the resistance during that period.

The model fit of the resistance can be called the motor's  $i$ - $u$  characteristic. If a static gain-scheduler is going to be used, this characteristic should be updated from time to time to ensure that the model fit represents the resistance attenuation as closely as possible for the given state of repair of the system of brush and slipring.

As for the temperature dependency, the temperature correction yielded a maximum difference between uncorrected and corrected resistance as  $R - R_0 \approx 0.55$  [ $\Omega$ ]. This was for the data obtained using the heaviest weight,  $m_w = 1.495$ , producing  $i_{rms} \approx 2.3$ . For currents below 0.7 [A], the current limit for the SCU, the difference had a maximum of  $R - R_0 \approx 0.073$  [ $\Omega$ ]. This suggests that the temperature dependency can be ignored when using the motor at current magnitudes below the current limit.

During the experiments, a possible dependency on the turning direction of the rotor was observed, where the resistance would be slightly different in the positive and negative directions. The resistance when the motor was stalled also seemed to be lower than when it was running. These effects should be investigated if better accuracy is desired using a gain-scheduler.

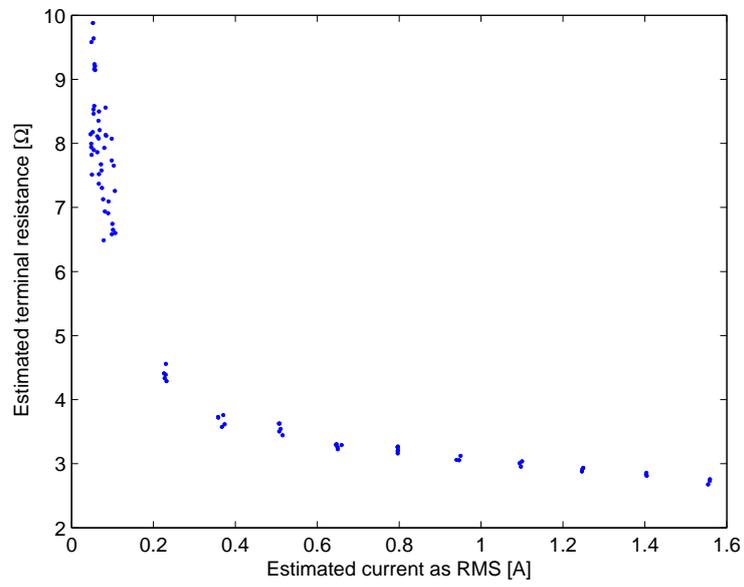


Figure 6.5: Terminal resistance as a function of current.

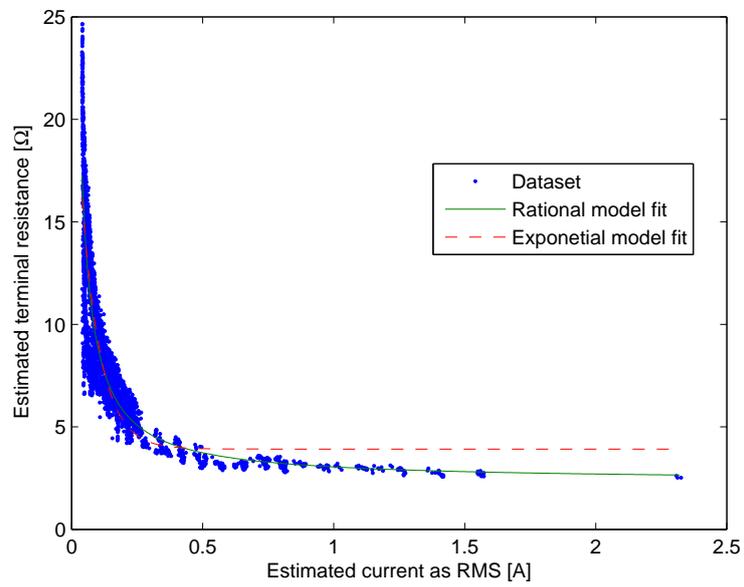


Figure 6.6: Fits for the rational and the exponential model.

### 6.3 Comparison of adaptive schemes

Figures 6.7, 6.8, 6.9, 6.10, 6.11, and 6.12 show the results from the runs using the sinusoidal reference. Two seconds worth of data are shown in each case, since the behavior is similar throughout all of the time-series. Figures 6.13, 6.14, and 6.10 show the results when using the generated trajectory reference.

We have chosen to present the voltage actuation signal, current measurement, current reference, and the current error for all the cases. The estimated resistance is presented for all but the case where a constant resistance was used in the adaptive control law and the case where the PI regulator was used.

The estimated current measurement bias from the EKF was subtracted from the current measurement outputted from the Butterworth filter in all the cases, and the measurement is therefore given as  $i_m = i_f - \hat{i}_b$ . The current error is given by  $e = i_r - i_m$ .

The resulting RMS values for the current error are calculated using the entire time-series for the error in all the cases. The RMS error values are presented in Tables 6.2 and 6.3.

Table 6.2: RMS errors for sinusoidal reference.

| Scheme       | RMSE   |
|--------------|--------|
| EKF          | 0.0274 |
| LMS          | 0.0424 |
| GS           | 0.132  |
| GS $^\alpha$ | 0.0549 |
| Const. R     | 0.318  |
| PI           | 0.517  |

Table 6.3: RMS errors for generated trajectory reference.

| Scheme       | RMSE   |
|--------------|--------|
| EKF          | 0.0237 |
| LMS          | 0.0346 |
| GS $^\alpha$ | 0.0463 |

The RMS values for the errors in Tables 6.2 and 6.3 suggest that the EKF provides the best accuracy, followed by the LMS estimator. It is interesting to note that the PI regulator performs even worse than simply using a constant resistance value in the adaptive control law/feedforward controller.

Looking at Figures 6.7 and 6.8, we can see that the EKF and the LMS estimator have very similar behavior. The EKF seems to provide better accuracy, especially when

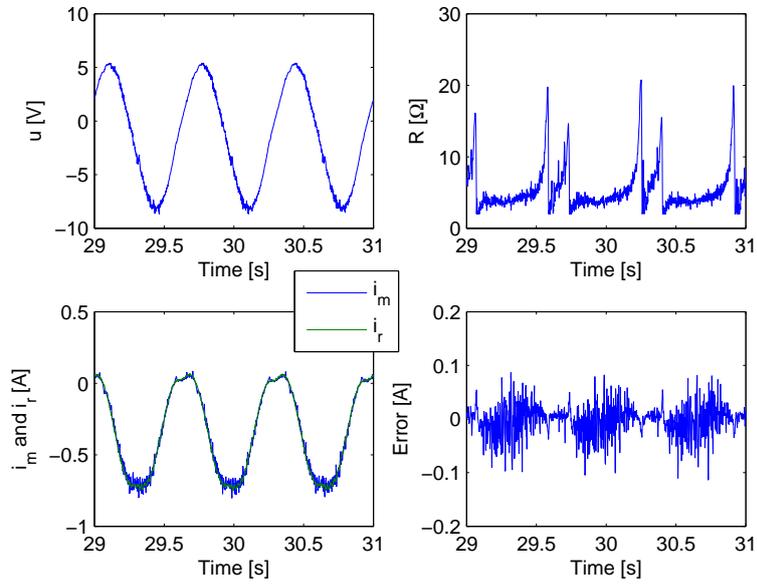


Figure 6.7: EKF when using the sinusoidal reference.

