INDEPENDENT OPERATION OF PARALLEL THREE-PHASE CONVERTERS FOR MOTOR DRIVE APPLICATIONS

by

W. Daniel Fingas

A thesis submitted in conformity with the requirements for the degree of Masters of Applied Science Graduate Department of Electrical and Computer Engineering University of Toronto

Copyright © 2009 by W. Daniel Fingas
Abstract

Independent Operation of Parallel Three-Phase Converters for Motor Drive Applications

W. Daniel Fingas
Masters of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto
2009

A motor drive consisting of two parallel voltage-sourced converters was developed and implemented. A parallel converter arrangement allows the system to be constructed in a modular fashion to gain economies of scale and redundancy. The converters are connected to common ac- and dc-buses without isolation and are controlled without inter-converter communication or a master/slave arrangement. The system was simulated and the results validated against an experimental setup. Both steady-state and dynamic load sharing were achieved through the use of drooped PI speed regulators. PI controllers were used to regulate the quadrature currents provided by each converter. Circulating 0-sequence current was regulated using P controllers. A linearized state-space model of the system was developed and an eigenvalue analysis was performed, showing system stability. Speed steps in simulation and in the laboratory demonstrated good response. The loss of one converter’s gating was emulated. The system continued to operate, showing an advantage of system redundancy.
Acknowledgements

I would like to thank Professor Peter Lehn for his guidance and patience even through late nights in the laboratory! His wisdom and approachability were very much appreciated.

Professor Francis Dawson and Jack Goldstein were also always willing to offer assistance.

I would also like to acknowledge and express thanks for the financial support of the people of Canada, provided through the Natural Sciences and Engineering Research Council of Canada (NSERC) in the form of a postgraduate scholarship for one year during the course of my research.

I also thank God for blessing me with this opportunity to learn in such a supportive and challenging environment.
## Contents

1 Introduction .............................................. 1
  1.1 Motivation ........................................... 1
  1.2 Project Objectives ................................... 2
  1.3 Literature Review .................................... 2
    1.3.1 Load Sharing ...................................... 3
    1.3.2 0-Sequence Circulating Current .................... 4
  1.4 Overview of Thesis ................................... 5

2 System Description ...................................... 6
  2.1 Overview ............................................. 6
  2.2 Permanent-Magnet Synchronous Machine .................. 7
    2.2.1 Machine Description ............................... 7
    2.2.2 Machine Loading ................................... 9
  2.3 Converters and Control Hardware ....................... 10
    2.3.1 Converter and $C_{dc}$ ............................. 10
    2.3.2 RT-Linux Controllers ............................. 10
    2.3.3 Anti-Aliasing Filters .............................. 10
  2.4 dc Side .............................................. 11
    2.4.1 dc Source ......................................... 11
    2.4.2 dc Chokes $L_{dc}$ ................................. 12
2.5 ac Filters .......................................................... 12
  2.5.1 Filter Description ............................................. 12
  2.5.2 Implementation Details ................................. 13
  2.5.3 Switching Ripple in the Two-VSC Configuration .... 14
  2.5.4 Characterization of 0-Sequence Switching Ripple .... 16
  2.6 Chapter Conclusion ............................................ 18

3 System Model ..................................................... 22
  3.1 Overview ...................................................... 22
  3.2 Machine Model ................................................ 22
    3.2.1 Rotating $qd0$ Reference Frame ...................... 23
    3.2.2 Machine Equations ..................................... 24
    3.2.3 Input-Output Block for Machine Equations .......... 26
  3.3 Averaged Converter Circuit ................................. 27
    3.3.1 $qd$ Averaged Model .................................. 28
    3.3.2 0-Sequence Averaged Model .......................... 29
  3.4 $qd$ Model of the Parallel Converter Arrangement ...... 30
    3.4.1 $qd$ State-Space Derivation ............................ 30
  3.5 0-Sequence Model of the Parallel Converter Arrangement .................. 34
  3.6 Complete Two-VSC System .................................... 35
    3.6.1 Control Considerations for the Two-Converter System 35

4 Controller Development ....................................... 37
  4.1 Control Methodology ........................................ 37
  4.2 0-Sequence Control ........................................ 38
    4.2.1 Proposed 0-Sequence Controller ..................... 39
  4.3 Machine Control ............................................. 40
4.3.1 Conventional Single Converter Field-Oriented Control 40
4.3.2 Extension to the Two-Converter Case 41
4.4 Closed-Loop Two-Converter System Model 43
  4.4.1 Speed Control Loop 44
  4.4.2 $q$-axis Control Loop 44
  4.4.3 $d$-axis Control Loop 45
  4.4.4 Closed-Loop Model 46
4.5 Stability Analysis 51
  4.5.1 Common- and Differential-Mode Currents 51
  4.5.2 Lunze Transform 52
  4.5.3 Common- and Differential-Mode System Model 54
  4.5.4 Eigenvalue Analysis 55

5 Results 59
  5.1 Overview 59
  5.2 Machine Model Validation 59
    5.2.1 Open-Loop Response 59
    5.2.2 Closed-Loop Response 63
  5.3 System Response to Speed Steps 65
    5.3.1 Low Speed Test 65
    5.3.2 High Speed Test 71
  5.4 Loss of Converter 76

6 Conclusions 79

A Motor Details 84

B Implementation Issues 89
  B.1 Common-Mode Signal Filters 89
B.2 Simulation Stability ................................. 90
B.3 Time-Step Size and Solver Choice ................. 90

C RT-Linux Control Code .............................. 91

D Description of RT-Linux Interface ................. 108


List of Tables

2.1 System Parameters ........................................... 7
2.2 Motor Parameters ........................................... 9
4.1 Controller gains and operating points ....................... 56
A.1 10-pin encoder pinout .................................... 86
List of Figures

2.1 System Schematic ................................................. 8
2.2 Screenshot of the control interface. .......................... 11
2.3 One of the two common-mode chokes constructed for this project. . . . . 14
2.4 Closed-loop path for the 0-sequence switching ripple current. ........ 16
2.5 Explanation of duty cycles for a requested voltage of 0.10 p.u.: (a) Converter 1’s voltage references, highlighting the time period examined in (b)-(d). (b) Requested voltages (normalized against the dc supply voltage $\frac{v_{tabc}}{v_{DC/2}}$) and the triangular carrier signal (c) Duty cycles resulting from the comparison and (d) Instantaneous $v_{qdb}$ for converter 1. ................. 19
2.6 The (a) minimum and (b) maximum 0-sequence driving voltage when two asynchronous converters are present, corresponding to the region around in-phase and 180° out-of-phase switching, respectively. As in Fig. 2.5, the requested voltage was 0.10 p.u. .............................................. 20
2.7 Experimentally observed 0-sequence ripple current, 6.7 A/V (i.e. 0.67 A/div). (a) Minimum ripple, corresponding to the converters near in-phase and (b) maximum ripple, when the converters are switching 180° out-of-phase. ......................................................... 21

3.1 Alignment of the Reference-Frame with respect to the Rotor Magnet . 24
3.2 The $qd0$ motor block ............................................... 26
3.3 The separate (a) $qd$ and (b) 0-sequence motor input/output blocks. . . . 27
3.4 A single converter with its input and output filters .......................... 28
3.5 Averaged $qd$ model for a single converter ................................. 28
3.6 Averaged $qd$ model for the two-converter system ......................... 29
3.7 Averaged model of the 0-sequence circuit ................................... 29

4.1 0-sequence feedback control model ........................................... 39
4.2 Conventional single-converter field-oriented control loop ............... 41
4.3 Field-oriented control loop extended for the parallel converter system . 43
4.4 Overall controller flow diagram .................................................. 47
4.5 Common-Mode Eigenvalues ......................................................... 57
4.6 Differential-Mode Eigenvalues ..................................................... 58

5.1 (a) Simulation and (b) experimental results for $v_q$ step from 15 to 20 V.
$\omega_e$ is [20 rad/s/div] and $i_q$ and $i_d$ are [5 A/div]. Experimental time is in ms. 61
5.2 (a) Simulation and (b) experimental results for a 5 V step in $v_d$. $\omega_e$ is 
[20 rad/s/div] and $i_q$ and $i_d$ are [10 A/div]. Experimental time is in ms. 62
5.3 (a) Simulation and (b) experimental results for $i_{qref}$ step from 0.74 to 15.94 
A. $\omega_e$ is [20 rad/s/div] and $i_q$ and $i_d$ are [5 A/div]. Experimental time is in ms. 64

5.4 Experimental: Response of VSC1 & VSC2 to a step in speed reference at 
t = 0.0 s from 62 to 124 rad/s (electrical) ........................................ 67
5.5 Experimental: Difference in Response Between VSC1 & VSC2 to a step in speed reference at t = 0.0 s from 62 to 124 rad/s (electrical) ......... 68
5.6 Simulation: Response of VSC1 & VSC2 to a step in speed reference at 
t = 0.1 s from 62 to 124 rad/s (electrical) ......................................... 69
5.7 Simulation: Difference in Response Between VSC1 & VSC2 to a step in speed reference at t = 0.1 s from 62 to 124 rad/s (electrical) .......... 70
5.8 Experimental: Response of VSC1 & VSC2 to a step in speed reference at 
\[ t = 0.0 \text{ s from 356 to 419 rad/s (electrical).} \] 72

5.9 Experimental: Difference in Response Between VSC1 & VSC2 to a step 
in speed reference at \( t = 0.0 \text{ s from 356 to 419 rad/s (electrical).} \) 73

5.10 Simulation: Response of VSC1 & VSC2 to a step in speed reference at 
\[ t = 0.1 \text{ s from 356 to 419 rad/s (electrical).} \] 74

5.11 Simulation: Difference in Response Between VSC1 & VSC2 to a step in 
speed reference at \( t = 0.1 \text{ s from 356 to 419 rad/s (electrical).} \) 75

5.12 Experimental test of the loss of gating signals for converter 2: from steady-
state, at \( t = 0.2 \text{ seconds. Shown are the speed and } qd0 \text{ currents for both} \) converters. 77

5.13 Simulation of the loss of gating signals for converter 2: from steady-state, 
at \( t = 0.2 \text{ seconds. Shown are the speed and } qd0 \text{ currents for both converters.} \) 78

A.1 Motor Information for Kollmorgen Goldline M-803-A. 88

D.1 Screenshot of the Control Interface 108
Chapter 1

Introduction

1.1 Motivation

Power electronic converters have increasingly been used in motor drive applications. These converters provide increased flexibility by allowing for low-speed, high-torque operation while still maintaining high efficiency and limiting the peak current. Conventional motor drives consist of a single controller which controls one or more three-phase dc-ac voltage-sourced converter (VSC) modules, where the number of modules required is a function of the power rating and module size.

A parallel arrangement of VSCs making use of autonomous controllers is proposed for motor drive applications. In this configuration, two or more converters will drive a single motor load. The two main advantages of this setup are modularity and redundancy. For modularity, a larger converter can be constructed from several smaller converters, each with separate control. As each controller/converter block pair is independent, they can be designed as modular elements and as a result benefit from economies of scale. Due to the independent converter blocks, \((n + 1)\) redundancy can easily be achieved by providing one more converter than is required for a given power rating. Because both the controller and converter are redundant, the converters do not introduce a single point of
To realize cost savings associated with parallel converters it is essential to limit additional hardware, such as bulky isolation transformers. For this reason, a configuration with directly-connected ac- and dc-buses is chosen.

1.2 Project Objectives

The objective of this thesis is to determine the feasibility of constructing an independently controlled parallel-VSC arrangement. To do this, the simplest characteristic parallel motor drive, consisting of two converters, is examined.

The project consists of the mathematical derivation of the system model, development of an appropriate controller, and validation through simulation and experiment. A linearized small-signal state-space model of the system and controller is developed in order to achieve system stability. The system is simulated in MATLAB/Simulink and the results are validated against an experimental system which was developed for this project.

Two main problems were encountered: load sharing (both dynamic and steady-state) and circulating 0-sequence current. The literature survey, presented in the next section, introduces the existing research on these two issues.

1.3 Literature Review

This section outlines the findings of the literature review. Of particular interest was research on power sharing, focused in the field of uninterruptible power supplies (UPSs) and the problem of circulating 0-sequence current, primarily centred in the area of parallel three-phase boost rectifiers (PTBRs).
1.3.1 Load Sharing

Load sharing, both dynamically and in steady-state, is problematic in parallel converter applications because converters have limited overload capacity, low output impedance, and are capable of a fast response [1]. Their controllers are typically also very sensitive to parameter variations.

The conventional approach to load sharing in motor drives is to use control interconnections and designate one module as the “master” over several “slave” units [2, 3], but this approach is not redundant [4]. In UPS applications, the problem of modular load sharing particularly for distributed, modular configurations, has been widely addressed [5, 6].

For the most part, these approaches make use of load-sharing droops based on the traditional power system droop method where the active power flow is dominated by the angle between the converter voltage vectors. Variations to this method include the addition of harmonics mitigation [4, 5] and the choice of other reference frames in which to apply the control [7, 8] in order to improve dynamic response. These methods require adaptation for a motor drive application because they are designed to operate when the line frequency is fixed.

Another parallel converter system where independent controllers have been studied is among the PTBR literature, but the emphasis has been on master/slave arrangements [9]. A non-linear control approach is proposed in [10], but a linear approach is preferred so that linear system theory applies.

A completely different approach to the problem of load sharing using current-sharing reactors [11], where the phase currents from each converter module are forced to be equal by coupling them through an inductor’s magnetic flux. This method, however, is not modular as the reactor must be specified in advance. Also, if one converter fails then current will not flow in the coupled reactor.

A third possibility is to use a machine with separate windings for each converter [12].
This addresses the reliability issues but is not inherently modular and requires a complete motor redesign.

Specifically in motor drive applications, [13] develops a motor drive consisting of independent converter modules. In this method, current sharing is performed by introducing an emulated impedance whose magnitude is significantly greater than that of the real interconnection impedance. However, this control relies on clock synchronization and its dynamic response is unknown.

1.3.2 0-Sequence Circulating Current

Where both the ac- and dc-bus are common, as is the case in this project, a path for 0-sequence circulating current exists. This has commonly been avoided by including isolation transformers on the ac side [4], however this is a costly and bulky solution. Mitigation of the 0-sequence circulating current has been addressed primarily for PTBRs, such as in [9, 10, 14]. Solutions employing converter communication [3], a supervisory control [15], synchronization [13] or a master-slave approach [16] have been proposed, however the underlying mechanism for the presence of 0-sequence circulating current has only recently been considered [9].

For the case of a PTBR, [14] introduced an independent controller to minimize the 0-sequence current using a modified space-vector modulation (SVM) scheme. This approach varies the duration of each zero vector to counteract whatever 0-sequence current is present, but suffers from saturation problems when the converter is operating at its limit and has unknown transient response. A non-linear approach has been suggested in [10], but a linear control is preferred.

A more comprehensive model of the PTBR has been developed by [9] and a controller to reduce the 0-sequence current is proposed. This method, however, relies on a common current reference provided by one of the converters. This is a type of master-slave arrangement, and as this reference is also a function of the number of converters it is not
modular unless it can be updated dynamically.

A more recent work [17] claims to develop a generalized system including common-mode passive and active elements, and a controller for a soft-switched motor drive system is developed from this. However, this controller cannot limit the 0-sequence currents at low frequencies which limits its applicability to this project.

1.4 Overview of Thesis

To accomplish dynamic and steady-state load sharing, the majority of these control methods rely on converter communication and synchronization, which is not desirable from a cost, simplicity, or reliability perspective. The autonomous controllers tend to rely on a modified power-system droop method which is not immediately applicable to a motor drive application.

Most methods for controlling circulating 0-sequence current also rely on inter-converter communication, a master/slave arrangement or extra hardware. These are either not modular solutions or add additional costs.

Therefore an independent, directly-connected parallel VSC configuration has important advantages in reliability and modularity. In order to implement it successfully, a new controller which takes into account 0-sequence circulating current and dynamic load sharing without relying upon inter-controller communication or a master/slave approach is required.

Chapter 2 describes the experimental system. The state space model of the system is given in Chapter 3. Chapter 4 develops the control strategies and evaluates the stability of the resulting closed-loop system.

The simulation model is validated against the experimental system and the effectiveness of the controller is presented in Chapter 5. Conclusions follow in Chapter 6.
Chapter 2

System Description

2.1 Overview

The chosen study system consists of two VSCs with common ac- and dc-buses driving a single permanent-magnet synchronous machine (PMSM). The VSCs are driven using separate controllers. The only common feedback signal is the rotor position of the PMSM. This configuration is the simplest parallel-converter arrangement and provides a basis for evaluating control performance both theoretically and experimentally. The system schematic is shown in Fig. 2.1 with important system parameters summarized in Table 2.1. This system was implemented in simulation and in the laboratory.

Section 2.2 introduces the motor, denoted as \textit{PMSM} in Fig. 2.1. The converters, \textit{VSC1} and \textit{VSC2}, the input capacitors \(C_{dc}\), and the associated control hardware (not shown in the figure) are described in Section 2.3.

The dc-side, consisting of the \textit{dc source} and the chokes \(L_{dc}\), is described in Section 2.4. On the ac side, the motor is connected to both converters through ac output filters (\textit{ac filter 1} and \textit{ac filter 2}) at each converter’s terminals. These ac filters are described in Section 2.5.

Components were sized in order to match, on a per-unit basis, an existing industrial
Table 2.1: System Parameters

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{dc}$ [V]</td>
<td>400</td>
</tr>
<tr>
<td>$C_{dc}$ [$\mu$F]</td>
<td>360</td>
</tr>
<tr>
<td>$L_{dc}$ [mH]</td>
<td>2.5</td>
</tr>
<tr>
<td>$f_{sample}$ [kHz]</td>
<td>11.12</td>
</tr>
<tr>
<td>$f_{switch}$ [kHz]</td>
<td>5.56</td>
</tr>
<tr>
<td>$C_{DM}$ [$\mu$F]</td>
<td>1.5</td>
</tr>
<tr>
<td>$C_{CM}$ [$\mu$F]</td>
<td>0</td>
</tr>
<tr>
<td>$L_{DM}$ [mH]</td>
<td>0.93</td>
</tr>
<tr>
<td>$L_{CM}$ [mH]</td>
<td>2.4</td>
</tr>
</tbody>
</table>

single-converter configuration.

2.2 Permanent-Magnet Synchronous Machine

2.2.1 Machine Description

The PMSM is a Kollmorgen GOLDLINE M-803-A. The datasheet, motor parameter measurements and encoder details are included as Appendix A. A few important parameters are listed in Table 2.2. The motor is a trapezoidally wound stepper-motor.

