Motor control under strong vibrations

Modelling and simulation of countermeasure dispenser for aircraft protection in Simulink

Andreas Fredlund
Tobias Persson
Abstract

Motor control under strong vibrations

Andreas Fredlund & Tobias Persson

This thesis intends to investigate and simulate key components and properties of a defensive electromechanical countermeasure dispenser unit mounted on an aircraft. The main focus was on comparing and evaluating different motor control strategies during strong vibrations: vibrations similar to those the dispenser unit might endure during its service life. The simulations of the performance and the mathematical modelling of the key components where done in MATLAB/Simulink. This made it possible to determine that Active Disturbance Rejection Control proved to be most promising when it comes to performance, but when it comes to implementation complexity the Linear Active Disturbance Rejection Control or the regular PID controller might be preferred instead. Another controller that occasionally showed good performance was the Fuzzy Logic controller but when it came to robustness and adaptiveness against uncertainties it couldn’t keep up. However, there are some mechanical features of the dispenser unit that affect the overall performance regardless of the chosen control strategy. The modelling in Simulink was made in such a way that allows easy modification and testing of such features. Although, to be able to represent the dynamics of the system in mathematical formulas, a few assumptions were made, which might implicate the exactness of the system but will still be a valid representation.
Populärvetenskaplig sammanfattning

En av produkterna under utveckling på Saab AB i Järfälla är en elektromekanisk motmedelsfällare som används av stridsflygplan och skyddar mot eventuella missilattacker. Den är uppbryggd av flertalet synkroniserade motorer vars huvuduppgift är att skicka ut skenmål för att missleda inkommande attacker. Eftersom det rör sig om livshotande attacker är det otroligt viktigt att funktionaliteten är utomordentligt pålitlig, även under de extrema miljöomständigheter som ett flygplan utsätts för. En av dessa omständigheter är starka vibrationer som kan uppstå från t.ex. planetets jetmotorer, vind och acceleration. Därför har detta examensarbete ägnats åt att undersöka hur en av motmedelsfällaren motorer, samt dess sammanhängande delar, påverkas av starka vibrationer. Undersökningen har utförts genom att modellera de vibrationskänsliga delarna och simulera deras egenskaper, samt att jämföra olika sätt att kontrollera och styra motorn. Denna undersökning har lett till att en kontrollalgoritmi vid namn Linear Active Disturbance Rejection Control visade sig vara bäst lämpad för uppgiften, dock med hårfint marginal. Det som visade sig vara av större betydelse för funktionaliteten var hur de sammanhängande delarna till motorn påverkades av de starka vibrationerna, då simuleringarna stundtals antydde att en fällning av skenmål var ogenomförbar. Detta önskade fenomen kan inte elimineras helt genom en förbättrad kontrollalgoritm utan kräver även andra mekaniska åtgärder. Därför har modellen i detta examensarbete även designats på ett sätt som möjliggör enkel modifiering och simulering av fysiska egenskaper innan de verkliga ändringarna görs. Detta kan ge större förståelse för förändringens innebörd och på så vis rättfärdiga eventuella mekaniska åtgärder, men framförallt undvika kostnaden och tid fotg för att beställa en ny del och testa den.
**Nomenclature**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADRC</td>
<td>Active Disturbance Rejection Controller</td>
</tr>
<tr>
<td>BLDC</td>
<td>Brushless DC</td>
</tr>
<tr>
<td>DOM</td>
<td>Degree Of Membership</td>
</tr>
<tr>
<td>ESO</td>
<td>Extended State Observer</td>
</tr>
<tr>
<td>FLC</td>
<td>Fuzzy Logic Controller</td>
</tr>
<tr>
<td>FOC</td>
<td>Field Oriented Control</td>
</tr>
<tr>
<td>GRMS</td>
<td>G Root Mean Square</td>
</tr>
<tr>
<td>LADRC</td>
<td>Linear Active Disturbance Rejection Controller</td>
</tr>
<tr>
<td>LESO</td>
<td>Linear Extended State Observer</td>
</tr>
<tr>
<td>LTD</td>
<td>Linear Tracking Differentiator</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple Input Multiple Output</td>
</tr>
<tr>
<td>NL</td>
<td>Negative Large</td>
</tr>
<tr>
<td>NLSEF</td>
<td>Nonlinear State Error Feedback Control Law</td>
</tr>
<tr>
<td>NM</td>
<td>Negative Medium</td>
</tr>
<tr>
<td>NS</td>
<td>Negative Small</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional Integral Derivative</td>
</tr>
<tr>
<td>PL</td>
<td>Positive Large</td>
</tr>
<tr>
<td>PM</td>
<td>Positive Medium</td>
</tr>
<tr>
<td>PS</td>
<td>Positive Small</td>
</tr>
<tr>
<td>PSD</td>
<td>Power Spectral Density</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation</td>
</tr>
<tr>
<td>SPWM</td>
<td>Sinusoidal PWM</td>
</tr>
<tr>
<td>SWPWM</td>
<td>Space Vector PWM</td>
</tr>
<tr>
<td>SISO</td>
<td>Single-Input Single-Output</td>
</tr>
<tr>
<td>TD</td>
<td>Tracking Differentiator</td>
</tr>
<tr>
<td>Z</td>
<td>Zero</td>
</tr>
</tbody>
</table>
Table of Contents
1 Introduction .................................................................................................................................. 5
  1.1 Background .......................................................................................................................... 5
  1.2 Objectives .................................................................................................................................. 6
  1.3 Structure .................................................................................................................................. 6
  1.4 Work Distribution .................................................................................................................. 7
2 The System .................................................................................................................................. 8
3 Theory ...................................................................................................................................... 10
  3.1 Control Theory ....................................................................................................................... 10
    3.1.1 PID Controller .............................................................................................................. 10
    3.1.2 Active Disturbance Rejection Control .............................................................................. 11
    3.1.3 Linear Active Disturbance Rejection Control ................................................................. 13
    3.1.4 Fuzzy Logic Controller ................................................................................................. 16
  3.2 Motor Basics .......................................................................................................................... 17
  3.3 Brushless DC motor ............................................................................................................. 19
    3.3.1 Mathematical Modelling of BLDC ................................................................................. 19
  3.4 Field Oriented Control .......................................................................................................... 21
  3.5 Gears ...................................................................................................................................... 28
  3.6 Springs .................................................................................................................................. 30
  3.7 Vibration Test .......................................................................................................................... 31
4 Methodology ............................................................................................................................... 35
  4.1 Simulink ................................................................................................................................ 35
    4.1.1 Overview of the System ................................................................................................. 35
    4.1.2 Field Oriented Control & BLDC ..................................................................................... 36
    4.1.3 Gears & Axles ............................................................................................................... 37
    4.1.4 Springs & Stack ............................................................................................................. 38
    4.1.5 Vibrations/Disturbances ............................................................................................... 38
    4.1.6 PID Controller .............................................................................................................. 38
    4.1.7 Active Disturbance Rejection Control .............................................................................. 39
    4.1.8 Linear Active Disturbance Rejection Control ................................................................. 40
    4.1.9 Fuzzy Logic Controller ................................................................................................. 41
  4.2 The Pick Motion ....................................................................................................................... 46
5 Result ......................................................................................................................................... 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Pick Motion without Vibrations</td>
<td>47</td>
</tr>
<tr>
<td>5.2</td>
<td>Pick Motion with Vibrations</td>
<td>50</td>
</tr>
<tr>
<td>5.3</td>
<td>Failed Separation due to Vibrations</td>
<td>54</td>
</tr>
<tr>
<td>5.4</td>
<td>Comparison of the Controllers without Vibrations</td>
<td>58</td>
</tr>
<tr>
<td>5.5</td>
<td>Comparison of the Controllers with Vibrations</td>
<td>59</td>
</tr>
<tr>
<td>5.6</td>
<td>Displacement of the Stack during a longer Simulation</td>
<td>61</td>
</tr>
<tr>
<td>6</td>
<td>Discussion &amp; Conclusion</td>
<td>63</td>
</tr>
<tr>
<td>6.1</td>
<td>Improvement and Future Work</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>References</td>
<td>65</td>
</tr>
</tbody>
</table>
1 Introduction

The next generation Electronic Warfare (EW) system has electro-mechanical countermeasure dispensers for aircraft protection that are made up of a distributed network of software nodes and uses brushless DC (BLDC) motors to control hatches and control arms. To maintain high performance as this environment is subjected to disturbances due to vibrations from e.g. the engines, wind and acceleration, the control strategy of the motors must be robust, accurate and adaptive. Other components, which are connected to the motors in one way or another, will also be under investigation since they too are affected by the vibrations and play an important role for the overall performance of the dispenser.

The current system that is under development is using a functioning control loop for controlling the motors, but the solution has its limitations. It is using libraries that have inaccessible source code, i.e. the developers at Saab are restricted to a certain set of functions and are unable to modify them. These libraries will ideally be replaced in the near future; either by a more suitable library that supports the demanded needs of the system, or by completely replace it with their own developed library. This master thesis will lay the foundation and theoretical background for the replacement of the current library.

1.1 Background

In the past decades, the field of controlled electrical drives has expanded rapidly due to advanced development in semiconductor devices and microprocessors with excellent computational speeds. It is technical improvements like these together with superior features over brushed DC motors that have made the way for the BLDC to grow in popularity. The advantageous features properties like high efficiency, long operating life, high dynamic properties, reliability and high torque to weight/size ratio. Because of these important advantages, the use of BLDC motors in application fields like electrical vehicles, robotics and aerospace has increased rapidly. The main concern in high performance application like these is to achieve smooth torque production. To be able to that, a very efficient control technique and commutation strategy is needed, which are also robust against disturbances and parameter variations and uncertainties [1]-[5].

The main commutation strategies that are being used are trapezoidal (also known as six-step), sinusoidal and field oriented control (FOC). This paper will investigate FOC because it has been proven to be the most versatile and optimal commutation strategy because of its smooth operation at low speed while still efficient at high speed. FOC can also be referred as vector control since it controls the current space vectors on a rotating reference frame which makes it possible for the controllers to operate on a DC signal [6].

A primitive, yet powerful choice for a controller is the proportional integral derivative (PID) controller, which has been around for nearly a century. It has been dominating various sectors of the industry due to its simplicity and accurate control but is starting to outlive its usefulness because it cannot keep up with the pursuit of efficiency. The PID controller is sensitive to signal noise since the derivative action in the controller might amplify it. In [7] Jingqing Han proposed the Active
Disturbance Rejection Controller (ADRC) which inherits from the PID controller and eliminates its weaknesses. ADRC is proven to be better than the classical PID, but the problem is to tune it to perfection. The complexity in the tuning process increases drastically since the number of tuning parameters increases. An alternative controller is the Linear ADRC (LADRC) which transforms the non-linear parts into linear. This leads to a far more simplified tuning process. Another controller that will be investigated is the Fuzzy Logic Controller (FLC). The FLC eliminates the need of mathematical modeling. Instead the parameters are tuned based on knowledge and experience of how a system behaves on certain inputs, which makes the solution very specific for different kinds of systems [7]-[11].

