IMPLEMENTATION AND TUNING OF AN EXTENDED EXPERT CONTROL SYSTEM FOR HELICOPTER AUTOROTATION

AND

DEVELOPMENT OF A NONLINEAR MODEL OF ELECTRIC DRIVES TO BE USED IN THE OPTIMIZATION OF TORQUE PERFORMANCE

A Thesis
Presented to
The Academic Faculty

by

Caroline R. Repola

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
George Woodruff School of Mechanical Engineering

Georgia Institute of Technology
December 2017

IMPLEMENTATION AND TUNING OF AN EXTENDED EXPERT CONTROL SYSTEM FOR HELICOPTER AUTOROTATION

AND

DEVELOPMENT OF A NONLINEAR MODEL OF ELECTRIC DRIVES TO BE USED IN THE OPTIMIZATION OF TORQUE PERFORMANCE

Approved by:

Dr. Jonathan Rogers, Advisor
Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Remco Leine, Advisor
Institut für Nichtlineare Mechanik
Universität Stuttgart

Dr. Nader Sadegh
Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Oliver Sawodny
Institut für Systemdynamik
Universität Stuttgart

Date Approved: September 8, 2017

ACKNOWLEDGEMENTS

None of the following would be possible without the work of Dr. Paul Neitzel of the Georgia Institute of Technology and Dr. Oliver Sawodny of Universität Stuttgart, the creators of this cooperative program between their two universities. I would like to thank them for giving me the opportunity to take part in this program, especially as part of the first class to include women engineers. They both provided guidance in the work of the thesis as well as in adjusting to the different environments of Atlanta and Germany.

I would next like to thank my advisor at Georgia Tech, Dr. Jonathan Rogers. During my first in-depth experience in research, he constantly encouraged me to continue looking deeper into the topic at hand. He also cultivated a lab of students where the veterans dispensed assistance and knowledge to those of us just starting out.

While in Germany, my work was completed as a student at Bosch. I would like to thank my advisors at the company, Tino Merkel and Michèle Hirsch. They were invaluable as mentors to my work. There were also many other Bosch employees and interns who helped with my work and helped me have a positive experience at the company.

Also providing support in Germany was Dr. Remco Leine, my advisor from Universität Stuttgart. Although he was not an expert on my research topic, he was able to bring a different perspective and expand my work in unexpected directions.

Finally, I want to thank my parents for every word of encouragement over the last 25 years and their support when I decided to move to the South and then across an ocean.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **iii**

LIST OF TABLES **vi**

LIST OF FIGURES **vii**

LIST OF SYMBOLS AND ABBREVIATIONS **ix**

SUMMARY **xi**

PART ONE **1**

CHAPTER 1. Introduction **2**

CHAPTER 2. Helicopter Operation and the Autorotation Controller **5**
2.1 Helicopter Operation **5**
2.1.1 Control of the Helicopter 6
2.1.2 Autorotation 7
2.1.3 States Describing the Behavior of a Helicopter 9
2.2 The Autorotation Controller **10**
2.2.1 The Stages of the Controller 12
2.2.2 Adjusting for a New Helicopter 14

CHAPTER 3. Model Development and Tuning **15**
3.1 Main Controller Design **15**
3.1.1 The Equations and Parameters 17
3.2 Tuning the Controller **20**
3.2.1 Steady-State Descent 21
3.2.2 Pre-Flare 23
3.2.3 Flare 24
3.2.4 Landing 27
3.2.5 Touchdown 29
3.3 Results **30**

CHAPTER 4. Conclusion **33**
4.1 Closing Remarks **33**
4.2 Future Work **33**

PART TWO **35**

CHAPTER 5. Introduction **36**

CHAPTER 6. Electric Drive Description **38**
6.1 Overview of the Electric Drive **38**
6.2 Control of the Electric Drive **39**

CHAPTER 7. Modeling the System **42**
7.1 Motivation **42**
7.2 Manually Switched Model **45**
7.2.1 Single Phase Model 45
7.2.2 Three Phase Model 56
7.2.3 Results 61
7.3 Non-Smooth Differential-Algebraic Equation Model **62**
7.3.1 Single Phase 63
7.3.2 Three Phase 70
7.3.3 Zero Crossing 77
7.3.4 Results 78
7.4 Comparison of the Models **82**
7.5 Future Use and Extensions of the Model **84**

CHAPTER 8. Model Predictive Control **86**
8.1 Motivation **86**
8.2 Structure of Model Predictive Control **87**
8.2.1 The Cost Function 89
8.2.2 Calculation of Torque 91
8.2.3 Optimizing the Switching Sequence 92
8.3 Further Use and Extensions **92**

CHAPTER 9. Conclusion **96**

REFERENCES **98**

LIST OF TABLES

Table 1	– Conditions for successful and marginal landings	21
Table 2	– Initial conditions used to demonstrate robustness of the controller	30
Table 3	– Autorotation Controller Parameters	30
Table 4	– Values of constants in the single and three phase circuits	45
Table 5	– Possible switch states for single phase circuit	47
Table 6	– Possible switch states for three phase circuit	56
Table 7	– Switch states for single phase circuit, with switch pairs view	64
Table 8	– Switch states for three phase circuit, with switch pairs view	70
Table 9	– Comparison of calculation times for single and three phase models using PLECS and Matlab	78

LIST OF FIGURES

Figure 1	– Momentum theory flow model for hover	6
Figure 2	– The difference in air flow through the main rotor in normal powered flight and during an autorotation	8
Figure 3	– Typical trajectory of a helicopter performing an autorotation	12
Figure 4	– An example of fuzzy logic transitions for the autorotation controller. The dashed lines indicate boundaries between the stages and the areas with labels are ‘flat’ meaning the controller is operating fully in that stage. The only exception is Touchdown, which does not reach saturation until ground level.	12
Figure 5	– Control system block diagram	15
Figure 6	– Velocity tracking controller block diagram for the longitudinal cyclic control value	17
Figure 7	– Successful autonomous autorotation. Shown is (from top left) altitude, vertical velocity, forward velocity, collective control, rotor rotation rate, pitch angle, the stage of the controller, and longitudinal cyclic control	30
Figure 8	– Successful automatic autorotations starting from various initial conditions, demonstrating robustness	32
Figure 9	– General block diagram of the electric drive	38
Figure 10	– Block diagram illustrating closed loop field oriented control of the electric drive	40
Figure 11	– System circuit diagram showing the three main components as their component parts, here in wye connection, which is used throughout this thesis	43
Figure 12	– Wye (left), used in this circuit, and delta (right) connections of three phase systems	44
Figure 13	– Single phase circuit with switches	46
Figure 14	– Possible 'simplified' circuits for single phase model; (a) when either both high side or both low side switches are closed or that diode is active; (b) and (c) when the switch pairs have opposite switches closed or opposite active diodes, the direction of the	48

voltage drop in the load elements depends on the sign of the current and which switch pair is high

Figure 15 – Partial circuit, representing behavior of open circuit case 55

Figure 16 – Three phase circuit with labeled switches 57

Figure 17 – Two circuits demonstrating the similar layout for two switch states 58

Figure 18 – The three general circuit configurations for the three phase model; (a) is when the switch pairs are all high or low, (b) is if two are high and one is low, and (c) is if one is high and two are low 59

Figure 19 – Graph of the function $\boldsymbol{Sgn}(\boldsymbol{x})$ 63

Figure 20 – Single phase circuit showing the possible paths of current for the four color-coded options in Table 7 64

Figure 21 – Single phase circuit with currents and lambdas 66

Figure 22 – Three phase circuit with a sketch demonstrating one possible path for the currents based on the color-corresponding row in Table 8 71

Figure 23 – Three phase circuit with currents and lambdas indicated 72

Figure 24 – Results comparing model to PLECS, single phase, load current, including switch commands 79

Figure 25 – Results comparing model to PLECS, single phase, capacitor voltage 80

Figure 26 – Results comparing model to PLECS, three phase 81

Figure 27 – Results comparing model to PLECS, three phase 82

Figure 28 – Diagram illustrating time horizons for model predictive control 88

Figure 29 – Search tree. Green nodes are inside the limit of the cost function, red nodes are outside, and black nodes are unexplored 94

LIST OF SYMBOLS AND ABBREVIATIONS

PART ONE

c	Blade chord, ft
DVE	Degraded Visual Environment
h	altitude above ground level ($-z$), ft
HITL	Human-in-the-Loop
m	Mass of the helicopter, slugs
N	Number of main rotor blades
PD	Proportional-Derivative control
PID	Proportional-Integral-Derivative control
R	Main rotor radius, ft
q	pitch rate in body reference frame, rad/s
u_{des}	forward speed command, ft/s
η	total magnitude of vehicle roll and pitch angles; $\eta \equiv \sqrt{\phi^2 + \theta^2}$,rad
η_{max}	maximum pitch angle commanded by the controller, deg
θ_{tr}	tail rotor collective blade pitch, rad
$\dot{\theta}_0$	collective control derivative, rad/s
θ_0	main rotor collective blade pitch, rad
θ_{1c}	main rotor lateral cyclic pitch, rad
θ_{1s}	main rotor longitudinal cyclic pitch, rad
ρ	Atmospheric density, slug/ft^3
Ω	main rotor rotation rate, rad/s

PART TWO

DAE	Differential algebraic equation
emf	Electromotive force, V
g_C	Charge in the capacitor, C
KPI	Key performance indicator
i_C	Current through the DC-Link side of the circuit, A
i_L,i_{L1},i_{L2}	Current or currents through the load side of the circuit, A
MPC	Model predictive control
ODE	Ordinary differential equation
q_i	Dummy charges, C
V	AC voltage source on the load, V
v_C	Voltage in the capacitor, V
λ_i	Representation voltages of the switches, V
τ,τ_d	Torque, desired torque, N·m

SUMMARY

This thesis covers two separate investigations under the topic of control. The first is the design and tuning of a fuzzy logic controller for Human-in-the-Loop (HITL) helicopter autorotation. The second is the exploration of an optimized pulse pattern for the control of an electric drive with focus on the development of the mathematical model of the drive.

Part One of this thesis discusses the autorotation controller. Helicopter autorotation is the operation a pilot performs when power is no longer supplied to the main rotor and an emergency landing is required. A controller was developed that allowed an autonomously controlled helicopter to perform an autorotation, an 'expert skill' more easily learned by human pilots. This controller is used in this thesis to create a tool that brings the computer and human together. The tuning process of the autorotation controller is described in detail. The controller used has five stages of operation; the transitions between these stages occur through a fuzzy logic determination. The results of the tuning bring about a successful autorotation in a simulated environment. The specific model of the controller developed in this thesis can be used in a different system to supply commands to a human pilot, aiding in the decisions during an autorotation.

Part Two of this thesis covers the development of the mathematical model of an electric drive and an optimization scheme to find a 'better' switching sequence for control. The goal of the model is to use it to find a better switching sequence, where better means fewer switching events as well as hitting targets of other key performance

indicators (KPIs). The idea explored in this thesis is controlling the drive based on direct manipulation of the switches instead of indirectly through voltage or current. The mathematical model focusing on the switches is important to develop to facilitate the exploration of this control. Two different methods for developing this model are described. The first is a manually switched model based on examining every possible state of the drive. The second method is a non-smooth differential algebraic equation (DAE) approach, a more sophisticated mathematical approach that describes every state of the drive in one set of equations. An optimization scheme using model predictive control (MPC) is described. The focus of the optimization is the torque output of the motor and the number of switching events. The optimization would use the model developed in the thesis.

PART ONE

IMPLEMENTATION AND TUNING OF AN EXTENDED EXPERT CONTROL SYSTEM FOR HELICOPTER AUTOROTATION

CHAPTER 1. INTRODUCTION

Harry Reasoner, an ABC news reporter, remarked during the Vietnam War, that "[a] helicopter does not want to fly. It is maintained in the air by a variety of forces and controls working in opposition to each other, and if there is any disturbance in this delicate balance the helicopter stops flying immediately and disastrously." The normal operation of a helicopter is a complex balancing act, and when a critical element, such as an engine, fails, that leads to the seemingly impossible task of landing the aircraft regardless. Helicopters can experience engine failure at any point in their flight and in many different conditions. Whether it is manned or unmanned, the goal is always to bring the vehicle back to the ground as safely as possible. This process is called autorotation, and occurs if power to the main rotor is lost. Without power, the pilot can still control the angle of the rotor and the blades and thus bring about a landing that has low forward and vertical speeds with the helicopter nearly level, which is the goal in a successful autorotation.

Human pilots are good at learning the 'expert' skill of autorotation, but it can be much more difficult for a computer to perform this task. However, in [1] an autorotation controller for an autonomous vehicle was developed. This control system can be combined with a piloted copter to have Human-in-the-Loop (HITL) control, where the pilot in a manned aircraft would be aided by the commands from a controller-calculated autorotation. This approach would be desired in a situation where there is degraded visual environment (DVE), since a pilot largely performs autorotation by judging the distance to the ground visually. In a DVE, it is difficult or impossible for the pilot to see

the ground and thus achieving autorotation is also difficult or impossible. These situations include night time—even if there are lights on the ground, distances are harder to judge in the dark—or during a weather event, such as fog, clouds, rain, or snow. With the United States' continued military presence in the Middle East, sandstorms are a concern that can also cause a DVE. The HITL control could also help in situations where the helicopter is flying slowly or at a low altitude when the failure occurs, meaning the pilot has less time to make decisions and execute the autorotation [1]. Although based on autonomous control, the goal of the controller and the HITL implementation is to create a pilot aid and not a crutch.

The work done in this thesis is part of a partnership with the University of Liverpool and funded by the United States Army. The autorotation controller is meant to be one part of a system that will aid pilots. The eventual result would be a guide for the pilot where the computer calculates and outputs the commands needed for a successful autorotation. Depending on the situation, the human pilot follows and implements those commands as closely as they can, or uses the supplied commands as a decision making aid. HITL control means different restrictions on the controller-supplied commands than if the autorotation is being performed by the controller as well, as was done in the controller in [1].

The next chapter of this thesis discusses helicopter autorotation and the autorotation controller that has been previously developed. The following chapter will cover the development of the autorotation controller such that it can work in the full controller created by the University of Liverpool and the tuning of that controller. Finally, the

results of the autorotation controller tuning will be discussed as well as the next steps in the life of the controller and any other future use or further features.

CHAPTER 2. HELICOPTER OPERATION AND THE AUTOROTATION CONTROLLER

2.1 Helicopter Operation

As indicated in the introduction, the operation of a helicopter is very complex, and as such will not be covered completely in this thesis, and is in fact the topic of many books. For our purposes, we will discuss some basics of normal operation of a helicopter and point out the differences in autorotation. Regardless of whether the helicopter is powered, the rotor is always the central aspect of helicopter operation. Almost all navigation is performed by controlling aspects of the rotor and the blades, and as long as the rotor is still spinning, a safe landing can be achieved.

Like any aircraft, a helicopter stays in the air by creating lift, or thrust, that works against the force of gravity. How this thrust is produced is described by momentum theory, which is based on fluid mechanics [2]. There is a constant flow of air through the rotor, and the mass of this air creates momentum. There is also a power transfer between the rotor and the air [3]. There are four different flight modes: axial climb, axial descent, forward flight, and hover. In each mode, there is still airflow through the rotor, but the direction may change, or the transfer of power is altered [2]. Figure 1 illustrates momentum theory for one of the flight modes, hover.

The rotational speed of the rotor determines the magnitude of thrust. The direction of that thrust is determined by the angle of the rotor. The rotor is often almost horizontal; vertical flight is the specialty of the helicopter [3].

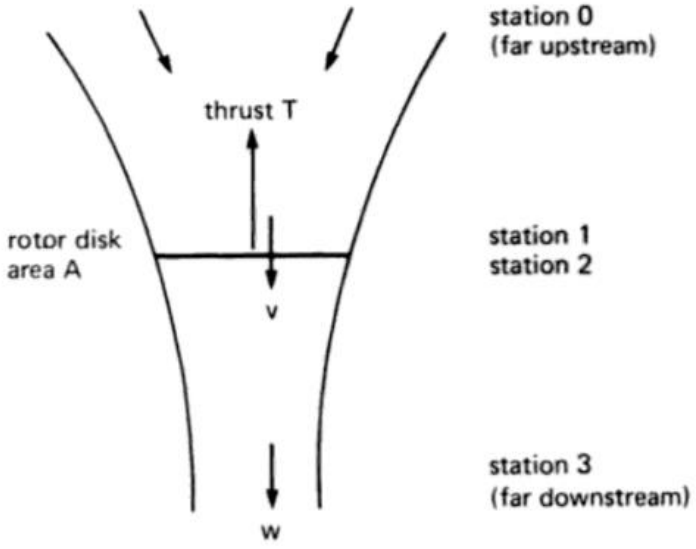

Figure 1 – Momentum theory flow model for hover [1]

2.1.1 Control of the Helicopter

There are four control elements of a helicopter: the pedal, θ_{tr}, (yaw control), lateral and longitudinal cyclic, θ_{1c} and θ_{1s}, (roll and pitch control), and the collective, θ_0, (thrust control). The first, the pedal, controls the tail rotor, while the following three all control different aspects of the main rotor. This control is performed through a swash-plate, a hub located on the shaft that connects the blades and does not require large amounts of force to move [4]. The collective controls all of the blades together, hence the name. By changing the pitch of the main rotor blades averaged over all azimuth angles, the collective changes the average total rotor thrust [3]. The two cyclic controls, lateral and longitudinal cyclic, control the angle of the blades with respect to the azimuth. This changes the tip path plane, changing the angle of the rotor with respect to the swash plate and thus the angle of the thrust vector [4].

In autorotation, only the longitudinal cyclic and collective controls affect the success of the autorotation and the other two are taken care of by either a separate velocity tracking controller, or, in the HITL case, by the pilot. This thesis discusses only the generation of the θ_{1s} and θ_0 control values. The longitudinal cyclic controls the pitch angle of the helicopter and impacts the vertical and forward velocity and acceleration. The collective controls the pitch of the main rotor blades and is important during autorotation because the angle of the rotor determines the rotation rate of the blades which provides energy to the helicopter and allows the required maneuvers in an autorotation to be performed.

2.1.2 Autorotation

In autorotation, power is no longer supplied to the rotor, so this can no longer be a consideration in performance. This means the helicopter must perform such that the rotor needs zero net power to ensure the helicopter will continue to fly [4]. The engine can no longer provide the energy to keep the blades rotating, but the airflow through the rotor can. Because the controls to the rotor remain available to the pilot, the blades can be adjusted to take advantage of the air flow through the rotor to keep it spinning and thus allowing controlled flight to continue.

Autorotation differs from normal operation in that the flow of air through the rotor is absolutely necessary to keep the blades spinning. Figure 2 below illustrates how the direction of airflow changes when a helicopter is in autorotation compared to normal operation, in this case forward flight. In autorotation, the blades act like a parachute, using the support from the mass of air flowing up through them to slow descent [4].

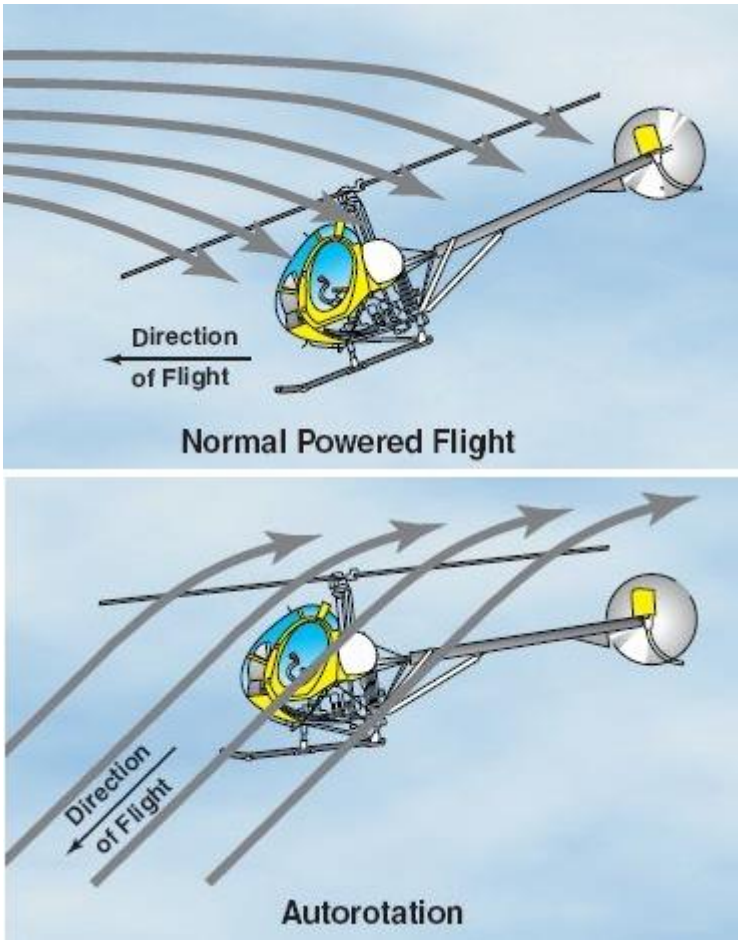

Figure 2 – The difference in air flow through the main rotor in normal powered flight and during an autorotation

The details of how an autorotation is performed will be covered further in this thesis, but Figure 3 is presented now to show the general shape of the trajectory. The helicopter initially enters a dive to maintain speed and keep rotors spinning, as indicated in the figure. The gravitational potential energy of the helicopter turns into kinetic energy in the rotors, resulting in a high rate of descent [3]. At the bottom of the dive, the helicopter must pull up sharply to slow the descent enough to make a comfortable landing. This is called the flare or flare-out. After the flare-out, the helicopter must still have enough time until touching down and distance to the ground to level out, ensuring

that the tail does not hit ground first. The specific commands that bring about this behavior will be explored in the following chapter.

