

Identification and Estimation of DC-Motor Dynamics for the Crazyflie Quadcopter

Erik Ottosson



LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6041
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2017 by Erik Ottosson. All rights reserved.
Printed in Sweden by Tryckeriet i E-huset
Lund 2017

Abstract

In this thesis methods for finding state space representation parameters for low inductance DC-motors are found and evaluated. Theory previously tested in simulations is verified experimentally and improved upon for both linear behaviour when the motor is controlled with a DC-source as well as nonlinear behaviour when controlled with pulse width modulation (PWM). For the linear behaviour a linear regression model with multiple regression sets proved satisfactory at identifying the motor dynamics. For the nonlinear behaviour caused by the PWM signal it was found that a Hammerstein model was sufficient to model the dynamics and several rotor speed estimators were tested and compared. Based on these results a design for a Crazyflie expansion board is suggested.

Popular Scientific Summary

To achieve agile flight of quadcopters it is important to control the rotor speed of their motors and make sure that they reach their desired values. This can be done by measuring the rotor speed and adapting the amount of current supplied to the motor thereafter. Measuring the rotor speed directly can be done in several ways and there are many sensors available for this task, for example optical encoders or hall sensors. For applications where these are not applicable, maybe because of restrictions on cost or weight, other approaches might be needed. In this thesis a different method was explored where the rotor speed is measured indirectly.

Instead of measuring the speed it is possible to measure the current being supplied to the motor and to use a simple linear model for the dynamics of a DC-motor, which describes the relation between rotation speed, voltage and current, to calculate the speed. This model used a certain number of parameters for which the values were unknown. To find the values of these parameters experiments were carried out where the rotor speed and current was measured for different voltage sequences. These data recordings was then used with so called linear regression, which is a method commonly used in statistics and for system identification, to find the values of the parameters. This method was proven to be feasible when the motor was supplied with a DC-source with variable amplitude.

The motors mounted on a quadcopter is however not supplied with a DC-source with variable amplitude but instead uses a fixed voltage source. The amount of current supplied to the motor is instead controlled by opening and closing a transistor so that the relation between the time it is open and the time it is closed is proportional to the desired current. When running the motor with this, so called pulse width modulation, it was discovered that the linear model previously used was not sufficient to describe the behaviour of the motor. It was also discovered that the current measuring became more difficult. The motors had very low inductance which means that the current changes very quickly giving us large amounts of noise. To handle this analog filters were used on the measurement signal as well as on the voltage supplied to the motor.

To capture this changed behaviour the motor model had to be expanded with a nonlinearity. The shape of this nonlinearity was found both by a trial and error

approach as well as a mathematical identification method. The later approach was deemed to give the best results while the trial and error method is simpler to implement on the actual hardware. Therefore it can be concluded that there are several ways to achieve satisfactory rotor speed estimation by measuring the motor current and that it is a feasible approach for applications where an indirect measurement method is needed.

Acknowledgements

I would like to thank my supervisor Professor Anders Robertsson for guidance and Ph.D. student Marcus Greiff for motivation and discussions. I would also like to thank Getachew Darge, research engineer at IEA at LTH, who helped me with lab work and Pontus Andersson who helped me with lab work and components. Finally I would like to thank Tobias Antonsson at Bitcraze AB who has been a great support in matters regarding the Crazyflie and design of the expansion board.

Contents

1. Introduction	11
1.1 Goals of Thesis	11
2. Crazyflie Platform	13
2.1 Motor Control	14
2.2 Expansion Port	15
3. Rotor Speed Estimation - DC-Input	17
3.1 Theory	17
3.2 DC-Input Experiment Setup	19
3.3 Identification	21
3.4 Estimator Evaluation	23
4. Rotor Speed Estimation - PWM-input	26
4.1 Pulse Width Modulation (PWM)	26
4.2 Hardware	28
4.3 Sensing Modeling	35
4.4 Measurements and Data Sets	39
4.5 Identification Methods	41
4.6 Results and Conclusion	47
4.7 Future Work	59
Bibliography	61
A. Expansion Board Designs	63
A.1 PCB without MCU	63
A.2 PCB with MCU	63
B. Landing Platform	67
B.1 Approximate Length Model	67
B.2 Complete Inverse Kinematics	68
B.3 Real Time Implementation	70

1

Introduction

During the greater part of 2016 Marcus Greiff, a student at LTH, worked on improving many parts of the Crazyflie platform which is a small size, open source quadcopter, further presented in chapter 2. His work covered several different topics ranging from dynamic modeling to state estimation and control. The results were well grounded in both theory and simulations as well as largely supported by real-time experiments.

One topic that was covered in theory and simulations but not experimentally was that of rotor control. The rotor speed control is presently implemented as an open-loop system and it was theoreticized that closing this loop would enable more aggressive flight.

This thesis therefore attempts to build further upon the work done by [Greiff, 2017] and to give experimental results from real-time implementations where results previously have been obtained from simulations.

Though the flight performance of the quadcopter was greatly improved, a topic not covered in the thesis of [Greiff, 2017] was that of landing. With the goal of enabling landing on moving surfaces a first step was taken to set up an experimental platform for future experiments, the results are covered in Appendix B.

1.1 Goals of Thesis

The thesis will be divided into two parts. The first is to develop the identification methods and measurement hardware needed to design an expansion board that will enable closed-loop control of the rotor speeds. The main goals are here to

- Verify identification simulation results of [Greiff, 2017].
- Implement rotor speed estimation.
- Design PCB for feedback loop measurements.

Chapter 1. Introduction

- Implement and compare control methods, mainly PID and model reference adaptive control (MRAC). (This goal was however not completed due to lack of time.)

The second focus is to make preparations for further work on autonomous landing on a moving surface.

- Program a moving platform to be able to follow angular and translational trajectories.
- Implement simple fly-over landing with laser height detector. (This topic was however not completed due to lack of time.)

2

Crazyflie Platform

The Crazyflie 2.0 is a small-size, low-weight quadcopter designed by the Swedish company Bitcraze AB. It is an open source project and is at the moment mainly used in research and education by several leading research labs around the world, for example MIT and ETH.

This chapter briefly presents details regarding the hardware and control structures already implemented on the Crazyflie that are the foundation upon which this thesis aims to improve. Many details are omitted and the interested reader is referred to [Greiff, 2017] and [Bitcraze, 2017a].



Figure 2.1 The Crazyflie 2.0 quadcopter. Picture obtained from [Bitcraze, 2017b].

2.1 Motor Control

Motor Hardware

The Crazyflie uses brushed coreless DC motors. The motors are connected to the battery where the connection to ground is controlled by one transistor per motor. These transistors are opened and closed by 328 kHz PWM-signals which has a duty cycle $d \in [0, 1]$, see Figure 2.2 for schematics.

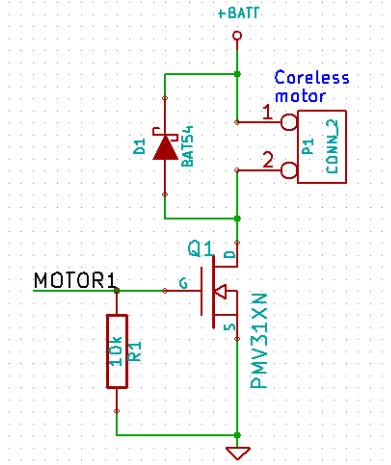


Figure 2.2 Schematics illustrating the wiring of one of the motors. MOTOR1 is the schematic name of the PWM output port of the microcontroller. Schematics obtained from [Bitcraze, 2017b].

Since the motor is not connected via an H bridge the rotors can only be run in one direction

Open Loop Control

The dynamics of the quadcopter is described by [Greiff, 2017] as a nonlinear MIMO state space model with 13 states and 4 input signals where the states represent the position, speed and rotation of the quadcopter and where the inputs, \mathbf{u} , are thrust, T , and torques, $\boldsymbol{\tau}$, formulated as the vector

$$\mathbf{u} = \begin{bmatrix} T \\ \boldsymbol{\tau} \end{bmatrix}. \quad (2.1)$$

Since the motors are controlled by setting the duty cycle, not a desired thrust or torque, a transformation from \mathbf{u} to d has to be obtained.

The thrust generated by each motor is assumed to be proportional to the rotor speed squared, $\dot{\theta}^2$, and the total thrust T to be the sum of the separate motor thrusts

$$T = \sum_{i=1}^4 k \dot{\theta}_i^2. \quad (2.2)$$

The torque generated by each motor is approximated as

$$\tau_i \approx b \dot{\theta}_i^2 + I \ddot{\theta} \quad (2.3)$$

where b is a drag constant and I is the rotor inertia. The parameter identification was carried out for the unknown parameters and the map $M_{\dot{\theta}^2 \rightarrow u}$ from rotor speed to control signals was defined as

$$[T \quad \tau]^T = M_{\dot{\theta}^2 \rightarrow u} \dot{\theta}^2 \quad (2.4)$$

A map $M_{d \rightarrow T}$ from duty cycle to thrust was also obtained experimentally as a second order polynomial.

Knowing these mappings the desired duty cycle can be calculated as

$$d = M_{d \rightarrow T}^{-1}(M_{\dot{\theta}^2 \rightarrow T}(M_{\dot{\theta}^2 \rightarrow u}^{-1}([T \quad \tau]^T))). \quad (2.5)$$

This also means that the rotor speed is estimated as a function of the duty cycle and is given as

$$\dot{\theta} = \sqrt{M_{\dot{\theta}^2 \rightarrow T}^{-1}(M_{d \rightarrow T} d)} \quad (2.6)$$

where $\dot{\theta} \geq 0$ since, as previously mentioned, the motors are only run in one direction.

2.2 Expansion Port

The Crazyflie is equipped with 20 input/output pins that can be used for customizing the quadcopter with extra hardware. This is usually done by adding an expansion board. The functionality of the expansions ports ranges from IO pins to communication and power supply. An illustration of pin functionality can be seen in Figure 2.3.

Crazyflie expansion port

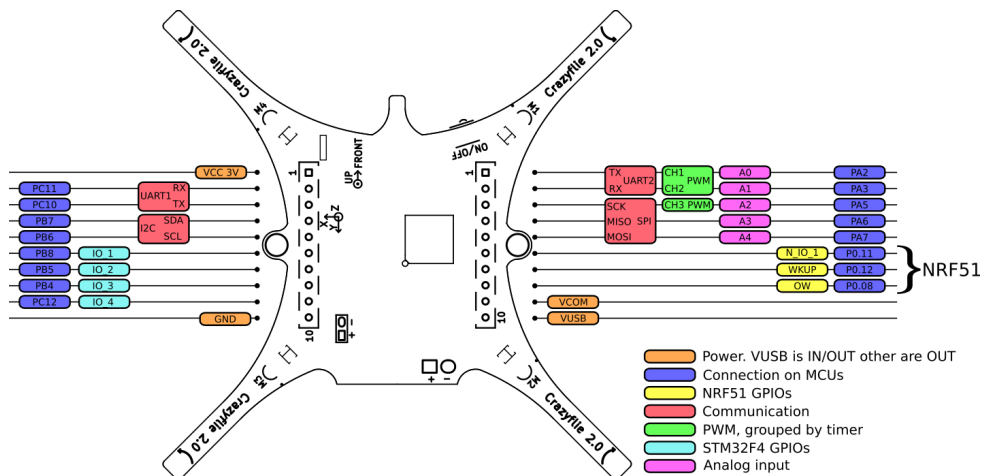


Figure 2.3 Expansion port pin functionalities. Figure obtained from [Bitcraze, 2017b].

In this thesis the analog input pins were the ones of main interest as well as those enabling SPI communication, as described in Appendix B.

3

Rotor Speed Estimation - DC-Input

In this chapter the motor model and parameter identification method used for the rotor speed estimation and closed loop control previously tested in simulations by [Greiff, 2017] is presented.

The validity of the method is investigated by applying the same methods experimentally. Improvements for the parameter identification are thereafter suggested.

3.1 Theory

The modeling and parameter identification of a brushed DC motor is covered by [Greiff, 2017] but will be restated here for readability and a slightly changed notation.

Rotor Model

We assume that the torque on the rotor is proportional to the current $i(t)$ and that there is viscous friction with the parameter b . Newton's second law of motion then gives us

$$J\ddot{\theta}(t) = T - b\dot{\theta}(t) \leftrightarrow \ddot{\theta}(t) = \frac{1}{J}(K_i i(t) - b\dot{\theta}(t)) \quad (3.1)$$

The inertia J was found to be nonlinear and dependent on whether $|\dot{\theta}|$ is increasing or decreasing so that it can be described by

$$J = \begin{cases} J^+ & \text{if } \text{sgn}(\dot{\theta}\ddot{\theta}) > 0 \\ J^- & \text{if } \text{sgn}(\dot{\theta}\ddot{\theta}) < 0 \end{cases} \quad (3.2)$$

By using Kirchoff's law we can model the relations between the current, voltage $V(t)$ and rotor speed $\dot{\theta}(t)$ in the motor if we assume the electro-motive force $e(t)$ to

be proportional to the rotor speed by a constant K_e and assume the motor to have an inductance L and a resistance R .

$$L \frac{di(t)}{dt} = -Ri(t) + V(t) - e(t) \Leftrightarrow \frac{di(t)}{dt} = \frac{1}{L}(-Ri(t) + V(t) - K_e \dot{\theta}(t)) \quad (3.3)$$

By disregarding the previously described mechanical nonlinearity and approximating it as linear these continuous differential equations can be discretised and put together into the discrete time state space form

$$x = \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} \quad x_{k+1} = \Phi x_k + \Gamma u_k \quad \Phi = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \quad \Gamma = \begin{bmatrix} \gamma_{11} \\ \gamma_{21} \end{bmatrix} \quad (3.4)$$

which will have the transfer functions

$$\begin{bmatrix} H_{V \rightarrow \dot{\theta}}(z) \\ H_{V \rightarrow i}(z) \end{bmatrix} = I(Iz - A)^{-1}B = \frac{1}{(z - \phi_{11})(z - \phi_{22}) - \phi_{12}\phi_{21}} \begin{bmatrix} (z - \phi_{22}\gamma_{11}) + \phi_{12}\gamma_{21} \\ \phi_{21}\gamma_{11} + (z - \phi_{11})\gamma_{21} \end{bmatrix} \quad (3.5)$$

These transfer functions can be expressed in a general form as:

$$H_{u \rightarrow y}(z) = \frac{B(z)}{A(z)}, \quad A(z) = z^2 + a_1z + a_0, \quad B(z) = b_1z + b_0 \quad (3.6)$$

for which a linear regression model is

$$y_k = \varphi_k^T \theta + e_k \quad (3.7)$$

with the regressor and the parameter vector

$$\varphi_k^T = [-y_{k-1} \quad -y_{k-2} \quad u_{k-1} \quad u_{k-2}], \quad \theta = [a_1 \quad a_0 \quad b_1 \quad b_0] \quad (3.8)$$

By calculating two different linear regressions, one for the rotor speed and one for the current, the transfer functions in Equation 3.5 are identified. It is assumed that the regression results in approximately equal values for a_1 and a_0 for both transfer functions. Once these transfer function parameters have been identified the original model parameters in Equation 3.4 can be calculated as

$$\begin{cases} \phi_{11} = -a_1 - \phi_{22} \\ \phi_{12} = -\frac{a_0 + (a_1 + \phi_{22})\phi_{22}}{\phi_{21}} \\ \phi_{21} = \frac{b_0^i - (a_1 + \phi_{22})\gamma_{21}}{\gamma_{11}} \\ \phi_{22} = \frac{a_0\gamma_{11}\gamma_{21} + b_0^{\dot{\theta}}b_0^i - b_0^{\dot{\theta}}a_0\gamma_{21}}{b_0^{\dot{\theta}}\gamma_{21} - b_0^i\gamma_{11}} \end{cases}, \quad \begin{cases} \gamma_{11} = b_1^{\dot{\theta}} \\ \gamma_{21} = b_1^i \end{cases} \quad (3.9)$$

With the state space model parameters found a Kalman filter can be implemented that uses current measurements as input to estimate the rotor speed.