Products and equipment that will be mounted on airplanes must meet certain standards and regulatory qualifications and must therefore undergo a variety of extreme environmental tests. One of these standards is the MIL-STD-810G which is specifically prepared for military applications but is often used on commercial products as well. The MIL-STD-810G standard provides several tests that are designed to address realistic environmental conditions a product might endure during its service life. One of these tests is the vibration test which is often performed by a specialized company using a vibration test shaker. During the vibration test there is an exceptional opportunity to also evaluate the product’s performance and, in this case, to investigate the motor and evaluate the control algorithms during extreme conditions [15].

1.2 Objectives

This thesis will investigate different trade-offs between control algorithms and will eventually be compared and evaluated against each other to be able to find the best one. To make the comparison interesting and relevant, a model of the BLDC motor connected to a couple of key components in the system will be generated. The model will be produced in MATLAB/Simulink and the resulting simulations will be considered when choosing a suitable control algorithm for controlling the BLDC motor in an eventual new software library. Another objective was to generate the simulation model as a helping tool when considering future mechanical changes in the real system.

1.3 Structure

The report starts with an explanation of the system with all its relevant key components and how they interact with each other. In the next section, theory behind relevant equations and explanations of different control strategies, motor basics, BLDC, FOC, gears, springs and vibrations are discussed. Next, the equations from the Theory section are used in the Methodology section to model the whole system with the four control strategies PID, ADRC, LADRC and FLC. In the Result section, graphs with relevant variables during different motions with and without disturbances are presented for each control strategy. The report ends with a discussion of how the control strategies differ from one another, possible improvements, future work and finally a conclusion of the entire work. Some parts in this thesis are company restricted and are therefore not publicly available. These parts are
described in Attachment 1-5. Some key word throughout the report has been replaced by an image instead of text in order to decrease the number of hits from search engines.

1.4 Work Distribution

The project was designed for two persons and most of the work has been done together but with distributed responsibilities. Tobias Persson was mainly in charge of the vibrations, Active Disturbance Rejection Control and Linear Active Disturbance Rejection Control, while Andreas Fredlund was mainly in charge of the physical equations of the gears and the Fuzzy Logic Controller.
2 The System

The main objective for the countermeasure dispenser system is to defend the airplane by ejecting decoys when under attack. The dispenser is filled with a stack of expendables containing radar and IR decoys and operates with a number of BLDC motors that pick and separate an expendable from the stack, eject the expendable and push the stack forward in a synchronized manner. In this master thesis the main focus was on the picking motor, simply called the “pick motor”, and its motion, Figure 1 a, b, c. However, the pick motor is directly dependent on how the rest of the system behaves. Therefore, the system physical characteristics that have an impact on the pick motor need to be modelled with proper equations. An overview of the system with the relevant parts is shown in Figure 1.

![Diagram of the system](image)

a. The system at rest.  
b. Picks an expendable from the stack.  
c. The expendable is sent to the eject phase.

Figure 1. Overview of the system containing springs, expendables and pick axles during a pick motion.

In the real system the left side of the springs is obviously connected to other parts of the dispenser, but in this thesis scope the springs are assumed to be attached to a fixed point instead, i.e. “the base”. The base is always contributing with a compression force (similar to the real system) which pushes the stack against the axles. By rotating the axles simultaneously, an expendable can fall into the recesses in the axles and then be separated from the stack. The pick motor is controlling the motion of these axles through a gear train, Figure 2.
The relevant equations and parameters that need to be investigated in order to get a realistic model are:

- Motor equations
- Spring equation
- Equations for gears and axles
- Friction between expendables and the dispenser
- Friction between expendables and pick axles
- Expressions of how the parts interact with each other

The dispenser is occasionally subjected to strong disturbances due to acceleration and vibrations. In the worst case scenario, the stack of expendables is displaced from the axles precisely when an ejection needs to be performed. This leads to no ejected decoy and no protection against an incoming attack.
3 Theory

3.1 Control Theory

In control theory, the most central part is to obtain an output signal from a system which is as close as possible to a desired value or a reference signal. One way to do that is with a feedback signal together with a controller. The block diagram in Figure 3 shows a standard closed loop system where the output signal \( y \) is subtracted from the desired value \( r \) and the error \( e \) is sent as an input to the controller. The controller takes the error and converts it into a command signal \( u \) which is the input to the process or the \textit{plant}.

![Figure 3. Standard representation of a closed loop system.](image)

3.1.1 PID Controller

The PID controller or the proportional integral derivative controller is a very common used controller in industrial fields. As the name reveals, the controller is based on proportional, integral and derivative terms. The PID controller is formed by adding these terms together in a linear fashion

\[
u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de(t)}{dt}\]  

which has the transfer function

\[
F(s) = F_p + F_i + F_d = K_p + K_i \cdot \frac{1}{s} + K_d s
\]

The proportional term corrects the current error while the integrating term considers the past values. In general, with larger \( K_p \) and \( K_i \) you get a faster response time but setting these values too big will lead to an unstable system. Often, a well-tuned PI controller is “good enough” to fulfill a systems requirement, but if you add a derivate term to the controller it allows you to have bigger \( K_p \) and \( K_i \) without losing stability. The derivative action predicts a future value of the error with the current rate of change of the error. With this predicted value the controller adjusts the control signal beforehand, thus reducing ripple and instability of the output signal. However, if the signal is very noisy, the derivative term will more often predict the wrong value and thus amplify the noise instead of reducing it. An overview of a PID controller in a closed loop system is presented in Figure 4.
3.1.2 Active Disturbance Rejection Control

The ADRC inherits the good parts from PID but also addresses its weaknesses. Weaknesses such as

- Drastic reference signal changes that causes the control signal to make unreasonable jumps
- The $D$ part of the PID is very noise sensitive
- The $I$ term reduces steady state error but introduces other complications

To be able to overcome these issues and limitations of the classical PID controller, three new technical solutions will be introduced. The three solutions are a tracking differentiator (TD), an extended state observer (ESO) and a nonlinear state error feedback control law (NLSEF), which together forms a typical structure of ADRC, Figure 5. The noise-tolerant TD is necessary to avoid sudden reference signal jumps that cause the control signal to make drastic changes that the systems dynamics can’t reasonably follow. Both the internal states and the external disturbances can be generalized as total disturbance and does not need to be expressively known. This generalized total disturbance can then be actively estimated and compensated by the ESO, which means that this solution is model independent and require little information about the plant to be controlled. Lastly, to be able to get the tracking of the error to approach zero as fast as possible, a NLSEF is introduced. The NLSEF can perform without the $I$ term in PID and thus avoiding the complications caused by it.

Figure 5. Typical ADRC topology.
ADRC is applicable to most nonlinear multi-input-multi-output (MIMO) systems, but for the sake of simplicity, let’s consider a second order single-input-single-output (SISO) system

\[ \ddot{y} = -a \dot{y} - by + w + bu \]  

3.1.2.1

where \( y \) is the output, \( u \) is the input and \( w \) is external disturbance of the system. The parameter \( a \) is unknown and \( b \) can be estimated with \( b_0 \), i.e. \( b_0 \approx b \), then Eq. 3.1.2.1 can be rewritten as

\[ \dot{y} = -a \dot{y} - by + w + (b - b_0)u + b_0u \]  

3.1.2.2

Let then \( f \) be a representation of both the unknown internal dynamics \(-a \dot{y} - by + (b - b_0)u \) and the external disturbances \( w \), such as

\[ f = -a \dot{y} - by + w + (b - b_0)u \]  

3.1.2.3

Eq. 3.1.2.2 together with Eq. 3.1.2.3 reduces the model of the plant to

\[ \dot{y} = f + b_0u \]  

3.1.2.4

with the assumption that \( f \) is differentiable and let \( \dot{f} = h \), the state space equation of the plant in Eq. 3.1.2.4 can be written as

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 + b_0u \\
\dot{x}_3 &= h \\
y &= x_1
\end{align*}
\]  

3.1.2.5

The ESO can now be constructed as

\[
\begin{align*}
fe_1 &= fal(e, \alpha_{01}, h), \quad fe_2 = fal(e, \alpha_{02}, h), \\
\dot{z}_1 &= z_2 - \beta_{01}e \\
\dot{z}_2 &= z_3 - \beta_{02}fe_1 + b_0u \\
\dot{z}_3 &= -\beta_{03}fe_2
\end{align*}
\]  

3.1.2.6

where \( z_1, z_2 \) and \( z_3 \) are estimated values for \( y, \dot{y} \) and \( f \). The inputs of the ESO are the control signal \( u \) and the output \( y \) from the plant. The nonlinear functions \( fal(e, \alpha, h) \) can be described as

\[
fal(e, \alpha, h) = \begin{cases} 
\frac{e}{h^{1-\alpha}} & \text{if } |e| \leq h \\
\text{sign}(e)|e|^\alpha & \text{if } |e| \geq h
\end{cases}
\]  

3.1.2.7

Moreover, the observer gains \( \beta_{01}, \beta_{02} \) and \( \beta_{03} \) are adjustable tuning parameters of the ESO and the parameters \( \alpha_{01} \) and \( \alpha_{02} \) can also be slightly adjusted for better performance. The control law can then take the form

\[ u = \frac{u_0 - z_3}{b_0} \]  

3.1.2.8
in order to reduce the plant in Eq. 3.1.2.5 to a double integrator plant, i.e.

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= u_0 \\
y &= x_1
\end{align*}
\] 3.1.2.9

which can be controlled by a nonlinear PD controller by making \( u_0 \) a function of the tracking error and its derivative, i.e.\[
\begin{align*}
u_0 &= \beta_1 \text{fal}(e_1, \alpha_1, h) + \beta_2 \text{fal}(e_2, \alpha_2, h), \quad 0 < \alpha_1 < 1 < \alpha_2 \quad 3.1.2.10
\end{align*}
\]

Putting Eq. 3.1.2.10 into Eq. 3.1.2.8 and the desirable control signal \( u \) is obtained with the NLSEF control law

\[
u = \frac{\beta_1 \text{fal}((v_1 - z_1), \alpha_1, h) + \beta_2 \text{fal}(v_2 - z_2), \alpha_2, h) - z_3}{b_0} \quad 3.1.2.11
\]

where \( \alpha_1, \alpha_2 \) and \( h \) are adjustable parameters of the NLSEF and \((v_1 - z_1)\) and \((v_2 - z_2)\) are the tracking error and its derivative. Here, \( z_1, z_2 \) and \( z_3 \) are the outputs from the ESO and \( v_1 \) and \( v_2 \) are the outputs from the TD. The TD can be solved by the following differential equation

\[
\begin{align*}
\dot{v}_1 &= v_2 \\
\dot{v}_2 &= -r \cdot \text{sign} \left(v_1 - v_0 + \frac{v_2 |v_2|}{2r}\right) \quad 3.1.2.12
\end{align*}
\]

where \( v_0 \) is the input reference signal, \( v_1 \) and \( v_2 \) are the desired tracking signal and its derivative and \( r \) is the tuning parameter of the TD. It can be selected depending on the physical limitations of the system and the larger \( r \) is, the faster \( v_1 \) reaches the reference signal.