The machine is fitted with a 4000-pulse encoder to provide the rotor position as feedback for the controllers. The rotor position is differentiated by the controllers to obtain the machine speed, expressed as $\omega_m$ [rad./s]. The electrical speed [electrical rads./s] is
Chapter 2. System Description

Figure 2.1: System Schematic
Table 2.2: Motor Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power [kW]</td>
<td>7.9</td>
</tr>
<tr>
<td>Voltage [V&lt;sub&gt;1-rms&lt;/sub&gt;]</td>
<td>250</td>
</tr>
<tr>
<td>Rated Continuous Current [A&lt;sub&gt;rms&lt;/sub&gt;]</td>
<td>24.9</td>
</tr>
<tr>
<td>Rated Electrical Speed [elec. Hz]</td>
<td>100</td>
</tr>
<tr>
<td>Rated Mechanical Speed [rpm]</td>
<td>2000</td>
</tr>
<tr>
<td>DC Resistance R&lt;sub&gt;s&lt;/sub&gt; [Ω]</td>
<td>0.18</td>
</tr>
<tr>
<td>q-axis Inductance L&lt;sub&gt;q&lt;/sub&gt; [mH]</td>
<td>6.0</td>
</tr>
<tr>
<td>d-axis Inductance L&lt;sub&gt;d&lt;/sub&gt; [mH]</td>
<td>3.3</td>
</tr>
</tbody>
</table>

given by:

\[
\omega_e = \frac{p}{2} \omega_m
\]

where \( p = 6 \) is the number of poles.

The machine was simulated assuming sinusoidal windings. The trapezoidal windings affect the current harmonics during voltage-control mode and cause increased voltage harmonics during current-control mode, but the simulation still accurately reflected the experimental results.

### 2.2.2 Machine Loading

The PMSM is loaded using a 5.6 kW dc dynamometer. The dynamometer is separately excited and operates with rated field current supplied by the lab’s 230 V dc supply. The dynamometer is loaded resistively to give a load torque characteristic of the form:

\[
T_L = k_L \omega_e
\]

In the experiments, \( k_L \) was typically varied between 0.03 and 0.1.
2.3 Converters and Control Hardware

2.3.1 Converter and $C_{dc}$

The converters are the 5 kVA, 3-phase voltage-sourced converters found in the lab fitted with 600 V fuses and higher-voltage capacitors to run using a higher dc-link voltage. For the dc-link capacitors $C_{dc}$, 360 $\mu$F film capacitors, 947C361K801CAMS by Cornell Dubilier, were chosen. These capacitors are rated for 800 Vdc, have much lower ESR than an equivalently-sized electrolytic capacitor, and can tolerate larger ripple currents. A capacitance of 360 $\mu$F at each converter provides 7.3 J per kW of motor rating. This value is five times larger than the minimum benchmark dc link capacitor for a drive cited in [18].

2.3.2 RT-Linux Controllers

Each VSC is controlled using a computer running Real-Time Linux. The two computers are completely independent but are each provided with motor position feedback and a common command signal. In an industrial application, the command signal would correspond to, for example, a common, external speed reference. In this project a common signal derived from a function generator served the same purpose.

The computers are programmed using C code and parameters can be varied in real-time using the control interface shown in Fig. 2.2. The code is given in Appendix C. The user-interface is described in Appendix D. Switching is at 5560 Hz. Sampling occurs at peaks and valleys of the carrier signal leading to a sampling frequency of 11.12 kHz.

2.3.3 Anti-Aliasing Filters

Anti-aliasing filters are installed on the feedback current and voltage signals. In a single converter system, the use of synchronous sampling allows sampling at twice the switching
frequency without observing switching ripple or aliasing. When a second, asynchronously-switched converter is present this is no longer true because converter 1 is unable to sample synchronously with converter 2’s switching. Because the switching ripple has a large amplitude, an anti-aliasing filter with significant attenuation at the switching frequency is required.

The laboratory setup uses second-order Sallen-Key low-pass filters with a cut-off frequency of $f_c = 1.85$ kHz with $\zeta = 0.8$. This provides theoretical attenuation of 20 dB at the switching frequency (5.56 kHz), and in-lab tests of the filter indicated 19 dB attenuation at the switching frequency.

2.4 dc Side

2.4.1 dc Source

The dc-bus provides the energy to operate the PMSM. The primary dc energy source ($C_{source}$) is a 38.4 mF series-parallel bank of 32 4800 µF 350 V electrolytic capacitors
which smooths the rectifier output and acts as a stiff dc source for the converters.

Upstream of this capacitor, the voltage step-up is accomplished using a three-phase variac followed by a 3:1 step-up transformer. The output is rectified using a 6-pulse diode bridge to provide the 400 V bus voltage, which is connected to the large capacitor through 5 mH chokes ($L_{source}$) on both the positive and negative phases to provide additional filtering.

To start up, the variac output voltage is increased from zero until the measured bus voltage is at 400 V. This avoids overvoltage which would result from ringing on the capacitors during startup.

In simulation, the dc source was treated as an ideal 400 V supply.

### 2.4.2 dc Chokes $L_{dc}$

Between the capacitor bank and each converter, additional dc chokes ($L_{dc} = 2.5 \, mH$) on each of the positive and negative lines provide impedance between the relatively small converter capacitors ($C_{dc}$) and the main dc source ($C_{source}$). This arrangement provides EMI filtering, mimics long lines (or a weak source) by limiting the rate of energy supply to the $C_{dc}$ and, as will be seen, provides additional 0-sequence impedance for the two-converter case (Section 2.5.3).

### 2.5 ac Filters

#### 2.5.1 Filter Description

The ac-bus consists of the interconnection between the two converter terminals and the motor terminal (the point of common connection, PCC). Each converter’s output passes through a filter consisting of a common-mode choke and a differential-mode inductor followed by differential-mode capacitors.
Both the capacitors and the inductors are sized according to the existing industrial drive and were intended to contain EMI resulting from the converter’s switching. For the capacitors, this means that a low inter-phase impedance exists at the switching frequency, while the inductors present a relatively high impedance to switching ripple currents.

A common-mode choke affects only the 0-sequence current (often called the “common-mode” current), while differential-mode inductors and capacitors affect the $q$- and $d$-axis currents (transformed versions of what are commonly called the “differential-mode” currents).\(^1\) The transformation into the $qd0$ frame is detailed in Section 3.2.1.

Section 2.5.2 gives practical details about the filter implementation and the three sections following introduce and attempt to quantify the switching ripple current resulting from the asynchronous parallel converter arrangement.

### 2.5.2 Implementation Details

In the laboratory, the converters are physically close together so the capacitances from both filters were implemented as a net differential-mode capacitance at the PCC. The per converter, phase-to-phase capacitance ($C_{DM}$) is 1.5 $\mu$F, resulting in a total capacitance of 3 $\mu$F phase-phase.

The common-mode chokes ($L_{CM}$) were constructed using 17 turns of AWG#8 wire (per phase) wrapped together around two 8x3 cm laminated transformer-steel U-cores bound together to form a continuous magnetic circuit. A small air gap was included to increase the device’s linearity and control the inductance. The common-mode inductance is 2.2 mH and the differential-mode inductance is 0.1 mH. One of the chokes is pictured in Fig. 2.3.

The differential-mode inductors ($L_{DM}$) are 0.83 mH three-phase units made by Rex

\(^1\)It is important not to confuse the terms common-mode and differential-mode which appear in this section with references to the common- and differential-mode currents in the remainder of this document. Those references refer to the common-mode current supplied to the motor and the differential-mode current which circulates in the two-converter case, as discussed in Section 4.5.
Figure 2.3: One of the two common-mode chokes constructed for this project.

Power Magnetics and are constructed using an E-I core pair with a small air gap. The common-mode inductance is less than 0.1 mH.

The common-mode choke has significantly higher impedance than the differential-mode inductor. Since only the 0-sequence current contributes to the flux, the cross-section of the magnetic core can be kept relatively small.

2.5.3 Switching Ripple in the Two-VSC Configuration

In the single converter configuration the ac filter was only required to contain EMI, with the motor’s inductance acting as an additional output filter for whatever switching ripple current was not absorbed by the ac filter. When a second converter is added without synchronizing the two converters’ switching, the filter takes on added importance because inter-converter ripple current can now exist.

The ac filter can be designed such that there will be essentially no $qd$ inter-converter
switching ripple because of the presence of the differential-mode capacitor at the PCC, which simplifies the design of the two-converter system. If the PCC looks like a ground to switching ripple, then the presence of another converter will not affect the closed-loop current path. This can be accomplished by ensuring that the differential-mode impedance of each inductor at the switching frequency is sufficiently larger than the differential-mode impedance presented by the capacitor. In the experimental system this is largely true because the inductor’s impedance is approximately three times larger than the net capacitive impedance at the PCC, thus it is assumed that the $qd$ switching ripple is not significantly affected by the addition of the second converter. Resizing of $L_{DM}$ due to the introduction of a second converter is therefore not considered.

A common-mode capacitor could also be added at the PCC, and this would have a comparable effect on the 0-sequence switching ripple. In this system, however, this capacitor was not present, so the addition of the second converter provides a closed-loop path for 0-sequence switching ripple. This path did not exist in the single converter case because the motor is an ungrounded three-wire device. This means that 0-sequence impedance is required between the converters in order to limit the 0-sequence switching current. Because a conventional three-legged, three-phase inductor provides very low impedance to 0-sequence current, either a specially designed differential-mode inductor or a separate common-mode choke is required.

For 0-sequence current, (that is current corresponding to a non-zero $(i_a + i_b + i_c)$, which implies a net flow of current through the converter) the loop is closed by the path shown in Fig. 2.4. The path goes through one converter, around the ac-side seeing only the common-mode chokes $L_{CM}$, through the other converter, and then through the dc side. The impedance on the dc side is given by the path connecting the top (bottom) rail of converter 1 to the bottom (top) rail of converter 2. Three parallel paths exist:

- $L_{dc}(\text{top, VSC 1}) \rightarrow V_{\text{source}} \rightarrow L_{dc}(\text{bottom, 2})$
- $C_{dc}(1) \rightarrow L_{dc}(\text{bottom, 1}) \rightarrow L_{dc}(\text{bottom, 2})$
• $L_{dc}(\text{top, }1) \rightarrow L_{dc}(\text{top, }2) \rightarrow C_{dc}(2)$

Because the capacitor $C_{\text{source}}$ is much larger than $C_{dc}$, the impedance through that path is approximately $2L_{dc}$ and the two other paths are not considered. The driving voltage is the difference between the two converters’ 0-sequence voltages.

![Diagram showing the closed-loop path for the 0-sequence switching ripple current.](image)

Figure 2.4: Closed-loop path for the 0-sequence switching ripple current.

The explanation of the 0-sequence switching ripple current and guidelines for sizing the common-mode chokes follow in the next section.

### 2.5.4 Characterization of 0-Sequence Switching Ripple

In order to understand the generation of 0-sequence switching voltage, consider the following situation. When the motor is turning slowly, its back-emf is small and thus in steady-state the average voltage requested by the controller will also be relatively small. This means that both controllers’ pulse-width modulators will implement this voltage using a duty cycle close to 50% on all three phases. If the same triangular carrier signal is compared with all three phases then, because the duty cycle is close to 50%, the three phases supplied by a single converter will have nearly in-phase switching. This is
shown in Fig. 2.5 and results in 0-sequence voltage \( v_0 = (v_a + v_b + v_c) / 3 \) at the switching frequency.

Even when the motor is not turning slowly, 0-sequence switching voltage will always exist because at least two of the phases need to be high (or low) at any given instant, giving a non-zero \( v_0 \). The worst case, however, is observed when the motor’s back-emf is small so that all three phases switch essentially together.

For the case of two asynchronous converters, the phase difference between their switching will vary periodically from 0 to 360°, with 180° corresponding to completely out-of-phase switching. At that point, although the duty cycles should be equal as both controllers request similar voltages, the converters will switch oppositely. If the duty cycle is close to 50%, this will mean that, for example, when phase A of converter 1 is switched high, converter 2 will be switched low. This switch configuration will last for approximately half of the switching period until which time the situation reverses, resulting in a large and alternating 0-sequence potential \( v_{01} - v_{02} \). Because both converters are present, the closed-loop path for 0-sequence discussed above allows 0-sequence switching ripple current to circulate. Fig. 2.6 shows the simulated 0-sequence driving voltage \( v_{01} - v_{02} \) when the two converters are passing through in-phase and 180° out-of-phase switching.

The maximum 0-sequence switching ripple can be estimated by considering the current in an inductor according to:

\[
\Delta i_0 \approx \frac{V}{L_{0\text{net}}} \frac{T_{\text{switch}}}{2}
\]

where \( \Delta i_0 \) is the 0-sequence switching ripple, \( V \) is the applied voltage (\( V_{dc} \) in this case) and \( L_{0\text{net}} = L_{01} + L_{02} + 2L_{dc} \) is the common-mode inductance around the 0-sequence closed-loop path of Fig. 2.4. This gives a peak-to-peak switching ripple of approximately 3.5 A.

Figure 2.7 shows 0-sequence ripple current observed in the laboratory. The test was run with a low motor speed, resulting in a large periodic \( v_0 \), and the scope screen was captured at the maximum and minimum points (corresponding to 180° out-of-phase and
in-phase switching). This result agrees very well with the approximation of Eqn. (2.3). A similar test performed in simulation showed a very similar response.

### 2.6 Chapter Conclusion

Simulations based on the system described here correspond well with laboratory results, indicating that the salient features have been captured. This is true both for low-frequency and switching ripple currents.

The periodic variation in the 0-sequence switching current could be eliminated by adding the common-mode capacitor $C_{CM}$ at the PCC. If sized appropriately, this would cause a constant 0-sequence switching ripple current even in the single converter case. The addition of the second converter would have minimal effect on the 0-sequence switching ripple, as was the case for $qd$ switching currents.

The inductors and capacitors limit the switching frequency ripple current by providing an impedance in the current path. This switching-frequency ripple current cannot be controlled by because the converters do not have sufficient bandwidth. The remaining chapters present the control of the low-frequency currents through time-averaged modelling and control.
Figure 2.5: Explanation of duty cycles for a requested voltage of 0.10 p.u.: (a) Converter 1’s voltage references, highlighting the time period examined in (b)-(d). (b) Requested voltages (normalized against the dc supply voltage $v_{\text{DC}/2}$) and the triangular carrier signal (c) Duty cycles resulting from the comparison and (d) Instantaneous $v_{q0}$ for converter 1.
Figure 2.6: The (a) minimum and (b) maximum 0-sequence driving voltage when two asynchronous converters are present, corresponding to the region around in-phase and 180° out-of-phase switching, respectively. As in Fig. 2.5, the requested voltage was 0.10 p.u.
Figure 2.7: Experimentally observed 0-sequence ripple current, 6.7 A/V (i.e. 0.67 A/div). (a) Minimum ripple, corresponding to the converters near in-phase and (b) maximum ripple, when the converters are switching 180° out-of-phase.
Chapter 3

System Model

3.1 Overview

In this chapter a linearized state-space model of a two converter system driving a single PMSM is developed. The chapter focuses on the development of an open loop model that can be interfaced to any specified controller equations. A proposed control design approach will follow in Chapter 4, along with the development of an associated closed-loop linearized model.

Section 3.2 identifies the PMSM equations, starting with the rotating reference frame which was chosen and the rationale for that choice. Section 3.3 develops the averaged converter models in the $qd$ and $0$ reference frames. This leads into Section 3.4, where the state-space $qd$ model of the converters plus machine is derived. Section 3.5 derives the $0$-sequence state-space model. The chapter concludes with the complete state-space description of the system in Section 3.6.

3.2 Machine Model

In this section the machine equations transformed into the rotating reference frame aligned with the machine’s rotor are presented. First, the specific equations used to
transform $abc$-frame quantities into the rotating $qd0$-frame quantities are given as multiple transforms exist in the literature. The state equations and the relationships between voltages and currents used for simulation and control are then presented.

### 3.2.1 Rotating $qd0$ Reference Frame

Choosing a rotating reference frame aligned with the machine’s rotor gives non time-varying inductances and allows for a linear control approach. This choice of reference frame is an integral part of conventional brushless dc, or field-oriented, control. The transformation and resulting machine equations are taken from [19].

The transform from the stationary (stator-side) $abc$-frame into the rotating $qd0$-frame is from Eqn. (6.9-17) in [19]:

\[ f_{qd0} = K f_{abc} \]  
\[ f_{abc} = K^{-1} f_{qd0} \]

where

\[ K = \frac{2}{3} \begin{bmatrix} \cos \theta & \cos(\theta - 2\pi/3) & \cos(\theta + 2\pi/3) \\ \sin \theta & \sin(\theta - 2\pi/3) & \sin(\theta + 2\pi/3) \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \]  
\[ K^{-1} = \begin{bmatrix} \cos \theta & \sin \theta & 1 \\ \cos(\theta - 2\pi/3) & \sin(\theta - 2\pi/3) & 1 \\ \cos(\theta + 2\pi/3) & \sin(\theta + 2\pi/3) & 1 \end{bmatrix} \]

The $d$-axis is chosen to be aligned with the rotor’s magnetic field, resulting in an induced voltage along the $q$-axis, as shown in Fig. 3.1. This results in torque being produced by $q$-axis current, as discussed in the next section.
3.2.2 Machine Equations

The machine equations for the PMSM are typically given using the machine’s flux linkages as the state variables. The flux linkage model is preferred for the machine model because it provides flexibility to allow for modelling non-linear relationships between current and flux (that is, non-constant inductances, though this was not used for this project).

The flux linkage states, $\lambda_q$, $\lambda_d$ and $\lambda_0$, are described by Eqns. (7.10-24), (7.10-25) and (7.10-26) from [19]:

$$\frac{d\lambda_q}{dt} = v_q - R_s i_q - \omega_e \lambda_d$$ (3.5)

$$\frac{d\lambda_d}{dt} = v_d - R_s i_d + \omega_e \lambda_q$$ (3.6)

$$\frac{d\lambda_0}{dt} = v_0 - R_s i_0$$ (3.7)

where the subscripts $q$ and $d$ indicate a quantity associated with either the $q$- or $d$-axis, and the subscript 0 indicates a quantity associated with the 0-sequence. $v_q$, $v_d$ and $v_0$ are the voltage at the machine terminals along each axis, $R_s$ is the stator resistance, and $i_q$, $i_d$, and $i_0$ are the machine currents. For the three-wire motor used here, there is no path to ground for 0-sequence current [19], resulting in the motor’s $i_0$ remaining zero throughout. This can be captured by modifying Eqn. (3.7) to capture the effective
Chapter 3. System Model

25

resistance to ground through the motor:

\[ \frac{d\lambda_0}{dt} = v_0 - (R_s + R_0)i_0 \] (3.8)

where \( R_0 \to \infty \) because of the three-wire motor.