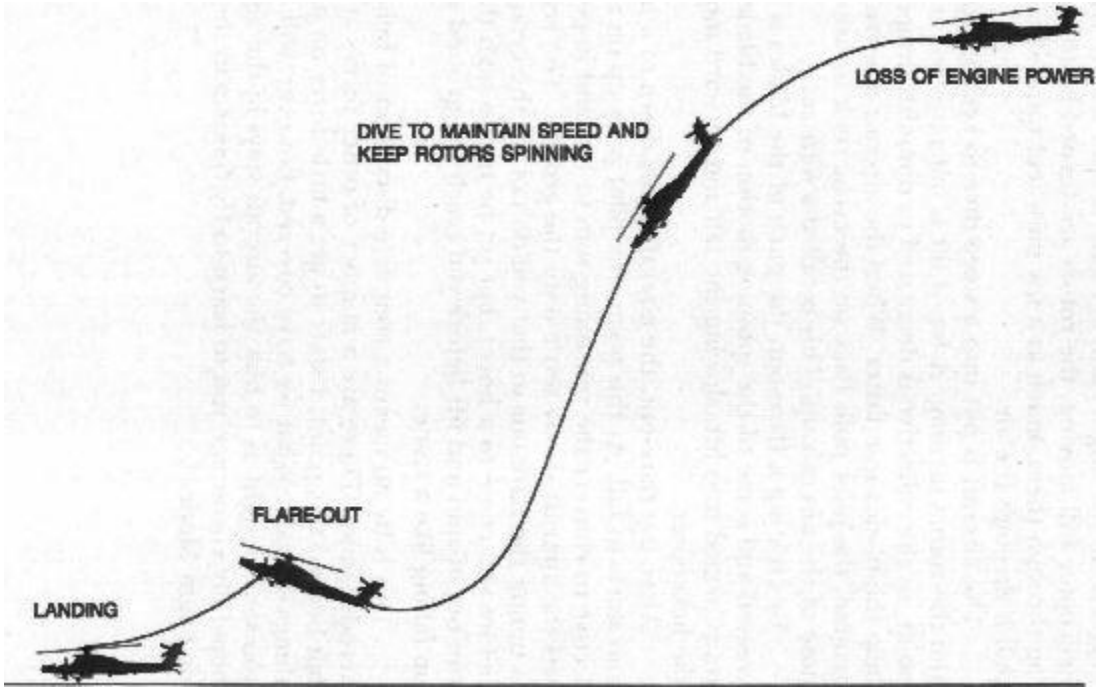

Figure 3 – Typical trajectory of a helicopter performing an autorotation

2.1.3 States Describing the Behavior of a Helicopter

There are many parameters that can describe the motion of a helicopter. The general full body motion and flight path of a helicopter can be described with 12 different states, some of which are related to the body-frame of the helicopter and some of which are related to the inertial frame. The values x, y, and z describe the location in space and are the location of the center of mass of the helicopter in reference to the inertial frame. In aviation, the reference frame used is North-East-Down; the choice of vertical down allows complying with the right-hand rule. Therefore, in this thesis, z is positive down but in later sections and chapters we also have h representing the altitude of the

helicopter, which is positive up. The angular position of the helicopter is denoted by ϕ (roll), θ (pitch), and ψ (yaw) which are again in reference to the inertial frame. The velocities of the helicopter, u, v, and w, are body-fixed (i.e., u is not the derivative of x), as are the angular velocities, p (roll rate), q (pitch rate), and r (yaw rate).

Only some of these values are of interest when tuning the autorotation controller. The vertical position, z or h, as discussed above, is necessary in many calculations and determination of other control parameters. The forward speed, which is found by

$$\dot{x} = u\cos\theta + v\sin\phi\sin\theta + w\cos\phi\sin\theta \tag{1}$$

is used in addition to u since we are concerned about the forward speed in reference to the inertial frame and not just in the body-fixed frame. Additionally, the vertical speed

$$\dot{h} = u\sin\theta - v\sin\phi\cos\theta - w\cos\phi\cos\theta \tag{2}$$

is used along with w for the same reason and to better estimate time to ground impact. The vertical speed $\dot{h}$ is the negative of $\dot{z}$. The pitch angle, θ, pitch rate, q, vertical speed, $\dot{h}$, and forward speed, $\dot{x}$ are all used to determine safe landing conditions and during tuning. An additional parameter, the rotor rotation rate, Ω, is also important during normal flight but especially during autorotation controller tuning.

2.2 The Autorotation Controller

As stated in the introduction, human pilots can learn to perform an autorotation fairly well. Teaching a computer is more difficult, but this has been achieved in [1] and a

helicopter was able to achieve autonomous autorotation. The elements of this controller that makes it effective are the use of fuzzy logic to transition between the five stages of operation and estimated time to ground impact to optimize the flare-out calculations [1]. During each stage, the goal is to get the helicopter in the correct state to continue to the next stage; the use of fuzzy logic allows for the gradual transition between stages so there are no control transients, or large differences in desired control which could lead to wild changes in behavior of the helicopter. The five stages are Steady-State Descent, Pre-Flare, Flare, Landing, and Touchdown. The weighting of the stages is determined by the current altitude of the helicopter, h, and time to impact, TTI, which is defined as

$$TTI_{\ddot{h}=0} = -\frac{h}{\dot{h}}. \tag{3}$$

Variations on TTI are also used in the Flare and Landing stages for calculations of control values. The vertical distance to the ground, velocity, and acceleration are very important, as this informs the amount of time and distance remaining to perform the autorotation maneuvers.

The transition between each phase uses trapezoidal fuzzy logic. Each transition is linear, i.e. the state that is halfway between Flare and Landing would result in control values that have a 50% contribution from each stage. Figure 4 below shows a graph of the fuzzy logic transition between stages. The exact values of the boundaries between each phase have to be determined during the tuning process and are different for different helicopters. The autorotation control could start with the helicopter in a state that locates in a later phase, although for design it will be assumed it starts in Steady-State Descent.

The controller will never go backwards, i.e. it can never go from Pre-Flare to Steady-State Descent, or once it has decreased the contribution of Flare it will not increase it again. The fuzzy logic also demonstrates that the operation of each phase is not strictly defined by height or time to impact and autorotation is a bit of an art.

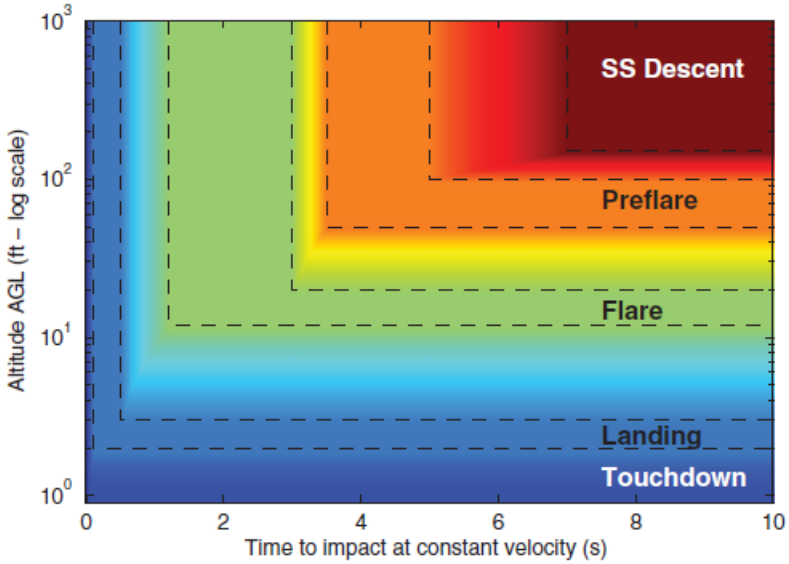

Figure 4 – An example of fuzzy logic transitions for the autorotation controller. The dashed lines indicate boundaries between the stages and the areas with labels are 'flat' meaning the controller is operating fully in that stage. The only exception is Touchdown, which does not reach saturation until ground level.

The autorotation control law at each stage results in three outputs: u_{des}, η_{max}, and $\dot{\theta}_0$. The first two are fed into a velocity tracking controller to give the desired θ_{1s} command. The final output is integrated to get the desired θ_0 command.

2.2.1 The Stages of the Controller

The goal of the first stage, Steady-State Descent, is self-evident, to get the helicopter to steady-state operation. The helicopter needs to have a constant forward velocity, $\dot{x}$, and the rotor must be kept spinning to maintain energy. During this stage the

pilot can also use the other controls to begin to navigate the helicopter to an appropriate landing site. Following Steady-State Descent is the Pre-Flare stage which is very similar to the previous and acts as a transition stage to the Flare stage. Constant forward velocity is still desired, but there is a limit placed on η_{max} so that if the helicopter was pitched down during Steady-State Descent, it will not have to command such a dramatic pitch up in the Flare stage.

The Flare stage is the critical stage where most of the action of the autorotation occurs. In this stage, the cyclic control is engaged to tilt the helicopter upwards and bring the vertical and horizontal velocities to lower and safe values for entering the Landing stage. The collective control is concerned with maintaining the kinetic energy in the rotor. In this stage there are no limits on the control inputs so that the helicopter can perform all necessary maneuvers to slow down. Like the previous two stages, Flare and Landing are very similar, again only adding a limit to $\eta_{\max}$ to ensure a smooth entrance into the final stage and a successful impact. In these stages, a TTI is constantly calculated, ensuring that the use of the available energy in the rotor is optimized. The Landing stage is used as a transition from Flare to Touchdown.

The final stage is Touchdown, where the helicopter finally reaches the ground. During the Landing and Touchdown stages, the helicopter is slowing down and leveling out. The controller is never fully in the Touchdown stage, based on the fuzzy logic, until it actually touches ground, at which point the autorotation is complete. In the Touchdown stage, all values are set to bring the helicopter to low forward and vertical velocities and to be almost level, for both comfort of the pilot and passengers but also to avoid bringing the tail rotor into contact with the ground. It is during the final transition

stage that the successful conditions for an autorotation on the helicopter states should be reached.

2.2.2 Adjusting for a New Helicopter

The autorotation controller has already been tuned for a Bell AH-1G Cobra, but the work of this thesis focuses on an SH-60 Seahawk. This helicopter is bigger than the previous and as such many values will have to increase. The controller, of course, does not have to change, as it was designed to work with many types of helicopters. However, the controller has many tuning parameters. Some are based on physical properties or assumptions of performance of the helicopter, and the initial guesses for the SH-60 can be made accordingly. Some are not, and the tuning has to be performed with the new helicopter in mind.

The tuning in this thesis starts from some values that are typical for a SH-60, such as the target velocities or angle limits, as well as using the values for the parameters that were tuned to the AH-1G. For example, although the higher weight of the Seahawk means the Steady-State Descent has to start at a higher altitude with more time until impact, the relative differences in the boundary values of the fuzzy logic stages for the AH-1G can inform those needed for the SH-60. The specifics of the tuning process are further discussed in the next chapter.

CHAPTER 3. MODEL DEVELOPMENT AND TUNING

3.1 Main Controller Design

As described in previous chapters, the work of this thesis is based on a previously developed controller, realized as part of a helicopter flight simulator written in C++. The controller in this thesis is developed in a way that allows it to be plugged in to another controller. This is why the autorotation controller only needs to produce control values for the collective and longitudinal cyclic and can rely on another controller or pilot to produce the control values for lateral cyclic and pedal.

The controller in this thesis is built in Simulink (developed by Mathworks), which usually takes an initial condition, runs a loop over time with evolving conditions, and gives the final output. However, the need to fit the controller into another system means that the Simulink controller that is the topic of this thesis can only run one step of control at a time. The state input information to the Simulink controller comes from a plant that is a separate helicopter flight simulator that takes in the commanded controller values. A simplified block diagram of just the autorotation controller is shown in Figure 5.

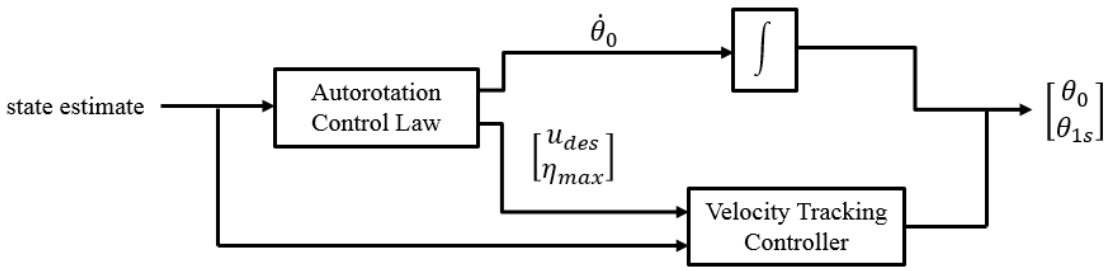

Figure 5 – Control system block diagram of the autorotation controller

The three output values from the autorotation control law block are calculated with the equations detailed in Section 3.1.1. This block is also where the fuzzy logic operates. The state estimate also provides information to determine the correct phase or mix of phases of the autorotation control, and the outputs have been determined based on that information.

There are two channels of control to produce the two control variables, θ_0 and θ_{1s}, as shown in Figure 5. The change in collective control is determined differently in each stage of control and then the time derivative must be taken. The first two stages of the autorotation controller are a PD controller tracking the rotor rotation rate. The next two stages track the vertical trajectory to ensure advantageous stage transitions and appropriate energy usage. The final stage simply reduces the rate of change of the collective. Each of these will be described further in later sections.

The longitudinal cyclic controller is the same design regardless of stage, simply using different values for u_{des} and η_{max}. The design of the longitudinal cyclic velocity tracking controller is shown in Figure 6; two nested-loop PID controllers track the desired velocity and determine the necessary pitch angle to achieve that velocity tracking. This is the same

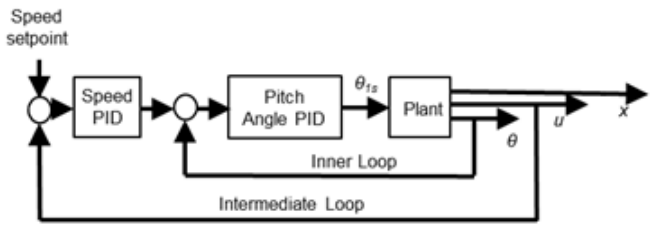

Figure 6 – Velocity tracking controller block diagram for the longitudinal cyclic control value

design for a controller that would be used when the helicopter is in normal operation with a powered rotor, without the optional outer loop controlling position. The two outputs from the autorotation control for the longitudinal cyclic channel are u_{des} and η_{max}. The former is used as the set point for the speed PID controller. The latter is used as a saturation limit before the pitch angle PID controller.

The final detail of the controller design is the time discretization. As explained already, the Simulink controller can only advance one step at a time. This means that each derivative and integral has to be discretized. To achieve this, we choose a time step of 50 milliseconds. This value was based on the update rate of the original helicopter simulator, indicating that it is sufficiently small to model the behavior. The advantage of using Simulink to design a controller is the availability of pre-built blocks. This advantage is lost, but instead we have the advantage of knowing exactly what is happening in each calculation. This also adds the necessity to store values from the previous update to be used in the next step to calculate the derivatives and integrals.

3.1.1 The Equations and Parameters

Inside the autorotation control law block in Figure 5 is a calculation of the three values needed to determine control. This is performed using the following equations. A detailed derivation can be found in [1]. The equations are presented first and separately from the tuning discussion so the similarities across stages can be clearly seen.

The control for the Steady-State Descent stage is described by:

$$u_{des} = \texttt{U\_AUTO} \tag{4}$$

$$\begin{aligned} \eta_{max}^{+} &= \texttt{SS\_DESCENT\_MAX\_ANGLE} \\ \eta_{max}^{-} &= \texttt{MIN\_ANGLE} \end{aligned} \tag{5}$$

$$\dot{\theta}_0 = \texttt{K\_D\_SS}\,\dot{\Omega} + \texttt{K\_P\_SS}(\Omega - \texttt{RPM\_AUTO}). \tag{6}$$

The control for the Pre-Flare stage is described by:

$$u_{des} = \texttt{U\_AUTO} \tag{7}$$

$$\eta_{max} = \texttt{PRE\_FLARE\_MAX\_ANGLE} \tag{8}$$

$$\dot{\theta}_0 = \texttt{K\_D\_SS}\,\dot{\Omega} + \texttt{K\_P\_SS}(\Omega - \texttt{RPM\_AUTO}). \tag{9}$$

The control for the Flare stage is described by:

$$u_{des} = \texttt{U\_TOUCHDOWN} \tag{10}$$

$$\begin{aligned} \eta_{max}^{+} &= \texttt{FLARE\_MAX\_ANGLE} \\ \eta_{max}^{-} &= \texttt{MIN\_ANGLE} \end{aligned} \tag{11}$$

$$\dot{\theta}_0 = \begin{cases} \frac{\texttt{K\_COL}}{\texttt{TAU}}\left(-\frac{2\left(h+\dot{h}TTI_F\right)}{TTI_F^2} - \ddot{h}\right) & for -\frac{2h}{\dot{h}} \geq TTI_F \\ \texttt{FAST\_COL\_INCREASE} & for -\frac{2h}{\dot{h}} < TTI_F \end{cases}. \quad (12)$$

The control for the Landing stage is described by:

$$u_{des} = \texttt{U\_TOUCHDOWN} \quad (13)$$

$$\eta_{max} = \texttt{LANDING\_MAX\_ANGLE} \quad (14)$$

$$\dot{\theta}_0 = \begin{cases} \frac{\texttt{K\_COL}}{\texttt{TAU}}\left(-\frac{2\left(h+\dot{h}\texttt{TTI\_L}\right)}{\texttt{TTI\_L}^2} - \ddot{h}\right) & for -\frac{2h}{\dot{h}} \geq \texttt{TTI\_L} \\ \texttt{FAST\_COL\_INCREASE} & for -\frac{2h}{\dot{h}} < \texttt{TTI\_L} \end{cases}. \quad (15)$$

The control for the Touchdown stage is described by:

$$u_{des} = \texttt{U\_TOUCHDOWN} \quad (16)$$

$$\eta_{max} = \texttt{TOUCHDOWN\_MAX\_ANGLE} \quad (17)$$

$$\dot{\theta}_0 = \texttt{TOUCHDOWN\_COL\_DECREASE} \quad (18)$$

Equations 5 and 11 are different than they appear in [1]. The previous definition was a boundary only defined by the pilot or velocity tracking controller. Since our controller defines the velocity tracking controller, a value for the limit on the angle needs to be defined for those stages as well. They are both also uneven, where the maximum commanded angle is allowed to be greater than the minimum commanded angle.

There are 16 control parameters used in Equations 4-18, although `TAU` and `K_COL` can be treated as one parameter, and one additional parameter `TTI_F_MAX` used in the calculation of TTI_F (See [1]). Other necessary parameters are the 14 boundary values that define the fuzzy logic control. The two loops of PID control in the velocity tracking controller introduce an additional six parameters. Including all of the parameters brings the total to 37. Some parameters are easy to prescribe, e.g. the target velocities. Some parameters overlap between stages, but even the ones that do not can still have an effect on the others. The tuning will be described stage-by-stage, but the interaction will still be noted. Possibly qualifying as the most important parameters are the boundary values for the fuzzy logic controller as these determine when each stage is active and can have effects on all five stages.

The focus of this thesis is the autorotation controller that will be used as part of a HITL system where the plant is the helicopter itself. However, to first achieve the tuning, the controller must be validated through a simulation model. In this thesis, the model used was a previously created system based on the ARMCOP model developed by Talbot and Chen (See [5], [6], and [7]). However, some aspects of this model were updated to provide increased fidelity. See [1] for more sources of how the model was created and details on the changes to the ARMCOP model.

3.2 Tuning the Controller

The tuning of this controller is not straightforward. Not only are there many different parameters for each of the five stages of operation, there are also the fuzzy logic boundary values that determine when the transitions between the stages should occur.

This means there are various factors at play, all of which are influencing each other. The overall goal for the tuning of the controller is to achieve a successful landing from the autorotation. Table 1 shows the conditions that define a successful landing. These conditions are specific to the SH-60 helicopter. This section will discuss the specific goals for tuning each stage, the parameters involved, and the interactions on and from other stages.

Table 1 – Conditions for successful and marginal landings

Parameter	Condition for Successful Landing	Condition for Marginal Landing
Pitch Angle, θ	<12°	<20°
Forward Speed, $\dot{x}$	<30 knots	<60 knots
Vertical Speed, $\dot{h}$	<8 ft/s	<15 ft/s
Pitch Rate, q	-30°/s<q<20°/s	-50°/s<q<40°/s

While tuning, the response to the states described at the end of Section 2.1.3 will be used to judge success, with the values in Table 1 used only for judging success of the final impact. However, it is also important to observe the behavior of the commanded control values. Rapidly oscillating control is not usually desired; in this case, keeping in mind the HITL goal, it should be avoided. While an autonomous controller could track those commands accurately, a human pilot would not be able to follow the commands quickly enough.