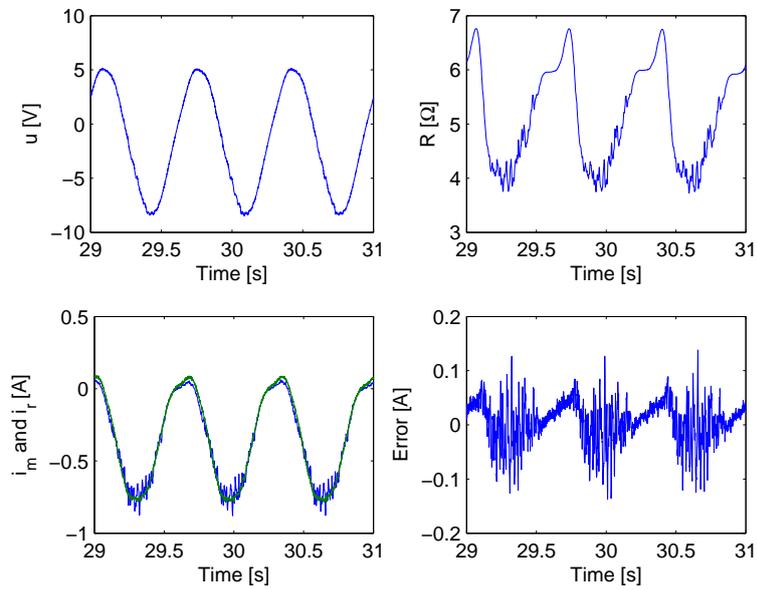


Figure 6.8: LMS when using the sinusoidal reference.

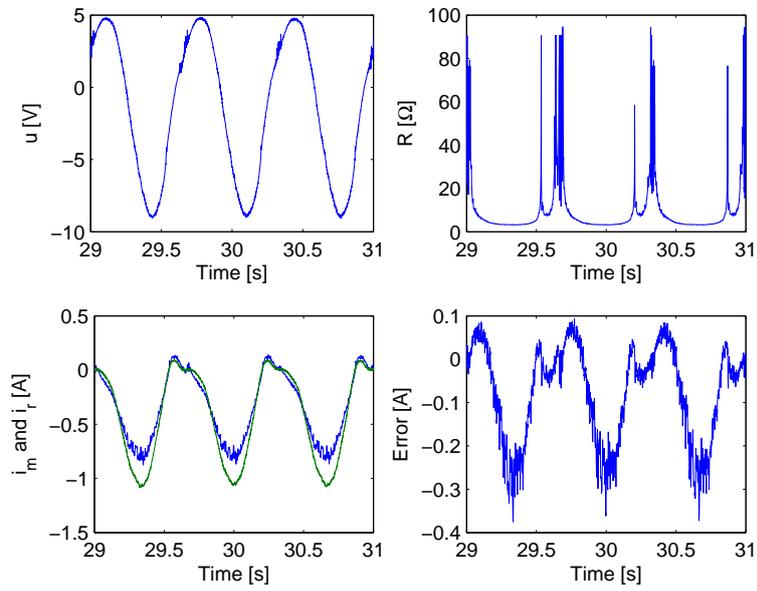


Figure 6.9: Gain-scheduler (GS) when using the sinusoidal reference.

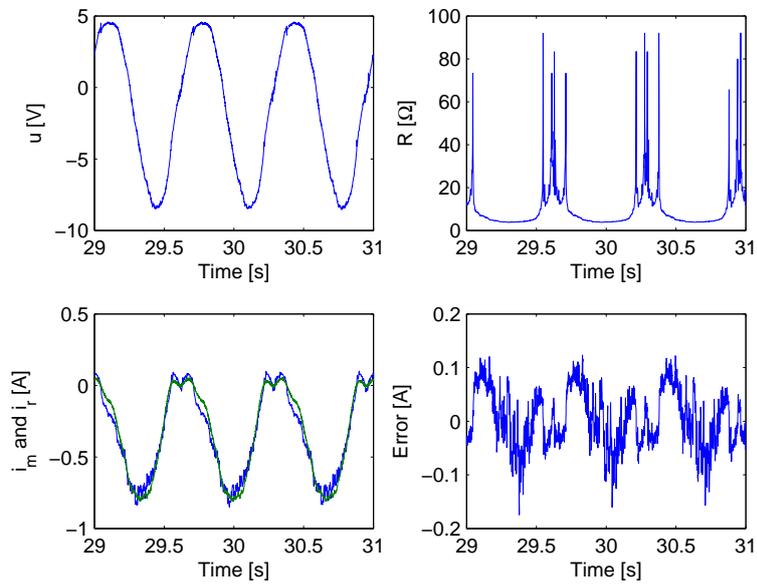


Figure 6.10: Gain-scheduler with one adaptive parameter ( $GS^\alpha$ ) when using the sinusoidal reference.

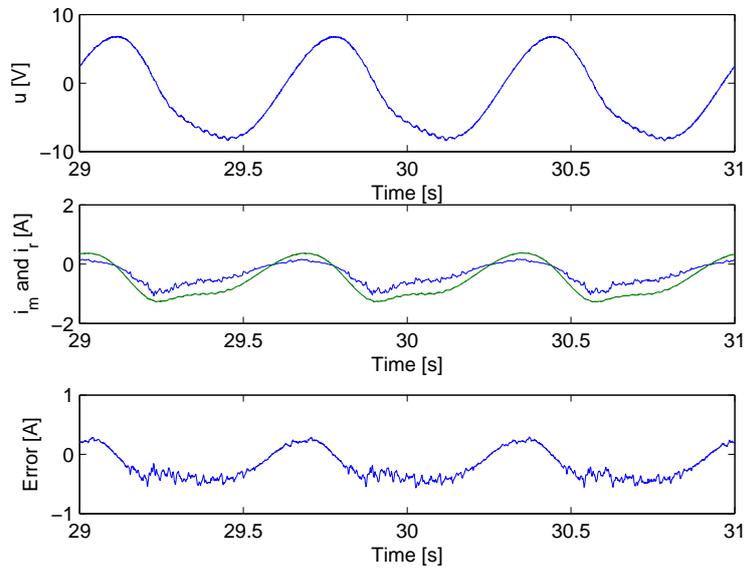


Figure 6.11: Result when having a constant resistance in the adaptive control law when using the sinusoidal position reference.

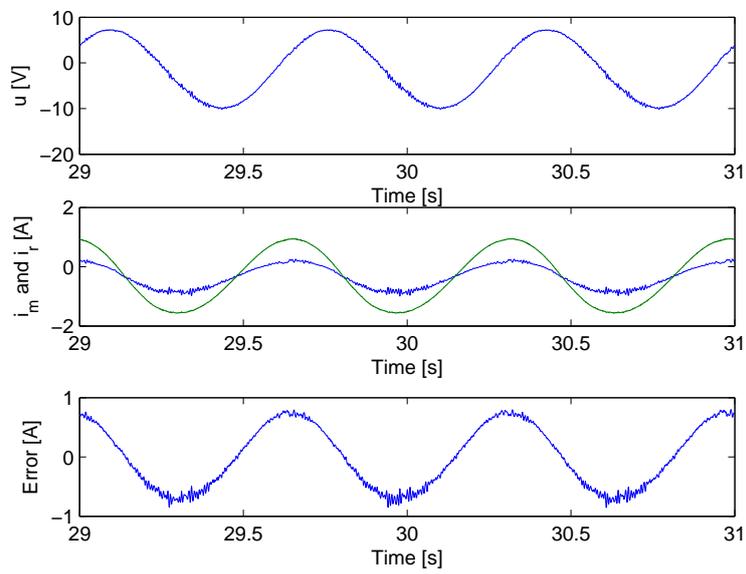


Figure 6.12: PI regulator when using the sinusoidal reference.