Linear Model Correlation Coefficient

To check the feasibility of a linear static model an analysis can be done of the correlation between input and output [Johansson, 2011]. For a finite discrete-time measurement series this can be done with the sample correlation coefficient

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3.10)$$

For a linear static system a correlation coefficient close to 1 should be obtained.

3.2 DC-Input Experiment Setup

The methods presented above were tested on a separate setup where a crazyflie motor was connected to a DC power source with a variable DC amplitude.

To measure the current, a 0.5Ω resistor was placed in series with the motor, between the motor and the battery. Measuring the voltage across this so called sense resistor gives us the current by Ohm's law. The placement of the sense resistor means that it is a high side current sensing. Therefore a differential amplifier has to be used [Regan et al., 2005], even though we do not necessarily want to amplify the measurement.

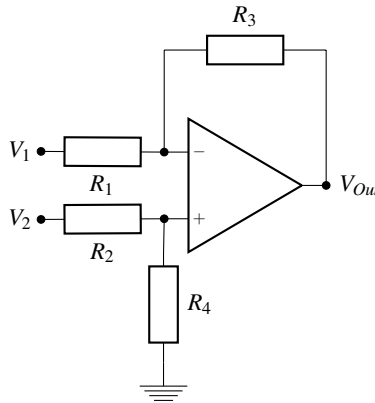


Figure 3.1 Unit gain differential amplifier where $R_1 = R_2 = R_3 = R_4 = 1M\Omega$ resistances. The op-amp used was a CA3140.

A general differential amplifier can be made of resistors and an op-amp as illustrated in Figure 3.1. The gain is determined from the resistances and the output voltage V_{out} is given by

$$V_{out} = -V_1\left(\frac{R_3}{R_1}\right) + V_2\left(\frac{R_4}{R_2 + R_4}\right)\left(\frac{R_1 + R_3}{R_1}\right) \quad (3.11)$$

A unit gain amplifier is obtained when all the resistors have equal resistances.

$$V_{out} = V_2 - V_1 \quad (3.12)$$

When Out+ is connected to V_2 and Out- is connected to V_1 , V_{out} will equal the filtered voltage drop across the motor but relative to ground and is connected to one of the analog inputs on the expansion port.

When the end of the resistor that is connected to the battery is connected to V_2 and the end that is connected to the motor is connected to V_1 , V_{out} will equal the voltage across the resistor. The current can then be obtained with Ohm's law.

An optical switch is used for RPM measurements and is connected to a digital input on the crazyflie. An interrupt is triggered and a timer is read each time the switch goes from low to high. The time difference between two consecutive reads is obtained from which the rotor speed can be calculated.

Measurements

A measurement sequence was carried out where the input voltage from the DC-source was varied in a sinusoidal-like way with different frequencies and amplitudes and rotor speed and current were logged with a frequency of 50 Hz. See Figure 3.2 for result plots.

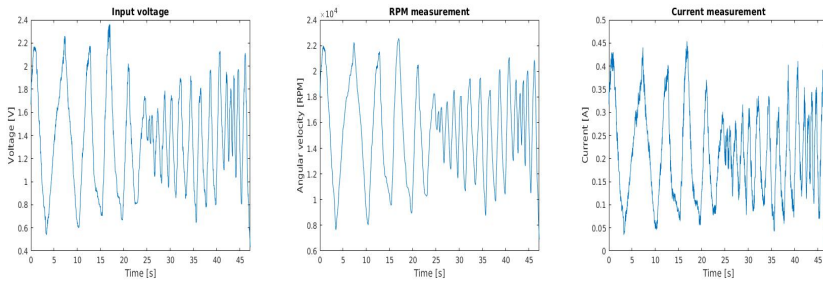


Figure 3.2 Left: Input voltage sequence. Center: Measured RPM. Right: Measured current.

The correlation coefficients, given by equation 3.10, for these sequences are $r_{u\theta} = 0.979$ and $r_{uI} = 0.981$ which indicate linear relations.

3.3 Identification

With the obtained data sets the identification methods described earlier could be used on the data sets seen in Figure 3.2 and the following parameters were acquired

Table 3.1 Identified parameter values using the technique described by [Greiff, 2017].

a_1^I	a_0^I	$a_1^{\hat{\theta}}$	$a_0^{\hat{\theta}}$	b_1^I	b_0^I	$b_1^{\hat{\theta}}$	$b_0^{\hat{\theta}}$
-0.7628	-0.1780	-1.8028	0.8042	0.2173	-0.2081	1314.6	-1299.5

Because the resulting values of a_1 and a_0 are different it is problematic to use these values in the state space form. An improved identification method is needed. To solve this issue two different methods were tried. First a brute force, parameter sweep method and then an improved regression model.

Parameter sweep (PS)

The transfer function $H_{V \rightarrow \hat{\theta}}(z)$ is identified with linear regression as previously described. The transfer function $H_{V \rightarrow I}(z)$ is then constructed by using the denominator polynomial of $H_{V \rightarrow \hat{\theta}}(z)$ and the numerator parameters, b_0^I and b_1^I , are found experimentally in simulations.

The numerator values are swept and each combination and its resulting transfer function is tested in simulink. The simulated response is compared with the measured and the coefficient combination that has the least squared difference between simulated and measured values is chosen.

Multiple Linear Regression (MLR)

As stated by [Ljung, 1987] it is possible to combine different regressor sets for different components of y . The θ that minimizes the LS criterion is given by

$$\hat{\theta} = \left[\frac{1}{N} \sum_{k=1}^N \boldsymbol{\varphi}(k) \Lambda^{-1} \boldsymbol{\varphi}(k)^T \right]^{-1} \left[\frac{1}{N} \sum_{k=1}^N \boldsymbol{\varphi}(k) \Lambda^{-1} \boldsymbol{\varphi}(k)^T \right] \quad (3.13)$$

where Λ can be used to give different weights to different components of y . Here set to the identity matrix.

Since we have two different regressor sets, one for the RPM, $\hat{\theta}$, and one for the current, I , we have the regressor matrix

$$\boldsymbol{\varphi}(k) = [\boldsymbol{\varphi}_{\hat{\theta}}(k) \quad \boldsymbol{\varphi}_I(k)] \quad (3.14)$$

where the separate regressor vectors for

$$\varphi_{\hat{\theta}}^T = [-y_{\hat{\theta}}(k-1) \quad -y_{\hat{\theta}}(k-2) \quad u(k-1) \quad u(k-2) \quad 0 \quad 0] \quad (3.15)$$

and

$$\varphi_I^T = [-y_I(k-1) \quad -y_I(k-2) \quad 0 \quad 0 \quad u(k-1) \quad u(k-2)] \quad (3.16)$$

are combined according to the parameter vector

$$\theta = [a_1 \quad a_0 \quad b_1^{\hat{\theta}} \quad b_0^{\hat{\theta}} \quad b_1^I \quad b_0^I]^T \quad (3.17)$$

The expression in the left parentheses of Equation 3.13 can in our case, since we have uniform weighting, then be further simplified as

$$\frac{1}{N} \sum_{k=1}^N [\varphi_{\hat{\theta}}(k) \quad \varphi_I(k)] \begin{bmatrix} \varphi_{\hat{\theta}}^T(k) \\ \varphi_I^T(k) \end{bmatrix} = \frac{1}{N} \sum_{k=1}^N (\varphi_{\hat{\theta}}(k) \varphi_{\hat{\theta}}^T(k) + \varphi_I(k) \varphi_I^T(k)) = \frac{1}{N} (\Phi_{\hat{\theta}}^T \Phi_{\hat{\theta}} + \Phi_I^T \Phi_I) \quad (3.18)$$

and a similar simplification is done on the expression in the right parentheses

$$\frac{1}{N} \sum_{k=1}^N [\varphi_{\hat{\theta}}(k) \quad \varphi_I(k)] \begin{bmatrix} y_{\hat{\theta}}(k) \\ y_I(k) \end{bmatrix} = \frac{1}{N} \sum_{k=1}^N (\varphi_{\hat{\theta}}(k) y_{\hat{\theta}}(k) + \varphi_I(k) y_I(k)) = \frac{1}{N} (\Phi_{\hat{\theta}}^T Y_{\hat{\theta}} + \Phi_I^T Y_I) \quad (3.19)$$

where

$$\Phi_{\hat{\theta}} = \begin{bmatrix} \varphi_{\hat{\theta}}^T(1) \\ \varphi_{\hat{\theta}}^T(2) \\ \vdots \\ \varphi_{\hat{\theta}}^T(N) \end{bmatrix}, \Phi_I = \begin{bmatrix} \varphi_I^T(1) \\ \varphi_I^T(2) \\ \vdots \\ \varphi_I^T(N) \end{bmatrix} \quad (3.20)$$

This means that the original least squares parameter estimation, Equation 3.13, can be calculated as

$$\hat{\theta} = (\Phi_{\hat{\theta}}^T \Phi_{\hat{\theta}} + \Phi_I^T \Phi_I)^{-1} (\Phi_{\hat{\theta}}^T Y_{\hat{\theta}} + \Phi_I^T Y_I) \quad (3.21)$$

3.4 Estimator Evaluation

Using the two methods described above, parameter sweep and multiple linear regression, we get the parameters presented in Table 3.2.

Table 3.2 Identified parameter values.

	a_1	a_0	$b_1^{\hat{\theta}}$	$b_0^{\hat{\theta}}$	b_1^I	b_0^I
PS	-1.7297	0.7324	0.1960	-0.1931	0.2078	-0.2062
MLR	-0.8954	-0.0540	0.5920	-0.5153	0.6440	-0.6182

These parameter values give the following open loop RPM responses, given by the transfer function $H_{V \rightarrow \hat{\theta}}(z)$, as shown in Figure 3.3.

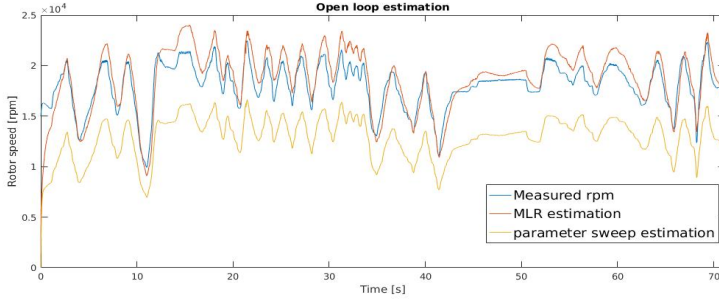


Figure 3.3 Open loop responses with parameters from parameter sweep and MLR.

Using the estimated parameters, discrete state space coefficients are calculated with Equation 3.9 and then implemented in a standard Kalman filter, as formulated by [Åström and Wittenmark, 2011], where the current is used as the measurement signal. $P(0)$ is chosen as the identity matrix and appropriate covariance matrices were found by trial and error. These observers are then tested with the test sequence and the results are displayed in Figure 3.4.

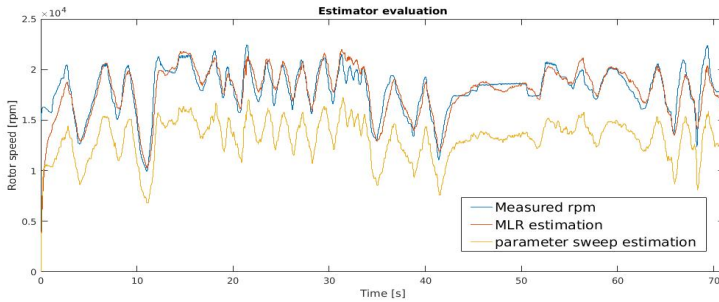


Figure 3.4 Results for the Kalman filter implementation of the system obtained with MLR and PS.

The estimated RPM was then compared with the measured and the size of the error was computed with the l^2 -norm

$$E = \sqrt{\sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (3.22)$$

The estimator errors are compared and their relative values are displayed in Table 3.3 where the method with the least error has been chosen as the reference value.

Table 3.3 Error norm relative values where the smallest error size, the result of the MLR Kalman, is chosen as reference.

<i>MLR Kalman</i>	<i>PS Kalman</i>	<i>MLR open</i>	<i>PS open</i>
1.0	5.622	1.490	5.80

It is now clear that the Kalman filter implementation of the MLR parameters results in the estimator that produces the smallest error.

Conclusion

One could question the reason for the greater static error of the parameter sweep approach. The difference in the method results can probably be attributed to the fact that the sweep does not have a high enough resolution when choosing parameter values. The optimal value is therefore missed.

Using two separate regression calculations for the transfer functions, $H_{V \rightarrow \dot{\theta}}(z)$ and $H_{V \rightarrow I}(z)$, results in denominator parameters that differ too much and therefore do not fulfill the requirements for calculating the state space parameters with Equation 3.9.

By using methods that force the denominator parameters to the same value this problem is avoided. A Kalman filter estimator based on the results from the multiple linear regression proved to give the best results followed by the open loop implementation of the same parameters.

