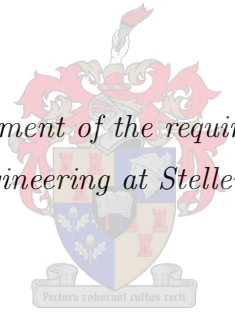
Modelling, Design and Implementation of a Small-Scale, Position Sensorless, Variable Speed Wind Energy Conversion System Incorporating DTC-SVM of a PMSG Drive with RLC Filter

by

Pieter Bouwer



Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Engineering at Stellenbosch University

Supervisor: Professor Maarten J Kamper
Department of Electrical & Electronic Engineering

March 2013

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2013

Abstract

Wind energy has proven to be a viable source of clean energy, and the worldwide demand is growing rapidly. Variable speed topologies, with synchronous generators and full-scale converters, are becoming more popular, and the effective control of these systems is a current trend in wind energy research.

The purpose of this study is the modelling, design, simulation and implementation of a small-scale, variable speed wind energy conversion system, incorporating the position sensorless direct torque control with space vector modulation, of a permanent magnet synchronous generator, including an RLC converter filter. Another aim is the development of a gain scheduling algorithm that facilitates the high level control of the system.

Mathematical models of the combined filter-generator model, in the stationary and rotating reference frames, are presented and discussed, from which equivalent approximate transfer functions are derived for the design of the controller gains.

The design of the controller gains, RLC filter components, gain scheduling concept and maximum power point tracking controller are presented. It is discovered that the RLC filter damping resistance has a significant effect on the resonance frequency of the system.

The system is simulated dynamically in both Simulink and the VHDL-AMS programming language. Additionally, the maximum power point tracking controller is simulated in the VHDL-AMS simulation, including a wind turbine simulator. The simulation results demonstrate good dynamic performance, as well as the variable speed operation of the system.

The practical results of torque and speed controllers show satisfactory performance, and correlate well with simulated results. The detailed gain scheduling algorithm is presented and discussed. A final test of the complete system yields satisfactory practical results, and confirms that the objectives of this thesis have been reached.

Acknowledgements

I am grateful to the following persons, without whom I would not have been able to complete this thesis:

- First and foremost, the Triune God, for redeeming me through His Son Jesus Christ, and comforting and leading me by His Holy Spirit.
- My father and mother, Johan and Nicolene Stemmet, and my two brothers, Hardus and Christo, for your love and support.
- My church family and friends who prayed for me and supported me.
- DB Hauptfleisch, for your steadfast friendship and insights.
- My supervisor, Professor Maarten Kamper, for providing me with guidance in my research and in life, as well as the opportunities to travel abroad to attend international conferences.

Thank you to the following persons for your technical support:

- Ivan Hobbs, for allowing me to use and build on your DSP software, and for your support in helping me understand it.
- David Groenewald, for providing me with support on the DSP software, as well as technical details.
- Jaco Serdyn, for helping me with the technical issues on the power electronic converter.
- Johannes Potgieter, for helping with the PMSG and other technical issues.

Contents

1 Introduction & Literature Summary **14**

1.1 Summary of Literature 14

1.1.1 Wind Generation 15

1.1.2 The Permanent Magnet Synchronous Generator 16

1.1.3 Power Converters 17

1.1.4 Converter Input LC Filter 17

1.1.5 Direct Torque Control in Wind Energy Conversion Systems . . . 18

1.1.6 VHDL-AMS 19

1.2 Proposed System Description 19

1.3 Overview of Thesis Structure 20

1.4 Summary and Conclusions 21

2 Theoretical Background and Modelling **22**

2.1 System Mathematical Modelling 22

2.1.1 Mechanical Model 23

2.1.2 Reference Frame Theory 23

2.1.3 Electrical Model 24

2.2 Space Vector Modulation 28

2.3 Direct Torque Control with Space Vector Modulation 31

2.3.1 Operation Principle 31

2.3.2 Approximate Transfer Functions 33

2.4 State Estimation 35

2.4.1 EEMF Observer 36

2.4.2 Flux Estimator 37

2.4.3 Speed estimation 38

2.5 RLC Filter 39

2.5.1 Resonance Damping 39

2.6 Wind Turbine Model 40

2.7 Gain Scheduling Concept 42
2.8 Summary & Conclusion 42

3 Design of Controllers, Estimators and Components 43
3.1 Digital Controller Design 43
3.1.1 Digital Torque Plant Derivation and Controller 43
3.1.2 Digital Flux Linkage Plant Derivation and Controller 45
3.1.3 Final Controller Gains 46
3.2 RLC Filter 48
3.3 Filter Resonance 51
3.4 Speed Controller 52
3.5 EEMF Observer and Flux Linkage Observer 53
3.6 Gain Scheduling Algorithm 53
3.7 Maximum Power Point Tracking 54
3.8 Summary and Conclusion 56

4 Simulation Methodology and Results 57
4.1 MATLAB Simulink Simulation Methodology 57
4.1.1 Top-Level Simulink System Model 58
4.1.2 Filter-Generator Model 58
4.1.3 EEMF Observer 59
4.1.4 Stator Flux Observer 60
4.1.5 Filter-Generator Flux Linkage and Torque Calculation 62
4.1.6 Speed Estimator 62
4.1.7 Direct Torque Controller 63
4.2 VHDL-AMS Simulation Methodology 64
4.2.1 EEMF Observer 65
4.2.2 Flux Observer 66
4.2.3 Filter-Generator Flux and Torque Calculation 67
4.2.4 Speed Estimation 67
4.2.5 Direct Torque Control 68
4.2.6 MPPT Controller 69
4.2.7 Wind Turbine Simulator 69
4.3 Simulation Results 70
4.3.1 Simulink Torque Step Simulation 70
4.3.2 Speed Estimation Comparison 73
4.3.3 VHDL-AMS Torque Step Simulation 75

4.3.4 VHDL-AMS Variable Speed MPPT Simulation 78
4.4 Summary and Conclusion 80

5 Practical Implementation & Results 82
5.1 Testing Methodology 82
5.2 Implementation Issues 84
5.3 Program Flow 85
5.4 Sub-system Test Results 89
5.4.1 State Estimattion 90
5.4.2 Torque Steps 92
5.4.3 Speed Control 93
5.5 WECS Test Results 95
5.5.1 Transition from Pre-cut-in to Cut-in 95
5.5.2 Transition from Cut-in to MPPT 95
5.5.3 Transition from MPPT to Max. Speed Limit 96
5.5.4 Re-Entry From Maximum Speed to MPPT 97
5.5.5 Cut-out Transition 99
5.6 Summary and Discussion 101

6 Conclusion 102
6.1 Summary and Conclusions 102
6.2 Recommendations for Future Work 105

A Derivation of the xy Reference Frame Filter-Generator Model 112

B Derivation of Approximate Parallel Structure DTC-SVM Transfer Functions 115
B.1 Flux Linkage 115
B.2 Torque 116

C Simulink Model Diagrams 118

D VHDL-AMS Simulation Code 126

E DSP Implementation Code 133

List of Figures

1.1 Representation of the system to be designed, simulated and implemented in this thesis. 20

2.1 Representation of the various coordinate reference frames in relation to one another. 23
2.2 Circuit model of the two-level, six switch IGBT converter (note that reverse diodes are not included in the representation) 24
2.3 A vector representation in the $\alpha\beta$, or stationary, reference frame of the operation principle of space vector modulation. 29
2.4 Representation of one switching period of SVM, using a double-edge PWM (DEPWM) switching algorithm. The reference voltage vector is in sector III, as in the vector diagram in Fig. 2.3. 31
2.5 A basic representation of the structure of DTC-SVM. 33
2.6 Controller representation of parallel-structure DTC-SVM. 33
2.7 Block diagram representation of the EEMF observer. 36
2.8 Block diagram representation of the combined voltage-current model flux estimator. 37
2.9 Diagram of an improved MRAS speed observer. 38
2.10 The wind turbine blade model. 40
2.11 Three-dimensional representation of the wind turbine blade power characteristics. A plane section reveals the turbine power versus rotational speed curve with wind speed as a parameter. 41

3.1 Digital torque controller diagram. 45
3.2 Digital flux linkage controller diagram. 46
3.3 Step response of the closed loop torque controller model. 47
3.4 Step response of the closed loop flux linkage controller model. 48
3.5 An equivalent lumped-element circuit diagram of the filter and generator, illustrating the input impedance as seen from the converter. 51

3.6 Lowering of system resonance frequency due to inclusion of damping resistance. 52
3.7 State diagram of gain scheduling algorithm. 53

4.1 Simplified representation of the top-level Simulink simulation, depicting the structure. 58
4.2 Implementation of the filter-generator continuous-time model in Simulink. 59
4.3 Representation of the EEMF observer of generator EMF, as implemented in the Simulink simulation. 60
4.4 Representation of the rotor position estimation, as implemented in the Simulink simulation. 60
4.5 Representation of the current model observer of generator stator flux, as implemented in the Simulink simulation. 61
4.6 Representation of the generator stator flux linkage observer, as implemented in the Simulink simulation. 61
4.7 Representation of the filter-generator flux calculation, as implemented in the Simulink simulation. 62
4.8 Representation of the filter-generator torque calculation block, as implemented in the Simulink simulation. 62
4.9 Representation of the derivative-filter method of rotor speed estimation, as implemented in the Simulink simulation. 63
4.10 Representation of DTC, as implemented in the Simulink simulation. . . 63
4.11 Diagram representing the structure of the VHDL-AMS simulation. . . . 65
4.12 Diagram representing the structure and functional significance of the proposed wind turbine simulator. 69
4.13 Simulation results of the torque step simulation in MATLAB Simulink, showing generator back EMF (a) and rotor angle (b), for an initial frequency of 20 Hz. 70
4.14 Simulation results of the torque step simulation in MATLAB Simulink, showing estimated stator flux linkage (a) and angle (b), for an initial frequency of 20 Hz. 71
4.15 Simulation results of the torque step simulation in MATLAB Simulink, showing estimated filter-generator torque (a) and filter-generator flux linkage magnitude (b), for an initial frequency of 20 Hz. 72

4.16 Simulation results of the torque step simulation in MATLAB Simulink, showing generator back EMF (a) and rotor angle (b), for an initial frequency of 40 Hz. 72
4.17 Simulation results of the torque step simulation in MATLAB Simulink, showing estimated stator flux linkage (a) and flux angle (a), for an initial frequency of 40 Hz. 73
4.18 Simulation results of the torque step simulation in MATLAB Simulink, showing filter-generator torque (a) and filter-generator flux linkage magnitude (b), for an initial frequency of 40 Hz. 74
4.19 Comparison of the rotor speed estimation of the simple filter method and the MRAS speed observer, with and without filtering, for an initial frequency of 20 Hz. 74
4.20 Comparison of the rotor speed estimation of the simple filter method and the MRAS speed observer, for an initial frequency of 40 Hz. . . . 75
4.21 Results of the VHDL-AMS torque step simulation, showing generator back EMF (a) and rotor position angle (b), for an initial frequency of 20 Hz. 76
4.22 Results of the VHDL-AMS torque step simulation, showing generator EM torque (a) and stator flux linkage (b), for an initial frequency of 20 Hz. 77
4.23 Results of the VHDL-AMS torque step simulation, showing generator back EMF (a) and rotor angle (b), for an initial frequency of 40 Hz. . 77
4.24 Results of the VHDL-AMS torque step simulation, showing generator EM torque (a) and stator flux linkage (b), for an initial frequency of 40 Hz. 78
4.25 Results of the MPPT VHDL-AMS simulation, showing generator electromagnetic torque and load (tubine) torque. 79
4.26 Results of the MPPT VHDL-AMS simulation, showing mechanical rotor speed and the input wind speed, represented by the dashed line. . . . 79
4.27 Results of the MPPT VHDL-AMS simulation, showing average generated electrical power and maximum available power that can be extracted from the wind (with the current turbine). 80

5.1 A representation of the laboratory test setup. 83

5.2 Flow diagram describing the generator interrupt service routine program flow. Here "State" bears reference to the state of the gain scheduling system, and not the physical state of the system (as in "State Estimation"). 86
5.3 Flow diagram describing the gain scheduling system flow of this system. 88
5.4 Practical estimated generator EMF ($\alpha\beta$ reference frame) at low speed. 91
5.5 Practical estimated generator EMF ($\alpha\beta$ reference frame) at high speed. 91
5.6 Practical estimated rotor position at low speed. 91
5.7 Practical estimated rotor position at high speed. 92
5.8 Measured electromagnetic torque (filter-generator) step in rotor speed acceleration direction (up-step), including corresponding simulated VHDL-AMS response. 93
5.9 Measured electromagnetic torque (filter-generator) step in rotor speed deceleration direction (down-step), including corresponding simulated VHDL-AMS response. 93
5.10 Measured phase *a* stator current and mechanical rotor speed response of speed control after *Cut-in*, and then a small speed step. 94
5.11 Measured phase *a* stator current and mechanical rotor speed response of speed control on load increase. 94
5.12 GS state diagram showing the state transition from the Pre-cut-in state to the Cut-in state. 95
5.13 Measured phase *a* stator current and mechanical rotor speed response of the state transition from Pre-cut-in to Cut-in 96
5.14 GS state diagram showing the state transition from the Cut-in state to the MPPT state. 96
5.15 Measured phase *a* stator current and mechanical rotor speed response of state transitions from Cut-in to MPPT. 97
5.16 GS state diagram showing the state transition from the MPPT state to the Max Speed Limit state. 97
5.17 Measured phase *a* stator current and mechanical rotor speed response of the state transition from MPPT to Max. Speed Limit. 98
5.18 Measured phase *a* stator current and mechanical rotor speed response of the state transition from Max speed re-entry to MPPT. 98
5.19 Mechanical torque sensor measurement for the re-entry transition from *Max. Speed Limit* to the *MPPT* state. 99
5.20 GS state diagram showing the state transition from the MPPT state to Cut-out, and then finally to Pre-cut-in. 99

5.21 Measured phase *a* stator current and mechanical rotor speed response of the state transition from *MPPT* to *Cut-out*. 100
5.22 Mechanical torque sensor output showing transitions from Pre-cut-in to Cut-in, to MPPT, to Cut-out and then finally to Pre-cut-in. 100

C.1 Top-level inverter/switching model of the Simulink simulation. 118
C.2 Top-level DEPWM pulse generation model of the Simulink simulation. 118
C.3 A component of the DEPWM pulse generation model of the Simulink simulation (First triggered block in Fig. C.2, clockwise from the top). . 119
C.4 A component of the DEPWM pulse generation model of the Simulink simulation (Second triggered block in Fig. C.2, clockwise from the top). 119
C.5 A component of the DEPWM pulse generation model of the Simulink simulation (Third triggered block in Fig. C.2, clockwise from the top). 119
C.6 Filter-generator model of the Simulink simulation. 120
C.7 EM torque calculation of the filter-generator model in Fig. C.6. 121
C.8 Top-level MRAS speed observer model of the Simulink simulation. . . . 121
C.9 Adaptive flux linkage current model of the MRAS speed observer. . . . 121
C.10 Modified dq to $\alpha\beta$ transformation of the adaptive flux linkage current model. 121
C.11 Top-level model of the Simulink simulation. 122
C.12 $\alpha\beta$ to dq transformation of the Simulink simulation. 123
C.13 dq to $\alpha\beta$ transformation of the Simulink simulation. 123
C.14 $\alpha\beta$ to xy transformation of the Simulink simulation. 123
C.15 xy to $\alpha\beta$ transformation of the Simulink simulation. 123
C.16 EEMF observer model of the Simulink simulation. 124
C.17 Stator flux linkage observer model of the Simulink simulation. 124
C.18 Current model of the stator flux linkage observer model. 124
C.19 DTC model of the Simulink simulation. 125

D.1 VHDL-AMS implementation of the RLC filter and PMSG circuit model. 126
D.2 VHDL-AMS implementation of the digital EEMF observer. 127
D.3 VHDL-AMS implementation of the digital flux linkage observer. 127
D.4 VHDL-AMS implementation of the digital derivative-filter speed estimation. 128
D.5 VHDL-AMS implementation of the digital speed controller. 128
D.6 VHDL-AMS implementation of digital DTC. 128

D.7 VHDL-AMS implementation of SVM and DEPWM switching signal generation. 129
D.8 VHDL-AMS implementation of the two-level IGBT converter circuit model. 130
D.9 VHDL-AMS implementation of the diode model. 130
D.10 VHDL-AMS implementation of the IGBT model. 131
D.11 VHDL-AMS implementation of the digital clock. 131
D.12 VHDL-AMS implementation of the reference frame coordinate transformations. 132

E.1 Code of the generator interrupt service routine, executed at each sample instance. 133
E.2 DSP implementation code of the EEMF observer. 134
E.3 DSP implementation of the flux observer. 135
E.4 DSP implementation of the derivative-filter speed estimation. 135
E.5 DSP implementation code of the DTC. 136
E.6 Code for states 0 and 1 of the gain scheduling system. 137
E.7 Code for state 2 of the gain scheduling system. 137
E.8 Code for states 3 to 5 of the gain scheduling system. 138
E.9 Code for states 6 and 7 of the gain scheduling system. 138
E.10 Code for state 8 of the gain scheduling system. 139

List of Tables

2.1 SVM boundary voltage vectors and the implications on converter switching and output voltages. 30
2.2 Coefficients of derived transfer function models of torque and flux linkage plants. 35

3.1 Final controller gains for the torque and flux linkage controllers. 46
3.2 Physical parameters of the system. 47
3.3 Final component values for the RLC filter. 51
3.4 Final gains of the EEMF observer, flux observer and speed controler. . 53
3.5 Description of the states of the GS algorithm. 54

5.1 Physical parameters of the WECS. 90

Chapter 1

Introduction & Literature Summary

We live in an ever growing and developing world where the hunger for energy is rapidly consuming our planet's limited natural resources. This unsettling realisation over the past few decades has initiated a global search for alternative energy sources. Wind energy has proven to be a viable source of clean energy, and has been used as early as Persian times.

According to [1], wind energy is the fastest growing renewable energy technology in the world. In [2] it is stated that in the 1990's wind generation capacity doubled every three years. In 2007, the USA alone had more than 16800 MW of installed wind capacity[3], and in Denmark, wind energy totalled 19.8 % of energy production in 2008[4].

This chapter commences with a summary of literature relevant to the thesis topic. The proposed system to be implemented is then presented and discussed, after which an overview of the thesis structure is given.

1.1 Summary of Literature

In the following section a summary of relevant literature is presented, which provides a background in the field of study of this thesis.

The general field of wind energy will be investigated first, after which literature dealing with selected topics such as the permanent magnet synchronous generator (PMSG), control strategies, VHDL-AMS and the output sinus filter will be discussed.

1.1.1 Wind Generation

The wind energy market is currently dominated by wind energy conversion system (WECS) topologies incorporating doubly-fed induction generators (DFIGs) with partial-scale converters; the interest in variable speed topologies, in general, is growing rapidly [5]. Interestingly, in an economic evaluation done in [6], it is seen that even though the energy capture of a fixed-speed system is substantially less than that of a variable speed system, the energy-per-cost of the fixed-speed system is superior. This cost analysis, however, was based on the cost of power electronic components at the time of writing.

Essentially, the biggest challenge in wind generation is the effective control strategies that govern the operation of the WECS. In [7], three important control objectives in wind generation are identified: control of the local bus voltage, maximum power capture, and minimisation of power loss in the generators. Maximum energy capture in a WECS uses the power coefficient, commonly referred to as C_p, to define the extent to which power is extracted from the wind. The power coefficient depends on the pitch angle and the tip speed ratio, which is proportional to the angular speed of the turbine [7].

The authors in [1] broadly classify wind energy conversion topologies into four categories, Type A through D. This classification is an excellent representation of the evolution of WEC topologies over the years. Type A is a purely fixed speed system, while Type B is a variable speed DFIG topology with limited controllability by varying rotor resistance. Types C and D incorporate power electronic converters, the latter incorporating a full scale converter.

Although it is found in [1] that the most rapidly growing of the topologies is of Type C, this thesis will focus on a Type D topology. It is believed that this topology will grow more rapidly in the coming years, especially in small-scale applications.

A main trend in the field of wind energy conversion is variable speed systems incorporating direct-drive permanent magnet synchronous genrators (PMSGs) with full scale power electronic converters (PECs). This trend can be attributed to improved energy capture, reduced mechanical stresses, higher efficiency and reliability [8]. According to [8] a variable speed system can capture up to 15% more energy than a comparable single fixed speed system. Another study estimates 15-23% less energy capture for a fixed speed system, as compared to a variable speed system, neglecting losses in the power electronic converter [6].

Wind speed usually varies very slowly compared to the time constants of the other sub-systems [9], generally a ten minute time constant is assumed [10]. Fast controllers

are thus not necessary for maximum power point tracking strategies.

In a study to determine the technical feasibility of variable speed stall-regulated wind turbine systems, it is found that variable speed operation significantly reduces loading on the wind turbine[5]. It however slightly increases structural loading on the tower. Stall-regulated methods are not necessarily in the scope of this thesis, but the concepts do make some valuable contributions to the topic of this thesis. The authors list some other advantages of variable speed operation that include reduced acoustic signature, increased power quality and the possible elimination of costly mechanical controls used in variable-pitch operation.

Another strong argument for variable speed strategies has to do with specific wind sites. Whereas a fixed-speed system is designed to be site-specific, operating as close as possible to the maximum power potential [5], variable speed systems are inherently much more flexible, and generic systems can be employed at a wider range of different wind sites. This is especially important when the majority of wind conditions are considered, which are, for the most part, less than ideal. As wind energy in general is relatively expensive at this point in time, it is imperative that energy capture is maximised in order that the cost-per-energy is kept to a minimum.

In a study in [11] on the control of variable speed fixed-pitch wind turbines, it is concluded that a properly designed variable speed system is capable of mimicking the power curve of a pitch-regulated (or variable-pitch) WECS. The article defines three regions of operation for the WECS, namely variable speed, fixed-speed and constant power operation. Stable and safe transitions between these regions are very difficult from a control perspective, the authors conclude.

According to [12] pitch controlled topologies are very costly, and fixed pitch topologies are becoming ever more popular, especially for low cost applications.

It is clear that the problem here is effective control of the wind turbine generator. Due to the stochastic nature of wind conditions, it is of utmost importance that the control systems governing the operation of the WECS be able to adapt to varying conditions.

1.1.2 The Permanent Magnet Synchronous Generator

The field of wind energy generation has mainly been dominated by the induction generator (IG), and more specifically the doubly-fed induction generator (DFIG), with a partial scale converter connected to the rotor windings. However in recent years, the incorporation of permanent magnet synchronous generators (PMSGs) have become an

increasingly attractive alternative [13].

The use of an IG in a WECS requires a gearbox, due to the high nominal rotational speeds of IGs as compared to the slow rotational speeds of wind turbines. Due to the high weight, mechanical losses, costs and maintenance requirements of gearboxes, it is desirable to do away with them. The incorporation of PMSGs in wind energy application addresses this problem, as high pole count PMSGs can operate at much lower speeds, eliminating the need for gearboxes[14][2]. A gear-less WECS is commonly referred to as being *direct-drive*.

Added advantages of PMSGs include higher power densities and inherently higher efficiencies than IGs. A possible disadvantage is the added losses of the full scale power converter in this topology; but one must also consider the fact that a full scale back-to-back converter will allow the PMSG-based WECS to operate under network disturbances [13], in addition to allowing full controllability.

1.1.3 Power Converters

Various types of power converters are used in distributed power systems, such as diode-bridge rectifiers, back-to-back inverters and matrix converters [15]. Variants of the back-to-back inverter include the three wire and four wire converters.

Due to significant advances in power electronic technology, specifically IGBT converters, and the proven reliability thereof [16], an IGBT converter is the technology of choice for this study. Advantages of using an an IGBT converter, as opposed to a traditional diode-based rectifier, include current harmonic reduction, the possibility to control output voltage, and four quadrant operation [17]. There is however a price to pay in terms of cost, as power electronic components remain expensive.

It has been shown that the use of active rectifiers allow for better energy capture, especially at low wind speeds, and allows more flexible control [15].

Factors that dictate the converter capacity include maximum instantaneous power as well as transients caused by the acceleration and deceleration of the wind turbine [11].

1.1.4 Converter Input LC Filter

An LC input filter is especially necessary if long cables are used from the converter to the generator [18], which is mostly the case in a WECS, where it is not possible to house the power electronics on the tower. The filter also provides generator friendly voltages, which extends the lifetime of the generators as well as reduces maintenance

issues, and higher power quality [19].

Transient or sustained resonances may occur with the presence of an LC filter, which may cause serious instability. It is therefore necessary to damp these resonances either actively or passively. Some active damping methods include shunt active power filters [20] as well as simulated virtual series resistances [1] [21] [22]. In this study a passive damping method is used, where damping resistances are placed in series with the LC filter capacitors – it is therefore more technically correct to refer to the filter as an RLC filter, as the resistance (damping) component of the filter now contributes significantly to the dynamic properties of the filter.

The caveats of the active damping methods mentioned are their complexity in design, and especially the implementation and expense of the shunt active damping methods in [20]. The method described in [22] and [21] also require higher switching frequencies. Although additional losses are incurred by implementing passive damping, the simplicity in design and implementation makes it the most attractive option for use in this thesis.

1.1.5 Direct Torque Control in Wind Energy Conversion Systems

The application of direct torque control (DTC) to WECSs is a relatively new concept. In [23] it is concluded that due to its insensitivity to physical parameters, inherent sensorless operation and ease in implementing variable speed strategies, DTC is more suited in WECSs than the classic field oriented control (FOC). The implementation in [23], however, does not implement an LC filter.

In [24] the authors apply DTC with space vector modulation (DTC-SVM) to wind energy, but their scheme is not direct-driven, sensorless or with an LC filter. In [25] classic DTC for maximum power point tracking (MPPT) control of a PMSG is used, but the system is neither sensorless nor with an LC filter.

In [26] a sensorless WECS is presented that uses classical DTC for variable speed MPPT control of an Interior-PMSG (IPMSG). It is concluded that DTC is very well suited for this type of application. The issue of a converter input filter or gearbox, however, is not addressed in said paper.

[1] In series with the filter capacitor

1.1.6 VHDL-AMS

With the increasingly mixed-signal nature of modern engineering designs, mixed-signal simulation languages are becoming increasingly prevalent. One such mixed-signal simulation language is VHDL-AMS. It is a variant of traditional VHDL that includes support for analogue modelling, and was standardised by IEEE only in 1999[27]. VHDL-AMS was born in 1994 when an IEEE working group began planning on how to extend VHDL to analogue applications[28].

VHDL-AMS modelling is categorised according to three types of applications: *Functional*, *behavioural* and *structural and physical*. Functional models consist of conditional and procedural statements. Behavioural and structural models, as the name suggests, describe the interface of a model and its interaction across this interface. Physical models describe the nature of a model in a specific physical domain. Any VHDL-AMS model is either one or a combination of these models[28].

Even 13 years after its standardisation, the technology is still very young, with endless possibilities for improvement. VHDL-AMS provides great opportunities to designers, and its diversity of application is clear[28].

1.2 Proposed System Description

The objective of this thesis is the modelling, design and implementation of a complete position sensorless WECS, incorporating direct torque control with space vector modulation (DTC-SVM) of a permanent magnet wind generator (PMSG) with an RLC input filter to the converter. It is believed that this topology addresses many of the problems identified in literature.

It must be emphasised that this is a small-scale system[2], which forms part of the motivation for a fixed pitch system. It is also part of the objectives of this research to implement a WECS with as few as possible moving parts, hence the direct-driven approach.

An overview of the proposed system, that is believed to satisfy the description given, is presented in Fig. 1.1. The gain scheduling (GS) algorithm selects between two controllers, $D_s(s)$ and $f(\hat{\omega}_r)$, which are fixed speed control and MPPT control respectively, according to the system's estimated variables. The *Estimation* block represents all the estimation components of the system that are required for the system

[2]There is no universal definition as of yet, but some sources agree on a power rating of less than 100 kW.

to achieve sensorless operation. The controllers generate a torque command m_f^* as an input to the DTC-SVM component, which in turn generates the duty cycle for the switching of the IGBT's. The generator currents $\vec{i}_s$ and voltages $\vec{v}_s$ are the only measured variables, from which all other system variables are estimated.

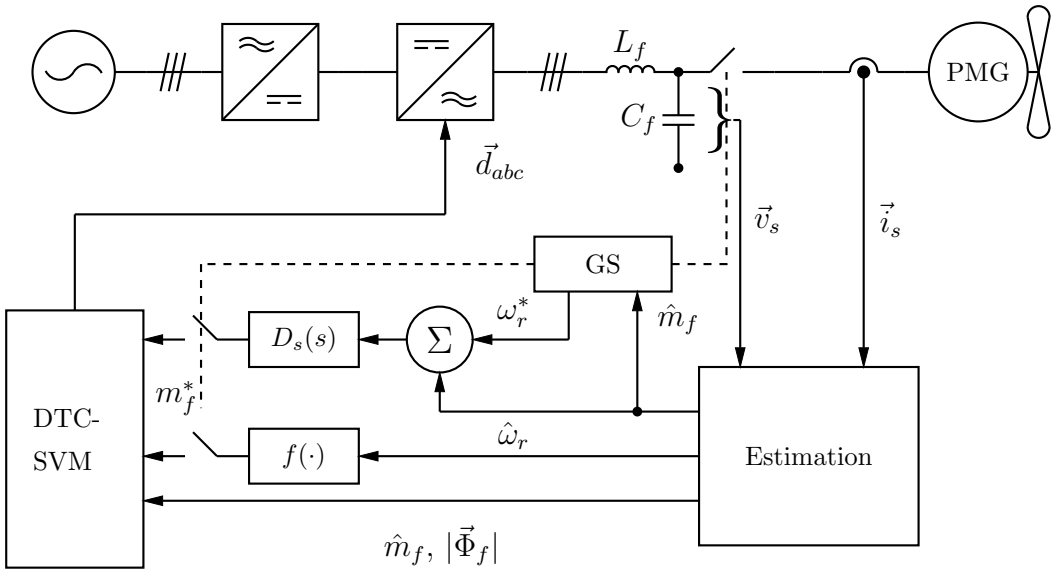


Figure 1.1: *Representation of the system to be designed, simulated and implemented in this thesis.*

As is clear from the diagram, this study is only concerned with the generator side of the WECS, and grid connection issues and transients are not dealt with in this thesis.

1.3 Overview of Thesis Structure

The background literature summary, along with the introduction of the thesis, is presented in this chapter. In the following chapter, titled *Theoretical Concepts and Modelling*, the mathematical modelling of the system is presented, as well as theoretical concepts that either underpin or contribute to the work done in this thesis; this also forms part of the literature study. Together, these two chapters form the introductory part of the thesis.

The designs of various sub-systems comprising the proposed WECS are presented in the *Design of Controllers, Estimators and Components* chapter. The simulation methodologies of both MATLAB Simulink and VHDL-AMS, including simulation results, are presented and discussed in the *Simulation Methodology & Results* chapter. This detailed technical part of the thesis concludes with the *Practical Implementation & Results* chapter, where the practical testing methodology is presented, practical

issues are discussed and finally test results are presented.

The thesis ends with the *Conclusion & Recommendations* chapter, where relevant conclusions are drawn from the work presented in the preceding chapters, and recommendations for future work in this line of study is presented. Relevant appendices are given after the *Bibliography*.

1.4 Summary and Conclusions

A summary of relevant literature is presented, whereafter the proposed system is presented and and overview of the thesis structure is given.

The trends in the literature are strongly guided towards variable speed systems. Although large scale systems are dominated by the DFIG, the PMSG seems to be gaining more and more acceptance in small-scale implementations, which is within the scope of this thesis.

It is clear from the literature that the problem faced in any variable speed WECS is a control problem, regardless of the topology. The section on power converters strengthens the idea to implement fully controllable IGBT converters, for reasons including reliability and flexibility. Without incorporating fully controllable power electronic converters, solutions to the problems identified in literature become implausible.

Although the application of DTC to WECS is a relatively new concept, the conclusions of the available literature endorse this. The consensus is that DTC is more suited to the control of WECSs than FOC.

Different operating modes for varying wind conditions and the transitions between these modes also pose complex control and regulation problems, and merit further investigation.

Chapter 2

Theoretical Background and Modelling

In this chapter, the theoretical concepts underlying the proposed system is presented and discussed. This chapter contributes toward the literature study, but is more focused towards the technical details in the literature that support the topic of this thesis.

Some foundational concepts are presented first, after which the ideal circuit model of the electrical system is presented, analysed and mathematically modelled. Reference frame theory and other mathematical transformations will then be applied to the standard model to obtain the final models that are used in the design and simulation.

The estimation theory of the system will then be presented. The chapter concludes with a brief overview of the mechanical and aerodynamic theoretical concepts that are relevant to this system.

2.1 System Mathematical Modelling

The generator to be utilised in the experiments is a radial flux, surface Permanent Magnet Synchronous Generator (PMSG). The standard, simplified model of a PMSG is widely publicised and mathematical derivations will therefore not be presented in this thesis.

Referring to the system overview diagram in Fig. 1.1, the total system to be modelled consists of the wind turbine, PMSG and an RLC filter. Firstly, the mechanical equations of the system will be presented, after which the modelling of the electrical system will be presented and discussed. The modelling of the wind turbine is presented and discussed in the last section of this chapter.

2.1.1 Mechanical Model

The mechanical equations that describe the dynamics of the generator rotor and turbine are

$$J\frac{d\omega_m}{dt} = m_e - m_L, \tag{2.1a}$$

$$\omega_r = N_p\omega_m, \tag{2.1b}$$

$$\omega_r = \frac{d\theta_r}{dt}. \tag{2.1c}$$

where J represents the combined inertia of the rotor and turbine, ω_m and ω_r the mechanical and electrical rotational speeds of the rotor, respectively, m_L and m_e the turbine torque and electromagnetic torque, respectively, and N_p the number of pole pairs of the PMSG. Note that friction is neglected in the mechanical model in (2.1).

2.1.2 Reference Frame Theory

The explanation of reference frame theory in this section is mostly based on work in [29].

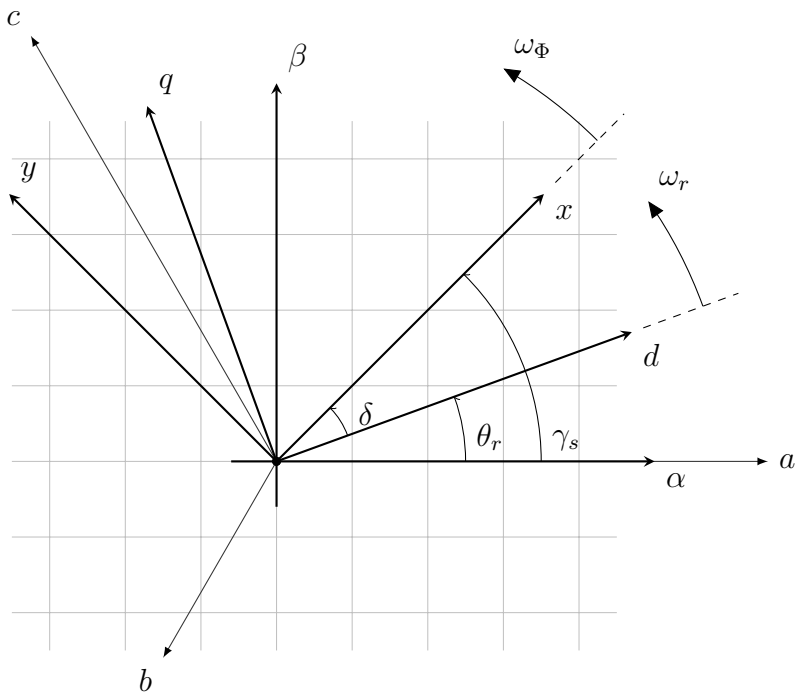


Figure 2.1: *Representation of the various coordinate reference frames in relation to one another.*

Variables of three phase machines can be mathematically transformed to arbitrary, fictitious reference frames, that rotate in synchronism with either the rotor or stator

magnetic field. This method was first introduced to simplify models of electric machines that have time-varying inductances.

To better visualise the various reference frames trigonometrically, refer to Fig. 2.1.

The *abc* reference frame, as the starting point, can be regarded as a three axis space of which the base vectors are not orthonormal. The $\alpha\beta$ transformation then transforms this *abc* space into an orthonormal coordinate space.

The rotor synchronously rotating (dq) and stator flux linkage synchronously rotating reference frames (xy) rotate in the direction indicated at angular frequencies of ω_r and ω_Φ, respectively, relative to the stationary reference frame ($\alpha\beta$). The rotational speeds ω_r and ω_Φ differ only under dynamic conditions. The expression for ω_Φ, in terms of ω_r, is given in (2.7).

2.1.3 Electrical Model

To achieve full vector control of the generator, a two-level IGBT voltage source inverter is used as the power electronic converter. A detailed electrical model of the complete generator-side electrical system, in the *abc* reference frame, is given in Fig. 2.2.

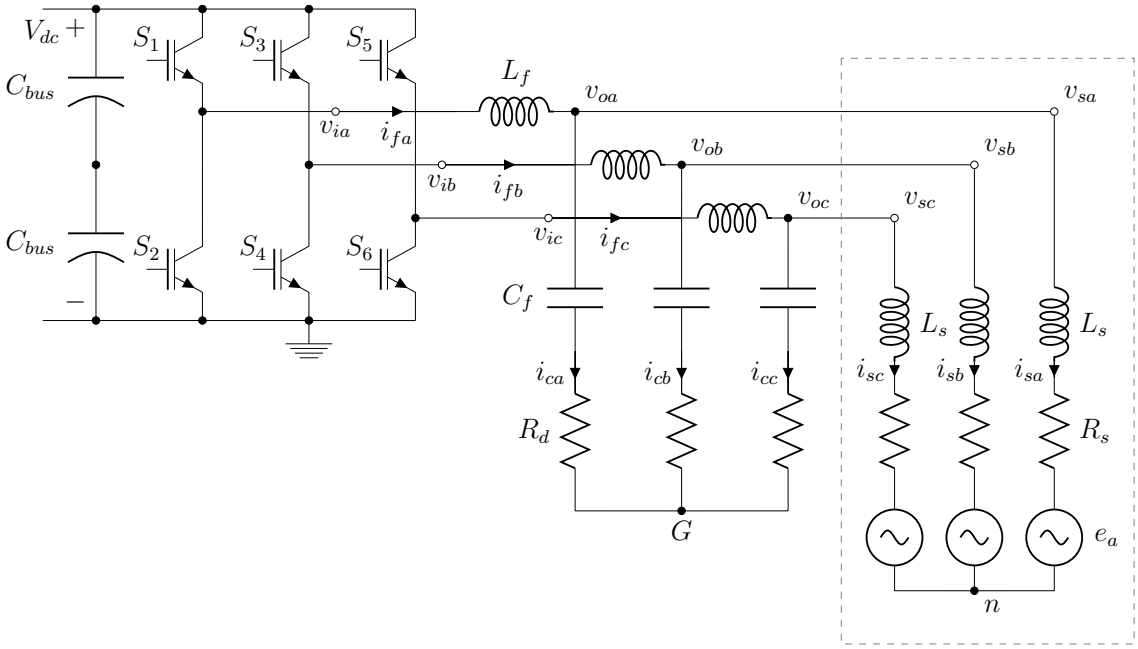


Figure 2.2: *Circuit model of the two-level, six switch IGBT converter (note that reverse diodes are not included in the representation)*

Using simple circuit analysis techniques, the mathematical equations describing the electrical model in Fig. 2.2 are obtained as

$$\vec{v}_i = r_f \vec{i}_f + L_f \frac{d\vec{i}_f}{dt} + \vec{v}_o, \tag{2.2a}$$

$$\vec{v}_s = R_s\vec{i}_s + L_s\frac{d\vec{i}_s}{dt} + \vec{e}_s = R_s\vec{i}_s + \frac{d\vec{\Phi}_s}{dt}, \tag{2.2b}$$

$$\vec{\Phi}_s = L_s\vec{i}_s + \Phi_m \begin{bmatrix} \cos(\theta_r) \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix}, \tag{2.2c}$$

$$\vec{e}_s = -\Phi_m\omega_r \begin{bmatrix} \sin\theta_r \\ \sin\left(\theta_r - \frac{2\pi}{3}\right) \\ \sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix}, \tag{2.2d}$$

$$\vec{v}_o = \frac{1}{C_f}\int \vec{i}_c dt + R_d\vec{i}_c. \tag{2.2e}$$

where $\vec{\Phi}_s$ is the stator flux linkage, Φ_m the permanent magnet flux and θ_r the rotor position of the PMSG.