3.2.1 Steady-State Descent

The first stage can be tuned independently; there is nothing in the control sequence before it and, depending of the height at which failure occurs, the helicopter will spend a significant amount of time in this stage and the following one, which has almost identical

controls and set points. As stated in Section 2.2.1, the goal of this phase is to maintain control of the helicopter and ensure there is enough energy to continue the autorotation. As in every stage of the controller, there is a target forward velocity which results from the cyclic control. The collective control tracks the main rotor rotation rate. Equation 6 shows that the collective control in this stage is PD control with respect to the rotor rotation rate.

The fuzzy logic boundary values are straightforward to assign here. There is no maximum of h or $TTI_{\dot{h}=0}$ as Steady-State Descent is the first stage. The minimum of these values is determine as the minimum altitude and time needed to perform a safe autorotation. This can be set according to knowledge of the type of helicopter, and adjusted later when further stages of the autorotation controller are tuned.

In the longitudinal cyclic channel, the PID loops are tuned. The desired forward velocity, `U_AUTO`, is set to an appropriate velocity for autorotation of an SH-60. The tracking of this value is tuned with the PID values in the two loops of the velocity tracking controller. The controller should be robust enough to reach a range of `U_AUTO` values from a range of initial forward velocities. These parameters are constant across all five stages of the autorotation controller, so they may need further adjustment as tuning continues. Here is introduced the recursive tuning that is necessary in a complicated controller such as this one. As parameters from an earlier stage are present in later stages and may be changed, the tuning of the earlier stage needs to be revisited to ensure that it is still behaving as needed. Parameters for other parts of the control may affect the forward speed as well. Also part of the velocity tracking controller are the pitch angle

limits, SS_DESCENT_MAX_ANGLE and MIN_ANGLE. The lower limit is defined based on normal helicopter operation limits while the upper limit has to be tuned to ensure that the target velocity can be reached in a reasonable amount of time. Because these limits are applied to the desired angle produced from the outer loop of the velocity control, they cannot assure that the actual pitch angle of the helicopter is within those bounds. The behavior of this state must be observed to ensure that it does not go too far beyond the commanded limit.

The collective control channel is PD control of the rotor rotation rate, as seen in Equation 6. The desired rotor rotation rate, RPM_AUTO, must be with 20% of the nominal rotor rotational speed. Like U_AUTO, this value is set and left, as this target is necessary to continue the autorotation. The tuning for the collective control of this stage then comes from the parameters K_P_SS and K_D_SS. These two values need to be tuned to cause a quick, regular PD response in Ω. At the beginning of the autorotation, the energy in the rotor must switch from being applied by the engine to being applied by the air. The rotation rate must be kept high enough to keep the aircraft flying.

At the end of the Steady-State Descent stage, the helicopter should have reached the target values. The vertical velocity as well as the forward velocity should be in steady-state mode. Due to the fuzzy logic, the control will already have partially passed to the next stage, beginning our discussion of the tuning of the Pre-Flare stage. The target forward velocity and rotor rotation rate will be the same in this stage.

3.2.2 Pre-Flare

As soon as tuning for this stage begins, the fuzzy logic becomes more involved, and individual stage tuning is more difficult. As observed from Equations 7-9, the control in this stage is almost identical to the control of the previous stage. The only difference in this stage is the angle limit, which ensures the helicopter is not pitched too high or low and can easily transition to the next stage, Flare. The similarity of these stages is because Pre-Flare is meant as a transitory stage between Steady-State Descent and Flare. The fuzzy logic boundary values should be set such that it does not take a long time for the controller to transition between Steady-State Descent and Pre-Flare.

Because of the similarity of the first two stages, the values tuned previously should still be valid. We only have to choose a value for `PRE_FLARE_MAX_ANGLE` and the tuning of this stage comes from choosing the boundaries for the fuzzy logic. We expect the autorotation controller to pass through this stage quickly, with a linear transition. The transfer of control to the Flare stage must give enough time and height to perform the autorotation, but not so much that it reaches a landing state before reaching zero altitude. `PRE_FLARE_MAX_ANGLE` is set to a lower value than `SS_DESCENT_MAX_ANGLE`. There is no change in the operation for the collective channel. The Pre-Flare stage needs to last long enough to ensure the helicopter reaches the state where the pitch angle is within the `PRE_FLARE_MAX_ANGLE` limit. Once it has achieved that, the next stage of the autorotation controller can begin.

3.2.3 Flare

The Flare stage is the central operation of the autorotation. At this point, the velocity tracking controller switches targets and the deceleration to a forward velocity

that is within the successful landing conditions begins. The collective channel has a more complex control, based on reaching a vertical velocity that is within the successful landing conditions for landing and achieving that in the time remaining until impact with the available kinetic energy.

As seen in [1], the boundaries for the fuzzy logic transitions are widest here. The time and distance needed to slow the SH-60 helps to determine the boundaries needed, though there is also some trial-and-error. The Flare stage should not begin at too high an altitude, or it will run out kinetic energy before it is close enough to the ground to safely land. On the other side, it is necessary to observe the behavior in the next two stages to ensure there is enough time after the Flare stage for the Landing and Touchdown stages. When exiting this stage, all values do not have be within the successful landing conditions, as there is still time in the final two stages to achieve those conditions. However, the forward and vertical velocities should both have decreased by about 50%. The altitude of the helicopter should be approximately equal to that of the fuzzy logic boundary value for the beginning of the transition to Landing. The rotor rotation rate will also start decreasing at the end of Flare, though it should remain constant until the Landing transition begins.

For the longitudinal cyclic channel, a new desired forward velocity, `U_TOUCHDOWN` is set. First this value must be chosen. It needs to be low enough to qualify as part of a safe touchdown. `U_TOUCHDOWN` can be chosen based on the safe landing conditions for the SH-60, shown in Table 1. The PID parameters in the velocity tracking controller have to be checked here to ensure they work for a new velocity. With the tuning of this stage, we can ensure that our velocity tracking controller is robust.

There may also be some recursive tuning, looking again at the behavior of the velocity during the first two stages. The final aspect of the cyclic channel is FLARE_MAX_ANGLE. As stated in Section 3.1.1, this value should be 'unlimited.' Because the velocity needs to decrease quickly, the helicopter needs to be able to pitch up dramatically. As in the Steady-State Descent, the limits do not assure that the actual pitch angle of the helicopter is within those bounds. However, by observation of the pitch angle, it can be checked that it is not too high. A FLARE_MAX_ANGLE that is too low will not bring the helicopter to U_TOUCHDOWN quickly enough, so the tuning comes in with setting a high enough limit that does not pitch up the helicopter too far.

In the collective channel, the controller first has to make a choice. In Equation 12, there are two options defined by comparing $-\frac{2h}{\dot{h}}$ with TTI_F where

$$TTI_F = TTL_E + \texttt{TTI\_L}. \tag{19}$$

The full definition of TTL_E can be found in [1]. It is based on an estimate of the kinetic energy remaining in the rotor. More kinetic energy means more gradual maneuvers can be performed to slow the descent, while less energy means the helicopter must flare more abruptly. The $-\frac{2h}{\dot{h}}$ term is an estimate of the actual time remaining until impact, assuming constant vertical velocity. Thus, comparing these two times determines if the control can track a desired descent rate or if it must simply quickly increase the collective command, which are the two options in Equation 12. The TTL_E term contains the parameter TTI_F_MAX which is used to define the total time the controller could spend in the Flare stage, i.e. this amount of time would result in the most gradual flare

trajectory. The two TTI parameters need to be tuned to give the helicopter enough time to perform the flare-out with either option.

The tuning of the collective control in this stage is based on the K_COL and TAU terms. K_COL can be tuned approximately by

$$\texttt{K\_COL} \approx \frac{6m}{\rho R^3 \Omega^2 N c} \tag{20}$$

which are all known properties of the helicopter. Thus tuning can be done solely through TAU. Because the collective control is tracking the vertical velocity, the tuning is performed by adjusting TAU to reduce the vertical velocity by about half by the time the controller is transitioning to the Landing stage. If there is less kinetic energy in the rotor and therefore less time to perform the flare-out, then the controller just uses the FAST_COL_INCREASE parameter. This value should be high and can be tuned the same way as TAU.

Although the helicopter will not be as slow as it needs to be to land, it will continue the trend towards the target velocities during the next stage, as seen by the similar control in Equations 12 and 15. The transition between Flare and Landing should be linear. By the time the rotor starts losing energy, the controller must transition to the Landing and Touchdown stages.

3.2.4 Landing

Like the Pre-Flare stage, the Landing stage is also a transitory stage in the controller. The control is very similar to that of the Flare stage and its purpose is very

similar to that of the Pre-Flare stage. After the flare-out, the helicopter needs to become level again and continue tracking towards U_TOUCHDOWN. The TTI_L term and the upper boundary for the fuzzy logic will be the same. The transition to Touchdown should occur quickly.

The longitudinal cyclic control is almost exactly the same as in the Flare stage with the important difference that LANDING_MAX_ANGLE is set low enough for the pitch angle of the helicopter to be within safe landing conditions by the end of the Landing stage. The value should be an intermediate value between FLARE_MAX_ANGLE and TOUCHDOWN_MAX_ANGLE. The fuzzy logic transition ensures that the pitch angle will not drop too sharply, as the contribution from the flare-out of the previous stage fades and cedes to the contribution from the Landing stage leveling out. The actual pitch angle should peak during the transition from Flare to Landing.

The collective control is also almost identical to that of the Flare phase except for the time to impact value used. This is clear from the definition of TTI_F which is the time remaining in the Flare stage plus TTI_L; the time remaining in the Flare stage is zero once the controller has switched to the Landing stage. Because the controller spends more time in the Flare stage, the parameters in this channel should be tuned based on the behavior of the helicopter in that phase and no further tuning should be required in this stage, except for the value of TTI_L.

As can be seen from the similar control of Flare and Landing, the tuning of these two stages will have effects on each other. The simple control in the final stage, Touchdown, also means that a lot of the behavior of the helicopter during that stage of

the controller depends on the behavior in the previous stage. The vertical and horizontal velocities should continue to decrease through this stage so they can reach their targets in Touchdown.

3.2.5 Touchdown

During the final stage of the controller, all states of the helicopter should be inside the boundaries for successful landing conditions. The control in this stage is not as complex as for the previous stages, as the controller does not spend much time in this stage. When it is in the Touchdown stage, there are also contributions from the Landing stage control. Tuning for this stage is not as involved as for the other stages, and is based on setting the parameters to low enough values to ensure the helicopter will make a successful landing. Only boundaries for when to enter the Touchdown stage are needed for the fuzzy logic conditions are needed, as the lower boundaries are simply when the helicopter impacts the ground, zero height and zero time to impact. The fuzzy logic boundaries for the final three stages all have a large effect on the behavior of the helicopter before landing, and much of the tuning of the last three stages is done recursively and concurrently.

The control for the longitudinal cyclic is based on limiting the pitch angle to be within the successful landing conditions. Thus, `TOUCHDOWN_MAX_ANGLE` will be very small. When tuning this parameter, the actual pitch angle needs to be almost level, so this angle limit needs to be lower than that for a successful landing, see in Table 1. The collective control is no longer tracking any state of the helicopter, but reducing the change in collective. During actual touchdown, there should not be a high rate of change

in collective control. By the end of the Touchdown stage, which is synonymous with impact, the helicopter should have low rates of change in almost every state.

3.3 Results

The tuning results in a controller that can successfully perform an autorotation with the ending conditions in Table 1 and the initial conditions in Table 2. The values of the parameters are in Table 3. These values, along with the values for the other parameters described in Section 3.1.1, are used to simulate a helicopter autorotation. The results of this are in Figure 7. To show the robustness of the controller, the autorotation has been modeled from the other initial conditions given in Table 3, with the results in Figure 8.

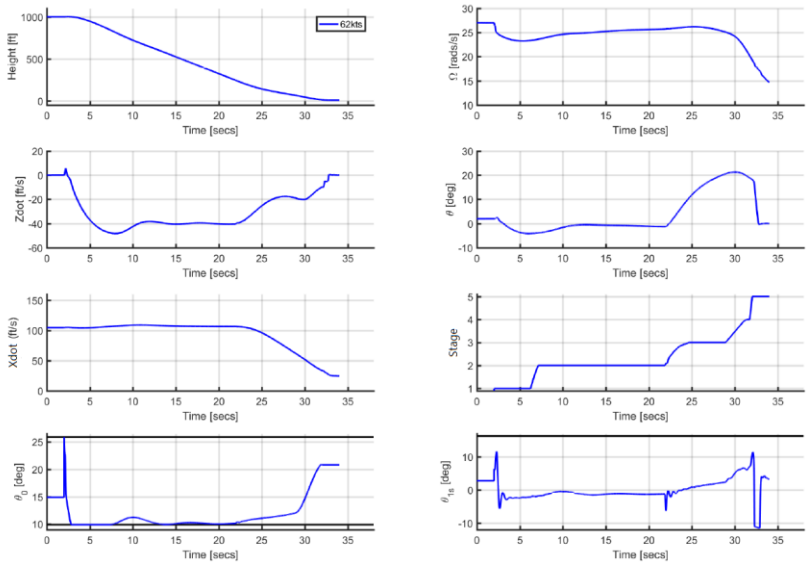

Figure 7 – Successful autonomous autorotation. Shown is (from top left) altitude, vertical velocity, forward velocity, collective control, rotor rotation rate, pitch angle, the stage of the controller, and longitudinal cyclic control

Table 2 – Initial conditions used to demonstrate robustness of the controller

Initial Velocity (knots)	Ground Speed (knots)	Descent Rate (ft/s)	Pitch Angle (deg)	Pitch Rate (deg/s)

62	16	0.35	-0.2	-0.1
40	26	0.6	-10	12
60	16	0.4	0	-0.3
80	18	9	15	-3.2

Table 3 – Autorotation Controller Parameters

Parameter	Definition	Value
RPM_AUTO	Desired main rotor rotation rate for the Steady-State Descent stage	26 rad/s
K_D_SS	Rotor speed time derivative gain for Steady-State Descent collective control	0.1 s^{-1}
K_P_SS	Gain on rotor speed for steady-state descent collective control	0.015 [nd]
TTI_L	Desired time to impact during the Landing stage	2 s
TTI_F_MAX	Maximum cap on the desired time to impact during the Flare stage	7 s
K_COL	Rotor collective gain for the Flare and Landing stages	$3.7x10^{-4}$ $rad \cdot s^2/ft$
TAU	Rotor collective adjustment time constant for Flare and Landing stages	0.8 s
FAST_COL_INCREASE	Collective adjustment rate for rapid adjustments during the Flare and Landing stages	20°/s
U_TOUCHDOWN	Desired forward velocity at touchdown	20 ft/s
U_AUTO	Desired forward speed for the Steady-State Descent stage	105 ft/s
LANDING_MAX_ANGLE	Maximum cap on pitch angle during the Landing stage	12°
TOUCHDOWN_MAX_ANGLE	Maximum cap on pitch angle during the Touchdown stage	1°
TOUCHDOWN_COL_INCREASE	Constant collective pitch rate during Touchdown stage	1°/s

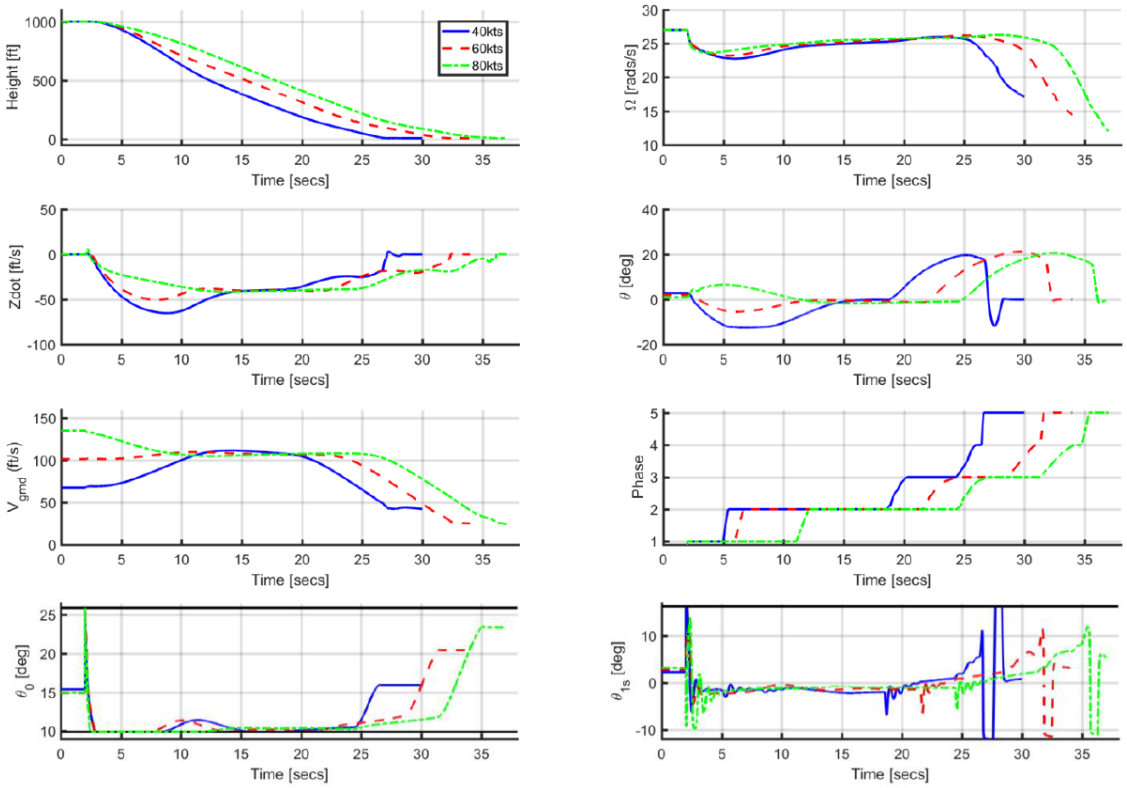

Figure 8 – Successful automatic autorotations starting from various initial conditions, demonstrating robustness

The states at the end of the autorotations in the results shown can be compared with the successful landing conditions. Many of the features described during the tuning can be observed also. The autorotation controller achieves successful autorotation and the next goal is to extend it to the HITL system.

CHAPTER 4. CONCLUSION

4.1 Closing Remarks

In Part One of this thesis, we have discussed the operation of a helicopter, the purpose of autorotation, and the tuning of a controller that helps a pilot perform an autorotation. After engine failure in a helicopter, it is desirable to perform a successful autorotation in any situation, whether in a DVE or not, whether the aircraft is controlled autonomously or by a human pilot. The controller developed in this thesis adapted one designed for autonomous use to one that can aid human pilots as part of a HITL system.

The tuning of a multi-stage, fuzzy logic control is complicated. Although about three dozen parameters are involved, and effects of one stage can be felt across the full control of the autorotation, an appropriate controller was created. The results presented in this thesis show that, with a simulated autonomously controlled helicopter, the parameters found bring about a successful autorotation for a SH-60 Seahawk. The controller used in that process was one designed to be modular and fit into the controller developed by Liverpool. When a human pilot is using the same control commands, they should also be able to perform a successful autorotation.

4.2 Future Work

As mentioned in the introduction, this controller was created to be used as part of a HITL system, developed by Liverpool University. The results were shown in this thesis still fully in a simulated environment, but the next step is to extend that to a system for a piloted helicopter. In the simulated or autonomous environment, the controls that are

produced by the autorotation controller are fed directly to the system modeling the helicopter flight or the autonomous controller. Instead, the controls could be sent to a cockpit display where the pilot attempts 'dot matching.' That is, the pilot sees the value of the collective and longitudinal cyclic the controller has calculated and works to track those controls. The result of the HITL system should be more consistent autorotations, i.e. the pilot always lands the helicopter within the successful landing conditions and there is uniform behavior over many autorotations and pilots. It should also result in improved performance of autorotations in a DVE.

Once this HITL autorotation system has been implemented, there are other directions for the controller to go. As currently realized, it only tracks forward speed and vertical speed and position, leaving the location of the landing site up to the human pilot. With information about the ground, such as maps with locations of mountains, lakes, or other geographical features, the controller could also guide the pilot to an ideal landing site. This controller was developed as a pilot aid in a DVE, because performing an autorotation becomes more difficult if the pilot cannot see the ground. In this case, the pilot would then also have a hard time finding a landing site on the ground. The controller could offer guidance in all control variables to ensure a safe landing in a good location.

PART TWO

DEVELOPMENT OF A NONLINEAR MODEL OF ELECTRIC DRIVES TO BE USED IN THE OPTIMIZATION OF TORQUE PERFORMANCE

CHAPTER 5. INTRODUCTION

Electric drives are an important device in a world where electromobility is becoming more popular. They appear in a wide-range of applications from household appliances (dishwashers and vacuum cleaners) to industrial manufacturing systems (conveyer belts and robots), from automobile applications, such as windshield wipers and power windows, to the motors on electric bikes, and more. The use of electric drives provides many benefits; they are in general more efficient than combustion engines and do not need gas as a power source [8]. An electric drive can provide higher efficiency while using a sustainable fuel source.