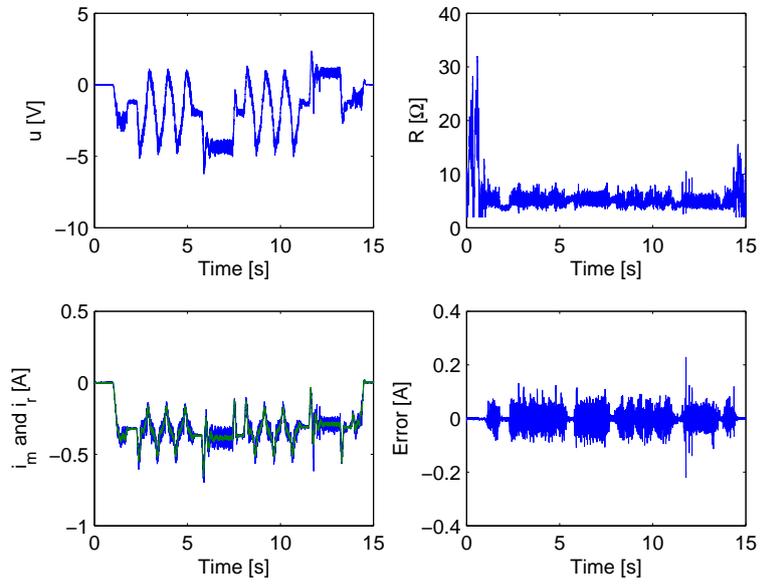


Figure 6.13: EKF when using the generated trajectory reference.

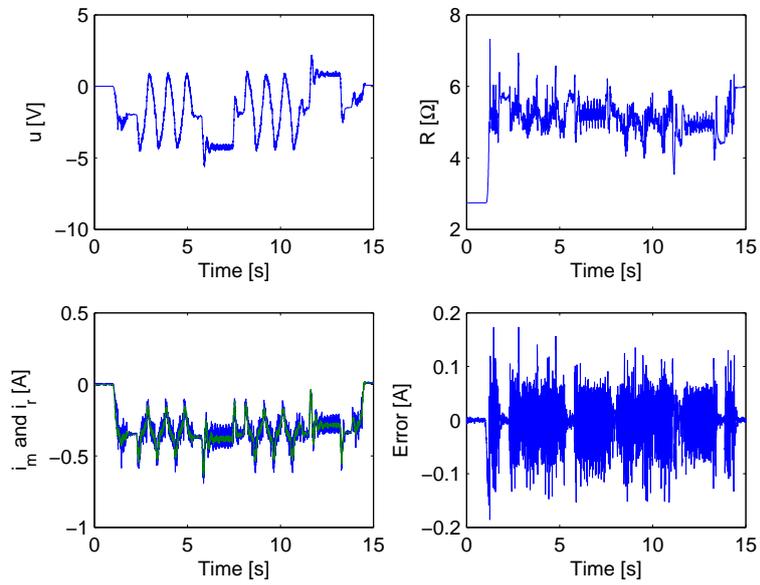


Figure 6.14: LMS when using the generated trajectory reference.

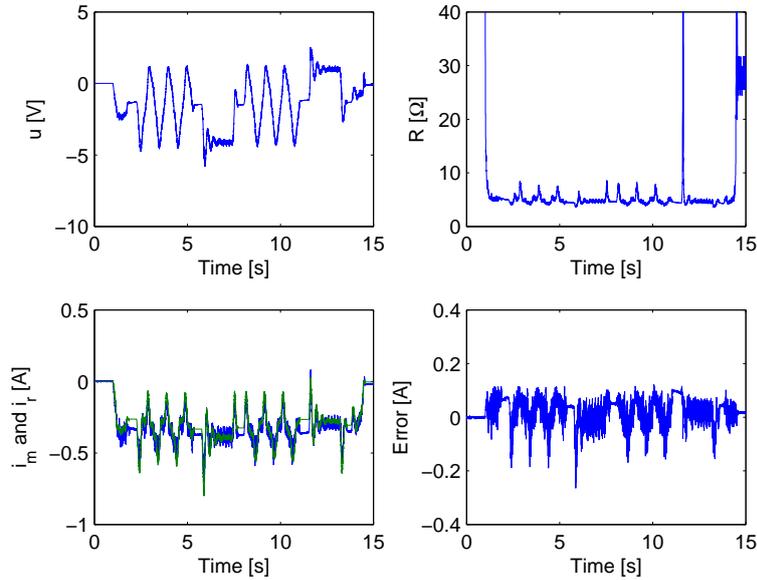


Figure 6.15: Gain-scheduler with one adaptive parameter ( $GS^\alpha$ ) when using the generated trajectory reference.

the magnitude of the current is small. When the magnitude of the current is small, the estimated resistance from the EKF can be seen to have high spikes, comparable to the ones found in the calculated resistance from the gain-scheduler (Figures 6.9 and 6.10.) This should indicate that the gain-scheduler at least has the correct behavior. Looking at the voltage actuation signal for the gain-scheduler, there seem to be some jerking when some of these spikes occur, which is undesirable. It indicates that the calculated resistance is too high at these points, and not in accordance with the actual resistance.

The static gain-scheduler performs rather badly, but is still better than using a constant resistance and the PI regulator. The gain-scheduler with one adaptive parameter performs much better. The estimated parameter  $\alpha$  was reduced to about 120, from the value of about 140 from Table 6.1. Recalling that the asymptote of the model is given by  $\gamma/\alpha$ , this indicates that the model provides a resistance that is too low compared to the true resistance. This can also be seen in Figure 6.9, where the measured current is smaller in magnitude than the current reference.

Looking at Figure 6.15, at the points where the motor is stalled, the measured current is typically of greater magnitude than the current setpoint, indicating that the calculated resistance is too high. This is in accordance with observations made during the experiments: the resistance seemed to be lower when the motor was stalled, than when it was running.

The motor used in the experiments suffered some abuse both when setting up the different experiments and during some of them. For example: in one instance the thread

attached to the mounted spindle became jammed, stalling the motor, and resulting in a winding temperature above the stated maximum permissible winding temperature. The state of repair of the brushes and slipping might thus have changed in the course of the experimentation. As such, the obtained model fit is probably less accurate than it might otherwise be. If time had permitted, a fresh dataset could have been obtained, fitted to the model, and the resulting parameter values tested in the gain-scheduler.

As a quick test of the notion that a better fit might have been obtained, we tried to find a fit to the model by hand using a heuristic approach. Since the large spikes in the resistance calculated using the gain-scheduler seemed somewhat unphysical, we tried to reduce the maximum resistance in the model by setting it to  $\beta = R(0) = 35$ . By setting the asymptote to something close to the nominal value for the resistance  $\gamma/\alpha = R(\infty) = 2.75$ , we could tune one parameter,  $\alpha$ , by setting  $\gamma = 2.75\alpha$ . We tried using  $\alpha = 85$ . This was reduced to about 70 by the estimator for  $\alpha$  when running the motor for a few seconds with the new fit. We henceforth set the initial value of  $\alpha$  to 70. We ran the two tests again with  $GS^\alpha$ , reducing  $\mu_{GS}$  to 25. The results are shown in Table 6.4 and Figures 6.16 and 6.17. As we can see, the results are close to what we got when using the LMS estimator. The largest error in Figure 6.17 is present when the motor is close to being stalled, indicating that a different set of model parameters should be used when this is the case.

This simple test suggests two things: the data obtained during the experiments to find the resistance's current dependency no longer represents the state the motor is in, and the nonlinear regression method might tend to find a value for  $\beta$  that is generally too high. Setting  $\beta$  to a reasonable constant value while doing the regression might prove to yield a better result when using the values obtained for the remaining two parameters in the gain-sceduler.

Table 6.4: RMS errors for gain-scheduler fitted by hand.

| Scheme      | RMSE   | Reference  |
|-------------|--------|------------|
| $GS^\alpha$ | 0.0417 | Sinusoidal |
| $GS^\alpha$ | 0.0349 | Trajectory |

### 6.3.1 Remarks

We attempted to tune the scaling constant/step length in the LMS estimator to see if it could perform better, but were unable to match the EKF. The LMS estimator also became unstable if the scaling constant was set too high, e.g.,  $\mu = 0.1$ .