4

Rotor Speed Estimation - PWM-input

This chapter covers the process of implementing the estimator described in the previous chapter. It was discovered that the motor exhibits nonlinear behaviour when controlled with PWM and therefore several estimation methods, based on either a direct mapping between measurement and RPM or a Hammerstein model, were tested and evaluated. It was also discovered that measuring the current is more difficult when using PWM instead of a DC-source. To handle this, three different hardware configurations for feedback measurements will be presented as well as filter designs for the measurement signal, .

4.1 Pulse Width Modulation (PWM)

When controlling a DC-motor with PWM we expect the dynamics of the motor to act as a low-pass filter and therefore smooth out the current so that most of the AC-components are removed and the DC-component remains. The concept is illustrated in Figure 4.1. If this is the case we should see that the results from the previous chapter should hold even though the input signal is changed from a DC-power source with variable amplitude to a PWM source with variable duty cycle.

PWM low pass filtering illustration

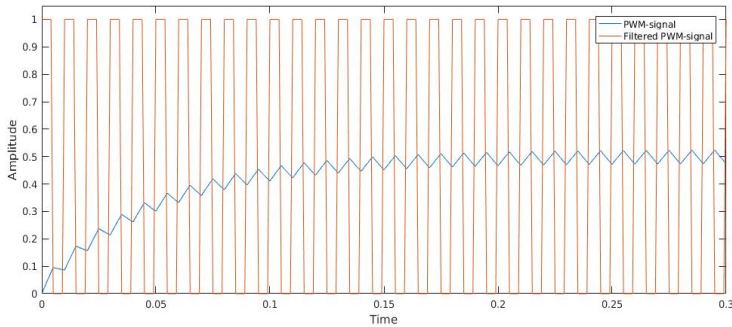


Figure 4.1 Illustration of low pass filtering of a pwm signal with a 50% duty cycle. Most of the periodic components are removed and the filtered signal has an average proportional to the duty cycle.

To verify the low pass filter characteristics of the motor a duty cycle of 0.5 was set and the voltage across a current sense resistor was measured with an oscilloscope and the result can be seen in Figure 4.2.

Motor current at 50 % duty cycle

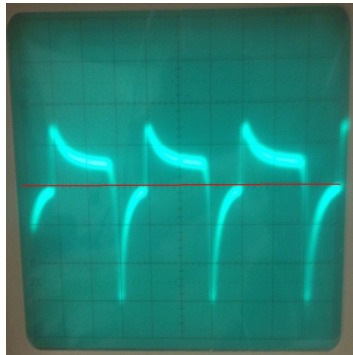


Figure 4.2 Measured motor current when a 50% duty cycle was applied. The lack of smoothing indicates a low inductance in the motor. Ground level is marked with a red line.

From Figure 4.2 it is clear that the motor does not have the desired low pass characteristic. This is not entirely unexpected since low inductance is characteristic

for coreless motors [MachineDesign, 2002]. This does not only influence the dynamics of the motor but it also makes it more problematic to measure the desired feedback variable. The differential amplifier used previously has its own dynamics and distorts different frequencies of the measurement signal unequally. It is therefore needed to filter the signal before amplifying it.

4.2 Hardware

Measurement Methods and Filter Design

Just as in the previous chapter where a DC-input was used, a viable measurement method is to measure the voltage across a sense resistor. However, it might also be possible to measure the motor voltage and use it as feedback variable since the low inductance of the motor should make it approximately proportional to the current. For both of these measurement approaches it is necessary to filter the signal before putting it to the amplifier because of the previously mentioned distortions. A low-pass RLC-filter was designed for this purpose and the details are covered below.

For the current sense approach it is difficult to design a filter that is aggressive enough to completely remove AC-components. The measurements can be significantly improved by also filtering the PWM-signal applied to the motor. This is usually done by connecting an inductor in series with the motor and a capacitor in parallel. The schematics for the complete setup with measurement signal filter and PWM-filter is illustrated by Figure 4.3.

An inductor with high enough inductance and low winding resistance was not found. The current sense configuration that was tested therefore only used a $100\mu F$ capacitor.

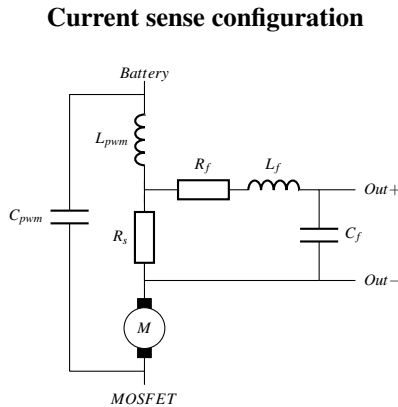


Figure 4.3 Current sense measurement setup. The Out-pins are connected to the amplifier.

When measuring the motor voltage it is not necessary to use a PWM-filter to get good measurements. Adding a capacitor might still be a good idea though since the changed dynamics might be easier to identify. It was also found that adding a capacitor increases the rotor speed. Two different configurations were therefore tried, one without and one with a $100\ \mu\text{F}$ capacitor.

Motor voltage measuring configurations

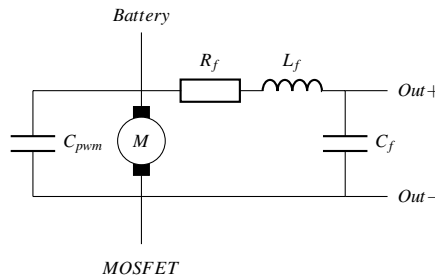


Figure 4.4 Motor voltage measurement setup. The Out-pins are connected to the amplifier.

To find parameters for the RLC-filter used for the measurements, calculations and simulations were done in Matlab and Simscape. Two feasible filters were tested on the motor voltage configuration without a PWM-filtering capacitor, Figure 4.4 with $C_{pwm} = 0$.

When testing the filters it was discovered that because of the poor performance of the OP-amp at high frequencies the PWM-frequency had to be lowered to 20 kHz from the normal frequency of 328 kHz.

Filter 1

When filtering the measurement signal the goal is to remove the AC-components, leaving the DC-component. Since the measured signal is periodic, as illustrated by Figure 4.2, the first DC-component, called the fundamental harmonic, should have the same frequency as the PWM-frequency [Smith, 1997], in our case 20 kHz, which means that a cutoff frequency below this value is desired.

The first filter that was tried had the following parameters: $R = 30\ \Omega$, $L = 2.2\ \text{mH}$, $C = 10\ \mu\text{F}$. These gave a promising cutoff frequency, as can be seen in Figure 4.5, and the Simscape simulations gave satisfactory results.

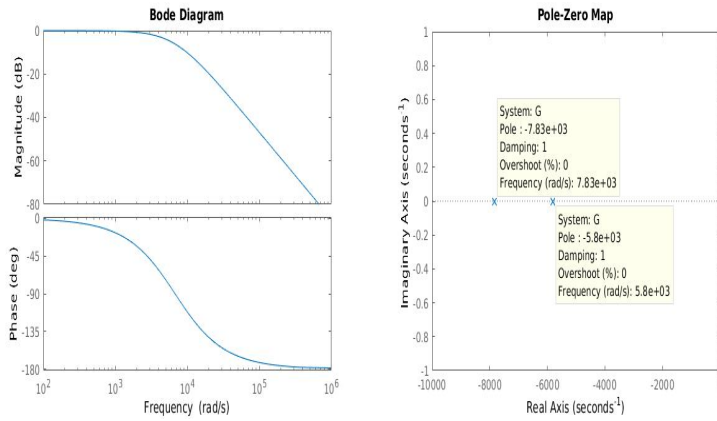


Figure 4.5 Bode plot and pole placement for RLC-filter 1. Poles located at -5800 rad/s and -7830 rad/s.

When tried on the real hardware the following results were obtained:

Filter 1 oscilloscope measurements

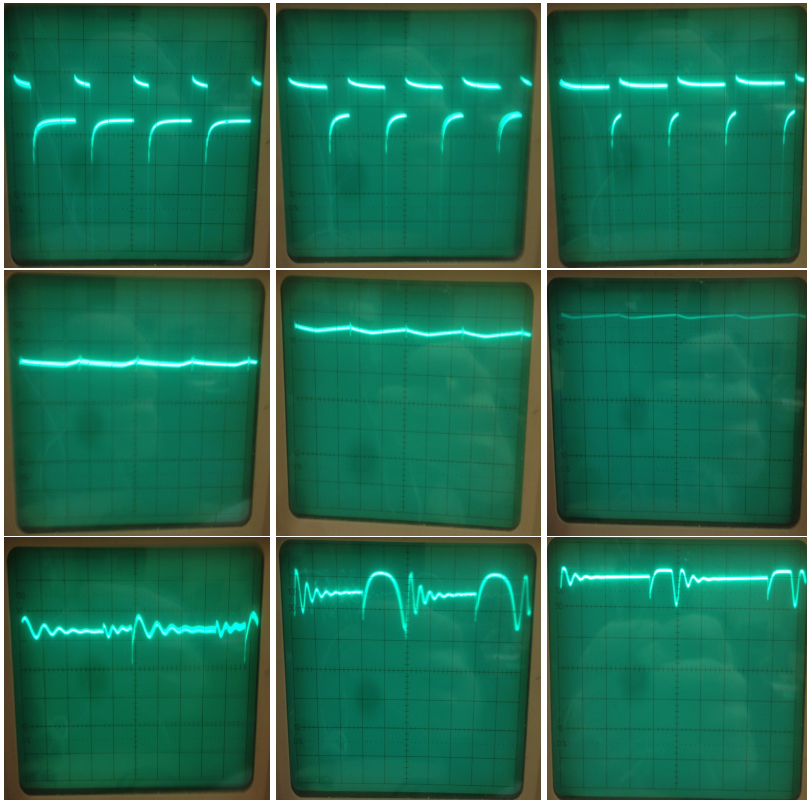


Figure 4.6 *Top row:* Voltage drop between BAT+ and Out-. *Middle row:* Voltage drop between Out+ and Out-. *Bottom row:* Voltage drop between V_{out} and ground. *Left column:* Duty cycle = 20%. *Middle column:* Duty cycle = 60%. *Right column:* Duty cycle = 100%. First row: X-axis: 20 μ s/div, Y-axis: 2 V/div. Second row: X-axis: 20 μ s/div, Y-axis: 1 V/div. Third row: X-axis: 10 μ s/div, Y-axis: 1 V/div.

We can see that the filter does not succeed in averaging the motor voltage drop into a constant value. The oscillative behaviour of the filtered signal is then further distorted by the amplifier. This result does not fulfill the requirements so a second filter was tried.

Filter 2

The aim of the second filter was to increase the impedance compared with the first one to evaluate if this would improve its performance. The following filter parameters were chosen: $R = 33k\Omega$, $L = 6.8mH$, $C = 10\mu F$, and the corresponding bode plot and pole placement can be seen in Figure 4.7. Compared with the first filter this pole placement should act more like a first order low-pass filter which has a lower cutoff frequency.

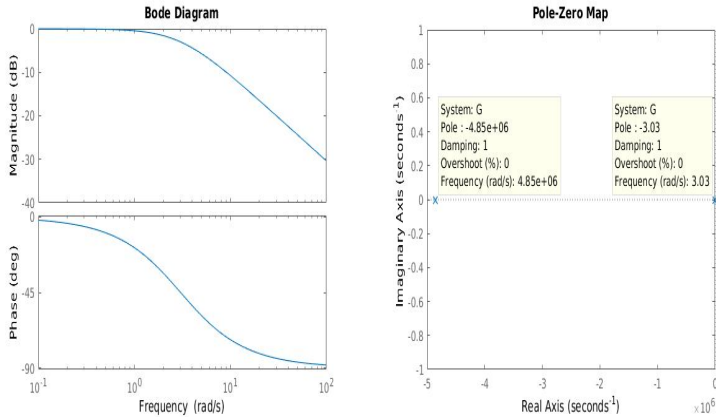


Figure 4.7 Bode plot and pole placement for RLC-filter 2. Poles placed at - 4850000 rad/s and -3.03 rad/s.

The filter was tried and the results were first measured with an oscilloscope and can be seen in the following table of figures.

Filter 2 oscilloscope measurements

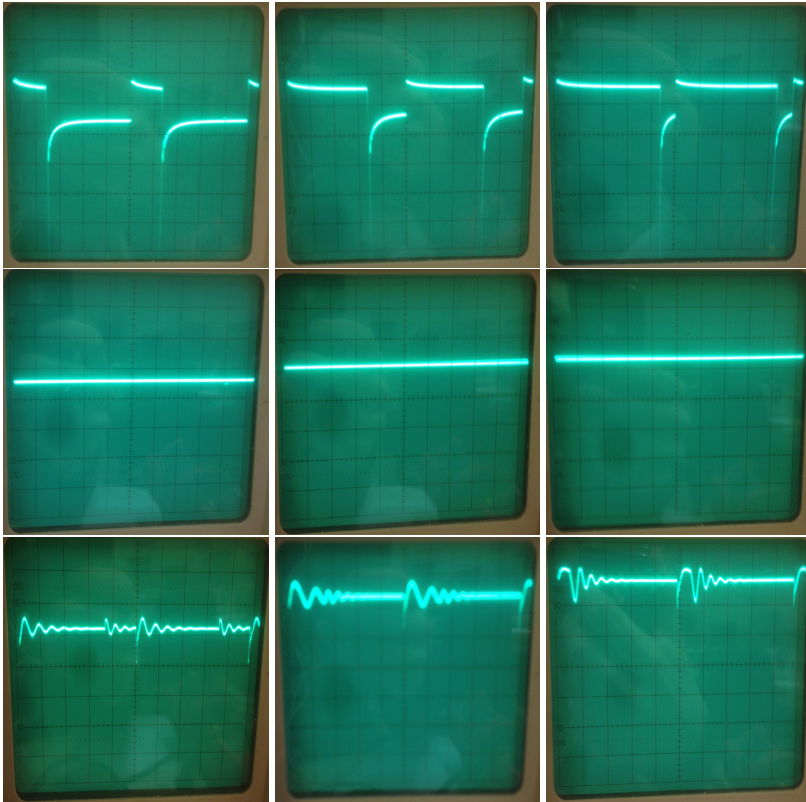


Figure 4.8 *Top row:* Voltage drop between BAT+ and Out-. *Middle row:* Voltage drop between Out+ and Out-. *Bottom row:* Voltage drop between V_{out} and ground. *Left column:* Duty cycle = 20%. *Middle column:* Duty cycle = 60%. *Right column:* Duty cycle = 100%. X-axis: 10 μ s/div, Y-axis: 2 V/div. Last row 1 V/div.

Since the filtered voltage before the amplifier seemed to be constant and increasing with an increased duty cycle the filter was deemed to be satisfactory. There are however still transients present, as can be seen in the bottom row, where they have been distorted by the amplifier. This distortion is further reduced by placing a capacitor at the amplifier output, as is illustrated in Figure 4.9.