### 3.1.3 Linear Active Disturbance Rejection Control

LADRC transforms the non-linear mechanisms of the traditional ADRC into linear while still preserving the robust control effects. It also utilizes the concept of bandwidth-parameterization tuning, which greatly simplifies the parameter tuning process. The structure of LADRC is very similar to ADRC, Figure 6. It consists of three parts: linear tracking differentiator (LTD), linear extended state observer (LESO) and a general PD controller.
The LTD tracks the reference signal in order to reduce the impact of signal mutation and outputs a smooth transient profile that the output can reasonably follow. The LESO actively estimates both the unknown internal dynamics and external disturbances and provides feedback for the controller. The plant can then be controlled with a simple PD controller. Let’s consider the second order SISO system as in Eq. 3.1.2.1 and recall the state space equation

$$\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 + b_0 u \\
\dot{x}_3 &= h \\
y &= x_1
\end{align*} \quad \text{3.1.3.1}$$

Based on the state model, a state observer that estimates $f$ can be constructed as

$$\begin{align*}
\dot{x} &= Ax + Bu + Eh \\
y &= Cz 
\end{align*} \quad \text{3.1.3.2}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ b_0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad E = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{3.1.3.3}$$

So for a second order system, a third order LESO must be designed as follows

$$\begin{align*}
\dot{z} &= Az + Bu + L(y - \hat{y}) \\
\hat{y} &= Cz
\end{align*} \quad \text{3.1.3.4}$$

where $L$ is the observer gain vector and after matrix multiplication we get

$$\begin{align*}
\dot{z}_1 &= z_2 + L_1(y - z_1) \\
\dot{z}_2 &= z_3 + L_2(y - z_1) + b_0 u \\
\dot{z}_3 &= L_3(y - z_1)
\end{align*} \quad \text{3.1.3.5}$$
where $z_1, z_2$ and $z_3$ are estimated values for $y, \dot{y}$ and $f$. If all three observer poles are placed at $-w_0$ then the following relation is satisfied

$$L_{1,2,3} = [\beta_1 = 3w_0 \quad \beta_2 = 3w_0^2 \quad \beta_3 = w_0^3]^T$$ 3.1.3.6

which makes the observer bandwidth $w_0$ the only parameter to tune in the LESO and the controller can then be chosen as

$$u = \frac{u_0 - z_3}{b_0}$$ 3.1.3.7

The controller together with a properly designed LESO that estimates $f$ correctly reduces the plant in Eq. 3.1.3.1 to approximately a pure double integrator plant, i.e.

$$\begin{cases}
\dot{x}_1 = x_2 \\
\dot{x}_2 = u_0 \\
y = x_1
\end{cases}$$ 3.1.3.8

This can then be controlled by a simple PD controller designed as follows

$$u_0 = k_p(v_1 - z_1) - k_d z_2$$ 3.1.3.9

where $v_1$ is the reference trajectory and the controller gains are selected by placing the closed loop system poles at $-w_c$ such as

$$k_p = w_c^2 \text{ and } k_d = 2w_c$$ 3.1.3.10

This pole placement technique makes the controller bandwidth $w_c$ the only parameter to tune in the PD controller. In Eq. 3.1.3.9, $v_1$ is output reference signal from the LTD and for a second order system, the LTD is designed as

$$\begin{cases}
\ddot{v}_1 = v_2 \\
\ddot{v}_2 = -k_1(v_1 - v) - k_2 v_2
\end{cases}$$ 3.1.3.11

where $v$ is the input reference signal, $v_1$ is the tracking signal and $v_2$ is the differential signal of the system. The adjustable parameters $k_1$ and $k_2$ are chosen as

$$\begin{cases}
k_1 = r^2 \\
k_2 = 2r
\end{cases}$$ 3.1.3.12

where $r$ is the speed factor and determines the tracking speed of the LTD.
3.1.4 Fuzzy Logic Controller

Some systems require a controller that can be very challenging to set up since the system only can be explained with complex mathematical models. However, when using a FLC the need for mathematical models is eliminated. What is needed for this controller is very good knowledge and experience of how the output of a system reacts to a certain input.

Boolean logic describes a state as 0 or 1 and nothing in between. Fuzzy logic on the other hand, can describe a state as a degree from 0 to 1, where 0 means that it is without doubt false, and 1 means without doubt true. In between, the state can be described as partly false and partly true. The degree of how true a statement is can differ from person to person, so the information is very vague and fuzzy. A simple example is to describe if the temperature in the shower is cold, warm or hot. In Figure 7 the membership functions for cold, warm and hot is shown. The functions take in a crisp value $x ^\circ C$ and associates it with a fuzzy value or a degree of membership, DOM. For example; if the temperature in the shower is 10$^\circ C$, then the DOM of cold is 1 and the DOM of warm and hot is 0. So it is without doubt that the water is cold. But if the water is 20$^\circ C$ then temperature is partly cold and partly warm. In this case the DOM of cold is 0.5 and the DOM of warm is 0.5. The same thing applies on the area between warm and hot. It’s a sliding “gray area” between the two states until it’s no longer warm and it is completely hot.

![Figure 7. Membership functions of cold, warm and hot.](image)

A general Fuzzy Logic controller is shown in the dotted box in Figure 8. The Fuzzifier, i.e. the DOM functions, brings a Fuzzy set to an inference engine which uses a rule base. The rule base consists of multiple rules for different membership functions and different actions. In the simple shower example, a rule could be

Rule 1: IF water IS cold THEN “turn on the heat”.

So, the rules are defined by one statement, or more, followed by an action. The inference engine run through all of these rules, combines them, and calculates a new fuzzy set. Let’s say that the DOM of cold is 1, then the output from the inference engine should be “turn on much heat”. If the DOM of cold is just 0.3, Rule 1 is still valid, but the output from the inference engine should be “turn on little
heat”. The new fuzzy set is then combined in the defuzzifier where real crisp outputs are calculated and can be used in the next step in the control loop.

![Block diagram of a system with Fuzzy Logic controller.](image)

In the shower example it is very easy to convert a crisp input to a crisp output through these steps, because you have experience of what is cold, warm and hot. It is more complicated to implement suitable membership functions, a rule base, an inference engine and a defuzzifier on a large unknown system. There is usually more than just one input to a controller and some of them may be more correlated to the output than others. The approach of getting a good FLC is:

1. Study the system
2. Set membership functions
3. Set rule base and define how to combine the fuzzy set in the inference engine
4. Define how to defuzzify the fuzzy set to crisp output
5. Repeat 1, 2, 3 and 4 and modify if needed.

### 3.2 Motor Basics

To be able to understand how motors convert electrical energy into mechanical energy, some basic mechanisms of the motors operation will be covered. Let’s first look at magnetomotive force $\mathcal{F}$. It is the ability of a coil to produce flux and is the product of the number of turns $N$ and the current $I$ of the coil, i.e.

$$\mathcal{F} = N \times I \quad 3.2.1$$

This can be used to calculate the magnetic flux $\Phi$, which is proportional to the $\mathcal{F}$ and can be described with

$$\Phi = \frac{\mathcal{F}}{R} \quad 3.2.2$$

where $R$ is the reluctance of the material. The reluctance gives a measure of how difficult it is for the magnetic flux to complete its circuit. Iron has very low reluctance and is therefore a good choice for a
magnetic core material in order to generate a higher flux. The magnitude of the magnetic flux density $B$ is the magnetic flux inside a body divided by the cross sectional area $A$ of the body, i.e.

$$B = \frac{\Phi}{A}$$  \hspace{1cm} (3.2.3)

To be able to generate torque, motors exploit the mechanical force $F$ which a current-carrying wire experience when placed in a magnetic field. This can be calculated with

$$\vec{F} = l \cdot (\vec{B} \times \vec{I})$$  \hspace{1cm} (3.2.4)

where $l$ is the current and $l$ is the length on the wire. The vector quantities in Eq. 3.2.4 can be described with Fleming’s left-hand rule, Figure 9, which is a simple and useful tool for finding the direction of the force.

![Fleming’s left-hand rule](image)

It becomes clear that the current needs to be perpendicular to the magnetic field in order generate maximum torque which then can be expressed by

$$T = F \cdot r$$  \hspace{1cm} (3.2.5)

where $r$ is the radius of the motors rotor. From this, we can conclude that the current is the only parameter which is variable during operation and is therefore the only parameter the torque depends on. All the other parameters are fixed and can be expressed by a single torque constant $k_t$. So, the torque can be expressed by

$$T = k_t \cdot I$$  \hspace{1cm} (3.2.6)

which tells how much torque a given motor produces per unit current. When the rotor rotates relative to the coils, there will be an induced voltage in the coils according to Faraday’s law, which can be referred to as back electromotive force (back EMF) since it counters the applied source voltage in a motor. At a constant speed with no mechanical load, the back EMF will grow until it’s equal to the source voltage. Simultaneously, the current drawn by the coils will fall to zero so the motor consumes only the energy from losses at this state. The magnitude of the back EMF, abbreviated as $e$, is proportional to the angular speed in rad/s and can be expressed as
where \( k_e \) is the back EMF constant that holds all fixed influential factors.

### 3.3 Brushless DC motor

There are many different types of motors but they all consist of a stationary frame, the stator, and a rotating frame, the rotor. In the case of a permanent magnet BLDC motor, the magnets are mounted on the rotor and the conductive coils are attached to the stator. A simple and typical BLDC motor has three equally spaced phases with one or more pole pairs on the rotor, Figure 10. Let’s denote the coils A, B and C. There are three different paths that current can flow in order to generate a magnetic field. Let’s pass a current through A and C. Then the small green arrows indicate the generated magnetic flux at each coil and the big arrow is the resulting flux generated from the combined magnetic field from A and C. The rotor will rotate until the south and north pole are aligned with the big arrow, with the north pole pointing at the same direction as the arrow tip. To maintain rotation, the magnetic flux needs to be switched continuously to get the permanent magnet to constantly chase the rotating magnetic field.

![Figure 10. Three phase BLDC motor with one pole pair.](image)

Also from equation Eq. 3.2.7 we concluded that the magnitude of back EMF depends on the angular speed. However, the waveform of the back EMF depends on the arrangement of the windings on the stator. For a BLDC motor, this arrangement contributes to a flux distribution that will produce a trapezoidal back EMF waveform.