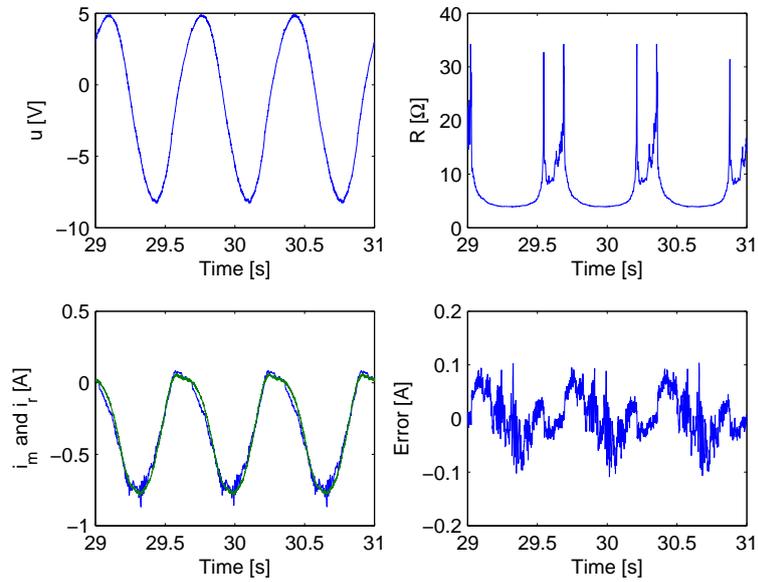


Figure 6.16: Gain-scheduler with one adaptive parameter ( $GS^\alpha$ ) when using the sinusoidal reference and parameters fitted by hand.

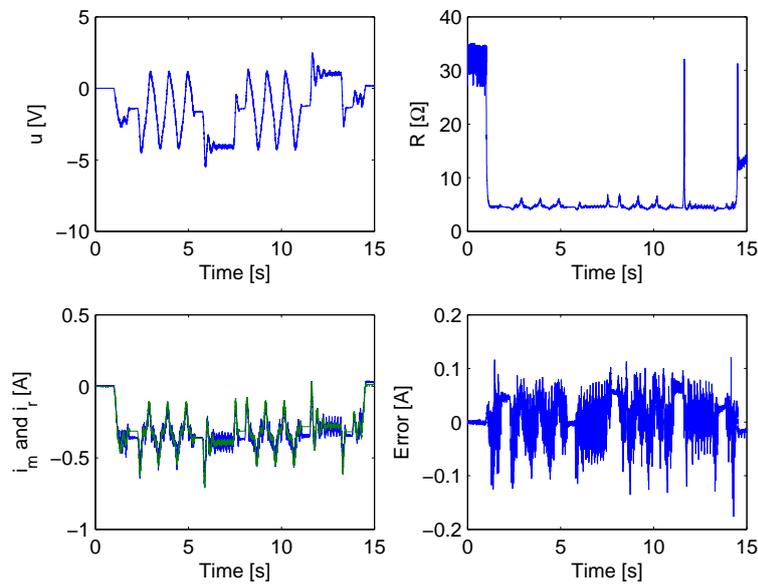


Figure 6.17: Gain-scheduler with one adaptive parameter ( $GS^\alpha$ ) when using the generated trajectory reference and parameters fitted by hand.

## 6.4 Conclusions

In the course of this investigation we have found that in order to control the motor's armature current using a discrete controller, current feedback provides very poor performance. The alternative that has been examined is feedforward control, and as we have seen, it requires accurate estimates of the motor's terminal resistance to perform well.

It was found that the resistance attenuation phenomenon due to current load is best fitted using a rational function. The resulting gain-scheduler based on this model did not perform as well as estimating the resistance using an LMS estimator (as is presently being done for the system under consideration) in the tests that we conducted. This is most likely due to unmodeled effects, resulting in a constantly changing  $i$ - $u$  characteristic for the motor. There is evidence that suggests that the gain-scheduler can be made to work better if the model parameters are routinely updated (perhaps using some physical insight when performing the regression), and some adaptive scheme is adopted to adjust some, or ideally all, the model's parameters while it is in use. This has the potential benefit of estimating (a) slowly varying parameter(s), rather than a quickly varying one, like the terminal resistance. The gain-scheduler should also be augmented with the behavior exhibited when the motor is stalled, and if possible, any unmodeled effects that have not been considered in this investigation.

The extended Kalman filter which was developed to provide resistance and current estimates for finding a model fit, was shown to be the best performing adaptive scheme, when used in conjunction with the adaptive law, in the tests that we conducted. The extended Kalman filter is very complex compared to the LMS algorithm, and might therefore not be suitable for implementation in the digital signal processor used on the segment control unit.

## 6.5 Recommendations

If it is desirable to take up and investigate further the feasibility of using a gain-scheduler for adapting to the changing resistance, it may be required to augment it with a model fit for the attenuation when the motor is stalled, as well as accommodating for unmodeled phenomena, either using a more complex model incorporating those phenomena, or using some form of online parameter estimation for the model parameters. One could perhaps suggest employing batch processing of current measurements and resistance estimates, using nonlinear regression, in higher layers of the control system, or while the system is idle. Using some physical intuition when fitting data to the model might also prove to yield better results in practice. It would also be natural to look at a larger sample of motors, in order to ascertain how much difference actually exist, motor-to-motor.

If the Kalman filter should be adopted, it can be recommended to look at the unscented Kalman filter in order to provide more accurate estimates. For this particular system, the unscented Kalman filter might also prove to be computationally lighter, since the unscented transform is used, instead of the linearization performed using the Jacobians (which have been shown to require much computation when they are updated.)

# Bibliography

- [1] Jens G. Balchen, Trond Andresen, and Bjarne A. Foss. *Reguleringsteknikk*. Institutt for teknisk kybernetikk, NTNU, 2003.
- [2] John Belcher and Walter Lewin. Lecture notes: Faraday's law, April 2002.
- [3] Dieter Betz. Relationship between contact resistance and wear in sliding contacts. *IEEE Transactions on Parts, Hybrids, and Packaging*, PHP-10(1):32–37, March 1974.
- [4] Robert Grover Brown and Patrick Y. C. Hwang. *Introduction to random signals and applied Kalman filtering*. John Wiley & Sons, Inc., 1997.
- [5] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, Inc., 1999.
- [6] M. Carroll Croarkin, William F. Guthrie, George W. Burns, Margaret Kaeser, and Gregory F. Strouse. Temperature-electromotive force reference functions and tables for the letter-designated thermocouple types based on the its-90. Monograph 175, National Institute of Standards and Technology, 1993.
- [7] Olav Egeland. *Servoteknikk*. Tapir Forlag, 1993.
- [8] Paul M. Fishbane, Stephen Gasiorowicz, and Stephen T. Thornton. *Physics for Scientists and Engineers*. Prentice-Hall, Inc., 1996.
- [9] Arthur Gelb, Joseph F. Kasper, Jr., Raymond A. Nash, Jr., Charles F. Price, and Arthur A. Sutherland, Jr. *Applied Optimal Estimation*. The M.I.T. Press, 1974.
- [10] Petros A. Ioannou and Jing Sun. *Robust Adaptive Control*. Prentice-Hall, Inc., 1996.
- [11] William J. Karnavas, Paul J. Sanchez, and A. Terry Bahill. Sensitivity analyses of continuous and discrete systems in the time and frequency domains. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(2):488–501, March/April 1993.
- [12] J. Y. Keller and M. Darouach. Optimal two-stage kalman filter in the presence of random bias. *Automatica*, 33(9):1745–1748, 1997.