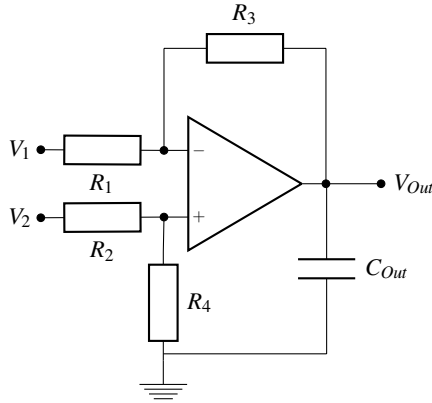


Figure 4.9 Unit gain differential amplifier with $1M\Omega$ resistances. The op-amp used was a CA3140. The output signal is filtered with a $10\ \mu\text{F}$ capacitor.

Measured Voltage to Current Mapping

To verify that the filter and amplifier also works for the current sense measurements we have to verify that the measured voltage across the sense resistor is proportional to the current. The motor is therefore replaced with a strictly resistive load and the setup is connected to a signal generator, providing the voltage $V(t)$, as illustrated in Figure 4.10.

Sensor linearity evaluation setup

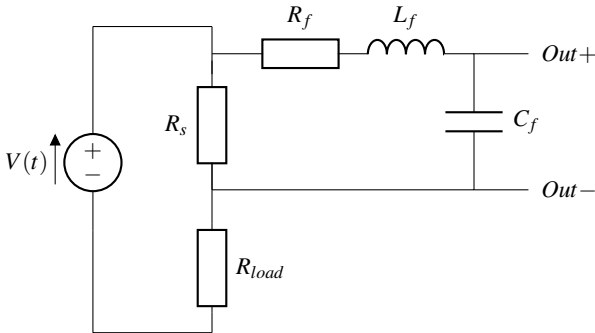


Figure 4.10 Schematics for experiment set up to find measured voltage to current mapping. The *Out*-ports were connected to the previously described amplifier. $R_s = 0.6\ \Omega$ and $R_{load} = 5\ \Omega$.

The sensor was tested with a ramp sequence and the amplifier output was

logged. The result can be seen in Figure 4.11.

Sensor linearity evaluation results

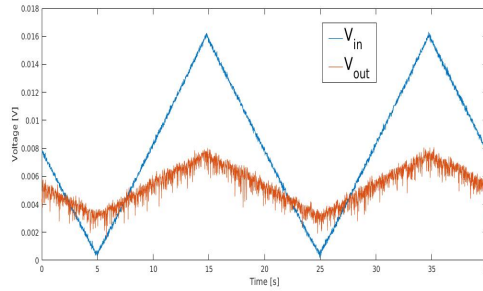


Figure 4.11 The input voltage, blue line, measured voltage, orange line, used to find the $y \rightarrow I$ map.

From this figure it seems that the relation between input u_k and output y_k is linear but with an offset which can be described by the map

$$M_{u_k \rightarrow y_k}(u_k) = a_1 u_k + a_0 \quad (4.1)$$

Therefore we try a linear regression model with the parameter vectors

$$\rho_k^T = [u_k \quad 1], \theta = [a_1 \quad a_0] \quad (4.2)$$

for which the following model parameters were obtained.

$$\begin{array}{c|c} a_1 & a_0 \\ \hline 0.3958 & 0.003 \end{array}$$

The current can now be calculated with the inverse mapping, $M_{u_k \rightarrow y_k}^{-1}$.

4.3 Sensing Modeling

Now that a capacitor has been added in parallel to the motor the model has to be updated. The motor is modeled with the standard dc-motor model presented in the previous chapter where the inductance and resistance of the filter and sensing resistor have been added to the motor parameters and where a capacitor is connected in parallel to the motor. A similar system model has previously been described by [Lyshevski, 1998]. The voltage source has an internal resistance to capture its non ideal behaviour. The circuit is illustrated in Figure 4.12.

Circuit diagram for motor model with PWM-filter

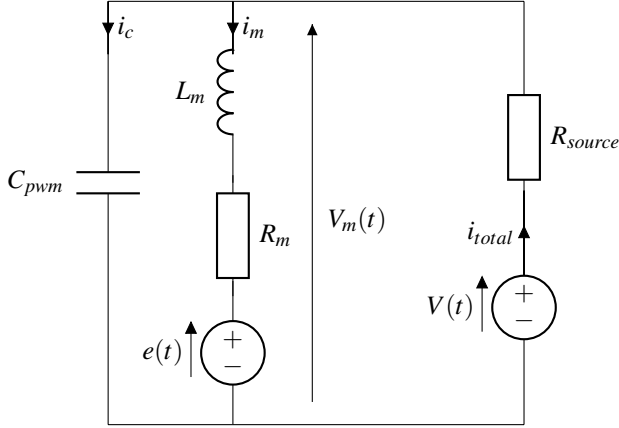


Figure 4.12 Equivalent circuit of the motor model with a PWM-filter and a non-ideal current source.

L_m is the sum of the motor's internal inductance and the added inductance L_{pwm} , R_m is the sum of the motor's internal resistance, R_s and the resistance of the added inductor L_{pwm} .

The mechanical behaviour of the motor is captured by Newton's law as described by equation 3.1 and the motor current dynamics are described as in equation 3.3 but with changed parameters because of the PWM filter.

$$L\dot{i}_m(t) = -Ri_m(t) + V_m(t) - e(t) \leftrightarrow \dot{i}_m(t) = \frac{1}{L}(-Ri_m(t) + V_m(t) - K_e\dot{\theta}(t)) \quad (4.3)$$

The current across the capacitor is given by

$$i_c(t) = C\frac{dV_m}{dt} \quad (4.4)$$

which can be substituted for the difference between the total current supplied by the battery and the motor current:

$$i_c(t) = i_{total}(t) - i_m(t) \quad (4.5)$$

The total voltage supplied will be equal to the sum of the motor voltage and the voltage drop in the voltage source. This is used to calculate the total current as

$$V(t) = V_m(t) + i_{total}(t)R_{source} \leftrightarrow i_{total}(t) = \frac{V(t) - V_m(t)}{R_{source}} \quad (4.6)$$

which finally lets us describe the derivative of the motor voltage as a function of $V(t)$, $V_m(t)$ and $i_m(t)$.

$$\dot{V}_m(t) = \frac{1}{C} \left(\frac{V(t) - V_m(t)}{R_{source}} - i_m(t) \right) \quad (4.7)$$

A new state space model can now be formulated.

$$x = \begin{bmatrix} \dot{\theta}(t) \\ i_m(t) \\ V_m(t) \end{bmatrix}, \dot{x} = \begin{bmatrix} -\frac{b}{J} & \frac{K_t}{J} & 0 \\ -\frac{K_e}{L} & -\frac{R}{L} & \frac{1}{L} \\ 0 & -\frac{1}{C} & -\frac{1}{CR_{source}} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{CR_{source}} \end{bmatrix} V(t) \quad (4.8)$$

The transfer functions for this system are

$$\begin{bmatrix} H_{V(s) \rightarrow \dot{\theta}}(s) \\ H_{V(s) \rightarrow i_m}(s) \\ H_{V(s) \rightarrow V_m}(s) \end{bmatrix} = I(Is - A)^{-1}B = \frac{1}{CR_{source} \det(Is - A)} \begin{bmatrix} \frac{K_t}{JL} \\ \frac{JL}{Js+b} \\ s^2 + s\left(\frac{bL+JR}{JL}\right) + \frac{bR+K_eK_t}{JL} \end{bmatrix} \quad (4.9)$$

If we do the name swap $R_{source} = R_s$ to make the expressions more compact, then

$$\det(Is - A) = \frac{s^3 JCLR_s + s^2(JCR_sR + JL + LCR_s b) + s(JR + bRCR_s + bL + JR_s + K_t K_e CR_s) + b(R + R_s) + K_t K_e}{JCLR_s} \quad (4.10)$$

which means that the transfer functions denominator becomes

$$CR_s \det(Is - A) = \frac{s^3 JL + s^2(JCR_sR + JL + LCR_s b) + s(JR + bRCR_s + bL + JR_s + K_t K_e CR_s) + b(R + R_s) + K_t K_e}{JL} \quad (4.11)$$

We can note that when $C = 0$ the denominator in equation 4.9 becomes

$$\frac{s^2(JL) + s(JR + JR_s + bL) + b(R + R_s) + K_t K_e}{JL} \quad (4.12)$$

This corresponds to the system without a capacitor and the transfer functions described by [Greiff, 2017].

We are interested in analysing the transfer function from $V(s)$ to $\dot{\theta}$ and how the capacitance effects the mean value. If we look at the static gain of the transfer function

$$H_{V(s) \rightarrow \dot{\theta}}(0) = \frac{K_t}{b(R + R_s) + K_t K_e} \quad (4.13)$$

where we can see that the transfer function in stationarity is independent of the value of C which means that we can not explain the increased rotor speed by simply looking at the static gain.

We will assume that except for this increased RPM the capacitor does not further influence the dynamics of the system and can simply be viewed as a low pass filter of the PWM-input. We assume that the two state representation with $\hat{\theta}$ and y , whether it is the current or the motor voltage, still is applicable.

4.4 Measurements and Data Sets

All measurements of RPM, current and motor voltage are from now on normalized with max values according to the following table.

Table 4.1 Max values used for normalisation

$\hat{\theta}$ [rpm]	V_m [V]	I [A]
24000	3.3	1

To check linearity the following duty cycle sequence, seen in Figure 4.13, was run.

Input step sequence.

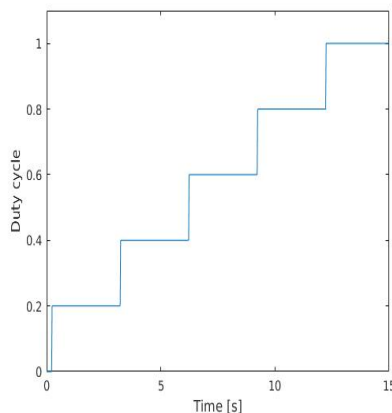


Figure 4.13 The stepwise increasing control signal sequence used for measuring linearity.

For which the following RPM and current/voltage was measured.

Hardware configuration step sequence responses.

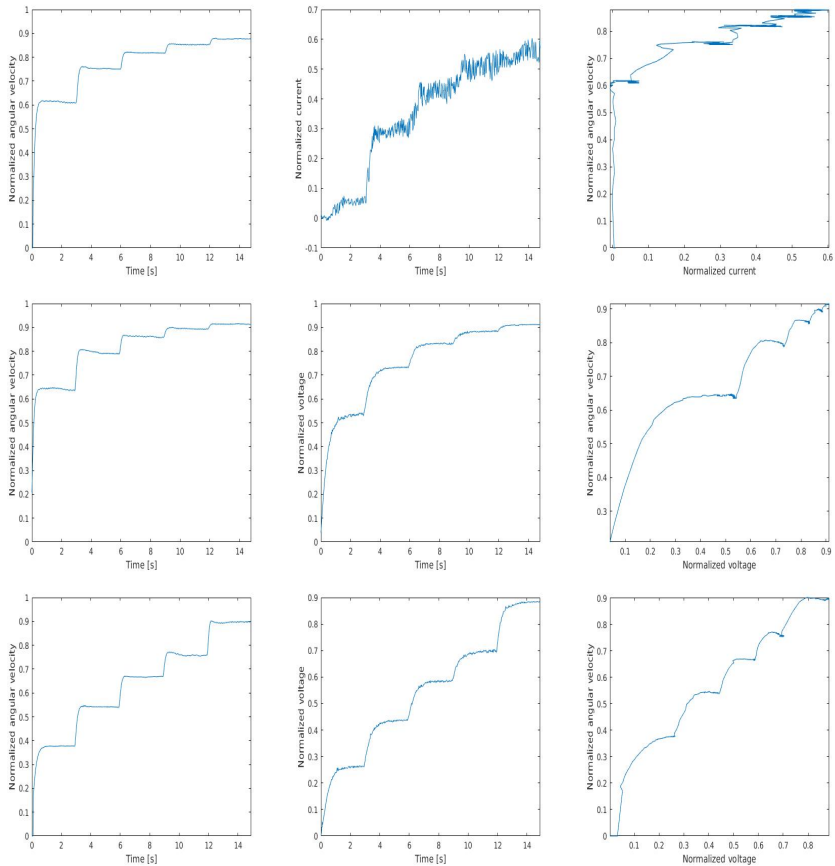


Figure 4.14 Measured RPM, left, feedback variable, middle, and RPM vs feedback variable, right, for the three hardware configurations. Top row: current sense. Middle row: motor voltage with capacitance. Bottom row: motor voltage without capacitance.

These measurements sequences have the correlation coefficients presented in Table 4.2.

Table 4.2 The correlation coefficients obtained from the measurements seen in Figure 4.14.

	$r_{u\theta}$	r_{uy}	$r_{\theta y}$
CS	0.837	0.942	0.891
$V_m \text{ cap}$	0.881	0.876	0.977
V_m	0.975	0.979	0.990

As previously mentioned a correlation coefficient close to 1 indicates linear relations. From these correlation coefficients and the plots in Figure 4.14 we can conclude that there are nonlinearities present, though to different degrees depending on configuration. A higher level of noise present in the current measurements compared with the voltage measurements can also be noted.

The data sets used for the identification was obtained by setting the duty cycle to $u = 0.5 + 0.3 \sin(t\omega(t))$ where the angular frequency increases linearly, $\omega(t) = \frac{2\pi t}{40}$, for 80 seconds. The data set used for verification of the identification was obtained by using a duty cycle sequence recorded from a test flight, the sequence can be seen in Figure 4.15.

In-flight duty cycle recording.

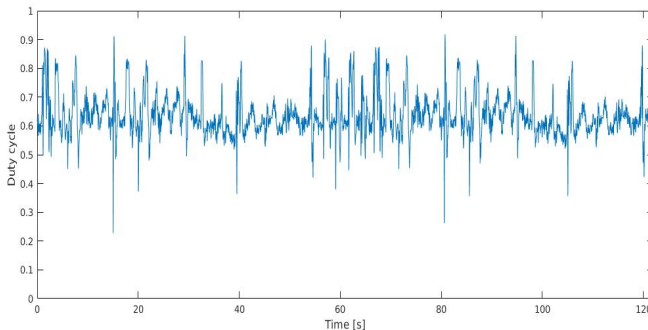


Figure 4.15 Duty cycle sequence obtained from a flying Crazyflie used to evaluate the aquired estimators.