The vectored (e.g. $\vec{x}_s$) variables in (2.2) represent vectors of three-phase components, a, b and c. This notation is also used to represent vectors of two dimensions in both the stationary and rotating reference frames further on. The expression for the stator flux linkage $\vec{\Phi}_s$ assumes that the three-phase system is positive phase sequenced.

What is important to note from the equations given in (2.2) is that the RLC filter output voltage is not the same as the line to neutral terminal voltage of the generator. The voltage between the generator and filter neutral points is not zero; from work done in [30] it can be shown that the equation in (2.3) holds true. This clearly shows coupling between the three phases and would make modelling the system by means of transfer function models quite difficult.

It is also assumed that the neutral points of the inverter and that of the RLC filter are connected, even though they are not in practice. The neutral point of the the inverter output can, however, be arbitrarily chosen as being the same as the RLC filter[1], which simplifies modelling.

Stationary Reference Frame ($\alpha\beta$)

In this section, the derivation of the stationary reference frame model of the electrical model, given in (2.2), is presented. The derivation of the stationary reference frame model is as follows:

$$\begin{bmatrix} v_{sa} \\ v_{sb} \\ v_{sc} \end{bmatrix} = \frac{1}{3} \cdot \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} v_{oa} \\ v_{ob} \\ v_{oc} \end{bmatrix}. \tag{2.3}$$

[1]This only dictates where the output is then "measured".

The voltages are transformed into the stationary reference frame($\alpha\beta$-reference frame) by using (2.3), (2.4a) and (2.4b).

$$\overline{K}_s = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \tag{2.4a}$$

$$\overline{K}_s^{-1} = \overline{K}_s^T, \tag{2.4b}$$

$$\begin{bmatrix} v_{s\alpha} \\ v_{s\beta} \end{bmatrix} = \frac{1}{3} \cdot \overline{K}_s \cdot \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \cdot \overline{K}_s^{-1} \cdot \begin{bmatrix} v_{o\alpha} \\ v_{o\beta} \end{bmatrix} \tag{2.4c}$$

$$= \begin{bmatrix} v_{o\alpha} \\ v_{o\beta} \end{bmatrix}. \tag{2.4d}$$

The output voltages of the RLC filter and the terminal voltages of the generator can now be used interchangeably. The derived equations in the $\alpha\beta$-reference frame are given in (2.5).

$$\vec{v}_{i,\alpha\beta} = R_f \vec{i}_{f,\alpha\beta} + L_f \frac{d\vec{i}_{f,\alpha\beta}}{dt} + \vec{v}_{o,\alpha\beta}, \tag{2.5a}$$

$$\vec{v}_{o,\alpha\beta} = \frac{1}{C_f} \int \vec{i}_c dt + R_d \vec{i}_c, \tag{2.5b}$$

$$\vec{v}_{s,\alpha\beta} = R_s \vec{i}_{s,\alpha\beta} + \frac{d\vec{\Phi}_{s,\alpha\beta}}{dt}, \tag{2.5c}$$

$$\vec{\Phi}_{s,\alpha\beta} = L_s \vec{i}_{s,\alpha\beta} + \Phi_f \begin{bmatrix} \sin\theta_r \\ \cos\theta_r \end{bmatrix}, \tag{2.5d}$$

$$m_e = \left(\frac{3}{2}\right) N_p \left(\Phi_{s\alpha} i_{s\beta} - \Phi_{s\beta} i_{s\alpha}\right). \tag{2.5e}$$

Stator Flux Linkage Synchronously Rotating Reference Frame (xy) Model

To implement parallel structure DTC-SVM, the system must be expressed in the stator flux linkage synchronously rotating reference frame (xy-frame). The reference frame in this regard rotates in synchronism with the stator flux linkage of the permanent magnet wind generator.

The system equations in the xy-reference frame are obtained by applying the transformation matrix in (2.6) to (2.5).

$$\overline{K}_{xy} = \begin{bmatrix} \cos\gamma_s & \sin\gamma_s \\ -\sin\gamma_s & \cos\gamma_s \end{bmatrix}. \tag{2.6}$$

The xy-reference frame equations of the system are then

$$\vec{v}_{i,xy} = R_f\vec{i}_{f,xy} + L_f\frac{d\vec{i}_{f,xy}}{dt} + L_f\omega_\Phi\begin{bmatrix} -i_{fy} \\ i_{fx} \end{bmatrix} + \vec{v}_{o,xy}, \tag{2.7a}$$

$$\vec{v}_{o,xy} = R_d\vec{i}_{c,xy} + \vec{v}_{c,xy}, \tag{2.7b}$$

$$\vec{i}_{c,xy} = C_f\frac{d\vec{v}_{c,xy}}{dt} + C_f\omega_\Phi\begin{bmatrix} -v_{cy} \\ v_{cx} \end{bmatrix}, \tag{2.7c}$$

$$\vec{v}_{s,xy} = R_s\vec{i}_{s,xy} + \frac{d\vec{\Phi}_{s,xy}}{dt} + \omega_\Phi\begin{bmatrix} 0 \\ \Phi_{sx} \end{bmatrix}, \tag{2.7d}$$

$$\Phi_{sx} = L_s i_{sx} + \Phi_m\cos\delta, \tag{2.7e}$$

$$0 = L_s i_{sy} - \Phi_m\sin\delta, \tag{2.7f}$$

$$m_e = \left(\frac{3}{2}\right)N_p\Phi_{sx}i_{sy}, \tag{2.7g}$$

$$\delta = \gamma_s - \theta_r, \tag{2.7h}$$

$$\omega_\Phi = \frac{d\gamma_s}{dt} = \omega_r + \frac{d\delta}{dt}, \tag{2.7i}$$

where ω_Φ is the filter-generator frequency, γ_s the stator flux position angle and δ the power angle of the PMSG, as defined in Fig. 2.1.

The coupling between the x and y components, as well as dependency on system frequency, is clear, and arise due to the inclusion of the RLC filter.

Filter-Generator Model

Due to the addition of the RLC filter, the terminal voltages of the generator, and consequently the torque and flux linkage, are no longer directly controlled. It is now necessary to define the filter-generator torque m_f and filter-generator flux linkage Φ_f, which will become the new controlled variables. The expressions of the new controlled variables are given in (2.8) [18][31].

$$\vec{\Phi}_{f,xy} = L_f\vec{i}_{f,xy} + \vec{\Phi}_{s,xy}, \tag{2.8a}$$

$$m_f = \left(\frac{3}{2}\right) N_p \left(\Phi_{fx} i_{fy} - \Phi_{fy} i_{fx}\right). \tag{2.8b}$$

The equations in (2.8) have the same form in both the $\alpha\beta$- and xy-reference frames, with the subscripts denoting the reference frames being the only difference.

Rotor Synchronously Rotating Reference Frame (dq) Model

Although it is common practice to assume equal d- and q-axis inductances in surface mounted permanent magnet machines, it is demonstrated that saturation effects do, in fact, cause a decrease in the d-axis inductance [32]. The transformation matrix from the $\alpha\beta$ to dq reference frame is

$$K_{dq} = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix}. \tag{2.9}$$

If the dq-transformation in (2.9) is applied to (2.5), the machine equations for a PMSG become

$$v_{sd} = R_s i_{sd} + L_d \frac{di_{sd}}{dt} - \omega_r L_q i_{sq}, \tag{2.10a}$$

$$v_{sq} = R_s i_{sq} + L_q \frac{di_{sq}}{dt} + \omega_r L_d i_{sd} + \omega_r \Phi_m, \tag{2.10b}$$

$$\Phi_{sq} = L_q i_q, \tag{2.10c}$$

$$\Phi_{sd} = L_d i_d + \Phi_m, \tag{2.10d}$$

$$m_e = \frac{3N_p}{2} \left(\Phi_q i_{sd} - \Phi_d i_{sq}\right). \tag{2.10e}$$

This concludes the mathematical models that will be used in the design and simulation of this system.

2.2 Space Vector Modulation

In this section, the theory and operation principle of space vector modulation (SVM) will be explained. In-depth theoretical analysis will not be presented, as this is not the goal of this thesis. The theory in this section is mostly adapted from [33].

SVM is an alternative inverter switch control method to traditional sinusoidal PWM, over which it has many advantages. Control structures are greatly simplified and power electronic hardware is exploited to the maximum. SVM inherently realises third harmonic injection, increasing the maximum output voltage of the inverter.

Important characteristics for the implementation of SVM are that there must be a single DC source to the converter, the system to be controlled has an insulated neutral and balanced three phase voltages.

SVM is realised in the stationary reference frame, as depicted in Fig. 2.3. SVM can be applied to many different topologies of power electronic converters, but only a two-level six switch converter topology, as shown in Fig. 2.2, is considered in this thesis. The following SVM scheme will therefore be constrained to this topology.

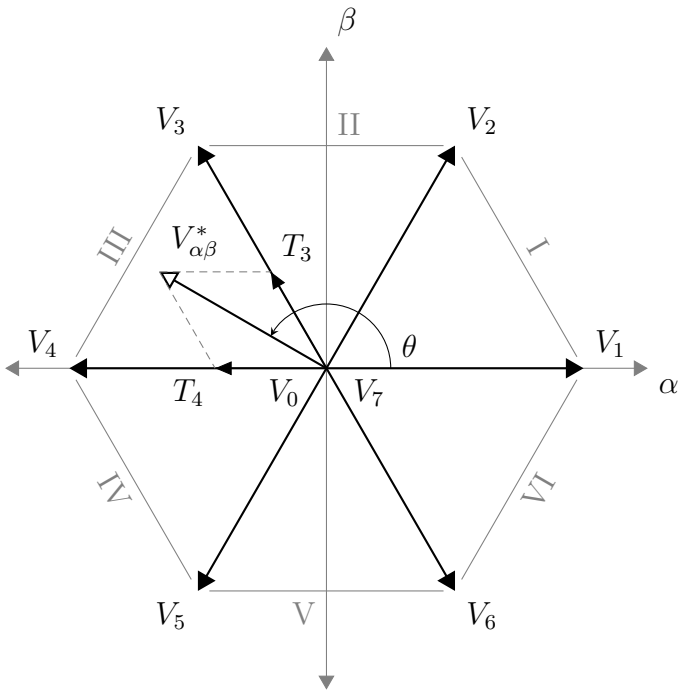


Figure 2.3: *A vector representation in the $\alpha\beta$, or stationary, reference frame of the operation principle of space vector modulation.*

The $\alpha\beta$ plane, as shown in Fig. 2.3, is divided up into six equal sectors, with discrete voltage vectors being the boundaries of these sectors. These discrete boundary voltage vectors are the only eight possible outputs of this topology of converter.

The boundary voltage vectors and their relationship to the converter in Fig. 2.2 are given in Table 2.1. The top level switches of the converter (S_1, S_3 & S_5), and the bottom level switches (S_2, S_4 & S_6) switch complimentary. In the table, a 0 represents an *open* or *off* switch, while a 1 represents a *closed* or *on* switch. The corresponding phase-to-ground voltages for the specific switching states are also given in the table.

The reference voltage $V^*_{\alpha\beta}$, which is the desired voltage at the inverter terminals, rotates in the $\alpha\beta$ plane at a certain angular frequency, with angle θ. This voltage vector is then projected as T_3 and T_4 (shown in the figure; subscripts corresponding to boundary voltage vectors) onto the boundary voltage vectors enclosing the current

Table 2.1: *SVM boundary voltage vectors and the implications on converter switching and output voltages.*

Vector	S_1	S_3	S_5	S_2	S_4	S_6	V_a	V_b	V_c
V_0	0	0	0	1	1	1	0	0	0
V_1	1	0	0	0	1	1	V_{dc}	0	0
V_2	1	1	0	0	0	1	V_{dc}	V_{dc}	0
V_3	0	1	0	1	0	1	0	V_{dc}	0
V_4	0	1	1	1	0	0	0	V_{dc}	V_{dc}
V_5	0	0	1	1	1	0	0	0	V_{dc}
V_6	1	0	1	0	1	0	V_{dc}	0	V_{dc}
V_7	1	1	1	0	0	0	V_{dc}	V_{dc}	V_{dc}

sector.

The projections will then ultimately determine the duty cycles of the power electronic switches. The size of the projection vectors determine the amount of time that the switches spend in a certain switching state for each switching period. An example timing diagram of one such switching period is given in Fig. 2.4 to illustrate this. The order in which the boundary vectors are switched in a specific switching period can be arbitrarily chosen, as long as the total switching time of a vector remains the same in any sequence. Note that V_0 and V_7 are the zero vectors, and can be interchangeably used.

From Fig. 2.4 it can be seen that each voltage vector is switched for a certain time. These times can easily be calculated as

$$t_3 = \frac{|T_3|}{|V_3|} T_s, \tag{2.11a}$$

$$t_4 = \frac{|T_4|}{|V_4|} T_s, \tag{2.11b}$$

$$t_0 = T_s - t_3 - t_4. \tag{2.11c}$$

Applicable total *on* and *off* switching times can then be calculated for each switch pair. The preceding explanation serves just as an illustration, and does not necessarily reflect the physical implementation of SVM. The theoretical basis, however, remains the same.

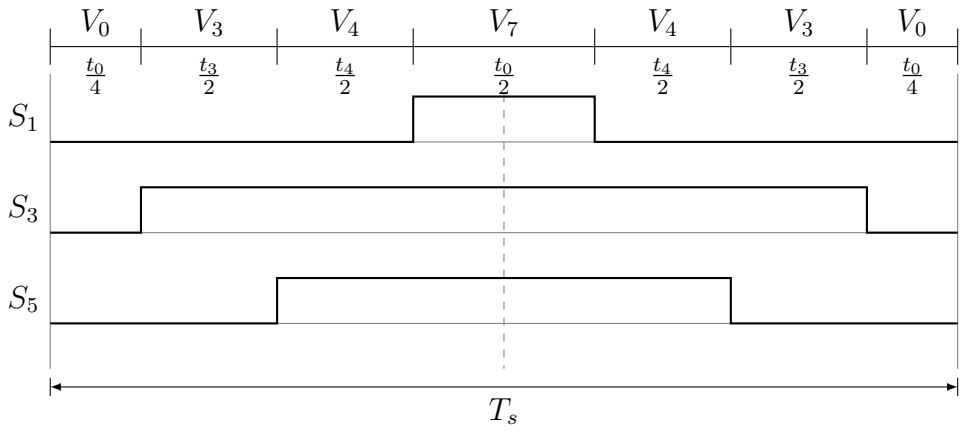


Figure 2.4: *Representation of one switching period of SVM, using a double-edge PWM (DEPWM) switching algorithm. The reference voltage vector is in sector III, as in the vector diagram in Fig. 2.3.*

2.3 Direct Torque Control with Space Vector Modulation

Classical direct torque control (DTC) will not be dealt with in this thesis, but it must be noted that DTC with space vector modulation (DTC-SVM) is a direct descendent of it. The inherent shortcomings of classic DTC, which include variable switching frequency, severe torque ripple and implementation complexity [34], is overcome by implementing DTC-SVM.

DTC-SVM has many advantages that strongly support the application thereof in this system. Most importantly, these advantages include:

- Direct control over electromagnetic torque, which allows for easy implementation of maximum power point tracking strategies in wind energy conversion systems[26].
- Requires a lower switching frequency, especially when compared to classical DTC[35].
- Offers very good speed control performance, which is a component of the WECS[35].

2.3.1 Operation Principle

The principle of operation of direct torque control with space vector modulation (DTC-SVM) will be explained in this section. Concepts are mostly adopted from [34] and [36]. As is shown in the model derivation of the combined filter-motor model, the

electromagnetic torque of the generator is not directly controlled as a result of the RLC filter. The principle, however, still remains the same. It will therefore be sufficient to assume for this section that the RLC filter is not included.

Although DTC can be applied to many different variations of synchronous machine (SMs), this thesis is mainly concerned with the control of a (theoretically) non-salient permanent magnet synchronous generator (PMSG).

The electromagnetic torque developed by the PMSG, in terms of dq-frame variables, can be expressed by the following equation:

$$m_e = \frac{3}{2} N_p \left(\frac{|\Phi_s| \Phi_m \sin \delta}{L_d} - \frac{|\Phi_s|^2 (L_q - L_d) \sin 2\delta}{2 L_d L_q} \right). \tag{2.12}$$

When a non-salient PMSG is considered, and we equate the static stator inductance to L_s, the expression in (2.12) reduces to

$$m_e = \frac{3}{2} N_p \frac{|\Phi_s| \Phi_m \sin \delta_\Phi}{L_s}. \tag{2.13}$$

The very principle of DTC can be deduced from (2.13), in that the electromagnetic torque m_e of the synchronous machine can be directly controlled by varying the torque angle δ, also referred to previously as the *power angle* (see Fig. 2.1), which is actually the angle difference between the PM rotor field and the rotating stator flux linkage. The torque angle can therefore be controlled by varying the stator flux angle γ_s, as shown in Fig. 2.1.

When one considers a rearrangement of (2.5c)

$$\frac{d\vec{\Phi}_{s,\alpha\beta}}{dt} = \vec{v}_{s,\alpha\beta} - R_s \vec{i}_{s,\alpha\beta}, \tag{2.14}$$

it is clear that the adjustment of the PMSG terminal voltage $\vec{v}_{s,\alpha\beta}$ has a direct effect on the derivative of the stator flux linkage. The appropriate voltage vector is then selected by the SVM, and DTC-SVM is realised.

The basic configuration of DTC with Space Vector Modulation, or DTC-SVM, is presented in Fig. 2.5.

Parallel Structure DTC-SVM

In work done in [34], two DTC-SVM schemes are compared, namely cascaded and a parallel structure DTC-SVM. It is concluded that the parallel structure is superior in terms of controller design flexibility and sensitivity to noise. The parallel structure of DTC-SVM will thus be implemented in this thesis. A diagrammatic representation of parallel structure DTC-SVM is shown in Fig. 2.6.

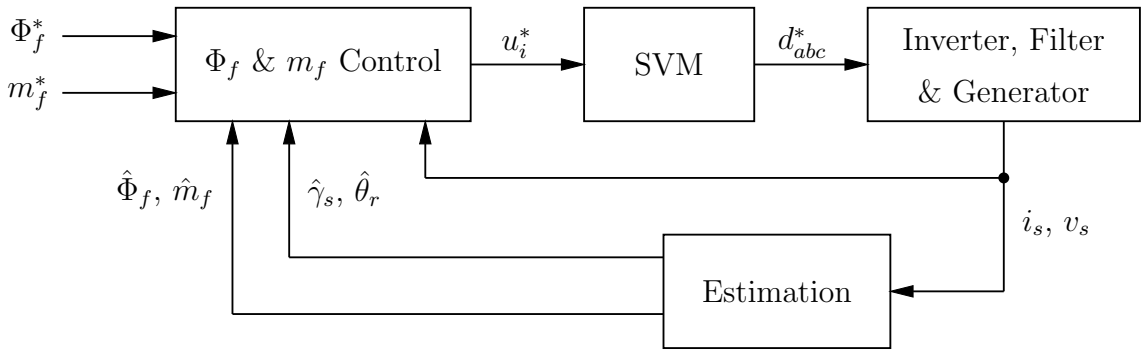


Figure 2.5: *A basic representation of the structure of DTC-SVM.*

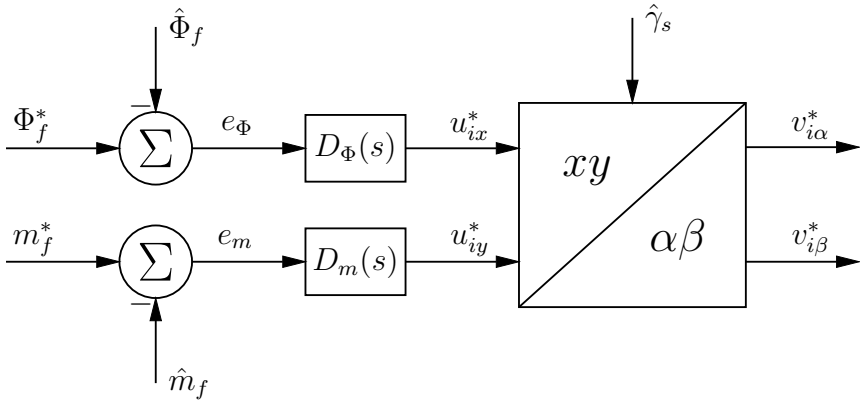


Figure 2.6: *Controller representation of parallel-structure DTC-SVM.*

The controllers in the diagram for the flux and torque are $D_\Psi(s)$ and $D_m(s)$, respectively. Generally these are PI-based controllers, because of their simplicity in implementation and design/tuning, but other controller structures can also be implemented, e.g. lead-lag compensators.

2.3.2 Approximate Transfer Functions

It is advantageous to derive mathematical models for the system that will aid in the design of the different controllers. These models will have to be approximated due to the complexity and multi-variable nature of this system. The inclusion of the LC filter significantly increases the complexity of these derivations, and therefore further necessitates approximation.

Building on work done in [34], approximate transfer functions for the system, including the LC input filter with damping resistances, are derived using the system equations expressed in the xy-reference frame (2.7).

The variables to be controlled here are the filter-generator flux linkage and torque, as shown in Fig. 2.6. If we assume for this derivation that the inverter is ideal, and

that the RLC filter is ideal, then the controller outputs are the x and y components of inverter output voltage. It can be shown that the filter-generator torque is mostly determined by the y component of the converter output voltage, and the flux linkage by the x component (see derivation in Appendix B). We therefore need transfer functions from v_{ix} to $|\vec{\Phi}_f|$, and from v_{iy} to m_f.

Simplifying assumptions that are made are that the coupling between the x and y components in (2.7) are neglected, the magnitude of flux linkage $|\vec{\Phi}_f|$ in some cases is assumed to be constant and equal to permanent magnet flux linkage Φ_m and the filter-generator frequency ω_Φ is assumed to be equal to the generator electrical frequency ω_r. Some coefficients have also been simplified during the derivation, based on relative parameter values. A more detailed explanation of the derivation can be found in Appendix B.

The approximate transfer functions are then obtained as

$$\begin{aligned} G_m(s) &= \frac{M_f(s)}{V_{iy}(s)} \\ &= \frac{A_m\left(\beta_3 s^3 + \beta_2 s^2 + \beta_1 s\right)}{\alpha_4 s^4 + \alpha_3 s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0}, \end{aligned} \tag{2.15a}$$

$$\begin{aligned} G_\Phi(s) &= \frac{\Phi_f(s)}{V_{ix}(s)} \\ &= \frac{A_\Phi\left(b_2 s^2 + b_1 s + b_0\right)}{a_3 s^2 + a_2 s + a_1 s + a_0}, \end{aligned} \tag{2.15b}$$

where $\Phi_f(s)$ represents the s-transfer function of filter-generator flux linkage magnitude ($|\vec{\Phi}_f|$).

The coefficients are presented in Table 2.2. These coefficients are expressed in terms of system parameters, of which values for this system are given in Table 5.1.

Due to their complexity, it would be difficult to use these transfer functions in the design of controllers, but they will enable one to observe the significant effect that the higher order poles have on the dynamic performance of the system. The effects of the damping resistances can also be observed using the full transfer functions.

The derived approximate transfer functions in (2.15) can be further simplified as

$$G_m(s) = \frac{A_m s}{s^2 + \alpha_1 s + \alpha_0}, \tag{2.16a}$$

$$G_\Phi(s) = \frac{1}{s + \frac{a_0}{a_1}}. \tag{2.16b}$$

Table 2.2: *Coefficients of derived transfer function models of torque and flux linkage plants.*

Flux Linkage Plant		**Torque Plant**	
Coefficient	*Expression*	*Coefficient*	*Expression*
A_Φ	$\frac{1}{L_f C_f}$	A_m	$\frac{3\Phi_m N_p}{2L_s}$
b_2	$L_f C_f$	β_3	$L_s C_f$
b_1	$R_d C_f$	β_2	$R_d C_f$
b_0	1	β_1	1
a_3	1	α_4	$L_f C_f$
a_2	$\frac{R_d}{L_f}$	α_3	$(R_d + r_f)C_f$
a_1	$\frac{1}{L_f C_f}$	α_2	1
a_0	$\frac{r_f}{L_f C_f L_s}$	α_1	$\frac{r_f}{L_s}$
		α_0	$\frac{N_p \Phi_m}{J}$

The closed loop transfer functions, including the PI controllers, are then obtained as

$$G_{m,CL} = \frac{A_m \left(k_{p,m} s + k_{i,m}\right)}{s^2 + \left(\alpha_1 + A_m k_{p,m}\right) s + \alpha_0 + A_m k_{i,m}}, \tag{2.17a}$$

$$G_{\Phi,CL} = \frac{k_{p,\Phi} s + k_{i,\Phi}}{s^2 + \left(a_0 + k_{p,\Phi}\right) s + k_{i,\Phi}}. \tag{2.17b}$$

where k_i and k_p denote integral and proportional gains of the PI controllers, respectively.

Control theory for second-order systems or simple manual tuning can now be easily applied to the closed loop transfer functions or to obtain the desired transient dynamic response. The maximum gains are further limited by the higher order poles contained in the more accurate transfer functions and any design should be verified for stability.

Digital equivalent transfer functions, which are a more accurate description of reality, are derived from the continuous transfer functions in the following chapter.

2.4 State Estimation

The core performance of a DTC scheme is ultimately determined by the quality and accuracy of its estimation of the system variables[37]. This is due to the fact that the

flux linkage and torque cannot be measured and must thus be obtained by estimation. It is therefore essential that the estimation methods relay as accurate information as possible on the state of the system.

One of the goals of this thesis is to achieve position sensorless control. It is therefore necessary to estimate the rotor position by some means, as the starting point.

The rotor angle is deduced mathematically from the generator back-EMF, as shown in (2.18b). The accuracy of the rotor angle estimation, the coordinate transformations and consequently the flux and torque estimation all depend on the quality of the back-EMF estimation.

$$\begin{bmatrix} e_{s,\alpha} \\ e_{s,\beta} \end{bmatrix} = \Phi_m \omega_r \begin{bmatrix} \cos\theta_r \\ -\sin\theta_r \end{bmatrix}, \tag{2.18a}$$

$$\theta_r = -\arctan\left(\frac{e_{s,\beta}}{e_{s,\alpha}}\right). \tag{2.18b}$$

2.4.1 EEMF Observer

An extended EMF (EEMF) observer is proposed in [38], of which a block diagram is presented in Fig. 2.7. The purpose of this observer is to estimate the generator back-EMF, which is to be used in the calculation of the rotor angle, as seen in (2.18b).

The observer in Fig. 2.7 is in the $\alpha\beta$ reference frame, and the signals are all vectors. Unless otherwise indicated, the signs of the summing functions are positive.

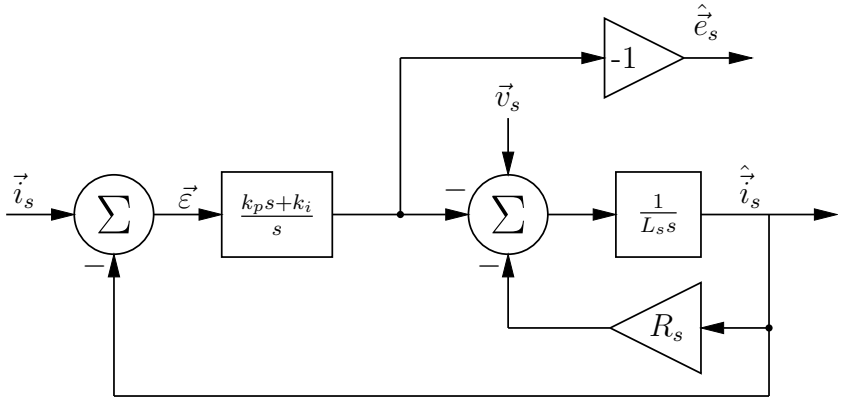


Figure 2.7: *Block diagram representation of the EEMF observer.*

This simple observer, which is a reduced-order observer, delivers surprisingly good performance, despite its simplicity. The precise mathematical analysis will not be dealt with, as this is not within the scope of this thesis.

It is evident from Fig. 2.7 that this observer is somewhat parameter sensitive. The PI current-error compensator section (see Fig. 2.7), however, mitigates this sensitivity to a large degree.

2.4.2 Flux Estimator

A measurement of the stator flux linkage of the machine, or more specifically the filter-motor flux linkage, is needed in the DTC algorithm. As it is not plausible to measure the stator flux directly, estimation techniques are needed to obtain an accurate measurement substitute.

If one considers the simple mathematical models of the PMSG presented in previous sections, it is clear that there are two distinctive models for obtaining an estimate of the stator flux linkage. Equation (2.5c) and equations (2.10c) and (2.10d) describe the voltage model and current model of stator flux linkage, respectively. A commonly used flux observer structure, that combine these models in a single estimator for improved accuracy and dynamic response, is presented in [39]. A block diagram representation of the combined flux observer is shown in Fig. 2.8.

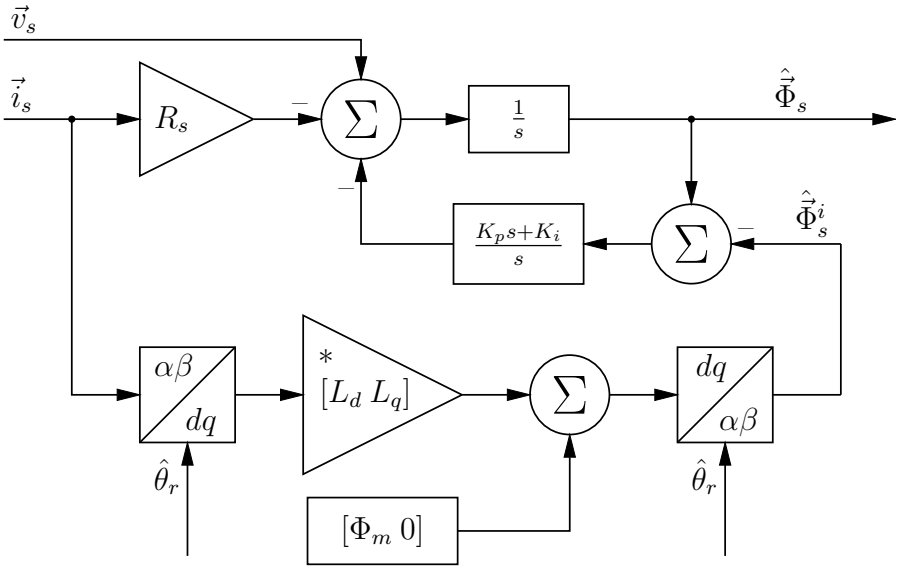


Figure 2.8: *Block diagram representation of the combined voltage-current model flux estimator.*

The observer in Fig. 2.8 is in the $\alpha\beta$ reference frame, and the signals are all vectors, with the exception of the position inputs to the coordinate transformation blocks. Unless otherwise indicated, the signs of the summing functions are positive.

The voltage model is modified to include a compensation term generated by a PI compensator. The error signal of the compensator is the difference between the total estimated stator flux $\hat{\vec{\Phi}}_s$ and the estimated value from the current model $\hat{\vec{\Phi}}_s^i$. Guidelines for obtaining optimal gains for the compensator is given in [39], but manual tuning produces satisfactory results.

2.4.3 Speed estimation

It is necessary to estimate the speed of the rotor for the speed controller and MPPT algorithm. Two methods are presented that are both be considered in this thesis.

Derivative-Filter Method

Ideally, one would be able to differentiate the estimated angle obtained from the EEMF observer. However, the reality of measurement noise, or specifically "estimator noise", will not allow accurate speed estimation due to the fact that differentiation of noise produces unusable results.

The differentiated angle will then have to be filtered to get rid of all the differentiation noise. This causes a delay and possibly a steady state error in the estimation of the rotor speed.

The z-transfer function of estimated rotor speed to estimated rotor position, using a combined derivative and filter, is given in eq. (2.19).

$$H_\omega(z) = \frac{\omega_r}{\theta_r} = \frac{\sigma}{T_s}\frac{(z-1)}{(z-(1-\sigma))}, \tag{2.19}$$

where the value of σ determines how much signal is filtered, i.e., the bandwidth of the filter, and T_s is the switching period of the converter.

MRAS Speed Observer

The concept of a model reference adaptive system (MRAS) can be illustrated by Fig. 2.9. This is an adapted version, as proposed in this thesis, of the MRAS rotor speed observer given in [39]. The difference is that the actual output of the full order flux observer, instead of just the voltage model, is used in the calculation of the error signal. This produces a faster response, better accuracy and less ripple in the estimated speed.

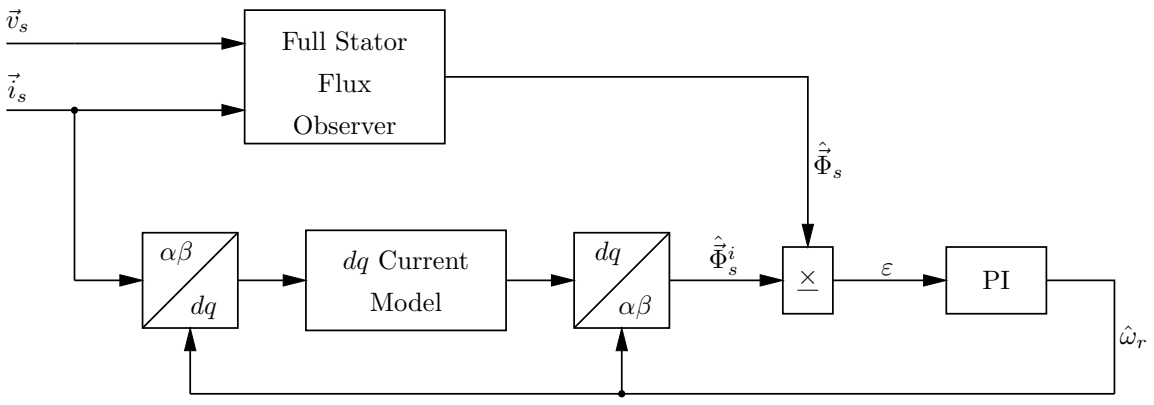


Figure 2.9: *Diagram of an improved MRAS speed observer.*

The observer in Fig. 2.9 is in the $\alpha\beta$ reference frame, and the signals are all vectors, with the exception of the cross product error and the estimated speed.

The MRAS observer adjusts the current model of stator flux linkage by compensating the rotor angular frequency ω_r used in the coordinate transformations, which is then also the output of the MRAS observer. The compensation is done by means of a simple PI compensator.

The error signal ε is generated by taking the vector cross product of the estimated flux from the flux estimator, and the estimated flux from the current model. This vector cross product produces a term that represents the phase difference between the vectors, shown mathematically in (2.20).

$$\varepsilon = \vec{\Phi}_s \times \vec{\Phi}_s^i = |\vec{\Phi}_s||\vec{\Phi}_s^i| \sin \psi \tag{2.20}$$

For small values of ψ, the error signal is proportional to the phase difference between the two estimated fluxes.

2.5 RLC Filter

It is an objective of this thesis to incorporate an RLC filter to filter the converter output voltages. The main reason for this is due to the phenomenon of travelling harmonic waves on long conductors, which causes overvoltages at the generator terminals, and can damage electrical equipment [40]. An added advantage is motor-friendly sinusoidal voltages, which increases the lifetime of the generator. The sinusoidal voltages also make measurement, and ultimately control, easier.

2.5.1 Resonance Damping

LC filters are inherently prone to resonances. It is therefore necessary to add suitable damping to ensure system stability [20]. A commonly used damping technique is to include a damping resistor in series with the filter capacitor. This limits the resonant currents generated by the capacitor.

A larger value for this damping resistance means more stability, but also introduces more I^2R losses which impacts efficiency. Efficiency-wise, this is a very ineffective solution, but it proves to be the simplest solution and worth the minor losses.

There are various active damping techniques in existence, as is shown in[20] and [21], but their implementation is complex and stretches beyond the scope of this thesis.

2.6 Wind Turbine Model

The wind turbine and its interaction with the environment is an extremely complex system. A very good in-depth analysis of the mechanical modelling of wind turbine models can be found in [4] and [41], from which most of the following section is adopted. The aim of this study is not the complex modeling and simulation of the mechanical and aerodynamic apects of the WECS, but rather the control aspects on the electrical side. It should therefore be noted that very simple models for the wind as well as the aerodynamic models of the turbine will be used to illustrate the concept.

A representation of a wind turbine system is presented in Fig. 2.10, which will aid in the explanation of a few concepts pertaining to WECSs. The figure represents a three-bladed turbine.

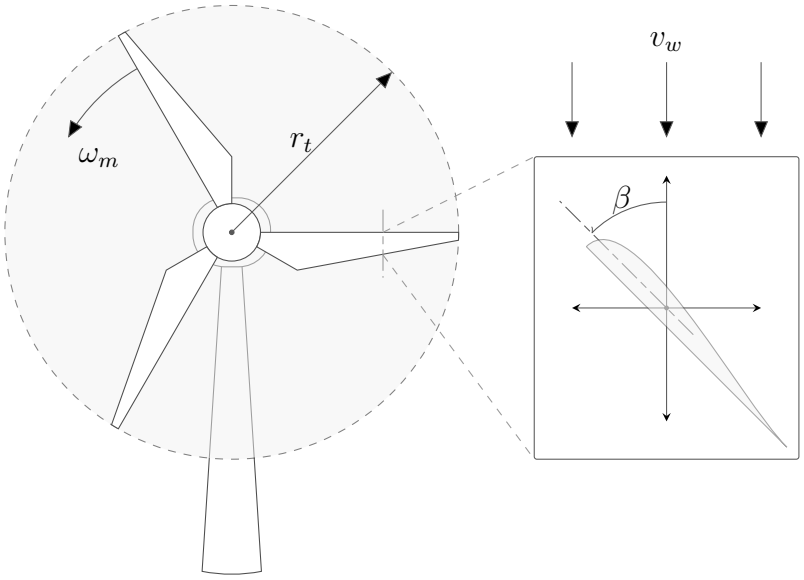


Figure 2.10: *The wind turbine blade model.*

Certain important definitions can now be derived from the diagram in Fig. 2.10. The first is the tip speed ratio (TSR), which represents the ratio of wind speed to turbine rotational speed. This quantity is defined in (2.21) and is of importance when considering the turbine blade power characteristics.

$$\lambda = \frac{\omega_m r_t}{v_w} \tag{2.21}$$

The second parameter is the blade pitch angle β. This is the angle that the individual turbine blades make relative to the incident wind vector, as shown in Fig. 2.10. Both these variables have an effect on the characteristics of the turbine, which are gauged by its power coeffiecient C_p. The power coefficient is a function of both the TSR and the blade pitch angle β.