- [13] Hassan A. Khalil. *Nonlinear Systems*. Prentice-Hall, Inc., 3rd edition, 2000.
- [14] John H. Lienhard IV and John H. Lienhard V. *A heat transfer textbook*. Phlogiston Press, 2006.
- [15] Kjell Malvig, editor. *Analog Signalbehandling*. Institutt for teknisk kybernetikk, NTNU, January 2002.
- [16] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer-Verlag New-York, Inc., 1999.
- [17] Chris C. Paige. Properties of numerical algorithms related to computing controllability. *IEEE Transactions on Automatic Control*, AC-26(1):130–138, February 1981.
- [18] Zoya Popović and Branko D. Popović. *Introductory Electromagnetics*. Prentice-Hall, Inc., 1999.
- [19] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C++*. Cambridge University Press, 2nd edition, 2002.
- [20] David A. Ratkowsky. *Handbook of Nonlinear Regression Models*. Marcel Dekker, Inc., 1989.
- [21] Dan Simon. *Optimal State Estimation*. John Wiley & Sons, Inc., 2006.
- [22] Eric A. Wan and Rudolph van der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of Symposium 2000 of Adaptive Systems for Signal Processing, Communication and Control*. IEEE, October 2000.
- [23] Bernard Widrow and Samuel D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, Inc., 1985.

# Appendix A

## Additional figures

### A.1 Open loop gain vs. frequency and resistance

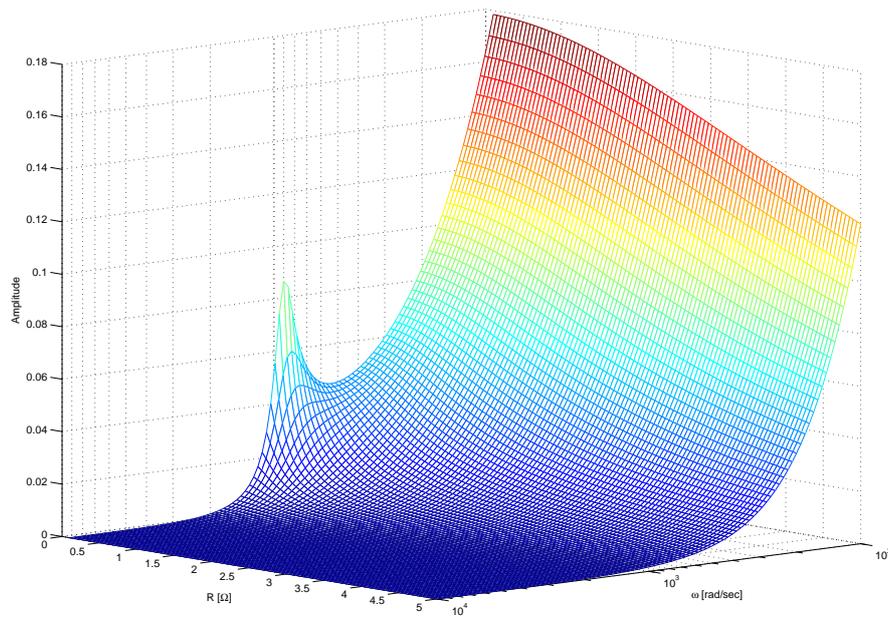


Figure A.1: Open loop gain vs. frequency and resistance ( $|\hat{G}_3(j\omega, R)|$ ).

## A.2 Closed loop gain vs. frequency and resistance

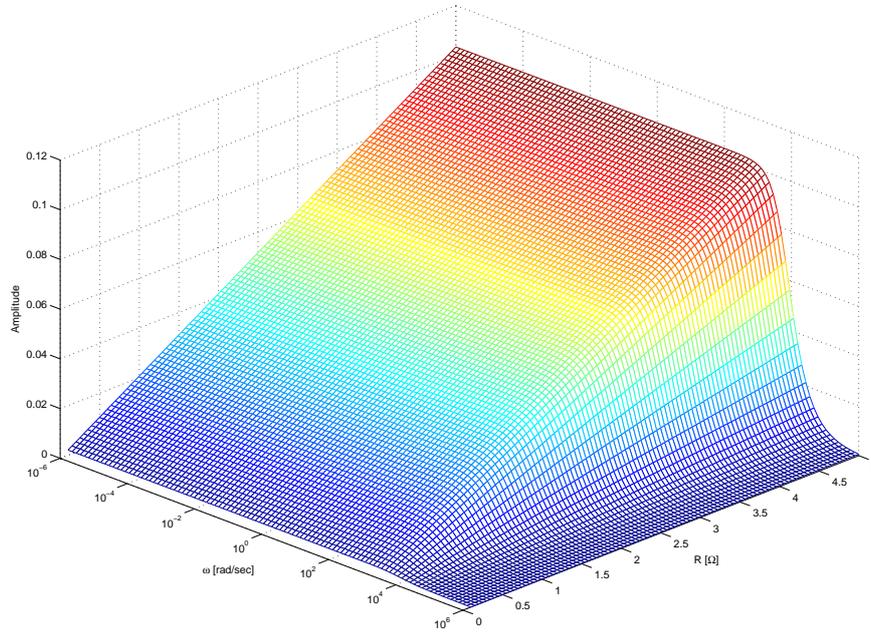


Figure A.2: Closed loop gain vs. frequency and resistance ( $|\hat{G}_T(j\omega, R)|$ ).

# Appendix B

## Contents of DVDs

There are five accompanying digital versatile discs (DVDs.) The experimental data takes up considerable space. The data from the experiments to determine the contact resistance have been divided into datasets. Here follows a brief description of the contents of each DVD.

**Disc No. 1** Contains experimental data. Dataset No. 1, 5, 6, 7, 8, 10, 11, data to show the bias rejection ability, data for the test performed with the spring to show the extended Kalman filters behavior, the data collected to when estimating the electromagnetic constant, the data from the comparisons, and the processed datasets.

**Disc No. 2** Contains experimental data. Dataset No. 2.

**Disc No. 3** Contains experimental data. Dataset No. 3.

**Disc No. 4** Contains experimental data. Dataset No. 4.

**Disc No. 5** Contains experimental data. Dataset No. 9, and the data for the temperature dependency experiment. Also contains the MATLAB scripts and Simulink models for processing the experimental data, files for the extended Kalman filter implementation, source code, Simulink models and related MATLAB scripts for executing the experiments, the MATLAB scripts used for the analysis and implementing and performing nonlinear regression, and miscellaneous files, such as datasheets, SPICE (Simulation Program with Integrated Circuit Emphasis) netlist for the Sallen-Key filter, the first implementation of the Gauss-Jordan algorithm, and the files used to determine commutation noise.

### B.1 MATLAB scripts

This section includes descriptions of the MATLAB scripts used. They are all located on disc No. 5. There are other MATLAB scripts on the DVD, but the ones described here are the most important that have been used in the making of the report.

### B.1.1 Analysis

#### **root\_locus.m**

Creates a root locus plot for current feedback.

Location: `matlab_scripts/analysis//oot_locus.m`

#### **compute\_discrete\_ss\_and\_jacobi.m**

Outputs discrete and linearized state and input matrices for combined DC motor and Butterworth filter model.

Location: `matlab_scripts/analysis/compute_discrete_ss_and_jacobi.m`

#### **observability\_analysis.m**

Observability analysis, discrete linearized system.

Location: `matlab_scripts/analysis/observability_analysis.m`

#### **gain\_vs\_resistance\_and\_frequency.m**

Creates Figures A.1 and A.2.

Location: `matlab_scripts/analysis/gain_vs_resistance_and_frequency.m`

### B.1.2 Experiments

#### **trajectory\_generator.m**

Calculates angular position reference trajectory, given start position, final position, acceleration interval, and desired angular velocity.