The work of this thesis is part of a larger project related to a holistic design process on the electric drive, looking simultaneously at the mechanical and electrical sides and how they interact with each other within a multi-domain optimization with several concurring objectives. The topic of this thesis is based on the electrical and control side of such a drive.

This thesis specifically explores control of the electric drive through direct manipulation of the switches in the B6-bridge. Normally, control is done by giving the inverter a voltage which is translated into open and closed times for the switches, as done in field oriented control (FOC) [9]. A similar method of control is space vector modulation, a subset of torque vector control [9]. We want to discover if direct control of the switches can improve the performance of the drive, whether that means a better response of the direct goals—torque, efficiency—and requirements of the drive or

looking into secondary key performance indicators (KPIs). These secondary KPIs include such metrics as torque ripple, noise vibration harshness (NVH), and power losses.

A suitable mathematical model of a portion of the drive is the focus of this thesis. This model increases understanding of the operation of the drive and helps to narrow in on the effects of direct manipulation of the switches. It will also lead to the goal of creating an optimal switching sequence based on various KPIs. A straightforward mathematical model that is not tied to specific software to solve it can be used offline and online when attempting to optimize the performance of the drive. The model of the switching can also be used modularly in optimization. By focusing on the switching within the power electronics, a detailed model can be developed that could eventually include other aspects of the drive.

The next chapter defines the electric drive and details the specific system used in this thesis. The following chapter is about the development of the mathematical model of the system. Finally, we have a discussion of model predictive control (MPC) and how it could be applied to this system. In each chapter, further work related to each process is suggested as well.

CHAPTER 6. ELECTRIC DRIVE DESCRIPTION

6.1 Overview of the Electric Drive

The electric drive, as will be considered in this thesis, consists of three main parts. These are illustrated in Figure 9. In the center are the power electronics. This is where the control of the drive occurs. In the figure, the input to this block is the switch pattern. The switch pattern, or switching sequence, controls the transistors that make up the B6-bridge inside the power electronics, which is the focus of this thesis. The B6-bridge is an arrangement of six transistors that act like switches, either allowing the power from the

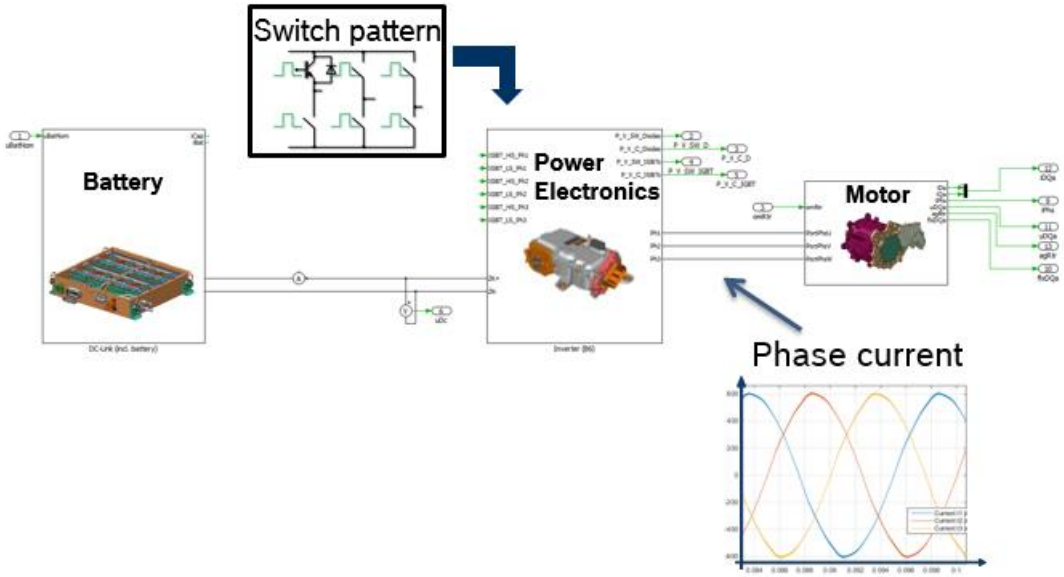

Figure 9 – General block diagram of the electric drive

battery to flow to the load or not. In the drive we consider in this thesis, they are IGBTs, although they will be modeled as ideal switches, meaning any losses or time to change state will be ignored. These switches receive commands to open or close at certain times.

The change of state of the switches is what translates the voltage supplied from the battery into a three phase current, also shown in Figure 9.

The battery supplies a constant voltage. Inside the DC-Link circuit is a capacitor capable of dissipating desired current peaks very quickly and meant to absorb any voltage or current feedback to the battery from the motor. This effect can be quantified by the DC voltage ripple, the variation in the voltage across the capacitor. Minimizing the ripple can allow for a smaller capacitor in the DC-link circuit, saving space in the drive, as the capacitor is currently a third of the volume of the inverter.

Finally, the power electronics are controlling the motor, also sometimes referred to as the load. The behavior of the motor can be quantified in many ways, including the torque, focused on in this thesis, the speed, and other KPIs, such as torque ripple, number of switching events, NVH, and power losses.

6.2 Control of the Electric Drive

Control of an electric drive is usually defined by controlling the behavior to reach either a target speed or a target torque. The input current to the drive is related to control of the speed while input voltage is for control of the torque. Current and voltage are of course related, as are speed and torque. The input current or voltage then indirectly determines a switching sequence, or the times at which each switch opens or closes. However, the basis of this thesis is the idea that the switching sequence can be prescribed and the control of the electric drive comes from the direct manipulation of the switches.

In Figure 10 is a block diagram for the control of the electric drive. In this case, we have current control of the torque. The input is a reference current, which corresponds to a particular voltage. This voltage is fed through some sort of pulse width modulation scheme to create a duty cycle that determines the operation of the switches. The output of the motor is the torque; we also see the secondary effects NVH and DC-Link power. In the control considered for this thesis, the input current and related voltage do not matter, as the switching sequence will be determined externally. The block diagram in Figure 10 demonstrates closed loop control, however in this thesis we focus on open loop control. We want to observe the effects on the outputs shown in this figure when applying various switching sequences. This means the switching sequence will be predetermined, without considering real time performance of the torque.

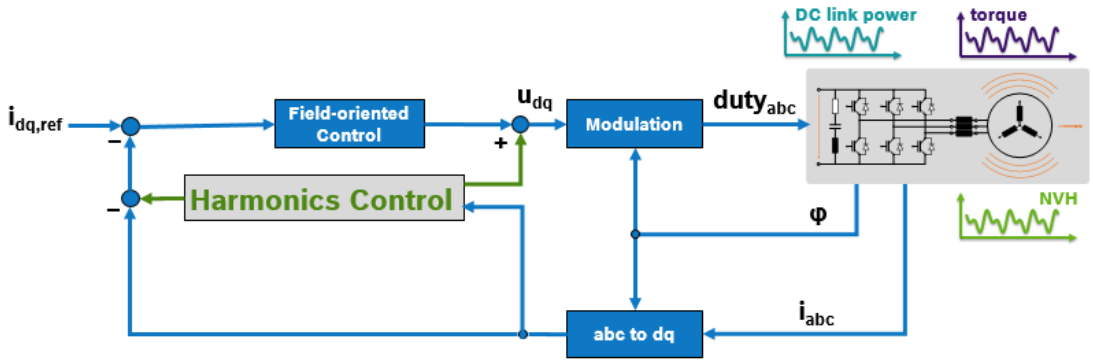

Figure 10 – Block diagram illustrating closed loop field oriented control of the electric drive

The focus on the switching sequence means that analysis of the drive in this thesis can focus on the B6-Bridge part of the power electronics. Only a simplified representation of the motor is needed as well as a similarly simple representation of the

power source. The next chapter features these three elements and the mathematical model thereof.

CHAPTER 7. MODELING THE SYSTEM

7.1 Motivation

The system of the electric drive, especially the switches, can be modeled in different ways. The program PLECS, for example, developed and distributed by the company PLEXIM, is a simulator with a focus on power electronic simulations. It can be embedded in Simulink (developed by Mathworks) and can be used to set up models relatively quickly and to easily run those desired models. There are also various models of the drive already created in Simulink and/or PLECS for use by Bosch. However, the available options allow for less flexibility and transparency which may be desirable to investigate switching patterns as one possible degree of freedom within the control of the electric drive. The model that will be developed and described within this thesis focuses on the battery, B6-bridge, and motor, and simplified versions of those elements, as discussed in the previous chapter.

One expects many advantages to using a modeling approach based on a differential equation description instead of one in PLECS (or a similar 'black box' simulator) or Simulink. The first is an advantage of the time it takes to run the simulation. When calculating a solution solely with Matlab, the solution should be significantly quicker than with PLECS or Simulink, especially if performing some sort of looped process. Having to call the external program in each loop iteration extends the time needed to run each simulation. Proof of this advantage will be demonstrated later in this chapter. Another advantage comes in greater insight and the potential to use the equations in

further work. When the behavior is described by a set of equations, it is much easier to pinpoint the desired values to solve for.

The model consists of three main parts, identified in Chapter 6: The B6-bridge that is the source of control and connects the DC power source to the 3-phase motor. See Figure 9 in the previous chapter for the basic representation; Figure 11 shows a more detailed circuit diagram. The central component of the system, both physically and in regards to the focus of this thesis, is the B6-bridge which affects the current through the load (motor) by manipulating how the voltage from the power source is transmitted, by the switching of power electronic elements, in this case the IGBTs. For the model in this thesis, the switches are represented as ideal IGBTs in anti-parallel with ideal diodes, i.e. there are no losses due to resistance.

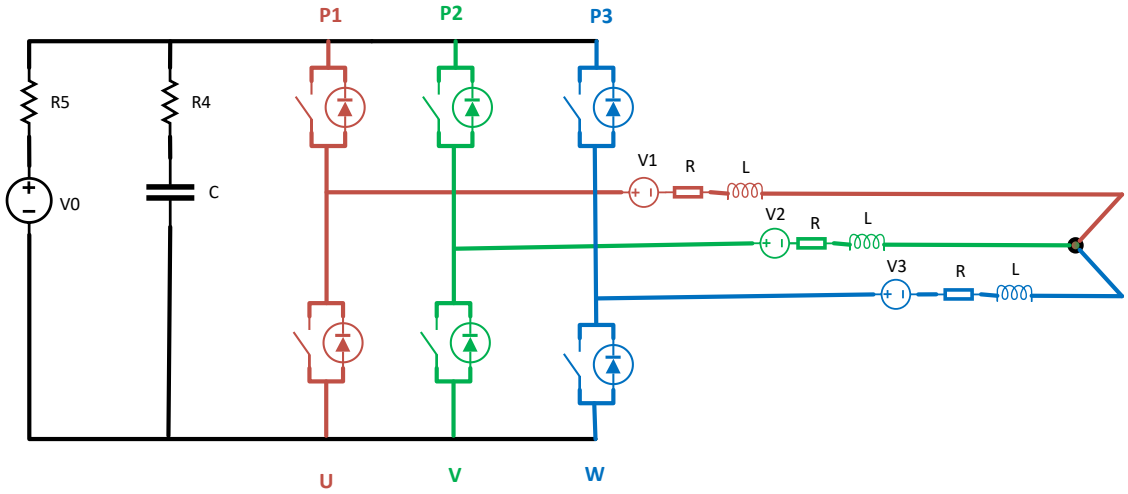

Figure 11 – System circuit diagram showing the three main components as their component parts, here in wye connection, which is used throughout this thesis

On one side of the B6-bridge, the DC side, we have the source, the DC battery, represented by a constant voltage source with a parasitic series resistor and a DC-link

capacitor. The capacitor acts as a storage of DC power and filters out the variations of the DC voltage. Also part of the DC-link model is the resistance for the capacitor, which would otherwise be ideal. The inductance in the battery is not included as for our purposes it is not necessary to consider effects from or on this element. The other side of the B6-bridge connects to the motor, or the load of the circuit, represented by three phases, each consisting of an ideal inductor in series with a resistor and an alternating current voltage source to represent the back electromotive force (emf). For simplicity, the resistance and inductance values are assumed to be the same for each phase. In the three phase model, the phases are connected in a wye formation. This connection is distinct from the delta connection, another common way three phase wires may be arranged, as shown in Figure 12. The ground, or neutral, wire shown in Figure 12 is not represented in the system analyzed in this thesis for simplicity.

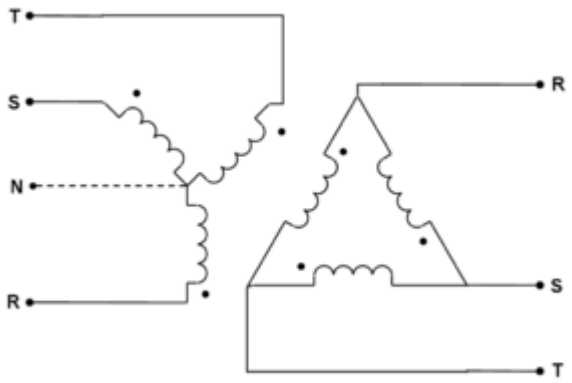

Figure 12 – Wye (left), used in this circuit, and delta (right) connections of three phase systems

Detailed in the following two sections are two different methods of deriving equations that describe the system. The first strategy, the manually switched model, is based on examining all possible equations based on the combinations of open and closed

switches and considering the linear time-invariant system for each configuration. The second strategy, creating a set of non-smooth differential algebraic equations (DAE), is a mathematically-based approach that simplifies the choice of equation when the switches change state. In both cases the single phase circuit is considered first, as a further simplification and well-suited device for understanding and explanation, then the three phase circuit is described.

Table 4 shows the constant physical quantities used in the following equations and the values applied to them in the calculations.

Table 4 – Values of constants in the single and three phase circuits

Constant	Value	Definition
L	101.7 µH	Inductor representing the motor
R	0.0038 Ω	Resistance of the inductor
C	1100 µF	Capacitor in DC-link
R_C	1.07 mΩ	Resistance of the capacitor
V_0	200 V	Battery voltage
R_S	0.10 Ω	Resistance of the battery voltage

7.2 Manually Switched Model

7.2.1 Single Phase Model

The system presented in Figure 11 in the previous section was a three phase system. To simplify initial calculations and as a tool to gain understanding in how to approach the three phase system solution, first the single phase system, circuit shown in

Figure 13, will be investigated. The difference to the three phase system is the load is only one phase of inductor, resistor, and voltage source connected to the DC battery by just two sets of switches.

In the single phase model, with four switches, there are 16 possible combinations of switch states, as shown in Table 5 below. The number of combinations comes from two possible states and four switches, so the total is calculated as 2^4, or 16. Here 0 indicates the switch is open and 1 indicates it is closed. The headings of the table, S1, S2, etc., represent the switches as shown in Figure 13. Immediately, we eliminate seven of these combinations, as it is not allowed for two switches of one switch pair (S1 and S2 or S3 and S4) to both be closed at the same time, as this will create a short circuit. The opposite, both switches being open at the same time, is not dangerous, and instead allows for the useful feature dead time. Dead time is simply a short gap (on the order of 1 μs) between the time one switch on a pair is told to open and the time the other switch is told to close. This is referred to the open circuit case.

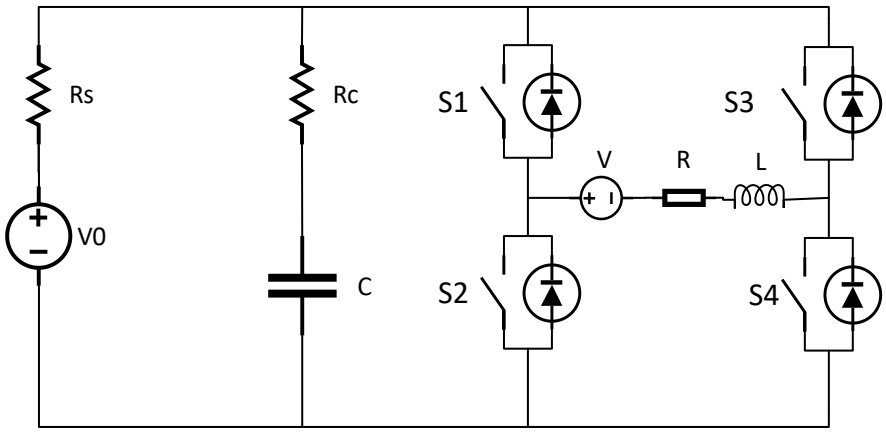

Figure 13 – Single phase circuit with switches

Table 5 – Possible switch states for single phase circuit

S1	S2	S3	S4
0	***0***	***0***	***0***
0	***0***	***0***	***1***
0	***0***	***1***	***0***
0	0	1	1
0	***1***	***0***	***0***
0	**1**	**0**	**1**
0	**1**	**1**	**0**
0	1	1	1
1	***0***	***0***	***0***
1	**0**	**0**	**1**
1	**0**	**1**	**0**
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

The possibilities have been narrowed down from 16 to 9, as indicated by the bold rows in Table 5, which are now the only ones that will be investigated. The next step is to sketch the possible circuits that result from the closed and open gate combinations, assuming a closed switch allows current to flow freely in both directions with cooperation from the anti-parallel diode and an open switch only allows the current to go one direction as only the anti-parallel diode is now in effect. Essentially, the switch and diode pairs are simplified to either open circuits or wires. This perspective allows the more complicated nonlinear effects of the diodes and switches to be ignored mathematically. The switch state will just instigate a choice of which set of differential equations to use.

From this graphically-inspired analysis, one can observe that different switching conditions lead to the same circuit topology. Thus, the 9 switch combinations lead to only 3 distinct circuits. These resulting three circuits are shown in the below circuit

drawings in Figure 14. Using Kirchhoff's Voltage Law, three different sets of first- or second-order differential equations can be obtained based on these circuits. The following equations correspond to the circuit drawings in Figure 14.

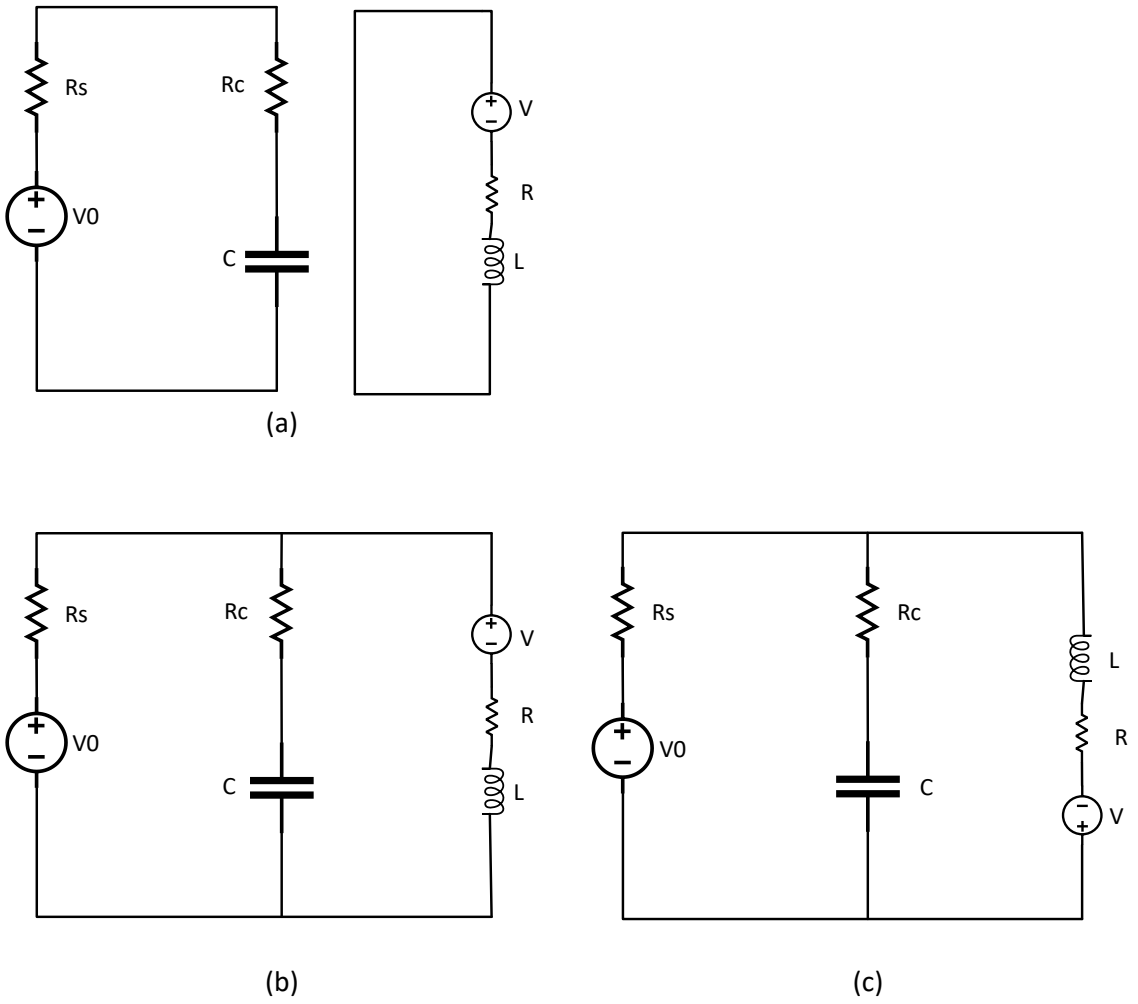

Figure 14 – Possible 'simplified' circuits for single phase model; (a) when either both high side or both low side switches are closed or that diode is active; (b) and (c) when the switch pairs have opposite switches closed or opposite active diodes, the direction of the voltage drop in the load elements depends on the sign of the current and which switch pair is high

$$(R_C + R_S)i_C(t) + \frac{1}{C}\int_0^t i_C(t)dt - V_0 = 0 \tag{21}$$

$$L\frac{di_L(t)}{dt} + Ri_L(t) - V(t) = 0 \tag{22}$$

$$(R_C + R_S)i_C(t) + R_C i_L(t) + \frac{1}{C}\int_0^t i_C(t)dt - V_0 = 0 \tag{23}$$

$$L\frac{di_L(t)}{dt} + (R + R_C)i_L(t) + R_C i_C(t) + \frac{1}{C}\int_0^t i_C(t)dt - V(t) = 0 \tag{24}$$

$$(R_C + R_S)i_C(t) - R_C i_L(t) + \frac{1}{C}\int_0^t i_C(t)dt - V_0 = 0 \tag{25}$$

$$L\frac{di_L(t)}{dt} + (R + R_C)i_L(t) - R_C i_C(t) - \frac{1}{C}\int_0^t i_C(t)dt - V(t) = 0 \tag{26}$$

Equations 21-26 use the properties from Table 4. They contain two states: currents i_C and i_L, which represent the currents in the two loops, the left and right loops, respectively in Figure 14; i_C is defined as the current through R_S and i_L is the current through R and L.