The total mechanical power delivered by the wind turbine is given by (2.22) [4], where ρ is the air density.

$$P_t = \frac{1}{2}\pi\rho r_t^2 v_w^3 C_p(\lambda, \beta) \tag{2.22}$$

This power coefficient describing the power characteristics of the turbine can be modelled in many different ways [4]. One way is by means of a lookup table consisting of experimentally determined data. A representation of such an experimentally defined power curve is shown in Fig. 2.11. Instead of the power coefficient, the graph in Fig. 2.11 directly gives the mechanical power output of the wind turbine.

An example of power curves for various wind speeds are presented in Fig. 2.11. These are contours on a plane that moves on the wind speed axis of the surface plot.

This thesis is specifically aimed at low to medium power generation levels for which fixed-pitch turbines are most commonly used, as is the case for this thesis. Referring to Fig. 2.10, the blade pitch angle is kept constant for fixed-pitch wind turbines.

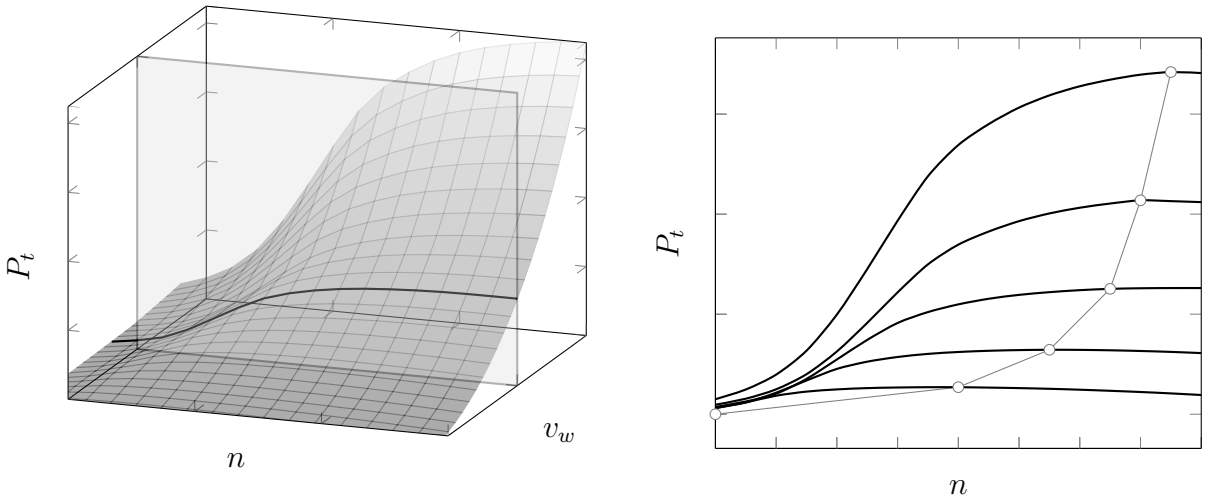


Figure 2.11: *Three-dimensional representation of the wind turbine blade power characteristics. A plane section reveals the turbine power versus rotational speed curve with wind speed as a parameter.*

Maximum power point tracking schemes aim to adjust the TSR in such a way so as to keep the power coefficient at a maximum – in essence following the maximum power point curve. An example of such a maximum power curve can be seen in Fig. 2.11. The absolute maximum value for C_p is called the Betz limit, which is about 0.59; this is a theoretically proven upper limit of the maximum power that can be extracted by any wind turbine[41].

2.7 Gain Scheduling Concept

Gain scheduling (GS), simply defined, is the dividing up of a system's entire operating range into smaller operating regions. This can be seen as a type of linearisation of a non-linear system. A more detailed exposition of GS theory is given in [41]. In this thesis the operation of the wind turbine system will be governed by such gain scheduling. This is needed due to the physical limitations inherent in any physical system.

The GS concept applied in this system is presented in the next chapter.

2.8 Summary & Conclusion

A mathematical model describing the system is developed for use in the design and simulation of the system. It is shown that the inclusion of the RLC filter significantly increases the complexity of the system model. Nevertheless, a very intuitive method for the derivation of approximate transfer functions is presented.

A theoretical explanation, with supporting literature, is given for each of the aspects and components that either comprise or has relevance to this system.

Chapter 3

Design of Controllers, Estimators and Components

In this chapter the design methodologies and specific design considerations of the controllers, estimators, physical components and other sub-systems comprising the system are presented. The chapter commences with the derivation of the equivalent digital controllers and transfer functions of the torque and flux plants defined in Section 2.3.2. The design methodology for the speed controller and observers are then motivated and presented.

The RLC filter design is done, after which the resonance effects of the damping resistance are investigated. The gain scheduling (GS) concept is presented and discussed, after which the maximum power point controller is derived.

3.1 Digital Controller Design

Approximate transfer functions for the torque and flux plants are derived in Section 2.3.2. These are necessary to design the the gains of the PI controllers for the plants. The practical setup, however, consists of a continuous time system controlled by a power electronic converter. It is then by implication a digital controller, necessitating the need to approach the design of the controller from a digital control perspective.

3.1.1 Digital Torque Plant Derivation and Controller

The equivalent digital transfer function of the torque plant is obtained by applying the z-transform to the continuous time transfer function model obtained in Section 2.3.2, and multiplying by the zero order hold function (ZOH):

$$G_m(z) = \frac{z-1}{z} \cdot \mathcal{Z}\left\{\frac{G_m(s)}{s}\right\} \tag{3.1}$$

$$= \frac{z-1}{z} \cdot \mathcal{Z}\left\{\frac{A_m}{s^2+\alpha_1 s+\alpha_0}\right\} \tag{3.2}$$

$$= \frac{z-1}{z} \cdot \mathcal{Z}\left\{\frac{A_m}{(z+\sigma_1)(z+\sigma_2)}\right\}., \tag{3.3}$$

where

$$\sigma_{1,2} = \frac{\alpha_1}{2} \mp \sqrt{\frac{\alpha_1^2}{4} - \alpha_0}. \tag{3.4}$$

The following transformation pair (for an *overdamped* system) is obtained from a standard table:

$$\frac{1}{(z+a)(z+b)} \Longleftrightarrow \frac{z\left(\mathrm{e}^{-aT_s} - \mathrm{e}^{-bT_s}\right)}{(b-a)\left(z-\mathrm{e}^{-aT_s}\right)\left(z-\mathrm{e}^{-bT_s}\right)}. \tag{3.5}$$

By applying (3.5) to (3.3), the transfer function becomes

$$G_m(z) = \frac{K_m(z-1)}{(z-p_{m,1})(z-p_{m,2})}. \tag{3.6}$$

The expressions for the gain K_m and poles p_m of the transfer function are

$$K_m = \frac{A_m\left(\mathrm{e}^{-\sigma_1 T_s} - \mathrm{e}^{-\sigma_2 T_s}\right)}{(\sigma_2 - \sigma_1)}, \tag{3.7a}$$

$$p_{1,m} = \mathrm{e}^{-\sigma_1 T_s}, \tag{3.7b}$$

$$p_{2,m} = \mathrm{e}^{-\sigma_2 T_s}. \tag{3.7c}$$

The digital equivalent transfer function of the proportional integral (PI) controller to be designed and used in the torque control loop, approximated by bilinear digital integration, is

$$D_m(z) = \frac{\lambda_{1,m} z + \lambda_{0,m}}{z-1}., \tag{3.8}$$

with

$$\lambda_{1,m} = \left(\frac{T_s}{2} k_{i,m} + k_{p,m}\right), \tag{3.9a}$$

$$\lambda_{0,m} = \left(\frac{k_{i,m} T_s}{2} - k_{p,m}\right). \tag{3.9b}$$

The equivalent digital control diagram for the torque controler, including a one sample period computation delay [33], is shown in Fig. 3.1.

The closed loop transfer function from m_f^* to m_f is then obtained as

$$G_{m,CL}(z) = \frac{D_m(z) \cdot \frac{1}{z} \cdot G_m(z)}{1 + D_m(z) \cdot \frac{1}{z} \cdot G_m(z)} \tag{3.10}$$

$$= \frac{K_m \lambda_{1,m}\left(z + \frac{\lambda_{0,m}}{\lambda_{1,m}}\right)}{z^3 - (p_{1,m} + p_{2,m}) z^2 + (K_m \lambda_{1,m} + p_{1,m} p_{2,m}) z + K_m \lambda_{0,m}}. \tag{3.11}$$

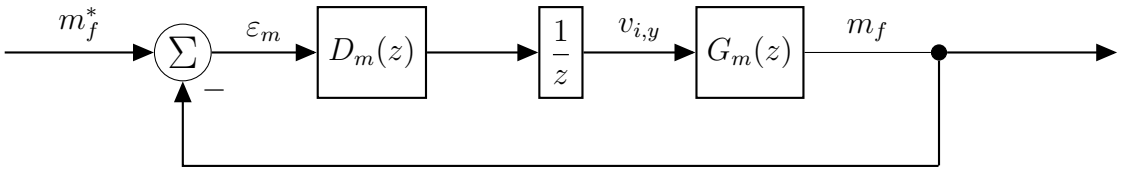


Figure 3.1: *Digital torque controller diagram.*

This approximate digital transfer function can now be used in the design of the controller parameters. Only a manual tuning method is used to obtain the gains for this controller.

3.1.2 Digital Flux Linkage Plant Derivation and Controller

The equivalent digital transfer function of the flux linkage plant is obtained in the same way as the torque plant:

$$G_\Phi(z) = \frac{z-1}{z} \cdot \mathcal{Z} \cdot \left\{ \frac{G_\Phi(s)}{s} \right\} \tag{3.12}$$

$$= \frac{z-1}{z} \cdot \mathcal{Z} \cdot \left\{ \frac{1}{s\left(s + \frac{a_0}{a_1}\right)} \right\}. \tag{3.13}$$

The following transformation pair is obtained from a standard table

$$\frac{1}{s\,(s+a)} \Longleftrightarrow \frac{z\left(1-\mathrm{e}^{-aT_s}\right)}{a\,(z-1)\left(z-\mathrm{e}^{-aT_s}\right)}. \tag{3.14}$$

By applying (3.14) to (3.13), the transfer function becomes

$$G_\Phi(z) = \frac{K_\Phi}{z - p_\Phi}. \tag{3.15}$$

The expressions for the gain K_Φ and pole p_Φ of the transfer function are

$$K_\Phi = \frac{1 - \mathrm{e}^{-\frac{a_0}{a_1}T_s}}{\left(\frac{a_0}{a_1}\right)}, \tag{3.16a}$$

$$p_\Phi = \mathrm{e}^{-\frac{a_0}{a_1}T_s}. \tag{3.16b}$$

The digital equivalent transfer function of the proportional integral (PI) controller to be designed and used in the flux linkage control loop, approximated by bilinear digital integration, is

$$D_\Phi(z) = \frac{\lambda_{1,\Phi} z + \lambda_{0,\Phi}}{z-1}. \tag{3.17}$$

with

$$\lambda_{1,\Phi} = \left(\frac{T_s}{2} k_{i,\Phi} + k_{p,\Phi}\right), \tag{3.18a}$$

$$\lambda_{0,\Phi} = \left(\frac{T_s}{2} k_{i,\Phi} - k_{p,\Phi}\right). \tag{3.18b}$$

The equivalent digital control diagram for the flux linkage controller, including a one sample period computation delay [33], is shown in Fig. 3.2.

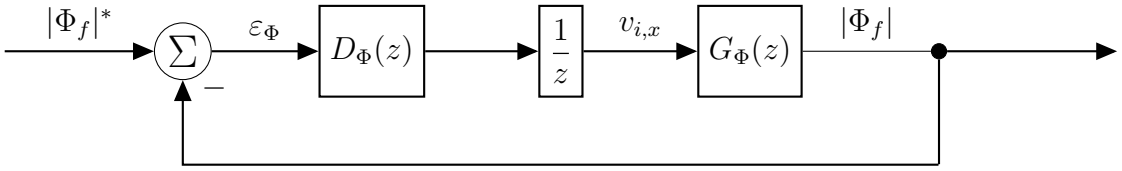


Figure 3.2: *Digital flux linkage controller diagram.*

The closed loop transfer function from $|\Phi_f|^*$ to $|\Phi_f|$ is then

$$G_{\Phi,CL}(z) = \frac{D_\Phi(z) \cdot \frac{1}{z} \cdot G_\Phi(z)}{1 + D_\Phi(z) \cdot \frac{1}{z} \cdot G_\Phi(z)} \tag{3.19}$$

$$= \frac{K_\Phi \lambda_{1,\Phi} \left(z + \frac{\lambda_{0,\Phi}}{\lambda_{1,\Phi}}\right)}{z^3 - (1 + p_\Phi) z^2 + (p_\Phi + K_\Phi \lambda_{1,\Phi}) z + K_\Phi \lambda_{0,\Phi}}. \tag{3.20}$$

Just like the torque plant equivalent transfer function, the flux plant equivalent transfer function can be used to design optimal gains for the flux linkage controller. These transfer functions can be used to do a detailed analytical analysis of the control system, however, due to time constraints, this has not been done for this thesis. The controller gains were instead determined by manual tuning.

3.1.3 Final Controller Gains

The final gains for the digital controllers are given in Table 3.1.

Table 3.1: *Final controller gains for the torque and flux linkage controllers.*

Parameter	Value
$k_{p,m}$	0.5
$k_{i,m}$	50
$k_{p,\Phi}$	2000
$k_{i,\Phi}$	5000

These gains are further limited by physical non-linearities that have not been taken into account in the model, so even if the approximate transfer functions are stable for better or faster responses, the physical system may become unstable. Another very important limit on the controller gains is the inverter output voltage limit, which is

loosely determined by the DC bus voltage. This should therefore be taken into account when the controllers are designed.

Controller Response

The physical parameters of the system are given in Table 3.2.

Table 3.2: *Physical parameters of the system.*

Parameter	Value
Rated line voltage ($V_{LL,\text{rated}}$)	400 V_{rms}
Rated current (I_{rated})	35 A_{rms}
Rated power (P_{rated})	15 kW_{rms}
Rated rotational speed (n_{rated})	150 rpm
Pole pairs (N_p)	20
Stator resistance (R_s)	0.4 Ω
Stator inductance (L_s)	10 mH
PM flux linkage (Φ_m)	1.0395 Wb-t
Filter inductor resistance (r_f)	30 $m\Omega$
Turbine moment of inertia (J)	300 kg·m^2
Switching period (T_s)	100 μs

The unit step response of the closed loop torque controller, for the system parameters in Table 3.2, is shown in Fig. 3.3. The response is very fast, with about 5% overshoot, and a 2% settling time of about 15 ms.

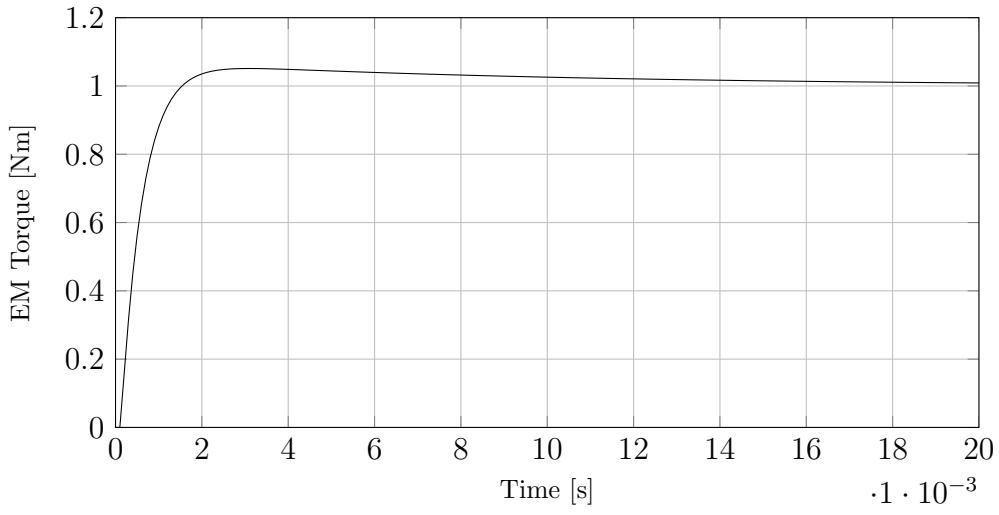


Figure 3.3: *Step response of the closed loop torque controller model.*

The unit step response of the closed loop flux linkage controller, for the same system, is shown in Fig. 3.4. The response is very fast, with no overshoot, and a rise time of about 1 ms.

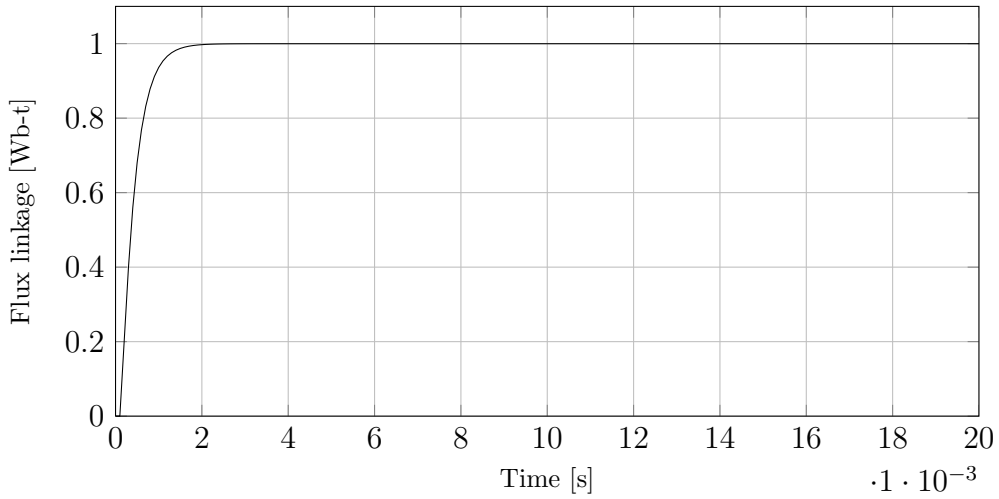


Figure 3.4: *Step response of the closed loop flux linkage controller model.*

3.2 RLC Filter

This design incorporates aspects of a comprehensive design strategy put forward in [42] and [18]. The parameters to be designed are the filter inductance L_f, the filter capacitance C_f and the damping resistance R_d. The resistance of the filter inductor represents filter losses, and can be neglected in this design procedure. The damping resistance is also neglected in the initial determination of the inductor and capacitor values. The effect thereof on the overall filter resonance is demonstrated later in this section.

The parameters of the system are given in Table 3.2. Base values to be used in the design of the filter component values are calculated as

$$\begin{aligned} Z_b &= \frac{V_{\text{rated}}^2}{P_{\text{rated}}} \\ &= \frac{\left(400\sqrt{\frac{2}{3}}\right)^2}{15 \times 10^3} \\ &\simeq 7.1\,\Omega, \end{aligned} \tag{3.21}$$

and

$$\begin{aligned} C_b &= \frac{1}{\omega_{max} Z_b} \\ &= \frac{1}{2\pi 50 Z_b} \\ &\simeq 448\,\mu\text{F}, \end{aligned} \tag{3.22}$$

where V_{rated} is the rated phase to neutral voltage.

The choice of capacitor is limited by the maximum allowable reactive power to be sourced, and affects the power factor of the machine. This is usually kept to within 5% of the rated machine power [42]. The fundamental frequency used to calculate the capacitor base value in (3.22) should be the maximum in the proposed operating range, which is $\omega_{max} = 50$ Hz, in this case.

$$\begin{aligned} C_f &\leq xC_b \\ &= 0.05 \times 448 \times 10^{-6} \\ &\simeq 22\,\mu\text{F} \end{aligned} \tag{3.23}$$

Let x be the percentage (fraction) of allowable reactive power, then (3.23) is used to calculate the maximum allowable value for the filter capacitor. This imposes an upper bound of approximately 22 μF on the value of the filter capacitor.

An upper bound on the value of the inductance is determined by the maximum allowable voltage drop across the filter inductor. This is usually less than 10% of the machine stator inductance [42].

$$\begin{aligned} L_{f,max} &= 0.1 \times L_s \\ &= 1000\,\mu\text{F}. \end{aligned} \tag{3.24}$$

Resonance must be avoided to mitigate the chances of the system becoming unstable. The filter resonant frequency must therefore be chosen to be far away from dominant frequencies in this system. So a factor of 5 is chosen for both the upper and lower bounds:

$$f_{res}^{max} = \frac{1}{5} f_s, \tag{3.25a}$$

$$f_{res}^{min} = 5 f_0. \tag{3.25b}$$

A simplified form of the equations describing the resonance frequency will be used in the design, neglecting the damping resistance.

$$f_{res} = \frac{1}{2\pi} \sqrt{\frac{1}{C_f} \left(\frac{1}{L_s} + \frac{1}{L_f} \right)}, \tag{3.26}$$

$$\frac{1}{(2\pi f_{res}^{max})^2 - \frac{1}{L_s}} < L_f < \frac{1}{(2\pi f_{res}^{min})^2 - \frac{1}{L_s}}. \tag{3.27}$$

If we choose a value for the filter capacitor, so that $C_f = 15\,\mu\text{F}$, then according to (3.27) (Note that the upper bound imposed by (3.24) is the lowest constraint)

$$L_{f,min} \simeq 440.78\,\mu\text{H} \tag{3.28a}$$

It is therefore concluded that a good value for the filter inductance is $L_f = 600\,\mu\text{H}$, keeping it as low as possible so as to limit the voltage drop over it.

According to [43], the optimal damping resistance can determined by the following equation, which has been derived based on the critical damping point of the system:

$$\begin{aligned} R_d &\geq 2\sqrt{\frac{L_f}{C_f}} \\ &\simeq 7\,\Omega \end{aligned} \tag{3.29}$$

A good standard value for the damping resistance is then $R_d = 10\,\Omega$. To estimate the losses in the damping resistance, we start by calculating the RMS value of filter capacitor current for the rated conditions in Table 3.2; the filter capacitor RMS current is then

$$I_c = \frac{V_o}{\sqrt{R_d^2 + \left(\frac{1}{\omega C_f}\right)^2}} \tag{3.30}$$

$$= \frac{\frac{400}{\sqrt{3}}}{\sqrt{10^2 + \left(\frac{1}{2\pi 50 \cdot 15 \times 10^{-6}}\right)^2}} \tag{3.31}$$

$$\simeq 1.08\,\text{A}_{rms}. \tag{3.32}$$

Where V_o is the RMS value of the RLC filter output voltage, in the stationary reference frame. The losses in the damping resistance are then simply calculated as

$$P_{R_d} = I_c^2 R_d \simeq 11.7\,\text{W}_{rms}. \tag{3.33}$$

Note that this calculation does not take higher-order harmonics into account, which may also contribute significantly to losses. A good margin must therefore be chosen when considering the power rating of the damping resistors.

The losses calculated in (3.33) are very small compared to the rated power of the generator. The inclusion of the damping resistor therefore does not have a significant effect on the efficiency of the system.

The final values of the RLC filter components are summarised in Table 3.3.

Table 3.3: *Final component values for the RLC filter.*

Parameter	Value
R_d	10 Ω
C_f	15 μF
L_f	600 μH

3.3 Filter Resonance

The resonance characteristics of the filter is of importance, as it will aid in the design to mitigate instabilities caused by said resonance. The resonant frequency of the filter is described as the frequency at which the input voltage and current to the circuit are in phase. This is the same as saying that the imaginary part of the equivalent input impedance of the circuit in Fig. 3.5 is zero, or that the phase angle is zero.

A simplified representation of the system is shown in Fig. 3.5. Notice that the parasitic resistance of the filter inductance and the stator winding resistance have been omitted, as their effect on the resonance characteristics of the filter is negligible. Refer to Table 3.2 for the values of specific system parameters.

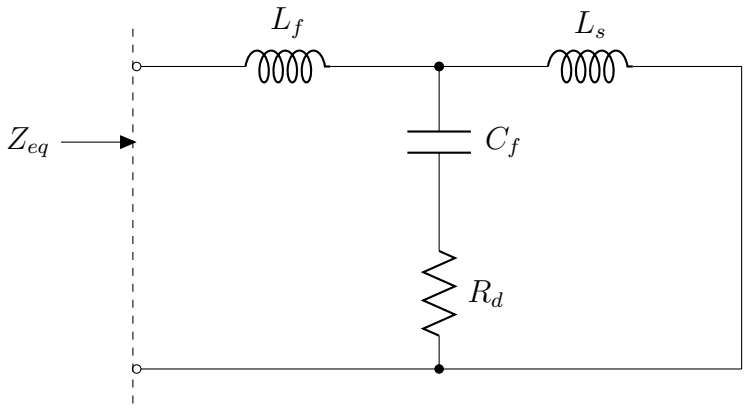


Figure 3.5: *An equivalent lumped-element circuit diagram of the filter and generator, illustrating the input impedance as seen from the converter.*

The equivalent input impedance to the circuit in Fig. 3.5 can be obtained as

$$\tilde{Z}_{eq} = \frac{(-\omega^2 L_s C_f R_d - \omega^2 L_f C_f R_d) + j(\omega L_s + \omega L_f - \omega^3 L_f L_s C_f)}{(1 - \omega^2 L_s C_f) + j\omega C_f R_d}. \tag{3.34}$$

We equate the angle of the impedance to zero, which finally yields the quartic equation in ω as

$$\omega^4 L_s L_s^2 C_f^2 + \omega^2 \left(L_s C_f^2 R_d^2 + L_f C_f^2 R_d^2 - 2 L_f L_s C_f - L_s^2 C_f\right) + L_s + L_f = 0. \tag{3.35}$$

The solution to (3.35) yields four solutions, of which only two are real:

$$\omega_{res} = \frac{\sqrt{\pm\sqrt{\xi_1 + L_s^4} - \xi_2 + L_s^2 + 2L_sL_f}}{L_s\sqrt{2C_fL_f}}, \tag{3.36a}$$

$$\xi_1 = R_d^4(C_f^2L_s^2 + 2C_f^2L_fL_s + C_f^2L_f^2) + R_d^2(-2C_fL_s^3 - 6C_fL_fL_s^2 - 4C_fL_f^2L_s), \tag{3.36b}$$

$$\xi_2 = R_d^2(C_fL_s + C_fL_f), \tag{3.36c}$$

where ξ_1 and ξ_2 represent the damping terms introduced as a result of the filter capacitor damping resistance. The two solutions in (3.36a) represent the low and high resonances, respectively, of which the latter is of concern.

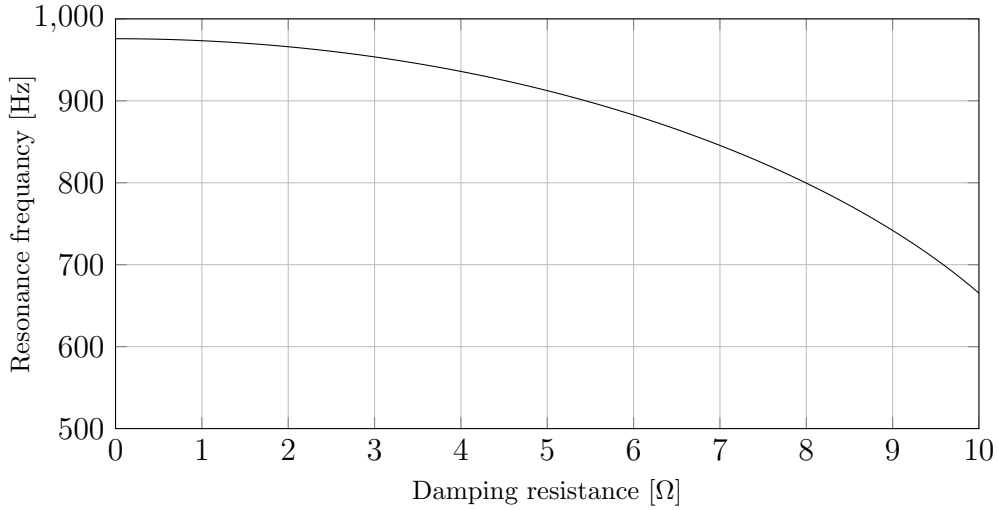


Figure 3.6: *Lowering of system resonance frequency due to inclusion of damping resistance.*

For the design of the LC filter elements, the damping resistance has been be omitted for simplicity. This is done with the assumption that the damping resistance will not have a major effect on the resonance frequency. However, the diagram in Fig. 3.6 shows that the resonance frequency decreases with an increase in the value of the damping resistance. As the function of the damping resistance is to damp out resonances, this is not a major concern, but it must be kept in mind so as to not design counter productively.

3.4 Speed Controller

The speed controller is also a simple PI controller that generates a torque command for the torque controller, based on the rotor speed error. The bandwidth of the speed controller is limited not only by the inertia of the turbine blades, but also by the quality

of the rotor speed estimation. It does, however, not have to be a very fast controller, due to the fact that wind speed variation is slow. No analytical design is done for the speed controller; the optimal gains are obtained by manual tuning.

3.5 EEMF Observer and Flux Linkage Observer

No analytical design has been done for the observers; only manual tuning of the gains to obtain satisfactory results. The final gains of the observers, including the speed controller, are given in Table 3.4, with the subscripts p and i denoting proportional and integral gains, respectively.

Table 3.4: *Final gains of the EEMF observer, flux observer and speed controler.*

	EEMF Observer	**Flux Observer**	**Speed Controller**
k_p	20	1000	10
k_i	1000	5000	5

3.6 Gain Scheduling Algorithm

This system is in essence a generation system, and the controller is therefore semi-passive as it waits for the changes from the energy source to determine what would be the appropriate control action to take.

The operating region of the system is divided into smaller operating regions for which specific conditions need to be satisfied.

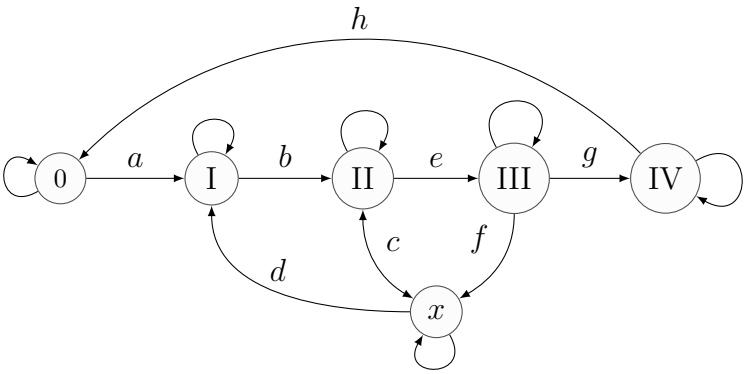


Figure 3.7: *State diagram of gain scheduling algorithm.*

For this system a simple gain scheduling (GS) algorithm is developed to facilitate the operation of the system. A state diagram representation of the GS algorithm is

given in Fig. 3.7.

The states in the diagram can be likened to the different operating regions of the WECS, which the system traverses as the wind speed changes. A different control strategy must be applied to each state, including the transitions between the states.

A summary these states and their descriptions is given in Table 3.5. The letters a through f represent transitions between the states that occur when a certain set of conditions are satisfied. The arrows on the transitions show the direction in which a transition can occur.

Table 3.5: *Description of the states of the GS algorithm.*

State	Name & Description
0	*Pre-states*: The generator is desynchronised and the system waits for the rotor speed (representing the wind speed) to be below a certain threshold, and then transitions to the next state once this threshold is breached.
I	*Pre-cut-in*: The sytem waits for the rotor speed to exceed a threshold before transitioning to the next state.
II	*Cut-in*: After synchronising with the generator, the generator speed is controlled at cut-in speed until the wind speed breaches a determined threshold, which is determined by monitoring electromagnetic torque generation. Transitioning can either be towards state x:*Cut-out* or state III:*MPPT*
III	*MPPT*: The generator torque is controlled as a function of rotor speed to achieve MPPT. This is the variable speed state of the WECS where it is desirable to spend most of the time.
IV	*Max Speed*: When the maximum speed constraint of the system is reached, the generator speed is controlled at this maximum speed. Transitioning can either be back towards state III or to state 0 (desynchronisation), depending on wind conditions.
x	*Cut-out*: The generator is controlled at a certain cut-out speed after certain conditions have been met in states II and III, and waits either for the wind speed to decrease sufficiently for desynchronisation, or to increase sufficiently to return to state II.

3.7 Maximum Power Point Tracking

There are various methods to track the maximum power points of the available wind energy, but they can be broadly placed into three categories, namely, anemometer

methods, hill climb search methods and calculation methods [44].

The calculation method is chosen because of its simplicity and ease of implementation, and according to [44], it's performance is comparable to that of the anemometer method. It is, however, dependent on the blade power characteristics and the overall system model; model inaccuracies can thus deteriorate performance.

The method presented below is similar to work done in [45], but differs in the definition of the torque characteristics. To obtain the relationship between torque and angular speed, we start with a set of blade power data, given here as a matrix of data points $\overline{P}_o$:

$$\overline{P}_o = \begin{bmatrix} p_{00} & \cdots & p_{0m} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nm} \end{bmatrix}_{n \times m}, \tag{3.37a}$$

$$\vec{V}_w = \begin{bmatrix} v_{w,0} & v_{w,1} & \vdots & v_{w,m} \end{bmatrix}^T_{m \times 1}, \tag{3.37b}$$

$$\vec{\Omega}_t = \begin{bmatrix} \omega_{t,0} & \omega_{t,1} & \vdots & \omega_{t,n} \end{bmatrix}^T_{n \times 1}, \tag{3.37c}$$

where the rows and columns of the matrix $\overline{P}_o$ represent the wind speed v_w and mechanical rotor speed ω_m, respectively. The vectors $\vec{V}_w$ and $\vec{\Omega}_t$ contain the values of corresponding wind speed and turbine rotational speed, respectively.

For every wind speed there is a maximum power that can be extracted from the available wind, which is dependent on the rotational speed of the turbine. If we step through the rows of the blade power matrix $\overline{P}_o$, i.e., the wind speed is varied incrementally, a new vector containing the maximum power for each wind speed can be formed:

$$\vec{P}^k_{max} = max[\overline{P}_o(k)(\dots)],\ k = 0, 1, \dots, n. \tag{3.38}$$

We can now use least squares regression, with the rotational speed vector $\vec{\Omega}_t$ as input, and the maximum power vector $\vec{P}_{max}$ as the output, to fit a polynomial that describes the maximum power curve (refer to Section 2.6). We know that the power produced by the turbine is proportional to the cube of the wind speed (see (2.22)). It is therefore best to fit a polynomial of the third degree to the data:

$$P_o = c_3\omega_m^3 + c_2\omega_m^2 + c_1\omega_m + c_0. \tag{3.39}$$

The relationship between torque and power is

$$P_o = m_L\omega_m. \tag{3.40}$$

Thus, the expression in (3.39) can be written as

$$\begin{aligned} m_L &= c_3\omega_m^2 + c_2\omega_m + c_1 + \frac{c_0}{\omega_m} \\ &\simeq c_3\omega_m^2 + c_2\omega_m. \end{aligned} \tag{3.41}$$

A further assumption that is made is that the mechanical output power P_o is equal to the generator electrical output power P_e. Substituting (2.1b) into (3.42), we obtain

$$\begin{aligned} m_L &= c_3\left(\frac{\omega_r}{N_p}\right)^2 + c_2\left(\frac{\omega_r}{N_p}\right) \\ &= k_2\omega_r^2 + k_1\omega_r. \end{aligned} \tag{3.42}$$

The purpose of the MPPT controller, based on a torque controller, is to cause the generator to produce the optimal torque, given by (3.42). From (2.1a) it can be seen that, dynamically, the load torque m_L and m_e are not equal. It can, however, be deduced that for steady state[1]

$$m_e \to m_L. \tag{3.43}$$

It can then be stated that the torque reference for the DTC-SVM controller, in the variable speed (or MPPT) region, is

$$m_f^* = f(\hat{\omega}_r) = k_2\hat{\omega}_r^2 + k_1\hat{\omega}_r. \tag{3.44}$$

Where the values of the constants are: $k_1 = \frac{c_3}{N_p^2} = -1.575$ and $k_2 = \frac{c_2}{N_p^2} = 0.0185$. The values for c_3 and c_2 are obtained by executing the maximum power curve fitting operation, as discussed earlier in this section.

3.8 Summary and Conclusion

Digital equivalent transfer functions for the torque and flux linkage plants are derived, as well as equivalent closed loop transfer functions for the design of the controller gains.

The RLC filter design is presented, as well as a full derivation of the resonance frequency including the damping resistance. This shows the effect of the damping resistance on the resonance frequency of the system, which is quite significant. This must be taken into account when designing the RLC filter.

Methodologies for the design of the speed controller gains are presented, as well as the various observer gains. Lastly, the design of the maximum power point control strategy is presented.

[1] In terms of rotor speed; assuming very slow varying wind speeds.

Chapter 4

Simulation Methodology and Results

The theoretical predictions of the system that are made in previous chapters are largely based on simplifications and assumptions. This is needed to construct a solution in an effective way, but it is not an entirely accurate prediction of reality. It is therefore advantageous to harness the power of modern computer technology so that more accurate behaviour of the system can be predicted.

The system is modelled in both MATLAB Simulink and the VHDL-AMS simulation language. The simulation results of both computer models are presented and compared. As VHDL-AMS is relatively new in this field of study, and not that widely implemented as of yet, its results are compared to results produced by MATLAB Simulink which is a much more mature and widely used simulation package.

Only the sensorless direct torque control has been incorporated in the Simulink simulation, as this is the dynamic backbone of this system. The VHDL-AMS simulation, in addition to the dynamic DTC simulation, simulates the MPPT component of the system, including a wind turbine simulator (WTS).

4.1 MATLAB Simulink Simulation Methodology

Simulations of the system are done in Simulink, as well as VHDL-AMS. The Simulink method is a graphical approach, in contrast to the completely coded method of VHDL-AMS. The Simulink simulation models will be presented first, which will serve as a basis for most of the VHDL-AMS models.

The rationale for simulating the system in both Simulink and VHDL-AMS, is so that the integrity of the VHDL-AMS simulation results can be verified against that of

an established standard, which is Simulink.

4.1.1 Top-Level Simulink System Model

The Simulink simulation consists of various blocks that contain the models of the different sub-systems. A simple representation of the main model is given in Fig. 4.1, and illustrates the flow and structure of the simulation. Only the *Filter-generator Model*, *State Estimation & Calculation* and *Direct Torque Control* bocks will be discussed in detail in this chapter. The actual top-level Simulink model of the system can be found in Fig. C.11 if more detail is desired.

For more detail on the *SVM & Inverter/Switching Model* block, refer to Fig. C.1 in Appendix C. The switching capabilities in this simulation are developed using only the standard blocks available within Simulink, as no power electronics toolbox is available.