Location: `ekf_implementation/trajectory_generator.m`

#### **spindle\_inertia.m**

Finds the moment of inertia for each of the spindles used.

Location: `ekf_implementation/spindle_inertia.m`

#### **set\_params.m**

Sets some parameters, such as sampling time, how many consecutive runs of a chosen Simulink model, and how long to run the model.

Location: `ekf_implementation/set_params.m`

#### **init.m**

Initializes variables before a run, such as the thermal model, controller gain(s), initial conditions to the EKF etc.

Location: `ekf_implementation/spindle_inertia.m`

### **thermal\_model.m**

Finds state estimator from thermal model.

Location: `ekf_implementation/thermal_model.m`

### **run\_model.m**

Sets up the WinCon environment, runs a selected Simulink model, and saves the recorded data.

Location: `ekf_implementation/run_model.m`

### **run\_different\_amps.m**

Performs an automated experiment sequence, varying the amplitude of the sinusoidal position reference.

Location: `ekf_implementation/run_different_amps.m`

### **run\_different\_freqs.m**

Performs an automated experiment sequence, varying the frequency of the sinusoidal position reference.

Location: `ekf_implementation/run_different_freqs.m`

### **dc\_motor\_init.m**

Sets up the needed variables for the DC motor simulator. The simulator was used to debug and validate the EKF implementation. Location: `ekf_implementation/dc_motor_init.m`

## **B.1.3 Experimental data processing**

### **load\_and\_process\_data.m**

Loads data as saved by WinCon, picks out the desired data, processes it using `process_data.m`, and saves the processed data to a new file.

Location: `experimental_data/load_and_process_data.m`

### **process\_data.m**

Processes inputted time-series, computing mean values, standard deviations, RMS-values, etc.

Location: `experimental_data/process_data.m`

### **load\_data.m**

Loads data as saved by WinCon, but does not process it. Suitable for looking at individual time-series.

Location: `experimental_data/load_data.m`

## B.1.4 Nonlinear regression

### **load\_processed\_data.m**

Loads processed data and prepares it is for use with a nonlinear regression method.

Location: `matlab_scripts/regression/load_processed_data.m`

### **linear\_regression.m**

Performs linear regression using Moore-Penrose pseudo-inverse.

Location: `matlab_scripts/regression/linear_regression.m`

### **nonlinear\_regression.m**

Performs nonlinear regression using Gauss-Newton iteration.

Location: `matlab_scripts/regression/nonlinear_regression.m`

### **run\_regression.m**

Performs nonlinear regression using MATLAB's `lsqnonlin` function with a chosen model.

Location: `matlab_scripts/regression/run_regression.m`

### **residual\_exponential\_vector.m**

Vectorized residual and Jacobian using the exponential model, suitable for `lsqnonlin`.

Location: `matlab_scripts/regression/residual_exponential_vector.m`

### **residual\_rational\_vector.m**

Vectorized residual and Jacobian using the rational model, suitable for `lsqnonlin`.

Location: `matlab_scripts/regression/residual_rational_vector.m`

## B.2 Extended Kalman filter implementation

This section includes descriptions of the files used in the implementation of the extended Kalman filter and the Simulink models used for the various experiments. The files are located on disc No. 5. These are the files that specify the implementation. Other files are present on the DVD, but are mostly files that have been automatically generated by Real-Time Workshop.

### **nr.h**

Header file for Gauss-Jordan elimination function, adapted from [19].

Location: `ekf_implementation/nr.h`

### **nrutil.h**

Header file for matrix and vector memory allocation functions, adapted from [19].  
Location: `ekf_implementation/nrutil.h`

### **nrutil.c**

Functions for matrix and vector memory allocation functions, adapted from [19].  
Location: `ekf_implementation/nrutil.c`

### **gaussj.c**

Gauss-Jordan elimination algorithm, from [19].  
Location: `ekf_implementation/gaussj.c`

### **constants.h**

Header file defining various constants.  
Location: `ekf_implementation/constants.h`

### **joint\_ekf\_R.h**

Header file for function that computes the discrete state-space matrices and their linearizations with respect to the terminal resistance,  $R$ .  
Location: `ekf_implementation/joint_ekf_R.h`

### **comp\_ss\_jacobi\_R.c**

Function that computes the discrete state-space matrices and their linearizations with respect to the terminal resistance,  $R$ .  
Location: `ekf_implementation/comp_ss_jacobi_R.c`

### **joint\_ekf\_R\_K.h**

Header file for function that computes the discrete state-space matrices and their linearizations with respect to the terminal resistance,  $R$ , and the electromagnetic constant  $K = K_T = K_E$ .  
Location: `ekf_implementation/joint_ekf_R_K.h`

### **comp\_ss\_jacobi\_R\_K.c**

Function that computes the discrete state-space matrices and their linearizations with respect to the terminal resistance,  $R$ , and the electromagnetic constant  $K = K_T = K_E$ .  
Location: `ekf_implementation/comp_ss_jacobi_R.c`

### **matrix\_operations.h**

Header file for functions that perform various matrix operations.

Location: `ekf_implementation/matrix_operations.h`

### **matrix\_operations.c**

Functions that perform various matrix operations, such as matrix addition and multiplication.

Location: `ekf_implementation/matrix_operations.c`

### **joint\_ekf\_R\_ib.c**

Implementation of the extended Kalman filter as a Simulink S-Function, estimating five states, torque load, current measurement bias, and terminal resistance.

Location: `ekf_implementation/joint_ekf_R_ib.c`

### **joint\_ekf\_R\_K\_ib.c**

Implementation of the extended Kalman filter as a Simulink S-Function, estimating five states, torque load, current measurement bias, terminal resistance, and electromagnetic constant.

Location: `ekf_implementation/joint_ekf_R_ib.c`

### **kalman\_R\_ib.mdl**

Simulink model implementing the EKF S-Function for estimating motor and Butterworth filter states, resistance, current measurement bias and torque load/disturbance. Also implements LMS estimator and Kalman filter for the thermal model.

Location: `ekf_implementation/kalman_R_ib.mdl`

### **kalman\_R\_K\_ib.mdl**

Simulink model implementing the EKF S-Function for estimating motor and Butterworth filter states, resistance, electromagnetic constant, current measurement bias and torque load/disturbance. Also implements LMS estimator and Kalman filter for the thermal model.

Location: `ekf_implementation/kalman_R_K_ib.mdl`

### **comparison.mdl**

Simulink model for comparing control schemes. Implements EKF, LMS estimator, gain-scheduler, and adaptive law/feedforward controller. Also implements a PI regulator.

Location: `ekf_implementation/comparison.mdl`

## Appendix C

# Experimental setups

The collected data are provided on the accompanying digital versatile discs (DVDs). Details for the experimental setup for each of the datasets are presented here. It is the data from the experiments to determine the contact resistance.