In Equations 21 and 22 (case (a)) we can see the decoupled case, where the DC-link is not affecting the behavior of the load. These equations can be solved simply, as each contains only i_C and i_L, respectively. The sets of Equations 23 and 24 (case (b)) and Equations 25 and 26 (case (c)) are very similar, as their circuits are too, where the only difference is the direction of the voltage source. The difference is in how the two loops interact. The terms that represent elements from the other loop (e.g., $R_C i_L$ in Equations 23 and 25) are positive in the first set of equations and negative in the second set.

7.2.1.1 Solving the Equations

Once we have constructed the equations that describe the relevant circuits, we need to solve them to obtain the temporal behavior of the dynamic system. The equations can be discretized and a Newton or Euler method solver can be applied. Matlab provides ODE solvers, such as `ode45` which only need the equation and initial conditions defined. The solution is straightforward for Equations 21 and 22 above as each equation only has one state variable. However, in Equations 23-26 above, the state variables are shared and i_C and i_L do not have the same highest order derivative, making a solution somewhat more difficult. This issue can easily be solved by taking derivatives of the equations until they are equal. Applying this approach, for example, to Equations 23 and 24 above results in the following:

$$(R_C + R_S)\frac{d^2 i_C(t)}{dt^2} + R_C\frac{d^2 i_L(t)}{dt^2} + \frac{1}{C}\frac{di_C(t)}{dt} = 0 \tag{27}$$

$$L\frac{d^2 i_L(t)}{dt^2} + (R + R_C)\frac{di_L(t)}{dt} + R_C\frac{di_C(t)}{dt} + \frac{1}{C}i_C(t) - \frac{dV(t)}{dt} = 0. \tag{28}$$

This also helps to eliminate the integral added from the capacitor and have the state i_C only in the zeroth or higher order derivatives, as can be seen in Equations 27 and 28. When the highest order derivatives match, the equations can be rewritten in the form

$$M\ddot{x} + D\dot{x} + Kx = f \tag{29}$$

which is a common formulation in many mechanical problems. A similar formulation can be used for electrical systems as well. In mechanical systems, M is the mass matrix,

associated with acceleration; in the electrical case it is the inductance matrix, related to the derivative of current. The damping matrix, D, is related to velocity; in the electrical case it is the resistance matrix, acting on the current. Finally, K is the stiffness matrix, characterizing the stiffness of a spring and relating to the spring's deformation (i.e. a position); in the electrical analog it is the capacitance acting on the charge. The forces, f, relate in the electrical case to the voltages from the voltage sources, which are the electrical driving force. In Equations 27 and 28, one could also write i_C and i_L as charges, q_C and q_L, drawing a better comparison between the mechanical and electrical representations. Using Equations 27 and 28 to give an example of the matrix form yields

$$\begin{bmatrix} R_C + R_S & R_C \\ 0 & L \end{bmatrix} \begin{bmatrix} \frac{d^2 i_C}{dt} \\ \frac{d^2 i_L}{dt} \end{bmatrix} + \begin{bmatrix} \frac{1}{C} & 0 \\ R_C & R + R_C \end{bmatrix} \begin{bmatrix} \frac{di_C}{dt} \\ \frac{di_L}{dt} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} i_C \\ i_L \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{V} \end{bmatrix}. \tag{30}$$

This is now a straightforward representation of a system of ODEs and can be easily rewritten as four 1st order ODEs. One potential inconsistency is how the general definitions of the matrices, described above, are not reflected in the exact definitions seen here. Matrix M has R and L values, instead of just L values, while all the R values should be in matrix D. This pattern continues with the mix of R and C values in matrix D and not all C values being in matrix K. This results from taking the derivatives of each equation. While only one derivative was needed for Equation 24, two were needed to ensure Equation 23 had a second order derivative to match. While Equation 30 can still be used to solve for the currents, it is unconventional and against some common sense.

We will see a better solution in Section 7.3, which discusses the non-smooth differential equation solver.

7.2.1.2 Initial Conditions

To solve the equations in the matrix form, the correct initial conditions are required. There are four initial conditions, representing the two states and their first derivatives. Because the set of differential equations to solve is chosen based on the current switch state, each time the switch state changes, a new set of equations is called as well as requiring a new set of initial conditions. This adds more complexity, which can easily lead to errors, and more calculation effort in the solution of these equations. The disadvantages are more pronounced in the three phase system which has more equations and more initial conditions. The initial current through the inductor is $i_L(0)$; together with the initial voltage of the capacitor, $v_C(0)$, they define

$$i_C(0) = \frac{-V_0 - sign\big(i_L(0)\big) * v_C(0) + R_S i_L(0)}{R_C + R_S}. \tag{31}$$

The initial values of $\frac{di_L(0)}{dt}$ and $\frac{di_C(0)}{dt}$ also have to be solved for.

For the first switch state in a simulation, $i_L(0)$ and $v_C(0)$ need to be prescribed. After that, the final value of i_L of each switch state is used as the first value of the next switch state. Unfortunately, this does not work for i_C or the derivatives, since they are calculated directly or indirectly from the initial voltage of the capacitor, and the changes in voltage during each change in switch state cause the current to jump, therefore

requiring Equation 31 to be evaluated each time. The necessary calculation comes from the definition of voltage in a capacitor as

$$V = \frac{1}{C}\int I\, dt \tag{32}$$

and the definition of voltage in an inductor as

$$V = L\frac{dI}{dt}. \tag{33}$$

The voltage in the capacitor must be continuous, but not differentiable, meaning the current in the capacitor may not be continuous. The opposite relationship between voltage and current in the inductor means that the current must be continuous, and it can easily be carried from one state to the next. These observations suggest that if the states were q_C and i_L instead of i_C and i_L the solution could be simpler. This will be implemented in the method described in Section 7.3.

7.2.1.3 Zero Crossing Behavior

There is one further step to be taken to fully describe the behavior. Equations 21-26 describe the behavior of the currents accurately if the diode and switch pairs are simplified to either open circuits or wires, which is sufficient if we only consider the cases where in a switch pair one switch is open and the other closed. However, there may also be the situations where both switches are open on a switch pair, the open circuit case, (this is related to the dead time behavior, described previously). This means that the wire assumption for the active diode is inaccurate and should truly be a ‘one-way’ wire.

When the switch state is one of the italicized rows in Table 5, the differential equations that describe the model change when the current i_L attempts to change sign. That is, in addition to the change of the switch states, which is an externally triggered event that changes the equations describing the model, there is also a change in equations due to the change of current direction, or sign, which can be considered an internally trigged event that will also result in a new set of equations needed to describe the model. Depending on which switches are closed, the current i_L will be able to change sign and change the circuit representation, or the current i_L will not be able to change sign and become zero; both cases result in the choice of a different set of equations.

Before the zero crossing event occurs, when a switch pair is in an open circuit state, depending on the sign of the initial current through the inductor, the switch pair can be assumed to be equivalent to a switch pair with one switch closed. For example, if the current is positive (here defined as flowing from right to left), then if the first switch pair (S1 and S2) has both switches open, as shown in Figure 15, this would be same as if S1 were closed, and if the current is negative, as it is as if S2 is closed. This is due to the diode part of the switch diode pair.

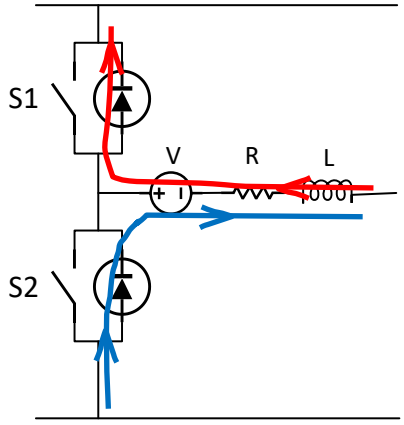

Figure 15 – Partial circuit, representing behavior of open circuit case

Once the critical point is reached, i.e. when i_L reaches zero and wants to change direction, this assumption can no longer be used, as the current will either flow through the opposite diode or neither diode. Instead, we must use a different equivalent circuit. Luckily, the describing circuits are not new ones, but are the three configurations which are already known. If the initial circuit is (b) or (c) from Figure 14, then after the zero crossing of i_L the new circuit is (a). If the initial circuit is (a), then the only change is that the current stops flowing in the load-side loop and continues as before on the DC-link side, as they are already separated.

In this case, where zero-crossing of the current i_L is treated as an additional trigger to switch between differential equations, the solution strategy has to be adjusted. As in the initial stage of the solution, one has the trigger from changes in the configuration of the switches; one has to consider changes in the sign of i_L (i.e. zero crossing) as a trigger to go from one set of differential equations to another one. A solution algorithm needs to find the point in time when zero-crossing of i_L occurs and adapt the temporal discretization to allow for an accurate solution. After that time point is found, normal

calculation with the original time step size would resume using new initial conditions calculated based on that time point. In this way, the two periods of operation (before and after the zero crossing) would be strung together.

7.2.2 Three Phase Model

Using the experience of solving for the single phase model, the description of the system of the three phase model follows a similar path. Figure 16 below shows the three phase circuit again, which is similar to the single phase, except there are now six switches used for control and three phases of the motor. Starting from the evaluation of all possible switch states we again exclude the short circuit states, i.e., those states with both switches on one switch pair closed, leaving only the options presented in Table 6. This reduces the total number of possibilities from 64 (from 2^6, since there are now six switches instead of only four) to 27, which is more manageable, and will be reduced further, similarly to the analysis for the single phase circuit. The focus is on the bold rows of Table 6, those with no open circuit case.

In this section, switch states will be referred to in their binary code, i.e. if the first switch set is open circuit, the second has high side engaged, and the third low side engaged, that would be indicated by 001001. This binary code can be seen in Table 6, when concatenating the switch states S1...S6 in one given row.

Table 6 – Possible switch states for three phase circuit

S1	S2	S3	S4	S5	S6		S1	S2	S3	S4	S5	S6
0	0	0	0	0	0		**0**	**1**	**0**	**1**	**1**	**0**
0	0	0	0	0	1		0	1	1	0	0	0
0	0	0	0	1	0		**0**	**1**	**1**	**0**	**0**	**1**

0	0	0	1	0	0
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	0	1
0	0	1	0	1	0
0	1	0	0	0	0
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	**1**	**0**	**1**	**0**	**1**

0	**1**	**1**	**0**	**1**	**0**
1	0	0	0	0	0
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	**0**	**0**	**1**	**0**	**1**
1	**0**	**0**	**1**	**1**	**0**
1	0	1	0	0	0
1	**0**	**1**	**0**	**0**	**1**
1	**0**	**1**	**0**	**1**	**0**

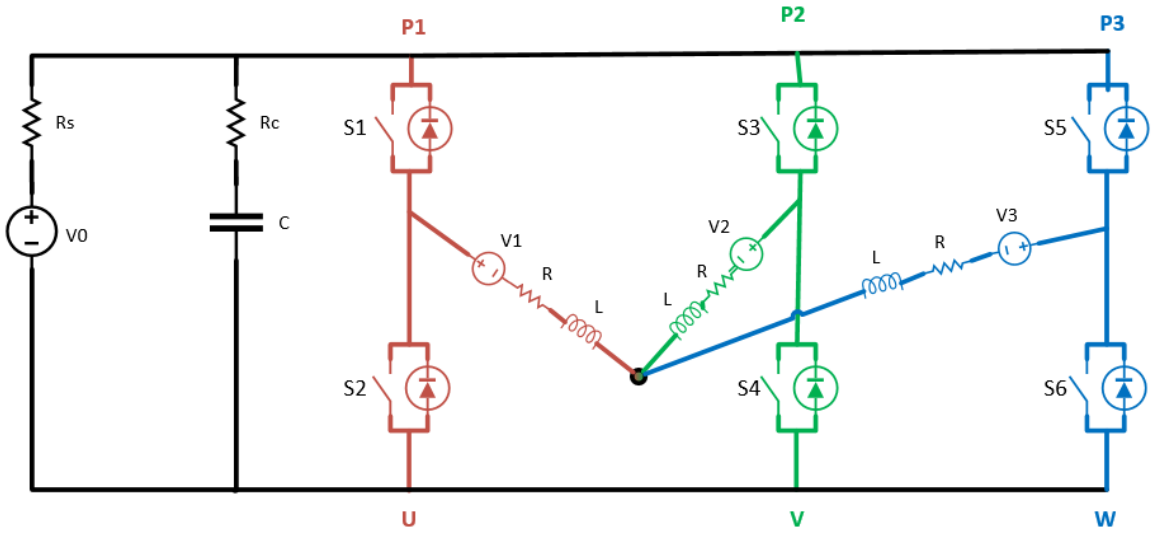

Figure 16 – Three phase circuit with labeled switches

A further reduction of the number of circuits to investigate can be obtained by observing that the switch pairs are fairly interchangeable. For example, switch states 010110 and 011001 would result in almost identical circuits, except that the phase which is connected to the positive side of the DC-link switches between the second and third phases. The circuits resulting from these two switch states are illustrated in Figure 17. Other combinations can be compared similarly, with the result that there are only three different circuits to consider, shown in Figure 18. This is achieved by introducing a

generic nomenclature, i.e. substituting A, B, and C for phase 1, 2, or 3 and using the differential equations as required by the effective switch state.

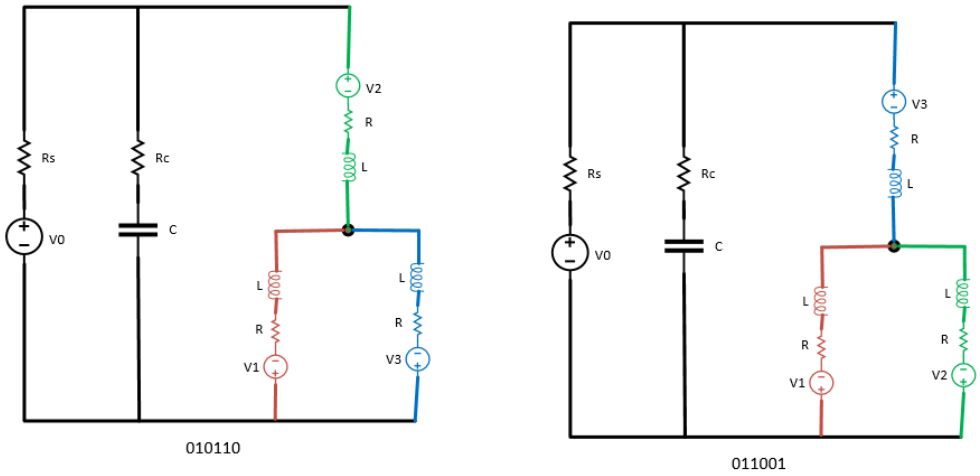

Figure 17 – Two circuits demonstrating the similar layout for two switch states

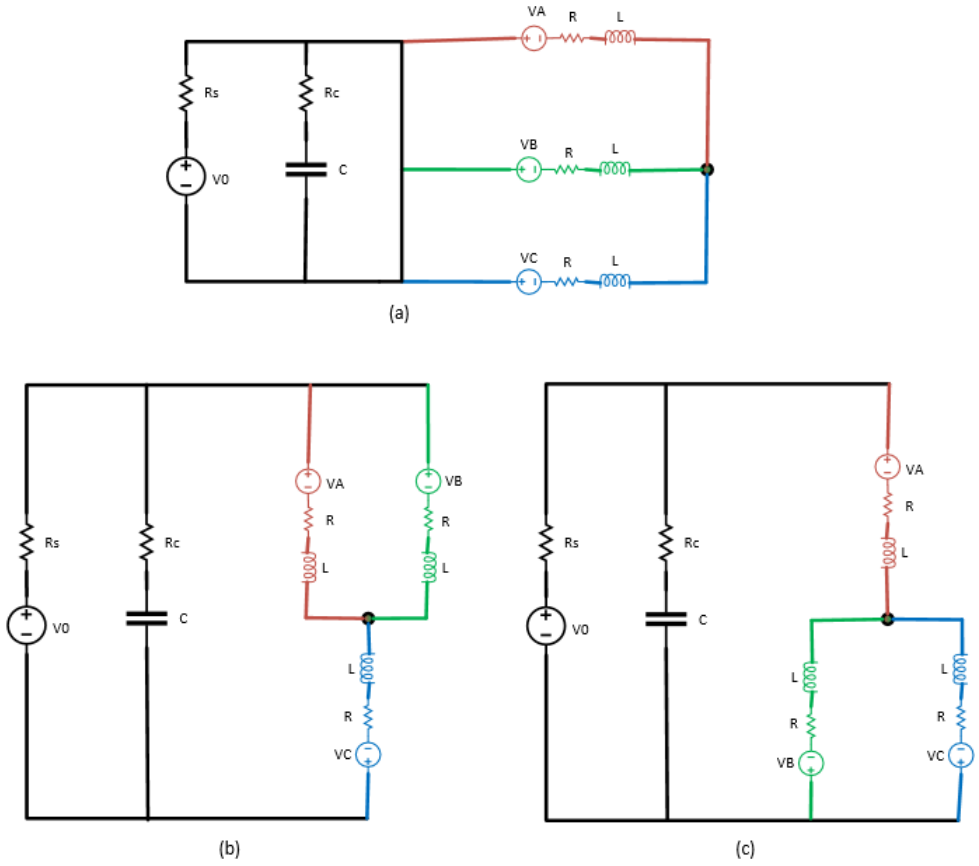

Figure 18 – The three general circuit configurations for the three phase model; (a) is when the switch pairs are all high or low, (b) is if two are high and one is low, and (c) is if one is high and two are low

As in Section 7.2.1, we can write equations for each of these circuit representations, and put them into matrix form so they can be solved by Matlab's ODE solver. From Figure 18 we can see that there are now three loops and three independent equations, one more than in the single phase representation. The relevant currents are i_C which goes through R_S as before, but now there are two currents, i_{LA} and i_{LB} that go through two of the phases. Although there are three phases, due to the y-connection and Kirchhoff's Current Law, only two currents need to be defined, since the third current can be derived from

$$i_{LC} = -(i_{LA} + i_{LB}) \tag{34}$$

The following equation is the set of ODEs for (b) in Figure 18.

$$\begin{bmatrix} R_C + R_L & -R_C & 0 \\ 0 & 2L & -L \\ 0 & -L & 2L \end{bmatrix} \begin{bmatrix} \frac{d^2 i_C}{dt^2} \\ \frac{d^2 i_{LA}}{dt^2} \\ \frac{d^2 i_{LB}}{dt^2} \end{bmatrix} + \begin{bmatrix} \frac{1}{C} & -\frac{1}{C} & 0 \\ -R_C & R_C + 2R & -R \\ 0 & -R & 2R \end{bmatrix} \begin{bmatrix} \frac{di_C}{dt} \\ \frac{di_{LA}}{dt} \\ \frac{di_{LB}}{dt} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{C} & \frac{1}{C} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_C \\ i_{LA} \\ i_{LB} \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{V}_B - \dot{V}_A \\ \dot{V}_C - \dot{V}_B \end{bmatrix} \tag{35}$$

This was developed similarly to the equations in the single phase case. The voltages have derivatives because we are able to represent the AC voltage sources as sine functions.

7.2.2.1 Initial Conditions

As in the single phase strategy, the derivation of the correct initial conditions at the beginning of each new time period with updated switch states are important and recurring. In the three phase case, there are six initial conditions to define for each iteration. The ABC substitution introduced above adds an extra difficulty with respect to the initial conditions, as the current values for initialization from the previous calculation period do not correspond directly to the next.