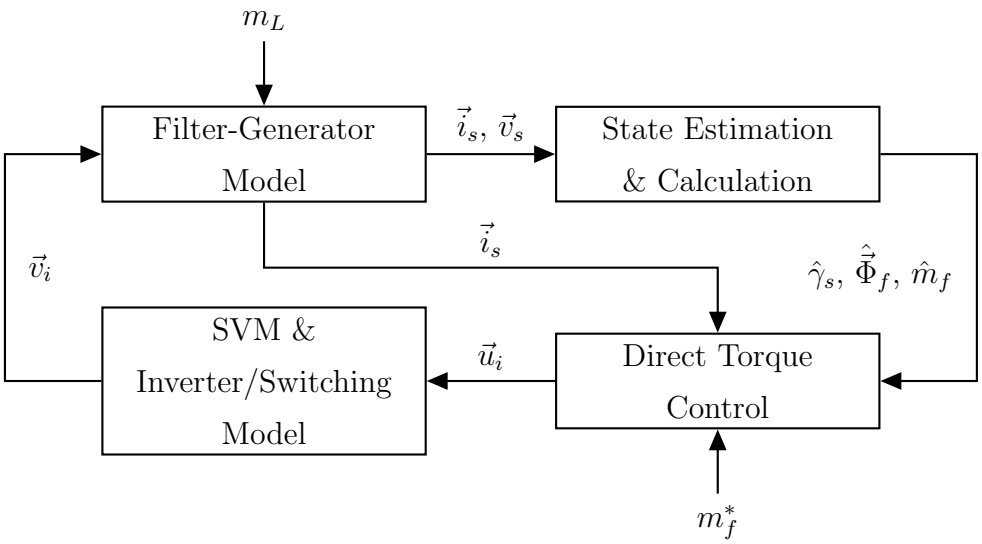
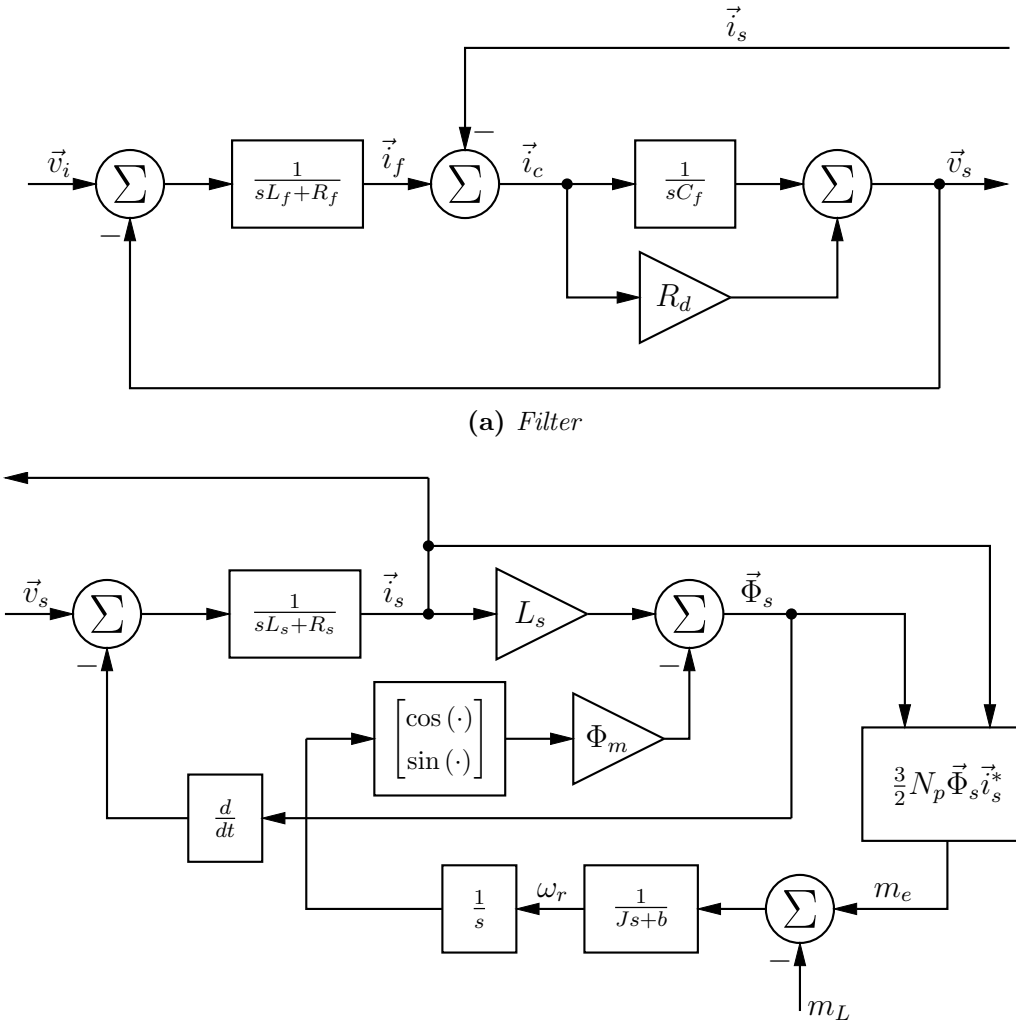


Figure 4.1: *Simplified representation of the top-level Simulink simulation, depicting the structure.*

4.1.2 Filter-Generator Model

The time-domain equations of the system in the stationary ($\alpha\beta$) reference frame (2.5) are transformed to the Laplace domain to obtain the transfer function model of the filter-generator circuit. A representation of this model, as implemented in the Simulink simulation, is given in Fig. 4.2. This representation is somewhat simplified, and only incorporates the mathematical model. Note that friction b is included in the model

for completeness' sake, even though it is zero in the actual simulation. The actual Simulink model can be found in Fig. C.6 if more detail is desired.



(a) *Filter*

(b) *Machine*

Figure 4.2: *Implementation of the filter-generator continuous-time model in Simulink.*

The benefits of transforming the system model to the $\alpha\beta$-frame is evident in the ease of implementation, with no coupling between the components. The models for both components of the stationary reference frame are virtually the same. This is a stationary frame vector model, with the only difference in the components being in the calculation of the generator EMF.

4.1.3 EEMF Observer

A representation of the digital transfer function implementation of the EEMF observer in Simulink is shown in Fig. 4.3. Note that this observer is inherently in the stationary reference frame; the signals in the diagram are therefore all two dimensional. A slight

difference in implementation, as compared to the concept presented in Fig. 2.7, is the removal of the "-1" block before the output of the estimated EMF. This is done away with by changing the signs of the PI controller summming function.

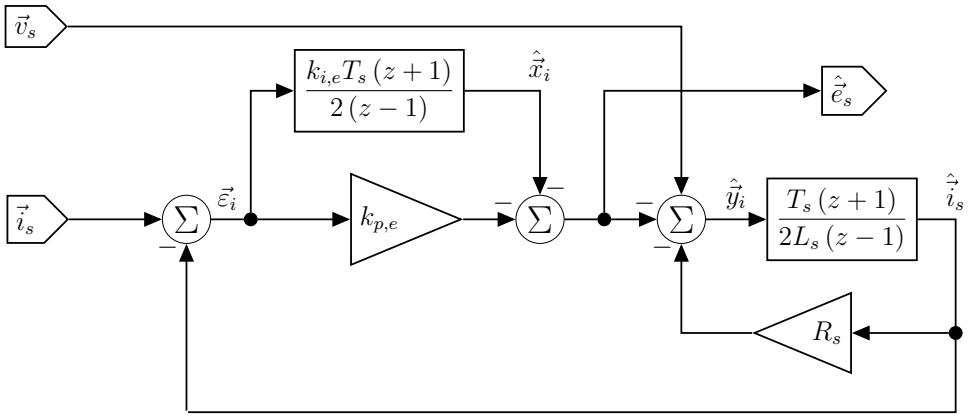


Figure 4.3: *Representation of the EEMF observer of generator EMF, as implemented in the Simulink simulation.*

The chosen digital approximation method for both digital integrators in Fig. 4.3 is the slightly more accurate trapezoidal or Tustin integration method. The motivation for this improved approximation is that of accuracy, as the quality of rotor position estimation is very important. The actual Simulink model can be found in Fig. C.16.

The estimated EMF produced by the EEMF observer in Fig. 4.3 is used in the calculation of the rotor position. The implementation of this calculation in Simulink is presented in Fig. 4.4. Other methods, such as phase-locked loops, can be used to improve the rotor position estimation. This simple method, however, already gives very good performance.

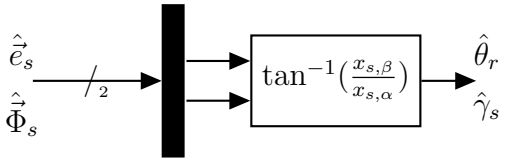


Figure 4.4: *Representation of the rotor position estimation, as implemented in the Simulink simulation.*

4.1.4 Stator Flux Observer

The current model of stator flux as implemented in Simulink is given in Fig. 4.5. The current model of stator flux resides in the rotor synchronously rotating reference frame (dq-frame), and is derived from (2.10c) and (2.10d). This model can be used exclusively

as the estimation method, but it is combined with the voltage model to improve the accuracy of the estimation.

As the measured current is in the $\alpha\beta$-frame, it is first transformed into the dq-frame to be used in the current model. The resulting estimated stator flux linkage is then be transformed back to the $\alpha\beta$-frame. It is here where the importance of the rotor position estimation becomes clear. The actual Simulink models for the $\alpha\beta$-dq and dq-$\alpha\beta$ transformations can be seen in Fig. C.12 and Fig. C.13, respectively.

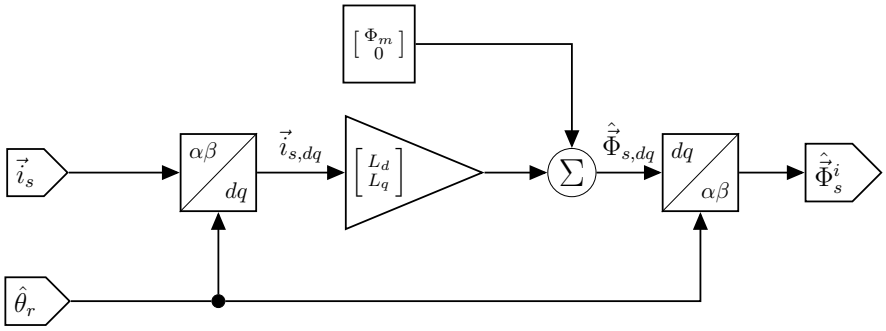


Figure 4.5: *Representation of the current model observer of generator stator flux, as implemented in the Simulink simulation.*

A representation of the combined voltage-current model flux observer, as implemented in the Simulink simulation, is given in Fig. 4.6. This model incorporates the previously discussed current model estimation to obtain a more accurate estimation, by compensating for integration errors and parameter uncertainty. This type of configuration also extends the speed range of the estimator, as the voltage model is more suited to higher frequencies, and the current model to lower frequencies.

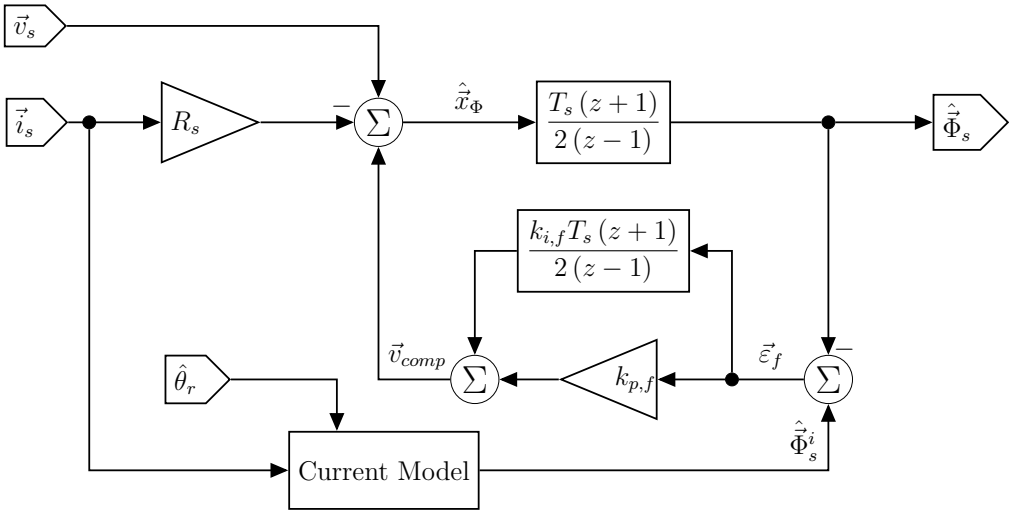


Figure 4.6: *Representation of the generator stator flux linkage observer, as implemented in the Simulink simulation.*

4.1.5 Filter-Generator Flux Linkage and Torque Calculation

The estimated parameters of the filter-generator system can now be calculated by Fig. 4.7 and Fig. 4.8. Concerning the DTC, this is the final step in the estimation process.

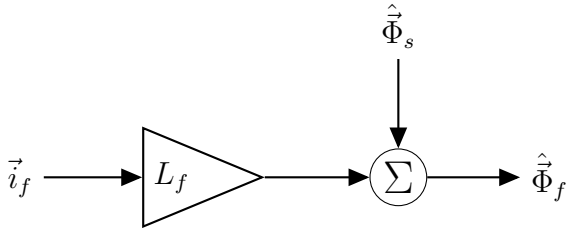


Figure 4.7: *Representation of the filter-generator flux calculation, as implemented in the Simulink simulation.*

The actual Simulink model implementation of these can be found in Fig. C.11. They are not contained within their own blocks, but reside in the top-level simulation model.

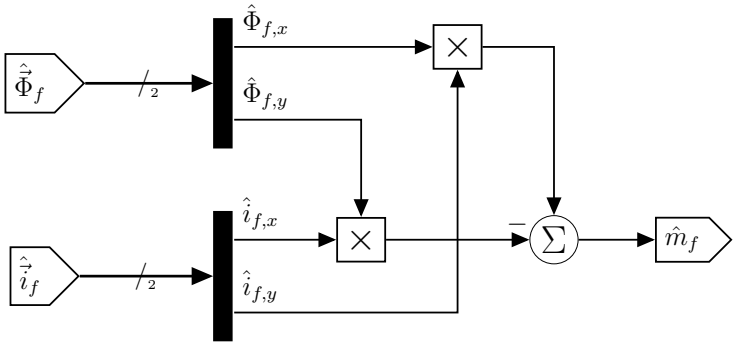


Figure 4.8: *Representation of the filter-generator torque calculation block, as implemented in the Simulink simulation.*

4.1.6 Speed Estimator

A representation of the derivative-filter method speed estimator (refer to (2.19)), as implemented in Simulink, is given in Fig. 4.9. A digital backward difference integration method is used, followed by a *wrap to zero* block to compensate for the wrapping of the estimated angle. The last stage of the estimation is then a simple digital filter. The actual Simulink model can be found in Fig. C.11.

The MRAS speed estimator is not included in this chapter, but can be found in Fig. C.8. It's adaptive current model, and speed-based dq-$\alpha\beta$ coordinate transformation are given in Fig. C.9 and Fig. C.10, respectively.

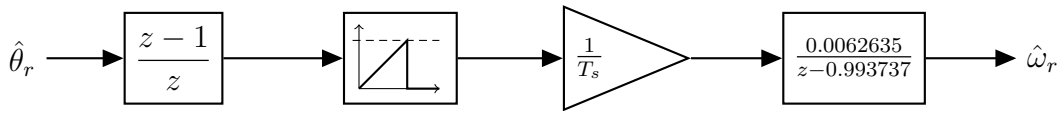


Figure 4.9: *Representation of the derivative-filter method of rotor speed estimation, as implemented in the Simulink simulation.*

4.1.7 Direct Torque Controller

A representation of the digital direct torque controller, as implemented in the Simulink simulation, is given in Fig. 4.10. It must just be noted that a $\frac{\pi}{2}$ angle is subtracted from the $\hat{\gamma}_s$ angle to compensate for the slight difference in transformation matrices. The diagram is a mixture of scalar and vector signals, but the vector signals in the diagram are clearly specified. The actual Simulink model implementation can be found in Fig. C.19.

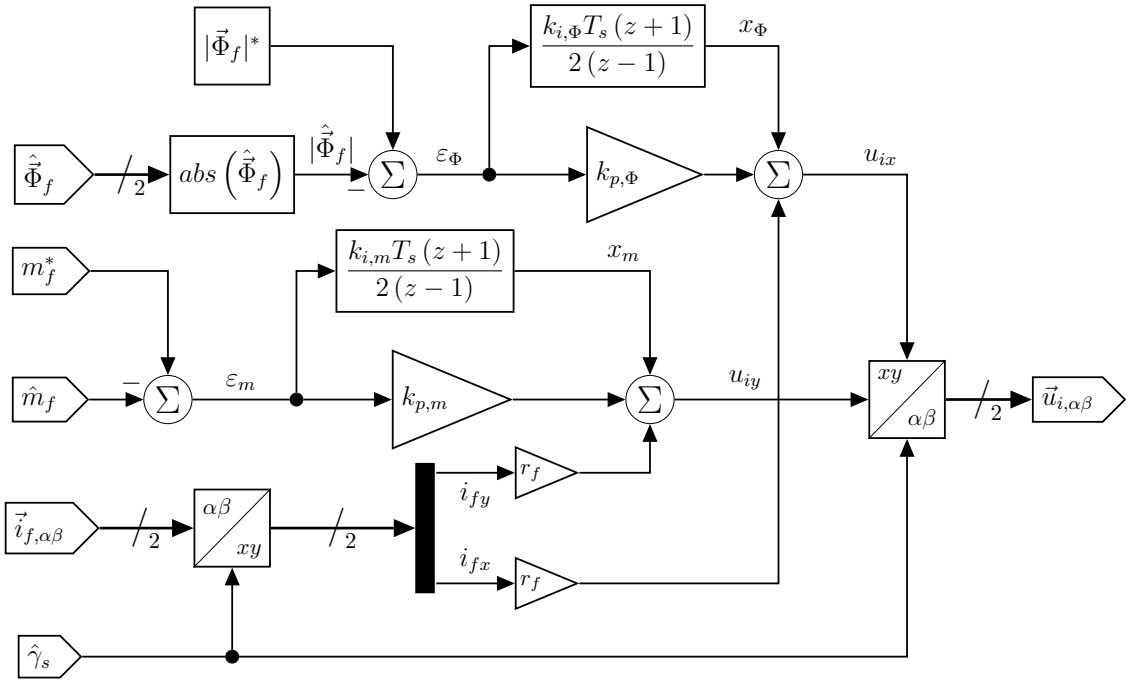


Figure 4.10: *Representation of DTC, as implemented in the Simulink simulation.*

The structure of the DTC represents the mathematical model of the system in the xy-frame(2.7), where the x-component relates to the flux linkage, and the y-component to the torque.

The controlled variables are the filter-generator flux linkage magnitude, and the filter-generator torque. For this system, the flux linkage magnitude is controlled as a constant, which is an assumption of the approximate transfer functions in Section 2.3.2. The flux controller is a simple PI controller that uses the Tustin integration approximation. The flux linkage part of the DTC produces the x-component of the control

voltage.

The torque controller is identical to the flux controller, with the exception of controller gains. The torque part of the DTC produces the y-component of the control voltage, after which the components of the control voltage are finally transformed to the $\alpha\beta$-frame, for use in the SVM for converter control. The actual Simulink implementation of the $\alpha\beta$-xy and xy-$\alpha\beta$ transformations can be found in Fig. C.14 and Fig. C.15, respectively.

4.2 VHDL-AMS Simulation Methodology

The most important part of the simulation study is to implement the system in a VHDL-AMS simulation framework. Several simulation packages supporting VHDL-AMS have been tried and tested, and SystemVision from Mentor Graphics was found to be the most competent package. The only caveat is the numerical solver, which is somewhat slow compared to the Simulink solver. The solver also sometimes fails to converge to a solution, causing the simulation to hang indefinitely; this can be mitigated by appropriate tuning of the solver parameters.

Although SystemVision also caters for graphical simulation design such as Simulink, the aim is to implement the simulation purely in VHDL-AMS. This philosophy will allow the simulation to be easily ported to any other simulation package that supports VHDL-AMS.

A diagram representing the top-level structure of the VHDL-AMS simulation is shown in Fig. 4.11. The actual simulation code is given in Appendix D.

The structure of the VHDL-AMS simulation is what is referred to as a *testbench*, which is a `ENTITY-ARCHITECTURE` combination that provides a framework for the simulation. System-wide parameters are declared within the `ENTITY` declaration, to be used as constants in the simulation body.

The working of the simulation is declared within the `ARCHITECTURE` declaration. The variables are declared in the preamble of the `ARCHITECTURE` while the system models are described within the body.

The physical structure of the system is made up of lumped elements connected by nodes, including simultaneous equations that form part of the generator model. A `PROCESS` statement takes care of digital measurement, control and switch signal generation.

A significant advantage of VHDL-AMS is the ease with which time-domain differential equations can be modelled. This modularity also allows for the simulation model

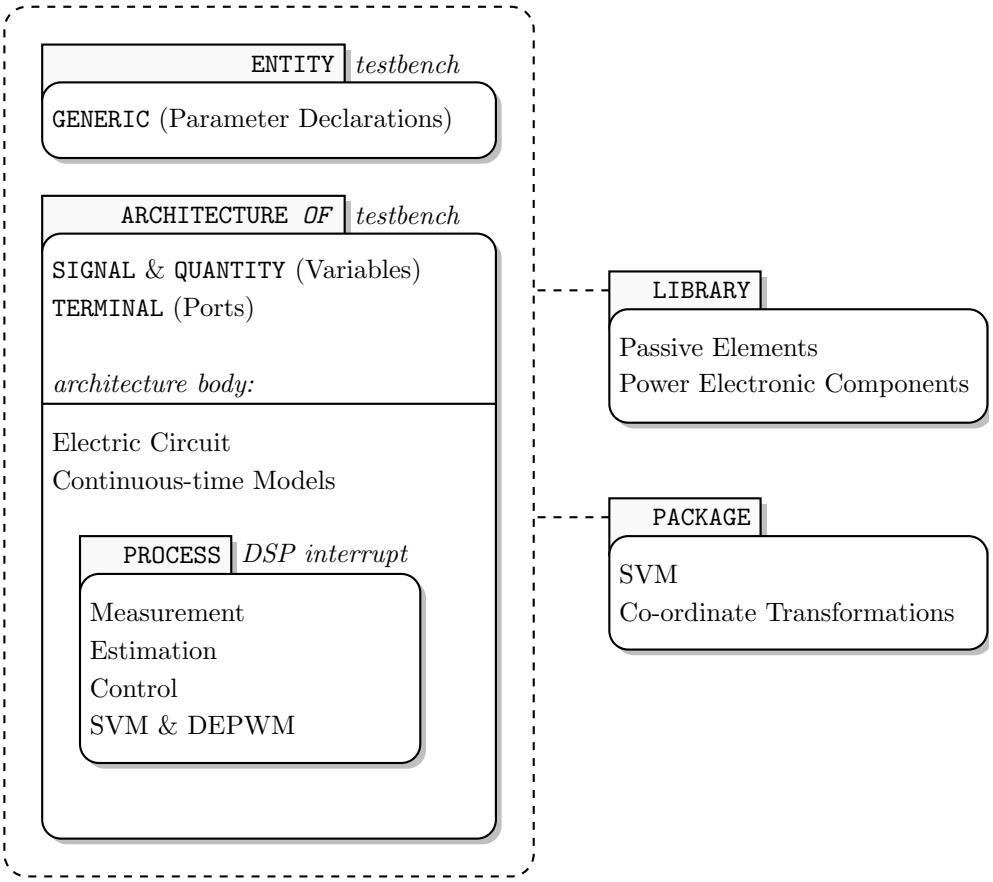


Figure 4.11: *Diagram representing the structure of the VHDL-AMS simulation.*

to represent the physical system much more closely in structure. The digital difference equations can also be input exactly as they will be implemented in practice.

The VHDL-AMS code of the converter and filter-generator model implementation can be found in Fig. D.1. The converter code is given is Fig. D.8. The converter model incorporates IGBT and reverse diode models, given in Fig. D.10 and Fig. D.9, respectively.

The digital diagrams in the previous section can now be used to derive the difference equations that will be used in the VHDL-AMS simulation, as well as the practical implementation on the DSP.

4.2.1 EEMF Observer

The EEMF observer difference equations are derived from the z-domain tansfer function model given in Fig. 4.3. The first stage of the estimation is the compensator section, which ultimately generates the output estimated EMF. These difference equations are obtained from Fig. 4.3. VHDL-AMS allows for new types to be declared, as well as associated operator overloading functions; in this system a *space vector* array type,

and its associated operator functions, is declared for vector calculations.

$$\varepsilon(k) = i_s(k) - \hat{i}_s(k-1), \tag{4.1a}$$

$$y_i(k) = y_i(k-1) + \frac{Ts}{2} \cdot (\varepsilon(k) + \varepsilon(k-1)), \tag{4.1b}$$

$$\hat{e}_s(k) = -(K_p \cdot \varepsilon(k) + K_i \cdot y_i(k)), \tag{4.1c}$$

$$\theta_r(k) = \arctan\left(\frac{\hat{e}_{s\beta}(k)}{\hat{e}_{s\alpha}(k)}\right), \tag{4.1d}$$

Difference equations for the current model section are given in (4.2) .

$$x(k) = u_s(k) - \hat{e}_s(k) - \hat{i}_s(k-1) \cdot R_s, \tag{4.2a}$$

$$\hat{y}_i(k) = \hat{y}_i(k-1) + \frac{T_s}{2} \cdot (x(k) + x(k-1)), \tag{4.2b}$$

$$\hat{i}_s(k) = \frac{1}{L_s} \cdot \hat{y}_i(k). \tag{4.2c}$$

These equations can be implemented *as-is* in the VHDL-AMS simulation, with the exception of delayed values, which have to be stored after each iteration. The VHDL-AMS code for the EEMF observer can be found in Fig. D.2.

4.2.2 Flux Observer

The flux observer has three main parts: the current model, compensator section and voltage model. The difference equations to be used in the VHDL-AMS simulations and in practice are derived from Fig. 4.5 and Fig. 4.6.

The difference equations for the stator flux linkage current model are given in (4.3).

$$\vec{i}_{s,dq}(k) = K_{dq} \cdot \vec{i}_s(k), \tag{4.3a}$$

$$\hat{\vec{\Phi}}_{s,dq}(k) = \vec{i}_{s,dq}(k) \cdot \begin{bmatrix} L_d & L_q \end{bmatrix} + \begin{bmatrix} \Phi_m \\ 0 \end{bmatrix}, \tag{4.3b}$$

$$\hat{\vec{\Phi}}^i_s(k) = K^{-1}_{dq} \cdot \hat{\vec{\Phi}}_{s,dq}(k). \tag{4.3c}$$

The matrices $\overline{K}_{dq}$ and $\overline{K}^{-1}_{dq}$ represent the dq-transformation and its inverse, respectively. All the coordinate transformations used in the VHDL-AMS simulation are implemented in a *package*, of which the code can be found Fig. D.12.

The voltage model output stage including compensation is given in (4.4).

$$\varepsilon(k) = \hat{\vec{\Phi}}^i_s(k) - \hat{\vec{\Phi}}_s(k-1), \tag{4.4a}$$

$$v_{comp}(k) = v_{comp}(k-1) + \varepsilon(k) \cdot \left(k_p + k_i'\right) + \varepsilon(k-1) \cdot \left(k_i' - k_p\right), \tag{4.4b}$$

$$\hat{\vec{x}}_\Phi(k) = \vec{v}_s(k) - R_s \cdot \vec{i}_s + \vec{v}_{comp}(k), \tag{4.4c}$$

$$\hat{\vec{\Phi}}_s(k) = \hat{\vec{\Phi}}_s(k-1) + \frac{T_s}{2} \cdot \left(\hat{\vec{x}}_\Phi(k) + \hat{\vec{x}}_\Phi(k-1)\right), \tag{4.4d}$$

$$\hat{\gamma}_s(k) = \arctan\left(\frac{\hat{\Phi}_{s,\beta}(k)}{\hat{\Phi}_{s,\alpha}(k)}\right). \tag{4.4e}$$

The VHDL-AMS code for the flux linkage observer can be found in Fig. D.3.

4.2.3 Filter-Generator Flux and Torque Calculation

The difference equations for the calculation of the filter-generator flux and torque are given in (4.5).

$$\hat{\vec{\Phi}}_f(k) = L_f \cdot \vec{i}_s(k) + \hat{\vec{\Phi}}_s(k), \tag{4.5a}$$

$$\hat{m}_f(k) = \frac{3N_p}{2} \cdot \left(\hat{\Phi}_{f,\alpha}(k) \cdot i_{f,\beta}(k) - \hat{\Phi}_{f,\beta}(k) \cdot i_{f,\alpha}(k)\right). \tag{4.5b}$$

The VHDL-AMS implementation for this can be found in Fig. D.3.

4.2.4 Speed Estimation

The speed estimator, as in the Simulink implementation in Fig. 4.9, is composed of a simple digital derivative with filter (refer to (2.19)). The implementation, however, is somewhat different than that of Simulink. The unwrapping of the estimated angle is done by an *if-then-else* statement, with the derivative and filter implemented as one difference equation. These equations are given in (4.6).

if $\hat{\theta}_r(k) > \hat{\theta}_r(k-1) + \frac{\pi}{2}$ **then**

$$\hat{\omega}_r(k) = 0.995 \cdot \hat{\omega}_r(k-1) + 0.005 \cdot \frac{1}{T_s} \cdot \left(\hat{\theta}_r(k) - \left(\hat{\theta}_r(k-1) + 2\pi\right)\right). \tag{4.6a}$$

else

$$\hat{\omega}_r(k) = 0.995 \cdot \hat{\omega}_r(k-1) + 0.005 \cdot \frac{1}{T_s} \cdot \left(\hat{\theta}_r(k) - \hat{\theta}_r(k-1)\right). \tag{4.6b}$$

end if

The estimation produced by this method still has a significant amount of ripple. To reduce this ripple without having a significant effect of the bandwidth of the digital

filter, another filter is placed in cascade with the first. The difference equation is given in (4.7).

$$\hat{\omega}_{r,1}(k) = 0.995 \cdot \hat{\omega}_{r,1}(k) + 0.005 \cdot \hat{\omega}_r^{(}k). \tag{4.7a}$$

The VHDL-AMS code for this can be found in Fig. D.4.

4.2.5 Direct Torque Control

The difference equations for the direct torque controller are derived from the diagram in Fig. 4.10. These are presented in (4.8).

$$\vec{i}_{s,xy}(k) = K_{xy} \cdot \vec{i}_s(k), \tag{4.8a}$$

$$\varepsilon_\Phi(k) = |\vec{\Phi}_f|^* - |\hat{\vec{\Phi}}_f|(k), \tag{4.8b}$$

$$\varepsilon_m(k) = m_f^*(k) - \hat{m}_f(k), \tag{4.8c}$$

$$x_\Phi(k) = x_\Phi(k-1) + \frac{T_s}{2} \cdot (\varepsilon_\Phi(k) + \varepsilon_\Phi(k-1)), \tag{4.8d}$$

$$x_m(k) = x_m(k-1) + \frac{T_s}{2} \cdot (\varepsilon_m(k) + \varepsilon_m(k-1)), \tag{4.8e}$$

$$u_x(k) = k_{p,\Phi} \cdot \varepsilon_\Phi(k) + k_{i,\Phi} \cdot x_\Phi(k) + r_f \cdot i_{f,x}(k), \tag{4.8f}$$

$$u_y(k) = k_{p,m} \cdot \varepsilon_m(k) + k_{i,m} \cdot x_m(k) + r_f \cdot i_{f,y}(k), \tag{4.8g}$$

$$\vec{u}_i(k) = K_{xy\to\alpha\beta} \cdot \vec{u}_{i,xy}(k). \tag{4.8h}$$

The actual VHDL-AMS implementation of the DTC can be found in Fig. D.6.

The inverter control voltage $\vec{u}_i$ is passed as an argument to a SVM function that generates the duty cycles for the next switching period. These duty cycles are then used to switch the power electronic switches of the converter according to double edge PWM (DEPWM). The VHDL-AMS code for the switch signal generation can be found in Fig. D.7.

This concludes the dynamic control for the system – any further control of the system will employ the DTC as a backbone to accomplish the desired operating conditions.

4.2.6 MPPT Controller

With reference to Section 3.7, the MPPT controller is only a torque reference generator, based on the estimated rotor speed. The difference equation for the MPPT controller, from (3.44), is given in (4.9).

$$m_f^*(k) = f\left(\hat{\omega}_r(k)\right) = 0.0185 \cdot \hat{\omega}_r^2(k) - 1.575 \cdot \hat{\omega}_r(k). \tag{4.9}$$

Its place in the VHDL-AMS simulation code can be seen in Fig. D.6.

4.2.7 Wind Turbine Simulator

To more accurately simulate the working of the system, wind turbine characteristics must be included. This is accomplished here by implementing a wind turbine simulator (WTS), derived from existing blade curve data. A simple diagram, illustrating the structure and functional significance of the WTS, is given in Fig. 4.12.

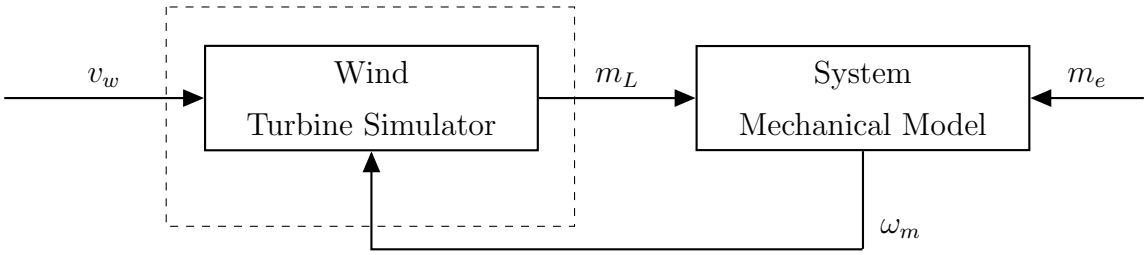


Figure 4.12: *Diagram representing the structure and functional significance of the proposed wind turbine simulator.*

The WTS consists of polynomial functions fitted to the blade curve data, taking as inputs of the wind speed and turbine rotational speed and outputting the associated turbine torque m_L as the load torque of the PMSG.

The polynomial coefficients are determined from the current wind speed by (4.10).

$$\begin{bmatrix} c_{\tau,0} \\ \vdots \\ c_{\tau,n} \end{bmatrix} = \begin{bmatrix} k_{v,00} & \cdots & k_{v,0n} \\ \vdots & \ddots & \vdots \\ k_{v,n0} & \cdots & k_{v,nn} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \vdots \\ v_w^n \end{bmatrix}. \tag{4.10}$$

The turbine torque, as a function of the turbine rotational speed, is then determined by (4.11).

$$m_L(\omega_m, v_w) = c_{\tau,n}(v_w) \cdot \omega_m^n + \cdots + c_{\tau,0}(v_w). \tag{4.11}$$

This model of the wind turbine is implemented in the VHDL-AMS simulation to demonstrate the variable speed MPPT operation of the system.

4.3 Simulation Results

In this section, simulation results are presented, demonstrating the dynamic performance of the system. The system is subjected to a torque step and run at initial speeds of 20 Hz and 40 Hz, which cover the boundaries of the operating range of the system. Only a positive torque step is simulated. The parameters of Table 3.2 are implemented in this simulation. The load torque is zero, except in the relevant tests where it is varied.

The Simulink simulation results will be presented and discussed first, after which the VHDL-AMS simulation results wil be presented. For the VHDL-AMS simulation, in addition to the torque step simulation, a wind turbine simulator (WTS) is implemented to demonstrate the variable speed MPPT working of the system.

4.3.1 Simulink Torque Step Simulation

The simulation results are presented in order of calculation precedence, starting with the estimated generator EMF in Fig. 4.13(a). The estimated EMF shows a short transient as the simulation starts, but stabilises after about 1ms with damped behaviour.

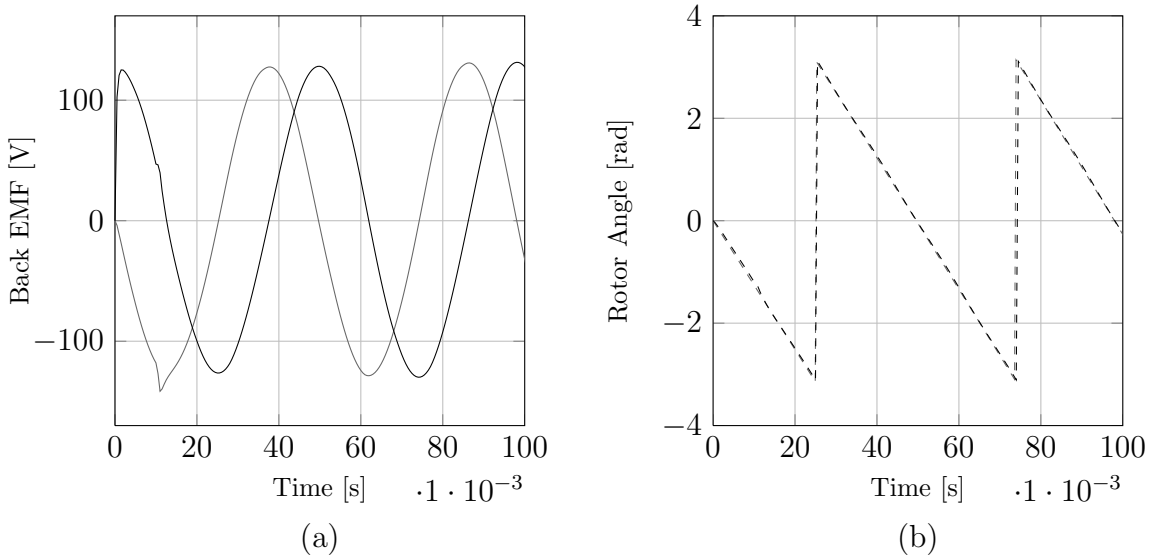


Figure 4.13: *Simulation results of the torque step simulation in MATLAB Simulink, showing generator back EMF (a) and rotor angle (b), for an initial frequency of 20 Hz.*

The estimated rotor position, calculated from the EMF voltage, is shown in Fig. 4.13(b). Note that the actual position is superimposed as a dashed, gray line. The estimated position follows the actual position accurately, as is clear from the plot. When the torque step is initiated at 10 ms, a quite severe transient appears on the estimated

EMF. The estimated rotor position, however, seems unaffected. This demonstrates the robustness of this simple technique.

The estimated flux linkage and the estimated stator flux linkage angle, along with the actual values, are shown in Fig. 4.14. The estimated values follow the actual values very accurately. A slight transient is observed in the estimated angle just after the torque step. This can be attributed to the flux observer's reliance on the stator current.