A sinusoidal position reference was used. When loading the motor with its own friction and inertia, here denoted by  $\mu$ , either the amplitude was fixed, while the frequency was varied, or the frequency was varied. Loading the motor with weights, the amplitude was fixed at  $A_r = 3\pi$ , and the frequency was fixed at  $\omega = 3\pi$ . The different references were then given by

$$\begin{aligned}\theta'_{r1} &= A_r \sin(\omega_{rn}t) \\ \theta'_{r2} &= A_{rn} \sin(\omega_r t) \\ \theta'_{r3} &= A_r \sin(\omega_r t).\end{aligned}$$

When stepping through the different frequencies we used

$$\omega_{rn} = n\frac{1}{4}\pi, \quad \{n : n \in N_j\},$$

and when stepping through different amplitudes we used

$$A_{rn} = n\frac{1}{4}\pi, \quad \{n : n \in N_j\},$$

taking the index,  $n$ , from one of the following sets:

$$\begin{aligned}N_1 &\in \{4, 5, 6, \dots, 20\} \\ N_2 &\in \{4, 6, 8, \dots, 20\}\end{aligned}$$

When loading the the motor with weights, we successively increased the load, using

weights with masses from the following sets:

$$M_1 \in \{0.150, 0.200, 0.250, 0.300, 0.350, 0.400, 0.450, 0.500, 0.550\}$$

$$M_2 \in \{0.600, 0.650, 0.700, 0.750, 0.800, 0.900, 1.000\}$$

$$M_3 \in \{1.495\}$$

$$M_4 \in \{0.200, 0.300, 0.400, 0.500\}$$

$$M_5 \in \{0.600, 0.700, 0.800, 0.900, 1.000\}$$

$$M_6 \in \{0.050, 0.100\}$$

$$M_7 \in \{0.050, 0.100, 0.150, 0.200, 0.250, 0.300, 0.350, 0.400, 0.450, 0.500, 0.550\}$$

$$M_8 \in \{0.600, 0.700, 0.800, 0.900, 1.000\}$$

We recorded a number of time-series for each load,  $N_{ts}$ . The length of the time-series is denoted  $\Delta t$ , and is presented in Table C.1, together with combinations of the above relations, which should specify how each dataset was produced. The motor typically ran for about 15 seconds before a time-series was recorded, in order to ensure that the amplitude of the reference had been ramped up to the desired magnitude, and that any unwanted transients in the extended Kalman filter had settled.

Table C.1: Experimental setup for each dataset.

| Dataset no. | Load  | Position reference                    | $N_{ts}$      | $\Delta t$ |
|-------------|---|---------------------------------------|---------------|------------|
| 1           | $\mu$                                       | $\theta_{r1}, N_1$                    | 3             | 60 [s]     |
| 2           | $\mu$                                       | $\theta_{r1}, N_1$                    | 30            | 60 [s]     |
| 3           | $\mu$                                       | $\theta_{r1}, N_1$                    | 30            | 60 [s]     |
| 4           | $\mu$                                       | $\theta_{r1}, N_1$                    | 30            | 60 [s]     |
| 5           | $\{m_w : m_w \in M_1 \cup M_2\}$            | $\theta_{r3}$                         | $\{15, 5\}$   | 60 [s]     |
| 6           | $\{m_w : m_w \in M_3\}$                     | $\theta_{r3}$                         | 3             | 60 [s]     |
| 7           | $\mu \wedge \{m_w : m_w \in M_4 \cup M_5\}$ | $\theta_{r1}, N_2 \wedge \theta_{r3}$ | $\{5, 5, 3\}$ | 60 [s]     |
| 8           | $\{m_w : m_w \in M_6\}$                     | $\theta_{r3}$                         | $\{10, 5\}$   | 60 [s]     |
| 9           | $\mu$                                       | $\theta_{r1}, N_1$                    | 50            | 15 [s]     |
| 10          | $\{m_w : m_w \in M_7 \cup M_8\}$            | $\theta_{r3}$                         | $\{30, 15\}$  | 15 [s]     |
| 11          | $\mu$                                       | $\theta_{r2}, N_1$                    | 15            | 15 [s]     |

# Appendix D

## Simulink diagrams

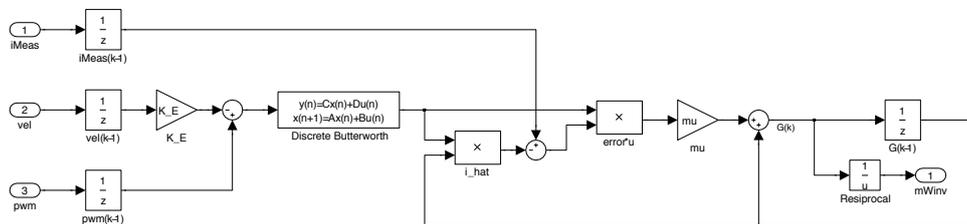


Figure D.1: Least mean squares parameter estimator.

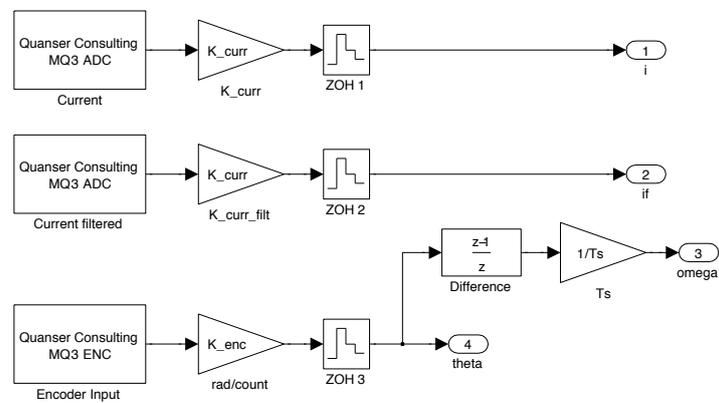


Figure D.2: Current and position input from MultIQ-3.

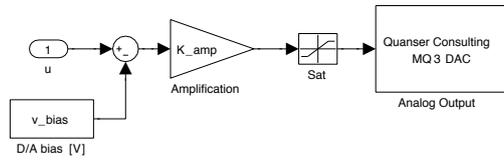


Figure D.3: Voltage output to MultitQ-3.

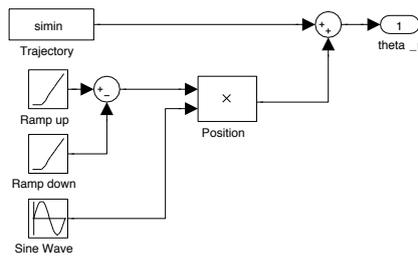


Figure D.4: Position reference generator.

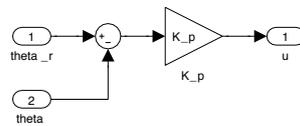


Figure D.5: Proportional controller for position.

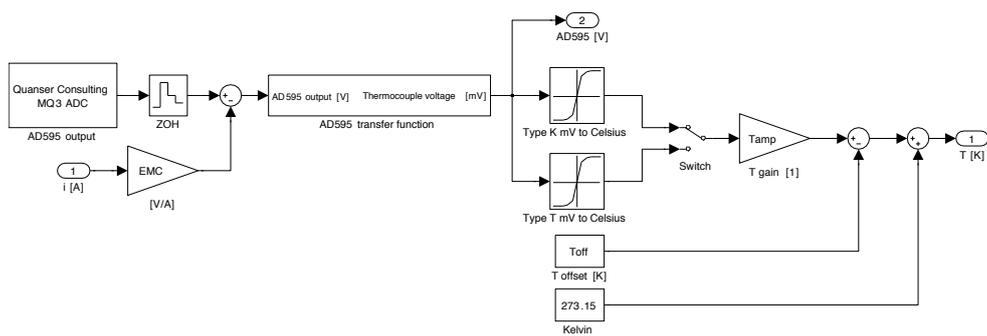


Figure D.6: Temperature measurement.

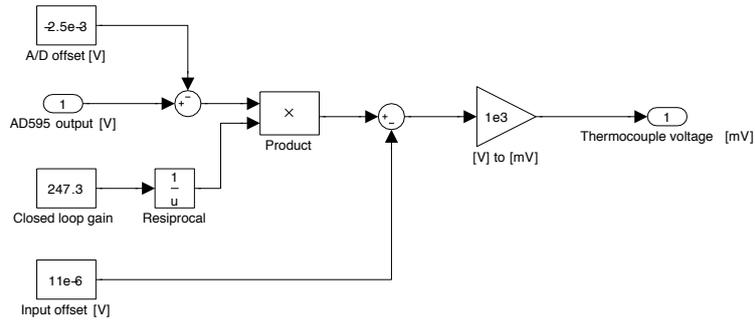


Figure D.7: AD595 transfer function.