4.5 Identification Methods

Because of the nonlinear behaviour of the configurations, the identification methods from the previous chapter are not directly applicable. Two different approaches

are tried to circumvent this limitation. One is to find a direct mapping from the measured signal to the rotor speed, thus eliminating the need to find a state space representation. The other is to complete the linear model with a nonlinearity on the input, a so called Hammerstein model which is described by [Narendra and Gallman, 1966]. This nonlinearity can be found both experimentally or with more mathematical approaches.

The Hammerstein identification methods described below are all described as ways of identifying transfer functions. The methods can however be extended so that a state space model is obtained by parameter sweep or multiple linear regression as described in the previous chapter. The greater performance and computational efficiency of the multiple linear regression approach makes it superior and will be the method used when state space parameters are needed.

Below, six different estimator models and the method that identifies the required parameters are presented.

Measurement to RPM Mapping

The most simple form of RPM estimation would be to find a direct mapping between the measured signal y , whether it is current or motor voltage, and RPM.

By studying the step responses in Figure 4.14 it seems like a power function can be a good approximation.

$$\hat{\theta} = ky^p \quad (4.14)$$

By using logarithms the parameters can be found using linear regression.

$$\log(\hat{\theta}) = p \log(y) + \log(k) \quad (4.15)$$

An attempt at improving the model was made by adding a constant term to equation 4.14.

$$\hat{\theta} = ky^p + c \quad (4.16)$$

This can not be solved with a linear regression model. The parameters were instead found using MATLABs `nlinfit` function.

Heuristic Hammerstein

We assume that the non linear system dynamics can be described as a linear system in combination with a nonlinearity on the input as illustrated by the block diagram i Figure 4.16.

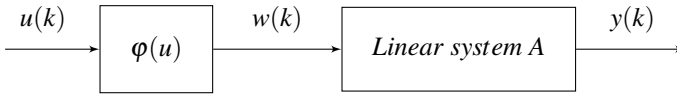


Figure 4.16 Hammerstein model block diagram.

By studying the plots in Figure 4.14 we conclude that the nonlinearity can be assumed to be approximately on the form

$$\varphi(u(k)) = 1 - e^{-au(k)} = w(k) \quad (4.17)$$

For which a value of $a = 4.2$ was found to be adequate by trial and error. By using equation 4.17 on the data sets the new input data set w_k was calculated and could be used to identify the linear system with multiple regression as described in the previous chapter.

The identified system can then be formulated as

$$x(k+1) = Ax(k) + B\varphi(u(k)) \quad (4.18)$$

which could be implemented in a Kalman filter or as open loop estimation.

Duty Cycle to Motor Voltage Transfer function

To account for the added capacitor and the extended state space model, presented in section 4.3, we study the relation between duty cycle and V_m . From equation 4.9 we can see that there is a transfer function from the input voltage to motor voltage. We assume that a transfer function can be added to the previous exponential nonlinearity as illustrated in Figure 4.17.

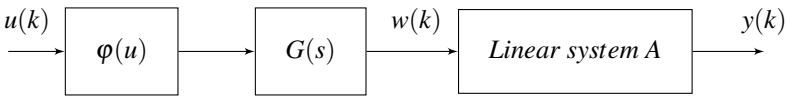


Figure 4.17 Hammerstein model block diagram with added transfer function $G(s)$ to model PWM-filtering.

The voltage across the motor was measured for the motor voltage configuration with a capacitor and a first order continuous transfer function was added to fit the simulated response to the measured in combination with the previously described non-linear function 4.17 but with a different exponential constant value, $a = 3.8$, found experimentally. The following transfer function was found by trial and error and gave promising results

$$G(s) = \frac{0.873}{0.35s + 1} \quad (4.19)$$

which can be seen in Figure 4.18.

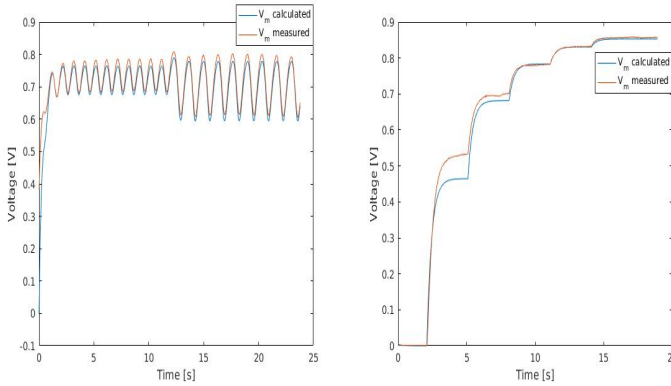


Figure 4.18 By measuring the voltage across the motor for the configuration with a capacitor the measurement results marked with brown lines were obtained. A transfer function and nonlinearity was then fitted heuristically in simulations to give the result marked with the blue line.

In the figure to the right a large error can be noticed for voltages below 0.55 V. This is assumed to be negligible since the voltage normally is higher during flight, as can be seen in Figure 4.15.

Linear Regression Hammerstein

So far the methods that have been used to handle the non-linearity have been of an experimental nature which can be both time consuming and yield poor results. A more systematic approach is therefore needed. By assuming the nonlinearity to be on the form of a unknown polynomial, methods to identify the transfer function presented below have been used on a similar nonlinear identification problem by [Kara and Eker, 2004].

Open loop

We consider a Hammerstein model on the form illustrated in Figure 4.16. The model for the linear part is given as

$$A(q^{-1})y(k) = B(q^{-1})w(k) + e(k) \quad (4.20)$$

where e_k is white gaussian noise, y_k is the output of the linear system and w_k is the input. Here

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_{n_a}q^{-n_a} \quad (4.21)$$

and

$$B(q^{-1}) = b_0 + b_1q^{-1} + b_2q^{-2} + \dots + b_{n_b}q^{-n_b} \quad (4.22)$$

where q^{-1} denotes the backward shift operator. We assume that the nonlinear mapping between u_k and w_k can be expressed as a polynomial of order n :

$$w(k) = \gamma_1 u(k) + \gamma_2 u^2(k) + \dots + \gamma_n u^n(k) \quad (4.23)$$

which means that equation 4.20 can be rewritten as

$$A(q^{-1})y(k) = B(q^{-1}) \sum_{j=1}^n \gamma_j u^j(k) + e(k). \quad (4.24)$$

The assumption that $\gamma_1 = 1$ can be made which means that equation 4.24 can be rewritten as

$$A(q^{-1})y(k) = B(q^{-1})u(k) + \sum_{j=2}^n \sum_{i=0}^{n_b} b_i \gamma_j q^{-i} u^j(k) + e(k). \quad (4.25)$$

By defining the polynomial

$$S_j(q^{-1}) = \gamma_j B(q^{-1}) = s_{j0} + s_{j1}q^{-1} + \dots + s_{jn_b}q^{-n_b} \quad (4.26)$$

the final equation is obtained:

$$A(q^{-1})y(k) = B(q^{-1})u(k) + \sum_{j=2}^n S_j(q^{-1})u^j(k) + e(k). \quad (4.27)$$

Which can be expressed on linear regression form with the regression vector

$$\phi^T = (-y(k-1), -y(k-2), \dots, -y(k-n_a), u(k), u(k-1), \dots, u(k-n_b), u^2(k), \dots, u^2(k-n_b), \dots, u^n(k), \dots, u^n(k-n_b)) \quad (4.28)$$

and the parameter vector

$$\theta = (a_1, a_2, \dots, a_{n_a}, b_0, b_1, \dots, b_{n_b}, s_{20}, \dots, s_{2n_b}, \dots, s_{n0}, \dots, s_{nn_b})^T. \quad (4.29)$$

Since the coefficients of the polynomial $S_j(q^{-1})$ are dependent on the polynomial $B(q^{-1})$ according to equation 4.26 it is not possible to identify the parameters in such a way so that $\gamma_1 \dots \gamma_n$ can be retrieved using linear regression. Linear regression can however be used to identify the parameters of the parameter vector if the

polynomials $S_j(q^{-1})$ instead are considered to be independent. Once the parameters are identified equation 4.25 can then be rewritten as

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k) + \sum_{j=2}^n \frac{S_j(q^{-1})}{A(q^{-1})}u^j(k) + e(k) \quad (4.30)$$

which can be used as an open loop estimator, as illustrated by the block diagram in Figure 4.19.

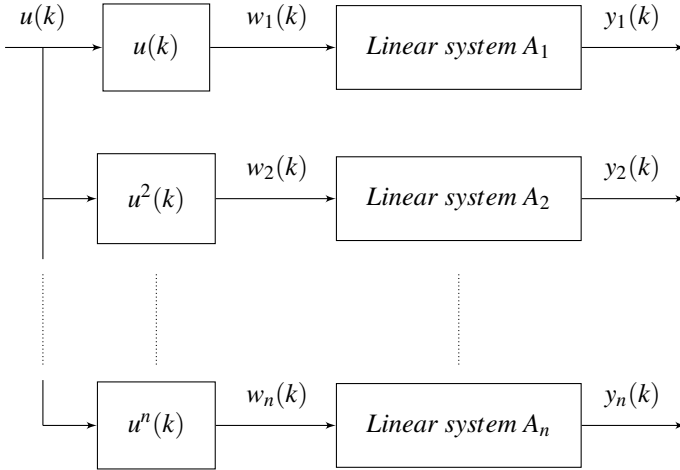


Figure 4.19 Hammerstein model approximation. If the nonlinearity is approximated as an unknown polynomial

The estimated output is given by the sum of the system outputs y_i

$$y(k) = \sum_{i=1}^n y_i(k). \quad (4.31)$$

An evaluation of the results of this method can be found in the next section.

This approach is not as computationally efficient as if the coefficients $\gamma_1 \dots \gamma_n$ had been identified but it should still be a feasible solution for applications without strict restrictions on the computation capacity.

State Space and Kalman Filter

An improvement to the open loop hammerstein model would be to express it on state space form and implement it in a Kalman filter.

By using multiple linear regression it is possible to identify the system and input matrices, $A_1 \dots A_n$ and $B_1 \dots B_n$ which then can be combined into a complete hammerstein model state space model

$$\dot{x}_h = A_h x_h + B_h u_h \quad (4.32)$$

where

$$A_h = \begin{bmatrix} A_1 & \mathbf{0}_{2 \times 2} & \dots & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & A_2 & \dots & \mathbf{0}_{2 \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \dots & A_n \end{bmatrix} \quad (4.33)$$

and

$$B_h = \begin{bmatrix} B_1 & \mathbf{0}_{2 \times 1} & \dots & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{2 \times 1} & B_2 & \dots & \mathbf{0}_{2 \times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2 \times 1} & \mathbf{0}_{2 \times 1} & \dots & B_n \end{bmatrix} \quad (4.34)$$

with the state and input vectors

$$x_h = \begin{bmatrix} x_1^{\dot{\theta}} \\ x_1^I \\ x_2^{\dot{\theta}} \\ x_2^I \\ \vdots \\ x_n^{\dot{\theta}} \\ x_n^I \end{bmatrix}, u_h = \begin{bmatrix} u \\ u^2 \\ \vdots \\ u^n \end{bmatrix}. \quad (4.35)$$

The output is given by equation 4.31 which has the matrix equivalent

$$y_h = C_h y, C_h = [0 \quad 1 \quad 0 \quad 1 \quad \dots \quad 0 \quad 1]. \quad (4.36)$$

4.6 Results and Conclusion

To evaluate the identification methods each was implemented and tested for all three hardware configurations; current sensing and motor voltage feedback with and without capacitance. Because of high measurement noise, noticeable in Figure 4.14, a low pass filter was added to the current sensing configuration. The estimators were tried on the test sequence shown in Figure 4.15. Every Hammerstein method is evaluated by analysing the open loop model response and comparing it with the measured RPM and Voltage/Current. A Kalman filter implementation of the linear

regression Hammerstein is also tested to verify that the state space representation presented in the previous section is feasible.

Error Analysis

Like in the previous chapter the error is calculated with the error norm as described in equation 3.22. The identified models for each hardware configuration are compared and the relative errors for the RPM estimators are displayed in Table 4.3 and for the current/voltage estimators in Table 4.4 where just as in the previous chapter the method with the least error has been chosen as the reference value. Here the estimator named *open* is the current duty cycle to RPM mapping implemented on the Crazyflie as given by equation 2.6 which will be used as a reference to evaluate if the developed estimators are an improvement to the existing open loop solution. *map1* and *map2* are the measurement to RPM mappings described by equation 4.14 and equation 4.16, *h ham* is the heuristic Hammerstein described by equations 4.17 and 4.18, $V \rightarrow V_m$ the Hammerstein model with the duty cycle to motor voltage transfer function, equation 4.19, and *ham kal* and *ham open* is the Kalman filter and open loop implementation of the Hammerstein models found through linear regression.

Table 4.3 Relative error norm for the RPM estimators for each hardware configuration. The smallest error norm has been chosen as the reference value for each configuration.

	<i>open</i>	<i>map1</i>	<i>map2</i>	<i>h ham</i>	$V \rightarrow V_m$	<i>ham kal</i>	<i>ham open</i>
<i>CS</i>	8.068	1.263	1.421	3.561	4.810	1.027	1
$V_m \text{ cap}$	4.572	1	1.245	2.402	2.362	1.631	1.531
V_m	5.159	2.069	2.496	1.156	1	1.975	1.967

Table 4.4 Relative error norm for the open-loop voltage/current estimators for each hardware configuration. The smallest error norm has been chosen as the reference value for each configuration.

	<i>h ham</i>	$V \rightarrow V_m$	<i>ham open</i>
<i>CS</i>	1.019	1.009	1
$V_m \text{ cap}$	2.018	2.690	1
V_m	1.299	1.003	1

To compare the hardware configurations the sample mean error $\bar{E} = \frac{E}{N}$, where E is the error size according to equation 3.22 and N is the number of samples, was

calculated for the methods that proved to have the least error according to Table 4.3 and Table 4.4.

Table 4.5 Sample mean error for the least error estimator of each hardware configuration for both RPM and current/voltage calculated from the normalized data sets.