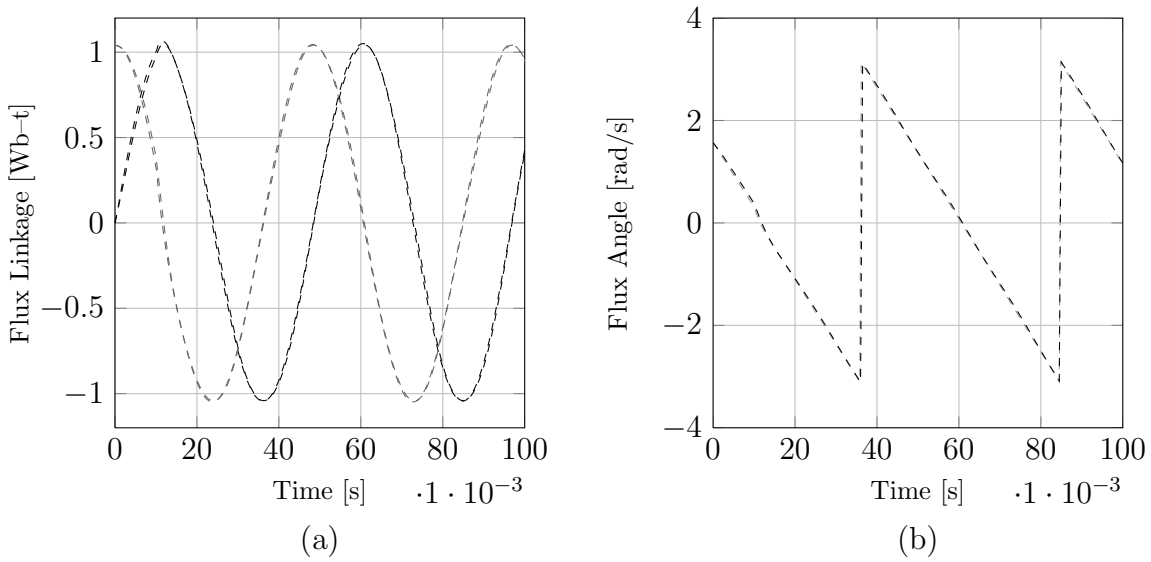


Figure 4.14: *Simulation results of the torque step simulation in MATLAB Simulink, showing estimated stator flux linkage (a) and angle (b), for an initial frequency of 20 Hz.*

The estimated filter-generator torque and estimated filter-generator flux linkage magnitude are shown in Fig. 4.15(a) and (b), respectively. The actual values of just the generator torque and flux linkage are superimposed as dashed gray lines for some comparison; the filter-generator variables are virtual, and have no actual physical counterparts. The difference in the filter-generator torque and the electromagnetic torque, at this frequency, is minuscule. A slight difference can be seen in the flux linkages.

The torque response is fast, with a rise time and settling time of about 3 ms and 20 ms, respectively. The flux linkage shows some transient behaviour on the torque step, but stabilises quickly at the reference value.

To demonstrate the dynamic range of the system in simulation, the same results are peresented at 40 Hz. The estimated generator EMF voltage and subsequent estimated rotor position are given in Fig. 4.16(a) and (b), respectively. The response of the EMF estimation at 40 Hz also stabilises quickly, with a damped response.

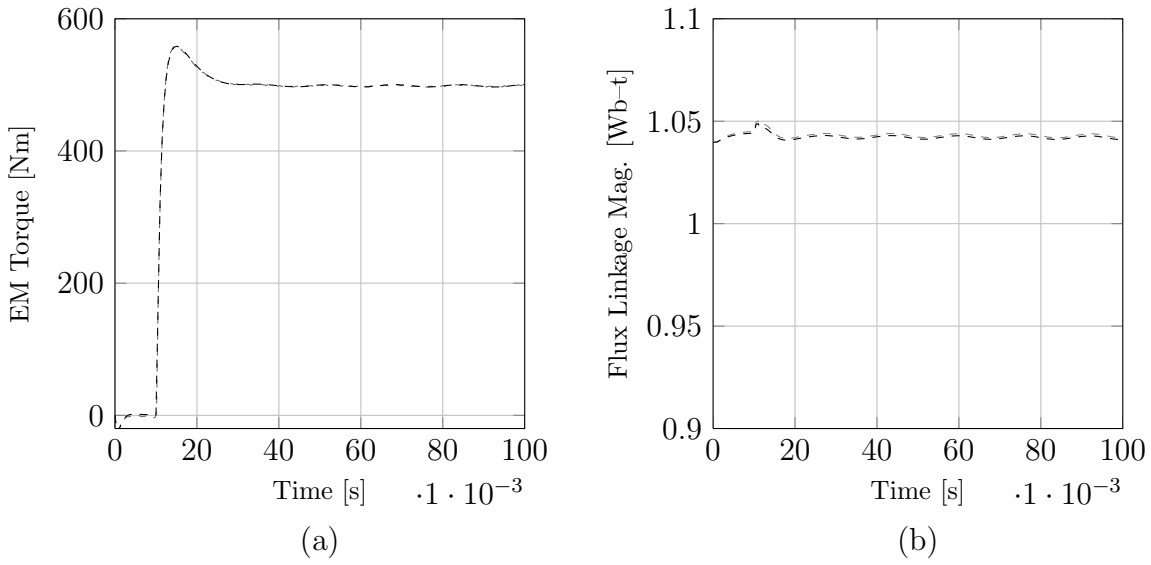


Figure 4.15: *Simulation results of the torque step simulation in MATLAB Simulink, showing estimated filter-generator torque (a) and filter-generator flux linkage magnitude (b), for an initial frequency of 20 Hz.*

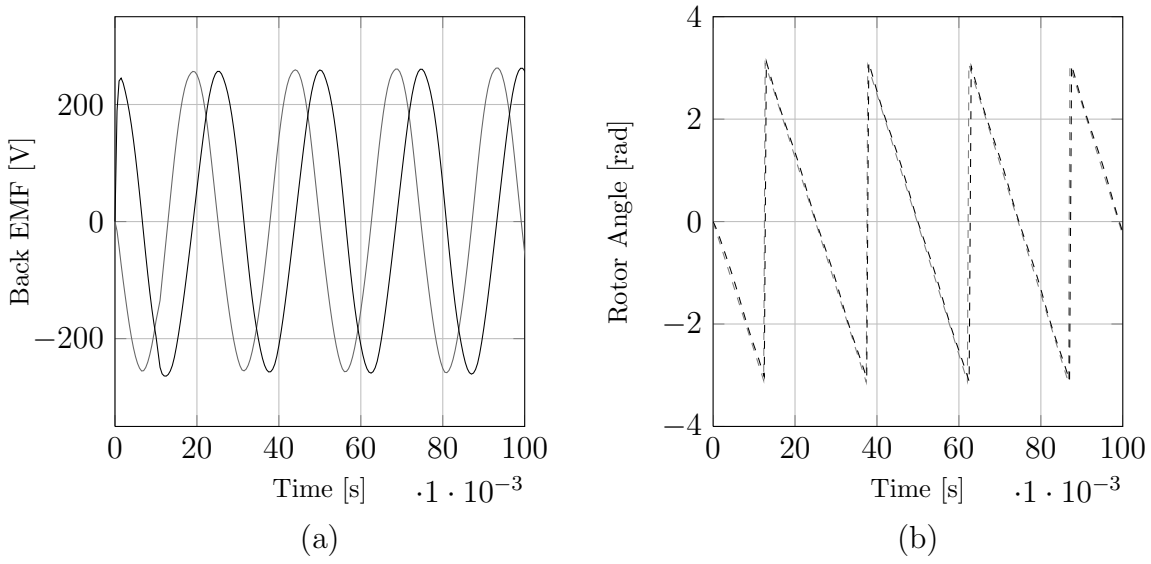


Figure 4.16: *Simulation results of the torque step simulation in MATLAB Simulink, showing generator back EMF (a) and rotor angle (b), for an initial frequency of 40 Hz.*

The calculated rotor position corresponds well to the actual rotor position, superimposed as a gray, dashed line. There are no major changes in performance; however if one looks closely, a slight oscillation can be observed on the estimated rotor position.

The estimated stator flux linkage and calculated stator flux angle are shown in Fig. 4.17(a) and (b), respectively, with the actual values superimposed as gray dashed lines. A slight steady state error can be seen in the amplitude of the estimated stator flux

linkage, although this does not seem to have any adverse effect on the calculated angle. Like the estimated rotor position, there is a slight oscillation present on the estimated flux linkage angle.

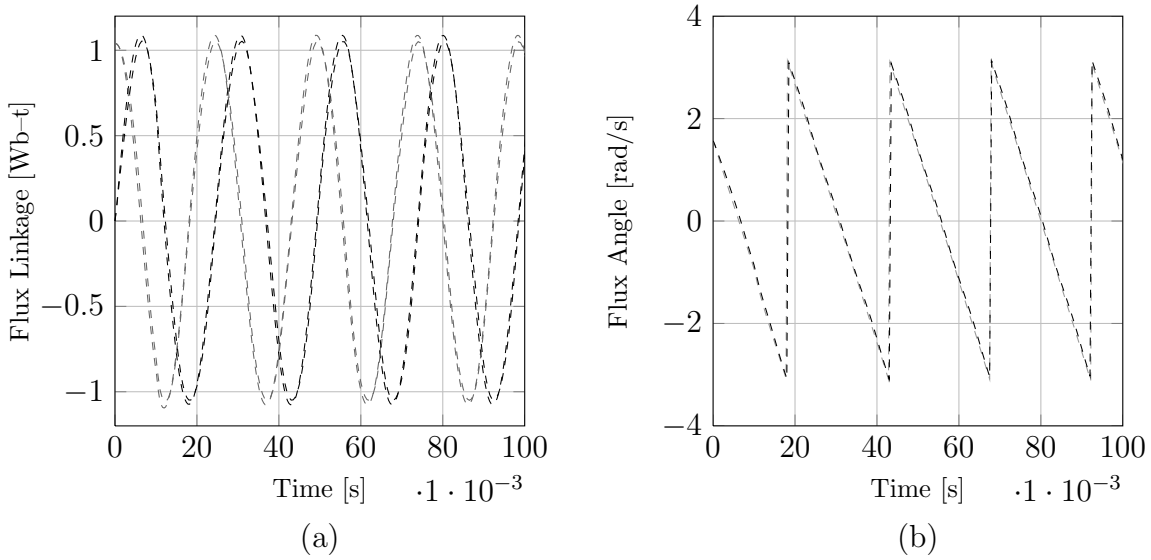


Figure 4.17: *Simulation results of the torque step simulation in MATLAB Simulink, showing estimated stator flux linkage (a) and flux angle (a), for an initial frequency of 40 Hz.*

The estimated filter-generator torque and flux magnitude, for an initial frequency of 40 Hz, are given in Fig. 4.18(a) and (b), respectively. The actual electromagnetic torque and stator flux linkage magnitude are superimposed as dashed gray lines, for some comparison.

The dynamic response of the torque seems unchanged from the 20 Hz initial frequency simulation result. The only difference is the increased torque ripple. There is now, however, a noticeable difference between the estimated filter-generator torque and the actual electromagnetic torque. The estimated filter-generator flux linkage magnitude also has a ripple.

The effects of the steady state error in the flux linkage estimation can be seen in the filter-generator flux, as the final value is slightly higher than the control reference. The transients in the torque step vanish quickly, although they are slightly more vivid than the 20 Hz initial frequency simulation.

4.3.2 Speed Estimation Comparison

Simulation results for the speed estimation techniques are shown here only for comparison, as they have no real function in the Simulink simulation.

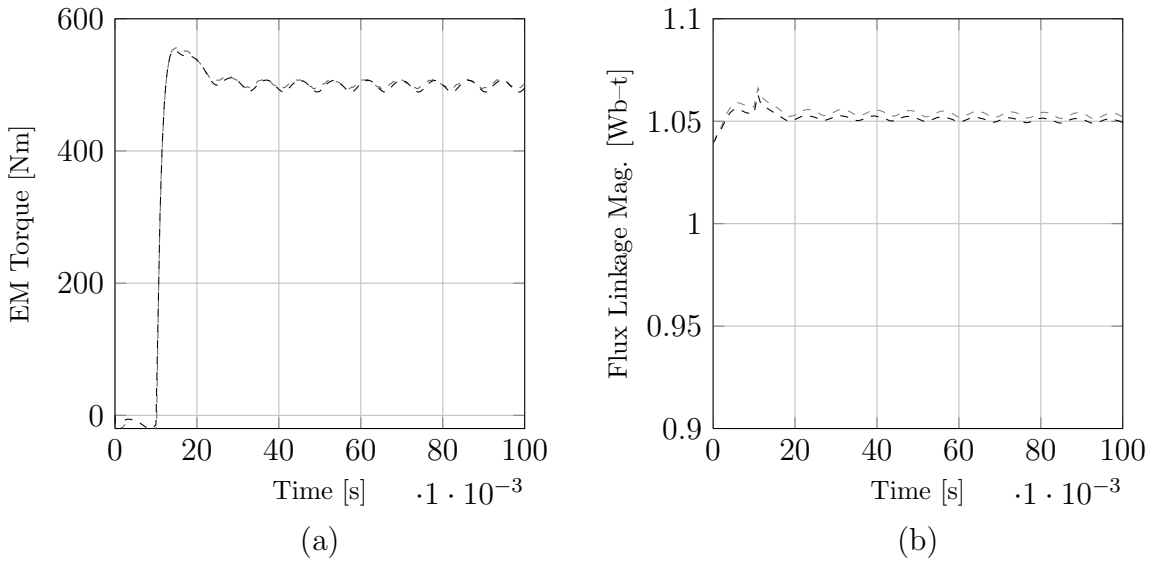


Figure 4.18: *Simulation results of the torque step simulation in MATLAB Simulink, showing filter-generator torque (a) and filter-generator flux linkage magnitude (b), for an initial frequency of 40 Hz.*

The estimated rotor speed for an initial frequency of 20 Hz, for the different rotor speed estimation techniques, is shown in Fig. 4.19. The actual rotor speed is also superimposed on the plot for comparison.

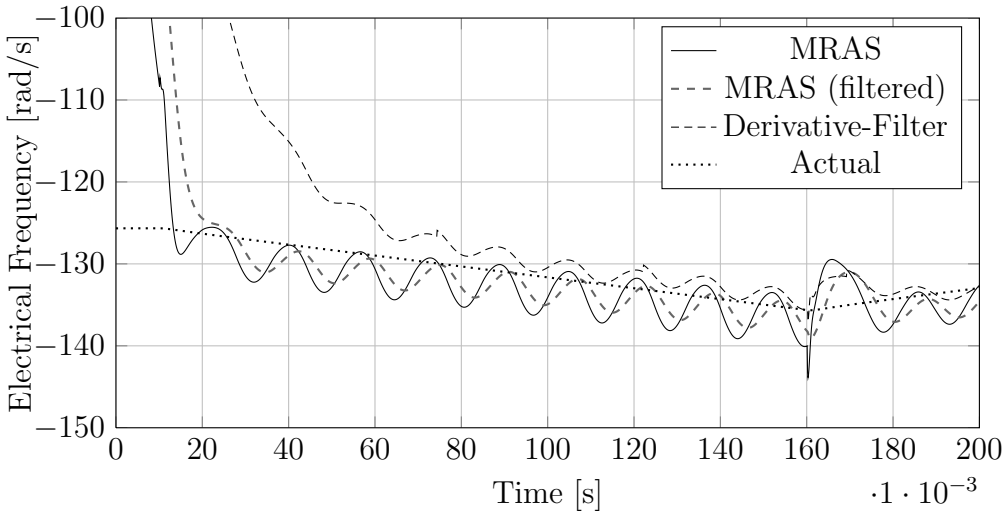


Figure 4.19: *Comparison of the rotor speed estimation of the simple filter method and the MRAS speed observer, with and without filtering, for an initial frequency of 20 Hz.*

The MRAS speed observer does have the fastest response, but it has the most severe ripple. Even when filtered, the estimated rotor speed produced by the MRAS still has more ripple than the derivative-filter method. A steady state error can also be observed for the result produced by the MRAS speed observer; the derivative-filter

method has better steady state accuracy than the MRAS.

The transient produced by the MRAS with the introduction of a second torque step at 160 ms is also worse than that of the derivative-filter method. The filtering stage of the MRAS observer does mitigate the transient to a large degree.

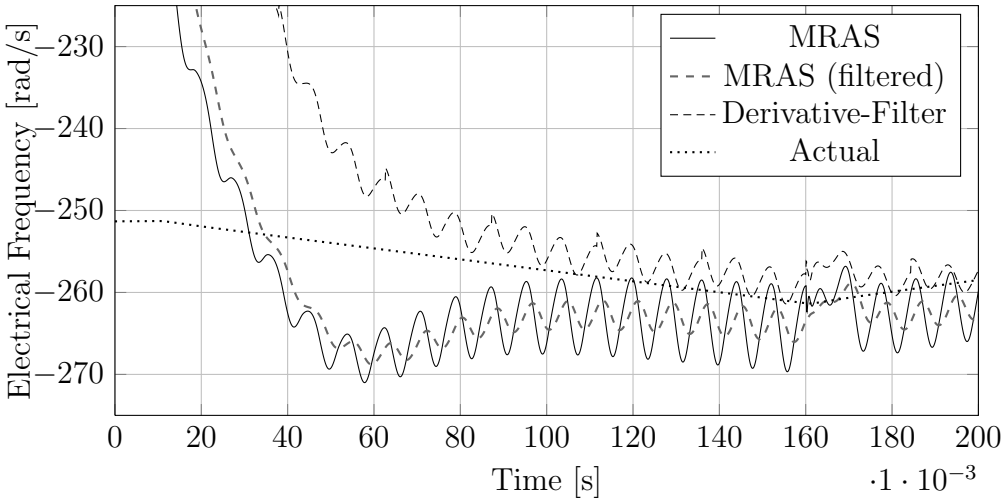


Figure 4.20: *Comparison of the rotor speed estimation of the simple filter method and the MRAS speed observer, for an initial frequency of 40 Hz.*

The speed estimator results for an initial frequency of 40 Hz is shown in Fig. 4.20. The speed estimation produced by the MRAS observer is practically unusable, due to extreme ripple. The filtering stage does, however, reduce the ripple to a level that is about the same as the derivative-filter method.

The steady state error of the MRAS speed observer is worse in this case, with the steady state accuracy of the derivative-filter method staying much the same.

The overall performance of the derivative-filter method is surprisingly superior to that of the MRAS observer. It is also a much simpler method in terms of implementation, and is much less computationally demanding.

4.3.3 VHDL-AMS Torque Step Simulation

The VHDL-AMS simulation results are presented in order of calculation, according to the procedural precedence of the DSP process.

The estimated EMF voltages in the stationary reference frame and the estimated rotor position angle, along with the actual EMF voltages and position angle, are given in Fig. 4.21. The superimposed gray, dashed lines represent the actual values.

The estimated EMF voltage converges towards the actual EMF voltages quickly and remains stable. Significant transients can be observed as the torque step is initiated

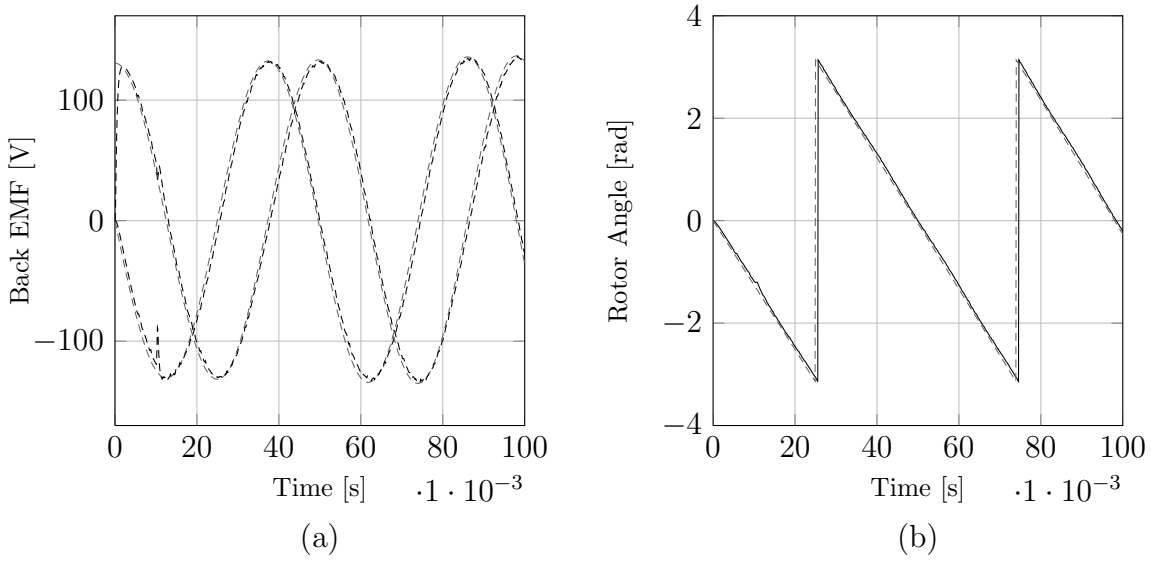


Figure 4.21: *Results of the VHDL-AMS torque step simulation, showing generator back EMF (a) and rotor position angle (b), for an initial frequency of 20 Hz.*

at 10 ms, but stability is regained quickly. The estimated rotor position angle also converges to the actual rotor position quickly, and is not noticeably affected by the transient in the estimated EMF voltage.

The estimated filter-generator torque and estimated filter-generator flux linkage magnitude are shown in Fig. 4.22(a) and (b), respectively. The actual values of the generator electromagnetic torque and flux linkage are superimposed as dashed, gray lines, for some comparison.

The response is somewhat different to that of the Simulink simulation – most noticeably the overshoot, which is less than 5 % in this case. There is also a high frequency ripple present on the filter-generator torque. The rise time and settling time, however, still correlate well with the Simulink results. A more noticeable difference can be seen between the filter-generator torque and the generator electromagnetic torque.

The filter-generator flux linkage converges to the reference value in about 5 ms. The initial value in this simulation for the filter-generator flux linkage differs from that in the Simulink simulation, hence the difference in initial response. The transient due to the torque step quickly dies away, and the controller stabilises at the constant reference value.

As in the Simulink simulation, the VHDL-AMS torque step simulation is also repeated for an initial frequency of 40 Hz. The results of the estimated EMF voltages and subsequent rotor position angle is shown in Fig. 4.23.

The performance of the EMF and rotor position angle estimation is still very good

at 40 Hz. The values converge quickly, and stabilise. A slight oscillation on the rotor position angle estimation can also be observed here, albeit somewhat less than in the Simulink simulation.

The estimated filter-generator torque and estimated filter-generator flux linkage magnitude, for an initial frequency of 40 Hz, are shown in Fig. 4.24(a) and (b), respectively, along with the actual values of the generator electromagnetic torque and stator flux linkage, superimposed as gray, dashed lines.

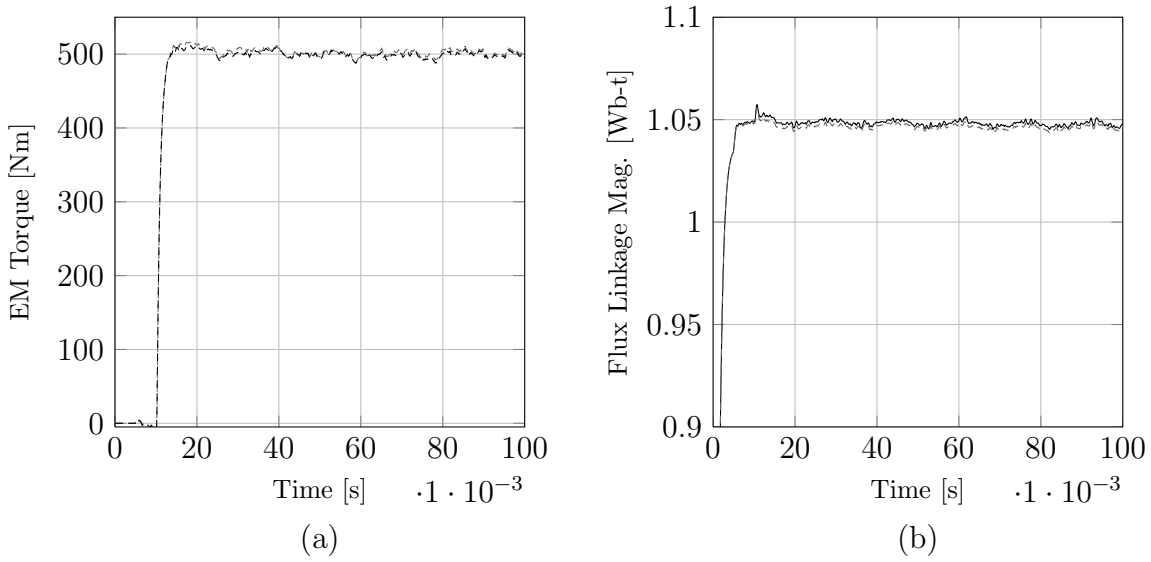


Figure 4.22: *Results of the VHDL-AMS torque step simulation, showing generator EM torque (a) and stator flux linkage (b), for an initial frequency of 20 Hz.*

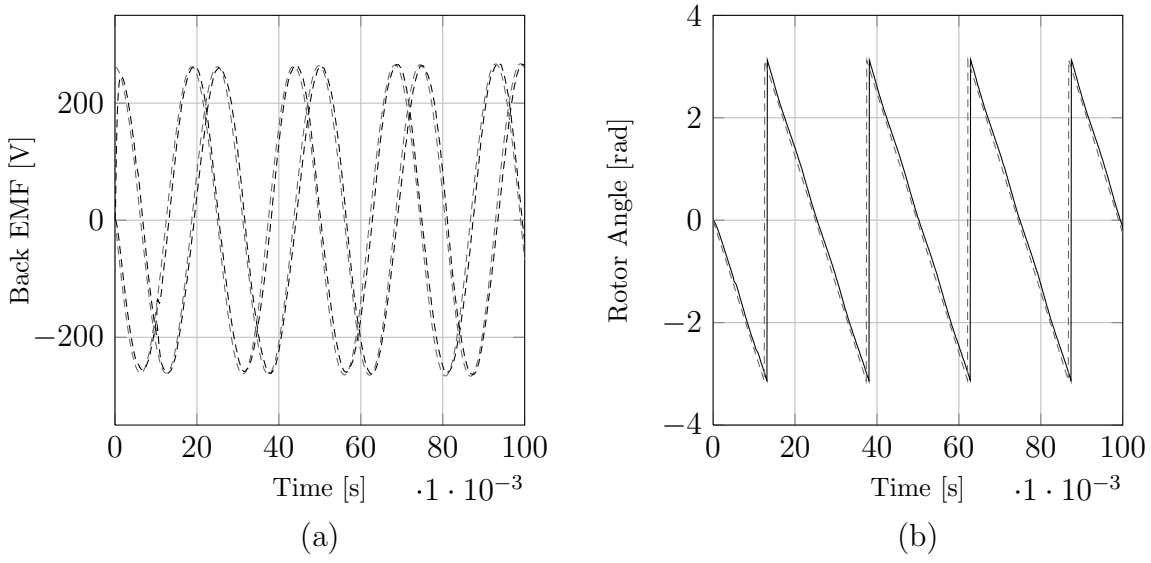


Figure 4.23: *Results of the VHDL-AMS torque step simulation, showing generator back EMF (a) and rotor angle (b), for an initial frequency of 40 Hz.*

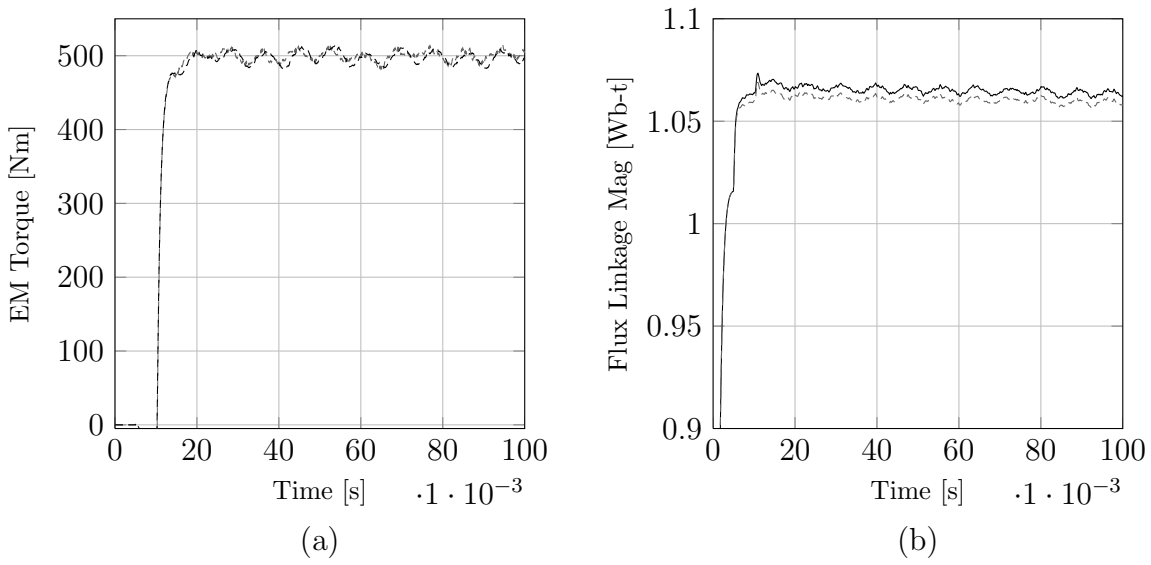


Figure 4.24: *Results of the VHDL-AMS torque step simulation, showing generator EM torque (a) and stator flux linkage (b), for an initial frequency of 40 Hz.*

The filter-generator torque response correlates well with the Simulink simulation, with the exception of the overshoot, which is even less than the 20 Hz initial frequency case, and the high frequency ripple. The flux linkage also converges quickly and stabilises after the transient due to the torque step. The effects of the steady state error of the flux observer can also be seen here, with the flux linkage somewhat higher than the reference value.

It can be seen that the VHDL-AMS results correlate very well with that of the Simulink simulation. The minor differences can be attributed to the differences in inverter models. The VHDL-AMS simulation uses mathematical models of the IGBT, see Fig. D.10, and the reverse diodes, see Fig. D.9, and a circuit model, while an ideal switch model is used in the Simulink simulation.

4.3.4 VHDL-AMS Variable Speed MPPT Simulation

The system under variable speed operation is implemented in the VHDL-AMS simulation. A simple wind model is developed in [2] that models wind speed as the combination of base, ramp, gust and random noise components. Only the ramp component of wind is utilised in this simulation. The WTS developed earlier in the chapter will be used to generate the load torque imposed on the generator rotor, based on the input wind speed.

The filter-generator torque and load torque over a 10 second interval is shown in Fig. 4.25. The system immediately changes as the variable speed algorithm is engaged

at about 2 s, after which convergence to the optimal working point commences.

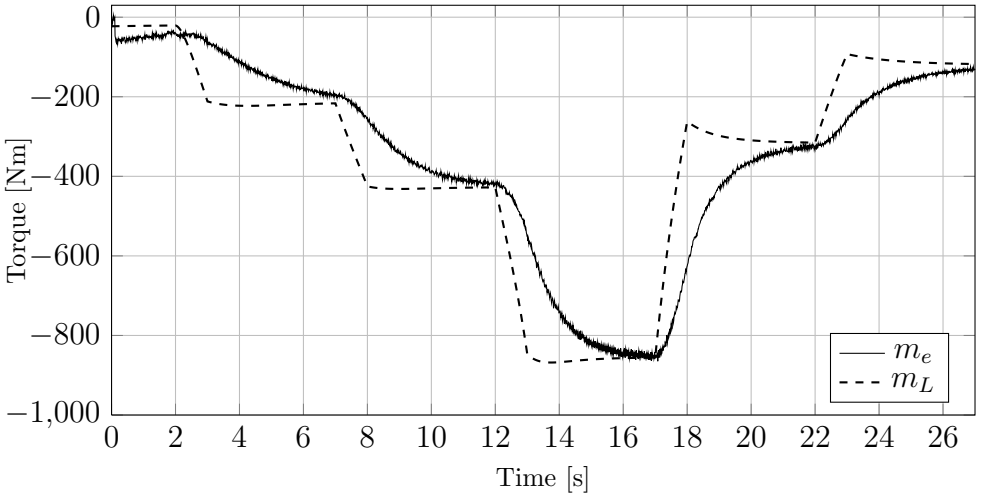


Figure 4.25: *Results of the MPPT VHDL-AMS simulation, showing generator electromagnetic torque and load (tubine) torque.*

The turbine rotational speed is shown in Fig. 4.26, along with the input wind speed to the WTS; the speed converges to the optimal working point after each wind speed ramp change.

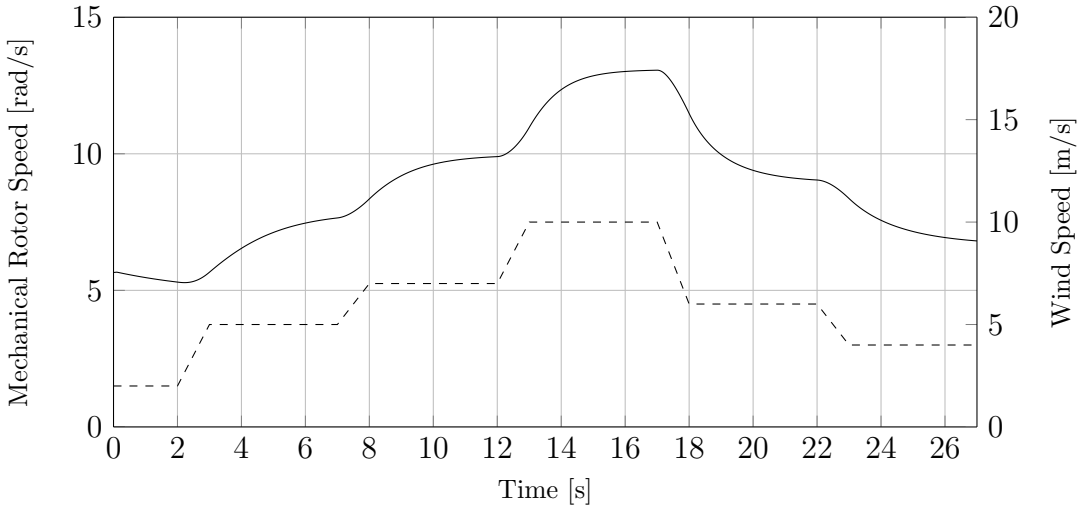


Figure 4.26: *Results of the MPPT VHDL-AMS simulation, showing mechanical rotor speed and the input wind speed, represented by the dashed line.*

The electrical output power is plotted against the maximum available power from the wind turbine, shown in Fig. 4.27. It can be seen that the system converges towards the maximum output power point, with the exception of the highest wind speed ramp change. This step is very large, and the system would require more time to converge completely.

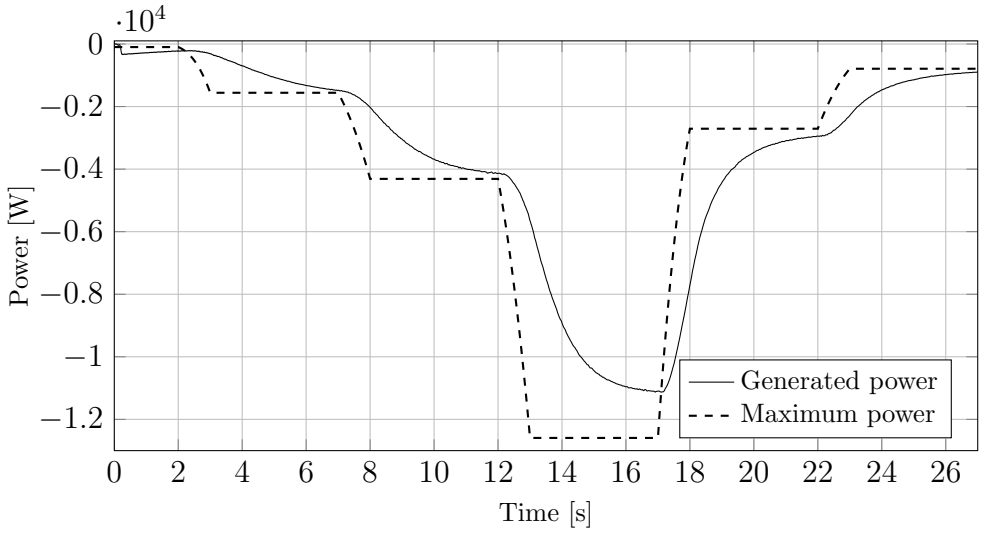


Figure 4.27: *Results of the MPPT VHDL-AMS simulation, showing average generated electrical power and maximum available power that can be extracted from the wind (with the current turbine).*

The relatively short duration of this simulation does not completely convey reality, as the wind speed conditions in practice change much slower. It does, however, demonstrate that the system performs very well, even when subjected to bad wind conditions. At the instances where the system seems to be generating more power than what is theoretically possible (represented by the dashed line), it is only inertially stored energy in the turbine that is being released.

4.4 Summary and Conclusion

Simulation models and methodologies for both the Simulink and VHDL-AMS simulations are presented and discussed. The results of a torque step simulation is presented for both simulation strategies. The speed estimation results of derivative-filter method and the MRAS speed observer are then compared. The chapter concludes with the results of a variable speed simulation that demonstrate the MPPT control of the system.

The estimation techniques used all give very good performance, with the exception of the speed estimator. The latter is the biggest problem in terms of estimation that this system faces. It is shown that the derivative-filter method is the preferred choice for speed estimation.

The torque and flux responses both convey very good performance. The torque response has minimal overshoot, settles fast and has very little steady state error. The flux linkage controller is only slightly affected by disturbances, such as the torque step.

It is seen that the overall results of the VHDL-AMS simulation correlate well with the Simulink simulation results.

The VHDL-AMS variable speed MPPT simulation demonstrates the most important function of the system, and very good results are obtained.

Chapter 5

Practical Implementation & Results

The previous chapters deal with the theoretical modelling, design and simulation of the system – all contributing towards the practical implementation. Even though theoretical simulations provide a good insight into the performance of a system, true design validation will only be obtained by means of physical implementation.

This chapter presents the physical implementation methodology that consists of the hardware configuration and design, as well as the software. Implementation issues that were encountered during implementation and testing are discussed. Tests results of the individual components of the system, and finally the full system, are then presented.

The chapter ends with conclusions and a discussion of the work presented in this chapter.

5.1 Testing Methodology

The practical test are performed on a laboratory test setup, of which a diagrammatic representation is given in Fig. 5.1. The hardware setup consists of a permanent magnet synchronous generator (PMSG) coupled to an induction motor via a prop shaft and gearbox.

The induction motor and gearbox emulate the wind and the wind turbine, which is driven by a Danfoss variable speed drive (VSD) using torque control. The torque reference is controlled by means of a portable analogue control knob, although it was initially intended to feed data to the VSD to more realistically simulate wind conditions. Time constraints did not allow the latter to be implemented.

The generator is connected to the power electronic converter by a three-phase cable through an RLC filter. The control algorithms are implemented on a Texas Instruments DSP, which controls the converter circuitry.