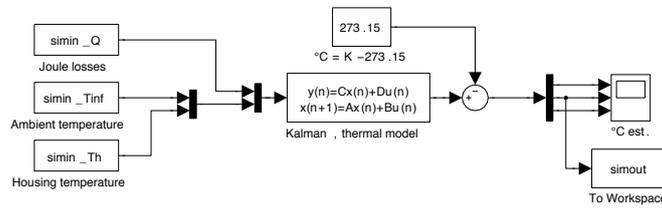


Figure D.8: Simulink diagram for offline reconstruction of thermal data.

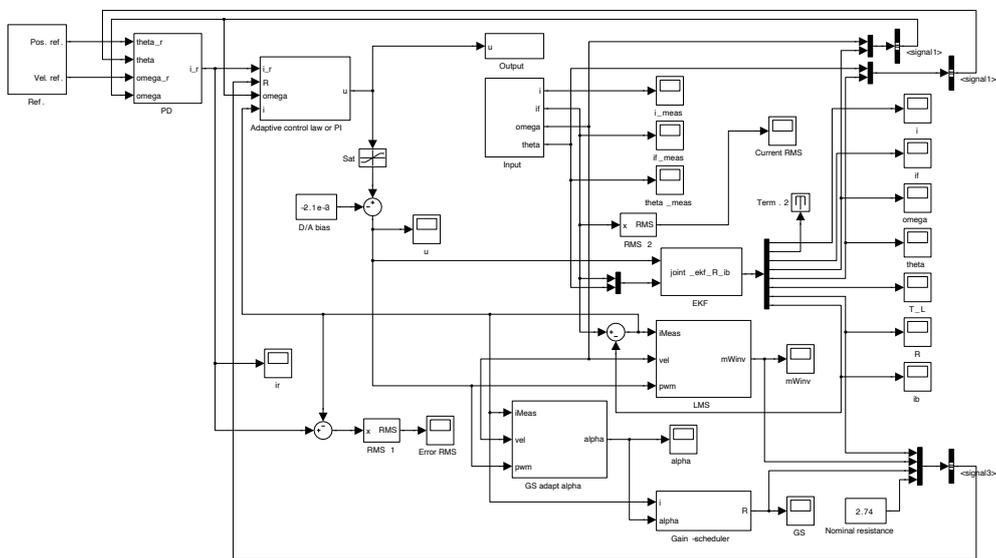


Figure D.9: Simulink diagram for comparison of adaptive schemes.

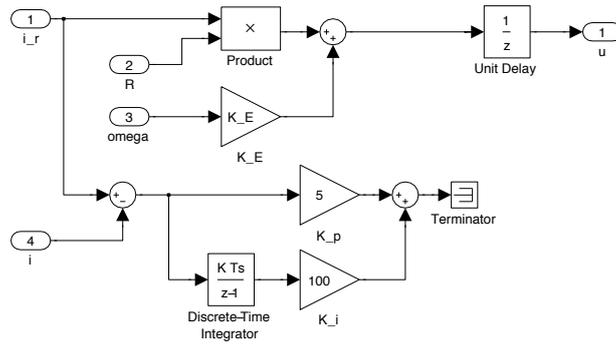


Figure D.10: Adaptive control law and PI current feedback regulator.

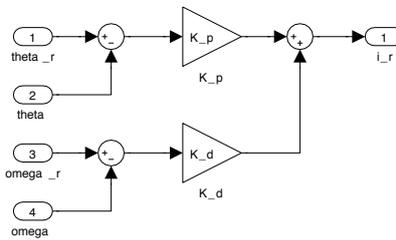


Figure D.11: PD regulator.

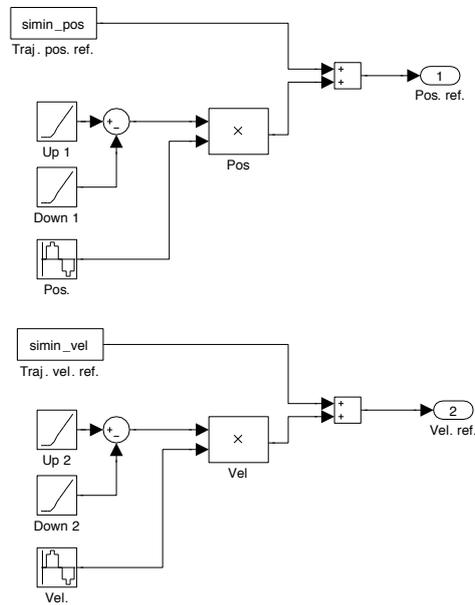


Figure D.12: Angular position and angular velocity reference input.

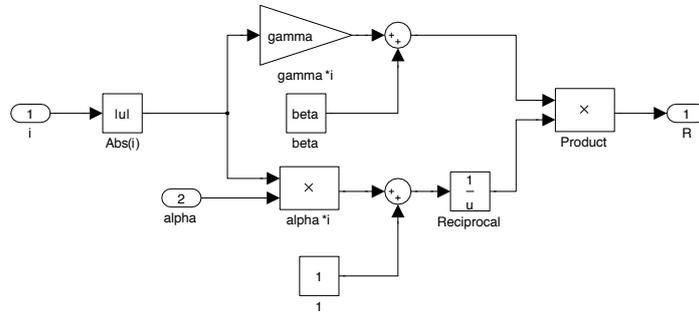


Figure D.13: Gain-scheduler.

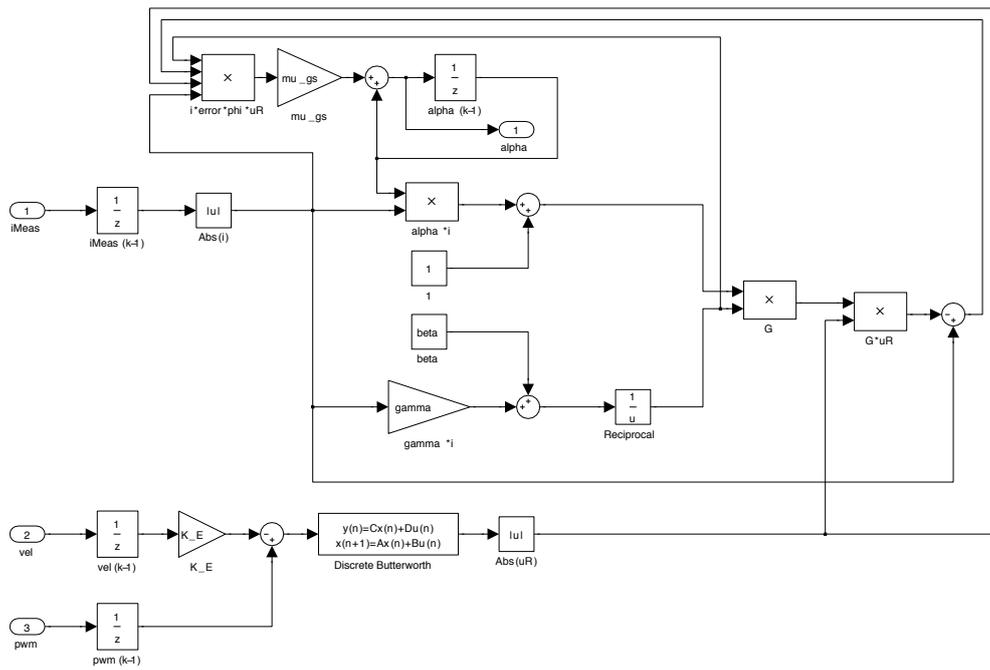


Figure D.14: Estimator for one of the gain-scheduler's parameters.