	CS	V_m cap	V_m
RPM	0.135×10^{-3}	0.337×10^{-3}	0.262×10^{-3}
Current/Voltage	0.723×10^{-3}	0.229×10^{-3}	0.407×10^{-3}

Apparently the current sense method yielded the smallest error for RPM estimation and motor voltage measuring with capacitance yielded the smallest error for the feedback variable estimation, which is current or voltage depending on the hardware configuration. It might be misleading to simply look at this error metric though since the estimator with the smallest error does not necessarily capture the dynamics of the system in the best way. For example, if we study the results for the RPM estimation in Figure 4.20 and Figure 4.21 and compare the current to RPM mapping with the heuristic Hammerstein we can see that the second method captures the behaviour better but has a static error that leads to the larger error size. A small static error can probably be compensated for by the outer control loop, though this has to be evaluated in flight, and should therefore be considered to be the better estimator. To measure how well the system captures the system dynamics we can study the variance of the error. An estimator which captures the system dynamics perfectly except for a static error would have zero variance. We therefore compute the population variance for each of the estimators and configurations. As with the error norm, the variances are displayed relative to the smallest variance in Table 4.6 and Table 4.7.

Table 4.6 Relative error variance of the RPM for the estimators and hardware configurations. The variance is used as a metric to compare how well the estimator captures the system dynamics.

	<i>open</i>	<i>map1</i>	<i>map2</i>	<i>h ham</i>	$V \rightarrow V_m$	<i>ham kal</i>	<i>ham open</i>
CS	77.497	5.179	5.459	1.122	2.089	1	1.0
V_m cap	34.084	1.605	2.155	1.250	1.842	1	1.251
V_m	7.161	1.877	2.308	1.924	1.420	1.002	1

Table 4.7 Relative error variance of the current/voltage for the open-loop estimators and hardware configurations.

	<i>h ham</i>	$V \rightarrow V_m$	<i>ham open</i>
<i>CS</i>	1.023	1	1.065
$V_m \text{ cap}$	1	1.033	1.299
V_m	6.207	5.731	1

Care has to be taken when looking at the variance since the noise of the measurements effect different estimator results unequally. The mappings from measurement signal to RPM are especially sensitive to this since the estimator results here are given without any filtering of the measurement signals.

Current Sense Configuration

Taking the results from Table 4.3 and Table 4.6 into consideration while studying Figure 4.21 and Figure 4.22 we see that the open and closed hammerstein models seem to be the best estimators for the RPM. The experimental Hammerstein has a larger static error but otherwise captures the dynamics nicely which also is illustrated by its low variance. The results for the current estimation is similar with the open Hammerstein resulting in the least error size and variance.

Current sense configuration, results for the measured current to rotor speed mapping.

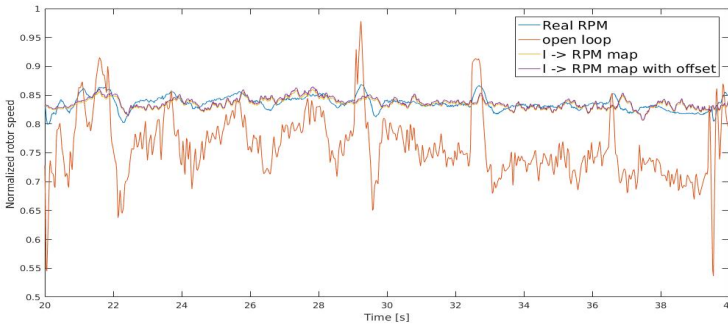


Figure 4.20 Results for the current sense configuration estimators. Here the $y \rightarrow RPM$ mappings are compared with the real value and the currently implemented open loop estimator. The mapping seems to capture the mean value but does not follow changes in the rotor speed fast enough.

Current sense configuration, results for the heuristic Hammerstein estimators.

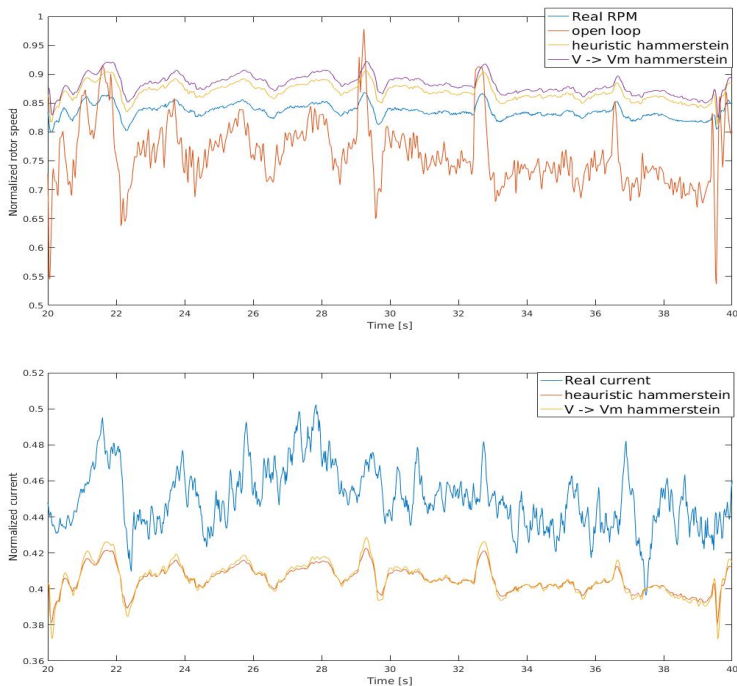


Figure 4.21 Results for the current sense configuration estimators. Here the heuristic hammerstein methods are compared with the real value and the currently implemented open loop estimator. Both models capture the dynamics in a good way but with a static error. The static error of the model with an added transfer models can be seen to have a slightly larger static error.

Current sense configuration, results for the linear regression Hammerstein estimator and its Kalman filter implementation.

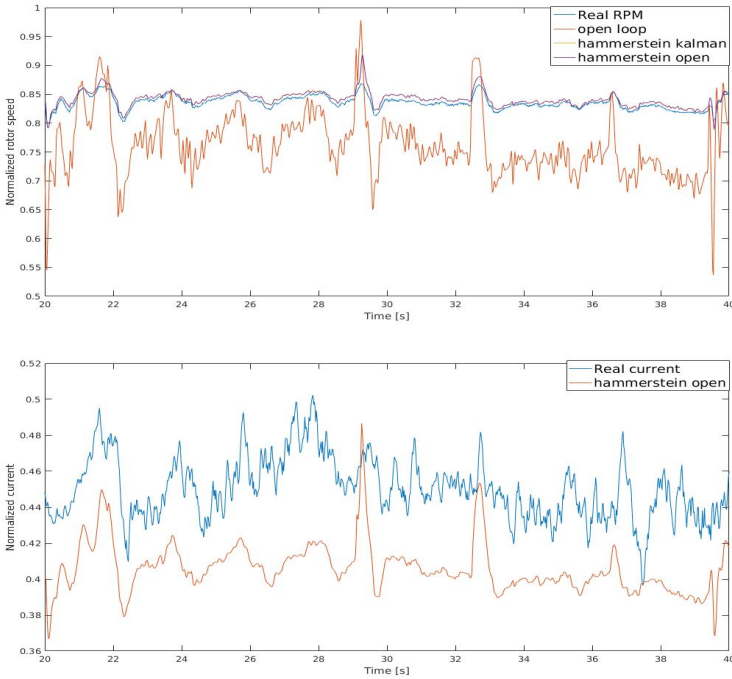


Figure 4.22 Results for the current sense configuration estimators. Here the linear regression hammerstein methods are compared with the real value and the currently implemented open loop estimator. Both estimators capture the dynamics and with a smaller static error than the heuristic models. No difference can be noticed between the open loop and the kalman implementation.

Voltage Measuring with Capacitance Configuration

The results for the motor voltage measuring configuration with a capacitance differ quite a bit from those presented in the previous section. The $V \rightarrow RPM$ mapping provides the smallest error norm but does not have the smallest error variance, which instead can be found with the Hammerstein Kalman implementation, for which the results can be seen in Figure 4.25. However by studying Figure 4.23 we can conclude that a part of the higher variance can be attributed once again to the noise of the measurement signal, though not as much because of the increased signal to noise ratio compared with the current sense configuration. Unfortunately the performance of the voltage estimation does little to clarify the situation since the method with the

least error size provides the highest variance. From these error metrics it is therefore difficult to tell which estimator is to be considered the best for this configuration.

Voltage measuring with capacitance configuration, results for the measured voltage to rotor speed mapping.

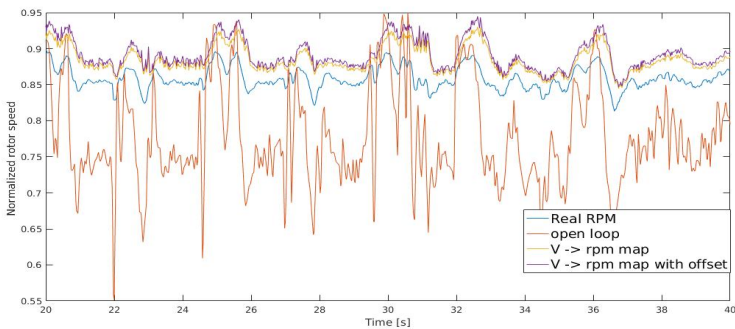


Figure 4.23 Results for the motor voltage measuring configuration with capacitor estimators. Here the $V \rightarrow RPM$ mappings are compared with the real value and the currently implemented open loop estimator. The estimated rotor speed seems to be following the actual speed quite accurately, however with a static error.

Voltage measuring with capacitance configuration, results for the heuristic Hammerstein estimators.

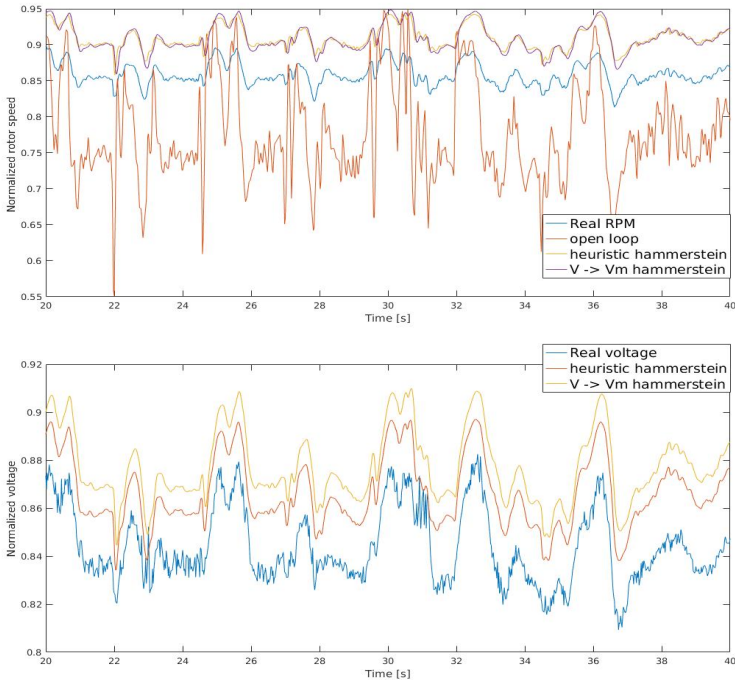


Figure 4.24 Results for the motor voltage measuring configuration with capacitor estimators. Here the heuristic hammerstein methods are compared with the real value and the currently implemented open loop estimator. Compared with the $V \rightarrow rpm$ mappings the static error is larger for these estimators but more accurately captures the dynamics.

Voltage measuring with capacitance configuration, results for the linear regression Hammerstein estimator and its Kalman filter implementation.

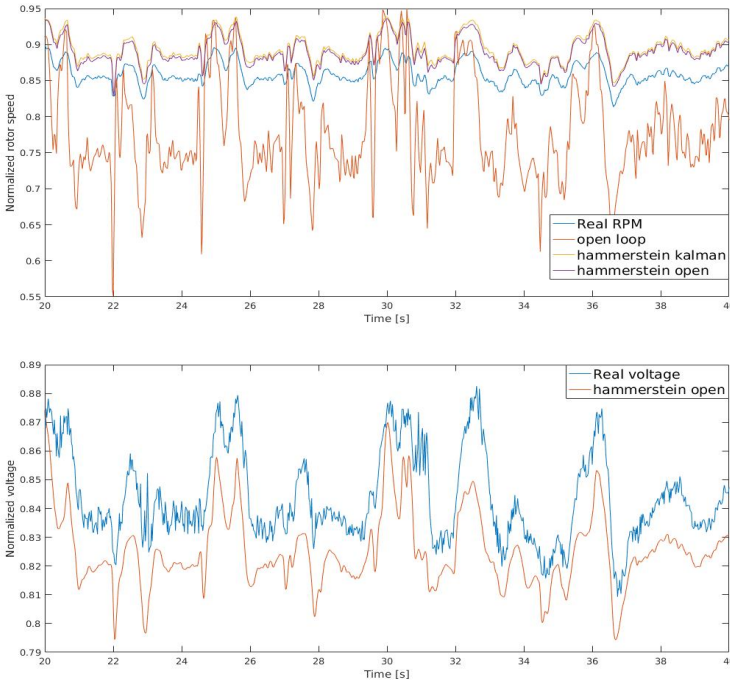


Figure 4.25 Results for the motor voltage measuring configuration with capacitor estimators. Here the linear regression hammerstein methods are compared with the real value and the currently implemented open loop estimator. Both estimators capture the dynamics but with a static error. The kalman filter implementation does not make any noticeable improvements.

Voltage Measuring without Capacitance Configuration

For the motor voltage configuration without capacitor it is obvious when studying Figure 4.27 that the heuristic and $V \rightarrow V_m$ Hammerstein methods, although having the smallest error, do a poor job at mimicking the system dynamics. The linear regression Hammerstein methods, for which the results can be seen in Figure 4.28, are therefore to be preferred for this configuration, despite the bigger error size noted in Table 4.3. The voltage to RPM mapping seen in Figure 4.26 seems to capture the dynamics well but is still outperformed by the Hammerstein method both regarding variance and error size.

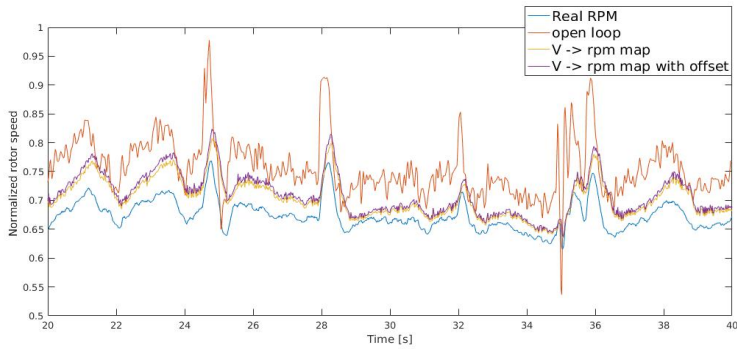
Voltage measuring configuration, results for the measured voltage to rotor speed mapping.