When the magnetics are linear, the flux linkages can be expressed in terms of the inductances and currents according to \( \lambda_q = L_q i_q, \lambda_d = L_d i_d + \lambda'_m \) and \( \lambda_0 = L_0 i_0 \), giving:

\[ L_q \frac{di_q}{dt} = v_q - R_s i_q - \omega_e (L_d i_d + \lambda'_m) \]
\[ L_d \frac{di_d}{dt} = v_d - R_s i_d + \omega_e L_q i_q \] (3.9)
\[ L_0 \frac{di_0}{dt} = v_0 - (R_s + R_0)i_0 \]

with \( \lambda'_m \) the amplitude of the flux linkages generated by the permanent magnets referred to the stator side.

Isolating the voltages on the left-hand side of Eqn. (3.9) gives:

\[ v_q = L_q \frac{di_q}{dt} + R_s i_q + \omega_e (L_d i_d + \lambda'_m) \]
\[ v_d = L_d \frac{di_d}{dt} + R_s i_d - \omega_e L_q i_q \] (3.10)
\[ v_0 = L_0 \frac{di_0}{dt} + (R_s + R_0)i_0 \]

To completely describe the machine, the equations relating the electrical and mechanical states and the equations describing the mechanical states are also required. The developed torque \( T_e \) is given by Eqn. (7.10-35) from [19]:

\[ T_e = \frac{3p}{2} \frac{\lambda'_m}{2} \] (3.11)

where \( p \) is the number of poles. Notice that the machine torque is proportional to, and is only a function of, \( i_q \)

\(^1\)This allows the coordinate transform to be properly oriented (as in Fig. 3.1). To orient the rotor’s \( d \)-axis with respect to the encoder’s reset, the fact that no torque results from non-zero \( i_d \) is exploited: a large current space vector is applied to the machine and when no torque is produced, the space vector is aligned with the \( d \)-axis (or the negative \( d \)-axis). The significance of the negative \( d \)-axis is that a positive \( i_q \) will result in negative speed.
where $T_L$ is the load torque and $J$ is the rotor plus load inertia. The electrical speed $\omega_e$, in [electrical rad./s], is related to the mechanical rotor speed $\omega_r$ according to:

$$\omega_e = \omega_r \frac{p}{2}$$

(3.13)

The electrical speed is chosen as a state variable instead of the mechanical speed because all the electrical calculations depend on $\omega_e$. The speed state equation, from Eqns. (3.11), (3.12) and (3.13), is given by:

$$\frac{d}{dt} \omega_e = \frac{p}{2} \left( \frac{3\mu \lambda_m i_q - T_L}{J} \right)$$

(3.14)

3.2.3 Input-Output Block for Machine Equations

These machine equations describe an input-output block of the form shown in Fig. 3.2 where $v_{qd0}$ and $i_{qd0}$ are related by Eqn. (3.10) and $\omega_e$ is related to $i_{qd0}$ by Eqn. (3.14). In these two sets of equations the 0-sequence quantities are decoupled from the $qd$ quantities. This allows the overall input-output block to be replaced with two separate blocks, Fig. 3.3, that are used to represent the machine in the development of the overall state-space system model.

![Diagram of PMSM with input and output](image-url)
3.3 Averaged Converter Circuit

To simplify the development of the state-space model of the two-converter system the averaged $qd$ and 0-sequence converter models are developed in this section. These models allow the formulation of simplified differential equations relating the converter and machine quantities. The separation of the $qd$ and 0-sequence models is justified because applying the $qd0$ reference frame transformation of Section 3.2.1 to a single converter results in a decoupled system. Additionally, the motor $qd$ and 0-sequence components were demonstrated to be decoupled in the previous section.

Fig. 3.4 shows a single converter plus its input and output filters (from the two-converter system, Fig. 2.1). The averaged converter model is a simplification of this circuit and has to be valid for those frequencies at which the motor operates, given by $\omega_e$. In this case, the upper limit is 100 Hz and the lower limit is approximately dc.

At these frequencies, the output capacitor $C_{DM}$ (and $C_{CM}$, were it present), which is sized to suppress EMI, has high impedance and can be neglected. The dc source can be treated as ideal because the capacitor $C_{source}$ is large enough that even at low frequencies its impedance is low. For very low frequencies, the impedance of the diode rectifier and the chokes $L_{source}$ is very low, meaning this approximation remains valid.

Further simplifications necessitate examining the $qd$- and 0-sequence separately, as was done for the switching ripple in Chapter 2. This is done in the next two sections.
3.3.1 $qd$ Averaged Model

For the $qd$ model, the switching is neglected and the voltage source converter is represented by its time-averaged value. In the model, it appears as a low-frequency voltage source connected to ground. On the ac side, the common-mode choke $L_{CM}$ provides very little differential-mode impedance. This means that the inductance is given by: $L_1 = L_{DM}$. The wire resistance of $L_{CM}$ and $L_{DM}$ is lumped as the single parameter $R_1$. In this system, $R_1 = 0.1 \, \Omega$.

These simplifications are shown in Fig. 3.5, with the capacitor $C_{DM}$ greyed out and shown only for reference. Assuming the output filters are identical, the averaged $qd$ model of the two separate converters can be connected to the motor’s input/output block of the previous section, as shown in Fig. 3.6.
3.3.2 0-Sequence Averaged Model

In Section 2.5.3 the 0-sequence current path, which depends on the presence of both converters, was introduced. To reiterate the findings there: on the ac side, the differential-mode inductor $L_{DM}$ does not affect the 0-sequence current, resulting in an inductance of $L_{01} = L_{CM}$. On the dc side, the impedance is $2L_{dc}$. The lumped resistance $R_1$ corresponding to the ac inductors is the same as in the $qd$ model.

Connecting the 0-sequence motor input/output block gives the averaged 0-sequence converter model of Fig. 3.7.
3.4  \(qd\) Model of the Parallel Converter Arrangement

This section derives the linearized state-space model for the \(qd\) currents in the open-loop system. The basis for this derivation is the averaged converter model coupled with the motor, shown in Fig. 3.6.

Section 3.5 gives the 0-sequence state-space model to complete the description of the open-loop two-converter motor drive.

3.4.1  \(qd\) State-Space Derivation

The conventions shown in Fig. 3.6 cause the motor currents \(i_{qd}\), which are normally chosen as states, to be linear combinations of the converter currents:

\[
\begin{align*}
    i_q &= i_{q1} + i_{q2} \\
    i_d &= i_{d1} + i_{d2}
\end{align*}
\]  
\(3.15\)

The state vector \(x_{qd}\) of the combined motor-converter \(qd\) system is chosen as:

\[
x_{qd} = \begin{bmatrix} \omega_e & i_{q1} & i_{d1} & i_{q2} & i_{d2} \end{bmatrix}^T
\]  
\(3.16\)

This choice was made in order to allow the local controllers to control local state variables and simplifies the extension of the analysis to the case where more than two converters are present.

Solving the circuit relations gives expressions for the \(qd\) currents in converters 1 and 2. The differential equations are expressed as a function of the state variables, the control voltages \(v_{tqd1}\) and \(v_{tqd2}\), and the motor terminal voltage \(v_{qd}\):

\[
\begin{align*}
    L_1 \frac{d}{dt} i_{q1} &= -R_1 i_{q1} - \omega_e L_1 i_{d1} - v_q + v_{tqd1} \\
    L_1 \frac{d}{dt} i_{d1} &= \omega_e L_1 i_{q1} - R_1 i_{d1} - v_d + v_{td1} \\
    L_1 \frac{d}{dt} i_{q2} &= -R_1 i_{q2} - \omega_e L_1 i_{d2} - v_q + v_{tqd2} \\
    L_1 \frac{d}{dt} i_{d2} &= \omega_e L_1 i_{q2} - R_1 i_{d2} - v_d + v_{td2}
\end{align*}
\]  
\(3.17\)
These equations are non-linear because $\omega_e$, $i_{q,1,2}$ and $i_{d,1,2}$ are all state variables. Note that in this form converter 1 is not coupled to converter 2 directly. The coupling occurs through the motor, appearing through the voltage $v_{qd}$.

The machine description, Eqn. (3.10), is used to rewrite $v_{qd}$ in terms of the state variables. This yields:

\[
\begin{align*}
  v_q &= L_q \frac{d}{dt} (i_{q1} + i_{q2}) + R_s (i_{q1} + i_{q2}) + \omega_e (L_d (i_{d1} + i_{d2}) + \lambda'_m) \\
  v_d &= L_d \frac{d}{dt} (i_{d1} + i_{d2}) + R_s (i_{d1} + i_{d2}) - \omega_e L_q (i_{q1} + i_{q2})
\end{align*}
\] (3.18)

The open-loop state equations are obtained by substituting Eqn. (3.18) into Eqn. (3.17) and augmenting that system with Eqn. (3.14), the state equation for $\omega_e$, with $i_q = i_{q1} + i_{q2}$:

\[
\frac{d}{dt} \begin{bmatrix} \omega_e \\ i_{q1} \\ i_{d1} \\ i_{q2} \\ i_{d2} \end{bmatrix} = \begin{bmatrix} 0 & \frac{3p^2}{8} \lambda'_m (i_{q1} + i_{q2}) \\ -R_t & -R_t & -\omega_e L_{dt} & -R_s & -\omega_e L_d & \lambda'_m \\ -R_s i_{q1} & -R_s i_{d1} - R_s i_{d2} + \omega_e (L_1 + L_d) i_{d1} + (L_1 + L_d) i_{d2} + \lambda'_m \\ -R_s i_{d1} - (R_1 + R_s) i_{q1} - (R_1 + R_s) i_{d2} + \omega_e (L_q i_{q1} + (L_1 + L_q) i_{q2}) \end{bmatrix} + \begin{bmatrix} -\frac{v_{TL}}{2J} \\ v_{tq1} \\ v_{td1} \\ v_{tq2} \\ v_{td2} \end{bmatrix}
\] (3.19)

Linearizing about an operating point (with large signal values denoted by a bar, e.g. $\bar{\omega}_e$) gives the system in a linear state-space form:

\[
\frac{d}{dt} \begin{bmatrix} \omega_e \\ i_{q1} \\ i_{d1} \\ i_{q2} \\ i_{d2} \end{bmatrix} = \begin{bmatrix} 0 & \frac{3p^2}{8} \lambda'_m & 0 & \frac{3p^2}{8} \lambda'_m & 0 \\ A_{q2} & -R_t & -\omega_e L_{dt} & -R_s & -\omega_e L_d \\ A_{d2} & -\omega_e L_{qt} & -R_t & \omega_e L_q & -R_s \\ A_{q2} & -R_s & \omega_e L_d & -R_t & \omega_e L_{dt} \\ A_{d2} & -\omega_e L_{qt} & -R_s & \omega_e L_{qt} & -R_t \end{bmatrix} \begin{bmatrix} \omega_e \\ i_{q1} \\ i_{d1} \\ i_{q2} \\ i_{d2} \end{bmatrix} + \begin{bmatrix} -\frac{v_{TL}}{2J} \\ v_{tq1} \\ v_{td1} \\ v_{tq2} \\ v_{td2} \end{bmatrix}
\] (3.20)
with $R_t = R_1 + R_s$, $L_{dt} = L_1 + L_d$, $L_{qt} = L_1 + L_q$ and:

$$
\hat{L} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & L_{qt} & 0 & L_q & 0 \\
0 & 0 & L_{dt} & 0 & L_d \\
0 & L_q & 0 & L_{qt} & 0 \\
0 & 0 & L_d & 0 & L_{dt}
\end{bmatrix}
$$

$$
\begin{align*}
A_{32} &= -L_{dt} \dot{i}_{d1} - L_d \dot{i}_{d2} - \lambda'_m \\
A_{42} &= -L_{qt} \dot{i}_{q1} - L_q \dot{i}_{q2} \\
A_{52} &= -L_{d} \dot{i}_{d1} - L_{dt} \dot{i}_{d2} - \lambda'_m \\
A_{62} &= -L_{q} \dot{i}_{q1} - L_{qt} \dot{i}_{q2}
\end{align*}
$$

The sparse leading matrix $\hat{L}$ results from the differentials in the motor voltage equations which depend on both converters. Inverting $\hat{L}$:

$$
\hat{L}^{-1} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & \frac{L_1 + L_d}{L_1(L_1 + 2L_q)} & 0 & \frac{-L_q}{L_1(L_1 + 2L_q)} & 0 \\
0 & 0 & \frac{L_1 + L_d}{L_1(L_1 + 2L_q)} & 0 & \frac{-L_q}{L_1(L_1 + 2L_q)} \\
0 & \frac{-L_q}{L_1(L_1 + 2L_q)} & 0 & \frac{L_1 + L_d}{L_1(L_1 + 2L_q)} & 0 \\
0 & 0 & \frac{-L_d}{L_1(L_1 + 2L_q)} & 0 & \frac{L_1 + L_d}{L_1(L_1 + 2L_q)}
\end{bmatrix}
$$

and pre-multiplying gives the small-signal state-space $qd$ model for the open-loop two-converter system:

$$
\frac{d}{dt} x_{qd} = A_{qd} x_{qd} + B_{qd} u_{qd}
$$
where

\[
A_{qd} = \begin{bmatrix}
0 & \frac{3g^2}{8J} \gamma_m' & 0 & \frac{3g^2}{8J} \gamma_m' & 0 \\
-\frac{R_1 L_{q1} + R_s L_1}{L_1 (L_1 + 2L_q)} & -\frac{\pi \omega (L_1 + L_3 + L_4)}{L_1 (L_1 + 2L_q)} & \frac{R_1 L_q - R_s L_1}{L_1 (L_1 + 2L_q)} & \frac{\pi \omega (L_q - L_d)}{L_1 (L_1 + 2L_q)} \\
\frac{R_1 L_{q1} - R_s L_1}{L_1 (L_1 + 2L_q)} & -\frac{\pi \omega (L_q - L_d)}{L_1 (L_1 + 2L_q)} & -\frac{R_1 L_q + R_s L_1}{L_1 (L_1 + 2L_q)} & -\frac{\pi \omega (L_1 + L_3 + L_4)}{L_1 (L_1 + 2L_q)} \\
\frac{\pi \omega (L_1 + L_3 + L_4)}{L_1 (L_1 + 2L_d)} & \frac{R_1 L_{q1} - R_s L_1}{L_1 (L_1 + 2L_q)} & \frac{\pi \omega (L_q - L_d)}{L_1 (L_1 + 2L_q)} & -\frac{R_1 L_q + R_s L_1}{L_1 (L_1 + 2L_q)} \\
\end{bmatrix}
\]

\[
B_{qd} = \begin{bmatrix}
-\frac{R}{2J} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{L_{q1}}{L_1 (L_1 + 2L_q)} & 0 & \frac{L_{q1}}{L_1 (L_1 + 2L_q)} & 0 & 0 \\
0 & 0 & \frac{L_{q1}}{L_1 (L_1 + 2L_d)} & 0 & \frac{L_{q1}}{L_1 (L_1 + 2L_d)} & 0 \\
0 & \frac{L_{q1}}{L_1 (L_1 + 2L_q)} & 0 & \frac{L_{q1}}{L_1 (L_1 + 2L_d)} & 0 & \frac{L_{q1}}{L_1 (L_1 + 2L_d)} \\
\end{bmatrix}
\]

\[u_{qd} = \begin{bmatrix}
T_L & v_{iq1} & v_{id1} & v_{iq2} & v_{id2}
\end{bmatrix}^T
\]

\[
A_{32} = -\frac{(L_1 + L_q + L_4) \overline{i_{d1}} + (L_d - L_q) \overline{i_{d2}} + \gamma_m'}{L_1 + 2L_q}
\]

\[
A_{42} = -\frac{(L_1 + L_q + L_4) \overline{i_{q1}} + (L_q - L_d) \overline{i_{q2}}}{L_1 + 2L_d}
\]

\[
A_{52} = -\frac{(L_d - L_q) \overline{i_{d1}} + (L_1 + L_q + L_d) \overline{i_{d2}} + \gamma_m'}{L_1 + 2L_q}
\]

\[
A_{62} = -\frac{(L_q - L_d) \overline{i_{q1}} + (L_1 + L_q + L_d) \overline{i_{q2}}}{L_1 + 2L_d}
\]

Examining \(A_{qd}\) shows the response of converter 1 to \(i_{qd1}\) is identical to the response of converter 2 to \(i_{qd2}\) and vice-versa, which is as expected. The interactions with the motor \((A_{32}, A_{42}, A_{52} \text{ and } A_{62})\) are also in agreement. This confirms that from the system's
perspective the two converters are identical. The $A_{qd}$ matrix shows that converter 1 is closely coupled to converter 2. The matrix $B_{qd}$ shows that converter 2’s control voltages have almost as much impact on converter 1 as do converter 1’s voltages. This makes the control problem more difficult because when the converters are independent converter 1 does not have access to converter 2’s control voltages without additional sensing circuitry.

### 3.5 0-Sequence Model of the Parallel Converter Arrangement

This section develops the state-space model for the 0-sequence currents.

The convention for current in Fig. 3.7 relates the converter and motor currents:

$$i_0 = i_{01} + i_{02}$$  \hspace{1cm} (3.23)

From the discussion of the equations describing the motor’s input/output block, Fig. 3.3b, recall that the impedance to ground through the block is very large (as $R_0 \rightarrow \infty$). This results in $i_0 = 0$, which gives, from Eqn. (3.23):

$$i_{01} = -i_{02}$$  \hspace{1cm} (3.24)

This means that the 0-sequence current is a state common to both converters.

The equations describing this state can be derived using KCL applied around the closed loop path:

$$v_{i01} - v_{i02} = (2L_{01} + 2L_{dc}) \frac{d}{dt} i_{01} + 2R_1 i_{01}$$  \hspace{1cm} (3.25)

where the resistance associated with $L_{dc}$ is assumed to be small. Rearranging and adopting the conventions $L_{0\text{net}} = 2L_{01} + 2L_{dc}$ and $R_{0\text{net}} = 2R_1$ gives:

$$\frac{d}{dt} i_{01} = -\frac{R_{0\text{net}}}{L_{0\text{net}}} i_{01} + v_{i01} - v_{i02}$$  \hspace{1cm} (3.26)

This is the state-space description of the low-frequency 0-sequence current.
3.6 Complete Two-VSC System

The open-loop state model consisting of both the $qd$ and the 0-sequence terms is given by augmenting the $qd$ state vector to give:

$$x_{OL} = \begin{bmatrix} \omega_e & i_{q1} & i_{d1} & i_{q2} & i_{d2} & i_{01} \end{bmatrix}^T$$  \hspace{1cm} (3.27)

which varies according to:

$$\frac{d}{dt} x_{OL} = A_{OL} x_{OL} + B_{OL} u_{OL}$$  \hspace{1cm} (3.28)

The state-transition matrix $A_{OL}$, the control matrix $B_{OL}$ and the inputs $u_{OL}$ are given as:

$$A_{OL} = \begin{bmatrix} A_{qd} & 0_{5x1} \\ 0_{1x5} & -\frac{R_{net}}{L_{net}} i_{01} \end{bmatrix}^T$$

$$B_{OL} = \begin{bmatrix} B_{qd} & 0_{5x1} & 0_{5x1} \\ 0_{1x5} & 1 & -1 \end{bmatrix}^T$$

$$u_{OL} = \begin{bmatrix} u_{qd} \\ v'_{01} \\ v'_{02} \end{bmatrix}$$

The sub-matrices are defined as in Eqn. (3.22) with $0_{5x1}$ and $0_{1x5}$ representing 5x1 and 1x5 matrices of zeros. These zeros show that the $qd$- and 0-sequence currents are decoupled.