The three current initial conditions come from the defined values of $i_{L1}(0)$, $i_{L2}(0)$, $i_{L3}(0)$, and $v_C(0)$. The first two load currents give the value for the third, as from Equation 34, but as the three currents cycle through the ABC definition, any two of the three initial values could be needed for a calculation. As in the single phase case, the initial voltage in the capacitor is used to calculate the initial current:

$$i_C(0) = \frac{-V_0 - sign\big(i_{LA}(0)\big) * v_C(0) + R_S i_{LA}(0)}{R_C + R_S} \tag{36}$$

The other three initial conditions, the derivatives of the three currents, also need to be calculated.

As described in Section 7.2.1.2, the load currents and the capacitor voltage are continuous but not differentiable. This means that only the final values of the load currents can be used as initial values for the next solution. The initial value of i_C has to be calculated according to Equation 36 for the same reasons presented in the previous

section. This also means the derivatives of the currents have to be recalculated as well as would be indicated by the non-differentiable nature of the currents.

Finally, the values used in these calculations have to be carefully defined according to the ABC definition. This substitution allows for fewer sets of equations to be used, but requires that with each new switch state, not only does the correct set of equations need to be chosen, but the three phases need to be redefined as A, B, or C.

7.2.3 Results

The single phase and three phase model were both created in Matlab and the results were compared with those derived from a PLECS simulation of a suitable model. However, they were only modelled to the point that a single switch state could be solved for. The complications of the ABC substitution made stringing together different switch states difficult to model, as well as the accuracy of the initial conditions calculated with each switch state change. This difficulty only adds to the limitations and issues mentioned in the above sections. For example, we see how the number of switch states increases greatly from single phase to three phase. If we wish to look at larger systems, the depiction of each switch state is no longer practical. Keeping track of initial conditions when the switch state changes also poses difficulties. While the Matlab method does save time over the PLECS method, there might be the possibility to speed up the calculations further when finding a more holistic approach, where the necessity to combine and initialize the adequate differential equations may be obsolete. All of this suggests there would be several advantages if there was one definition of the system that unified all the various circuits depicted above.

7.3 Non-Smooth Differential-Algebraic Equation Model

The goal of this section is to derive a holistic modeling approach, a way to describe the model as a whole and not try to simplify the switch and diode pairs to either open circuits or wires, but find a mathematical representation of them. The method developed in this section is based on work done in [10], where a similar non-smooth model of a buck converter circuit was created. In this thesis, we are not using as much of the deep mathematics as are behind the approach in [10]. But, based on insights from the method built in Section 8.2, we develop a model similar to the end-result of [10].

Also in discussed in Section 8.2 were many drawbacks to that model and the way it was developed. Some advantages of the non-smooth model include the development of just one set of equations and a redefinition of the states of the system. As will be shown in this section, the initial conditions and the solution are simpler as well. This method will solve many of the issues brought up in the previous section. It will also produce a comprehensive description of the drive.

The element that was used in [10] and distinguishes this method is the addition of a function that describes the nonlinear behavior of the switch and diode. Figure 19 below shows an example of the equation that will be applied later. This represents a two-way switch where the direction of the current can change based on some outside input.

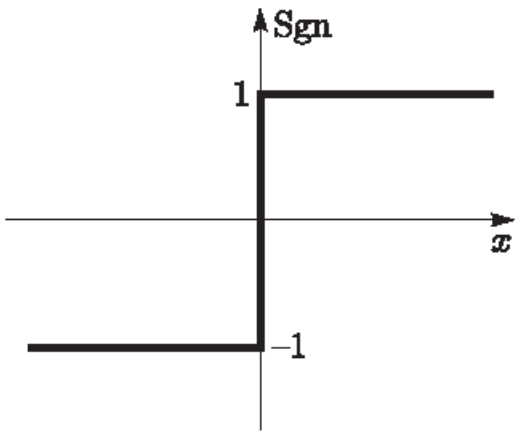

Figure 19 – Graph of the function $Sgn(x)$

7.3.1 Single Phase

For the sake of simplicity, only the cases where one switch of each pair is open and the other one is closed will be considered. This means we ignore dead time (the open circuit case) and assume ideal performance where switches open and close at exactly the

same time. In this case, instead of considering each switch S1 through S4 separately, the state of each switch pair is used for the description. Each state of a switch pair can now be indicated by 1 or -1, indicating, respectively, high or low switch engagement, as seen in Table 7, where P1 and P2 denote the two switch pairs. For single phase, this means there are only four options (2^2, two switch pairs with two options each). These are comparable to the bold but not italicized lines in Table 5.

Table 7 – Switch states for single phase circuit, with switch pairs view

P1	P2
1	1
1	-1
-1	1
-1	-1

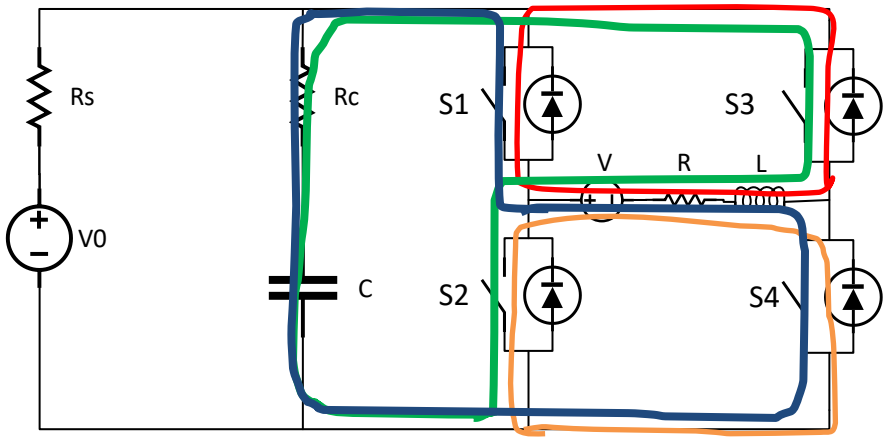

Figure 20 – Single phase circuit showing the possible paths of current for the four color-coded options in Table 7

The first step is to formulate equations for the circuit as a whole. In this case, we can start from four equations, as there are four loops in the circuit and thus four independent currents, as shown in Figure 20. This can be proved using Kirchhoff's Current Law. To include the switch/diode pairs S1…S4, a voltage termed λ_i is

introduced at each pair, with the index *i* corresponding to the index of the switch. Each λ_i is zero, if current is conducted through the switch, i.e. zero voltage drop at the conducing switch. If no current is flowing, λ_i is greater than 0. Since we are disregarding open circuit cases, the λ terms will always be complimentary. If λ_1 is 0, λ_2 will be non-zero, and vice versa. This holds for λ_3 and λ_4 as well. With this introduction of the λ_i, the resulting equations are:

$$-V_0 + R_S\dot{q}_1 + R_C(\dot{q}_1 - \dot{q}_2) + \frac{1}{C}(q_1 - q_2) = 0 \tag{37}$$

$$-\frac{1}{C}(q_1 - q_2) - R_C(\dot{q}_1 - \dot{q}_2) + \lambda_1 + \lambda_2 = 0 \tag{38}$$

$$-V + R(\dot{q}_3 - \dot{q}_4) + L(\ddot{q}_3 - \ddot{q}_4) - \lambda_2 + \lambda_4 = 0 \tag{39}$$

$$-L(\ddot{q}_3 - \ddot{q}_4) - R(\dot{q}_3 - \dot{q}_4) - V - \lambda_1 + \lambda_3 = 0 \tag{40}$$

where the state variables are the charges, q_i. Figure 21 shows the circuit with the currents and lambdas indicated. A few substitutions and assumptions can be made to reduce this to two equations with only two state variables. Some assumptions will be made from the work of the previous section, from which we generally know what the equations will look like.

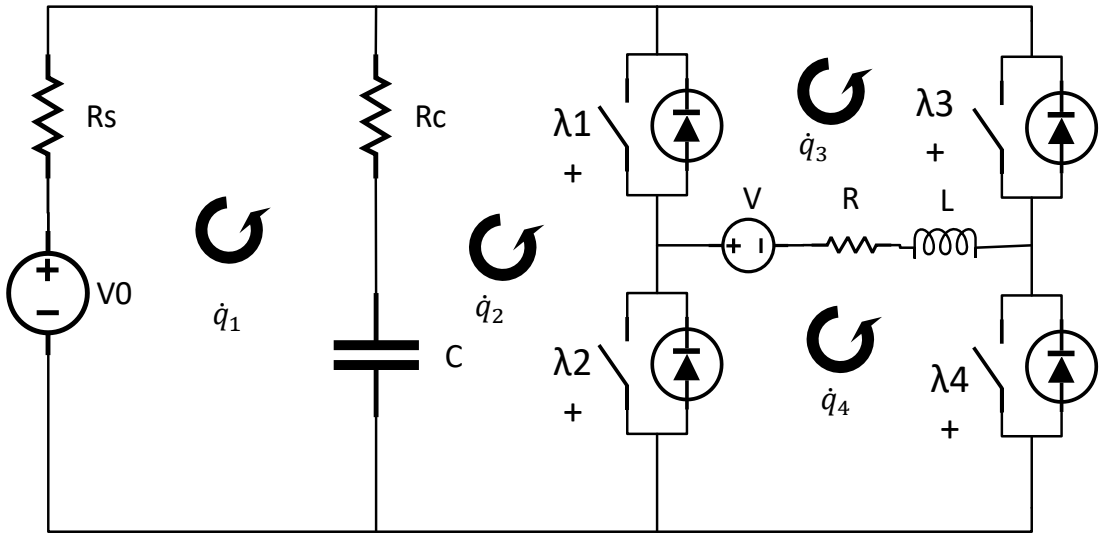

Figure 21 – Single phase circuit with currents and lambdas

The first simplification is to redefine the state variables. We define

$$g_C = q_1 - q_2 \tag{41}$$

$$i_C = \dot{q}_1 - \dot{q}_2 \tag{42}$$

$$i_L = \dot{q}_3 - \dot{q}_4 \tag{43}$$

where g_C is the charge in the capacitor, i_C is the current through the capacitor, and i_L is the current through the load. We can also notice that Equations 39 and 40 are redundant, as the definitions of P1 and P2 require the complementarity of λ_1 and λ_2 and of λ_3 and λ_4. If we substitute Equations 41-43 into Equations 37-40 and use the assumption of complementarity, we have:

$$-V_0 + R_S(i_C - \dot{q}_2) + R_C i_C + \frac{1}{C} g_C = 0 \tag{44}$$

$$-\frac{1}{C}g_C - R_C i_C + \lambda_1 + \lambda_2 = 0 \tag{45}$$

$$-V + Ri_L + L\frac{di_L}{dt} - \lambda_2 + \lambda_4 = 0. \tag{46}$$

Although $\dot{q}_2$ still appears in Equation 44, from examination of Figure 20 we can see that this value is either i_L or 0. Therefore, we can introduce here our first choice function, where the value of P1 and P2, as well as the direction of the current i_L decides if a term is included in the set of equations or not. This function is indicated as $Sgn(P1, P2)$ because the result changes sign based on the comparison of P1 and P2 (See Figure 19). As the next step we solve for λ_2 in Equation 45 and substitute this into Equation 46. From this we end up with two equations, which we expect, not just because we already saw this result, but because from Figure 20 we can see that each option in Table 7 leaves the circuit with only two loops. These two equations are

$$-V_0 + R_S(i_C - Sgn(P1, P2)i_L) + R_C i_C + \frac{1}{C}g_C = 0 \tag{47}$$

$$-V + Ri_L + L\frac{di_L}{dt} + \frac{1}{C}g_C + R_C i_C - \lambda_1 + \lambda_4 = 0 \tag{48}$$

where the choice function, or sign function, is defined as

$$Sgn(P1, P2) = \begin{Bmatrix} 1 \ if \ P1 > P2 \\ 0 \ if \ P1 = P2 \\ -1 \ if \ P1 < P2 \end{Bmatrix} \tag{49}$$

where the first and third rows refer to either the blue or green circuits in Figure 20 and the second row refers to the red and orange circuits in Figure 20.

The final step is to insert the sign function into Equation 48. The remaining λ_i terms indicate that there needs to be some contingency on the state of the switches. From observing Figure 20 again, we see that the position of the switches controls if DC link side of the circuit is connected to the load side of the circuit. This gives the final equations:

$$-V_0 + R_S(i_C - Sgn(P1, P2)i_L) + R_C i_C + \frac{1}{C} g_C = 0 \tag{50}$$

$$-V + Ri_L + L\frac{di_L}{dt} + Sgn(P1, P2)(\frac{1}{C} g_C + R_C i_C) = 0 \tag{51}$$

The i_L term in Equation 50 and the i_C and g_C terms in Equation 51 create the link between the two sides of the circuit in mathematical terms. We now consider g_C and i_L the state variables. This is simpler than using i_C and i_L because the highest order derivatives are now equal.

7.3.1.1 Solving the Equations

By defining g_C and i_L to be the state variables, the difference in order of highest derivatives is removed, as i_C is defined as the first derivative of g_C. Rather than take derivatives to rewrite the equations, we follow the line of thought given in [10] that treats the system as one of a set of differential algebraic equations (DAE) instead of ordinary

differential equations (ODE). When Equations 50 and 51 are rewritten, the DAE representation becomes clearer:

$$i_C = \left(V_0 + Sgn(P1, P2)R_S i_L - \frac{1}{C} g_C\right)\left(\frac{1}{R_C + R_S}\right) \tag{52}$$

$$\frac{di_L}{dt} = \left(\frac{1}{L}\right)\left(V - Ri_L - Sgn(P1, P2)\left(\frac{1}{C} g_C + R_C i_C\right)\right). \tag{53}$$

These equations are part of a DAE system. The first equation can be solved algebraically, assuming we know the values of $i_L$and g_C, which must be defined as initial conditions, for the current through the inductor and for the voltage across the capacitor divided by the capacitance, respectively. The second equation, however, is a differential equation because it has both i_L and the deritvative of i_L.

When the initial i_L^A and g_C^A are defined, the following sequence of equations will solve for i_L^E and g_C^E, as well as i_C^E. The scheme used is a modified version of Moreau's time-stepping method, where i_L is approximated with an Euler step and g_C with the trapezoidal rule [10]. An intermediate step g_C^M for the capacitor charge is defined halfway between the beginning and the end of the integration interval Δt:

$$i_C^A = \left(V_0 + Sgn(P1, P2)R_S i_L^A - \frac{1}{C} g_C^A\right)\left(\frac{1}{R_C + R_S}\right) \tag{54}$$

$$g_C^M = g_C^A + \frac{\Delta t}{2} i_C^A \tag{55}$$

$$i_L^E = i_L^A + \frac{1}{L}\Delta t\left(V - Ri_L^A - Sgn(P1, P2)\left(\frac{1}{C} g_C^M + R_C i_C^A\right)\right) \tag{56}$$

$$g_C^E = g_C^M + \frac{\Delta t}{2}\left(V_0 + Sgn(P1, P2)R_S i_L^E - \frac{1}{C} g_C^M\right)\left(\frac{1}{R_C + R_S}\right) \tag{57}$$

$$i_C^E = \left(V_0 + Sgn(P1, P2)R_S i_L^E - \frac{1}{C} g_C^E\right)\left(\frac{1}{R_C + R_S}\right). \tag{58}$$

These calculations are repetitively performed in a loop and any switching event leads to a change in P1 or P2 and thus a change in the output of the choice function and a new set of equations to solve. Thus the equations with the choice function are always valid, regardless of the switch state. The internal switching event, zero crossing of i_L is not covered by the choice function because it will not occur as we have ignored the open circuit case.

7.3.2 Three Phase

As in Section 7.2, the three phase case is more complicated than the single phase case. Again, the open circuit case is ignored, leaving only the states shown in Table 8, which correspond to the bold rows in Table 6. The development of the model for the three phase circuit is very similar to that for the single phase circuit in Section 7.3.1. This section will be slightly abridged because of that relationship.

Table 8 – Switch states for three phase circuit, with switch pairs view

P1	P2	P3
1	1	1
1	1	-1
1	-1	1
1	-1	-1
-1	1	1
-1	1	-1
-1	-1	1

-1 -1 -1

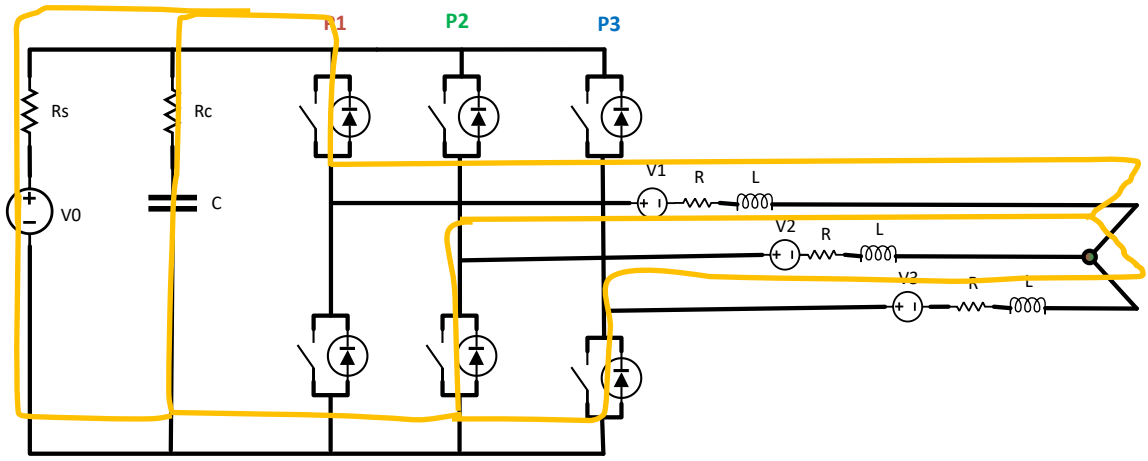

Figure 22 – Three phase circuit with a sketch demonstrating one possible path for the currents based on the color-corresponding row in Table 8

The first step is the same as in the single phase case, writing the equations for the circuit as a whole. From Kirchhoff's Current Law, it can be shown that there are six independent currents. Although this is harder to see in the circuit in Figure 22 than in the single phase case, there are only six loops in the circuit. We can then write the equations using Kirchhoff's Voltage Law. See Figure 23 for the circuit loops used.

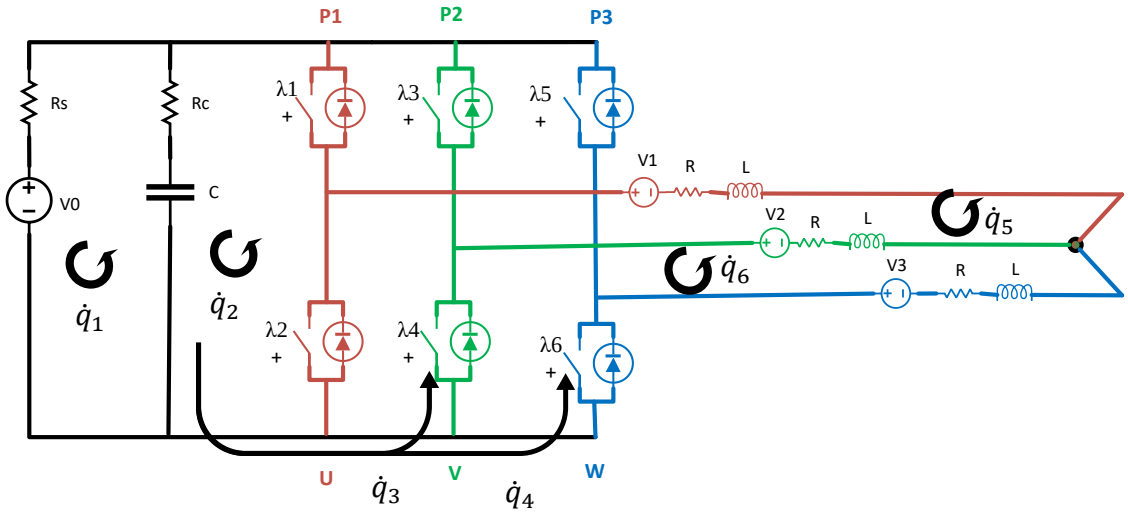

Figure 23 – Three phase circuit with currents and lambdas indicated

$$-V_0 + R_S\dot{q}_1 + R_C(\dot{q}_1 - \dot{q}_2 - \dot{q}_3 - \dot{q}_4) + \frac{1}{C}(q_1 - q_2 - q_3 - q_4) = 0 \tag{59}$$

$$\frac{1}{C}(q_1 - q_2) + R_C(\dot{q}_1 - \dot{q}_2) = \lambda_1 + \lambda_2 \tag{60}$$

$$\frac{1}{C}(q_1 - q_2 - q_3) + R_C(\dot{q}_1 - \dot{q}_2 - \dot{q}_3) = \lambda_3 + \lambda_4 \tag{61}$$

$$\frac{1}{C}(q_1 - q_3) + R_C(\dot{q}_1 - \dot{q}_3) = \lambda_5 + \lambda_6 \tag{62}$$

$$V_1 + R\dot{q}_5 + L\ddot{q}_5 - L(\ddot{q}_5 - \ddot{q}_6) - R(\dot{q}_5 - \dot{q}_6) - V_2 + \lambda_2 - \lambda_4 = 0 \tag{63}$$

$$V_2 + R(\dot{q}_6 - \dot{q}_5) + L(\ddot{q}_6 - \ddot{q}_5) - L\ddot{q}_6 - R\dot{q}_6 - V_3 + \lambda_4 - \lambda_6 = 0 \tag{64}$$

Equations 59-64 are similar to the ones found in the single phase case, Equations 37-40. These equations are non-unique; it is important to define Equations 63 and 64 so that a λ term from each phase is included. From the work done in Section 7.2.2, we

know that we only need three equations to describe the behavior of any switch state. To reach that goal, we can first make some substitutions:

$$g_C = q_1 - q_2 - q_3 - q_4 \tag{65}$$

$$i_C = \dot{q}_1 - \dot{q}_2 \tag{66}$$

$$i_{L1} = \dot{q}_5 \tag{67}$$

$$i_{L2} = \dot{q}_6 \tag{68}$$

where g_C is the charge in the capacitor, i_C is the current through the capacitor, i_{L1} is the current through the first phase, and i_{L2} is the current through the third phase.