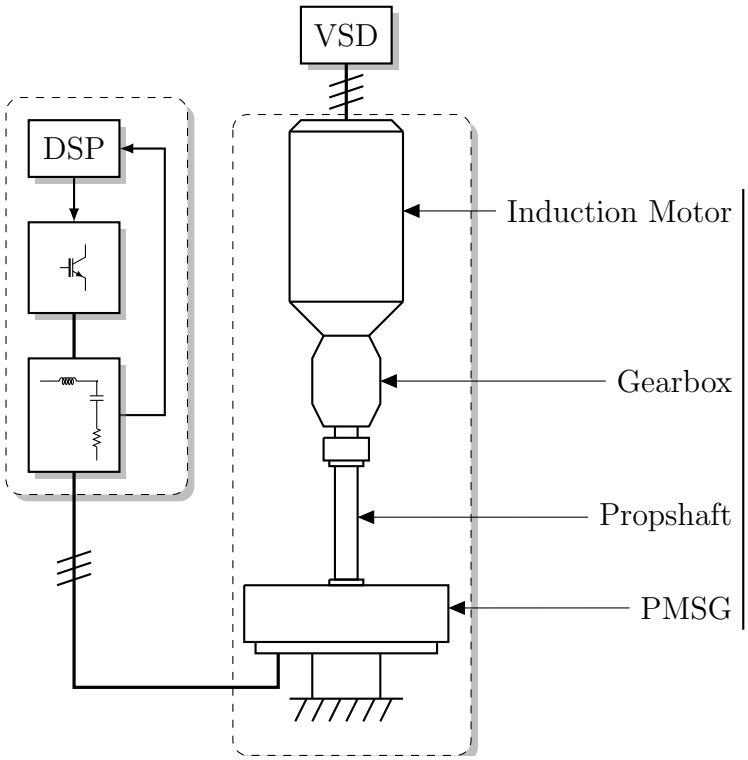


Figure 5.1: *A representation of the laboratory test setup.*

Voltage and current probes are used to measure the filter capacitor line to neutral voltages and generator phase currents, respectively. The rotor speed is measured by means of a shaft encoder.. Torque measurements are obtained by a rotational torque meter fixed to the prop shaft.

The converter used in this system had to be slightly modified to comply with the requirements of the design. The first modification includes the moving of the current sensors to the generator-side of the sinus filter, so that the clean stator current could be measured rather than the noisy inductor current. The filter capacitor damping resistances were also included.

Existing software from a previous implementation on the same hardware is re-used and modified for this implementation. This will be futher discussed in the relevant section.

The estimated variables in the DSP software is obtained by logging data in the DSP memory, and then transmitting the data through RS232 to a computer. Only very short time windows could be captured, due to the very limited flash memory available on the DSP.

A digital to analogue converter (DAC) was tested to be implemented for real-time readings from the estimators, but due the difficulties in implementation and quality of readings, it was decided not to incorporate it into the tests.

5.2 Implementation Issues

This section describes some issues that were encountered during the implementation and testing of the system.

Electromagnetic interference (EMI) caused major problems in this implementation. The first encounter with this phenomena was on the implementation of the designed filter capacitances. Although the capacitors were theoretically sufficient for application in this system, the excess EMI caused random behaviour on the electronics of the system. The previous, larger filter capacitors were then used instead.

The converter housing did not have sufficient space to mount the damping resistances, as they are quite large, so they had to be kept outside the housing, and connected to the filter capacitors by cables. These cables broadcast the EMI through the entire converter, as it was only much later discovered. The effects weren't immediately visible, but every now and then the converter would de-synchronise completely and go offline. It was later discovered that the DSP received a signal that triggered the stop sequence, that was caused by the EMI from the cables connecting the filter capacitors and damping resistances.

After some initial tests, the converter software started reporting voltage unbalance errors on the DC bus capacitors. After numerous tests, dis-assemblies and assemblies of the converter, and re-forming of the bus capacitors, the culprits were discovered. Two of the six DC bus capacitors were faulty, and did not hold charge either sufficiently or at all. They were replaced, and then re-formed.

In one of the tests, a bug in the newly developed generator control software caused the converter to synchronise incorrectly. This caused large inrush currents from the grid through the grid-side IGBT modules, which caused them to fail silently. When the grid-synchronisation sequence was initiated again, the IGBTs did not respond to the control signals from the DSP, and failed to start switching. This caused prolonged large currents to flow through the soft-start resistors, and they failed very loudly. They were replaced, along with the grid-side IGBT module, and testing continued.

The control circuitry of the generator-side IGBT module incurred damage during the course of testing, presumably during the same incident when the grid-side IGBT module had failed[1]. During a test, after synchronisation with the generator, the generator-side IGBT module failed very loudly; It completely destroyed itself and its control circuitry. The IGBT module was replaced with the last available spare, and

[1] A project engineer that assisted with the implementation also later discovered some errors in the software that could have contributed to the failure of the IGBT modules.

the tests were completed.

5.3 Program Flow

The most important and challenging work is done in software development. Presented in this section are details on the DSP software and specific algorithms relevant to this study.

The existing software takes care of the basic functions for operation, such as measurement & ADC, protection and the grid-side control of the back-to-back converter. The latter consists of the following:

- Initial charging of the DC-link bus capacitors from the utility network through soft-start resistors,
- boost charging though PWM switching,
- synchronising with the electricity network,
- and then finally controlling the DC bus voltage.

The grid-side control keeps the DC bus voltage constant by transferring power across the DC-link. The details of the grid-side will not be discussed, as it is outside the scope of this work. For more detail on this see [30].

The generator interrupt service routine (ISR), which is called at each sampling instant, is represented by the flow diagram in Fig. 5.2. The actual C-code for the generator ISR can be found in Fig. E.1.

The ISR commences with obtaining measurements from the ADC and executing the protection algorithms of the system. The next step is the *Estimation* sequence, which represents the EEMF observer, flux observer and speed estimation. The actual code for these can be found in Fig. E.2, Fig. E.3 and Fig. E.4, respectively.

The next step is a mechanism that forms part of the gain scheduling (GS) algorithm. Depending on the state that the system is in, the algorithm either continues or bypasses the control algorithms. If the system finds itself in a controlled state, the *Speed Controller* step, which generates the torque command for the *Torque Controller*, is either bypassed or selected. The actual code for the DTC and speed controllers can be found in Fig. E.5.

The *Torque Controller* detail is expanded to the left of the main ISR flow. Another one of the GS mechanisms inside this method controls how the torque reference is

selected. Depending on the state of the system, the torque reference is either zero (*Pre-states* and *Pre-cut-in*), generated by the speed controller or by the MPPT controller.

Finally, the control voltages produced by the flux and torque controllers are passed to the SVM function, which in turn calculates the duty cycles for the converter switches.

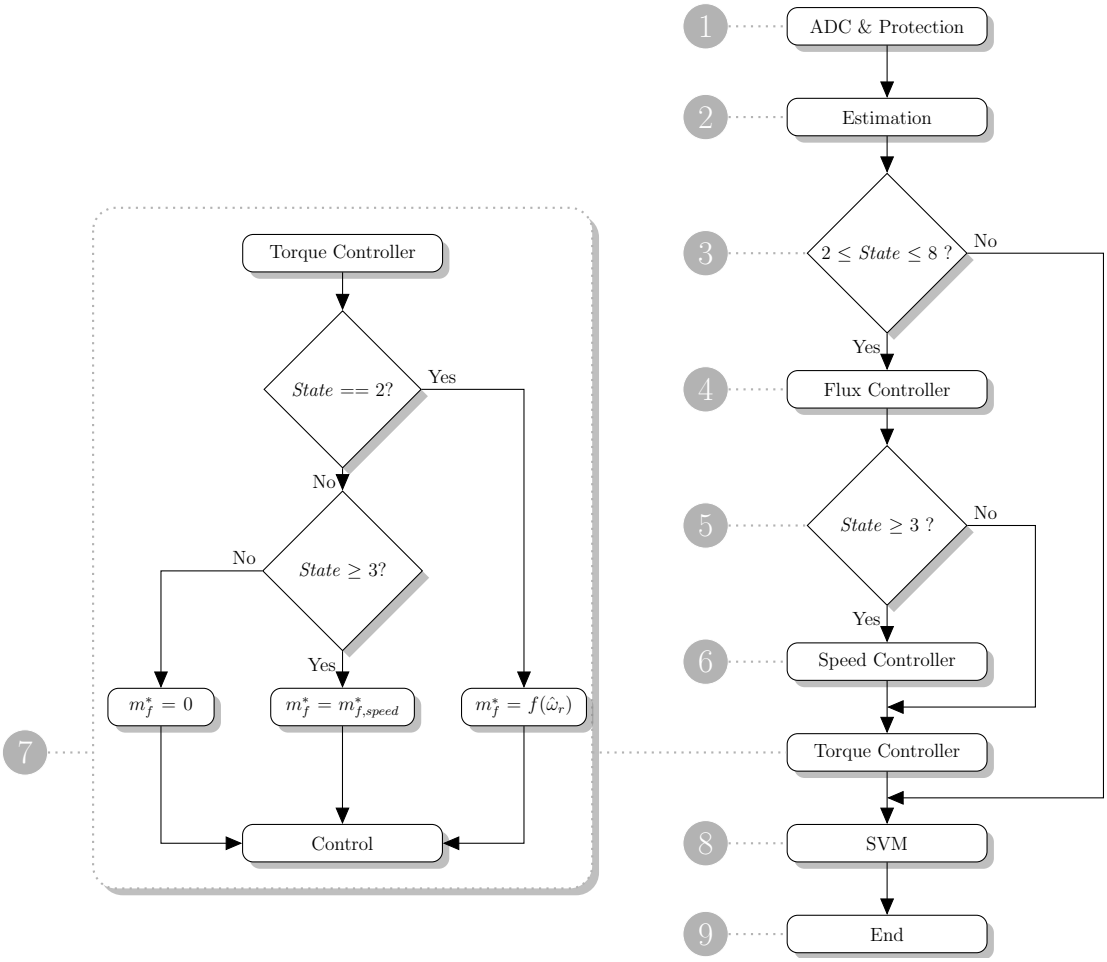


Figure 5.2: *Flow diagram describing the generator interrupt service routine program flow. Here "State" bears reference to the state of the gain scheduling system, and not the physical state of the system (as in "State Estimation").*

Wind is inherently a stochastic energy source, and must therefore be regarded as that. An innovative GS algorithm is required for the stable and effective high-level control of a WECS, with sufficient contingencies in place.

The crux of the system, which is the GS algorithm, is represented by the flow diagram in Fig. 5.3. This flow diagram can be seen as the detailed version of the conceptual state diagram first presented in Fig. 3.7. The conceptual states' relationship to the actual DSP states are presented in the table in Fig. 5.3. The flow diagram is also categorised according to these states for easy comparison to the conceptual design.

Once the system enters a state, it loops perpetually until the specific system variable

being monitored breaches specified threshold values. The blocks marked S_1 through S_7 are *transition sequences* that consist of the setting of initial values and assigning of controller references, as required for the next state.

Unlike the generator ISR, which is executed in discreet intervals, the GS state machine is a continuous process that monitors system variables and changes states, starting values and control references accordingly.

The DSP software has a higher level of system operation, that functions in terms of *modes*. When the grid-side converter is de-synchronised from the electricity network, the system is in the *Offline* mode. Once it has successfully synchronised with the network, and the DC bus voltage is controlled, the mode changes to the *Ready-Online* mode.

The *Online* mode is initiated when the generator-side converter is ready to synchronise with the generator. The system now monitors the rotational speed of the generator to determine the appropriate control action.

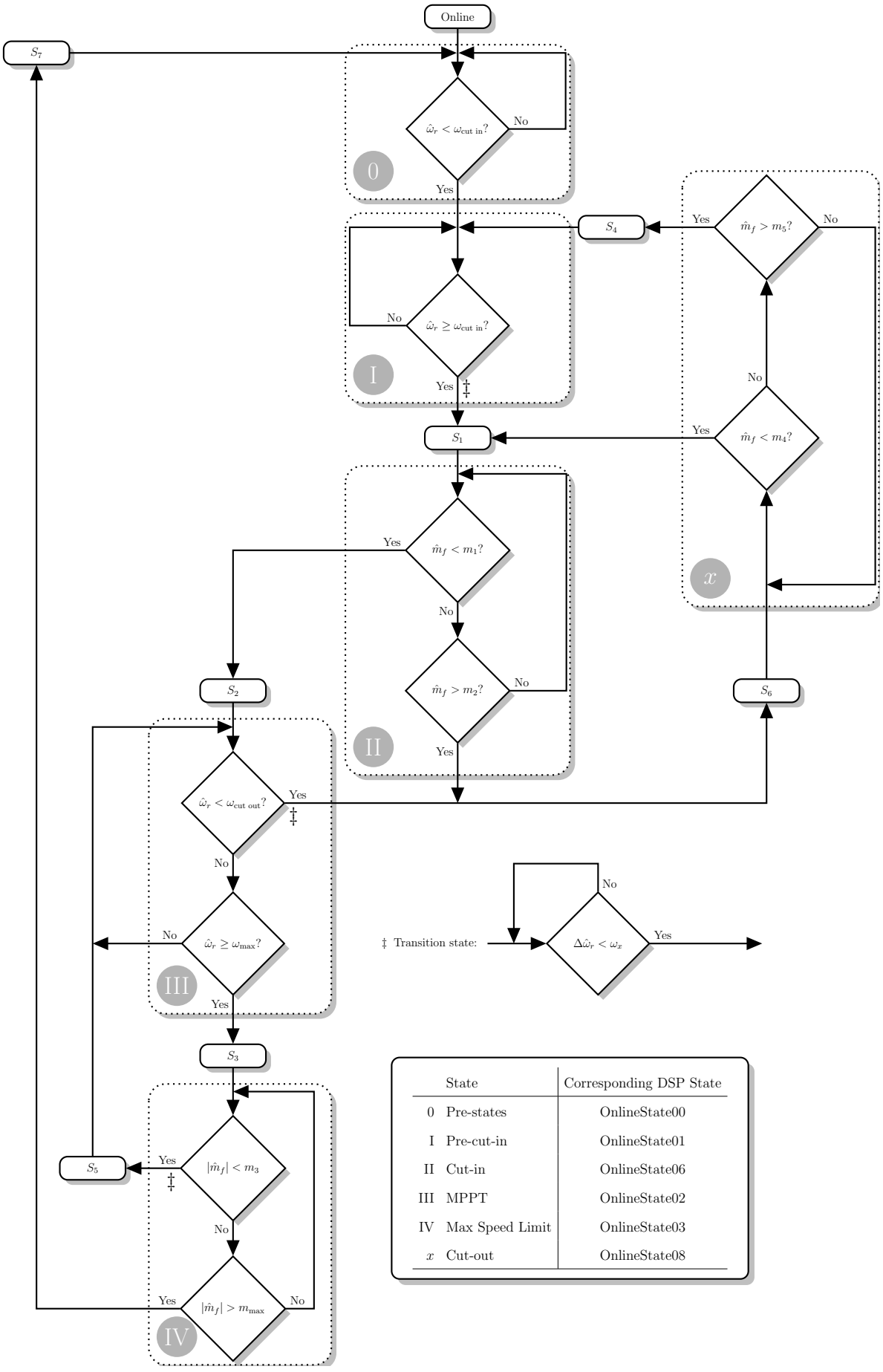
The system is initially in the *Pre-states* state, waiting for the rotational speed to be in a safe region, that is, below $\omega_{\text{cut in}}$, for synchronising. Once this constraint is satisfied, the GS algorithm transitions the system to the *Pre-cut-in* state. The system is now ready to synchronise with the generator, and will transition to the *Cut-in* state once the $\omega_{\text{cut in}}$ threshold is breached.

The ‡-symbol represents a *transition state*, as indicated in the flow diagram. In this state, the system waits for the rotational speed to stabilise, in order to prevent limit cycles. This additional state is required in some transitions, and it was decided upon implementation where the need arose.

The *Cut-in* state is also a type of transitional state, put in place to mitigate the occurrence of limit cycle behaviour. Once the system adopts this state, the generator rotational speed is controlled at $\omega_{\text{cut in}}$. The system filter-generator torque is now monitored to see if and when the wind speed increases sufficiently to transition to variable speed operation ($MPPT$), or conversely, decreases enough to transition to the *Cut-out* state.

When the filter-generator torque breaches the m_2 threshold, the system transitions to the variable speed $MPPT$ state, which is the desired operating region of the WECS. The generator rotational speed is monitored in this state, and is governed by the threshold parameters $\omega_{\text{cut out}}$ and ω_{max}. If the former is breached, the system transitions to the the *Cut out* state. If the latter is breached, the system transitions to the *Max. Speed Limit* state.

The *Max. Speed Limit* state is implemented as a protective measure to cater for



	State	Corresponding DSP State
0	Pre-states	OnlineState00
I	Pre-cut-in	OnlineState01
II	Cut-in	OnlineState06
III	MPPT	OnlineState02
IV	Max Speed Limit	OnlineState03
x	Cut-out	OnlineState08

Figure 5.3: *Flow diagram describing the gain scheduling system flow of this system.*

the limitations of the WECS. The generator is controlled at $\omega_{\max}$, and the transition decision is once again based on the filter-generator torque. If the wind speed decreases enough, the system re-enters the *MPPT* state. Conversely, if the wind speed becomes too high, the upper threshold of filter-generator torque is breached, and the system transitions back to the *Pre-states* state. Here the system waits for the wind speed to enter a safe region for synchronisation.

The *Cut-out* state can be seen as the intermediate state before generator de-synchronisation, and is also implemented to mitigate possible instabilities in state transitions. The generator is controlled at $\omega_{\text{cut out}}$, and the filter-generator torque is monitored in order to infer wind behviour. If the wind speed decreases sufficiently, the system de-synchronises, and transitions to the *Pre-cut-in* state. If the wind increases sufficiently, the system transitions to the *Cut-in* state.

There is no analytical method to determine the threshold values, and they are system specific. They have been determined by trial and error in this system.

5.4 Sub-system Test Results

In this section, the results obtained from the laboratory tests of the sub-systems comprising the WECS, are presented. These sub-systems are the estimators, DTC and speed controller. The gain scheduling (GS) algorithm is demonstrated in the tests of the full WECS, in the next section.

In terms of the estimation, results of the estimated EMF and subsequent rotor position are presented for completeness, even though there is no effective way to gauge their performance, other than estimation quality. The DTC is then tested by means of torque steps, and finally the speed controller.

The parameters of the physical system are presented in Table 5.1. It should be noted that the value of filter capacitance is not the same the designed value. This is discussed earlier in Section 5.2.

The larger filter capacitance has, as a consequence, much larger currents flowing through it. The losses in the damping resistances are therefore much more significant. When one performs the calculations in (3.32) and (3.33) with the larger capacitance, the losses in the damping resistance jump to approximately 113 W_{rms}.

This is still insignificant in terms of the system's efficiency, but it does have consequences when considering the power rating of the damping resistors. Ventilation must now be taken into account, as a significant amount of heat must now be evacuated. In this system, 150 W damping resistors are used, which are mounted to a heat sink for

Table 5.1: *Physical parameters of the WECS.*

Parameter	Value
Rated line voltage	400 V
Rated current	35 A_{rms}
Rated rotational speed	150 rpm
Rated torque	1000 Nm
Pole pairs (N_p)	20
Stator resistance (R_s)	0.4 Ω
Stator inductance (L_s)	10 mH
PM flux linkage (Φ_m)	1.0395 Wb-t
Filter inductance (L_f)	600 μH
Filter inductor resistance (r_f)	30 $m\Omega$
Filter capacitance (C_f)	47 μF
Damping resistance (R_d)	10 Ω
Turbine moment of inertia (J)	300 kg·m^2
Switching frequency (f_s)	10 kHz

heat evacuation. The maximum operating frequency of this system is limited to 40 Hz, mainly due to this issue.

5.4.1 State Estimattion

In this section practical results of the estimation are presented . Because of the difficulty in gauging in a practical system, the flux observer output will not be presented.

The EMF and rotor position estimation results are obtained at a low speed, in the range of 15-18 Hz, and a higher speed, in the range of 35-40 Hz. The results for the low and high speeds are given in Fig. 5.4 and Fig. 5.5, respectively. The voltages are in the $\alpha\beta$-frame.

As is clear from the plots, the estimated EMF voltages are far from the near-perfect sinusoidal results obtained in the simulations. The quality of the estimation does, however, improve significantly at higher speeds. This is due to the fact that the estimation is based on the generator EMF voltage, which increases in magnitude with an increase in speed.

The estimated rotor position obtained from the EMF voltages, for low and high speeds, are given in Fig. 5.6 and Fig. 5.7, respectively.

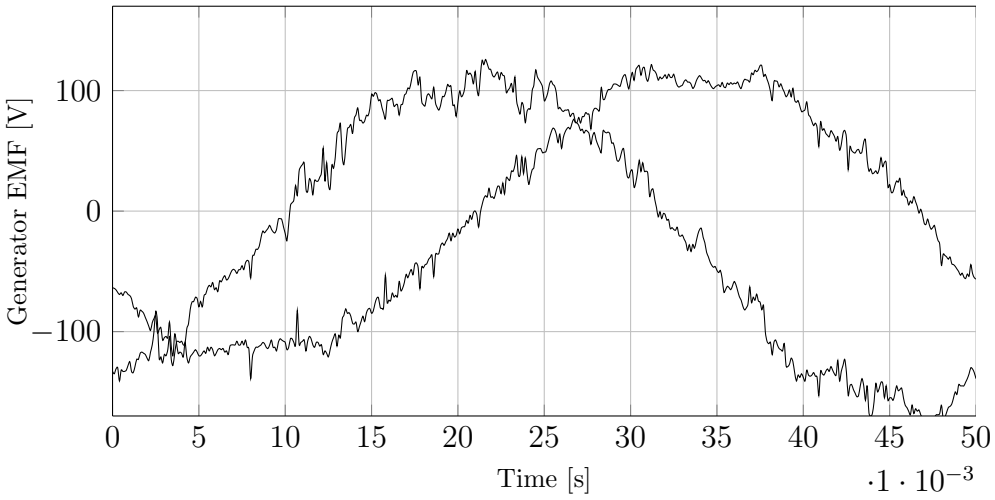


Figure 5.4: *Practical estimated generator EMF ($\alpha\beta$ reference frame) at low speed.*

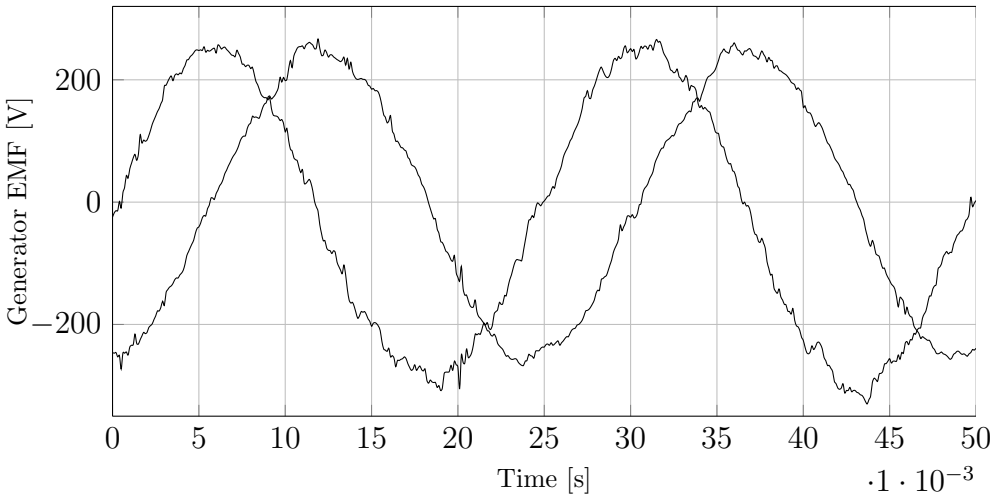


Figure 5.5: *Practical estimated generator EMF ($\alpha\beta$ reference frame) at high speed.*

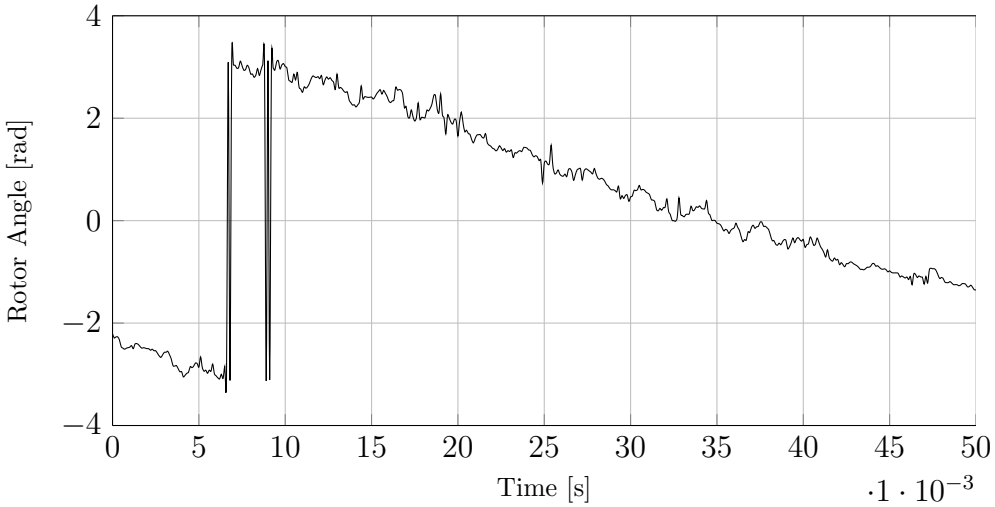


Figure 5.6: *Practical estimated rotor position at low speed.*

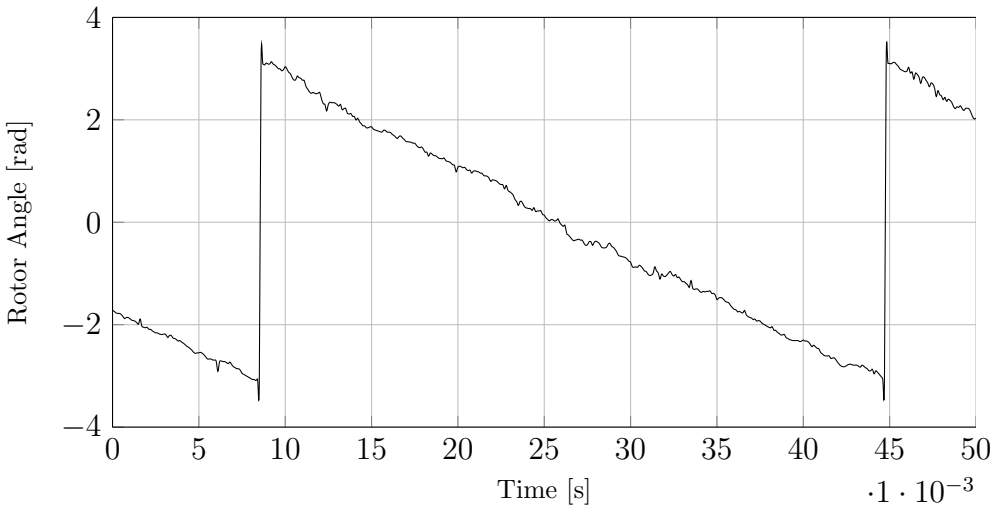


Figure 5.7: *Practical estimated rotor position at high speed.*

As the rotor position estimation is obtained from the EMF voltages, its quality is dependent on the quality of the EMF voltage estimation. This is clear from the results, as the low speed position estimation resembles the corresponding noisy, low speed EMF estimation.

It is possible to remove most of the noise, which is mostly due to the converter switching, by digital filtering.

5.4.2 Torque Steps

Practical results of the operation of the DTC are presented here. Positive and negative 300 Nm steps are given as references to the torque controller. The positive torque steps are initiated at an initial frequency of about 18 Hz, while the negative steps are initiated at a frequency of about 30 Hz.

The response of the system to the positive 300Nm step is given in Fig. 5.8, along with the simulated response in VHDL-AMS. The output from the torque sensor was unusable in this test, due to severe noise.

It is clear that the practical response differs somewhat from the simulated response, mostly in terms of damping. The settling times and steady state correlate acceptably. A possible reason for the difference in response is a difference in the actual inertia of the rotating parts, compared to the simulated value for inertia.

Another possible explanation is the reaction of the VSD controller. It is already known that the VSD, under torque control, is to blame for the severe ripple seen on the torque response.

The result for the negative 300 Nm torque step are given in Fig. 5.9, along with

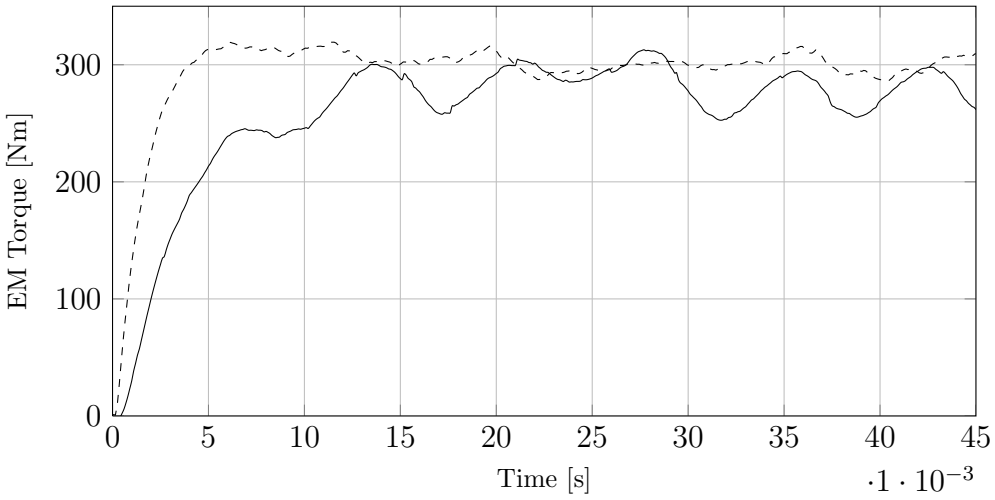


Figure 5.8: *Measured electromagnetic torque (filter-generator) step in rotor speed acceleration direction (up-step), including corresponding simulated VHDL-AMS response.*

the simulated VHDL-AMS response. It is clear that the ripple on the torque response for negative step is less than for the positive step.

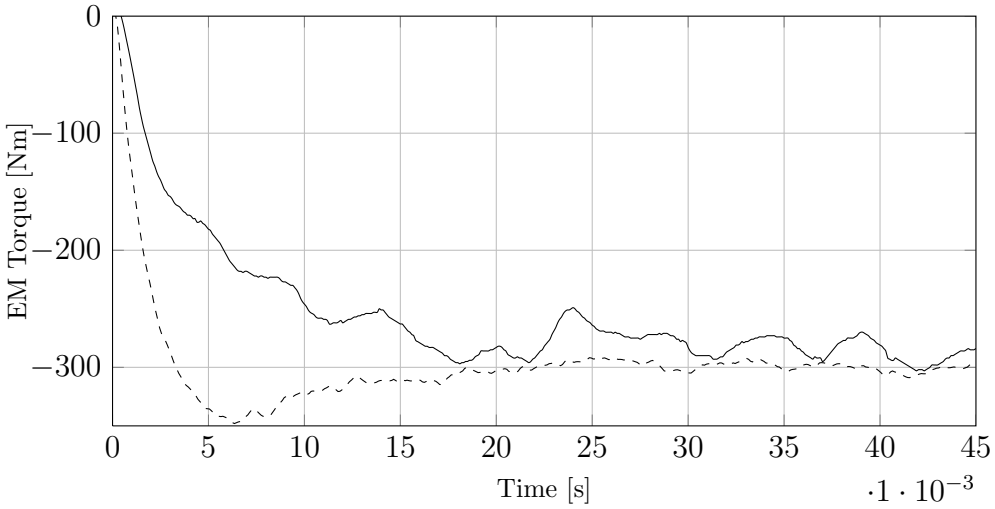


Figure 5.9: *Measured electromagnetic torque (filter-generator) step in rotor speed deceleration direction (down-step), including corresponding simulated VHDL-AMS response.*

The torque responses of the DTC are satisfactory and correlate well with the predicted responses, except for the difference in damping.

5.4.3 Speed Control

The practical results of the constant speed controller are presented in this section. A speed step and disturbance rejection test are performed. A caveat of this controller is that it relies strongly upon the speed estimation, which in this system is not a very

good estimation.

After synchronisation, the speed is controlled at *Cut-in* speed; a small step is then made in the speed reference of about 10 %, of which the response is given in Fig. 5.10.

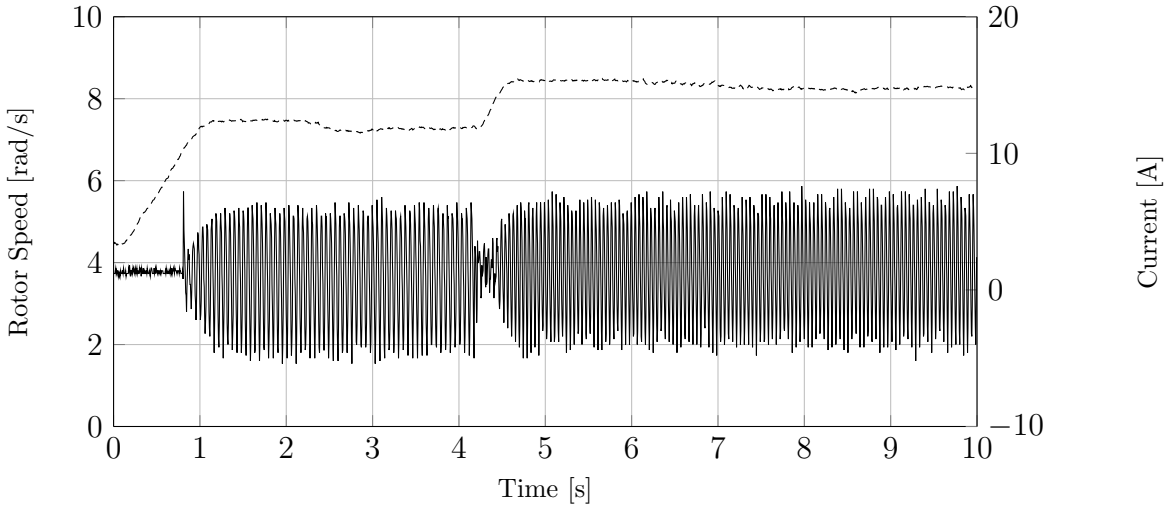


Figure 5.10: *Measured phase* a *stator current and mechanical rotor speed response of speed control after* Cut-in*, and then a small speed step.*

The controller responds to the step well, with a slight overshoot, and settles after about 3 s. This type of bandwidth is acceptable in this system, as the wind speed is generally assumed to vary slowly.

The response of the speed controller to an increasing load disturbance is given in Fig. 5.11. The speed increases only slightly and stabilises, with a continued increase in load.

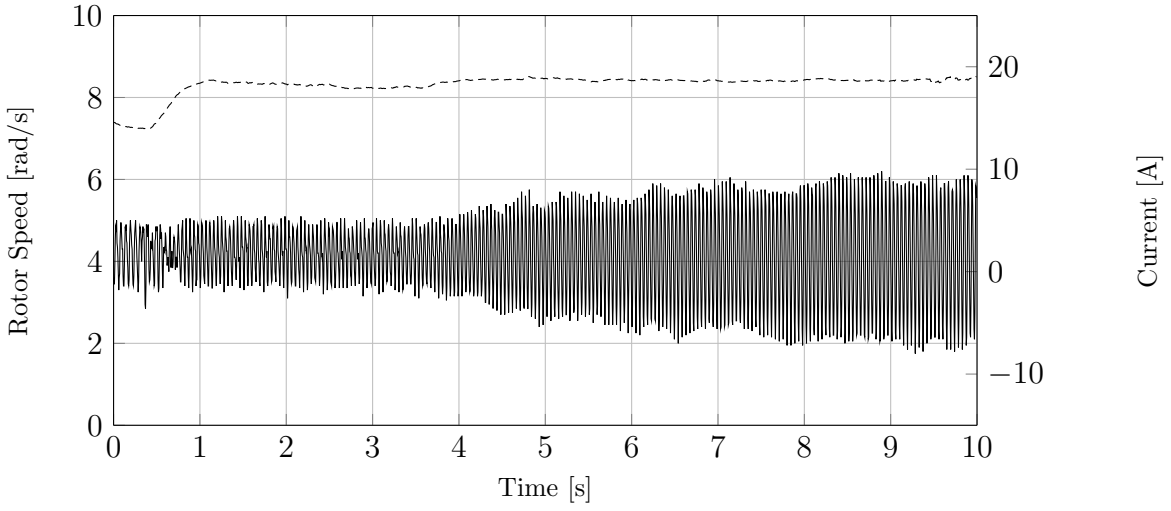


Figure 5.11: *Measured phase* a *stator current and mechanical rotor speed response of speed control on load increase.*

These results verify acceptable performance of the speed controller, which is an integral part of this system.

5.5 WECS Test Results

Test results of the complete WECS are presented in this section. The system is subjected to loads resembling increasing and decreasing wind speeds, in order that the most important state transitions are covered. Rotor speed and phase current are plotted to demonstrate the response of the system.

The results presented in this section are the culmination of the work done in this thesis.

5.5.1 Transition from Pre-cut-in to Cut-in

The concept gain scheduling (GS) state diagram, highlighting the transition from *Pre-cut-in* to *Cut-in*, is shown in Fig. 5.12.

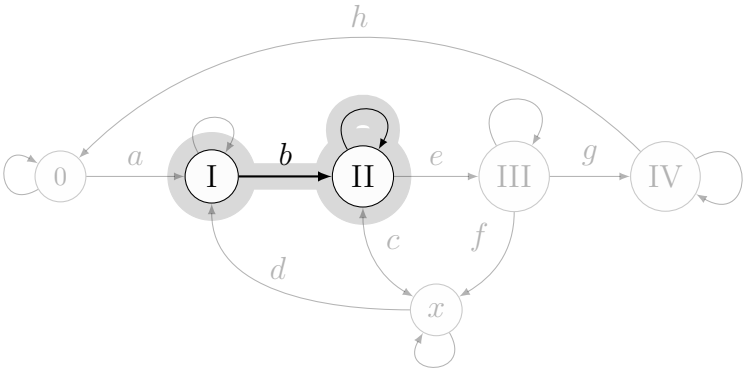


Figure 5.12: *GS state diagram showing the state transition from the Pre-cut-in state to the Cut-in state.*

The measured mechanical rotor speed and phase *a* stator current for this transition are given in Fig. 5.13. It can be seen that the generator mechanical rotor speed increases until it reaches the cut-in speed just after 3 s. The converter synchronises with the generator, which is then controlled at the cut-in speed.

The state transition is successful and the system now resides in the *Cut-in* state.

5.5.2 Transition from Cut-in to MPPT

The concept GS state diagram, highlighting the transition from *Cut-in* to *MPPT*, is shown in Fig. 5.14.