3.6.1 Control Considerations for the Two-Converter System

Several features of the open-loop two-converter system make the control difficult. In the single converter case, all state variables were associated directly with the controller, but in the two-converter system there are states which are associated primarily with the other converter. Due to the current/voltage relationship imposed by the motor, converter 1 is coupled to converter 2. The same 0-sequence current state, however, is shared by both
converters. It is also completely decoupled from the rest of the system. These issues are addressed in the next chapter.
Chapter 4

Controller Development

4.1 Control Methodology

The control objective is regulation of the motor speed using two independently controlled converters. Each converter controller receives position feedback from the motor. Each controller is also supplied with the same reference speed. Without inter-converter communication, each converter has access only to its own converter’s output currents and local dc-bus voltage ($C_{dc}$ voltage).

In order to facilitate speed regulation using linear control techniques and to exploit the decoupling between the $qd$ and 0-sequence quantities, the two-converter system is controlled using field-oriented control techniques. The system state variables for control are the $qd\theta$ currents for each converter, the machine speed, and whatever states are associated with the control. In order to have the 0-sequence voltage as a control parameter for the voltage-sourced converters, sinusoidal PWM (SPWM) is used rather than space-vector modulation (SVM).

To implement a fully modular system, the controllers are required to be both independent and scalable. To address scalability, it is desired that the same control be applied to each converter. To ascertain whether the closed-loop system is stable, an eigenvalue
Two challenges were expected as a result of the literature review. First, it is necessary that the converters share current equally transiently and in steady-state. The current sharing is not inherent because the converters act as stiff voltage sources, and it is possible to set up a significant circulating current while maintaining a specified operating point of the motor.

Second, the converters are able to produce a 0-sequence voltage. Use of two converters on the same dc- and ac-buses without isolation provides a path for 0-sequence current to flow. This current has no path through the motor, so it has no direct impact on the motor’s operation, but this circulating 0-sequence current introduces extra losses into the system and reduces the maximum $q$- and $d$-axis currents which the converter can supply. Contrary to the 0-sequence switching ripple discussed in Chapter 2 which is predominantly influenced by reactor design, circulation of low frequency 0-sequence current must be mitigated through control action.

The 0-sequence control is developed in Section 4.2. The problem of the machine control, which consists of a speed regulator and current controllers for the $q$- and $d$-axes, is presented in Section 4.3. Section 4.4 develops the closed-loop state-space model of the linearized system. The stability of the resulting system is analyzed in Section 4.5.

### 4.2 0-Sequence Control

As the 0-sequence quantities are decoupled from the $qd$ quantities in Eqn. (3.28), the 0-sequence quantities can be controlled independently from the machine quantities. This also indicates that the 0-sequence current has no direct impact on the control objective. Instead, the 0-sequence current can be thought of as a parasitic, where the control tries to minimize the value of this parasitic.

The time-averaged 0-sequence current results from slight differences in converter dead-
times and other non-idealities. Any 0-sequence current which flows eats into the maximum current which can flow in any given switch and increases losses without contributing to any control objective, and thus ideally the time-averaged 0-sequence current should be zero.

4.2.1 Proposed 0-Sequence Controller

The 0-sequence model, Eqn. (3.26), has two control inputs while there is only a single state to be controlled. The feedback control loop is shown in Fig. 4.1. Choosing both controllers \( C_{01}(s) = C_{02}(s) = C_0(s) \) maintains the modular approach. The signals \( \text{err1} \) and \( \text{err2} \) represent sensor errors and noise in the circuit. If either is non-zero, then choosing a PI control for \( C_0(s) \) will result in integral wind-up even if the references are equal.

![Figure 4.1: 0-sequence feedback control model](image)

For this reason, a proportional control \( C_0(s) = k_0 \) is chosen such that:

\[
v_{t01} = -k_0i_{01}
\]

\[
v_{t02} = -k_0i_{02}
\]

The references \( i_{01}^* \) and \( i_{02}^* \) are set to zero. The closed-loop 0-sequence model is then given...
by:
\[
\frac{d}{dt}i_{01} = -\frac{R_{\text{0net}} + 2k_0}{L_{\text{0net}}}i_{01}
\]
which is stable for all \( k_0 \geq -\frac{R_{\text{0net}}}{2} \). Assuming that both controllers are identical, as has been done here, allows \( k_0 \) to be tuned as if there is only one controller with a plant of \( 2/ (sL_{\text{0net}} + R_{\text{0net}}) \).

Unlike the development in Section 2.5.4 where the 0-sequence switching ripple was examined, the control in this section is for low-frequency 0-sequence current. Although the impedance seen by the switching ripple and the averaged model is the same, the converters do not have sufficient bandwidth to control the switching ripple and the controller developed here must not respond to switching-frequency 0-sequence current. If the bandwidth is too high, then the low-frequency assumptions can be violated, resulting in the performance of the \( qd \) control being degraded. This places a limit on the maximum viable value of \( k_0 \). In this project, the chosen \( k_0 \) yields a closed-loop bandwidth of approximately \( f_{\text{switch}}/100 \).

4.3 Machine Control

Although the 0-sequence control is important, the control objective is regulation of the motor speed. To accomplish this, the quadrature currents must be regulated appropriately.

4.3.1 Conventional Single Converter Field-Oriented Control

For the single converter case, conventional field-oriented control can be used to regulate the speed when position feedback is available [20]. Field-oriented control is outlined briefly in this section. The converter currents are equal to the machine currents, so controlling the converter currents directly controls the machine currents.

The machine torque \( T_e \) is proportional to \( i_q \). Speed regulation is achieved using an
outer control loop to assign $i_q^\ast$. By using PI-controllers, the speed and $q$-axis current can be forced to track their references with zero steady-state error (in the presence of step-changes). Although $i_q$ is forced to track its reference, this capability is not strictly necessary because the speed controller will automatically adjust its output ($i_q^\ast$) if there is a difference between the speed and its reference.

To achieve maximum torque per unit current, $i_d$ is controlled to zero. This results in the closed-loop control structure for the single-converter case shown in Fig. 4.2. In order to decouple the $q$- and $d$-axis currents from each other, the cross-coupling terms are fed forward.

Assuming proper tuning of all PI gains, the resulting system exhibits a fast response and is stable.

### 4.3.2 Extension to the Two-Converter Case

The conventional field-oriented control is adapted for the two-converter case. The addition of the second converter adds the need to maintain system stability and ensure both dynamic and steady-state load-sharing. Adding a second converter also creates a path for 0-sequence current. Fortunately, the 0-sequence quantities are decoupled and
are controlled using the method developed in Section 4.2.1.

By examining the open-loop system of Eqn. (3.28), it is can be seen that each converter $k$’s currents $i_{qk}$ and $i_{dk}$ are closely coupled, through the motor, to the $q$- and $d$-axis currents in the other converter(s). As well, each converter’s currents are directly impacted by the other converter’s control voltage through the $B_{OL}$-matrix. This is potentially problematic because each converter has knowledge only of its own currents $i_{qk}$ and $i_{dk}$, effectively preventing a proper cancellation of the cross-coupling between the $q$- and $d$-axis currents.

In order to address the need for steady-state current sharing, a droop term is added to the speed PI-control loop to account for any speed measurement errors. As in the single converter case, there is no need to put a droop on the $q$-axis current PI-controller. $i_{qk}^*$ will be modified by each converter’s speed control loop as required. If the system is stable, dynamic load-sharing occurs as a result of the identical controllers. This choice of control maintains modularity and does not require inter-converter communication. System stability is verified in Section 4.5.

Unlike the 0-sequence case, no droop is required for the $d$-axis current PI-control loop because the motor is able to sink any current which results from sensor errors or noise. The proposed control loop is shown in Fig. 4.3.

The values to use for the cancellation of the cross-coupling $L$ terms can be chosen based on the closed-loop two-converter model presented in the next section. Converter 1 and converter 2 use identical controllers, so $C_{q1}(s) = C_{q2}(s) \doteq C_q(s)$ and $C_{d1}(s) = C_{d2}(s) \doteq C_d(s)$. The speed-control loop controllers are equal also: $C_{\omega 1}(s) = C_{\omega 2}(s) \doteq C_\omega(s)$. 

4.4 Closed-Loop Two-Converter System Model

The model developed in this section closes the loop for the system of Eq. (3.28) using the control described in the previous section, with particular reference to Fig. 4.3. For each converter $k$, that control can be summarized as:

- PI-control feedback $C_d(s)$ is applied to $i_{dk}$ to set $v_{tdk}$ with the references $i_{dk}^* = 0$ for each converter with feed-forward terms to cancel some of the coupling with the $q$-axis

- PI-control feedback is applied to $i_{qk}$ to set $v_{tqk}$ with the references $i_{qk}^*$ set by the speed control loops with feed-forward terms to cancel some of the coupling with the $d$-axis

- Drooped PI-control applied to $\omega_e$ to set $i_{qk}^*$ with the reference $\omega_e^*$ set externally
The presence of three PI-control loops (per controller) adds three states per controller to the system. These will be donated as $x_{sk}$, for the speed controllers’ integrators, and $x_{qk}$ and $x_{dk}$ for the $q$- and $d$-axis controllers’ integrators.

In order to close the loop, the values of control inputs need to be specified. From $u_{OL}$ in Eq. (3.28), these inputs are: $v_{tqk}$, $v_{tdk}$, $v_{t0k}$ (for each converter) and $T_L$. The voltages are the control inputs while $T_L$ is the mechanical loading supplied by the dynamometer and is described by Eq. (2.2) for a resistively loaded dc machine. For the purposes of the closed-loop analysis, $k_L = 0.05$ is assumed.

### 4.4.1 Speed Control Loop

The drooped PI-control used for the speed loop, giving the $i^*_{qk}$, has this transfer function where the PI is given by $C_\omega(s) = k_\omega(s + a_\omega)/s$:

$$
i^*_{qk} = \frac{k_\omega(s + a_\omega)}{s(1 + k_\omega D_\omega) + k_\omega a_\omega D_\omega} (\omega^*_e - \omega_e) \quad (4.3)
$$

where $D_\omega$ is the droop coefficient. This can be rewritten in state-space form, with the state $x_{sk} = \omega^*_e - \omega_e - D_\omega i^*_{qk}$ described by:

$$
dt x_{sk} = \frac{1}{1 + k_\omega D_\omega} (\omega^*_e - \omega_e) - \frac{k_\omega a_\omega D_\omega}{1 + k_\omega D_\omega} x_{sk} \quad (4.4)
$$

The output from the speed control loop is given by:

$$
i^*_{qk} = \frac{k_\omega}{1 + k_\omega D_\omega} (\omega^*_e - \omega_e) + \frac{k_\omega a_\omega}{1 + k_\omega D_\omega} x_{sk} \quad (4.5)
$$

### 4.4.2 $q$-axis Control Loop

The closed-loop $q$-axis control loop depends on $i^*_{qk}$, and defines the control voltage $v_{tqk}$ using this transfer function:

$$
v_{tqk} = (i^*_{qk} - i_{qk}) \frac{k_q(s + a_q)}{s} + \omega_e L_{cd} i_{dk} \quad (4.6)
$$
where $L_{cd}$ is the inductance associated with the cancelled cross-coupling term, and is free to be chosen. The state-space model in terms of the integrator state $x_{qk}$ using Eq. (4.5) for $i^*_{qk}$ is:

$$
\frac{d}{dt} x_{qk} = i^*_{qk} - i_{qk} = \frac{k_\omega}{1 + k_\omega D_\omega} (\omega^*_e - \omega_e) + \frac{k_\omega a_\omega}{1 + k_\omega D_\omega} x_{sk} - i_{qk} \quad (4.7)
$$

The output voltage is written in state-space form by subbing Eq. (4.5) into Eq. (4.6):

$$
v_{tqk} = k_q a_q x_{qk} + k_q \left( \frac{k_\omega}{1 + k_\omega D_\omega} (\omega^*_e - \omega_e) + \frac{k_\omega a_\omega}{1 + k_\omega D_\omega} x_{sk} \right) - k_q i_{qk} + \omega_e L_{cd} i_{dk} 
= k_q a_q x_{qk} + \frac{k_q k_\omega a_\omega}{1 + k_\omega D_\omega} x_{sk} + \frac{k_q k_\omega}{1 + k_\omega D_\omega} (\omega^*_e - \omega_e) - k_q i_{qk} + \omega_e L_{cd} i_{dk} \quad (4.8)
$$

Linearizing gives the converter voltage:

$$
v_{tqk} = k_q a_q x_{qk} + \frac{k_q k_\omega a_\omega}{1 + k_\omega D_\omega} x_{sk} + \frac{k_q k_\omega}{1 + k_\omega D_\omega} (\omega^*_e - \omega_e) - k_q i_{qk} + \omega_e L_{cd} i_{dk} + i_{dk} L_{cd} \omega_e \quad (4.9)
$$

completing the description of the $q$-axis closed-loop control voltage.

### 4.4.3 $d$-axis Control Loop

The closed-loop $d$-axis control loop always has zero input and defines the output control voltage $v_{tdk}$ according to the transfer function:

$$
v_{tdk} = -i_{dk} \frac{k_d (s + a_d)}{s} + \omega_e L_{cq} i_{qk} \quad (4.10)
$$

Using the state $x_{dk}$, defined as:

$$
\frac{d}{dt} x_{dk} = -i_{dk} \quad (4.11)
$$

allows the state-space description for the $d$-axis control voltage:

$$
v_{tdk} = k_d a_d x_{dk} - k_d i_{dk} - \omega_e L_{cq} i_{qk} \quad (4.12)
$$

Linearizing gives:

$$
v_{tdk} = k_d a_d x_{dk} - k_d i_{dk} - \omega_e L_{cq} i_{qk} - i_{qk} \omega_e L_{cq} \quad (4.13)
$$
4.4.4 Closed-Loop Model

The overall closed-loop model is shown in Fig. 4.4.

The control voltages given in Eqns. (4.1), (4.9), (4.13) and the loading description Eqn. (2.2), define the control inputs in $u_{OL}$. Along with the control matrix $B_{OL}$, these allow the closed-loop feedback control to be described.

The augmented closed-loop model uses the state vector $\hat{x}$:

$$\hat{x} = \begin{bmatrix} \omega_e & i_{q1} & i_{d1} & i_{q2} & i_{d2} & i_{01} & x_{s1} & x_{q1} & x_{d1} & x_{s2} & x_{q2} & x_{d2} \end{bmatrix}^T$$

Later, it will be advantageous to re-order the state vector and associated closed-loop state-transition matrix to group the $qd$ states for a single converter-controller pair together, followed by the differential-mode 0-sequence state $i_{01}$.

The only remaining input to this system is $\omega_e^*$. The control parameters, introduced in the previous sections, are $k_\omega$, $a_\omega$ and $D_\omega$ (speed loop), $L_{cd}$, $k_q$ and $a_q$ ($q$-axis current loop), and $L_{cq}$, $k_d$ and $a_d$ ($d$-axis current loop). The closed-loop form is given by:

$$\frac{d}{dt} \hat{x} = \hat{A}\hat{x} + \hat{B}\hat{u}$$

with $\hat{u} = \omega_e^*$.

Reordering the state vector $\hat{x}$ to $x_{CL}$ where:

$$x_{CL} = \begin{bmatrix} \omega_e & i_{q1} & i_{d1} & x_{s1} & x_{q1} & x_{d1} & i_{q2} & i_{d2} & x_{s2} & x_{q2} & x_{d2} & i_{01} \end{bmatrix}^T$$

(4.16)

gives:

$$\frac{d}{dt} x_{CL} = A_{CL}x_{CL} + B_{CL}u$$

with $u = \hat{u}$. This allows the system to be more easily extended to the case of more than two converters because the states associated with each converter’s $qd$ model are grouped together. First, this means that another converter can be added without a complicated reordering of the state-transition matrix. Second, that inter-converter interactions can be identified.
Figure 4.4: Overall controller flow diagram
The significance of the placement of $\omega_c$ at the beginning of and $i_{01}$ at the end of the state vector will be discussed in Section 4.5.2.

This choice of $x_{CL}$ causes the $A_{CL}$ matrix to have the form:

$$A_{CL} = \begin{bmatrix} -\frac{p_{kL}}{2j} & A_{1\omega} & A_{2\omega} \\ A_{\omega 1} & A_{11} & A_{21} \\ A_{c1} & 0_{3x5} & 0_{11x1} \\ A_{\omega 2} & A_{12} & A_{22} \\ 0_{3x5} & A_{c2} & A_0 \end{bmatrix} \quad (4.18)$$

The first row shows how the speed is affected by itself (through the dynamometer loading $-\frac{p_{kL}}{2j}$) and the converter currents/integrator states through $A_{1\omega}$ and $A_{2\omega}$. Because both converters contribute identically to the machine currents, $A_{1\omega} = A_{2\omega} = A_{c\omega}$.

The speed affects the converter currents and integrator states through $A_{\omega 1}$ and $A_{\omega 2}$. Converter 1’s currents interact with themselves and with the converter 1’s integrators through $A_{11}$, and with converter 2’s states through $A_{21}$. The zeros ($0_{3x5}$) indicate that converter 1’s integrators interact only with the states associated with converter 1, which is as expected based on the independent control approach.