To reduce from six equations to three, we can use the λ terms to substitute Equations 60-62 into Equations 63 and 64. Solve for λ_2 with Equation 60, λ_4 with Equation 61, and λ_6 with Equation 62. This results in

$$V_0 + R_S i_C + R_C(i_C - \dot{q}_3 - \dot{q}_4) + \frac{1}{C}(g_C) = 0 \tag{69}$$

$$V_2 + Ri_{L1} + L\frac{di_{L1}}{dt} - L\left(\frac{di_{L1}}{dt} - \frac{di_{2L}}{dt}\right) - R(i_{L1} - i_{L2}) - V_1 + \lambda_2 - \lambda_4 = 0 \tag{70}$$

$$V_3 - R(i_{L1} - i_{L2}) - L\left(\frac{di_{L1}}{dt} - \frac{di_{2L}}{dt}\right) - L\frac{di_{2L}}{dt} - Ri_{L2} - V_2 + \lambda_4 - \lambda_6 = 0 \tag{71}$$

Similar to the single phase case, we can recognize that the $\dot{q}_3$ and $\dot{q}_4$ in Equation 69 are just i_{L1} and i_{L2}, respectively. This can be seen by inspecting the circuit. We also need to recognize where the choice functions need to go. This is based on which terms

create the connection between the load side of the circuit and the battery side, as well as connecting the two loops of the load circuits. The choice functions are more complicated in the single phase case, as there are three sets of switches, and they can interact. The equations with the choice functions are given in the DAE form:

$$i_C = Sgn(P1,P2)\frac{R_C}{(R_C+R_S)}i_{L1} + Sgn(P2,P3)\frac{R_C}{(R_C+R_S)}i_{L2} - \frac{1}{C(R_C+R_S)}g_C + \frac{1}{(R_C+R_S)}V_0 \tag{72}$$

$$\frac{di_{L1}}{dt} = -\frac{3R+Sgn2(P1,P2,P3)R_C}{3L}i_{L1} + Sgn(P2,P3)\frac{R_C}{3L}i_{L2} + Sgn3(P1,P2,P3)\left(\frac{R_C}{3L}i_C + \frac{1}{3LC}g_C\right) + \frac{1}{3L}(2V_1 - V_2 - V_3) \tag{73}$$

$$\frac{di_{L2}}{dt} = Sgn(P1,P2)\frac{R_C}{3L}i_{L1} - \frac{3R+Sgn2(P1,P2,P3)R_C}{3L}i_{L2} + Sgn3(P1,P2,P3)\left(\frac{R_C}{3L}i_C + \frac{1}{3LC}g_C\right) + \frac{1}{3L}(-V_1 - V_2 + 2V_3) \tag{74}$$

Equation 72 is similar to Equation 50 in the single phase version, and the choice function is the same as defined in Equation 49. This is also used when the load current is in the equation for the opposite one, i.e. the i_{L2} term in the $\frac{di_{L1}}{dt}$ equation. However, in the

equations for the load currents, the choices for the DC-Link terms and the same load current terms depend on all three phase legs. These functions are defined as:

$$Sgn2(P1,P2,P3) = \begin{Bmatrix} 2 & if\ P1 = P2\ \neq\ P3 \\ 1 & if\ P1 = P3\ \neq\ P2\ or\ P1 \neq P2 =\ P3 \\ 0 & if\ P1 = P2 = P3 \end{Bmatrix} \tag{75}$$

$$Sgn3(P1,P2,P3) = \begin{Bmatrix} 2 & if\ P1 < P2\ and\ P1 < P3 \\ 1 & if\ P1 < P2\ or\ P1 < P3 \\ 0 & if\ P1 = P2 = P3 \\ -1 & if\ P1 > P2\ or\ P1 > P3 \\ -2 & if\ P1 > P2\ and\ P1 > P3 \end{Bmatrix} \tag{76}$$

Instead of just positive, negative, or zero, there is now the possibility that the term will appear twice. This is due to the nature of the 3 phases and how we have defined our two independent currents. These functions can be defined by examining the circuit and referring to the equations found for the manually switched method. The first choice function, which corresponds to the opposite load current showing up in the other equation, only has positive values because the two load currents always have the same relative sign. The direction of the relationship between the load side and the DC-Link side can change, and the second choice function has negative and positive results.

7.3.2.1 Solving the Equations

With only one set of equations to discretize, the solution is much easier than in the manually switched method. The solution process is the same as that for the single phase, only with one additional equation to account for the second independent load current.

$$i_C^A = \left(V_0 + Sgn(P1,P2)R_S i_{L1}^A + Sgn(P2,P3)R_S i_{L2}^A - \frac{1}{C} g_C^A\right)\left(\frac{1}{R_C + R_S}\right) \quad (77)$$

$$g_C^M = g_C^A + \frac{\Delta t}{2}(i_C^A - Sgn(P1,P2) i_{L1}^A - Sgn(P2,P3) i_{L2}^A) \quad (78)$$

$$i_{L1}^E = \frac{\Delta t}{3L}\Big(-3R + Sgn2(P1,P2,P3)R_C \frac{dt}{3L} i_{L1}^A + Sgn(P2,P3)R_C i_{L2}^A + Sgn3(P1,P2,P3)\left(R_C i_C^A + \frac{1}{C} g_C^M\right) + 2V_1 - V_2 - V_3\Big) \quad (79)$$

$$i_{L2}^E = \frac{\Delta t}{3L}\Big(Sgn(P1,P2)R_C i_{L1}^A - 3R + Sgn2(P1,P2,P3)R_C i_{L2}^A + Sgn3(P1,P2,P3)\left(R_C i_C^A + \frac{1}{C} g_C^M\right) + -V_1 - V_2 + 2V_3\Big) \quad (80)$$

$$g_C^E = \left(g_C^M + \frac{\Delta t}{2}\left(V_0 - \left(1 - \frac{R_C}{R_C + R_S}\right)(Sgn(P1,P2)R_S i_{L1}^E + Sgn(P1,P2)R_S i_{L2}^E)\right)\right)\left(1 + \frac{\Delta t}{2C(R_C + R_S)}\right)^{-1} \quad (81)$$

$$i_C^E = \left(V_0 + Sgn(P1,P2)R_S i_{L1}^E + Sgn(P2,P3)R_S i_{L2}^E - \frac{1}{C} g_C^E\right)\left(\frac{1}{R_C + R_S}\right). \quad (82)$$

These equations are from the same scheme described in Section 7.3.1.1. They are solved in a loop where P1, P2, or P3 could change at any time, therefore eliminating the necessity to keep track of initial conditions when the switch state changes. For the first step of the solution, the initial values of g_C, i_{L1}, and i_{L2} must be defined as they were in the single phase example. The internal trigger events, zero crossing of either i_{L1}, i_{L2}, or

$i_{L1} + i_{L2}$, are not covered by the choice functions because they will not occur as we have ignored the open circuit case.

7.3.3 Zero Crossing

Up to this point, we have ignored the open circuit case in both the single phase and three phase formulations, and therefore the currents i_L, or i_{L1}, i_{L2}, or $i_{L1} + i_{L2}$, have been allowed to change signs with no change in equation. If one tries to add this case (by defining P1, P2, or P3 as 0), the solution is not accurate anymore. One step to incorporating this case into the model is by creating a translation from that 0 definition of P1, P2, or P3 to either a 1 or -1 definition. In the single phase case, P1 is equal to the sign of i_L and P2 is equal to the opposite of the sign of i_L. In the three phase case, the value of P* is simply the sign of the current in that phase leg. This new definition satisfies the solution for the first part of operation, before the zero crossing occurs.

After any one of the load currents change sign, the equations necessarily must change. To implement zero crossing in the single phase model, conditions have to be added to the Sgn function that change the result to zero when the sign of i_L changes, because the zero crossing results in the two circuit loops being separated, i.e. the red or orange loops in Figure 20. One choice function also must be added to the $\frac{di_L}{dt}$ equation to zero out the result in the case when the loops are already separated initially.

The zero crossing implementation in three phase is harder to achieve. The changes that can occur are more numerous, especially as there are three different currents (because the two independent load currents combine for the third dependent current) that

can change sign, and the behavior of the circuit could change multiple times if more than one current crosses zero. By trying to examine every possible circuit, the choices become too numerous and we run into similar problems as when trying to develop the equations as in the manually switched case. The correct choice functions could be found with more advanced mathematical descriptions.

7.3.4 Results

The non-smooth DAE approach is implemented in Matlab, with the results compared to PLECS shown below. Figure 25 and Figure 24 show the results for the single phase case. The external switch commands are shown as well; it can be seen in the graph of load current that the solver also responds to the internal trigger when the load current crosses zero. In the graph of capacitor voltage, some error can be observed between PLECS and Matlab. However, this error is not seen in the three phase results in Figure 26 and Figure 27. The three phase is the more important, because this is the actual model of the drive and the model that will be used in a controller or optimization.

For a string of just 5 commands, the time to perform the solution is greatly decreased for Matlab by a factor of about 30-40, shown in Table 9. Not only is the factor of the decrease notable, the graphs show that Matlab is calculating more points than PLECS and still calculating faster. With a variable time step, the speed of the Matlab results could probably be increased more.

Table 9 – Comparison of calculation times for single and three phase models using PLECS and Matlab

Single Phase Plecs	0.178 s	Three Phase Plecs	0.171 s

Single Phase Matlab	0.00402 s	Three Phase Matlab	0.00580 s

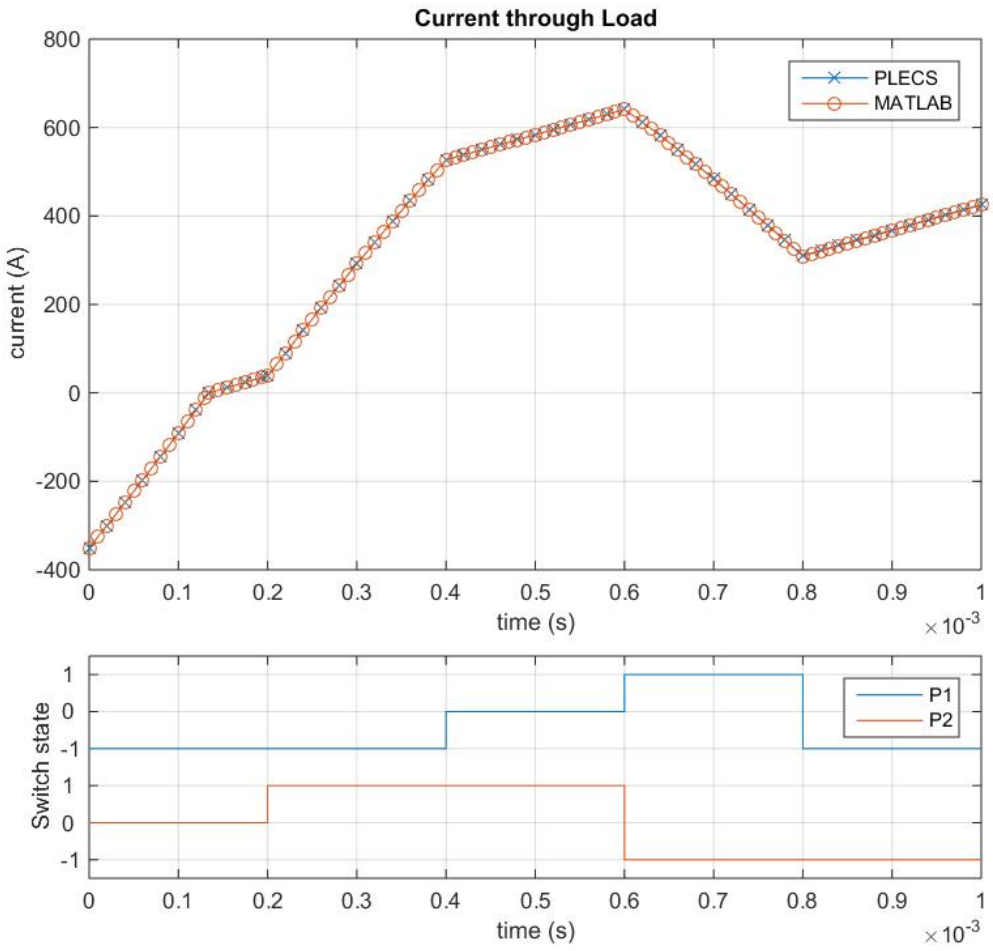

Figure 24 – Results comparing model to PLECS, single phase, load current, including switch commands

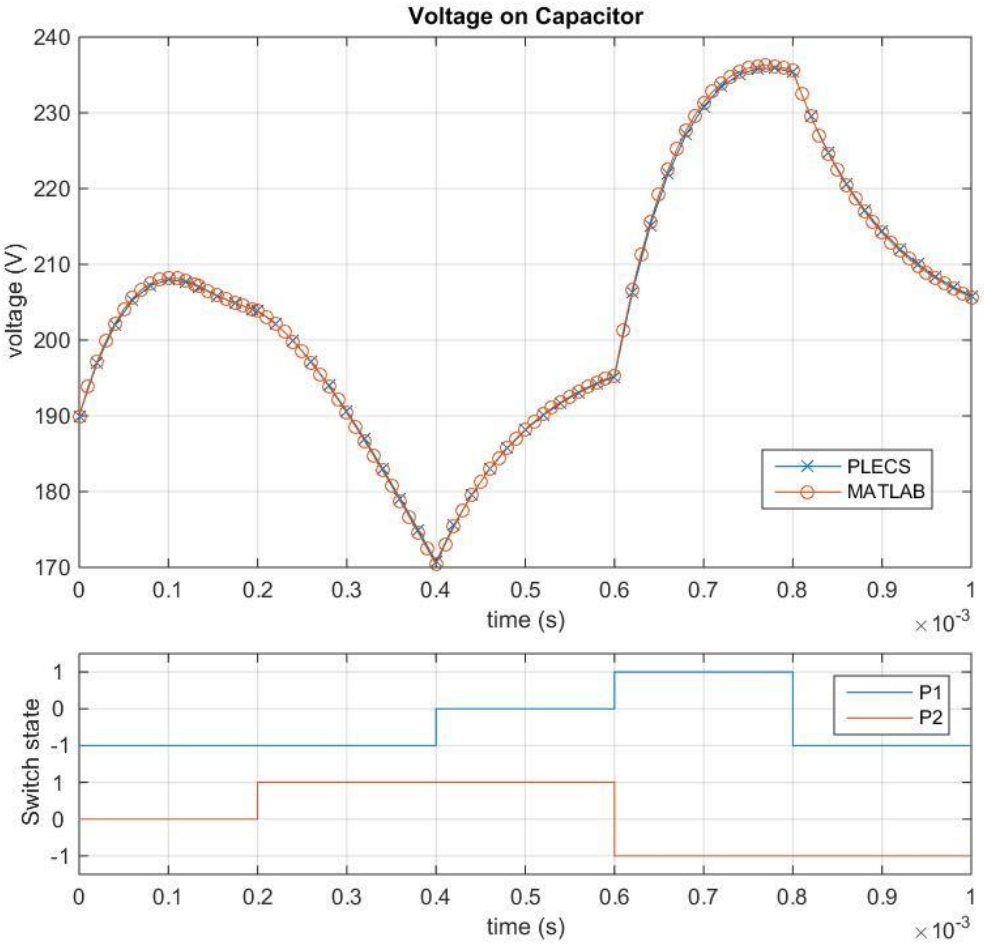

Figure 25 – Results comparing model to PLECS, single phase, capacitor voltage, including switch commands

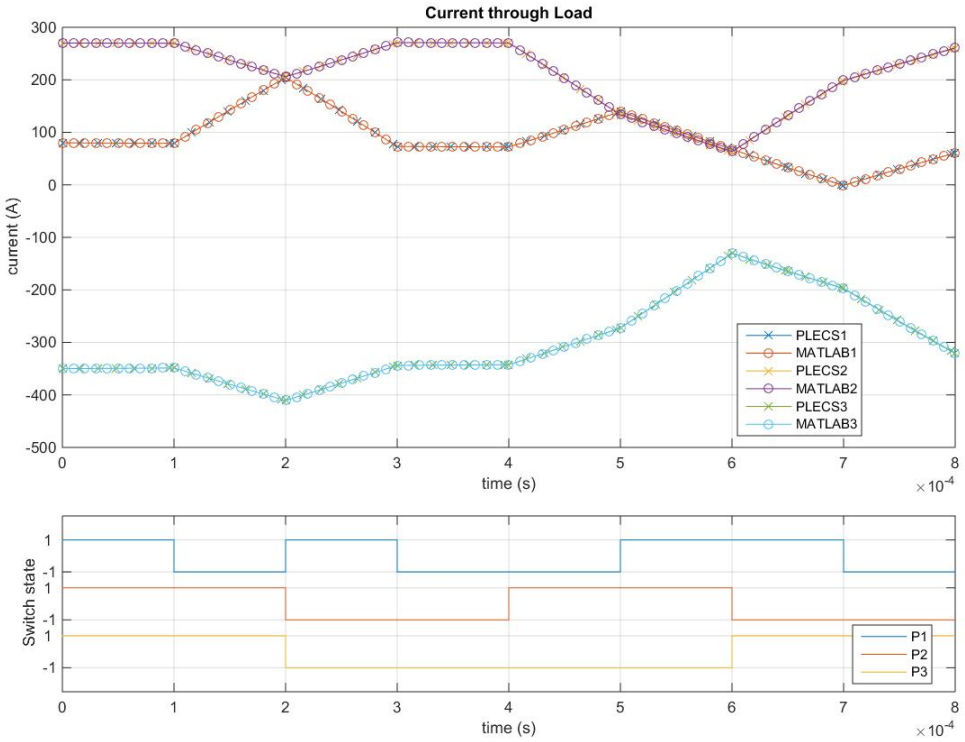

Figure 26 – Results comparing model to PLECS, three phase, load current, including switch commands

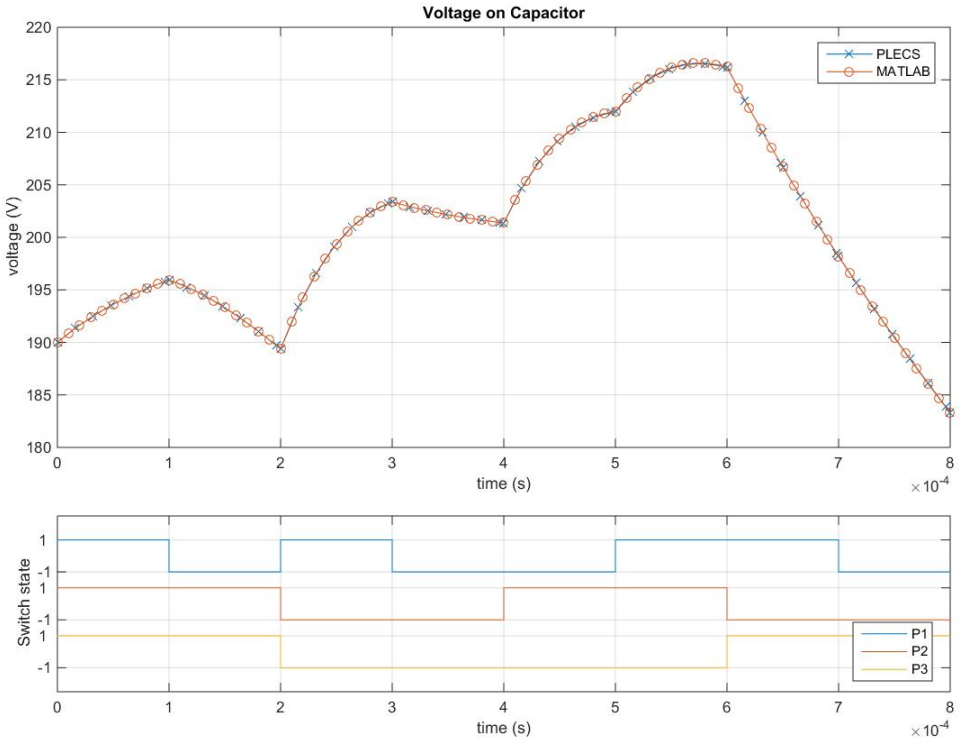

Figure 27 – Results comparing model to PLECS, three phase, capacitor voltage, including switch commands

7.4 Comparison of the Models

The development of both models has been described and the results of the non-smooth DAE model have been presented. This approach is superior in both method and result to the manually switched approach. Although insights of the solution of the manually switched method approach were used to find the final equations of the non-smooth DAE model, the mathematical basis of the second approach provides a path to easier development for larger systems. From an interpretation point of view, the result is superior because it shows a connectedness between the equations that the first does not.