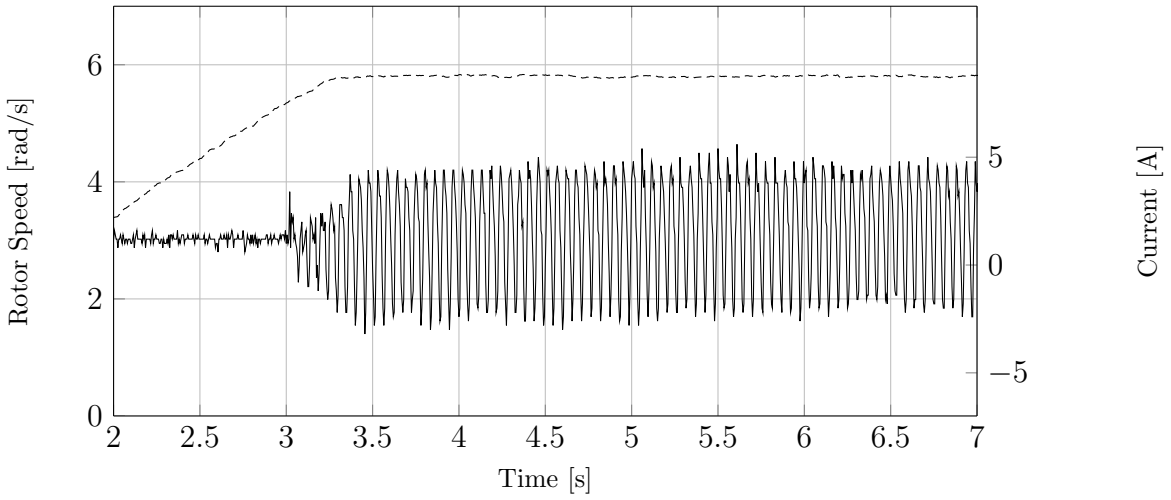


Figure 5.13: *Measured phase* a *stator current and mechanical rotor speed response of the state transition from Pre-cut-in to Cut-in*

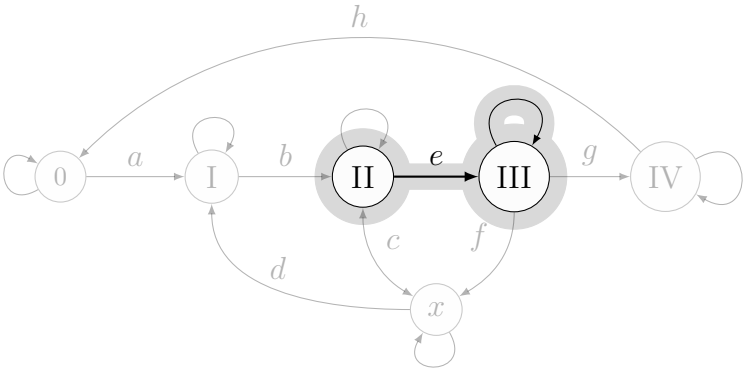


Figure 5.14: *GS state diagram showing the state transition from the Cut-in state to the MPPT state.*

The measured mechanical rotor speed and phase *a* stator current for this transition are given in Fig. 5.15. The load increases steadily, represented by the increasing phase current, and the system then transitions to the *MPPT* state at about 4.5 s. The variable speed and corresponding load increase is clear after the transition.

This state transition is successful and the system now resides in the *MPPT* state.

5.5.3 Transition from MPPT to Max. Speed Limit

The concept GS state diagram, highlighting the transition from *MPPT* to *Max. Speed Limit*, and vice versa, is shown in Fig. 5.16.

The measured mechanical rotor speed and phase *a* stator current for this transition are given in Fig. 5.17. The maximum speed is reached at about 2 s, and the system transitions to the *Max. Speed Limit* state, where the generator is controlled at the

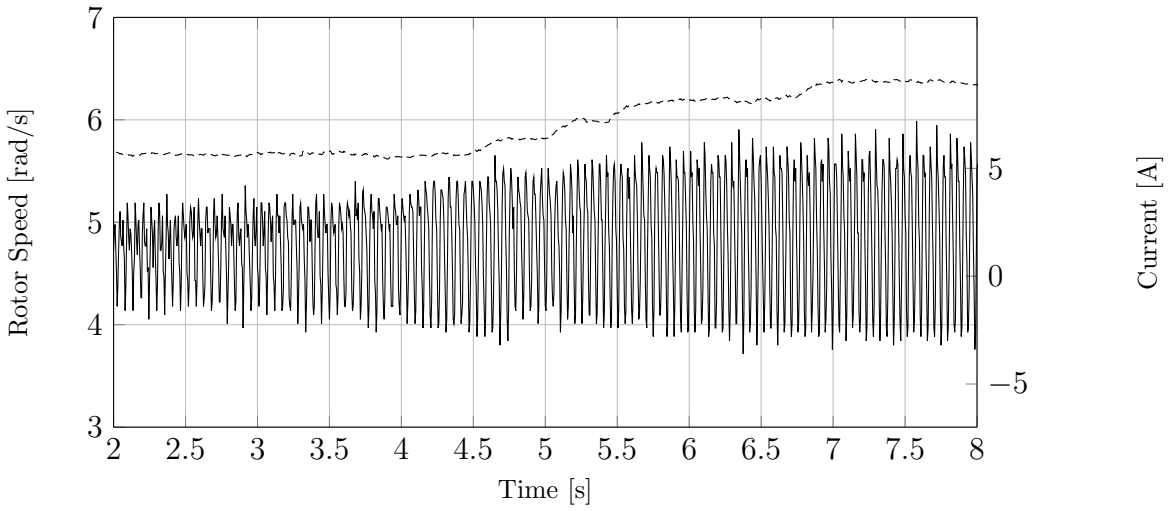


Figure 5.15: *Measured phase* a *stator current and mechanical rotor speed response of state transitions from Cut-in to MPPT.*

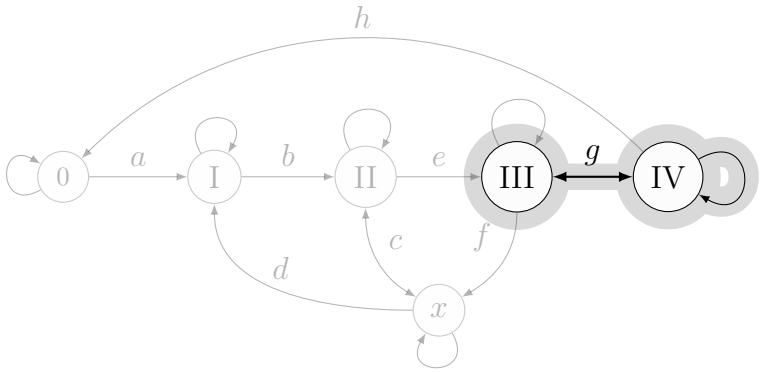


Figure 5.16: *GS state diagram showing the state transition from the MPPT state to the Max Speed Limit state.*

maximum speed of 40 Hz. This translates to roughly 12.5 rad/s in mechanical rotor speed, as in the plot.

As is clear from the plot, here are some oscillations on the measured current. This can be attributed to the torque control system of the VSD driving the induction motor.

This state transition is successful and the system now resides in the *Max. Speed Limit* state. If the load increases sufficiently, the converter will de-synchronise, and the system will transition to the *Pre-states* state. The latter transition is not tested in the practical implementation.

5.5.4 Re-Entry From Maximum Speed to MPPT

The measured mechanical rotor speed and phase *a* stator current for this transition are given in Fig. 5.18. As the load is decreased, an initial decrease in rotor speed can

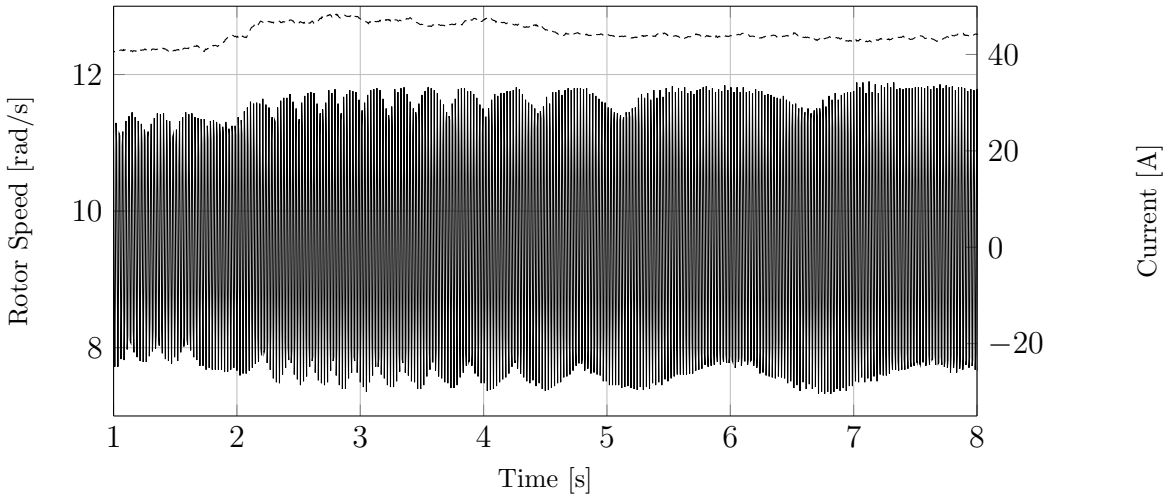


Figure 5.17: *Measured phase* a *stator current and mechanical rotor speed response of the state transition from MPPT to Max. Speed Limit.*

be observed. This is simply the speed controller that can not keep up with the rate of load decrease. The actual re-entry transition to the *MPPT* state occurs at about 4.2 s. A transient on the current makes the transition clear.

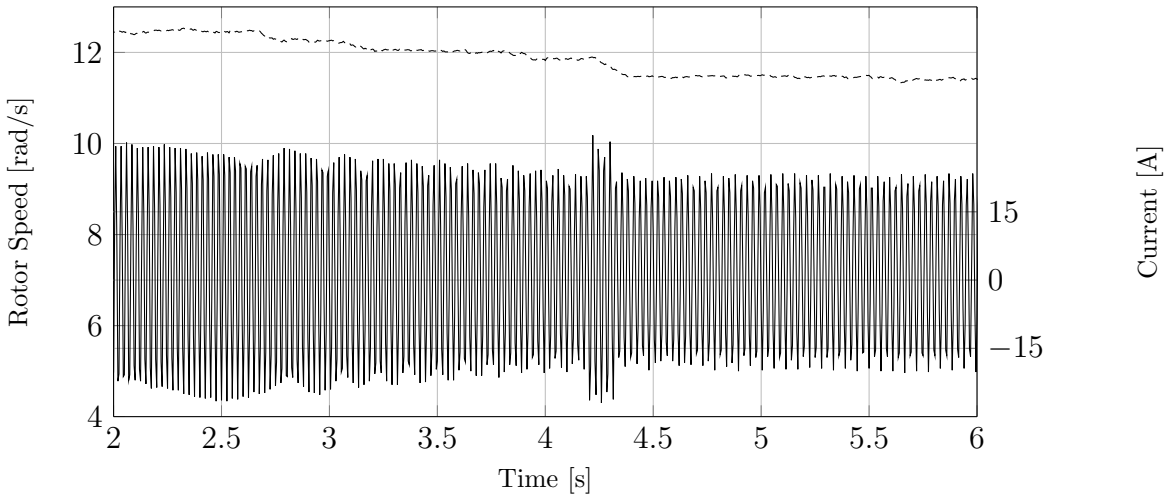


Figure 5.18: *Measured phase* a *stator current and mechanical rotor speed response of the state transition from Max speed re-entry to MPPT.*

The measurement from the mechanical torque sensor for this transition is given in Fig. 5.19. It is interesting to note that the developed torque prior to the transition nears the rated torque of the generator; this is dictated by the turbine blade power characteristics. The transient in the stator current is also reflected in the torque measurement.

The system successfully limits the specified maximum speed, and then successfully re-enters the variable speed (*MPPT*) state when the conditions permit it.

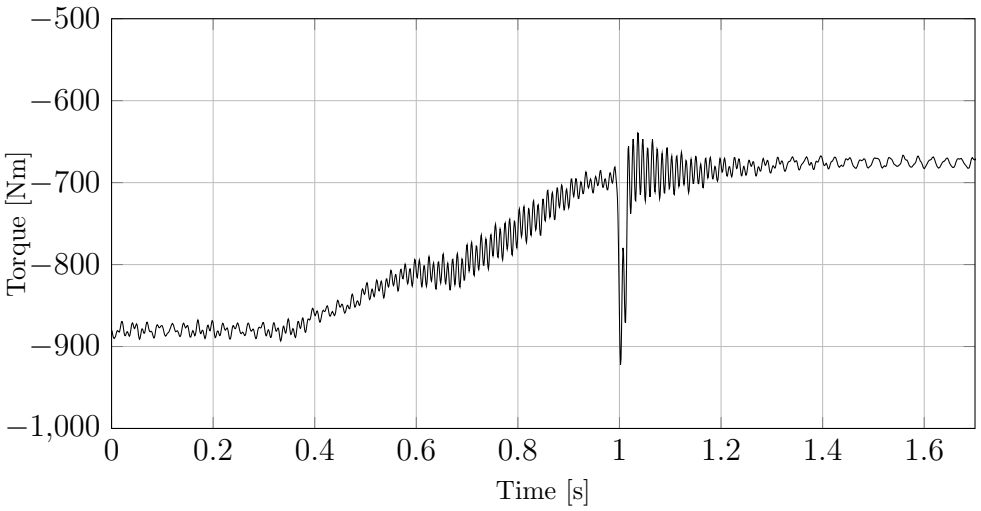


Figure 5.19: *Mechanical torque sensor measurement for the re-entry transition from* Max. Speed Limit *to the* MPPT *state.*

5.5.5 Cut-out Transition

The concept GS state diagram, highlighting the transition from *MPPT* to *Cut-out*, and then finally to *Pre-cut-in*, is shown in Fig. 5.20.

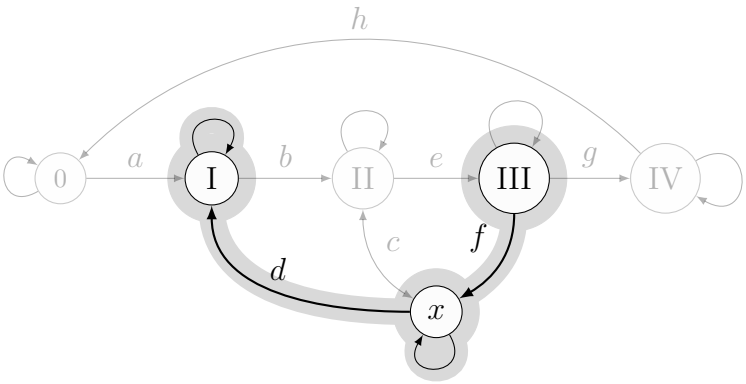


Figure 5.20: *GS state diagram showing the state transition from the MPPT state to Cut-out, and then finally to Pre-cut-in.*

The mechanical rotor speed and phase *a* stator current for this transition are given in Fig. 5.21. The system is initially in the *MPPT* region, with the load, and consequently the speed, steadily decreasing. When the speed is below the cut-in speed, the system transitions to the *Cut-out* state. This transition occurs at about 6.5 s.

When the load decreases sufficiently, the system de-synchronises, and transitions to the *Pre-cut-in* state. This final transition occurs at about 8.5 s.

The system functions as desired, and the transitions are successful.

A mechanical torque sensor measurement, showing transitions from *Pre-cut-in* to

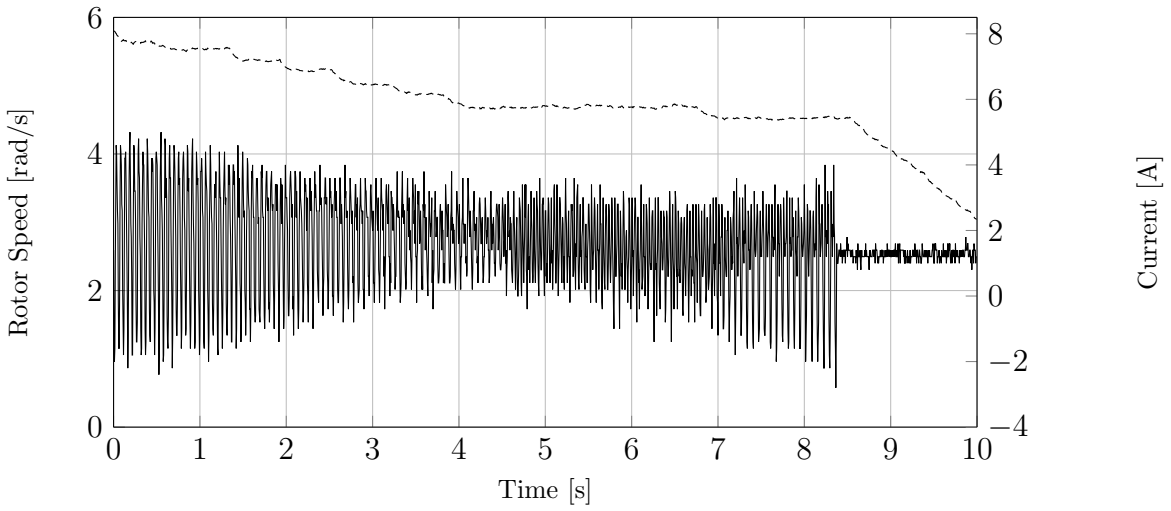


Figure 5.21: *Measured phase* a *stator current and mechanical rotor speed response of the state transition from* MPPT *to* Cut-out.

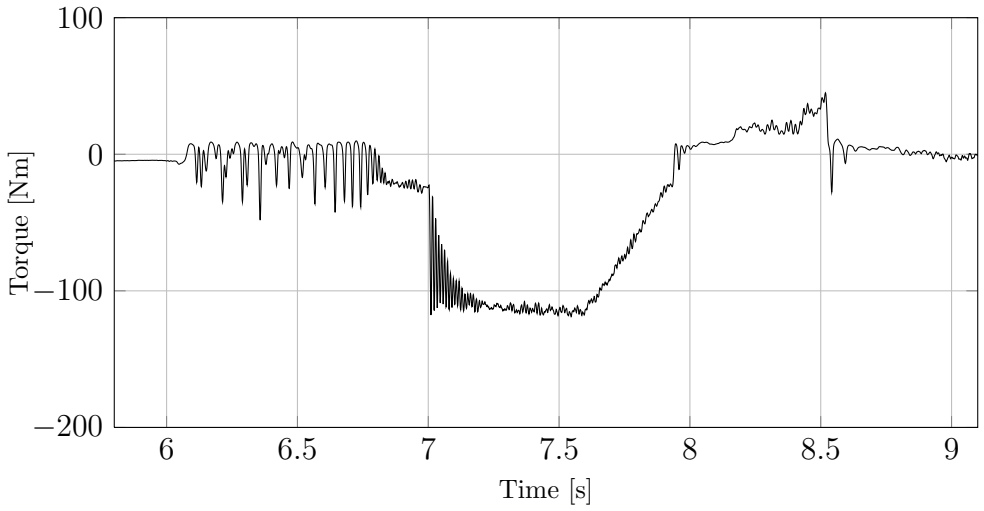


Figure 5.22: *Mechanical torque sensor output showing transitions from Pre-cut-in to Cut-in, to MPPT, to Cut-out and then finally to Pre-cut-in.*

Cut-in, *Cut-in* to *MPPT*, *MPPT* to *Cut-out*, and then finally back to *Pre-cut-in*, is given in Fig. 5.22.

The system transitions to *Cut-in* just after 6 s, to *MPPT* at about 7 s, to *Cut-out* just before 8 s, and then finally de-synchronises just after 8.5 s. These transitions occur in a very short time of about 2.5 s, from synchronisation to de-synchronisation. This demonstrates the robustness of the system to sudden changes in wind conditions.

5.6 Summary and Discussion

The practical implementation, testing methodology and test results of the system in question are presented. The testing methodology is given, after which a summary of implementation issues encountered is presented.

Test results of the various sub-systems comprising the WECS are presented, and their performance is scrutinised and validated. The practical torque step response of the system is compared to the simulated response in VHDL-AMS. They are found to correlate well, except for differences in damping.

The speed controller is tested by subjecting it to a speed step, as well as testing its robustness to load disturbances. The results are satisfactory, and valid for implementation in the full WECS.

Finally, the complete WECS is tested by varying the load, and allowing the GS system to traverse the states. It does so successfully, and the working of the complete system is verified. A final test, represented by mechanical torque measurements, demonstrates the robustness of the system, even when subjected to sudden changes in wind.

Chapter 6

Conclusion

6.1 Summary and Conclusions

The initial goals of this thesis, as set out in Section 1.2, can be summarised as follows: To model, design and implement a complete position sensorless wind energy conversion system (WECS), incorporating direct torque control with space vector modulation (DTC-SVM) of a permanent magnet generator (PMSG), with an RLC filter.

In the first chapter of this thesis, the above mentioned goal statement is introduced and supported by a summary of relevant literature. From the literature discussed, it is clear that the effective control of a WECS is the biggest challenge in this field of study. It is also believed, as is stated in Section 1.2, that the proposed system conforms to the current, and more importantly, future technological trends, as indicated by the literature. The objective, then, of the research conducted and presented in the subsequent chapters, is to find a solution to the previously stated challenge, with regards to the proposed system.

In the second chapter, mathematical models of the system are derived and various fundamental concepts are presented and discussed. Reference frame theory is applied to the base electrical model of the system, in order to obtain the stationary and rotating reference frame models. It is clear, especially when considering the stator flux synchronously rotation reference frame (xy reference frame), that the complexity of the system model increases significantly with the inclusion of the RLC filter. The complete model of the system is designated as the *filter-generator* model.

The coordinate transformations are not only necessary to simplify the design and simulation of the system, but are required by the sub-systems comprising the WECS. For example, the vector control method that is implemented in this system, viz. parallel structure DTC-SVM, requires the system model to be expressed in the stator flux

synchronously rotating reference frame, whereas the EMF estimation method requires a stationary frame model.

Approximate transfer functions of the flux linkage and torque controller plants, referring to the parallel structure DTC-SVM, are derived presented; these transfer functions are helpful when designing or tuning the PI controller gains. The coupling effects between the components of the xy reference frame are neglected in this derivation. The need for an RLC filter is also motivated, and the implications of its inclusion are discussed.

In the third chapter, equivalent closed-loop digital transfer functions for the parallel structure DTC are derived, and then used to obtain satisfactory gains for the conrollers. These transfer functions can be used to analytically determine the optimal gains, but only manual tuning is used in this system. The gains for the EEMF and flux linkage observers, as well as the speed controller, are given. These are also determined by manual tuning.

The design of the RLC filter components is presented, neglecting the effect of the damping resistance. An expression for the resonance frequency of the filter-generator model is derived. A plot of the resonance frequency versus the value of the damping resistance shows quite a significant decrease in resonance frequency, with an increase in damping resistance. This actually shows that a great deal of consideration must be given to the presence of the damping resistance when designing the other filter components.

A gain scheduling (GS) state machine concept, with the objective of governing the top-level control of the WECS, is presented. The maximum power point tracking (MPPT) controller of the system, which is essentially a rotor speed dependent torque reference, is presented and discussed. This algorithm is inherently sensitive to turbine blade parameter variation, as its coefficients depend on turbine blade power data. The simplicity, however, of implementing MPPT with a torque controller system is clear.

In the fourth chapter, simulation models and methodologies of the system and simulation results, are presented. An integral part of this thesis is the implementation of the system in a VHDL-AMS simulation, as as this not widely publicised. The system is simulated in both Simulink and VHDL-AMS, of which the former is mostly done to verify the latter.

A torque step simulation is done for both the Simulink and VHDL-AMS simulations, where a 500 Nm torque step command is given to the DTC-SVM; this is done with initial system frequencies of 20 Hz and 40 Hz, so that the whole operating range of the WECS is covered. The results demonstrate the good performance of the estimators,

as well as the DTC-SVM. The response of the torque controller is fast, with minimum overshoot, and the steady state error is minor. The flux linkage controller demonstrates exceptional accuracy and disturbance rejection, as its main objective is constant flux control.

The VHDL-AMS simulation results correlate very well with the Simulink simulation resuls. The only noticeable anomalies are a difference in the overshoot of the torque controller response, the flux linkage magnitude steady state error and a high frequency component in the torque and flux linkage responses of the VHDL-AMS simulation. Some of these differences can be attributed to the more accurate IGBT and reverse diode models used in the VHDL-AMS simulation.

A comparison of the performance of the MRAS speed estimator and the derivative-filter method is presented and discussed. It is concluded that the much simpler derivative-filter method yields superior performance, and over a wider operating range, than more complex MRAS speed observer.

In the final part of the chapter, a strategy is presented to evaluate the performance of the MPPT controller in Simulation. A wind turbine simulator (WTS) is implemented to more accurately model the effect of the turbine. The system in the VHDL-AMS simulation, incorporating the WTS, is subjected to wind speed inputs that resemble quite rapidly changing wind conditions. The system performs very well, and tracks the maximum power points quickly. This is an important simulation, as it demonstrates that, in theory at least, the variable speed state of the system does what is has been designed to do.

In the fifth and final chapter, practical implementation and testing methodologies of the system are presented. Practical issues that were encountered during implementation and testing are also discussed.

Existing software that is responsible for grid synchronisation of the grid-side converter and DC bus voltage control, which is extended and modified for this implementation, is briefly discussed. The software that is developed for this system is then discussed. The final implementation of the gain scheduling (GS) is presented and discussed, with the help of flow diagrams.

The practical test results commence with results of the estimation at low and high speeds, of which only the EMF observer and subsequent rotor position estimation are presented. The EMF estimation at low speeds is very noisy, with an equally noisy rotor position estimation; the noise is mitigated to a large degree by digital filtering.

The results of torque steps of ± 300 Nm are presented and compared to simulated responses of the VHDL-AMS simulation. The results correlate well, but deviate in

terms of damping. This can be attributed to inaccurate models and system parameters, such as the moment of inertia of the rotating parts, and also the torque controller of the VSD.

Results of the speed controller response to a speed step, as well as subjecting the system to a load disturbance, are presented. The tests show good, stable performance of the speed controller, as well as sufficient robustness to disturbances. Although the response of the speed controller seems slow, it is more than sufficient when considering the large average time constant of wind.

The test results of the complete WECS are then finally presented. The system is subjected to mechanical load variation, representing wind speed variation. The GS algorithm successfully traverses the various states with stable transitions. Torque measurement results of a final test demonstrates the good performance of the system, even under rapidly changing wind conditions.

With reference to the previous summary of the work done, it can confidently be stated that the objectives of this thesis have been reached.

6.2 Recommendations for Future Work

To stimulate critical thinking towards future research topics, the following recommendations are given:

- More analysis can be done on the derived filter-generator model of the system.
- The approximate transfer functions presented in Section 2.3.2 can be used for more detailed analysis of the system, such as analysis on the closed loop dynamics, including pole-zero and bode analysis.
- More robust estimation methods, such as Kalman filtering, should be investigated.
- An investigation on the stability of the proposed MRAS speed observer method could be done.
- Mathematical analysis and modelling of the gain scheduling system, so that analytical methods can be developed to determine the optimal thresholds.
- Analytical design of the estimators.
- A comparison study on the performance of the proposed system as compared to other similar systems, and even different topologies.

- It would be interesting to implement this system in practice, to evaluate its performance when subjected to real wind.

It is the hope of the author that this work will not only contribute to the field of study, but that the shortcomings of the research will give rise to more research endeavours.

Bibliography

[1] A. D. Hansen and L. H. Hansen, "Market penetration of wind turbine concepts over the years," in *European Wind Energy Conference and Exhibition*, Brussels, Belgium, 2007.

[2] A. Rolan, A. Luna, G. Vazquez, D. Aguilar, and G. Azevedo, "Modeling of a variable speed wind turbine with a Permanent Magnet Synchronous Generator," in *IEEE International Symposium on Industrial Electronics*. Seoul, Korea: IEEE, Jul. 2009, pp. 734–739.

[3] T. Ackermann, G. Ancell, L. Borup, P. Eriksen, B. Ernst, F. Groome, M. Lange, C. Mohrlen, A. Orths, J. O'Sullivan, and M. de la Torre, "Where the wind blows," *IEEE Power and Energy Magazine*, vol. 7, no. 6, pp. 65–75, Nov. 2009.

[4] A. Perdana, "Dynamic Models of Wind Turbines A Contribution towards the Establishment of Standardized Models of Wind Turbines for Power System Stability Studies," Ph.D. dissertation, 2008.

[5] E. Muljadi, K. Pierce, and P. Migliore, "A Conservative Control Strategy for Variable-Speed Stall-Regulated Wind Turbines," in *ASME Wind Energy Symposium*, Reno, NV, USA, January 2000.

[6] J. Stegmann and M. Kamper, "Economic and Efficiency Evaluation of Different Battery Charging Wind Generator Systems," in *Southern African Universities Power Engineering Conference*, Johannesburg, South Africa, January 2010, pp. 205–210.

[7] M. Karrari, W. Rosehart, and O. P. Malik, "Comprehensive control strategy for a variable speed cage machine wind generation unit," *IEEE Transactions on Energy Conversion*, vol. 20, no. 2, pp. 415–423, June 2005.

[8] H. Li and Z. Chen, "Overview of different wind generator systems and their comparisons," *IET Renewable Power Generation*, vol. 2, no. 2, pp. 123–138, 2008.

[9] B. Singh and G. K. Kasal, "Voltage and frequency controller for a three-phase four-wire autonomous wind energy conversion system," *IEEE Transactions on Energy Conversion*, pp. 509–518, June 2008.

[10] R. Vermaak, M. J. Kamper, and J. H. Potgieter, "Grid-Connected VSC-HVDC Wind Farm System and Control Using Permanent Magnet Induction Generators," in *IEEE International Conference on Power Electronics and Drive Systems*, Taipei, Taiwan, November 2009, pp. 897–902.

[11] K. G. Pierce and P. G. Migliore, "Maximizing Energy Capture of Fixed-Pitch Variable-Speed Wind Turbines," in *AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, January 2000.

[12] G. Putrus, M. Narayana, M. Jovanovic, and P. S. Leung, "Maximum power point tracking for variable-speed fixed-pitch small wind turbines," in *20th International Conference and Exhibition on Electricity Distribution*, Prague, Czech Republic, June 2009, pp. 1–4.

[13] S. A. Diaz, C. Silva, J. Juliet, and H. A. Miranda, "Indirect sensorless speed control of a PMSG for wind application," in *International Electric Machines and Drives Conference*, Miami, Florida, USA, May 2009, pp. 1844–1850.

[14] T. A. Haskew, "Characteristic study of vector-controlled direct driven permanent magnet synchronous generator in wind power generation," in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, Piscataway, NJ, USA, Jul. 2008, pp. 1–9.

[15] R. Teodorescu and F. Blaabjerg, "Flexible control of small wind turbines with grid failure detection operating in stand-alone and grid-connected mode," *IEEE Transactions on Power Electronics*, vol. 19, no. 5, pp. 1323–1332, September 2004.

[16] Z. Chen and E. Spooner, "Wind Turbine Power Converters: A Comparative Study," in *International Conference on Power Electronics and Variable Speed Drives*, London, UK, September 1998, pp. 471–476.

[17] C. Cecati, A. DellAquila, A. Lecci, and M. Liserre, "A Current-Sensorless Three-Phase Active Rectier with Fuzzy-Logic Control ," in *IEEE Industry Applications Conference*, vol. 4, Seattle, Washington, October 2004, pp. 2609–2614.

[18] J. Pontt, J. Rodriguez, M. Rotella, S. Kouro, C. Silva, and H. Farias, "Output sinus filter for medium voltage drive with direct torque control," in *Fourtieth IAS*

Annual Meeting. Conference Record of the 2005 Industry Applications Conference, Kowloon, Hong Kong, China, October 2005, pp. 204–209.

[19] A. Sapin, P. Steimer, and J.-J. Simond, "Modeling, simulation and test of a three-level voltage source inverter with output LC filter and direct torque control," in *38th IAS Annual Meeting on Conference Record of the Industry Applications Conference*, Salt Lake City, Utah, USA, October 2003, pp. 492–498.

[20] W. Zhao and G. Chen, "Comparison of active and passive damping methods for application in high power active power filter with LCL-filter," in *International Conference on Sustainable Power Generation and Supply*, Nanjing, China, Apr. 2009, pp. 1–6.

[21] K. Hatua, A. Jain, D. Banerjee, and V. Ranganathan, "Active Damping of Output LC Filter Resonance for Vector Controlled VSI Fed AC Motor Drives." *IEEE Transactions on Industrial Electronics*, no. 99, p. 1, 2011.

[22] N. V. S. K. Srighakollapu, "Sensorless Maximum Power Point Tracking Control in Wind Energy Generation using Permanent Magnet Synchronous Generator," in *34th Annual Conference of the IEEE Industrial Electronics Society*, Orlando, Florida, USA, Nov. 2008, pp. 2225–2230.

[23] Z. Xu, T. Pang, and X. Dianguo, "Instantaneous torque control of a permanent magnet wind power generator without a position sensor," in *IEEE International Conference on Electrical Machines and Systems*, Wuhan, China, Oct. 2008, pp. 1343–1347.

[24] C. Busca, A.-i. Stan, T. Stanciu, and D. I. Stroe, "Control of Permanent Magnet Synchronous Generator for Large Wind Turbines," in *IEEE International Symposium on Industrial Electronics*, no. 1, Bari, Italy, 2010, pp. 3871–3876.

[25] J. M. Kwon, J. H. Kim, S. H. Kwak, and H. H. Lee, "Optimal power extraction algorithm for dtc in wind power generation systems," in *IEEE International Conference on Sustainable Energy Technologies*, no. 2, SMU Conference Centre, Singapore, Nov. 2008, pp. 639–643.

[26] Y. Inoue, S. Morimoto, and M. Sanada, "Output maximization using direct torque control for sensorless variable wind generation system employing IPMSG," in *13th IEEE International Power Electronics and Motion Control Conference*, Poznań, Poland, Sep. 2008, pp. 1859–1865.

[27] T. Schneider, J. Mades, and M. Glesner, "An Open VHDL-AMS Simulation Framework," in *IEEE International Workshop on Behavioral Modeling and Simulation*, Orlando, Florida, USA, 2000, pp. 89–94.

[28] J.-J. Charlot, N. Lews, T. Zimmer, and H. Levi, "VHDL-AMS for Mixed Technology and Mixed Signal, An Overiew," in *9th IEEE Mediterranean Conference on Control and Automation*, Dubrovnik, Croatia, 2001.

[29] P. C. Krause, *Analysis of Electric Machinery*. New York: McGraw-Hill, 1986.

[30] I. K. Hobbs, "Evaluation and implementation of anti-islanding methods for converter fed distributed generation," Master's thesis, University of Stellenbosch, Stellenbosch, South Africa, 2009.

[31] S. Mastellone, G. Papafotiou, and E. Liakos, "Model Predictive Direct Torque Control for MV drives with LC filters," in *13th European Conference on Power Electronics and Applications*, Barcelona, Spain, Sep. 2009, pp. 1–10.

[32] P. Bouwer, J. H. Potgieter, and M. J. Kamper, "Modelling and dynamic performance of a direct-drive direct-grid slip permanent magnet wind generator," in *IEEE International Electric Machines & Drives Conference*, Niagara Falls, Canada, May 2011, pp. 137–142.

[33] S. Buso and P. Mattavelli, *Digital control in power electronics*, ser. Synthesis Lectures on Power Electronics. Morgan & Claypool, Nov. 2006, vol. 2.

[34] D. Swierczynski, "Direct Torque Control with Space Vector Modulation (DTC-SVM) of Inverter-Fed Permanent Magnet Synchronous Motor Drive," Ph.D. dissertation, 2005.

[35] D. Swierczynski and M. P. Kazmierkowski, "Dircet Torque Control of Permanent Magnet Synchronous (PMSM) Using Space Vector Modulation (DTC-SVM) - Simulation and Experimental Results," in *Proceedings of the IEEE International Symposium on Industrial Electronics*, L'Aquila, Italy, 2002, pp. 751–755.

[36] B. Wu, *High-Power Converters and AC Drives*. John Wiley & Sons, 2006.

[37] G. Buja and M. Kazmierkowski, "Direct Torque Control of PWM Inverter-Fed AC MotorsA Survey," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 4, pp. 744–757, Aug. 2004.

[38] S. Yang, X. Zhang, C. Zhang, Z. Xie, and F. Li, "Sensorless control for PMSG in direct-drive wind turbines," in *The 2nd IEEE International Symposium on Power Electronics for Distributed Generation Systems*, Hefei, China, Jun. 2010, pp. 81–84.

[39] M. P. Kazmierkowski, R. Krishnan, and F. Blaabjerg, *Control in Power Electronics*, 1st ed. Academic Press, 2002.

[40] J. Guzinski, "Closed loop control of ac drive with LC filter," in *13th IEEE International Power Electronics and Motion Control Conference*, Poznań, Poland, Sep. 2008, pp. 994–1001.

[41] F. D. Bianchi, H. D. Battista, and R. J. Mantz, *Wind Turbine Control Systems: Principles, Modelling and Gain Scheduling Control*. Springer-Verlag London Limited, 2007.

[42] M. Liserre, F. Blaabjerg, and S. Hansen, "Design and Control of an LCL-Filter-Based Three-Phase Active Rectifier," *IEEE Transactions on Industry Applications*, vol. 41, no. 5, pp. 1281–1291, Sep. 2005.

[43] A. A. Adam and K. Gulez, *Torque Control*, M. T. Lamchich, Ed. InTech, 2011.

[44] G. Moor and H. Beukes, "Maximum power point trackers for wind turbines," in *35th IEEE Annual Power Electronics Specialists Conference*, Aachen, Germany, 2004, pp. 2044–2049.

[45] A. Mirecki, X. Roboam, and F. Richardeau, "Comparative Study of Maximum Power Strategy in Wind Turbines," in *IEEE International Symposium on Industrial Electronics*, Palais des Congrès Expositions, Ajaccio, France, 2004, pp. 993–998.

Appendix A

Derivation of the xy Reference Frame Filter-Generator Model

The xy-$\alpha\beta$ coordinate transformation in (2.6) is applied to the (slightly modified) stationary frame ($\alpha\beta$) model (2.5) of the filter-generator:

$$\vec{v}_{i,xy} = R_f \vec{i}_{f,xy} + L_f K_{xy} \frac{d\vec{i}_f}{dt} + \vec{v}_{o,x}, \tag{A.1a}$$

$$\vec{v}_{o,xy} = \vec{v}_{c,xy} + R_d \vec{i}_{c,xy}, \tag{A.1b}$$

$$\vec{i}_{c,xy} = C_f K_{xy} \frac{\vec{v}_c}{dt}, \tag{A.1c}$$

$$\vec{v}_{s,xy} = R_s \vec{i}_{s,xy} + K_{xy} \frac{d\vec{\Phi}_s}{dt}, \tag{A.1d}$$

$$\vec{\Phi}_{s,xy} = L_s \vec{i}_{s,xy} + \Phi_f K_{xy} \begin{bmatrix} \sin\theta_r \\ \cos\theta_r \end{bmatrix}, \tag{A.1e}$$

which produces some unsolved, non-linear terms, of which the solutions constitute the remainder of this derivation. Vectors with no subscript are implicitly assumed to be stationary ($\alpha\beta$) frame variables.