The state-transition matrices associated with converter 2 are mirrors of those associated with converter 1, meaning that $A_{22} = A_{11} = A_s$, $A_{21} = A_{12} = A_n$ and $A_{c1} = A_{c2} = A_c$. $A_{\omega 1}$ and $A_{\omega 2}$ are not equal as they depend on large signal quantities (e.g. $i_{q1}$), but they mirror each other. The final column and final row of zeros correspond to the 0-sequence current which is decoupled from the rest of the system.
The matrix $A_{CL}$ can be equivalently rewritten as:

$$
A_{CL} = \begin{bmatrix}
-k_{c}A_{m} & 0 \\
A_{m1} & A_{m2} & 0_{1	imes 1} \\
A_{m2} & 0_{1	imes 1} & A_{m0}
\end{bmatrix}
$$

(4.19)

The sub-matrices of this form of the closed-loop state-transition matrix are given as:

$$
A_{m1} = \begin{bmatrix}
-k_{c}A_{m} & \frac{-L_{1}+L_{q}+L_{q}-L_{q}L_{q}/L_{1}}{L_{1}+2L_{q}} \frac{L_{q}-L_{q}-L_{q}L_{q}/L_{1}}{L_{1}+2L_{q}} \frac{L_{q}}{L_{1}+2L_{q}} - L_{1}+L_{q}+\frac{L_{q}}{L_{1}+2L_{q}} \frac{L_{q}-L_{q}-L_{q}L_{q}/L_{1}}{L_{1}+2L_{q}} \frac{L_{q}}{L_{1}+2L_{q}}
\end{bmatrix}
$$

$$
A_{m2} = \begin{bmatrix}
-k_{c}A_{m} & \frac{-L_{1}+L_{q}+L_{q}-L_{q}L_{q}/L_{1}}{L_{1}+2L_{q}} \frac{L_{q}-L_{q}-L_{q}L_{q}/L_{1}}{L_{1}+2L_{q}} \frac{L_{q}}{L_{1}+2L_{q}} - L_{1}+L_{q}+\frac{L_{q}}{L_{1}+2L_{q}} \frac{L_{q}-L_{q}-L_{q}L_{q}/L_{1}}{L_{1}+2L_{q}} \frac{L_{q}}{L_{1}+2L_{q}}
\end{bmatrix}
$$

$$
A_{cw} = \begin{bmatrix}
\frac{3p^{2}}{8J}\lambda_{m} & 0 & 0 & 0
\end{bmatrix}
$$

$$
A_{s} = \begin{bmatrix}
\frac{(R_{1}+k_{s})L_{q}}{L_{1}(L_{1}+2L_{q})} & \frac{-L_{1}+L_{q}L_{q}/L_{1}}{L_{1}+2L_{q}} \frac{L_{q}}{L_{1}+2L_{q}} \frac{-L_{1}+L_{q}L_{q}/L_{1}}{L_{1}+2L_{q}} \frac{L_{q}}{L_{1}+2L_{q}} & \frac{k_{s}k_{c}L_{q}}{L_{1}(L_{1}+2L_{q})} & \frac{k_{s}a_{m}L_{q}}{L_{1}(L_{1}+2L_{q})} & 0
\end{bmatrix}
$$
with \( L_t = L_1 + L_q + L_d \).

\[
A_n = \begin{bmatrix}
\frac{(R_1+k_\omega)L_q - R L_1}{L_1(L_1+2L_q)} & \frac{L_q - L_d - L_{cd} L_q/L_1}{L_1+2L_q} & -k_\omega k_\omega a_\omega L_q/L_1 & -k_\omega a_q L_q/L_1 & 0 \\
\frac{L_q - L_d + L_{cd} L_q/L_1}{L_1+2L_d} & \frac{(R_1+k_\omega)L_q - R L_1}{L_1(L_1+2L_d)} & 0 & 0 & \frac{-k_\omega a_q L_d/L_1}{L_1(L_1+2L_d)}
\end{bmatrix}
\]

\[
A_c = \begin{bmatrix}
0 & 0 & -k_\omega a_\omega D_\omega & 0 & 0 \\
-1 & 0 & \frac{k_\omega a_\omega}{1+k_\omega D_\omega} & 0 & 0 \\
0 & -1 & 0 & 0 & 0
\end{bmatrix}
\]

\[
A_0 = \frac{-R_0\text{net} - 2k_0}{L_0\text{net}}
\]

The control input matrix \( B_{CL} \) is given as:

\[
B_{CL} = \begin{bmatrix}
0 \\
\frac{k_\omega k_\omega}{(L_1+2L_q)(1+k_\omega D_\omega)} & 0 \\
\frac{k_\omega}{(L_1+2L_q)(1+k_\omega D_\omega)} & 0 \\
\frac{1}{1+k_\omega D_\omega} & \frac{k_\omega}{1+k_\omega D_\omega} & 0 \\
\frac{1}{1+k_\omega D_\omega} & \frac{k_\omega}{1+k_\omega D_\omega} & 0
\end{bmatrix}
\]

(4.20)
4.5 Stability Analysis

4.5.1 Common- and Differential-Mode Currents

The modes relating the converter current to the motor are the common modes of the system. The common-mode currents $i_q$ and $i_d$ are the sum of the converter currents:

$$i_q = \sum_k i_{qk} \quad (4.21)$$

$$i_d = \sum_k i_{dk}$$

where $k$ is the converter index. Here, $k$ is only 1 and 2 because there are two converters.

The states can be redefined in terms of these common-mode currents. When taken in concert with the machine speed $\omega_e$, which is also a common-mode state, the associated eigenvalues will dictate the response of the common-mode. As the control objective is machine control and the common-mode $q$ and $d$ currents are the machine currents, the common-mode response is of the most interest.

The inter-converter differential-modes are not directly related to a control objective, but they must be stable. Depending on how the differential-modes are excited, these modes may not require as much damping as the common modes. The difference currents, which the differential-modes are associated with, are given by:

$$i_{qk,DM} = i_q/N - i_{qk} \quad (4.22)$$

$$i_{dk,DM} = i_d/N - i_{dk}$$

Although differential-mode currents are defined for each converter in the system (in this case 2 converters) one of these currents is redundant. This was encountered already for the case of the 0-sequence currents $i_{01}$ and $i_{02}$ where $i_{02}$ was redundant. For the two-converter case, only the differential-mode currents defined with respect to converter 1 are chosen as states.
The 0-sequence state $i_{01}$ is inherently a differential-mode current. This may be shown by evaluating the 0-sequence common-mode current:

$$i_0 = \sum_k i_{0k}$$

(4.23)

The current $i_0$ is zero because there is no path for 0-sequence current into the motor. If the differential-mode definition in Eqn. (4.22) is applied to the 0-sequence current then:

$$i_{0k,DM} = \frac{i_0}{N} - i_{0k}$$

$$= 0 - i_{0k}$$

$$= -i_{0k}$$

(4.24)

thus $i_{0k}$ are differential-mode currents. For the two-converter case, $k = 1$ only.

### 4.5.2 Lunze Transform

The Lunze transform [21, 22, 23] provides an algorithmic method of separating the common- and differential-mode quantities from a state-transition matrix. Here, it is applied to $A_{CL}$. The resulting matrices allow the common- and differential-mode eigenvalues to be computed and the effect of the control on these eigenvalues can be determined.

The Lunze matrix is defined for the two-converter system as:

$$L_z = \frac{1}{2} \begin{bmatrix} 1 & 0_{1x5} & 0_{1x5} & 0 \\ 0_{5x1} & 1_{5x5} & 1_{5x5} & 0 \\ 0_{5x1} & -1_{5x5} & 1_{5x5} & 0 \\ 0 & 0_{1x5} & 0_{1x5} & 1 \end{bmatrix}$$

(4.25)

The new state is given by:

$$z = L_z x_{CL}$$

(4.26)
this gives $z = [z_c \ z_d]$ where $z_c$ consists of a speed state, the common-mode currents from Eqn. (4.21) and the common-mode integrator states. $z_d$ consists of the differential $qd$ currents and integrator states plus the 0-sequence current state.

Previously the state vector was reordered so that the machine speed $\omega_e$ was the first state and the 0-sequence current $i_{01}$ was the last. When the Lunze transform is applied, these single states are not transformed because a sum or difference is not meaningful for a single state. That the states are identical is evidenced in Eqn. (4.25) by the rows/columns with only a single 1 (e.g. the first row and column). These states, however, are inherently common- or differential-mode, as discussed earlier.

The state-transition matrix is to be partitioned into four sub-matrices:

$$A = \begin{bmatrix}
A_{CM} & A_{DM-CM} \\
A_{CM-DM} & A_{DM}
\end{bmatrix} \quad (4.27)$$

where $A_{CM}$ describes the impact of the common-mode states resulting from common-mode states, $A_{DM}$ describes the same for the differential-mode, and $A_{DM-CM}$ and $A_{CM-DM}$ describe the interactions between the two sets of states. To accomplish this partitioning, all the single common-mode states must be placed at the start of the state vector. In this case, only $\omega_e$. The differential-mode states must be associated with $A_{DM}$, which is done by placing the states at the end of the state vector. In this case, only $i_{01}$.

This format allows the common-mode eigenvalues and the differential-mode eigenvalues to be isolated and makes clear what interactions between the common- and differential-modes exist.

As an example of the benefits of this decoupling, take the motor feedback. Here, it is chosen as varying linearly with speed. However, because it is part of the common-mode which is not affected by the transform, the load characteristic affects only the common modes of the system. Thus in extending to an $N$-converter system, the particular choice of load characteristic will not impact the stability of the differential-modes.
Applying the Lunze transform to Eqn. (4.17) gives:

\[
\frac{d}{dt} z = A_z z + B_z u
\]  

(4.28)

where \( A_z = L_z A_{CL} L_z^{-1} \) and \( B_z = L_z B_{CL} \). The input \( u \) is not affected by the transform.

\( A_z \) has the form:

\[
A_z = \begin{bmatrix}
A_c & 0_{6x6} \\
A_{zd} & 0_{6x5} & A_d
\end{bmatrix}
\]  

(4.29)

\( A_c \) contains the common-mode terms and \( A_d \) the differential-mode terms. The zeros in the top-right show that the common-mode is not affected by the differential-modes at all. \( A_{zd} \), however, indicates that the differential-mode can be excited by \( \omega_e \). The common-mode \( A_c \) is given as:

\[
A_c = \begin{bmatrix}
-\frac{pk_j}{2J} & \frac{3p^2 L_m}{8J} & 0 & 0 & 0 & 0 \\
A_{c21} & -\frac{R_1+2R+k_q}{L_1+2L_q} & -\frac{L_1+2L_d-L_{cd}}{L_1+2L_q} \omega_e & -\frac{k_q k_\omega a_{\omega}}{(1+k_\omega D_\omega)(L_1+2L_q)} & \frac{k_q a_q}{L_1+2L_q} & 0 \\
A_{c31} & \frac{L_1+2L_q-L_{cd}}{L_1+2L_d} \omega_e & -\frac{R_1+2R+k_d}{L_1+2L_d} & 0 & 0 & \frac{k_d a_d}{L_1+2L_d} \\
\frac{-2}{1+k_\omega D_\omega} & 0 & 0 & -\frac{k_d a_d}{1+k_\omega D_\omega} & 0 & 0 \\
\frac{-2k_q}{1+k_\omega D_\omega} & -1 & 0 & \frac{k_q a_q}{1+k_\omega D_\omega} & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0
\end{bmatrix}
\]  

(4.30)

with

\[
A_{c21} = -\frac{2k_q k_\omega}{(1+k_\omega D_\omega)(L_1+2L_q)} - \frac{L_1+2L_d-L_{cd}}{L_1+2L_q} \left( \overline{i_{d1}} + \overline{i_{d2}} \right) - \frac{2\lambda'_m}{L_1+2L_q}
\]

\[
A_{c31} = -\frac{L_1+2L_q+L_{cq}}{L_1+2L_d} \left( \overline{i_{q1}} + \overline{i_{q2}} \right)
\]
and the differential-mode matrices are:

\[
A_d = \begin{bmatrix}
-\frac{R_1+R_q}{L_1} & -\frac{L_1-L_{eq}}{L_1} & \frac{k_qk_oa_o}{(1+k_oD_o) L_1} & \frac{k_oa_q}{L_1} & 0 & 0 \\
\frac{L_1-L_{eq}}{L_1} & -\frac{R_1+R_d}{L_1} & 0 & 0 & \frac{k_oa_d}{L_1} & 0 \\
0 & 0 & -\frac{k_oa_oD_o}{1+k_oD_o} & 0 & 0 & 0 \\
-1 & 0 & -\frac{k_oa_d}{1+k_oD_o} & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{2R_1+R_q}{L_{on}}
\end{bmatrix}
\]

(4.31)

and

\[
A_{\omega d} = \begin{bmatrix}
\frac{L_1-L_{eq}}{L_1} (\bar{i}_{d1} - \bar{i}_{d2}) & \frac{L_1+L_{eq}}{L_1} (\bar{i}_{q1} - \bar{i}_{q2}) & 0 & 0 & 0 & 0 \\
\end{bmatrix}^T
\]

(4.32)

### 4.5.4 Eigenvalue Analysis

In order to evaluate the eigenvalues of these matrices, control parameters and an operating point were chosen. These are given in Table 4.1.

If converters with different ratings are used, the droop \(D_o\) can be assigned in proportion to the converter rating. This addresses the requirement for steady-state load sharing. The gains associated with the speed and current control loops can probably be adjusted to ensure dynamic load sharing for the case of unequal converters, but further research on this issue is required to determine what the gains should be.

The eigenvalues associated with the common-mode state transition matrix \(A_c\) are shown in Fig. 4.5 as speed increases from 0 to its maximum. All of the eigenvalues are stable, though as motor speed increases the system displays poorer damping.

The differential-mode eigenvalues are shown in Fig. 4.6, again as speed varies from 0 to maximum. All of the eigenvalues are stable. Among the differential-mode eigenvalues, two are unaffected by the change in the motor’s operating point (speed). The eigenvalue at -234 is associated with the 0-sequence current, and the eigenvalue at -0.9 is associated with the differential \(x_s\) state (associated with the integrators for the speed-control loop).
### Machine Loading

| $k_L$ [N·m/(electrical rad./s)] | 0.05 |

### Speed Controller

| $k_\omega$ [A/(electrical rad./s)] | 0.3 |
| $a_\omega$ [s] | 10 |
| $D_\omega$ [(electrical rad.)/s/A] | 0.33 |

### q- and d-axis Controllers

| $k_q,k_d$ [V/A] | 2 |
| $a_q,a_d$ [1/s] | 75 |
| $L_{cq}$ [mH] | 6.0 |
| $L_{cd}$ [mH] | 3.3 |

### 0-Sequence Controller

| $k_0$ [V/A] | 2 |

### Operating Point

| $i_{qk}$ [A] | 5 |
| $i_{dk}$ [A] | 0 |

Table 4.1: Controller gains and operating points
Based on \( A_{\omega d} \), the differential-mode is only excited when there exists a difference in operating point (\( i_{d1} \neq i_{d2} \)) between converter 1 and converter 2 and the speed changes. Thus, for normal operation when the converters are sharing the load equally (assuming their droops \( D_{\omega} \) are equal), the differential-mode eigenvalues are not excited.

In order to cancel the cross-coupling terms associated with \( A_c \) (\( A_{c23} \) and \( A_{c32} \)), the parameters \( L_{cq} \) and \( L_{cd} \) should be set as \( L_{cd} = L_1 + 2L_d \) and \( L_{cq} = L_1 + 2L_q \). If these values are chosen for \( L_{cq} \) and \( L_{cd} \), however, the cross-coupling terms in the differential-mode matrix \( A_d \) (\( A_{d21} \) and \( A_{d12} \)) become significantly larger.

For the two-converter case the system remains stable, but this is an example where optimizing the common-mode response is detrimental to the differential-mode stability. If more converters were present, it might not be possible to cancel the cross-coupling
terms effectively using this control strategy. Additional research would be required to determine whether a suitable control can be found which would allow the common modes to be optimized while not adversely affecting the differential modes.
Chapter 5

Results

5.1 Overview

This chapter validates the simulation model by comparing simulation and experimental results. The system stability demonstrated in these results also supports the validity of the eigenvalue analysis.

First, the machine model is validated against the experimental PMSM in Section 5.2. Section 5.3 presents results for two different speed steps in order to confirm that the converters share the load dynamically and in steady-state. This also demonstrates system stability. In Section 5.4 demonstrates the system redundancy associated with the two-converter arrangement.

5.2 Machine Model Validation

5.2.1 Open-Loop Response

The open-loop response was obtained using a single converter and applying steps to $v_{tq1}$ and $v_{td1}$, the $q$- and $d$-axes voltages. Setting the voltages determines the converter’s switching times directly.
These results are shown in Figs. 5.1a and 5.1b for the $q$-axis and in Figs. 5.2a and 5.2b for the $d$-axis.

Due to the non-linear load characteristic, which was not modelled in detail, some discrepancy exists between the experimental and simulation result for the $v_q$ test, but overall the agreement is acceptable. The transient amplitudes are similar, as are the waveform shapes of both $i_q$ and $i_d$ for both tests.

A ripple-current is present on the experimental results for $i_q$ due to the trapezoidal windings of the PMSM. For higher speed operation, the frequency of this ripple would be increased and the system inductance provides more attenuation. For the $i_d$ test no ripple is present as the rotor is not turning.

There is an offset in the experimental results due to the voltage loss in the converter. This is particularly noticeable in the $i_d$ test due to the low voltage being requested, but the machine dynamics—indicated by the time constant and magnitude of the current change—are in good agreement.
Figure 5.1: (a) Simulation and (b) experimental results for $v_q$ step from 15 to 20 V. $\omega_e$ is [20 rad/s/div] and $i_q$ and $i_d$ are [5 A/div]. Experimental time is in ms.
Figure 5.2: (a) Simulation and (b) experimental results for a 5 V step in $v_d$. $\omega_e$ is [20 rad/s/div] and $i_q$ and $i_d$ are [10 A/div]. Experimental time is in ms.
5.2.2 Closed-Loop Response

The closed-loop response is obtained by applying PI-controllers to the $q$-and $d$-axis currents. The $q$-axis current is stepped from 0.74 to 15.94 A (64% of rated), corresponding to a step in $T_e$ from 0 to 20 N·m, while the $d$-axis current reference is fixed at 0 A. The results are shown in Figs. 5.3a and 5.3b.

The simulation and experimental results are very similar, though there is some dynamic coupling between the $q$- and $d$-axes in the experimental setup which is not present in the simulation. The rise and fall times are also similar, though the experimental controller is slightly faster.
Figure 5.3: (a) Simulation and (b) experimental results for $i_{q_{ref}}$ step from 0.74 to 15.94 A. $\omega_e$ is [20 rad/s/div] and $i_q$ and $i_d$ are [5 A/div]. Experimental time is in ms.
5.3 System Response to Speed Steps

In order to validate the effectiveness of the control, two steps in the speed reference $\omega^*_e$ were applied. The response of the $qd$ and 0-sequence currents and the machine speed was observed.

For each test, a series of 16 graphs, split into 4 figures, is shown. The first 8 graphs show experimental results and the last 8 graphs replicate the first 8, but the simulation results are shown. The first figure has 5 graphs: the first graph shows the applied speed step and the motor speed and the next 4 graphs show the $qd$ and 0-sequence currents for both converters. The second figure, showing the sixth, seventh and eighth graphs, shows the common- and differential-mode currents associated with the $qd$-axes and the differential-mode 0-sequence current. These are derived from the current states described in Eqn. (4.26).

The current references $i^*_q$ were limited to 15 A for all tests, providing a nominal maximum current of 30 A.