### 3.3.1 Mathematical Modelling of BLDC

The voltage of the three phases in a BLDC motor can be defined as
\[
\begin{bmatrix}
V_{an} \\
V_{bn} \\
V_{cn}
\end{bmatrix} = R \begin{bmatrix}
i_a \\
i_b \\
i_c
\end{bmatrix} + L \begin{bmatrix}
di_a \\
di_b \\
di_c
\end{bmatrix} + \begin{bmatrix}
e_a \\
e_b \\
e_c
\end{bmatrix}
\]

where \( V_{an}, V_{bn} \) and \( V_{cn} \) are the stator phase to neutral voltages, \( R \) is the phase to phase resistance, \( i_a, i_b \) and \( i_c \) are the stator phase currents, \( L \) is the phase to phase inductance and \( e_a, e_b \) and \( e_c \) are the induced back EMF. By solving \( \frac{di_{a,b,c}}{dt} \) from Eq. 3.3.1 and taking the Laplace transform we get

\[
\frac{di_a}{dt} = \frac{1}{L} (V_{an} - Ri_a - e_a), \quad i_a = \frac{1}{Ls + R} (V_{an} - e_a)
\]
\[
\frac{di_b}{dt} = \frac{1}{L} (V_{bn} - Ri_b - e_b), \quad i_b = \frac{1}{Ls + R} (V_{bn} - e_b)
\]
\[
\frac{di_c}{dt} = \frac{1}{L} (V_{cn} - Ri_c - e_c), \quad i_c = \frac{1}{Ls + R} (V_{cn} - e_c)
\]

The back EMF can be described with

\[
e_{a,b,c} = k_e \cdot \omega \cdot f_{a,b,c}(\phi)
\]

where \( k_e \) is the back EMF constant, \( \omega \) is the mechanical rotational speed and \( f(\phi) \) is a unit trapezoidal function. The trapezoidal function is defined in Table 1 and produces three 120° phase shifted unit trapezoidal shaped waveforms.

<table>
<thead>
<tr>
<th>( \phi ) [rad]</th>
<th>( f_a(\phi) )</th>
<th>( f_b(\phi) )</th>
<th>( f_c(\phi) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 0 ) - ( \frac{\pi}{3} )</td>
<td>1</td>
<td>-1</td>
<td>( 1 - \frac{6\phi}{\pi} )</td>
</tr>
<tr>
<td>( \frac{\pi}{3} ) - ( \frac{2\pi}{3} )</td>
<td>1</td>
<td>( \frac{6\phi}{\pi} - 3 )</td>
<td>-1</td>
</tr>
<tr>
<td>( \frac{2\pi}{3} ) - ( \pi )</td>
<td>5 - ( \frac{6\phi}{\pi} )</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>( \pi ) - ( \frac{4\pi}{3} )</td>
<td>-1</td>
<td>1</td>
<td>( \frac{6\phi}{\pi} - 7 )</td>
</tr>
<tr>
<td>( \frac{4\pi}{3} ) - ( \frac{5\pi}{3} )</td>
<td>-1</td>
<td>9 - ( \frac{6\phi}{\pi} )</td>
<td>1</td>
</tr>
<tr>
<td>( \frac{5\pi}{3} ) - ( 2\pi )</td>
<td>( \frac{6\phi}{\pi} - 11 )</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

The electromagnetic torque of the motor can be calculated as
\[ T_e = \sum_{n=a,b,c} k_t I_n f_n(\phi) \quad 3.3.4 \]

where \( k_t \) is the torque constant and \( I_n \) is the current in each winding. The motor mechanical motion can be described as

\[ J \frac{d\omega}{dt} + B\omega = T_e - T_L \quad 3.3.5 \]

where \( T_L \) is the torque load, \( J \) is the rotor inertia and \( B \) is the damping constant. Taking the Laplace transform of this equation, the rotor speed as a function of torque can be expressed as

\[ \omega = \frac{1}{Js + B} (T_e - T_L) \quad 3.3.6 \]

To get the rotors electrical rotational speed \( \Omega \) we use the following formula

\[ \Omega = p \cdot \omega \quad 3.3.7 \]

where \( p \) is the number of pole pairs on the rotor.

### 3.4 Field Oriented Control

One way of establishing a rotating magnetic field is through vector control also known as FOC. FOC is based on controlling the current in each phase of the motor so a resulting vector, which represents the magnetic field, is rotating. The reference system in Figure 11 represents three coils separated by 120°. When applying current in phase a, b and c they are producing a magnetic field in each direction. The currents that flow into the motor follow Kirchhoff’s current law

\[ \sum_{k=1}^{3} I_k = i_a + i_b + i_c = 0 \quad \rightarrow \quad i_c = -(i_a + i_b) \quad 3.4.1 \]

So, when looking at the vector \( i_c \) one can see that it has the same magnitude as \( i_a \) and \( i_b \) combined but with opposite sign. The resulting current vector \( i_s \) is the three current vectors \( i_a, i_b \) and \( i_c \) added together. Three sine-waves shifted by 120° generate a smooth rotation of the resulting vector.
Figure 11. The abc-frame of a motor. The a, b and c axis are representing three coils with three current phases. A smooth rotation of the vector $i_s$ is established by three sine-waves shifted by 120°.

The $i_s$ vector is just a two-dimensional vector and can therefore be expressed in a frame with two axes instead of three. The Clarke transform allows us to express the same vector in an $\alpha\beta$-frame instead of the $abc$-frame which is shown in Eq. 3.4.2, Eq. 3.4.3 and Figure 12.

$$\begin{bmatrix} i_\alpha \\ i_\beta \\ i_\gamma \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$  

3.4.2

Since $i_a + i_b + i_c = 0$ the transformation can be simplified to

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$  

3.4.3

Figure 12. $\alpha\beta$-frame of the motor and the transformed signals.
As mentioned earlier, FOC is based on controlling the current in each phase. This can be done by controlling $i_\alpha$ and $i_\beta$ but this is very complex to do since the motor is rotating and the component along the two axis are time varying. It is more convenient to jump onto the motor rotating reference frame, so the components along the axis remains constant. This can be done with the Park transform shown in Eq. 3.4.4 and Figure 13.

$$
\begin{align*}
    i_d &= i_\alpha \cdot \cos(\phi) + i_\beta \cdot \sin(\phi) \\
    i_q &= -i_\alpha \cdot \sin(\phi) + i_\beta \cdot \cos(\phi)
\end{align*}
$$

![Figure 13. Rotating reference system leads to constant components along the axes.](image)

The rotating vector $i_s$ is now reflected in the rotating reference system $dq$. The $d$-axis current is perfectly aligned with the rotor flux and the $q$-axis is quadrature, $90^\circ$ to rotor flux. The only component that produces torque of the rotor is the one that is quadrature to the rotor flux i.e. the current that is aligned with the $q$-axis, $i_q$. The goal is to point the resulting current vector $i_s$ along the $q$-axis to produce maximum torque. In the example in Figure 13 the produced torque is very poor since only $\sim 50\%$ of the magnitude of $i_s$ is reflected onto the $q$-axis. This is where the control of the currents comes in. Both the measured $i_d$ and $i_q$ currents are compared with commanded current values, $i_{q \text{ref}}$ and $i_{d \text{ref}}$, which is based on the demanded rotation of the motor. $i_{d \text{ref}}$ is often set to 0 since it doesn’t contribute to any torque while $i_{q \text{ref}}$ is increased if you want the motor to accelerate and decreased if you want it to deaccelerate. The error signals $\text{error}_q = i_{q \text{ref}} - i_q$ and $\text{error}_d = i_{d \text{ref}} - i_d$ are fed to controllers which are producing error correction voltages $v_d$ and $v_q$, Figure 14.
Figure 14. Commanded currents compared with the measured currents which are fed into a controller that produces correction voltages.

These correction voltages are in the rotating reference frame, but the coils in the motor are in the stationary reference frame. The rotating reference frame voltages need to be converted to the stationary reference frame using inverse Park transformation

\[
\begin{align*}
    v_\alpha &= v_d \cdot \cos(\phi) - v_q \cdot \sin(\phi) \\
    v_\beta &= v_d \cdot \sin(\phi) + v_q \cdot \cos(\phi)
\end{align*}
\]

The \(v_\alpha\) and \(v_\beta\) voltages are then converted into three voltages using inverse Clarke transformation

\[
\begin{bmatrix}
    v_a \\
    v_b \\
    v_c
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & 0 \\
    -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\
    -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 0
\end{bmatrix}
\begin{bmatrix}
    v_\alpha \\
    v_\beta
\end{bmatrix}
\]

These reference voltages are then used in a pulse width modulation (PWM) technique to control the inverter and adjust the voltage in each coil. A two level, three phase inverter consisting of six switches are used to control a three phase motor, Figure 15.

Figure 15. Two level, three phase inverter with six switches connected to a BLDC motor.
The two dominating modulation techniques are sinusoidal PWM (SPWM) and space vector PWM (SVPWM). In SPWM, the three voltages $v_a$, $v_b$, and $v_c$ acquired from the inverse Clarke transformation are compared with a high frequency triangular signal to generate the control signals for the inverter's switches, Figure 16. For example, if the phase voltage $v_a$ is above the high frequency triangular signal then the signal for the switch S1 will be high, Figure 17. The switches are controlled in pairs (S1, S2), (S3, S4), (S5, S6) and are opened and closed in a complementary fashion, i.e. when S1 is open, S2 is closed and vice versa.

Figure 16. High frequency triangular signal and the three phase voltages.

Figure 17. The PWM signals controlling the switches.
The maximum peak phase voltage are either $\frac{V_{dc}}{2}$ or $-\frac{V_{dc}}{2}$ depending on the state of the switches.

The other technique, space vector PWM, is a more advanced PWM technique that provides better utilization of the DC bus voltage and contributes to less switching losses and is therefore often favored [12, 13, 14]. The principle of SVPWM is to take the voltages acquired from inverse Park transformation $v_\alpha$ and $v_\beta$ to determine the reference voltage $v_{ref}$ and the angle $\theta$, i.e.

$$v_{ref} = \sqrt{\left(\frac{v_\alpha^2 + v_\beta^2}{3}\right)}$$

$$\theta = \arctan\left(\frac{v_\beta}{v_\alpha}\right)$$

This is then used to determine in which of the six sectors the rotating reference voltage is currently in, Figure 18.

![Figure 18](image.png)

Figure 18. The possible space vectors. In this case $V_{ref}$ lies between V1 and V2.