In the non-smooth model, the continuity between phases is accounted for when the switches change state. It is also much easier to solve in Matlab.

The non-smooth implementation performs better, because of when the choice of equations is made. In the manually switched method, before the equations can be solved, the program must go through a series of 'if' statements to determine which set of equations to use, then assign the A, B, and C phases correctly. It must also calculate or assign six initial conditions before beginning the solution. This process occurs each time the switch state changes. In the non-smooth model, the choice is simply a short function added to the script that solves the equations and incorporated into the single set of equations that describe the model. Although the program checks the choice at every time step of the discretization, and the switch state does not change at that rate, it is simply within the normal operation when the switch state does change. There is no difference in operation when the switch state changes, meaning the (only) three initial conditions do not have to be recalculated or switched around, and are simply from the final value of the previous iteration.

The implementation of the open circuit case is also handled more elegantly in the non-smooth model. In the separate differential equations model, which uses a Matlab solver, first the original case is calculated, then that solution is checked for a change in sign of the current, and the solution must be recalculated from this point. This is a lot of unnecessary extra calculation and backtracking. In the non-smooth case, when the zero crossing is detected, the equations can change immediately, and if back tracking is required to find the point of change more accurately, less of the solution needs to be

recalculated. Additionally, the initial conditions for the new set of equations are simply taken from the last computed time step.

Finally, the non-smooth DAE method was much easier to implement in Matlab, to the point that the switched method was not fully developed in Matlab. We showed in Section 7.3.4 that the DAE method is faster than PLECS. In the inaccurate simulations using the switched method, the calculation times were slower or equal to the time for PLECS.

7.5 Future Use and Extensions of the Model

Further work on developing the system model could go in two directions: expanding the capabilities of the solver or incorporating representations of additional physical elements. The first type of expansion includes different treatment of the AC voltage source and adding the open circuit case as a dead time option. Additional or adjusted physical parameters include the definition of the inductance in the load, adding a delta connected circuit to the model, or adding other elements.

One aspect of expanding the solver further is how the voltage on the load (the back emf) is handled in the calculations. Because of the small enough discretization of time, this voltage is assumed as constant on each time step, and the sine wave is realized by the discretization. By changing this to a continuous function where the voltage can vary over the time step, the solution could become independent of the size of the discretization, meaning a larger time step would not give any reduction in accuracy due to the estimation for the voltage value.

The open circuit case is not currently implemented in the solution for the three phase circuit. Usually, the state of the switches is defined as high or low and the zero state, or open circuit case, is only used as the dead time tool. If the program were expanded, it could include an option of dead time of a certain length, and the only regular input to the switches would remain the high or low choice. Dead time is the small window of time between when one switch of a phase opens and the other closes, and helps prevent a short circuit in real life operation.

The value of the inductance on the load is calculated based on the operating point and this value is assumed constant. However, it can change based on the current amplitude. The motor description from finite element analysis may be better described by an inductivity which depends on current.

The work done above is for a wye-connected circuit, but the equations for a delta-connected circuit would be different (See Figure 12). The model is a drive with just a single three phase control; this could be expanded to drives with redundant control, or an Nx3 phase machine, where N is 2 or higher.

The model, as described in section above, is a simplified one. Many physical aspects have been removed or simplified. The model could be easily expanded by adding elements in, such as the inductor in the DC-link. There is always a trade off in detail in models with the computation time it takes to solve them. It may be that adding more elements does not increase the accuracy of the model and adds to the computation time. The model as it has been developed is a good representation of the system and the suggestion of adding more parts is only to see if they help in the optimization process.

CHAPTER 8. MODEL PREDICTIVE CONTROL

8.1 Motivation

Having a fast and sufficiently accurate model description at one's disposal means it is now possible to use the model to find a sequence of switching events that will lead to an optimized performance of the drive. The optimization can be performed with respect to various KPIs, such as torque ripple, number of switching events, NVH, DC ripple, etc. The control of the drive is through the switching of the gates in the B6-bridge and the performance is based on the change of current resulting from the switching events. The goal of the optimization would be to balance a minimization of the number of switching events with optimal performance from the other various KPIs, focusing initially on torque performance. One optimization strategy is a Monte Carlo approach where many different sequences of switching events could be tested and the results compared. These sequences could be created randomly, both in the commanded switch combination and in the time point that the switches open or close, or there could be sophisticated design intent behind the choices of sequences. Either way, a Monte Carlo approach requires tens of thousands of combinations or more to be tested and compared to find the range of the best ones. The drawback of having to apply a large computational effort is furthered by the fact it might be hard to classify and understand why those optimal sequences performed better. (See [11] for information about Monte Carlo optimization.)

A different approach is to use Model Predictive Control (MPC), an optimization technique which incorporates control, as implied by the name. While it is often used online during the operation of the system being controlled, we propose to use explicit

MPC. Explicit here simply means a superior sequence of switching events is found offline, before the control is actually implemented in a simulation or actual system. MPC is similar to Monte Carlo, in that it checks many different possibilities. However, the time scale is smaller and a cost function is used to continually check for superior performance. The cost function in MPC allows for guidance of the 'superior performance,' and the KPIs can be selected for and balanced. Although this chapter proposes using explicit MPC, this can easily be adapted to an online process. In this way, the MPC approach has an advantage over the Monte Carlo one because it has a path to control built in.

The idea to use MPC in this manner has been done before. A very similar example can be seen in [12]. This thesis uses a simpler model of the drive, but shows that the MPC approach is effective in finding superior performance. That work was based on research by Geyer, as seen in [13] and [14].

This chapter discusses how the MPC approach can be applied in this case. The approach is based on the model developed in Chapter 7, but it has not been applied yet. The MPC controller is built and has been verified, but significant results have not been found. Since promising results have been found in the references listed above, the MPC optimization should produce applicable results in this case as well. This chapter offers the structure of the optimization, potential issues, predicted outcomes, and future use.

8.2 Structure of Model Predictive Control

The purpose of MPC is to turn an infinite optimization problem into a series of finite ones. This is done by predicting a series of short time horizons rather than the full

period of operation. To run an MPC optimization, one needs the measurements of the states and a model of the system. The optimization algorithm always works on a fixed time horizon (usually a few time steps) and mainly consists of two stages. Figure 28 below illustrates the procedure. First, the optimization problem is solved for the prediction horizon, starting from the measured states. The results of the system are predicted by the equations, using every possible switch state. The cost function for each sequence of switch states is calculated and the minimum is found. In the second stage, the calculated optimal control is implemented for a shorter time horizon, usually only one step ahead. In the next iteration, the time horizon is shifted forward by the control horizon and the two stages are performed again. In online operation, this process continues as long as the system is running. In a simulation, or our explicit version, a total time of operation would be defined.

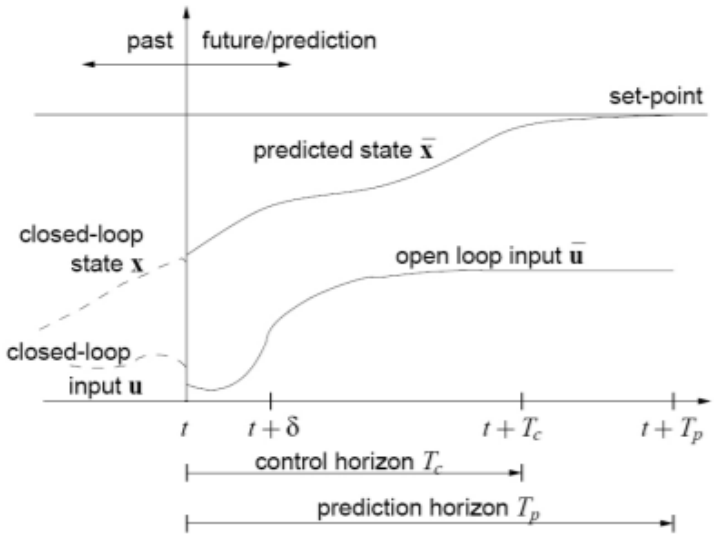

Figure 28 – Diagram illustrating time horizons for model predictive control

The prediction horizon requires a string of possible switch states; these strings are built from the switch states described in Table 8. There are eight switch states in the table; for three time steps there are 8^3=512 combinations and for four time steps, 4096. Depending on the size of the time step and the total time to be modeled, this process will then have to be repeated a few thousand times. Checking every possibility in this way is called full enumeration. We do not want to look too much farther in the future, as we will not get much more information. Five time steps is the reasonable limit for the time horizon, which would mean 32,768 possible combinations. At this point, we are approaching comparable numbers to a Monte Carlo simulation. However, there is an advantage in understanding the minimization more clearly and having the link to the control. Additionally, each trial in the MPC approach is much shorter than Monte Carlo, as it only calculates the short fixed time horizon instead of the full time period under investigation.

One must also choose the length of the time step used in the optimization. A smaller time step is desired to ensure too much accuracy is not lost through the discretization. However, if the time step is chosen to be too small, the number of iterations needed to produce an optimized control sequence for a full simulation increase. We are proposing to use explicit MPC, so we do not have to worry about computation time while online. In an online case, the time step would have to be long enough that the next control command could be calculated before it is needed. The suggested time step is 25 μs. This was used in [12] and is small enough to ensure accuracy in the discretization.

8.2.1 The Cost Function

In any optimization problem, a cost function needs to be defined to represent the terms to be minimized. In a continuous time problem, the cost function is an integral and in a discrete time problem, a sum. In an MPC formulation, the bounds of the cost function are finite. The cost function has terms relating to the input and output. Minimization is then achieved through a balance between the two (i.e. there may be a line of constant minimum where the output and input can vary in their contribution to the minimization).

The cost function proposed for this MPC problem is designed to minimize the output that is the torque ripple, while the input is represented by the number of switching events.

$$J = \sum_{i=k}^{k+N} (\tau(i) - \tau_d)^2 + \beta S(i), \qquad \beta > 0 \tag{83}$$

The first term of Equation 83 is for minimizing the torque ripple, by comparing the calculated torque to the desired torque. Torque ripple is not desired as it reduces the accuracy in the torque output of the motor. The second term penalizes switching events. Switching is accompanied by switching losses. If switching occurs less often or at better times (e.g. at a smaller value of the switched current), the losses are reduced. Therefore, the thermal load on the switches is reduced and the efficiency of the inverter is higher. The positive scaling factor β is used to balance the effect on the cost from the two terms. Depending on the value of the torque ripple, both terms need to be around the same magnitude so that they both have an approximately equal effect on the cost, but β can also be used to weight one term more than the other.

8.2.2 Calculation of Torque

The cost function uses torque as the output, but the output presented in Chapter 7 was the current. This means there needs to be some relationship between current and torque. These two quantities can be related analytically, but we propose using a numerical relationship instead. The related torque values are found by inputting the output currents into look-up tables of appropriate torque values. These tables were created using finite element analysis of the drive.

Using the look-up tables instead of an analytical relationship has advantages and disadvantages. One drawback is the data from the FEA must be for a drive with the correct properties, such as size or operating temperature. This means that it may be necessary for many analyses to be run to optimize different drives. However, any equation between torque and current would most likely be empirically based regardless. The relationship between the two is not easily represented mathematically, as there can be saturation and nonlinear effects. This means that these analyses must be done in either case, and once the information has been gathered once, it is usable into the future. The look-up tables also require some interpolation (and potentially extrapolation, if the output current values are outside of those analyzed initially). This can lead to inaccuracies, but with a spline inter-/extrapolation these inaccuracies are reduced.

There are also advantages to using the look-up tables. The harmonics of the torque behavior can be easily separated, and thus can be analyzed in the optimization individually. The main harmonic can be optimized, and the rest sequentially. A harmonic can be isolated and inspected if it is producing high levels of torque ripple.

Additionally, the look-up tables of torque values had already been produced before the work on this thesis began, so they were already available for use.

8.2.3 Optimizing the Switching Sequence

As explained in Section 8.1, this MPC based optimization has not yet been performed with this model. In Chapter 7, we developed the model and in this chapter we have described how MPC could work in this case. Because similar work was done in [12] (also a thesis researched at Bosch), this is a topic that will be explored further in projects at the company. The promising results of that thesis also show the value of continuing the work of the optimization of this model.

There is not going to be one 'best' switching sequence, but many superior ones. To find these optimal sequences, it will be necessary to discover the effects of the different parameters in the cost equation on the optimization. The most obvious is the tuning of the scaling factor β, balancing reducing torque ripple and minimizing the number of switching events. The time horizon can also be adjusted, from three to five steps. In that case, we would want to discover if a longer horizon better predicts the optimal control, or if the effects of each switch state are more immediately apparent. Similarly, the time step can be changed as well to check for similar effects as the time horizon adjustments while also paying attention to the accuracy of the discretization.

8.3 Further Use and Extensions

The first extension that could be easily applied is to add terms to the cost function. Currently, the cost for a switching event is constant. As explained above, the magnitude

of the switching current at the time of the switching event changes the severity of the switching event. This term in the cost function could instead be related to the value of that current, to penalize switching events at higher currents more heavily. If a switch spends too long a time in the open state, this can cause noise. Separate terms for each switch could be added to make it advantageous in the minimization to close a switch that has been open for a certain period of time.

Other KPIs have been mentioned throughout the thesis. To add them to the cost function they need to be able to be represented through outputs or states of the model. The DC ripple is indicated by the current through the capacitor, which is easily found from the model. However, a KPI such as NVH or power losses are more complicated. The model may need to be extended to incorporate these terms into the cost equation.

The MPC scheme could be more flexible. It is currently discretized with a constant time step. As in the model, it may be advantageous to make variable time steps possible, or even a continuous case. This would mean that the switching events could occur at any time and would not be restricted to only $t + k\Delta t, k = 1,2,3 \ldots$ time points. Another element of flexibility—that would also need to be considered in the model—is the possibility of variable speed. This means the operating point of the drive would change.

As discussed, the first goal of the MPC approach is to show, using the mathematical model that we have created, that a switching sequence created through this optimization results in superior performance as compared to space vector pulse width modulation or other traditional control. Model predictive control is usually used online in systems, where the model predicts behavior and then the optimized control is applied to

the actual system. The actual output is then used as the initial state in the next prediction horizon. In this case, we are using the model to both predict behavior and process the control. This means that the results from using the optimal switching sequence should be exactly as they were during the optimization process. This can all be performed offline and the switching sequence can first be applied to other models of the electric drive, such as those mentioned in some section. After that, the switching sequence can be applied to a physical drive to see how the theoretical results compare to the predicted. The ultimate goal would be, after showing that the optimized switching sequence does result in superior performance, to install an online MPC controller on the electric drive. The work done in [12] was also leading to this goal.

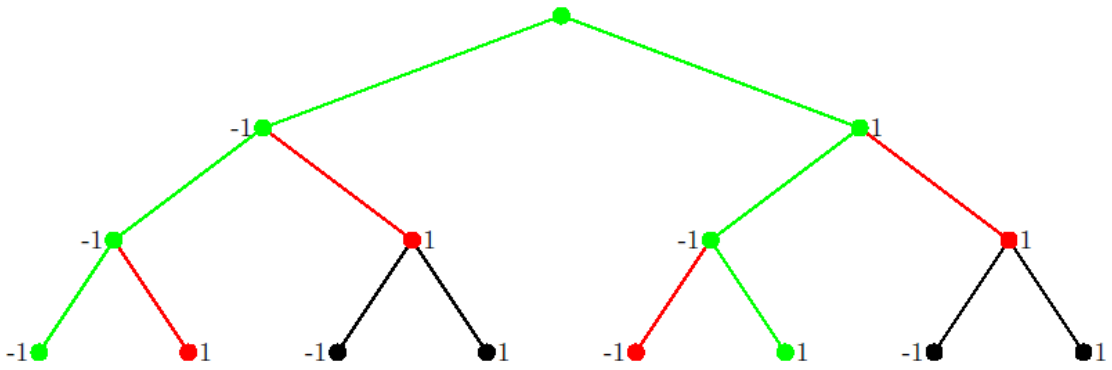

Figure 29 – Search tree. Green nodes are inside the limit of the cost function, red nodes are outside, and black nodes are unexplored [12]

Whether offline or online, it is always desirable to decrease the time needed to run the calculations. The full enumeration process for finding the minimum cost function suggested in this thesis is the longest possible process. Although finding the optimal control at each time step takes only a few seconds, when this is done for each time step, it takes on the order of tens of minutes. To be able to be more efficient in finding optimal

sequences offline, this should be sped up. If an online process is ever desired, then the calculation at each time step should be faster as well. One way to do this is to reduce the number of search paths and avoid the 'worst case complexity' of having to check each and every option. This reduction can be done by imposing a limit on the cost function and not exploring any path where the cost function goes over that limit, and was used in [12] and is illustrated in Figure 29. Although this would mean evaluating the cost function at each step and not just at the end of the time horizon, it would also potentially eliminate many search paths.

Although the optimization procedure has not yet been implemented, the model described in this thesis and the process detailed in this chapter should be able to be used together to find 'better' switching sequences.

CHAPTER 9. CONCLUSION

In this Part Two of thesis, we have discussed the electric drive, developed a mathematical model, and presented an optimization scheme. This thesis describes a step in the goal towards a redesigned and optimized electric drive. Electric drives are very important in a world that is relying more and more on renewable sources of energy. Increasing their efficiency and improving other aspects of behavior will only increase their usage.

The creation of a mathematical model proved to be a complicated process, but the insights gained and the model itself were worth the work. The development of the model led to two ways of thinking of the system. It can either be a set of different circuits that can be separately analyzed or it can be represented by one set of equations with terms that are 'turned on' or 'turned off.' Although the second method, the non-smooth DAE model, proved more useful and simpler to model, the first manually switched model was helpful in developing the second. All the steps in this process should help in the continuing work. The non-smooth model can be used in the optimization of the switching sequence but is also a good starting place for expanding the model and having the drive more fully described mathematically instead of in other modeling software, such as PLECS or Simulink. The initial examination of all possible circuits can also help others to better understand what is happening when the switches change state.

Even though this thesis does not present results for the optimization of the switching sequence, a cost function has been developed and the process fully described. MPC has been applied in other similar cases, showing that the suggestions of this thesis

have a basis in past work. The MPC optimization can be used, tuned, and improved based on details provided in this thesis.

This thesis presents many opportunities for this work to be taken further. As has been said, this work is only one part in a larger holistic design process. The creation of the model is a large step forward and the approach of the MPC optimization has been shown to work in other cases. Putting the two together should lead to an optimized drive that is capable of meeting targets for various KPIs.

REFERENCES

[1] Sunberg, Z. N., Miller, N. R. and Rogers, J. D., "A Real-Time Expert Control System for Helicopter Autorotation," *Journal of the American Helicopter Society*, Bd. 60, 2015.

[2] Johnson, W., Helicopter Theory, New York: Dover Publications, Inc., 1994.

[3] Leishman, J. G., Principles of Helicopter Aerodynamics, New York: Cambridge University Press, 2000.

[4] Seddon, J. and Newman, S., Basic Helicopter Aerodynamics, Chichester, UK: Wiley, 2011.

[5] Talbot, P. D., Tinling, B. E., Decker, W. A. and Chen, R. T. N., "A Mathematical Model of a Single Main Rotor Helicopter for Piloted Simulation," TM-84281, NASA, 1982.

[6] Chen, R. T. N., "A Simplified Rotor System Mathematical Model for Piloted Flight Dynamics Simulation," TM-78575, NASA, 1979.

[7] Chen, R. T. N., "Effects of Primary Rotor Parameters on Flapping Dynamics,"

Technical Paper TP-1431, NASA Ames Rsearch Center, 1980.

[8] Mirchevski, S., “Energy Efficiency in Electric Drives,” *Electronics,* Bd. 16, pp. 46-9, 2012.

[9] Veltman, A., Pulle, D. W. and DeDoncker, R. W., Fundamentals of Electric Drives, Heidelberg: Springer, 2007.

[10] Glocker, C., “Models of non-smooth switches in electrical systems,” *Internation Journal of Circuit Theory and Applications,* Bd. 33, pp. 205-234, 2005.

[11] Robert, C. P. and Casella, G., “Monte Carlo Optimization,” in *Introducing Monte Carlo Methods with R*, New York, Springer, 2010, pp. 125-65.

[12] Holley, J., “Fault-Tolerant Model Predictive Control of Redundant Permanent Magnet Synchronous Machines,” Master’s Thesis, University of Freiburg, 2016.

[13] Geyer, T. and Quevedo, D. E., “Multistep Direct Method Predictive Control for Power Electronics-Part 1: Algorithm,” *Proc. IEEE Energy Conversion Congress and Exposition,* pp. 1154-61, 2013.

[14] Geyer, T., “Model Predictive Control in Power Electronics: An Introduction,” ABB Corporate Reserach, Switzerland, 2016.