5.3.1 Low Speed Test

Figs. 5.4-5.7 show the low speed, moderate torque test. Current sharing between the converters is excellent (first graph, Fig. 5.7), and at this low speed there is very little cross-coupling between the $q$- and $d$-axes. The 0-sequence current is well controlled.

In the experimental setup, the speed step (first graph, Fig. 5.4) causes the speed controller to saturate $i^*_{q1,2}$ approximately 15 ms after the speed step is applied, and the references remain in saturation until the motor speed reaches the new $\omega^*_e$. The speed overshoot is approximately 10% of the final value. Steady-state current changes from approximately 4 A before the step to 8 A following the step (second and third graphs). This corresponds to a load change from approximately 1/4- to 1/2-rated.

The ripple on the quadrature currents (second and third graph in Fig. 5.4) is due to
the machine's trapezoidal windings.

Both experimentally and in simulation, the two converters respond essentially identically, excepting the switching ripple, and the cross-coupling in the simulation is very similar to the experimental case. The agreement between the simulation and experiment is excellent.
Figure 5.4: Experimental: Response of VSC1 & VSC2 to a step in speed reference at \( t = 0.0 \) s from 62 to 124 rad/s (electrical).
Figure 5.5: Experimental: Difference in Response Between VSC1 & VSC2 to a step in speed reference at \( t = 0.0 \) s from 62 to 124 rad/s (electrical).
Figure 5.6: Simulation: Response of VSC1 & VSC2 to a step in speed reference at $t = 0.1$ s from 62 to 124 rad/s (electrical).
Figure 5.7: Simulation: Difference in Response Between VSC1 & VSC2 to a step in speed reference at $t = 0.1$ s from 62 to 124 rad/s (electrical).
5.3.2 High Speed Test

Figures 5.8 to 5.11 show a speed step of 10\% up to 2/3-rated speed (1133 rpm to 1333 rpm). The load is about 1/3-rated. The high- and low-speed tests are very similar. For the high-speed case, the lighter load causes the speed controller’s response to be faster, the overshoot to be slightly higher, and the saturation in the current to last for a shorter time.

The most salient difference is the larger cross-coupling, where the spike on $i_d$ reaches approximately 50\% of the spike in $i_q$. The large cross-coupling partly results from the choice of $L_{cq}$ and $L_{cd}$, which were set to the machine inductances rather than the optimal values determined from the eigenvalue analysis. The parallel converter arrangement means an extra cross-coupling term, resulting from the other converter’s currents, is present. The other converter’s currents are not measured, so this cannot be perfectly cancelled. At low speed, this effect is negligible.

In Fig. 5.9, the currents $i_{q1}$ and $i_{q2}$ are marginally different, resulting in a non-zero $\Delta i_q$. This is caused by one of the controllers measuring a different speed than the other, due to dropped encoder-counts. The droop on the speed reference causes the converters to supply different $i_q$ currents, which is the desired operation.

In simulation, there is no significant difference between the converters’ response. For the experimental case, the interface component variation account for the small difference in response between converters ($\Delta i_q$ and $\Delta i_d$ on the first and second graph, Fig. 5.9).

Overall, the performance of the simulated system closely matched that of the experimental system. The controllers caused the converters to share the load equally and the system’s response was stable.
Figure 5.8: Experimental: Response of VSC1 & VSC2 to a step in speed reference at $t = 0.0$ s from 356 to 419 rad/s (electrical).
Figure 5.9: Experimental: Difference in Response Between VSC1 & VSC2 to a step in speed reference at $t = 0.0$ s from 356 to 419 rad/s (electrical).
Figure 5.10: Simulation: Response of VSC1 & VSC2 to a step in speed reference at $t = 0.1$ s from 356 to 419 rad/s (electrical).
Figure 5.11: Simulation: Difference in Response Between VSC1 & VSC2 to a step in speed reference at $t = 0.1$ s from 356 to 419 rad/s (electrical).
Chapter 5. Results

5.4 Loss of Converter

This section demonstrates the ability of the system to continue operating despite the loss of one of the converters. In Fig. 5.12, the experimental system starts in steady-state at approximately 1/3-rated speed. At $t = 0.2$ converter 2’s gating was shut off to emulate the loss of that controller.

Fig. 5.13 shows a simulation of the same case. The response is almost identical to the response of the laboratory system, indicating that the simulation results are reliable for this type of test.

As a result of the loss of gating, a speed dip of less than 10% occurs. The motor settles to a new steady-state speed in approximately 800 ms. This new speed is slightly less than the original speed as a result of the droop $D_\omega$. The plots of the quadrature currents show that until converter 2 lost its gating signals the load was shared equally. At that point, converter 2 no longer provides $q$-axis current and converter 1 begins to increase $i_{q1}$ to compensate, as the same net $i_q$ is required to maintain the motor’s speed. Also note that at $t = 0.2$ s. when converter 2 is lost, converter 1’s current immediately jumps to the net $i_q$ as a result of the requirement for continuity in the currents resulting from the machine inductance. The $q$-axis controller quickly returns $i_{q1}$ to the pre-transient state, which causes cross-coupling with the $d$-axis, resulting in a transient in $d$-axis current also.

The increase in $i_{q1}$ is driven by speed control loop whose error signal is the difference between the motor speed and its reference. As the motor dynamics are slow, this compensator also reacts slowly.

The 0-sequence current is initially regulated to zero and the loss of one of the converters does not affect its regulation, although the 0-sequence switching ripple is reduced because the applied voltage $v_0$ is smaller. Note that a closed-loop path for 0-sequence current still exists through the disabled converter 2.
Figure 5.12: Experimental test of the loss of gating signals for converter 2: from steady-state, at $t = 0.2$ seconds. Shown are the speed and $qd0$ currents for both converters.
Figure 5.13: Simulation of the loss of gating signals for converter 2: from steady-state, at \( t = 0.2 \) seconds. Shown are the speed and \( qd0 \) currents for both converters.
Chapter 6

Conclusions

A parallel motor drive consisting of two converters directly connected to a common dc source and a single PMSM was developed. The converters are controlled independently without converter inter-communication and do not depend on a master/slave arrangement.

A laboratory model of the system was developed. The system is described with a particular emphasis on the implications of the transition to the two-converter drive from the single-converter drive. The 0-sequence switching ripple was characterized and a new model describing the circulating 0-sequence current was identified.

A new linearized state-space model of the system was developed using the rotating $qd0$ reference frame aligned with the PMSM’s rotor. This mathematical formulation is well-suited to the study of multi-converter motor drives.

The transformation also resulted in the 0-sequence components being decoupled from the rest of the system. The $qd$ currents are coupled through the motor and each converter is affected by the other converter’s control voltages.

To ensure steady-state load sharing, a new drooped PI speed control scheme controls the converters’ $q$-axis current references. The control exploits the decoupled nature of the $qd$ and 0-sequence quantities and uses PI controllers to control $qd$ current loops. The
controllers do not communicate and only depend on the machine position and the locally available current and voltage information. The stability of the closed-loop system was confirmed using an eigenvalue analysis.

The system was simulated in MATLAB/Simulink and the simulation was validated against the laboratory model by comparing the open- and closed-loop responses and applying a step in speed. The usefulness of the parallel converter arrangement was demonstrated through a test wherein one of the converters stopped gating and the system continued to operate.

Future research in the area of parallel motor drives should focus on the extension of the two converter system presented here to the case of $N$ converters. This will likely necessitate the investigation of a new control method capable of stabilizing the larger system and holds significant potential for providing more cost-effective and reliable drive structures.

Another area for future work is to run the system sensorless (without the encoder), possibly using the voltage at the PCC to determine the machine position.
Bibliography


converter fed microgrids,” *To be published in IEEE Transactions on Power Systems*.


[10] S. Mazumder, “Continuous and discrete variable-structure controls for parallel three-

modulated inverter with parallel connected transistors using current-sharing reac-
tors,” *Power Electronics, IEEE Transactions on*, vol. 8, no. 2, pp. 186–191, Apr
1993.


[13] B. Shi and G. Venkataramanan, “Parallel operation of voltage source inverters with


control of a multiconverter consisting of a parallel transformerless connection of


Appendix A

Motor Details

Most machine parameters are listed in Fig. A.1, however some tests indicated discrepancies between the datasheet and actual values.

Magnet Strength

The strength of the magnetic field, denoted $\lambda'_m$ [V/s] with the $'$ indicating a quantity referred to the stator side, can be determined in several ways. From the datasheet rated values and (3.11):

$$\lambda'_m = \frac{4T_e}{3P_i q_s}$$ (A.1)

which gives $\lambda'_m = 0.281$. Using the same relation and measuring the torque gave $\lambda'_m = 0.278$.

Alternatively, based on (3.9): if $\omega_e$ is large then setting $i_d = 0$ using $v_{dref} = 0$, measuring $i_q$, and applying a known $v_q$ allows $\lambda'_m$ to be determined according to:

$$\lambda'_m = \frac{v_q - R_s i_q - L_q \frac{di_q}{dt}}{\omega_e}$$ (A.2)

In all cases the result was close to the datasheet value.
Appendix A. Motor Details

Resistance

The system resistances were measured with the laboratory milliohm meter. The machine’s stator resistance was measured as 0.354 $\Omega_{i-1}$, giving $R = 0.177 \Omega$. The line resistance from converter terminal to motor terminal was measured at about 0.1 $\Omega$.

Inductance

As only one value for inductance is given on the motor’s datasheet: 16.3 $mH_{i-1}$, the machine was assumed to have a non-salient rotor (with surface-mounted magnets). However, after determining that the cross-coupling terms did not correspond well with the predicted values, additional tests to determine the real machine inductance were performed. As there seemed to be rotor saliency (that is, $L_q \neq L_d$, normally because the magnets are embedded in the rotor, resulting in a change in the cross-section of the magnetic material), tests were performed along both the $q$- and $d$-axes. The first method was to apply a step on the $q$-axis voltage under a locked-rotor condition. The second method involved operating at high speed and determining the inductance to eliminate the cross-coupling terms.

For the first method, a large clamp was attached to the shaft to physically lock the rotor, setting $\omega_e = 0$. As a result, any changes are associated with the electrical circuit. By applying a voltage $v_d$ and measuring the slope of the current, the inductance can be approximated based on the voltage applied and the approximate slope of the current at $t = 0$ (the instant when voltage is applied):

$$L_d \approx v_d \frac{\Delta i(0)}{\Delta t(0)}$$

The same process can be repeated to determine $L_q$.

The second method involves operating with $\omega$ large and varying $L_d$ ($L_q$) to cancel the cross-coupling terms.
<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Channel A Output</td>
</tr>
<tr>
<td>B</td>
<td>Channel B Output</td>
</tr>
<tr>
<td>C</td>
<td>Channel C Output</td>
</tr>
<tr>
<td>D</td>
<td>DC+ Input</td>
</tr>
<tr>
<td>E</td>
<td>— (internal connection to D)</td>
</tr>
<tr>
<td>F</td>
<td>DC Return</td>
</tr>
<tr>
<td>G</td>
<td>Case Ground</td>
</tr>
<tr>
<td>H</td>
<td>Channel (\bar{A}) Output</td>
</tr>
<tr>
<td>I</td>
<td>Channel (\bar{B}) Output</td>
</tr>
<tr>
<td>J</td>
<td>Channel Z Output</td>
</tr>
</tbody>
</table>

Table A.1: 10-pin encoder pinout

Averaged together, these results gave the true machine inductances: \(L_d = 3.3 \, mH\) and \(L_q = 6.0 \, mH\).

**Encoder**

A 4000-pulse encoder is fixed to the rotor of the PMSM. No information specific to this particular encoder was available, but the encoder uses the standard 10-pin layout of Table A.1[24].

The encoder should be supplied with approximately 10 V. Channels A and B are the counters while channel Z is the reset pulse. The encoder output was extremely noisy and a large common-mode filter was placed on the encoder’s channel Z output.

At high speeds, the encoder’s reset pulse is small enough that VSC2’s controller was unable to consistently recognize the reset pulse, rendering the speed measurement unreliable. Thus, no high speed tests were performed because controller 2 could not
operate.

Controller 2 was unable to read positive $\omega_e$, so only positive-speed tests were performed in the laboratory. This is likely an FPGA issue.
### Kollmorgen GOLDLINE

#### 80x PERFORMANCE DATA

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Symbols</th>
<th>Units</th>
<th>B-802-A</th>
<th>B-802-B</th>
<th>B-802-C</th>
<th>B-804-A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horsepower</td>
<td>HP</td>
<td>HP</td>
<td>10.6</td>
<td>13.6</td>
<td>8.2</td>
<td>14.5</td>
</tr>
<tr>
<td>Kilowatts</td>
<td>kW</td>
<td>kW</td>
<td>7.9</td>
<td>10.1</td>
<td>6.1</td>
<td>10.8</td>
</tr>
<tr>
<td>Speed at Rated Power</td>
<td>N Rated</td>
<td>RPM</td>
<td>2000</td>
<td>2750</td>
<td>1600</td>
<td>1500</td>
</tr>
<tr>
<td>Max Operating Speed</td>
<td>N Max</td>
<td>RPM</td>
<td>2000</td>
<td>2750</td>
<td>1600</td>
<td>1500</td>
</tr>
<tr>
<td>Continuous Torque (stall) at 40°C</td>
<td>Tc</td>
<td>lb-ft</td>
<td>31.0</td>
<td>30.0</td>
<td>30.0</td>
<td>58.0</td>
</tr>
<tr>
<td>Continuous Torque (stall) at 25°C</td>
<td>Tc</td>
<td>lb-ft</td>
<td>32.9</td>
<td>31.8</td>
<td>31.8</td>
<td>61.5</td>
</tr>
<tr>
<td>Continuous Line Current</td>
<td>Ic</td>
<td>Amps RMS</td>
<td>24.9</td>
<td>32.4</td>
<td>18.9</td>
<td>35.0</td>
</tr>
<tr>
<td>Peak Torque</td>
<td>Tp</td>
<td>lb-ft</td>
<td>96.0</td>
<td>95.3</td>
<td>95.3</td>
<td>171.0</td>
</tr>
<tr>
<td>Peak Line Current</td>
<td>Ip</td>
<td>Amps RMS</td>
<td>130.0</td>
<td>129.0</td>
<td>129.0</td>
<td>232.0</td>
</tr>
<tr>
<td>Max Theoretical Acceleration</td>
<td>Z</td>
<td>rad/ sec²</td>
<td>26700</td>
<td>26500</td>
<td>26500</td>
<td>27600</td>
</tr>
<tr>
<td>Back EMF (line-to-line) ±10%</td>
<td>Kb</td>
<td>VRMS/kRPM</td>
<td>102.0</td>
<td>76.0</td>
<td>130.0</td>
<td>136.0</td>
</tr>
<tr>
<td>Max line-to-line Volts</td>
<td>VMax</td>
<td>Volts RMS</td>
<td>250.0</td>
<td>250.0</td>
<td>250.0</td>
<td>250.0</td>
</tr>
<tr>
<td>DC Res at 25°C (line-to-line) ±10%</td>
<td>Rm</td>
<td>Ohms</td>
<td>0.361</td>
<td>0.200</td>
<td>0.568</td>
<td>0.230</td>
</tr>
<tr>
<td>Inductance (line-to-line) ±30%</td>
<td>Lm</td>
<td>mH</td>
<td>16.3</td>
<td>9.4</td>
<td>25.7</td>
<td>13.0</td>
</tr>
<tr>
<td>Rotor Inertia (B, BE, EB-80x)</td>
<td>Jn</td>
<td>lb-ft-sec²</td>
<td>0.00360</td>
<td>0.00360</td>
<td>0.00360</td>
<td>0.00620</td>
</tr>
<tr>
<td>(M, ME-80x)</td>
<td></td>
<td>kg-m²</td>
<td>0.00488</td>
<td>0.00488</td>
<td>0.00488</td>
<td>0.00840</td>
</tr>
<tr>
<td>Weight (B, BE-80x)</td>
<td>Wt</td>
<td>lb</td>
<td>79</td>
<td>79</td>
<td>79</td>
<td>112</td>
</tr>
<tr>
<td>(M, ME-80x)</td>
<td>Wt</td>
<td>kg</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>50.6</td>
</tr>
<tr>
<td>Static Friction</td>
<td>Tf</td>
<td>lb-ft</td>
<td>0.47</td>
<td>0.47</td>
<td>0.47</td>
<td>0.67</td>
</tr>
<tr>
<td>Thermal Time Constant (B, BE, ME-80x)</td>
<td>TCT</td>
<td>Min</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>48</td>
</tr>
<tr>
<td>Thermal Time Constant (EB-80x)</td>
<td>TCT</td>
<td>Min</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>70</td>
</tr>
<tr>
<td>Viscous Damping Z Source</td>
<td>Fi</td>
<td>lb-ft/kRPM</td>
<td>0.175</td>
<td>0.175</td>
<td>0.175</td>
<td>0.221</td>
</tr>
<tr>
<td>Motor Constant at 25°C</td>
<td>Km</td>
<td>lb-ft/√W</td>
<td>1.79</td>
<td>1.79</td>
<td>1.82</td>
<td>3.00</td>
</tr>
<tr>
<td>Thermal Resistance at stall</td>
<td>Rth</td>
<td>°C/Watt</td>
<td>0.23</td>
<td>0.24</td>
<td>0.25</td>
<td>0.18</td>
</tr>
<tr>
<td>Number of Poles</td>
<td></td>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

Figure A.1: Motor Information for Kollmorgen Goldline M-803-A.
Appendix B

Implementation Issues

B.1 Common-Mode Signal Filters

In the experimental setup when both converters are switching, a significant amount of noise is present on the input and output channels for converter 2’s controller. This noise is not present on converter 1’s controller, and seems to form a closed-loop through the computer. The anti-aliasing filter and input filters at the computer’s terminal block, each of which is designed to attenuate only differential-mode signals, has little or no effect on this signal, suggesting that this is a common-mode problem. The input channels are all differential-mode inputs, so it is unclear why the common-mode is causing such a problem, and the source for this noise is not known.

To mitigate the noise, a common-mode choke was added to each of the input and output signals. As the desired signal is entirely differential-mode, this should have a negligible effect on the phase and magnitude of the measured signal, but it provides very high-impedance to the problematic common-mode noise. The chokes were constructed from magnetic toroids and consist of approximately 20 turns each.
B.2 Simulation Stability

In order to preserve system stability during simulation, resistors are required in series with the EMI capacitors at the PCC. An acceptable value is 10 Ω, but a smaller value such as 1 Ω results in extremely large currents occurring in simulation. The value should be as small as possible as the EMI capacitors have an impedance of about 5 Ω at the switching frequency.