The circle inside the hexagon represent the largest output voltage magnitude achieved with SVPWM and is equal to $\frac{V_{dc}}{\sqrt{3}}$. The space vectors $V_0 - V_7$ are representations of the different states of the inverters switches, Table 2.
The goal is to generate a switching pattern that will approximate $v_{ref}$. This can be done by determine the switching durations $T_a$, $T_b$ and $T_0$ using

$$T_a = \frac{\sqrt{3} \cdot T_s \cdot V_{ref}}{V_{dc}} \left( \sin \left( \frac{\pi}{3} - \theta + \frac{n-1}{3} \pi \right) \right), \quad n = 1, 2, ..., 6$$

$$T_b = \frac{\sqrt{3} \cdot T_s \cdot V_{ref}}{V_{dc}} \left( \sin \left( \theta - \frac{n-1}{3} \pi \right) \right), \quad n = 1, 2, ..., 6$$

$$T_0 = T_s - T_a - T_b$$

where $T_s$ is the sampling time and $n$ is the sector number. Using Table 2 and the switching time durations, the switching sequence can be generated. For example, if the reference voltage is inside sector 1, i.e. between V1 and V2, then V0 will be applied for duration of $\frac{T_0}{4}$, then V1 for $\frac{T_a}{2}$, then V2 for $\frac{T_b}{2}$, then V7 for $\frac{T_0}{4}$ and then the sequence is reversed, Table 3. The same strategy can be used to get the switching pattern for the remaining sectors as well. Once the switching pulse pattern is known, the correct output phase voltage of the inverter can be applied.

Table 2. Space vectors with representative states and switches.

<table>
<thead>
<tr>
<th>Space vector</th>
<th>State</th>
<th>On switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0</td>
<td>[000]</td>
<td>S2, S4, S6</td>
</tr>
<tr>
<td>V1</td>
<td>[100]</td>
<td>S1, S4, S6</td>
</tr>
<tr>
<td>V2</td>
<td>[110]</td>
<td>S1, S3, S6</td>
</tr>
<tr>
<td>V3</td>
<td>[010]</td>
<td>S2, S2, S6</td>
</tr>
<tr>
<td>V4</td>
<td>[011]</td>
<td>S2, S3, S5</td>
</tr>
<tr>
<td>V5</td>
<td>[001]</td>
<td>S2, S4, S4</td>
</tr>
<tr>
<td>V6</td>
<td>[101]</td>
<td>S1, S4, S5</td>
</tr>
<tr>
<td>V7</td>
<td>[111]</td>
<td>S1, S3, S5</td>
</tr>
</tbody>
</table>

Table 3. Switching pattern for the six sectors.

<table>
<thead>
<tr>
<th>Sector</th>
<th>$T_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\frac{T_0}{4}$</td>
</tr>
<tr>
<td>1</td>
<td>V0</td>
</tr>
<tr>
<td>2</td>
<td>V0</td>
</tr>
<tr>
<td>3</td>
<td>V0</td>
</tr>
<tr>
<td>4</td>
<td>V0</td>
</tr>
<tr>
<td>5</td>
<td>V0</td>
</tr>
<tr>
<td>6</td>
<td>V0</td>
</tr>
</tbody>
</table>

FOC can be summarized with the following steps and Figure 19.

- Measure the current in each phase in the motor
- Transform the measured currents to rotating reference system
  - Clarke transform
- Park transform
- Compare the measured current with desired current
- Take the error into a controller and generate correction voltages; \( V_d, V_q \)
- Transform the correction voltage to stationary reference frame
  - Inverse Parke transform
- Apply the correction voltages on each phase using a PWM technique

![Block diagram of Field Oriented Control topology.](image)

Figure 19. Field Oriented Control topology.

### 3.5 Gears

A motor is often connected to a set of gears or a gear train. The ratio between two gears are defined as

\[
R = \frac{\omega_1}{\omega_2} = \frac{\dot{\omega}_1}{\dot{\omega}_2} = \frac{r_2}{r_1} = \frac{n_2}{n_1}
\]

where \( \omega \) is the angular velocity, \( \dot{\omega} \) is the angular acceleration, \( r \) is the radius, and \( n \) is the number of teeth. In the most common case when there are multiple gears attached in series the gear ratio is

\[
\frac{n_2}{n_1} \cdot \frac{n_3}{n_2} \cdot \frac{n_4}{n_3} \cdots \cdot \frac{n_i}{n_{i-1}} = \frac{n_i}{n_1}
\]

which shows that the equation can be reduced by only looking at the first and the last gear. However, when two gears are mounted on the same shaft as in Figure 20, one need to consider all the steps in the gear train to get the correct gear ratio.
Denoting the red gear as gear 1, blue as gear 2, yellow as gear 3 and green as gear 4, the transformation in angular velocity can be calculated in the following steps

\[
\omega_1 = \frac{n_2}{n_1} \omega_2 = \{\omega_2 = \omega_3\} = \frac{n_2}{n_1} \omega_3
\]

\[
\omega_3 = \frac{n_4}{n_3} \omega_4
\]

\[
\omega_1 = \frac{n_2 n_4}{n_1 n_3} \omega_4
\]

with the gear ratio

\[
R = \frac{n_2 n_4}{n_1 n_3}
\]

Recall that the torque equation of a motor is

\[
T_e = J_e \frac{d\omega}{dt} + B_e \omega + T_L
\]

The gear train contributes as a part of the torque load \(T_L\) to the motor due to inertia, \(J\) and damping, \(B\) in each gear. Suppose that gear 1 is the driving gear i.e. the gear that is connected to the motor shaft, and that an external force \(F_{\text{ext}}\) is applied to gear 4 as shown in Figure 20. The total torque from the external force and gear 2, 3, 4 acting as a load to gear 1 can be defined by writing a free body diagram, looking at the forces in each connection and describe the torque equation in each gear. The torque equation for gear 4 is

\[
J_4 \cdot \dot{\omega}_4 = F_{\text{ext}} \cdot r_4 - F_{34} \cdot r_4 - B_4 \omega_4
\]

Due to Newton’s third law, the forces between gear 4 and gear 3 can be expressed as

\[
F_{34} = F_{43}
\]

and due to the gear ratio, the angular velocity and acceleration can be expressed as
\( \omega_4 = \frac{n_3}{n_4} \cdot \omega_3 \)  \hspace{1cm} 3.5.8

\( \dot{\omega}_4 = \frac{n_3}{n_4} \cdot \dot{\omega}_3 \)  \hspace{1cm} 3.5.9

By rewriting Eq. 3.6.6 and putting in Eq. 3.6.7, Eq. 3.6.8 and Eq. 3.6.9, the force from gear 4 applied to gear 3 is

\[
F_{43} = \frac{1}{r_4} \left( F_{\text{ext}} \cdot r_4 - \frac{n_3}{n_4} \left( B_4 \cdot \omega_3 + J_4 \cdot \dot{\omega}_3 \right) \right)
\]  \hspace{1cm} 3.5.10

Gear 2 and 3 are connected on the same shaft and thus have the same torque, \( \omega \) and \( \dot{\omega} \). The inertia is \( J_{2,3} = J_2 + J_3 \) and the damping is \( B_{2,3} = B_2 + B_3 \). The torque equation for gear 2 and 3 is

\[
J_{2,3} \cdot \dot{\omega}_{2,3} = F_{43} \cdot r_3 - F_{12} \cdot r_2 - B_{2,3} \cdot \omega_{2,3}
\]  \hspace{1cm} 3.5.11

Newton’s third law and the gear ratio between gear 1 and 2 is used and the equation is rewritten to

\[
F_{21} = \frac{1}{r_2} \left( F_{43} \cdot r_3 - \frac{n_1}{n_2} \left( B_{2,3} \cdot \omega_1 + J_{2,3} \cdot \dot{\omega}_1 \right) \right)
\]  \hspace{1cm} 3.5.11

Finally, the torque that is applied to gear 1 from external forces and from gear 2, 3 and 4 due to inertia and damping is

\[
T_L = F_{21} \cdot r_1
\]  \hspace{1cm} 3.5.12

### 3.6 Springs

The equation that describes the motion of an object that is connected to a spring is

\[
m \frac{d^2 x}{dt^2} + c \frac{dx}{dt} + k x = 0
\]  \hspace{1cm} 3.6.1

which is the equation for the damped harmonic oscillator without any external force. Here, \( m \) is the mass of the object, \( c \) is the viscous damping coefficient, \( k \) is the spring constant and \( x \) is the displacement from the equilibrium position, \( x_0 \). Consider a system as in Figure 21 where a spring is connecting the outer frame (the base) with an object. The base is subjected to vibrations which cause the object to move. This is known as base excitation.
The force that is applied to the object is the force from the spring, the movement of the base and the frictional force from the surface between the object and the base. The motion of the object can be expressed as

\[ m \left( \frac{d^2x}{dt^2} + \frac{d^2y}{dt^2} \right) + c \frac{dx}{dt} + kx + S\text{gn}\left( \frac{dx}{dt} \right) F_f = 0 \]  \hspace{1cm} (3.6.2)

\[ m \left( \frac{d^2x}{dt^2} \right) + c \frac{dx}{dt} + kx + S\text{gn}\left( \frac{dx}{dt} \right) F_f = -m \frac{d^2y}{dt^2} \]  \hspace{1cm} (3.6.3)

where \( x \) is the displacement from the equilibrium position relative to the base, \( \frac{d^2y}{dt^2} \) is the acceleration of the base, \( F_f \) is the frictional force and \( S\text{gn}\left( \frac{dx}{dt} \right) \) is the direction the object is moving relative to the base. The object starts to move when the external force exceeds the static frictional force \( F_f = \mu_s m g \), where \( \mu_s \) is the coefficient of static friction and \( g \) is the gravitational acceleration. Once this is fulfilled the frictional force is defined as \( F_f = \mu_k m g \), where \( \mu_k \) is the coefficient of kinetic friction and points at the opposite direction of \( \frac{dx}{dt} \). Two or more springs that are connected in parallel as in Figure 22 acts as a single spring with the spring constant \( K = K_1 + K_2 + K_3 + K_4 \).

### 3.7 Vibration Test

To be able to qualify products to meet certain standards they must undergo a variation of tests. These tests are also a great way to evaluate the products performance and to detect deficiencies and
shortcomings. MIL-STD-810G is one of the standards that provides a wide variety of tests that are specifically designed to simulate and replicate the environmental conditions the product might experience during its service life. One of these tests is the vibration test where the product is placed on a vibration test shaker which introduces vibrations to the product. During a traditional vibration test where the tested item is exposed to sine signals that vary in frequency, amplitude and phase. While it might not be the best representation of real world vibrations but is useful when studying resonance responses since it excites only one frequency at a time. Random vibration test on the other hand, are non-periodic and composed of a continuous spectrum of frequencies. This makes it more similar to real world conditions since it excites all resonant frequencies simultaneously. It also makes it much harder to analyze considering its absolute value can’t be calculated at a specific instant. However, the probability of the occurrence of a particular amplitude can be predicted using sufficient knowledge of the past behavior of the vibrations. Random vibration can be represented in the frequency domain by a power spectral density function (PSD) and since it is a statistical measurement of the motion, that is the result of an averaging process, an infinite number of different signals in the time domain could have generated the same PSD. Essentially what it describes is how the power of the vibration signal is distributed over a range of frequencies and can be defined as a set of amplitude breakpoints at given frequencies, Figure 23.