Figure 4.26 Results for the motor voltage measuring configuration without capacitor estimators. Here the $y \rightarrow RPM$ mappings are compared with the real value and the currently implemented open loop estimator.

Voltage measuring configuration, results for the heuristic Hammerstein estimators.

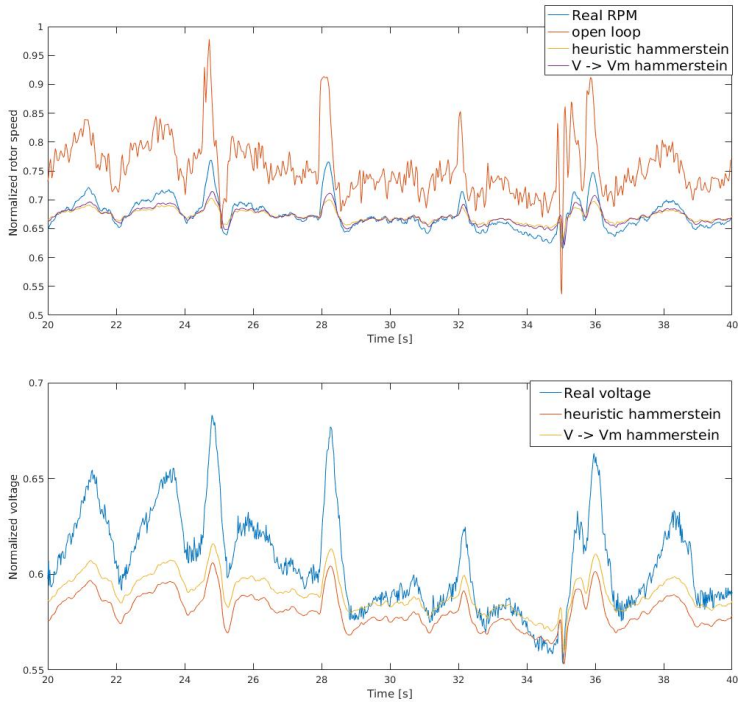


Figure 4.27 Results for the motor voltage measuring configuration without capacitor estimators. Here the heuristic hammerstein methods are compared with the real value and the currently implemented open loop estimator. The estimators does a poor job at capturing the rotor speed dynamics even though the mean error is low.

Voltage measuring configuration, results for the linear regression Hammerstein estimator and its Kalman filter implementation.

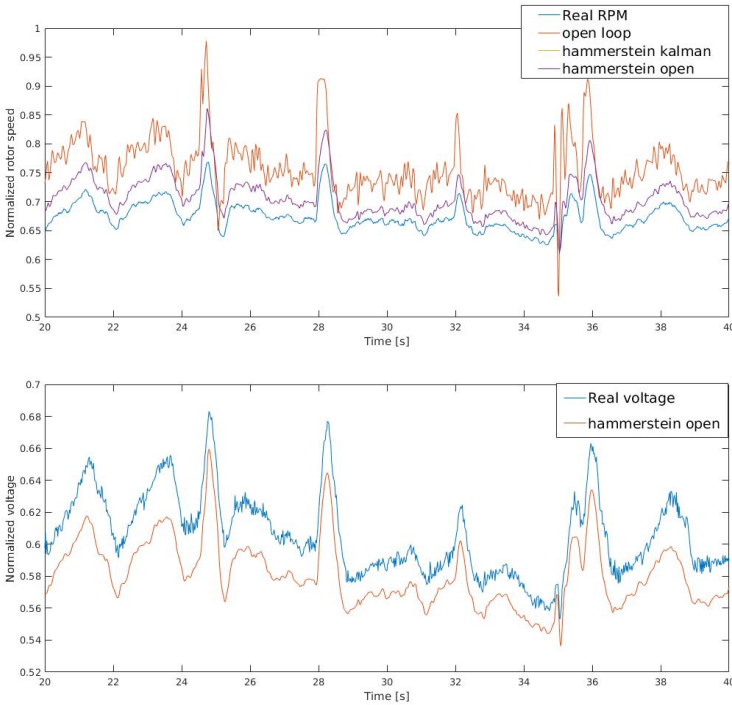


Figure 4.28 Results for the motor voltage measuring configuration without capacitor estimators. Here the linear regression hammerstein methods are compared with the real value and the currently implemented open loop estimator. The estimators captures the dynamics but has a static error. The Kalman filter implementation did not manage to improve the result in any noticeable way compared to the open loop estimator.

Estimator Methods Conclusions

$y \rightarrow RPM$ mapping: The measurement to RPM mapping is the simplest way to construct a closed loop estimator and it proved to be surprisingly efficient, especially for the motor voltage measuring configurations. This estimator can easily be implemented on the Crazyflie and is not computationally demanding. The identification process is also simple since the mapping obtained via linear regression proved to be the best one, and no time has to be spent on finding parameters experimentally.

Heuristic Hammerstein: The heuristically determined Hammerstein method succeeds in capturing the dynamics of the system for both hardware configurations that

uses a PWM-filtering capacitance. The worse performance for the configuration without this filtering is explained by looking at Figure 4.14 and noting that the non-linearity is different compared to the previous ones which leads us to the main drawback of this method. Namely that it is time consuming to experimentally find the coefficients for the nonlinearity and an optimal solution is not guaranteed. Though if decent parameter values are found the estimator is easily implemented both as an open loop and in a Kalman filter.

$V \rightarrow V_m$ **Hammerstein:** The attempt at extending the nonlinearity with a transfer function to mimic the $V \rightarrow V_m$ was not successful. Surprisingly, it is the method that results in the least RPM error norm for the voltage measuring configuration without capacitor. This is likely more of a coincidence and a result of the fact that the methods that are better at capturing the system dynamics have a static error. The fact that it fails to outperform the heuristic Hammerstein method for the current sense configuration proves that this was not an effective method.

Linear Regression Hammerstein: These methods have proven to be the best at capturing the dynamics of the system even though a static error might be present. The drawback is the increased computational cost which for the open loop estimator increases linearly with the polynomial order. The results for the Kalman implementation here can be seen as an indication that it is possible to have a filter that outperforms the open loop estimator though this depends on correctly tuned covariance matrices for which further analysis is needed. The Kalman filter implementation depends on multiplication of large matrices for which the computational time and its effect on the feasibility for a real time implementation has to be analysed.

Hardware Configuration Conclusions:

If a static error is acceptable it is possible to find estimators for all three configurations. The question which configuration is to be preferred is therefore not decided from this criterion. It is instead a question of production cost and energy efficiency. Since adding a capacitance in parallel to the motor increases the RPM it might mean that it is more energy efficient since the duty cycle can be lowered while maintaining the same RPM. That this corresponds to a lower energy consumption has to be verified. If the capacitance does not contribute to greater power consumption the motor voltage measuring configuration with a capacitor should be the preferred solution since the cost and space needed for the current sense resistor can be saved.

4.7 Future Work

The Hammerstein method has to be considered to be the approach with the most potential though there are improvements that can be made. As previously mentioned further analysis of the covariance matrices has to be done for a successful

Kalman filter implementation. The alternative would be to find a method that correctly finds the nonlinearity polynomial coefficients in equation 4.23. Further work is also needed to minimize the static error that is the result of the current identification. As previously mentioned a small static error can be handled by the outer control loop but it is still to be considered preferable to remove it. Implementing the estimators on the Crazyflie and testing them as a part of a simple control loop, for example a PID controller, is also something that needs to be done before finally settling for a specific hardware configuration and estimator. Expansion boards for on flight testing have been designed and can be found in appendix A, they have however not yet been produced and tested.

Bibliography

- Åström, K. J. and B. Wittenmark (2011). *Computer-Controlled Systems*. Dover Publications.
- Bitcraze (2016). *Loco positioning deck*. URL: <https://wiki.bitcraze.io/projects:lps:deck> (visited on 2017-07-05).
- Bitcraze (2017a). *Bitcraze website*. URL: <https://www.bitcraze.io/> (visited on 2017-06-03).
- Bitcraze (2017b). *Crazyflie kit electronics explained*. URL: [https://wiki.bitcraze.io/projects:crazyflie:hardware:explained?s\[\]=battery&s\[\]=specification](https://wiki.bitcraze.io/projects:crazyflie:hardware:explained?s[]=battery&s[]=specification) (visited on 2017-05-21).
- Greiff, M. (2017). *Modelling and control of the crazyflie quadrotor for aggressive and autonomous flight by optical flow driven state estimation*. eng. Student Paper.
- Johansson, R. (2011). *System Identification*. Dover Publications.
- Kara, T. and I. Eker (2004). *Nonlinear modeling and identification of a DC motor for bidirectional operation with real time experiments*. Energy Conversion and Management.
- Liu, K., J. M. Fitzgerald, and F. L. Lewis (1993). *Kinematic Analysis of a Stewart Platform Manipulator*. Vol. 40. IEEE Transactions on Industrial Electronics.
- Ljung, L. (1987). *System Identification, Theory for the User*. Prentice-Hall, Inc.
- Lyshevski, S. E. (1998). *Nonlinear control of mechatronic systems with permanent-magnet DC motors*. Mechatronics.
- MachineDesign (2002). *Coreless dc motors*. URL: <http://www.machinedesign.com/basics-design/coreless-dc-motors> (visited on 2017-07-23).
- Narendra, K. S. and P. Gallman (1966). *An Iterative Method for the Identification of Nonlinear Systems Using a Hammerstein Model*. IEEE transaction on automatic control.
- Regan, T., J. Munson, G. Zimmer, and M. Stokowski (2005). *Current Sense Circuit Collection*. Linear Technology.

Smith, S. W. (1997). *The Scientists and Engineer's Guide to Digital Signal Processing*. California Technical Pub; 1st edition.

A

Expansion Board Designs

To implement the current or voltage sensing described in this report on the Crazyflie a expansion board has to be developed. In this section two PCB-design concepts are briefly presented, a simple one without a microcontroller unit (MCU) and a more complex one with a MCU, which should be printed on a 25×27 mm board. The amplifier and the filters used would be the same for both designs but the components used during the system identification experiments in this thesis report should not be considered optimal. Before being produced a more thorough evaluation would be needed to find optimal filter parameter values and amplifiers.

A.1 PCB without MCU

The simplest way to construct an expansion board for the feedback measurements is to add the differential amplifier and the filters described previously in Chapter 4 and connect the amplifier outputs to the analog inputs of the Crazyflie expansion ports. However, the Crazyflie normally uses another expansion board for position estimating, the Loco positioning deck [Bitcraze, 2016], which uses the available analog input pins for SPI (Serial peripheral interface bus) communication. These pins are therefore not available. A local ADC IC with a SPI serial interface, for example a Microchip MCP3004, can therefore be used to both convert the analog values and transfer them to the MCU of the Crazyflie. A circuit diagram for this design can be seen in Figure A.1. A problem with using this design might be delays in the communication between the ADC on the expansion board and the MCU.

A.2 PCB with MCU

To eliminate the risk of measurement signal delays that can arise when using SPI is to have a separate MCU, programmed via serial wire debug (SWD), on the expansion board with its own analog inputs as well as output pins with PWM-support. The MCU would obtain rotor speed reference values from the MCU on the Crazyflie

using UART communication. This would mean that the expansion board acts as a servo drive for the motors. To control the motors with the added MCU, motor control transistors like the ones on the Crazyflie have to be added to the expansion board, see Figure A.2.

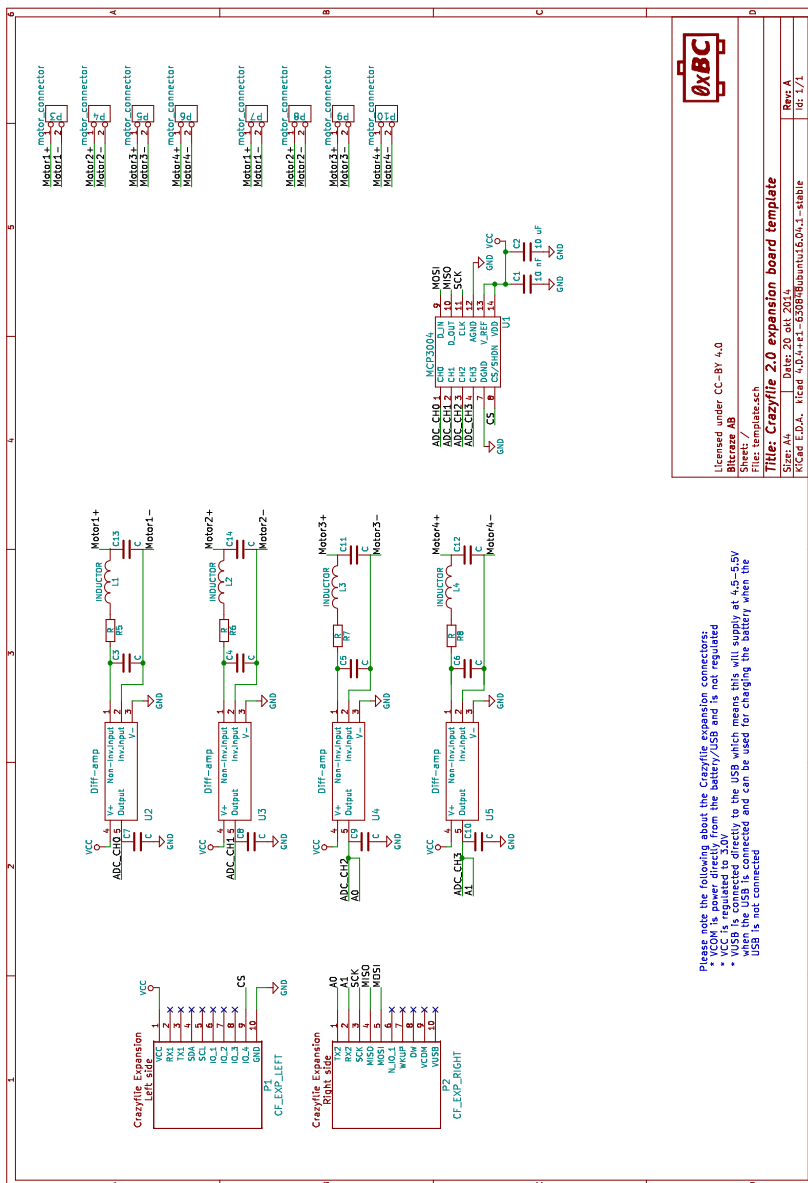


Figure A.1 A simple expansion board design for measuring motor voltage. The measured values are locally converted in an ADC IC and communicated to the Bitcraze MCU using SPI.