B.3 Time-Step Size and Solver Choice

In order to attain accurate results for the common-mode (0-sequence) current in particular, a small time-step is required. For most simulations, 2 µs characterizes the process well, but the dynamics of the common-mode current does not exhibit periodicity and should not be considered reliable. To get an accurate simulation of the common-mode current, a time-step of 1 µs, or possibly even smaller, must be used.

The solver chosen is a stiff, fixed time-step solver: ode2. Variable time-step solvers tended to take too long and require too small of a minimum time-step.
Appendix C

RT-Linux Control Code

// description:
// This controller has been developed to control the PMSM. The code
// was originally modified by Dale Dolan with the sole goal to act as a
torque controller (using current reference only).

// Modifications were made in the summer & fall of 2008 by Danny Fingas
// in order to test the validity of a MATLAB model of a system
// incorporating the PMSM. Code specific to the torque sensor was
// removed.

// Original (c) 2003 Hung Shyu

// rtce_boolean_param gating_on false "Gating on. Choose ONLY ONE OF:"
// rtce_line
// rtce_boolean_param classical false "Open-Loop Sinusoidal Voltage Source"
// rtce_boolean_param voltage_control false "q-d Voltage Control"
// rtce_boolean_param torque_control false "Manual Torque and Id References"
// rtce_boolean_param speed_control true "Use Drooped Speed Control to set Iqref"

// rtce_line
// rtce_scalar_meter Thetadisp "%0.2f deg." "Rotor Angle (theta)"
// rtce_scalar_meter [deg."]
// rtce_scalar_meter avg_elec_speed "%0.2f rad" "Average Elec. Speed [r/s]"
// rtce_scalar_meter w_filtereddisp "%0.2f rad" "elec. freq. (filtered) [rads]"
// rtce_scalar_meter [rads]"
// rtce_scalar_meter w_unfiltereddisp "%0.2f rad" "elec. freq. [rads]"
// rtce_line
// rtce_scalar_meter v_DCdisp "%0.1f V dc" "dc-link Voltage"
// rtce_scalar_meter Vlimitdisp "%0.2f " "Vlimit"
// rtce_scalar_meter Vlength_corrected "%0.2f " "
// rtce_scalar_meter Vlengthdisp "%0.2f " "Vlength"
// rtce_scalar_meter Iqrefdisp "%0.2f " "Iqref (if applicable)"

// rtce_new_column

// rtce_scalar_slider LPCO 40.0 0.1 100.0 "LPF freq. [Hz] for speed"
// rtce_scalar_param angle_delta_ref 42.0 −180 +179 "Angle [Degree]"
// rtce_scalar_param lambda_mag 0.272 −1.0 1.0 "lambda mag"
// rtce_scalar_param Lq 0.0040 −0.015 0.015 "Lq"
// rtce_scalar_param Ld 0.0033 −0.015 0.015 "Ld"

// rtce_line
// rtce_scalar_slider frequency 0 0 6 "Open–loop Volt. Freq [Hz]"
// rtce_scalar_slider ma 0 0 1.0 "Open–loop ma"
// rtce_scalar_slider adjust_angle 0.000 −179.000 180.000 "adjust angle [degrees]"
// rtce_line
// rtce_scalar_param Vqref 0.0 −130.0 130.0 "Vqref (q–d Voltage Control)"
// rtce_scalar_param Vdref 0.0 −130.0 130.0 "Vdref (q–d Voltage Control)"
// rtce_line
// rtce_scalar_param Tref 0.0 −20.0 40.000 "Tq Ref. [N*m] (Manual Ref.)"
// rtce_scalar_param Idrefmanual 0.0 −30.0 30.000 "Idref (Manual Reference)"

// rtce_new_column
// rtce_scalar_param wrefSet −157.1 −628.3 628.3 "Speed Reference [radians]"
// rtce_scalar_meter wrefDisplay "%0.2f radians" "Tracked Speed"
// rtce_scalar_param operationalChangeDelta 20.0 −600 600 "Step Size for Op. Change"
// rtce_scalar_param I0refset 0.0 −20.0 20.0 "Zero–Sequence Current Reference [A]"
// rtce_scalar_param Idrefset 0.0 −20.0 20.0 "D–Axis Current Reference [A]"
// rtce_scalar_slider Kpq 2.0 0.0 10.0 "q–axis Proportional Gain Kpq"
// rtce_scalar_slider Kiq 150.0 0.0 600.0 "q–axis Integral Gain Kiq"
// rtce_scalar_slider Kpd 2.0 0.0 10.0 "d–axis Proportional Gain Kpd"
// rtce_scalar_slider Kid 150.0 0.0 600.0 "d–axis Integral Gain Kid"
// rtce_scalar_slider Kpzero 2.0 0.0 10.0 "Zero–Sequence Proportional Gain Kpzero"
// rtce_scalar_slider Kpw 0.3 0.01 2.0 "Speed Loop Proportional Gain Kpw"
// rtce_scalar_slider aw 10.0 0.1 100.0 "Speed Loop Zero"
Appendix C. RT-Linux Control Code

Placement aw”

// r t c e _ s c a l a r _ s l i d e r D w 0. 3 3 0. 0 2. 0 " S p e e d D r oo p "

#define _V E R B O S E // for output of PRINTF
#include "rtce_module.h"
#include "rtce_control.h"
#include "acromag.h"
#include "../fpga/pwm.h"

// include automatically created parameter definitions
#include "parser_output.h"

// get the shared memory support for parameter exchange
#include "rtce_shared_mem_param.h"
#include "rt_math.h"
#include "math_tools.h"
#include "control_tools.h"

static float signal [64];
#define SIGNAL(i) signal[i]

double second_order_lpf(double x, double *x0, double *x1,
 double *y0, double *y1, double Q, double wc, double Tsample);

double first_order_lpf(double x, double *x0, double *y0, double wc,
 double Tsample);

static int rtce_create_sharedmem(void) {
  int result = rtce_create_sharedmem_for_this_number_of_items(
    RTCE_NUM_SCALAR_PARAMS,
    RTCE_NUM_BOOLEAN_PARAMS,
    RTCE_NUM_SCALAR_METERS,
    RTCE_NUM_SCALAR_SLIDERS);
  // only the control-module may reset the select-index on startup,
  // if the userinterface resets this index on startup,
  // it changes the parameter that might be used by a running
  controller
  if ((result == 0) && (RTCE_NUM_CONFIRM_PARAMS != 0))
    *rtce_ctrl_memory_block_select_index = 0;
  return result;
}

// module section

// this controller can be built as a module
// in this case only this file needs to be recompiled

#ifndef MODULE
// ATTENTION: no floating point in these functions !!!!
int init_module(void) {
}
PRINTF("init_module - begin\n");
PRINTF(" $Id: ECE315 Lab-2, Hung Shyu $\n");
rte_create_sharedmem();
rte_pwm_set_ts_for_pwm_and_AD(200);
PRINTF("init_module - end\n");
return 0;
}

void cleanup_module(void) {
    PRINTF("cleanup_module - begin\n");
rte_reset_control();
    PRINTF("cleanup_module - end\n");
}

#endif

// CONTROL FUNCTIONS
// is started when the thread is created
// floating point and all symbol support

hrtimer_t previous_t;
hrtimer_t current_t;
static hrtimer_t t_ns_start; // time in ns between bootup and controller

void rte_init_control(void) {
    PRINTF("init_control - begin\n");
    {
        int i;
        for (i=0; i<32; i++) ROUTE(i+ 0) = &SIGNAL(i);
        for (i=0; i<16; i++) ROUTE(i+32) = &IN(i);
        for (i=0; i<16; i++) ROUTE(i+48) = &OUT(i);
        for (i=0; i<64; i++) rbi->route[i] = i;
    }
    rtce_route = &(rbi->route[0]);
    t_ns_start = gethrtimer();
    PRINTF("init_control - end\n");
    return;
}

// reset global variables if necessary ...
void rte_reset_control(void) {
}

// stop the controller if it has it's own timing
void rte_stop_control(void) {
    rtce_pwm_stop();
}

// calc the time that has elapses since t_ns and show it
static void __inline__ update_execution_time_meter(hrtimer_t t_ns) {
    hrtimer_t t_delta;
// that's approx. the time the control loop needs
// to execute in ns:
t_delta = gethrtime() - t_ns_start - t_ns;
}

static void __inline__ gating_main_switch(char switch_var) {
// old_gating_on = TRUE on init ensures that
// the gating on controller startup can not be on by default
static char old_gating_on = TRUE;

if (old_gating_on != switch_var) {
    old_gating_on = switch_var;
    if (switch_var)
        rtce_pwm_start();
    else
        rtce_pwm_stop();
}
}

void rtce_run_control(void) {
// DECLARE GLOBAL VARIABLES
hrtimer t t_ns; //, t_delta1=0; // Required??

RTCE_SCALAR_TYPE t;

// RTCE_SCALAR_TYPE Ia, Ib, Ic;
// RTCE_SCALAR_TYPE I_alpha, I_beta, Vt_alpha, Vt_beta;
// RTCE_SCALAR_TYPE Va, Vb, Vc, V_0, V_DC_unfiltered;
// RTCE_SCALAR_TYPE in_dk = 0, out_dk = 0, in_qk = 0, out_qk = 0, in_0k
// = 0, out_0k = 0;
// RTCE_SCALAR_TYPE in_wk = 0, out_wk = 0;
// RTCE_SCALAR_TYPE uq = 0, ud = 0, u0 = 0;
// RTCE_SCALAR_TYPE Idref = 0.0, Id0ref = 0.0;
// RTCE_SCALAR_TYPE L_0, L_d, L_q, Vtd = 0, Vtq = 0, Vt0 = 0;
// RTCE_SCALAR_TYPE Vta, Vtb, Vtc;

float duty_cycle_new_a, duty_cycle_new_b, duty_cycle_new_c;

int temp;
double Tsample;
double Ia, Ib, Ic, Vta, Vtb, Vtc;
// double Va, Vb, Vc;
double operationalChange;
double v_DC, v_DC_unfiltered = 0, Vlimit = 0, Vlength = 0;
double w_filtered, w_unfiltered, Theta, wref = 0;
static double Iqref = 0, Idref = 0, I0ref = 0;
// double I_alpha, I_beta, Vt_alpha, Vt_beta, V_0;
double in_dk = 0, out_dk = 0, in_qk = 0, out_qk = 0;
double in_0k = 0, out_0k = 0; // Kizero;
double in_wk = 0, out_wk = 0, out_qwk = 0, out_pwk = 0, Kiw;
double uq = 0, ud = 0, u0 = 0;
double L_0, L_d, L_q, Vtd = 0, Vtq = 0, Vt0 = 0;
// int smallest, middle, largest, tmp;
int slot_count, delta_slot_count, estimSlotCount, curInstSpeed;
int orig_slotcount, orig_slotcount0; // temporary
int i;
static float rotor_angle_m = 0.0, rotor_angle_e = 0.0;
static float sinusoidal_ref_angle = 0.0;
static int storedDeltaSlotCount[5] = {0,0,0,0,0}, errorCount = 0;
static double recentAvg = 0.0;
static double in_qk1 = 0, out_qk1 = 0;
static double in_dk1 = 0, out_dk1 = 0;
static double in_0k1 = 0, out_0k1 = 0;
static double in_wk1 = 0, out_iwk1 = 0;
static int slot_count0 = 0;
static double theta_unfiltered0 = 0, theta_unfiltered1 = 0,
theta_filtered0 = 0, theta_filtered1 = 0;
double theta_unfiltered, theta_filtered;
static double w_unfiltered0 = 0, w_unfiltered1 = 0,
w_filtered0 = 0, w_filtered1 = 0;
static double avg_elec_speed_unfilt0 = 0, avg_elec_speed_unfilt1 = 0,
avg_elec_speed0 = 0, avg_elec_speed1 = 0;
static double v_DC_unfilt0 = 0, v_DC_unfilt1 = 0, v_DC0 = 0, v_DC1 = 0;
static double Iqold = 0, I_dold = 0, I_0old = 0;
double I_qnew, I_dnew, I_0new;

const double pi = 3.1415926;
// const int ts = 125; // 125us sample time => 8 kHz
const int ts = 90; // 90us sample time => 11.12 kHz = 5.56*2 kHz
(5.56 kHz switching)
const double numSlots = 4000.0; // The number of slots associated
// with this encoder (e.g. 4000 pulses/rotation).
// Uses "double" to ensure correct arithmetic
downstream.
const double maxDeltaSlotCount = 38.0; // Based on machine max. speed,
// the maximum change in slot count during a single
// iteration. Max. speed = 2000 rpm, w/ Ts <
1/5560 Hz
const double maxCurrent = 15.0; // Max current which converter can
supply.

Tsample = ts / 1000000.0;

// calculate the time
// note: t_ns should be set in the first statement here,
// as it will be used to calculate the approx time for the control−loop
// t_ns = gethrtimes() − t_ns_start; // gethrtimes(): time in ns since
// bootup
// t = t_ns*0.000000001; // ns—>s

temp = rtce_acromag_get_ts();
if (ts != temp) rtce_pwm_set_ts_for_pwm_and_AD(ts);
if (rbi->ts != ts) rbi->ts = ts;
// Process Input Channels
Ia = IN(0)*5.0 + 0.05; // current sensor 5A/V VSC1: 0.04, VSC2: 0.06;
Ib = IN(1)*5.0 + 0.04; // current sensor 5A/V VSC1: 0.04, VSC2: 0.04;
Ic = IN(2)*5.0 + 0.03; // current sensor 5A/V VSC1: 0.04, VSC2: 0.02;
v_DC.unfiltered = IN(3)*70.0; // voltage sensor 70V/V
operationalChange = IN(4); // allows an external command signal to be read

// Voltage sensor 30V/V
Vc = IN(7)*30.0; // voltage sensor 30V/V

// DC Capacitor Voltage Protection. Voltage rises if PMSM generating.
v_DC = v_DC.unfiltered;
second_order_low_pass(v_DC, 10.0, Tsample); */
v_DC = second_order_lpf(v_DC.unfiltered, &v_DC.unfilt0, &v_DC.unfilt1,
&v_DC0, &v_DC1, 0.707, 2.0*pi*1, Tsample);

if (v_DC > 450.0)
gating_on = FALSE;

SET_PWM_SWITCH(PWM6Q, PWMOFF);
slot_count = READ_SLOT_COUNT();

orig_slotcount = slot_count;
orig_slotcount0 = slot_count0;

// Determine whether the new "slot_count" is reasonable, or whether a value
// obtained assuming the speed is constant (linear interpolation)
// should be used.

// system initialization

// This allows the system to be initialized properly, and the glitch can be
// observed as happening only once consecutively.

delta_slot_count = slot_count0 - slot_count;

// check to see whether there’s an anomaly in change in slot_count
// this anomaly could be associated with:
// – wrap-around
// – acceleration/deceleration
// – a spurious slot_count measurement
// A 3–point median filter on speed is implemented.

// check for wrap-around and compensate
if (abs(delta_slot_count) > maxDeltaSlotCount) {
    if (abs(delta_slot_count + numSlots) < maxDeltaSlotCount ||
        abs(delta_slot_count - numSlots) < maxDeltaSlotCount) {

// Change the delta_slot_count based on wrap-around.

// for positive rotation
if (abs(delta_slot_count + numSlots) < maxDeltaSlotCount) {
    delta_slot_count = delta_slot_count + numSlots;
}
// for negative rotation
else {
    // if (abs(delta_slot_count - numSlots) < maxDeltaSlotCount)
    delta_slot_count = delta_slot_count - numSlots;
}

// if delta_slot_count has changed by more than 1 from the recent average then it is
// not correct. If it is not correct, then do not update the history term.
if (delta_slot_count > recentAvg + 1.0 ||
    delta_slot_count < recentAvg - 1.0) {
    if (recentAvg >= 0)
        curInstSpeed = recentAvg + 0.5; // +0.5 gives rounded instead of
        // truncated value
    else curInstSpeed = recentAvg - 0.5; // -0.5 for case of negative speed
} else {
    curInstSpeed = delta_slot_count;
    // update recent average (over 5 time steps). This only considers
    // values which have
    // not been discarded. storedDeltaSlotCount[0] is the oldest value
    for (i = 0; i < 4; i++) {
        storedDeltaSlotCount[i] = storedDeltaSlotCount[i + 1];
    }
    storedDeltaSlotCount[4] = curInstSpeed;
    recentAvg = 0;
    for (i = 0; i < 5; i++) {
        recentAvg = recentAvg + storedDeltaSlotCount[i];
    }
    recentAvg = recentAvg / 5;
}

// set estimSlotCount based on the results of the median filter if required
// It is assumed that multiple glitches will not happen consecutively.
// To prevent getting 'stuck' in assuming that the speed is one value, while
// the real speed is something else, this condition is forced.
// Note that a single glitch has 1 'up' and a corresponding 'down'
reading

// 2 examples of glitches:
// count is: ... 1750 -> 1755 -> 1758 -> 1765 -> 1770 ... ('down' + 'up ')
// count is: ... 1750 -> 1755 -> 1760 -> 2000 -> 1770 -> 1775 ...
if ((curInstSpeed != delta_slot_count) && errorCount < 2) {
    estimSlotCount = slot_count0 - curInstSpeed;
    errorCount = errorCount + 1;

    // account for wrap-around if necessary
    if (estimSlotCount > numSlots) {
        estimSlotCount = estimSlotCount - numSlots;
    } else if (estimSlotCount < 0) {
        estimSlotCount = estimSlotCount + numSlots;
    }
    slot_count0 = estimSlotCount;
} else {
    estimSlotCount = slot_count;
    errorCount = 0;
    slot_count0 = slot_count;
}

// slot_count0 = slot_count;  // caused problems.
// Use the estimated slot_count, as the real slot_count may not be correct
// slot_count0 = estimSlotCount;  // caused much more serious problems

// Determine the current rotor position based on slot_count.
// This gives the angle, Theta, to be used for the qd0 transform.
rotor_angle_m = 2.0*pi*estimSlotCount/numSlots;  // Mechanical rotor angle, in radians.
rotor_angle_e = 3.0*rotor_angle_m;  // Elec. rotor angle. 6 pole machine (3 pole-pairs).
Theta = rotor_angle_e + (angle_delta_ref)/360.0*2.0*pi;  // angle associated with d-axis
Theta = -Theta;  // Positive rotor motion => negative turning sensed.
    // That is to say, theta counting DOWN is positive by this convention.
    // e.g. +ve speed is 3999-->0 and -ve speed is 0-->3999
// low-pass filter Theta
theta_unfiltered = Theta;