![Figure 23. An example of a PSD graph.](image)

The unit of the Y-axis is root mean square acceleration per unit bandwidth $G_{RMS}^2/Hz$ or $G^2/Hz$ for the sake of brevity. Given the amplitude breakpoints in Figure 23, the overall $G_{RMS}$ value can be
calculated by taking the square root of the area under the PSD curve. At first glance this might appear trivial, but the calculations are not completely straightforward since it’s a log-log graph. The area under a straight line between two breakpoints, \((f_1, a_1)\) and \((f_2, a_2)\), can be calculated using following equations

\[
slope = \frac{\log \left( \frac{a_2}{a_1} \right)}{\log \left( \frac{f_2}{f_1} \right)}, \quad offset = \frac{a_1}{f_1^{slope}}, \quad area = \frac{offset}{slope + 1} \left( f_2^{slope+1} - f_1^{slope+1} \right)
\]

So for example between \((f_1 = 20, a_1 = 0.01)\) and \((f_2 = 100, a_2 = 0.04)\) from Figure 23, we get \(area = 2.04G^2\). The same equations follow for each pair of breakpoints and finally we take the square root of the sum of the individual areas to get the overall \(G_{RMS}\) value

\[
\sqrt{2.04 + 12 + 22.38} = 6.03G_{RMS}
\]

With the \(G_{RMS}\) value it’s possible to predict the average number of times a certain acceleration amplitude might occur in a given duration. This can easily be done if the random vibration is assumed to be Gaussian distributed, which means that values within one standard deviation (\(\sigma\)) of the mean accounts for 68.27% of the total number of samples. So for example let’s use 6.03 \(G_{RMS}\), it’s equal to \(\sigma\) when the mean is zero, then the distribution of particular amplitudes is as Table 3 demonstrates.

<table>
<thead>
<tr>
<th>(\sigma = 6.03 G_{RMS})</th>
<th>Probability Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-\sigma &lt; x &lt; \sigma)</td>
<td>68.27%</td>
</tr>
<tr>
<td>(-2\sigma &lt; x &lt; 2\sigma)</td>
<td>95.45%</td>
</tr>
<tr>
<td>(-3\sigma &lt; x &lt; 3\sigma)</td>
<td>99.73%</td>
</tr>
<tr>
<td>(-4\sigma &lt; x &lt; 4\sigma)</td>
<td>99.99%</td>
</tr>
</tbody>
</table>

An arbitrary time series of the random vibration signal can be generated from the PSD with 6.03 \(G_{RMS}\) for a desired duration. In Figure 24 this has been done for a 2 second duration and the signal has a 68.27% tendency to remain within \(\pm 6.03G\) and on very rare occasions exceeding \(\pm 18.09G\).
During vibration tests the shaker applies random vibration from a given PSD to the test item for a given time and is then monitored by accelerometers that provide feedback to the shaker. Amplitudes outside the $\pm 3\sigma$ limit are neglected because of design purposes which removes the most damaging high peak accelerations during test simulation. This might implicate long fatigue damage simulations but is still a valid assumption considering the amplitudes outside $\pm 3\sigma$ limit have a probability of appearing only 0.27% of the total data points.
4 Methodology

4.1 Simulink

Simulink is a graphical programming environment for simulating, modeling and analyzing dynamical system. This gives the opportunity to investigate the behavior of the system and to validate it before moving to hardware. Since Simulink is developed by MathWorks, it enables tight integration with the rest of the MATLAB environment and comes with a broad collection of predefined building blocks which can be dragged and dropped and eventually connected to perform a specific task [18].

4.1.1 Overview of the System

The entire system in Simulink is shown in Figure 25, where all the vital components are defined in subsystems. The Reference Speed subsystem calculates the desired angular velocity on the BLDC motor and allows the ability to plan different motion sequences. The Controller subsystem takes in the desired angular velocity and the motors actual angular velocity and outputs a voltage signal for the motor based on the input information and this is where the PID controller, ADRC, LADRC or FLC are implemented. The FOC & BLDC subsystem contains the entire FOC and the BLDC motor which is explained in Section 4.1.2. The four exchangeable controllers are only considered for regulating the speed signal. The two current controllers within the FOC & BLDC subsystem remain unchanged during all of the simulations in order to reduce the implementation and tuning time.

The Gears & Axles subsystem, reviewed later in Section 4.1.3, works in both directions. It takes in the motors velocity and acceleration and uses it in equations through the gear train in order to calculate the velocity of the pick axles, but it also takes in the disturbance applied on the pick axles and uses it in equations to get the torque load leading back to the motor. Lastly, there is the Vibrations & Stack & Springs subsystem which consists of the vibrations, functions which controls the movement of the stack and also the characteristics and the connections of the springs. The output from this subsystem is the disturbance from the stack due to strong vibrations and is explained in greater detail in Section 4.1.4 & 4.1.5. Also, there are two memory blocks and four scopes in the model. The memory blocks are there to delay the feedback signal with two time steps because of design reasons and the scopes provide the ability to view the signals in a graph plotted against the simulation time. The exact Simulink files, showing all of the details, can be found in Attachment 3.
4.1.2 Field Oriented Control & BLDC

The entire structure of FOC with two consistent PID controllers is displayed in Figure 26 and was implemented based on Section 3.4. The current on each phase in the motor is measured and fed into the Clarke transformation, which are based on Eq. 3.4.3, and later fed into the Park transformation, Eq. 3.4.4. The measured and transformed current signals are then fed into the two PID controllers, which control the voltages $v_d$ and $v_q$. The voltages are then transformed back to the stationary reference frame and fed into the SVPWM which is based on the end of Section 3.4. The three phase voltages from the SVPWM block are then applied to the BLDC motor. The BLDC subsystem has been modelled in Simulink according to the equations in Section 3.3.1 and is shown in Figure 27. The exact motor parameters are presented in Attachment 2.
4.1.3 Gears & Axles

The motor is driving the two pick axles through the gear train in Figure 2, Section 2 and was modelled according to the equations in Section 3.5. The numbered subsystems in Figure 28 stand for a certain gear or axle and the exact equations and parameters such as inertia, damping, radius, mass and number of teeth are presented in Attachment 1. The input disturbance is the resulting disturbance from the stack, reviewed in the next section, and is assumed to be divided equally on the two pick axles.
4.1.4 Springs & Stack

The four springs in parallel are modelled in Simulink according to equations in Section 3.6, where the stack is the object that is connected to the springs. The stack is assumed to be a solid body with no internal damping but would in reality contribute with damping and other uncertainties. The vibrations, reviewed in the next section, are affecting the base of the springs which in return are affecting how the stack is moving with reference to the base. However, the movement of the stack is restricted since it’s contained in a confined area. This confined area, which is defined in the Springs & Stack subsystem, also provides resistance to the stacks movability in the form of friction. The stacks resulting force and displacement are converted to the disturbance that affects the two pick axles. The connection from vibrations to the resulting disturbance that is sent to the axles is shown in Figure 29. The exact spring parameters are presented in Attachment 5.

![Simulink model of vibrations, springs and stack.](image)

4.1.5 Vibrations/Disturbances

The disturbances that affect the springs and the stack in the simulations originates from a MIL-STD-810G vibration test performed on the real system. The used data was presented as a PSD graph and comes from measured values from a sensor that was placed on the dispenser during the test. The data points from the sensor were loaded into MATLAB and used in a MATLAB script (Attachment 3) to generate a time series with acceleration as a function of time. The time series containing over 250,000 data points was transferred from MATLAB to Simulink and applied to the modelled system. Worth mentioning here is that because the vibrations are random, the time series will be slightly different every time it’s generated. This behavior is of course desired but might be misleading when it comes to the comparison of the controllers. Therefore, the same time series data is used when the controllers are compared against each other but during other simulations different data is used to cover as many outcomes as possible.

4.1.6 PID Controller

The PID controller is very simple and so widely used that Simulink supports an already implemented PID controller block with advanced features such as the ability to tune the PID gains automatically with a given input and output. This is very helpful considering it also offers the ability to follow the response time while adjusting the gains. The D term of the PID controller is completely disregarded in this thesis since there are substantial disturbances acting on the system and the D term is known to amplify such disturbances. Another reason is that the D term sometimes introduces numerical errors...
during simulations with fixed time step in Simulink. The PI controller block and how it is connected to the motor is shown in Figure 30.

![Figure 30. PI controller connected to the BLDC motor.](image)

### 4.1.7 Active Disturbance Rejection Control

The ADRC controller was implemented in Simulink according to Section 3.1.2 and is shown in Figure 31. The input of the TD is the reference speed and the outputs are calculated with Eq. 4.1.2.12. The ESO is designed according to Eq. 4.1.2.6 and produces an estimation of the motors output and the total disturbance which is used in the NLSEF that is based on Eq. 4.1.2.10. These three parts together with the estimated gain $b_0$ forms the ADRC controller algorithm. To demonstrate what Eq. 3.1.2.12 would look like in Simulink, the ESO subsystem is shown in Figure 32 where the MATLAB functions $fal1$ and $fal2$ are based on Eq. 3.1.2.7.

![Figure 31. Simulink model of the ADRC controller.](image)
The tuning process of ADRC turned out to be very complicated considering there are so many parameters to tune and adjust. In the ESO alone there are three observer gains $\beta_1$, $\beta_2$ and $\beta_3$ and two parameters $\alpha_1$ and $\alpha_2$ to adjust and fine-tune for optimal results. On the upside however, many adjustable parameters enable more combinations and freedom in the tuning process, which might yield a better result if done correctly. Unfortunately, no apparent combinations or relationships between the parameters for this particular system were found, except those mentioned in Section 3.1.2, and the tuning consisted of trial and error.