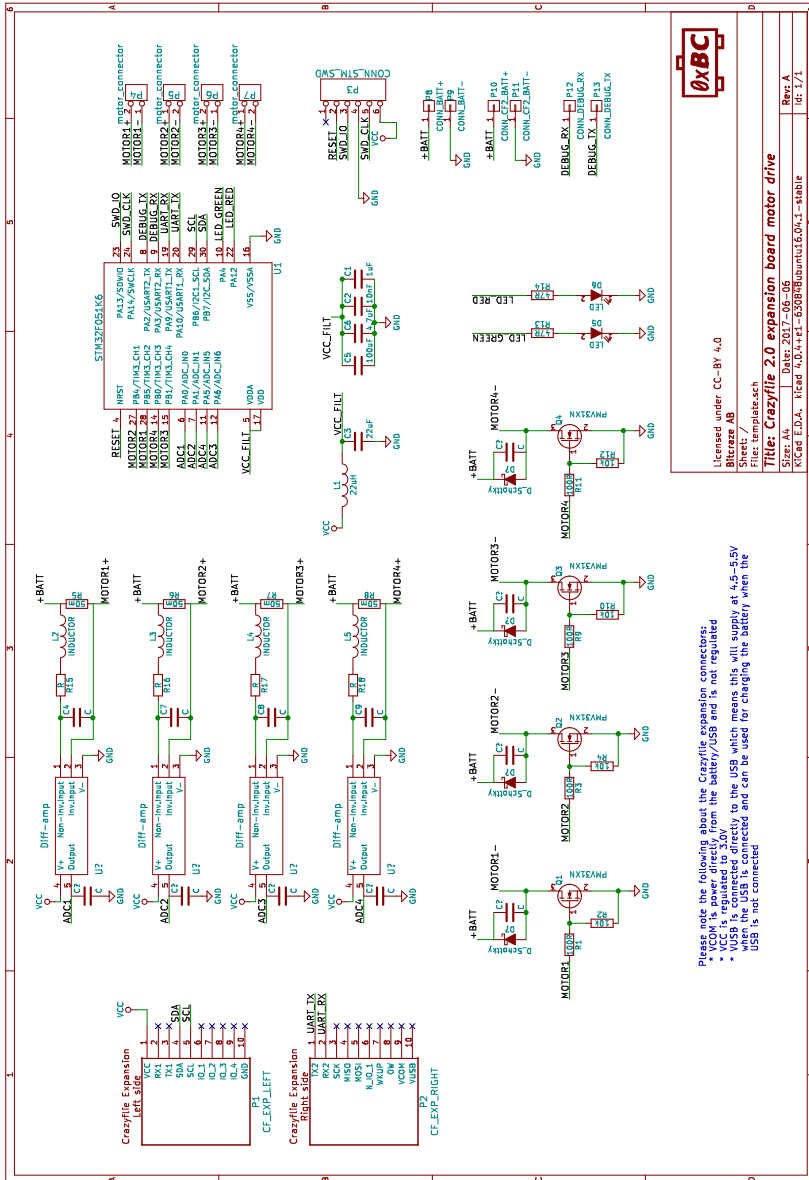


Figure A.2 An expansion board design using a on board MCU for controlling the motors. This MCU gets rotor speed reference values from the main MCU on the Crazyflie via UART communication.

B

Landing Platform

Since a goal of the thesis work was to implement functionality on the quadcopter to land on moving surfaces, a moveable platform had to be obtained. The 6-DOF Stewart-like-platform seen in figure B.1 was used. It is controlled by an Arduino UNO and uses 6 rotational servos for arm positioning.

Landing platform

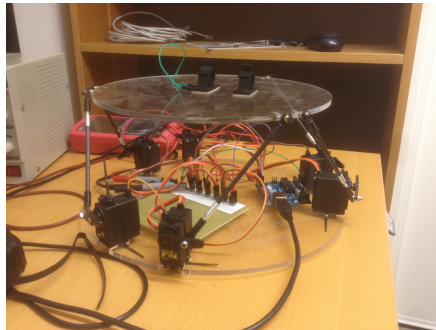


Figure B.1 The stewart-like-platform.

In this section we derive the equations describing the kinematics of the landing platform.

B.1 Approximate Length Model

In the case where the platform is controlled by linear actuators the inverse kinematics have a unique solution which has been derived by [Liu et al., 1993].

An inertial frame (X, Y, Z) is fixed at the center of the base platform with the Z axis pointing upward. Another coordinate system (x, y, z) is fixed at the center of the top platform. The location of the origin of the top frame relative the base

frame is denoted by $[p_x, p_y, p_z]$ and (α, β, γ) represents the rotation angles defined by rotating the top frame first about the X axis with α degrees, then about the Y axis with β degrees and then about the Z axis with γ degrees.

The transformation from the top to the base frame is described by the transformation matrix

$$\mathbf{T}_{BASE}^{TOP} = \begin{bmatrix} \cos(\beta)\cos(\gamma) + \sin(\alpha)\sin(\beta)\sin(\gamma) & -\cos(\beta)\sin(\gamma) + \sin(\alpha)\sin(\beta)\cos(\gamma) & \cos(\alpha)\sin(\beta) & p_x \\ \cos(\alpha)\sin(\gamma) & \cos(\alpha)\cos(\gamma) & -\sin(\alpha) & p_y \\ -\sin(\beta)\cos(\gamma) + \sin(\alpha)\cos(\beta)\sin(\gamma) & \sin(\beta)\sin(\gamma) + \sin(\alpha)\cos(\beta)\cos(\gamma) & \cos(\alpha)\cos(\beta) & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.1})$$

which is a function of the location and rotation of the top frame.

The coordinates of a top frame node can be expressed in base frame coordinates by the calculation

$$\begin{bmatrix} X_{Ti} \\ Y_{Ti} \\ Z_{Ti} \\ 1 \end{bmatrix} = \mathbf{T}_{BASE}^{TOP}(p_x, p_y, p_z, \alpha, \beta, \gamma) \begin{bmatrix} x_{Ti} \\ y_{Ti} \\ z_{Ti} \\ 1 \end{bmatrix} \quad (\text{B.2})$$

The desired lengths of the actuators are then determined by

$$L_i = \sqrt{(X_{Ti} - X_{Bi})^2 + (Y_{Ti} - Y_{Bi})^2 + (Z_{Ti} - Z_{Bi})^2} \quad (\text{B.3})$$

Now taking into account that we are using servos instead of linear actuators an approximation is made to obtain the desired servo angle. We assume that the rotation of the servo arm is made in the same plane as the base and top node and that the desired angle can be obtained from trigonometry as

$$\theta = \arccos\left(\frac{L_i^2 + l_{si}^2 - l_{li}^2}{2L_i l_{si}}\right) \quad (\text{B.4})$$

where θ is the deviation from the servo position where the servo arm and the platform arm are parallel, l_{si} is the length of the servo arm and l_{li} is the length of the arm connecting the servo arm to its respective top platform node.

B.2 Complete Inverse Kinematics

Denote the servo arm endpoint position \mathbf{x} , the servo center position \mathbf{x}_B and the position of the top node \mathbf{x}_T . All positions are expressed in base frame coordinates. For a desired top platform position the problem is to find the corresponding servo arm positions and can be stated as three constraint equations.

$$\begin{cases} |\mathbf{x} - \mathbf{x}_B| = l_s \\ |\mathbf{x}_T - \mathbf{x}| = l_l \\ \mathbf{a}^T \mathbf{x} = b \end{cases} \quad (\text{B.5})$$

Here the first equation is the constraint of the servo arm length, the second equation is the constraint given by the length of the platform arm. The third equation constrains the servo arm to one plane.

The equations can be simplified by rewriting the absolute values

$$\begin{cases} \mathbf{x}^T \mathbf{x} - 2\mathbf{x}_B^T \mathbf{x} = l_s^2 - \mathbf{x}_B^T \mathbf{x}_B \\ \mathbf{x}^T \mathbf{x} - 2\mathbf{x}_t^T \mathbf{x} = l_t^2 - \mathbf{x}_t^T \mathbf{x}_t \\ a^T \mathbf{x} = b \end{cases} \quad (\text{B.6})$$

$$\begin{cases} \mathbf{x}^T \mathbf{x} - 2\mathbf{x}_B^T \mathbf{x} = l_s^2 - \mathbf{x}_B^T \mathbf{x}_B \\ 2(\mathbf{x}_B^T - \mathbf{x}_t^T) \mathbf{x} = l_t^2 - l_s^2 + \mathbf{x}_B^T \mathbf{x}_B - \mathbf{x}_t^T \mathbf{x}_t \\ a^T \mathbf{x} = b \end{cases} \implies \begin{cases} \mathbf{x}^T \mathbf{x} - 2\mathbf{x}_B^T \mathbf{x} = l_s^2 - \mathbf{x}_B^T \mathbf{x}_B \\ a^T \mathbf{x} = b \\ c^T \mathbf{x} = d \end{cases} \quad (\text{B.7})$$

where $c = 2(\mathbf{x}_B^T - \mathbf{x}_t^T)$ and $d = l_t^2 - l_s^2 + \mathbf{x}_B^T \mathbf{x}_B - \mathbf{x}_t^T \mathbf{x}_t$

The system of equations with three unknowns now consists of one quadratic and two linear equations. By solving the linear part of the system, we should be able to obtain a solution depending on one parameter

$$\begin{cases} a^T \mathbf{x} = b \\ c^T \mathbf{x} = d \end{cases} \implies \mathbf{A} \mathbf{x} = \begin{bmatrix} b \\ d \end{bmatrix} \implies \mathbf{A} \mathbf{x} = [\hat{\mathbf{A}} | \mathbf{A}_3] \begin{bmatrix} \hat{\mathbf{x}} \\ x_3 \end{bmatrix} \quad (\text{B.8})$$

$$\hat{\mathbf{A}} \hat{\mathbf{x}} = \begin{bmatrix} b \\ d \end{bmatrix} - \mathbf{A}_3 x_3 = h(x_3) \implies \hat{\mathbf{x}} = \hat{\mathbf{A}}^{-1} h(x_3) \quad (\text{B.9})$$

if we choose the parameter $x_3 = t$

$$\mathbf{x} = \begin{bmatrix} K_1 + K_2 t \\ K_3 + K_4 t \\ t \end{bmatrix} \quad (\text{B.10})$$

where

$$\begin{cases} K_1 = \hat{\mathbf{A}}_{1,1}^{-1} b + \hat{\mathbf{A}}_{1,2}^{-1} d \\ K_2 = -(\hat{\mathbf{A}}_{1,1}^{-1} a_3 + \hat{\mathbf{A}}_{1,2}^{-1} c_3) \\ K_3 = \hat{\mathbf{A}}_{2,1}^{-1} b + \hat{\mathbf{A}}_{2,2}^{-1} d \\ K_4 = -(\hat{\mathbf{A}}_{2,1}^{-1} a_3 + \hat{\mathbf{A}}_{2,2}^{-1} c_3) \end{cases} \quad (\text{B.11})$$

$$t = p \pm \sqrt{p^2 + q} \quad (\text{B.12})$$

where $p = \frac{x_{B1} K_2 + x_{B2} K_4 + x_{B3} - K_1 K_2 - K_3 K_4}{1 + K_2^2 + K_4^2}$ and $q = \frac{l_s^2 - \mathbf{x}_B^T \mathbf{x}_B + 2(K_1 x_{B1} + K_3 x_{B2}) - K_1^2 - K_3^2}{1 + K_2^2 + K_4^2}$

The solution that will be used in the implementation is the one with the negative sign.

To calculate the desired angle that is to be used as reference value for the servo, the angle between the vector $(\mathbf{x} - \mathbf{x}_B)$ and the Z-axis unit vector is obtained with their dot product.

$$\phi = \arccos\left(\frac{x_3 - x_{B3}}{|\mathbf{x} - \mathbf{x}_B|}\right) \quad (\text{B.13})$$

A matlab implementation was made where the platform followed sinusoidal displacement and rotations in p_z and α , both simultaneously and independently, showing that the angles are calculated correctly.

Platform kinematic simulations

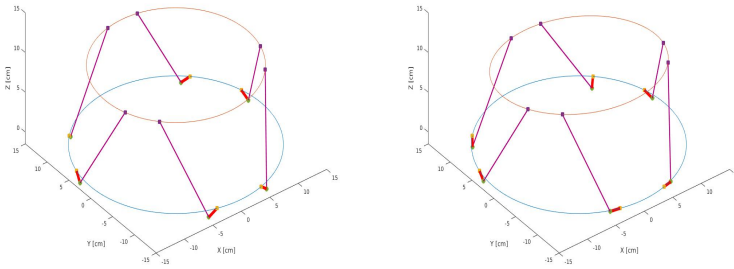


Figure B.2 A visualization of the Stewart platform matlab simulation used to verify the inverse kinematics. *Left:* The simulated platform when displacement and rotation are set to zero. *Right:* The simulated platform with a rotation around the x-axis. These simulation results indicate that the servo angles are calculated correctly.

B.3 Real Time Implementation

The platform controller is implemented as a ROS-node on a host PC. From a desired position and rotation this node calculates the corresponding servo angles and uses a publisher to send the data to the Arduino. The Arduino uses the <ros.h> library and a subscriber can therefore be set up which reads the servo angles sent by the PC. These angles are then set as reference values to the servos.

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER'S THESIS	
		<i>Date of issue</i> November 2017	
		<i>Document Number</i> ISRN LUTFD2/TFRT--6041--SE	
<i>Author(s)</i> Erik Ottosson		<i>Supervisor</i> Anders Rantzer, Dept. of Automatic Control, Lund University, Sweden Bo Bernhardsson, Dept. of Automatic Control, Lund University, Sweden (examiner)	
		<i>Sponsoring organization</i>	
<i>Title and subtitle</i> Identification and Estimation of DC-Motor Dynamics for the Crazyflie Quadcopter			
<i>Abstract</i> <p>In this thesis methods for finding state space representation parameters for low inductance DC-motors are found and evaluated. Theory previously tested in simulations is verified experimentally and improved upon for both linear behaviour when the motor is controlled with a DC-source as well as nonlinear behaviour when controlled with pulse width modulation (PWM). For the linear behaviour a linear regression model with multiple regression sets proved satisfactory at identifying the motor dynamics. For the nonlinear behaviour caused by the PWM signal it was found that a Hammerstein model was sufficient to model the dynamics and several rotor speed estimators were tested and compared. Based on these results a design for a Crazyflie expansion board is suggested.</p>			
<i>Keywords</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316			<i>ISBN</i>
<i>Language</i> English	<i>Number of pages</i> 1-69	<i>Recipient's notes</i>	
<i>Security classification</i>			

<http://www.control.lth.se/publications/>