// compensate history terms if wrap-around
if (abs(theta_unfiltered - theta_unfiltered0) > maxDeltaSlotCount/4000*3*2*pi) {
    // this is a case of wrap-around, assuming all the glitches have
    // previously been successfully removed.
    if (theta_unfiltered > theta_unfiltered0) {
        theta_unfiltered0 = theta_unfiltered0 + 3*2*pi;
        theta_filtered0 = theta_filtered0 + 3*2*pi;
    } else {
        // for case of (theta_unfiltered < theta_unfiltered0)
\[
\theta_{\text{unfiltered}} = \theta_{\text{unfiltered}} - 3 \times 2 \times \pi;
\]
\[
\theta_{\text{filtered}} = \theta_{\text{filtered}} - 3 \times 2 \times \pi;
\]

// first-order filter for \(\theta\)
\[
\theta_{\text{filtered}} = \left( \frac{\left( \theta_{\text{unfiltered}} + \theta_{\text{unfiltered}} \right) \cdot T_{\text{sample}} + \left( \frac{2}{1850 \times 2 \times \pi} - T_{\text{sample}} \right) \cdot \theta_{\text{filtered}}}{(T_{\text{sample}} + 2 / (1850 \times 2 \times \pi))} \right);
\]
\[
\theta_{\text{filtered}} = \theta_{\text{filtered}};
\]
\[
\theta_{\text{unfiltered}} = \theta_{\text{unfiltered}};
\]

// wrap-around of \(\theta_{\text{filtered}}\) if required
if \(\theta_{\text{filtered}} < (-3 \times 2 \times \pi - \text{angle}_\text{delta}_\text{ref} / 360.0 \times 2.0 \times \pi)\) {
    \[
    \theta_{\text{filtered}} = \theta_{\text{filtered}} + 3 \times 2 \times \pi;
    \]
} else if \(\theta_{\text{filtered}} > -\text{angle}_\text{delta}_\text{ref} / 360.0 \times 2.0 \times \pi\) {
    \[
    \theta_{\text{filtered}} = \theta_{\text{filtered}} - 3 \times 2 \times \pi;
    \]
}

// // alpha-beta-zero transform, using the scaling factor for amplitude-invariance
// // This transform is rotated 90 deg. (lead) from the direct transform found in Krause.
// // To compensate, \text{angle}_\text{delta}_\text{ref} should be set to 132.0 deg = 42 + 90
// \[
I_{\alpha} = 2.0 / 3.0 \times (I_a - 0.5 \times I_b - 0.5 \times I_c);
\]
\[
I_{\beta} = 2.0 / 3.0 \times (0.866 \times I_b - 0.866 \times I_c);
\]
\[
I_0 = 2.0 / 3.0 \times (0.707 \times I_a + 0.707 \times I_b + 0.707 \times I_c);
\]

// // qd transform
\[
I_q = -\sin(\theta_{\text{filtered}}) \times I_{\alpha} + \cos(\theta_{\text{filtered}}) \times I_{\beta};
\]
\[
I_d = \cos(\theta_{\text{filtered}}) \times I_{\alpha} + \sin(\theta_{\text{filtered}}) \times I_{\beta};
\]

// Current Transform
// direct transform to \(qd0\) from \(abc\)
// defined in Krause "Electromechanical Motion Devices" (1st ed.) eqs. 6.9–19 & 6.9–20
\[
I_{q\text{new}} = 2.0 / 3.0 \times (I_a \times \cos(\theta_{\text{filtered}}) + I_b \times \cos(\theta_{\text{filtered}} - 2.0 \times \pi / 3.0) + I_c \times \cos(\theta_{\text{filtered}} + 2.0 \times \pi / 3.0));
\]
\[
I_{d\text{new}} = 2.0 / 3.0 \times (I_a \times \sin(\theta_{\text{filtered}}) + I_b \times \sin(\theta_{\text{filtered}} - 2.0 \times \pi / 3.0) + I_c \times \sin(\theta_{\text{filtered}} + 2.0 \times \pi / 3.0));
\]
\[
I_{0\text{new}} = 2.0 / 3.0 \times 0.5 \times (I_a + I_b + I_c);
\]

// 2-point averaging filter to help alleviate switching effects which are present as a result of the other converter.
\[
I_{q\text{old}} = (I_{q\text{new}} + I_{q\text{old}}) / 2.0;
\]
\[
I_{d\text{old}} = (I_{d\text{new}} + I_{d\text{old}}) / 2.0;
\]
\[
I_{0\text{old}} = (I_{0\text{new}} + I_{0\text{old}}) / 2.0;
\]

// Voltage Transform - currently, only \(V_0\) is used to droop the \(I_{0\text{ref}}\),
the ZS reference

// direct transform to qd0 from abc
// defined in Krause "Electromechanical Motion Devices" (1st ed.) eqs.
6.9–19 & 6.9–20

// V_q = 2.0/3.0*(Va*cos(theta_filtered) + Vb*cos(theta_filtered - 2.0*pi/3.0) + Vc*cos(theta_filtered + 2.0*pi/3.0));
// V_d = 2.0/3.0*(Va*sin(theta_filtered) + Vb*sin(theta_filtered - 2.0*pi/3.0) + Vc*sin(theta_filtered + 2.0*pi/3.0));
// V_0 = 2.0/3.0*0.5*(Va + Vb + Vc);

// Rotor speed [electrical radians/sec]
w_unfiltered = 3.0*curInstSpeed*2.0*pi/numSlots/Tsample;

/*
  second_order_low_pass(w_unfiltered , 2.0*pi*LPCO, Tsample);
*/

w_filtered = second_order_lpf(w_unfiltered , &w_unfiltered0 ,
  &w_unfiltered1, &w_filtered0, &w_filtered1 ,
  0.707, 2.0*pi*LPCO, Tsample);

// determine average electrical speed (for display only) using a very
slow LPF (1 Hz)
/*
  avg_elec_speed = w_unfiltered;
  second_order_low_pass(avg_elec_speed , 2.0*pi*10, Tsample);
*/

avg_elec_speed = second_order_lpf(w_unfiltered , &avg_elec_speed_unfilt0 ,
  &avg_elec_speed_unfilt1, &avg_elec_speed0, &avg_elec_speed1 ,
  0.707, 2.0*pi*1, Tsample);

// set controller history terms to zero if the system is not gating
if (!gating_on) {
  out_dk1 = 0.0;
  out_qk1 = 0.0;
  out_0k1 = 0.0;
  out_iwk1 = 0.0;
  in_dk1 = 0.0;
  in_qk1 = 0.0;
  in_0k1 = 0.0;
  in_wk1 = 0.0;
}

if (torque_control) {
  // Set Iqref based on the torque reference (Tref), as per Dale
  Dolan’s code. He used
  // 1st order approximation and provided the 2nd order as an option.
  Iqref = 0.7602*Tref + 0.7353; // 1st order formulation
  // Iqref = 0.0007*Tref*Tref + 0.7446*Tref + 0.7855; // 2nd order
  // formulation
  Idref = Idrefmanual;
}
if (speed_control) {
    // Speed Controller
    // − sets Iqref, clipping at +/- maxCurrent
    // − is drooped based on Iqref

    // set parameters based on zero placement & gain
    Kiw = Kpw*aw;
    wref = wrefSet;
    if (operationalChange > 5) {
        wref = wref + operationalChangeDelta;
    }
    wrefDisplay = wref;

    in_wk = wref - w_filtered;
    in_wk = in_wk - Dw*Iqref; // Droop based on Iqref

    // PI, discretized with ZOH. The integrator and proportional
    components are
    // separate to allow for windup prevention which matches the
    Simulink case.
    out_iwk = out_iwk1 + Kiw*Tsample*in_wk1;
    out_pwk = in_wk*Kpw;

    // limit the integrator to prevent windup
    if (out_iwk > maxCurrent) out_iwk = maxCurrent;
    if (out_iwk < -maxCurrent) out_iwk = -maxCurrent;
    out_iwk1 = out_iwk;

    out_wk = out_iwk + out_pwk;

    //out_wk = Kpw*in_wk1; // Proportional Control

    // Limit out_wk to +/- maxCurrent
    // Clipping it here clips Iqref
    if (out_wk > maxCurrent) out_wk = maxCurrent;
    if (out_wk < -maxCurrent) out_wk = -maxCurrent;

    in_wk1 = in_wk;
    Iqref = out_wk;

    // Set Iqref to zero, as no d–axis current is desired for this
    operational mode
    Iqref = Iqrefset;

    // Zero–Sequence Current Reference (droop)
// Iref = Irefreset − Dzero * V_0;
// Iref = Irefreset − Dzero * I_0;
}

// Current Controllers are only enabled when required
if (torque_control || speed_control) {
    // q-axis Controller
    in_qk = Iqref − I_q;
    out_qk = out_qk1 + Kpq * in_qk + (Kiq * Tsample − Kpq) * in_qk1; // PI, 
    // discretized with ZOH
    out_qk = Kpq * in_qk1; // Proportional Control
    out_qk1 = out_qk;
    in_qk1 = in_qk;
    uq = out_qk;

    // d-axis Controller
    in_dk = Idref − I_d;

    // PI, discretized with ZOH
    out_dk = out_dk1 + Kpd * in_dk + (Kid * Tsample − Kpd) * in_dk1;

    // Proportional Control
    out_dk = Kpd * in_dk1;
    out_dk1 = out_dk;
    in_dk1 = in_dk;
    ud = out_dk;

    if (speed_control) {
        // 0-axis controller
        /* rtce_scalar_slider azero 59 0.0 500.0 "Zero-Sequence Integral
        Gain Kizero was 2.95"
        rtce_scalar_slider Dzero 0.33 0.0 2.0 "Zero-Sequence Droop"
        */
        in_0k = I0ref − I_0;
        Kizero = Kpzero * azero;
        in_0k = I0ref − I_0;

        // PI, disc. w/ ZOH
        out_0k = out_0k1 + Kpzero * in_0k + (Kizero * Tsample − Kpzero) * in_0k1;
        /*
        // Proportional Control
        out_0k = Kpzero * in_0k;

        // history terms
        out_0k1 = out_0k;
        in_0k1 = in_0k;
        u0 = out_0k;
        */
    } else
// set the requested zero-sequence to zero
u0 = 0;

// Specify output voltage vectors: cancel cross-coupling terms and possibly also the
// motor's back emf (along the q-axis)
Vtd = ud - w_filtered*Lq*I_q;
Vtq = uq + w_filtered*Ld*I_d; // + w_filtered*lambda_mag;
Vt0 = u0;

// Use voltage_control
// Specify the voltage space vector to create.
if (voltage_control) {
    Vtd = Vdref;
    Vtq = Vqref;
    Vt0 = 0.0;

    /* This allows a sinusoidal d-q reference to be created.
       * Vtd = Vdref*cos(sinusoidal_ref_angle);
       * Vtq = Vqref*cos(sinusoidal_ref_angle);
       * sinusoidal_ref_angle = sinusoidal_ref_angle+2.0*pi*frequency*Tsamp;
       * if (sinusoidal_ref_angle > 2*pi)
       *    sinusoidal_ref_angle = sinusoidal_ref_angle - 2.0*pi;
       */
}

FLICTDIMENT=sqrt(Vtd*Vtd + Vtq*Vtq);
Vlimit=v_DC/2.0; // PWM modulation limit

if (Vlength > Vlimit) {
    Vtd = Vtd*Vlimit/Vlength;
    Vtq = Vtq*Vlimit/Vlength;
    // resize the history terms if necessary
    out_qk1 = out_qk1*Vlimit/Vlength;
    out_dk1 = out_dk1*Vlimit/Vlength;
}

Vlength_corrected = sqrt(Vtd*Vtd + Vtq*Vtq); // For the display meter

// Vt_alpha = Vtd*cos(theta_filtered) - Vtq*sin(theta_filtered);
// Vt_beta = Vtd*sin(theta_filtered) + Vtq*cos(theta_filtered);
//
// Vta = Vt_alpha + 0.707*Vt0;
// Vtb = -0.5*Vt_alpha + 0.866*Vt_beta + 0.707*Vt0;
// Vtc = -0.5*Vt_alpha - 0.866*Vt_beta + 0.707*Vt0;
// Direct transform from qd0 to abc frame. See abc->qd0 for equation reference.
Vta = Vtq*cos(theta_filtered) + Vtd*sin(theta_filtered) + Vt0;
Vtb = Vtq*cos(theta_filtered - 2.0*pi/3.0) + Vtd*sin(theta_filtered - 2.0*pi/3.0) + Vt0;
Vtc = Vtq*cos(theta_filtered + 2.0*pi/3.0) + Vtd*sin(theta_filtered + 2.0*pi/3.0) + Vt0;

// Duty cycle based on SPWM
duty_cycle_new_a = Vta/(vDC/2.0);
duty_cycle_new_b = Vtb/(vDC/2.0);
duty_cycle_new_c = Vtc/(vDC/2.0);

// For classical operation, there is no control and the VSC acts as a sinusoidal voltage source
if (classical) {
    sinusoidal_ref_angle = sinusoidal_ref_angle + 2.0*pi*frequency*Tsample;
    if(sinusoidal_ref_angle > 2.0*pi)
        sinusoidal_ref_angle = sinusoidal_ref_angle - 2.0*pi;

duty_cycle_new_a = ma*sin(sinusoidal_ref_angle - adjust_angle*pi/180.0);
duty_cycle_new_b = ma*sin(sinusoidal_ref_angle - 2.094395102 - adjust_angle*pi/180.0);
duty_cycle_new_c = ma*sin(sinusoidal_ref_angle - 4.188790205 - adjust_angle*pi/180.0);
}

// SET_PWM Q1(duty_cycle_new_a);
// SET_PWM Q2(duty_cycle_new_b);
// SET_PWM Q3(duty_cycle_new_c);

SET_PWM_P1(duty_cycle_new_a);
SET_PWM_P2(duty_cycle_new_b);
SET_PWM_P3(duty_cycle_new_c);

gating_main_switch(gating_on);

//

*************************************************************************

// BEGIN - PREPARE VALUES FOR TALK & SCOPE - YOU MAY ADD SIGNAL/METERS
*************************************************************************

// Output to METERS in "TALK"
#define display_cut_off_freq (20.0)

Iqrefdisp = Iqref;
Theadisp = theta_filtered;
w_filtereddisp = w_filtered;
w_unfiltereddisp = w_unfiltered;
v_DCDisp = v_DC;
VlimitDisp = Vlimit;
VlengthDisp = Vlength;

SIGNAL(0) = Iqref;
SIGNAL(1) = Idref;
SIGNAL(2) = orig_slotcount;
SIGNAL(3) = v_DC_unfiltered;
SIGNAL(4) = theta_filtered;
SIGNAL(5) = Id;
SIGNAL(6) = Iq;
SIGNAL(7) = I0;
SIGNAL(8) = w_filtered;
SIGNAL(9) = w_unfiltered;
SIGNAL(10) = wref;
SIGNAL(11) = Vtb;

// END - PREPARE VALUES FOR TALK & SCOPE - YOU MAY ADD SIGNAL/METERS

// note: update_execution_time_meter() should be the last statement of the
// control-loop as it will calculate the approx AVERAGE time for the
// control-loop
update_execution_time_meter( t_ns );

// Routing table support function
void rtce_assign_data(void) {
    int i;
    for(i=0; i< MAX_DATA_BUF; i++) {
        SHOW(i) = *(ROUTE( rbi->route[i] ) );
    }
}

// function to operate second order low-pass filter
// pass in: x = x[n] , x0 = x[n-1] , x1 = x[n-2] , y0 = y[n-1] , y1 = y[n-2],
// Q = 1/(2*zeta) , wc = omega_critical , Tsample = sample period
// returns: second_order_lpf = y[n]
//
// Call as (with Q = 0.5 = 1/(2*zeta), fc = 1000):
// speed = second_order_lpf(speed_unfilt , &speed_unfilt0 , &speed_unfilt1 , 
// &speed_filt0 , &speed_filt1 , 0.5, 2.0*pi*1000, Tsample);
double second_order_lpf(double x, double *x0, double *x1, double *y0, double *y1, double Q, double wc, double Tsample) {
    double tmp = (wc*wc*Tsample*Tsample*(x + (*x0)*2 + *x1)
                 - (*y0)*(2*wc*wc*Tsample*Tsample - 8)
                 - (*y1)*(4 - 2*Tsample*wc/Q + wc*wc*Tsample*Tsample))
    / (4 + 2*Tsample*wc/Q + wc*wc*Tsample*Tsample);
    *y1 = *y0;
    *y0 = tmp;
    *x1 = *x0;
    *x0 = x;
    return tmp;
}

double first_order_lpf(double x, double *x0, double *y0, double wc, double Tsample) {
    double tmp = ((x + *x0)*Tsample + (2/wc - Tsample)*(*y0)/(Tsample + 2/wc));
    *x0 = x;
    *y0 = tmp;
    return tmp;
}
Appendix D

Description of RT-Linux Interface

The graphical user interface (GUI) is shown in Fig. D.1. The code to generate the GUI is given at the start of Appendix C.

![Figure D.1: Screenshot of the Control Interface](image)

The left-most column contains the gating check-box, the control scheme choice, and measured/calculated quantities. Only one control scheme is active at any time.

Note that the orientation of the $q$- and $d$-axes is not known until the encoder has reset. An encoder reset is guaranteed when the rotor has completed one revolution. The
orientation of the result pulse is adjusted using the *adjust angle* slider in the middle column.

*Vlimit* is the maximum size of the applied space vector (defined based on *dc-link Voltage*). *Vlength corrected* is the clipped version of *Vlength* (clipped based on *Vlimit*). *Iqref* is the requested $i_q$ current.

In each mode of operation, only some of the sliders/parameters are active. The modes of operation are:

- **Open-Loop Sinusoidal Voltage Source** makes the converter act as a sinusoidal voltage source. The relevant control parameters are in the middle column: *Open-loop Volt. Freq.* and *Open-loop ma*.

- **$q$-$d$ Voltage Control** makes the converter apply voltage along the $q$- and $d$-axes (open-loop). Parameters are $Vqref$ and $Vdref$.

- **Manual Torque and Id References** applies manual $i_q$ and $i_d$ references to closed-loop current controllers. It is intended only for calibration of a single-converter, so the 0-sequence loop is not active. The $i_q$ reference is calculated based on the requested $Tq. Ref$ and $Idref$ sets the $i_d$ reference. The $q$, $d$ and 0 gains (right-most column) are active.

- **Use Drooped Speed Control to set Iqref** sets the $i_q$ reference using the drooped PI speed loop reference speed *Speed Reference*. The reference for $i_d$ is now given by *D-Axis Current Reference* and *Zero-Sequence Current Reference* is also active. Also enabled is *Step Size*, which affects the *Tracked Speed* when the external signal is high (in which case *Tracked Speed* is the sum of *Speed Reference* and *Step Size*). The *Speed Loop* gains and the *Speed Droop* term are also active in this mode.

The parameters can be varied while the system is running, however the gating should be turned off before changing the mode of operation. All measured signals are displayed in real-time.
The *LPF freq [Hz] for speed* is used to filter the feedback speed used for cancellation of the cross-coupling terms and for input to the speed control loop.