4.1.8 Linear Active Disturbance Rejection Control

As mentioned in Section 3.1.3 the LADRC transforms the non-linear parts of the traditional ADRC into linear ones and its total structure is shown in Figure 33. The LTD takes in the reference speed and calculates the tracking signal using Eq. 4.1.3.11 & Eq. 3.1.3.12. The LESO is designed according to Eq. 4.1.3.5 & Eq. 3.1.3.6 and outputs estimations based on the angular velocity of the motor and the control signal. This is then used in the PD controller which is based on Eq. 3.1.3.9 & Eq. 3.1.3.10. These three parts combined concludes the LADRC algorithm.
There are three parameters that are needed to take in consideration when tuning the LADRC, the tracking speed $r$ in the LTD, the observer bandwidth $w_0$ in the LESO and the controller bandwidth $w_c$ in the PD controller. The parameter $r$ is mainly used to soften the control signal and the larger $r$ is, the faster $v_1$ tracks the input signal $v_0$. This means that $r$ can be increased until the impact of signal mutation becomes apparent. For the tuning process of the observer bandwidth $w_0$, there are three limiting factors: sampling rate, sensor noise and dependency on the model of the plant. Generally, $w_0$ should be set as high as the hardware and software limitations allows and are usually somewhere between $w_0 = 5\sim 10r$. After an appropriate $w_0$ has been selected, a common rule of thumb to follow is $w_0 = 3\sim 5w_c$, where $w_c$ is the bandwidth of the controller. After the three parameters have been set using the above procedure, different design trade-offs can be made by incrementally increasing or decreasing the parameters individually. [9]

### 4.1.9 Fuzzy Logic Controller

A FLC was implemented in Simulink as shown in Figure 34 & 35. The inputs to the controller are the error of the angular velocity and the change in error in angular velocity, whereas the output of the controller is the change in voltage.
After going through the steps described in Section 3.1.4, the membership functions, the rule base, the inference engine and defuzzifier were carefully chosen. The membership functions are applied on the inputs and are described in Table 4 and Figure 36 & 37.
Table 4: Definition of range of the membership functions of which the output is non-zero.

<table>
<thead>
<tr>
<th>Error</th>
<th>Range</th>
<th>Change in Error</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative Large, NL</td>
<td>(-∞ - 200)</td>
<td>NL</td>
<td>(-∞ -6)</td>
</tr>
<tr>
<td>Negative Medium, NM</td>
<td>(-300 -100)</td>
<td>NM</td>
<td>(-9 -3)</td>
</tr>
<tr>
<td>Negative Small, NS</td>
<td>(-200 0)</td>
<td>NS</td>
<td>(-6 0)</td>
</tr>
<tr>
<td>Zero, Z</td>
<td>(-100 100)</td>
<td>Z</td>
<td>(-3 3)</td>
</tr>
<tr>
<td>Positive Small, PS</td>
<td>(0 200)</td>
<td>PS</td>
<td>(0 6)</td>
</tr>
<tr>
<td>Positive Medium, PM</td>
<td>(100 300)</td>
<td>PM</td>
<td>(3 9)</td>
</tr>
<tr>
<td>Positive Large, PL</td>
<td>(200 ∞)</td>
<td>PL</td>
<td>(6 ∞)</td>
</tr>
</tbody>
</table>

Figure 36. Membership functions for Error in angular velocity.

Figure 37. Membership functions for change in error in angular velocity.
The rule base has changed during the work, but it is based from the beginning on the reasoning that if the error is positive, i.e. \( \omega_{ref} > \omega \), then the input to the motor, the voltage, should be increased and vice versa if the error is negative. The other input, change in error, wasn’t as intuitively as the error but with the reasoning that if the error is zero and that the change in error is positive, i.e. \( E(k) > E(k - 1) \), then it means that in this timestep, the error is bigger than the last timestep. Since the error is zero it also means that the error in the last timestep was negative which means that the slope of the curve, \( \omega \), is negative, hence the voltage should be increased to compensate for the negative change. Taking these conclusions into consideration and with a trial and error method, a final rule base was concluded. Since the membership functions are defined in seven different ways for both inputs, the rule base consists of the combined \( 7 \cdot 7 = 49 \) rules and is described in the following list and Table 5.

- Rule 1: IF error IS negative large AND dError IS negative large THEN output IS negative large.
- Rule 2: IF error IS negative large AND dError IS negative medium THEN output IS negative large.
- ...
- Rule 25: IF error IS zero AND dError IS zero THEN output IS zero.
- Rule 26: IF error IS zero AND dError IS positive small THEN output IS positive small.
- ...
- Rule 48: IF error IS positive large AND dError IS positive medium THEN output IS positive large.
- Rule 49: IF error IS positive large AND dError IS positive large THEN output IS positive large.

<table>
<thead>
<tr>
<th>dError/error</th>
<th>NL</th>
<th>NM</th>
<th>NS</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>1 NL</td>
<td>8 NL</td>
<td>15 NL</td>
<td>22 NL</td>
<td>29 NS</td>
<td>36 Z</td>
<td>43 PS</td>
</tr>
<tr>
<td>NM</td>
<td>2 NL</td>
<td>9 NL</td>
<td>16 NL</td>
<td>23 NM</td>
<td>30 Z</td>
<td>37 PS</td>
<td>44 PM</td>
</tr>
<tr>
<td>NS</td>
<td>3 NL</td>
<td>10 NL</td>
<td>17 NS</td>
<td>24 NS</td>
<td>31 PS</td>
<td>38 PM</td>
<td>45 PL</td>
</tr>
<tr>
<td>Z</td>
<td>4 NL</td>
<td>11 NM</td>
<td>18 NS</td>
<td>25 Z</td>
<td>32 PS</td>
<td>39 PM</td>
<td>46 PL</td>
</tr>
<tr>
<td>PS</td>
<td>5 NL</td>
<td>12 NM</td>
<td>19 NS</td>
<td>26 PS</td>
<td>33 PM</td>
<td>40 PL</td>
<td>47 PL</td>
</tr>
<tr>
<td>PM</td>
<td>6 NM</td>
<td>13 NS</td>
<td>20 Z</td>
<td>27 PM</td>
<td>34 PL</td>
<td>41 PL</td>
<td>48 PL</td>
</tr>
<tr>
<td>PL</td>
<td>7 NS</td>
<td>14 Z</td>
<td>21 PS</td>
<td>28 PL</td>
<td>35 PL</td>
<td>42 PL</td>
<td>49 PL</td>
</tr>
</tbody>
</table>

The rules are implemented in the inference engine where Mandani’s minimum operator is used and is defined by

\[
\mu_{E \cap dE}(x) = \min(\mu_E(x), \mu_{dE}(x))
\]

where \( \mu_E(x) \) and \( \mu_{dE}(x) \) is the truth value of error and change in error respectively for a given \( x \) i.e. the outputs from the membership functions.

**Example:** Let’s say that the error is -75 rad/s and the change in error is -2 rad/s. The error goes into the membership functions in Figure 36 and the truth value, \( \mu_E(-75) \), for negative small and zero is 0.75 and 0.25 respectively. The change in error goes into the membership functions in Figure 37 and the truth value, \( \mu_{dE}(-2) \), for negative small and zero is \( \sim 0.67 \) and \( \sim 0.33 \) respectively. The only
rules that gives a truth value different from zero is rule 17, 18, 24 and 25. The combined truth value \( \mu_{E \cap dE} \) for each rule is then:

- Rule 17: \( \mu_{17} = \mu_{E \cap dE}(x) = \min(\mu_E(x), \mu_{dE}(x)) = \min(0.75, 0.67) = 0.67 \)
- Rule 18: \( \mu_{18} = \min(0.75, 0.33) = 0.33 \)
- Rule 24: \( \mu_{24} = \min(0.25, 0.67) = 0.25 \)
- Rule 25: \( \mu_{25} = \min(0.25, 0.33) = 0.25 \)

with the outputs

- Rule 17: Output(17) = NM
- Rule 18: Output(18) = NS
- Rule 24: Output(24) = NS
- Rule 25: Output(25) = Zero

To get a real crisp output from this information, values for each output NL, NM, NS, Z, PS, PM and PL is defined as

- NL = -3
- NM = -1.5
- NS = -0.5
- Z = 0
- PS = 0.5
- PM = 1.5
- PL = 3

These values are then combined with the truth values in the defuzzifier with the centroid method defined by

\[
\sum_{i=1}^{n} \frac{\text{output}(i) \cdot \mu_i}{\mu_i} = \frac{\text{output}(17) \cdot \mu_{17} + \text{output}(18) \cdot \mu_{18} + \text{output}(24) \cdot \mu_{24} + \text{output}(25) \cdot \mu_{25}}{\mu_{17} + \mu_{18} + \mu_{24} + \mu_{25}} \]

which for the given example is

\[
\frac{-1.5 \cdot 0.67 - 0.5 \cdot 0.33 - 0.5 \cdot 0.25 - 0.25}{0.67 + 0.33 + 0.25 + 0.25} = -0.8633.
\]

which is the output, \( dU \), of the controller. In the next timestep the output is added on the current Voltage, \( U(k + 1) = U(k) + dU \) and fed into the motor.
4.2 The Pick Motion

Before the result is covered, the pick motion that is under investigation has to be reviewed in greater detail to be able to understand the outcomes of the motion. Recall the overview of the system in Section 2. It describes the pick motion where an expendable is separated from the stack and then sent to the eject phase. The motion of the pick axles can be divided into three phases in a period of $2\pi$ i.e. one turn. Denoting the rotating angle of the pick axles as $\alpha$, the opening angle of the recesses as $\theta_1$, and the closing angle of the recesses as $\theta_2$, the phases during perfect conditions with no disturbances can be defined as

- **Phase 1: $\alpha = [0 \ \theta_1]$**
  The stack is pushing on the axles due to the initial compression of the springs, which causes a friction between the expendable and the axles. This friction is seen as a resisting torque load to the motor which forces the motor to work harder.

- **Phase 2: $\alpha = (\theta_1 \ \theta_2)$**
  In this phase an expendable can fall into the recesses and be separated from the stack. During this phase the stack is pushed forward both by the initial compression of the springs plus by an external pushing mechanism to ensure that a separation will be performed. This affects the motor with a torque load with opposite sign compared with the friction i.e. it “helps” the motor to rotate.

- **Phase 3: $\alpha = [\theta_2 \ 2\pi]$**
  This phase completes a full turn of the pick motion and the conditions during this phase are exactly the same as phase 1, i.e. friction as resisting torque.

As mentioned earlier, this is during perfect conditions and demonstrates the wanted behavior of each phase. However, when the dispenser is under strong vibrations it can cause the stack to be displaced from the pick axles. This happens when the force due to vibrations overcomes the initial compression force during phase 1 and 3, and during phase 2 it has to overcome the initial compression force plus the external pushing mechanism force. This is of course an unwanted behavior since the pick axles are unable to grab and separate an expendable from the stack which will result in an unsuccessful ejection of the expendable.
5 Result

The Result section consists of simulations of the pick motion during different conditions. Relevant information such as angular velocity, position, torque, force and disturbance are presented. All of the graphs covering the velocity and the position of the pick axle have negative Y-axis because of design purposes originated from Saab. All X-axis and some Y-axis are censored due to company restricted information. The uncensored version of the Result is presented in Attachment 4.

5.1 Pick Motion without Vibrations

The first motion reviewed is the pick motion using a PI controller and with no vibrations acting on the system. The reference velocity and the actual velocity of the pick axles can be seen in Figure 38, where the impact on the motion from the pick phases described in Section 4.2 becomes apparent. First the motor tries to accelerate to follow the reference velocity but inertia and damping from the gear train and friction on the pick axles is resisting. Then the recesses of the pick axles appear and a sudden torque shock helps the motor which results in a sudden overshoot in the velocity and the motor needs to compensate quickly. Then the separation of expendables happens and the recesses close which again resist the velocity but this time the motor want to deaccelerate. The position of the pick axles can be seen in Figure 39, where the axles complete a full turn, i.e. $-2\pi$. Also, the displacement of the stack can be seen in Figure 40, where zero on the Y-axis indicates that the stack lies against the pick axles and the sudden positive displacement is when the recesses open and the stack is pushed in and an expendable is separated. After the separation, the recesses close and the position of the stack is reset back to zero. The last graph reviewed for this particular motion is Figure 41, where the torque acting on the pick axles and the torque leading back to the motor can be seen. This might be the most interesting graph since it reveals the interaction between the motor and the pick axles through the gear train. It also reveals that in the beginning the torque is increasing until it overcomes the static friction from the stack, then the kinetic friction is acting instead. After the pick motion is done, the torque is zero since the pick axles is not moving and therefore not contributing to any torque.
Figure 38. The reference angular velocity and the actual angular velocity of the pick axles during the pick motion.

Figure 39. The reference angular position and the actual angular position of the pick axles during the pick motion.
Figure 40. The displacement of the stack relative to the pick axles. Positive position indicating that it is pushed into the recesses and negative position indicating that the stack is displaced away from the axles.

Figure 41. The torque load acting on the pick axles and the torque load leading back to the motor.
5.2 Pick Motion with Vibrations

The next motion is still the pick motion with a PI controller but this time the system is exposed to strong vibrations. In Figure 42 & 43 the pick axle’s velocity and position can be seen. They still follow the same pattern as before for the same reasons but this time with vibrations causing the velocity’s shakiness. In Figure 44 the displacement of the stack can be seen and the positive displacement indicates that the stack is pushed into the recesses and separates an expendable. The negative displacement indicates that the stacks is actually letting go of the pick axles caused by the vibrations. This can also be seen in Figure 45 where the torque on the pick axles becomes zero every time the stack is letting go. The segment of the vibrations acting on this motion can be seen in Figure 46. The resulting force affecting the stack due to the vibrations and the initial compression and external compression force can be seen in Figure 47 and the resulting acceleration of the stack can be seen in Figure 48. The stack can only have positive acceleration when the recesses appear or when the displacement of the stack is negative. This is because the stack is contained in a confined area and has limited movability.

![Figure 42. The reference angular velocity and the actual angular velocity of the pick axles during the pick motion with vibrations.](image-url)
Figure 43. The reference angular position and the actual angular position of the pick axles during the pick motion with vibrations.

Figure 44. The displacement of the stack relative to the pick axles. Positive position indicating that it is pushed into the recesses and negative position indicating that the stack is displaced away from the axles.
Figure 45. The torque load acting on the pick axles and the torque load leading back to the motor.

Figure 46. The time series of the vibrations during a successful separation.
Figure 47. The resulting force on the stack due to vibrations and initial and external compression force.

Figure 48. The stacks acceleration due to the resulting force and its bound movability.
5.3 Failed Separation due to Vibrations

This time the pick motion is still executed with a PI controller but with another section of the vibrations data. The vibrations during this motion lead to a failed separation of an expandable because the resulting force cause the stack to be displaced from the pick axles when the recesses appear. This might not be evident in Figure 49 & 50 where the pick axle’s velocity and position follow the reference in a good manner. It’s in Figure 51 where it becomes clear that the stack is displaced away from the axles during the moment when the recesses appear (indicated with the red arrow). Also in Figure 52, the torque load on the axles is zero except when the stack comes back to the pick axles and then contributes with resisting torque. The segment of the vibration data used during this motion is shown in Figure 53 and the resulting force acting on the stack is shown in Figure 54. In Figure 55 the stacks acceleration can be seen. The negative acceleration in the beginning is causing the displacement of the stack and the positive acceleration thereafter is pushing back the stack to the axles but too late to be able to grab and separate an expandable since the moment of the recesses has passed.

![Graph](image)

**Figure 49.** The reference angular velocity and the actual angular velocity of the pick axles during the pick motion with vibrations.
Figure 50. The reference angular position and the actual angular position of the pick axles during the pick motion with vibrations.

Figure 51. The displacement of the stack relative to the pick axles. Positive position indicating that it is pushed into the recesses and negative position indicating that the stack is displaced away from the axles. Red arrow indicating when the separation is supposed to happen.
Figure 52. The torque load acting on the pick axles and the torque load leading back to the motor.

Figure 53. The time series of the vibrations during a failed separation.
Figure 54. The resulting force on the stack due to vibrations and initial and external compression force.

Figure 55. The stacks acceleration due to the resulting force and its bound movability.
5.4 Comparison of the Controllers without Vibrations

The four different controllers are simulated during the same conditions and plotted against each other in the same graph to be able to compare the performance and determine which of the controllers would be most suitable for the task. The first motion that is considered is the pick motion without vibrations, which gives a good indication on how closely the controllers can follow the reference signal during the separation of an expendable. All the controllers follow the reference velocity of the pick axles in Figure 56 in a relative nice way but during the separation window there are a sudden change to the velocity and ADRC and LADRC proves to be the most robust controllers in this window. In Figure 57 the angular position of the pick axles is presented and all of the controllers are following the reference signal in a nice manner.

Figure 56. Angular velocity of the pick axles during the pick motion without vibrations.
5.5 **Comparison of the Controllers with Vibrations**

The controllers are again compared during the pick motion but this time with vibrations acting on the system. In Figure 58 the angular velocity of the pick axles can be seen and in Figure 59 the angular position is presented. The vibrations are affecting the velocity and the ADRC are overcompensating slightly in this case, which also makes its position lag behind the others. The FLC is reacting in such a way that causes a small gain in its position. The LADRC and the PI controller are marginally better at compensating for the disturbance and therefore might be favored over the others.
Figure 58. Angular velocity of pick axles during the pick motion with vibrations.

Figure 59. Angular position of pick axles during the pick motion with vibrations.
5.6 Displacement of the Stack during a longer Simulation

Data for the displacement of the stack was collected during a longer simulation to give an indication of how many ejections fail due to vibrations displacing the stack from the pick axles. This phenomenon is displayed in Figure 60 and a zoomed in version is presented in Figure 61. The stack is assumed to be maxed out during the entire simulation and will therefore not lose weight after a successful ejection. This assumption was made because the worst case scenario is the most interesting and revealing in this case. The small positive spikes in the figures are successful separations and the negative spikes are when the stack is displaced from the axles. The red dots are indications for when the separations are supposed to be performed and as can be seen in the figures, the stack is displaced a relative large part of the simulations time and as a result several separations will be compromised.

![Displacement of the stack](image)

Figure 60. The displacement of the stack during a longer simulation. The red dots indicating when the separations are supposed to happen.
Figure 61. A zoomed in version of Figure 60.
6 Discussion & Conclusion

The aim of this master thesis was to investigate which control algorithm is most suitable to control the pick motor of an expendable ejection system and with this information lay a foundation for developing a new software library. A great deal of work has been put into establishing a valid model of the BLDC and tuning each control algorithm. The tuning and comparison was first based on step responses of the motor with and without eligible disturbances. However, during the thesis more and more knowledge of the system was gained which led to the modelling of the gear train, the stack, the springs, the pick motion and the generation of the actual vibrations. Once all these parts were integrated to a complete system a more relevant and interesting comparison of the control algorithms could be made.

The results from Section 5.4 & 5.5 can be seen from different perspectives. If the goal is to succeed with a separation of an expendable from the stack, the choice of control algorithm does not have a significant impact considering they all succeed with the pick motion during extreme conditions. If a separation has failed, it’s not due to the chosen control algorithm; it’s due to the vibrations causing the stack to be displaced from the pick axles. This is something that can’t be controlled with the pick motor since it completes a full pick motion even if it can’t grab and separate an expendable.

The other point of view is if the goal is more focused on the performance in the form of rise time, overshoot, undershoot and such qualities, then the best controllers turned out to be ADRC and LADRC. However, how well the controllers perform is strongly dependent on the tuning process. A well-tuned FLC can for example perform better than an inferiorly tuned LADRC. With this in mind, the comparison of the controllers is challenging since it is hard to know if each controller is tuned optimally. It is also in a developer’s interest that the control algorithm is easy to implement and tune. This reason together with fairly good performance might be the reasons why the PID controller is the most widely used controller in industrial control systems. FLC on the other hand, would be very time consuming to implement and to tune optimally. It did however show great promise when it came to performance but unfortunately couldn’t deliver in the end. The reason for this might be that new information about the modelled system was continuously gained during the thesis which led to many small changes that rendered the currently tuned FLC almost useless and the whole process had to be done all over again. This repetition led to less time spent on the tuning process each time and might have caused the mediocre performance in the end. This was also partly the case for the ADRC and may therefore mean that it could perform even better. The LADRC and the PI controller had a simpler tuning process and are therefore more adaptive to system changes.

ADRC and LADRC showed really good performance but the LADRC was much easier to implement and to tune. If Saab were to develop a new software library, the recommendation for control algorithms would be LADRC and PI, simply because both deliver good performance and are easy to implement and tune. Also considering that there are three controllers required in FOC, the easy to implement and tune attributes are of extra importance.
6.1 Improvement and Future Work

The assumptions and simplifications that have been made in this master thesis have of course an impact on how reliable the model is. The assumption that the stack is a solid body isn’t completely true since the expendables are placed in plastic containers which have an elastic behavior during extreme conditions. This elasticity causes the stack to behave similar to a spring with its own specific characteristics. This is the first thing that should be investigated in future work. Another assumption that has been made was that the stack is pushing with equal distributed force on the two pick axles. If a more detailed model of the stack was to be produced, this could also be considered to produce a more fully fledged solution.

In the real system the expendables in the stack are hooked together which requires a small force to separate them. This affects the torque load to the pick motor in the beginning of phase 2 in the pick motion i.e. the separation phase. This extra torque load hasn’t been taken into consideration in the current model and the forces that keep the expendables together are assumed to be 0 N.

The vibration data that was taken from the MIL-STD-810G vibration test was measured on the outside of the dispenser and in this model the vibrations are applied directly on the base of the springs inside the dispenser. If a more detailed model of the whole dispenser unit containing other parts which are outside this thesis scope were to be produced, the vibrations could be applied more correctly.

Despite the assumptions and simplifications which makes the model slightly incorrect, it can still be used to validate data, compare control algorithms and simulate a change in the system before moving on to hardware e.g. if the gear train or the motor were to be substituted. This property will give the developers at Saab more qualified arguments regarding changes to the systems physical parameters.

Simulink hasn’t been used in the countermeasure dispenser department at Saab earlier but this master thesis will hopefully encourage them to consider an implementation of the program in their work procedure. It is a very powerful, easy to use software and can be used to facilitate the decision making.
7 References


65