- Solution of (A.1c):

  The *chain rule* re-arranged:

$$g(t) \cdot \frac{d}{dt}\left[f(t)\right] = \frac{d}{dt}\left[f(t) \cdot g(t)\right] - \frac{d}{dt}\left[g(t)\right] \cdot f(t) \tag{A.2}$$

Then applying the chain rule in (A.2) to (A.1c) we obtain:

$$K_{xy}\frac{d\vec{v}_c}{dt} = \frac{d}{dt}\left[K_{xy}\vec{v}_c\right] - \frac{d}{dt}\left[K_{xy}\right]\vec{v}_c \tag{A.3}$$

$$= \frac{d\vec{v}_{c,xy}}{dt} - \frac{d}{dt}\left[K_{xy}\right]\vec{v}_c, \tag{A.4}$$

and

$$\frac{d}{dt}\left[K_{xy}\right] = \frac{d\gamma_s}{dt}\begin{bmatrix} -\sin\gamma_s & \cos\gamma_s \\ -\cos\gamma_s & -\sin\gamma_s \end{bmatrix}, \tag{A.5}$$

with

$$\gamma_s = \theta_r + \delta, \tag{A.6a}$$

$$\frac{d\gamma_s}{dt} = \frac{d\theta_r}{dt} + \frac{d\delta}{dt}. \tag{A.6b}$$

By combining (A.5) and (A.6), we obtain

$$\frac{d}{dt}\left[K_{xy}\right]\vec{v}_c = \left(\omega_r + \frac{d\delta}{dt}\right)\begin{bmatrix} -\sin\gamma_s & \cos\gamma_s \\ -\cos\gamma_s & -\sin\gamma_s \end{bmatrix}\begin{bmatrix} \vec{v}_{c,\alpha} \\ \vec{v}_{c,\beta} \end{bmatrix} \tag{A.7}$$

$$= \omega_\Phi \begin{bmatrix} \vec{v}_{c,y} \\ -\vec{v}_{c,x} \end{bmatrix}, \tag{A.8}$$

with let $\omega_\Phi = \omega_r + \dfrac{d\gamma_s}{dt}$.

Substituting (A.4) and (A.8) into (A.1c), we obtain

$$\vec{i}_{c,xy} = C_f\frac{d\vec{v}_{c,xy}}{dt} + C_f\omega_\Phi\begin{bmatrix} -\vec{v}_{c,y} \\ +\vec{v}_{c,x} \end{bmatrix}. \tag{A.9}$$

- Solution of (A.1a):

  By applying the chain rule, and following the same reasoning, we obtain

$$\vec{v}_{i,xy} = R_f\vec{i}_{f,xy} + L_f\frac{d\vec{i}_{f,xy}}{dt} + L_f\omega_\Phi\begin{bmatrix} -\vec{i}_{fy} \\ \vec{i}_{fx} \end{bmatrix} + \vec{v}_{s,xy}. \tag{A.10}$$

- Solution of (A.1d):

  Once again by applying the chain rule, and following the same reasoning, we obtain

$$\vec{v}_{s,xy} = R_s \vec{i}_{s,xy} + \frac{d\vec{\Phi}_{s,xy}}{dt} + \omega_\Phi \begin{bmatrix} 0 \\ \Phi_{sx} \end{bmatrix}. \tag{A.11}$$

- Solution of (A.1e):

  Finally, by evaluating and simplifying (A.1e), we obtain

$$\vec{\Phi}_{s,xy} = L_s \vec{i}_{s,xy} + \Phi_f \begin{bmatrix} \cos\delta \\ -\sin\delta \end{bmatrix}. \tag{A.12}$$

This concludes the derivation of the xy-frame model of the filter-generator.

Appendix B

Derivation of Approximate Parallel Structure DTC-SVM Transfer Functions

The derivation of the approximate transfer functions of the flux linkage and torque plants of the combined filter-generator model is presented in detail in this chapter.

B.1 Flux Linkage

The objective here is to obtain

$$G_{\Phi}(s) = \frac{\Phi_{fx}(s)}{V_{ix}(s)}. \tag{B.1}$$

First let us consider the expanded expression for the filter-generator flux linkage from (2.7e) and (2.8a):

$$\Phi_{fx} = L_f i_{fx} + L_s i_{sx} + \Phi_m, \tag{B.2}$$

and we know from basic circuit theory that

$$i_{sx} = i_{fx} - i_{cx}, \tag{B.3a}$$

$$i_{cx} = C_f \frac{dv_{sx}}{dt}. \tag{B.3b}$$

Note that the cross-coupling term in (B.3b) has been omitted for simplicity, as the goal here is not exact transfer functions. The inclusion of the cross-coupling term would make it implausible or maybe even impossible to derive de-coupled equivalent transfer functions.

Substituting (B.3a) and (B.3b) into (B.2), taking the derivative, and simplifying, we obtain:

$$\frac{d\Phi_{fx}}{dt} = (L_f + L_s)\frac{di_{fx}}{dt} - L_s C_f \frac{d^2 v_{cx}}{dt^2}. \tag{B.4}$$

Taking the Laplace transformation of (B.4), it becomes

$$s\Phi_{fx} = sL_s I_{fx} - s^2 L_s C_f V_{sx}. \tag{B.5}$$

Substituting (B.3b) into (2.7b) and taking the Laplace transform, we obtain:

$$V_{sx} = V_{cx}\left(1 + R_d C_f s\right). \tag{B.6}$$

Substituting (B.6) into (B.5), it becomes

$$s\Phi_{fx} = sL_s I_{fx} - \frac{s^2 L_s C_f}{1 + R_d C_f s}. \tag{B.7}$$

Taking the Laplace transform of (2.7d), omitting the cross-coupling terms and re-arranging, we obtain

$$I_{fx} = \frac{1}{r_f}\left(V_{ix} - s\Phi_{fx}\right), \tag{B.8}$$

By applying simple circuit theory, and omitting cross-coupling terms, we obtain

$$V_{sx} = V_{ix} - r_f I_{fx} - sL_f I_{fx}. \tag{B.9}$$

Let (B.9) and (B.8) are substituted into (B.7); simplifying, we obtain

$$G_\Phi(s) = \frac{\Phi_{fx}(s)}{V_{ix}(s)} = \frac{\dfrac{1}{L_f C_f}\left(L_f C_f s^2 + R_d C_f s + 1\right)}{s^3 + \dfrac{R_d}{L_f}s^2 + \dfrac{1}{L_f C_f}s + \dfrac{r_f}{L_f C_f L_s}}, \tag{B.10}$$

where it has been assumed in the derivation that $\frac{L_s}{r_f} >> R_d C_f$. This concludes the solution of the objective in (B.1).

B.2 Torque

The objective here is to obtain

$$G_m(s) = \frac{m_f(s)}{V_{iy}(s)}. \tag{B.11}$$

Taking the derivative of (2.7b), Laplace transforming, and re-arranging, we obtain

$$I_{cy} = \frac{sC_f}{1 + R_d C_f s} V_{sy}. \tag{B.12}$$

Taking the Laplace transform of the y-components of (B.3a) and (B.3b), and omitting the cross-coupling terms, we obtain

$$I_{sy} = I_{fy} - \frac{C_f s}{1 + R_d C_f s} V_{sy}. \tag{B.13}$$

Taking the Laplace transform of the y-component of (2.7a), omitting the coupling terms and re-arranging, we obtain

$$V_{sy} = V_{iy} - r_f I_{fy} - s L_f I_{fy}, \tag{B.14}$$

and then substituting (B.14) into (B.13), it becomes

$$I_{sy} = I_{fy} - \frac{C_f s}{1 + R_d C_f s} \left(V_{iy} - I_{fy} r_f - s L_f I_{fy} \right). \tag{B.15}$$

Substituting (2.7e) into the y-component of (2.7d), and taking the Laplace transform, it becomes

$$V_{iy} = r_f I_{fy} + s L_s I_{sy} + \Phi_m \omega_r. \tag{B.16}$$

where it is assumed that that $\delta \simeq 0$, which means that $\omega_\Phi \simeq \omega_r$. It is also assumed that the stator flux linkage is constant and $\Phi_{sx} \simeq \Phi_m$.

Now taking the Laplace transform of (2.1a), with the previous assumption that $\delta \simeq 0$, i.e. the load is zero, and assuming that $m_f \simeq m_e$, it becomes

$$s\omega_r = \frac{N_p}{J} m_f. \tag{B.17}$$

We know that

$$m_e = \frac{2N_p}{3} i_{fy} \Phi_{sx}, \tag{B.18}$$

and assuming $\Phi_{sx} \simeq \Phi_m$, then taking the Laplace transform and re-arranging, (B.18) becomes

$$I_{fy} = \frac{2}{3 N_p \Phi_m} m_f. \tag{B.19}$$

Let

$$G_c(s) = \frac{C_f s}{1 + R_d C_f s}, \tag{B.20}$$

then substituting (B.15) into (B.16), it becomes

$$V_{iy} = r_f I_{fy} + s L_d \left(I_{fy} - G_c(s) \cdot V_{iy} + r_f G_c(s) I_{fy} + s L_f G_c(s) \cdot I_{fy} \right) + \Phi_m \omega_r. \tag{B.21}$$

Substituting (B.18) and (B.17) into (B.21), we finally obtain

$$G_m(s) = \frac{m_f(s)}{V_{iy}(s)} = \frac{\frac{3\Phi_m N_p}{2L_s} s \left(L_s C_f s^2 + R_d C_f s + 1 \right)}{L_f C_f s^4 + C_f \left(r_f + R_d \right) s^3 + s^2 + \frac{r_f}{L_s} s + \frac{N_p \Phi_m}{J}}, \tag{B.22}$$

assuming that $1 >> \frac{r_f R_d C_f}{L_s}$. This completes the solution of the objective in (B.11).

Appendix C

Simulink Model Diagrams

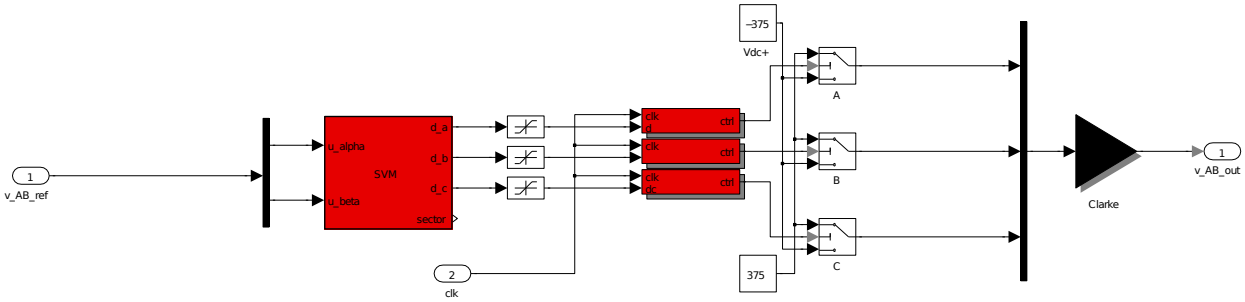


Figure C.1: *Top-level inverter/switching model of the Simulink simulation.*

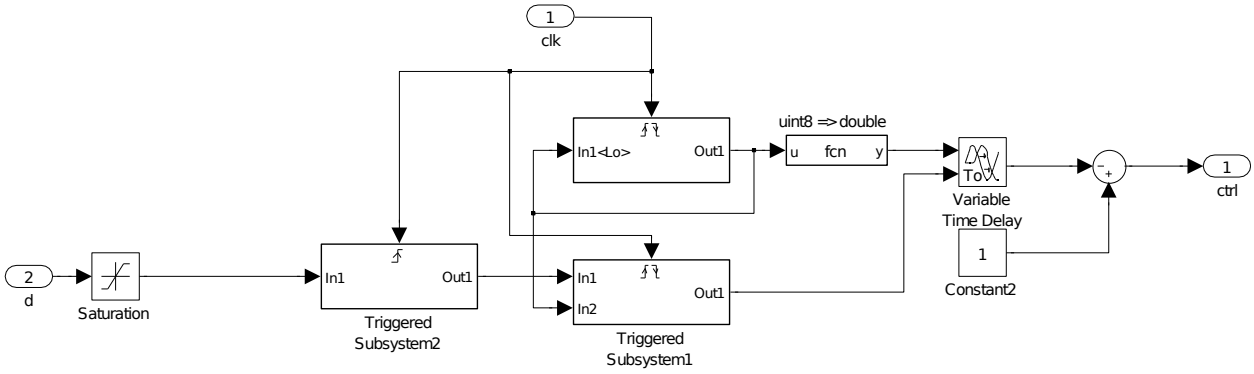


Figure C.2: *Top-level DEPWM pulse generation model of the Simulink simulation.*

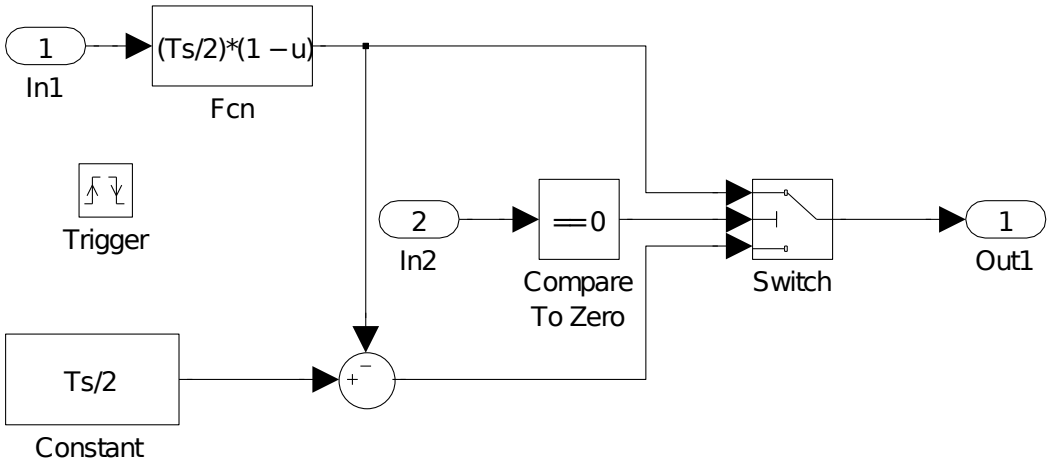


Figure C.3: *A component of the DEPWM pulse generation model of the Simulink simulation (First triggered block in Fig. C.2, clockwise from the top).*

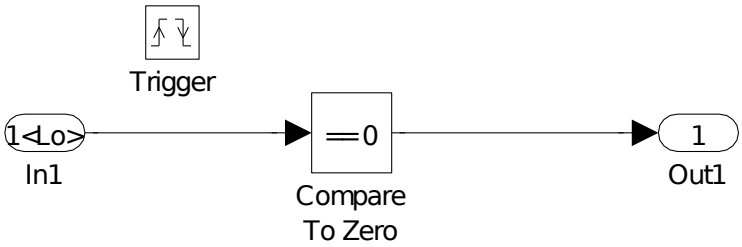


Figure C.4: *A component of the DEPWM pulse generation model of the Simulink simulation (Second triggered block in Fig. C.2, clockwise from the top).*

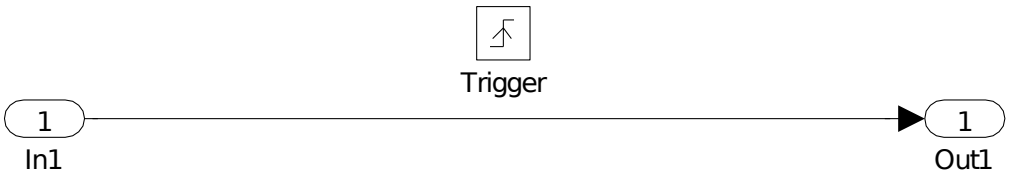


Figure C.5: *A component of the DEPWM pulse generation model of the Simulink simulation (Third triggered block in Fig. C.2, clockwise from the top).*

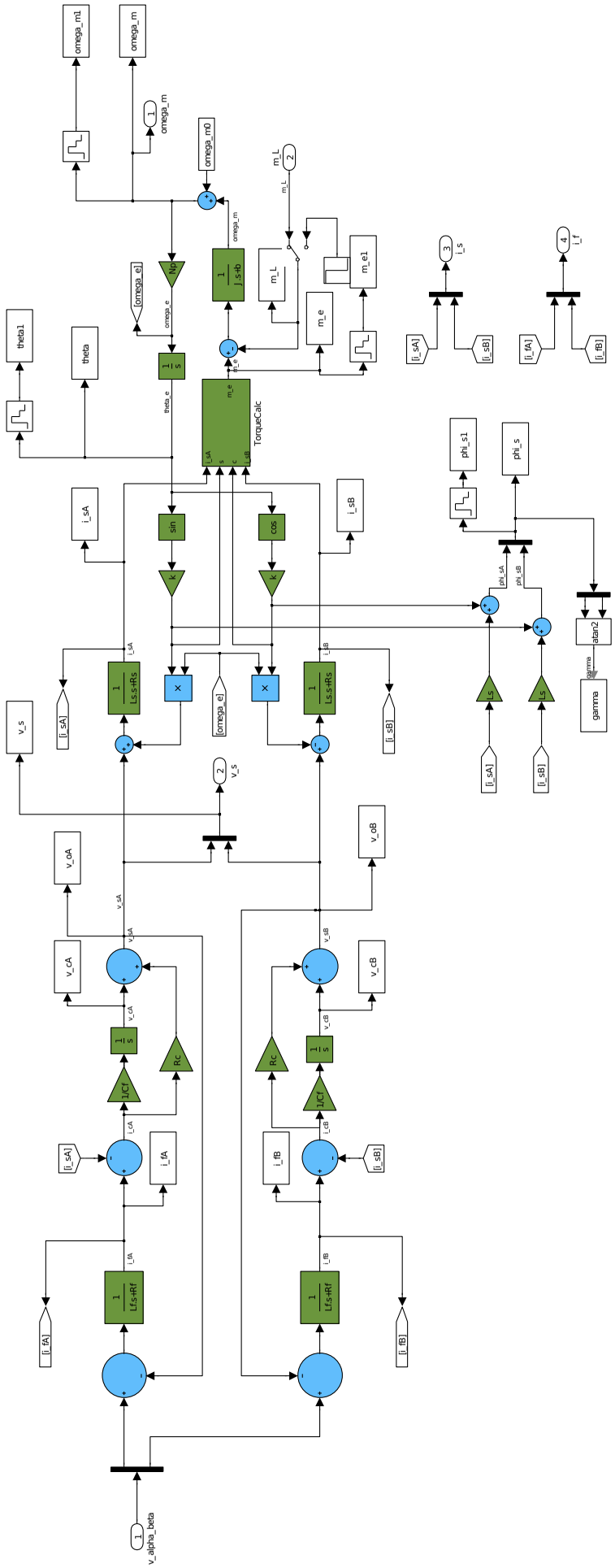


Figure C.6: *Filter-generator model of the Simulink simulation.*

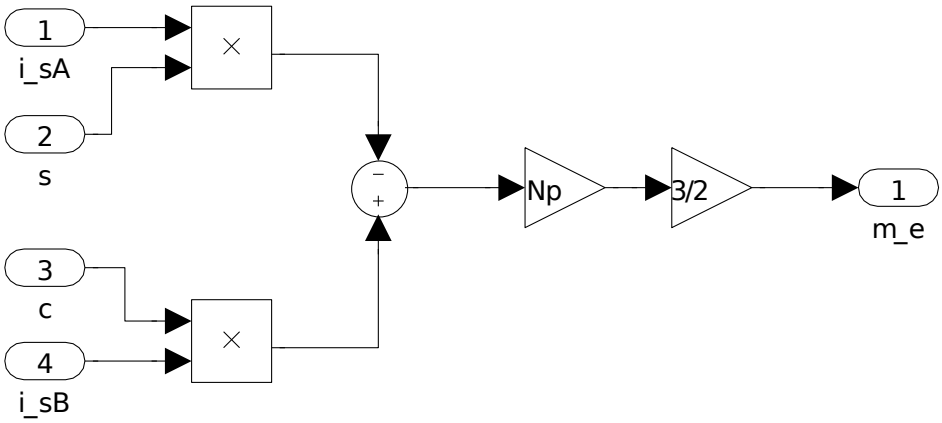


Figure C.7: *EM torque calculation of the filter-generator model in Fig. C.6.*

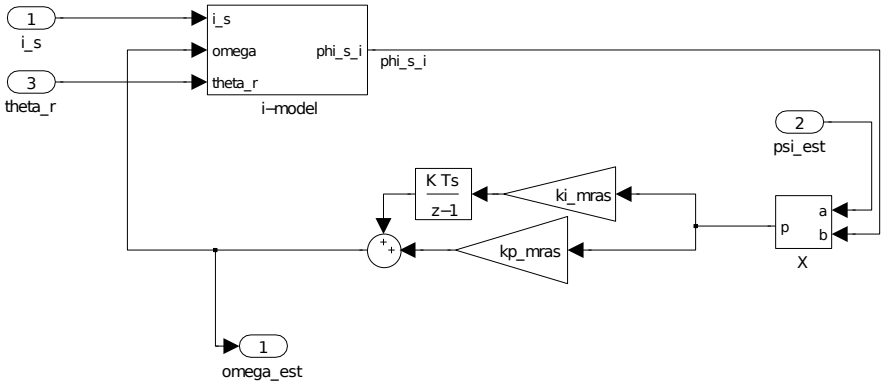


Figure C.8: *Top-level MRAS speed observer model of the Simulink simulation.*

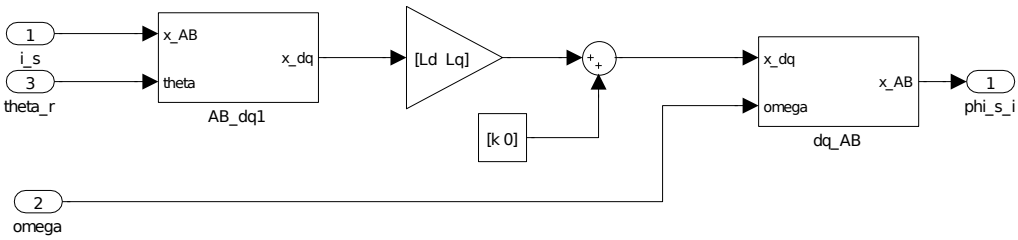


Figure C.9: *Adaptive flux linkage current model of the MRAS speed observer.*

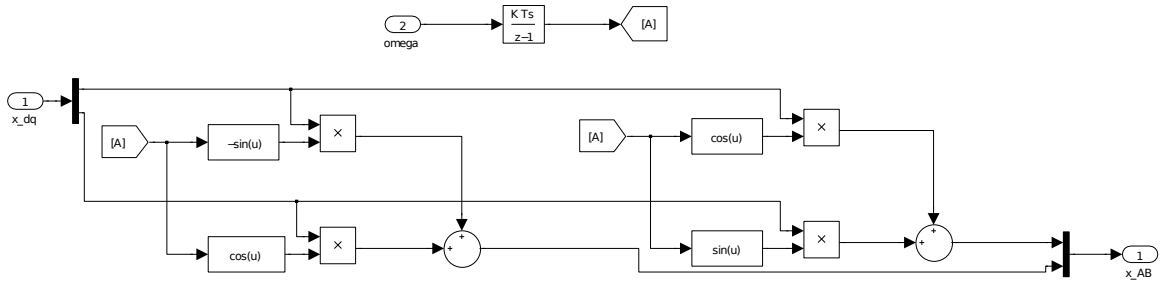


Figure C.10: *Modified dq to $\alpha\beta$ transformation of the adaptive flux linkage current model.*

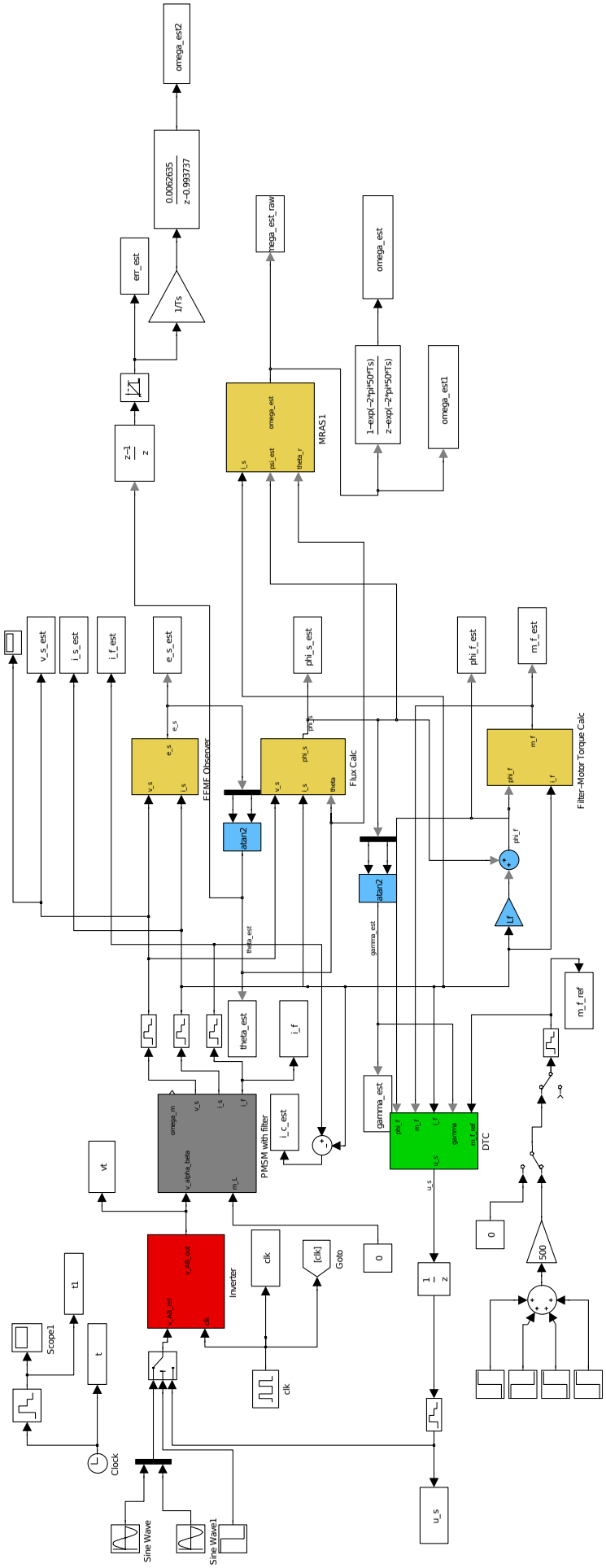


Figure C.11: *Top-level model of the Simulink simulation.*

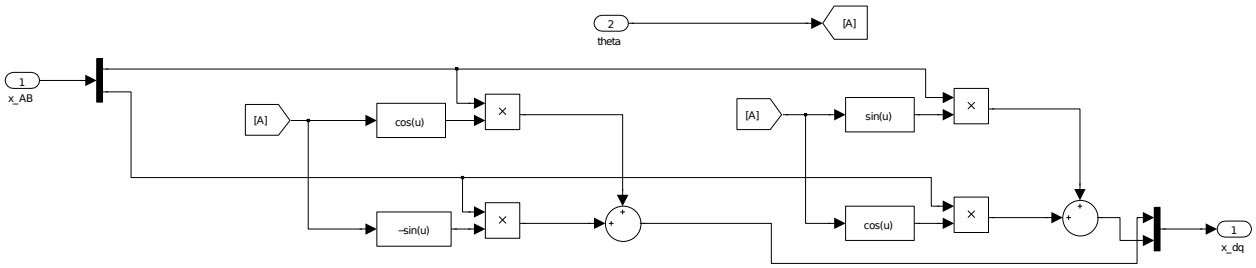


Figure C.12: *$\alpha\beta$ to dq transformation of the Simulink simulation.*

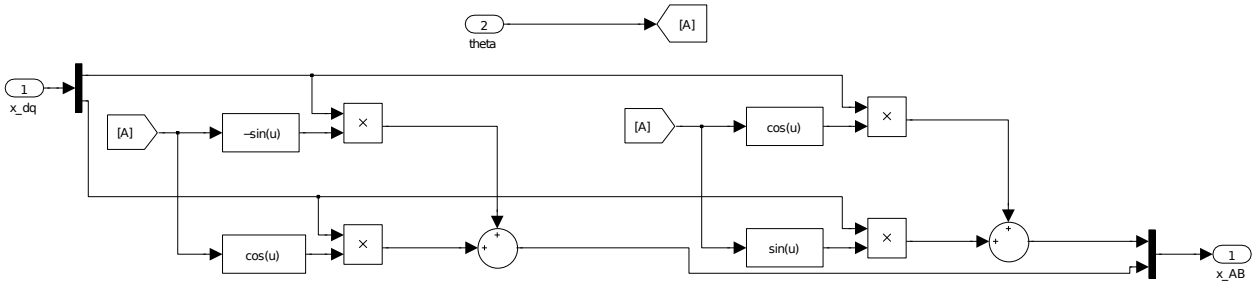


Figure C.13: *dq to $\alpha\beta$ transformation of the Simulink simulation.*

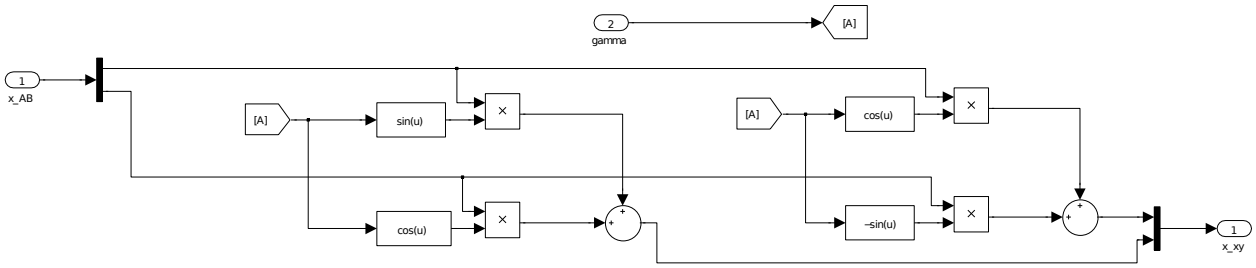


Figure C.14: *$\alpha\beta$ to xy transformation of the Simulink simulation.*

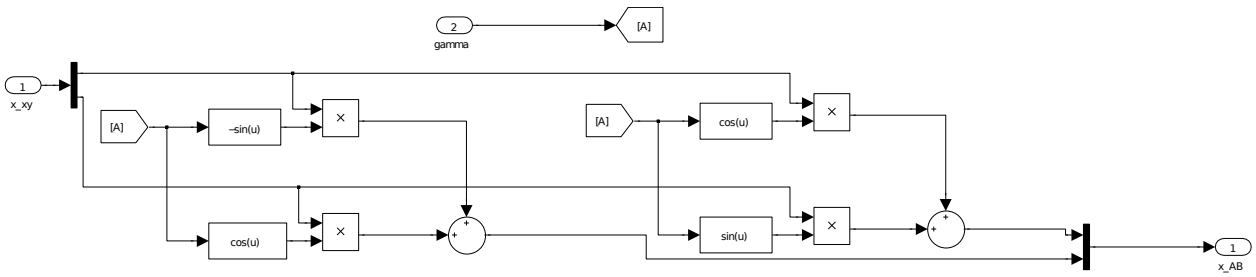


Figure C.15: *xy to $\alpha\beta$ transformation of the Simulink simulation.*

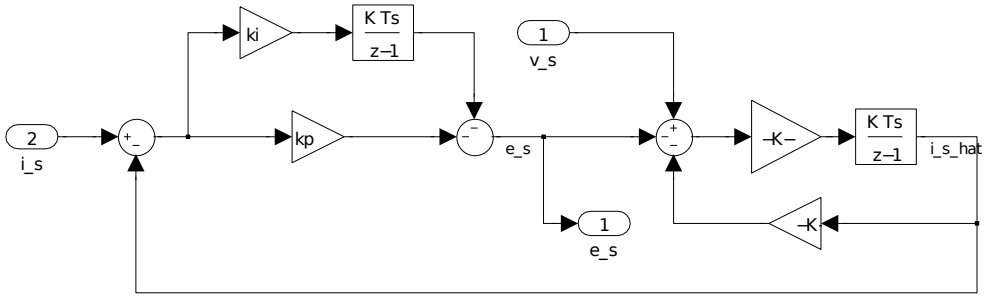


Figure C.16: *EEMF observer model of the Simulink simulation.*

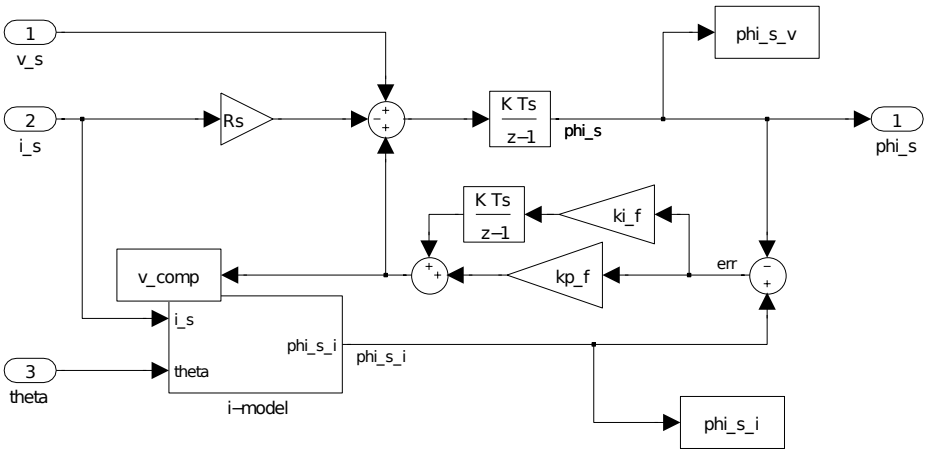


Figure C.17: *Stator flux linkage observer model of the Simulink simulation.*

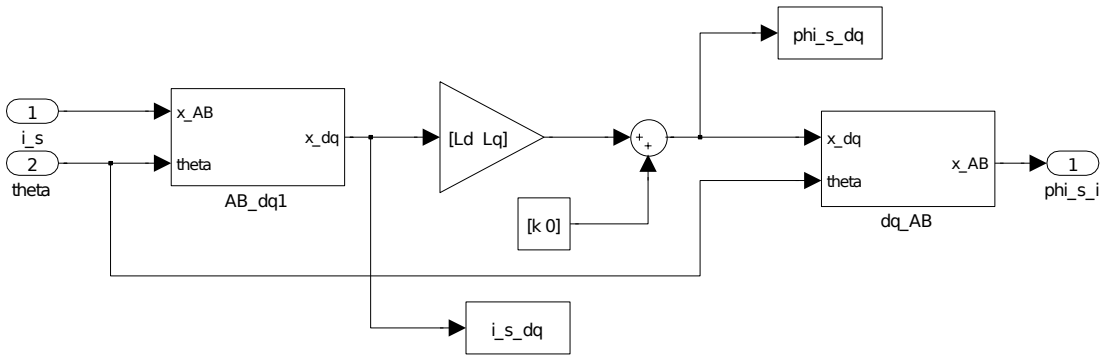


Figure C.18: *Current model of the stator flux linkage observer model.*

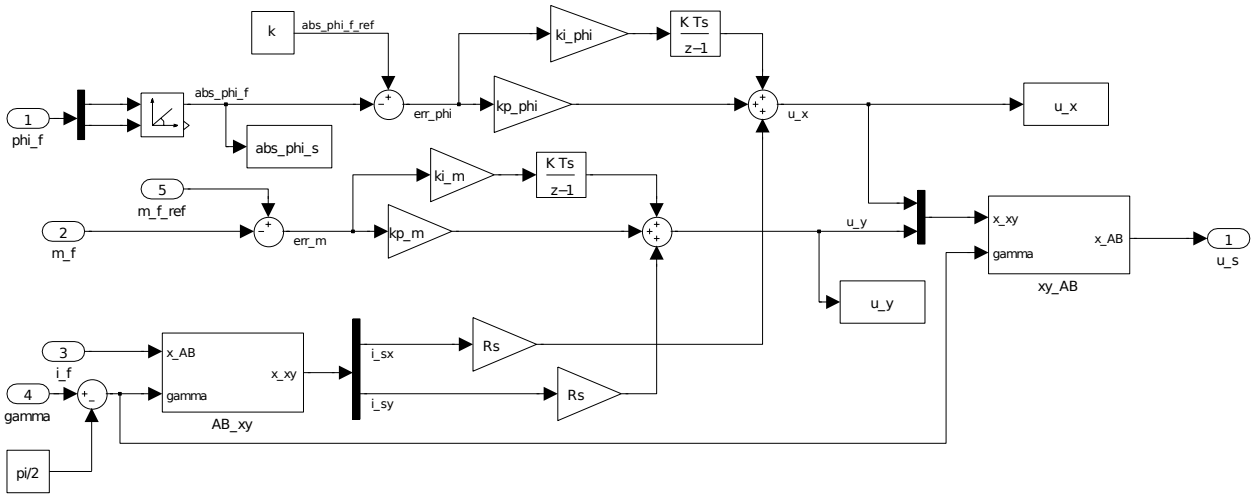


Figure C.19: *DTC model of the Simulink simulation.*

Appendix D

VHDL-AMS Simulation Code

```
-- Circuit:
  --LABEL: ENTITY:    ARCHITECTURE: PARAMETERS:      PORTS:
     -- DC Voltage source and bus capacitors
  dcp:   entity dc_source  (default)  generic map ( V=>Vdc/2.0 )   port map (  n1, electrical_ref     );
  dcm:   entity dc_source  (default)  generic map ( V=>Vdc/2.0 )   port map (  electrical_ref, n2     );
  -- Inverter IGBT bridge:
  inverter:  entity inverter_bridge  (default)        port map ( n1, n2, a0, b0, c0, a_top_ctrl, a_bottom_ctrl,
                                                                                    b_top_ctrl, b_bottom_ctrl,
                                                                                    c_top_ctrl, c_bottom_ctrl  );
  -- RLC Filter:
  Lfa:  entity inductor   (default) generic map ( L=>Lf, r0=>Rf ) port map ( p=>a0, m=>a1, vm=>vL_a, im=>if_a );
  Lfb:  entity inductor   (default) generic map ( L=>Lf, r0=>Rf ) port map ( p=>b0, m=>b1, vm=>vL_b, im=>if_b );
  Lfc:  entity inductor   (default) generic map ( L=>Lf, r0=>Rf ) port map ( p=>c0, m=>c1, vm=>vL_c, im=>if_c );
  Cfa:  entity capacitor  (default) generic map ( C=>Cf )   port map ( p=>a1, m=>a2, im=>ic_a, vm=>vc_a );
  Cfb:  entity capacitor  (default) generic map ( C=>Cf )   port map ( p=>b1, m=>b2, im=>ic_b, vm=>vc_b );
  Cfc:  entity capacitor  (default) generic map ( C=>Cf )   port map ( p=>c1, m=>c2, im=>ic_c, vm=>vc_c );
  Rca:  entity resistor   (default) generic map ( R=>rc )   port map ( p=>a2, m=>nc      );
  Rcb:  entity resistor   (default) generic map ( R=>rc )   port map ( p=>b2, m=>nc      );
  Rcc:  entity resistor   (default) generic map ( R=>rc )   port map ( p=>c2, m=>nc      );
  -- Breaker:
  Synca:  entity varistor   (default)         port map ( p=>a1, m=>a3, r=>Rsync    );
  Syncb:  entity varistor   (default)         port map ( p=>b1, m=>b3, r=>Rsync    );
  Syncc:  entity varistor   (default)         port map ( p=>c1, m=>c3, r=>Rsync    );
  -- Generator:
  Lsa:  entity inductor   (default) generic map ( L=>Ls )   port map ( p=>a3, m=>a4      );
  Lsb:  entity inductor   (default) generic map ( L=>Ls )   port map ( p=>b3, m=>b4      );
  Lsc:  entity inductor   (default) generic map ( L=>Ls )   port map ( p=>c3, m=>c4      );
  Rsa:  entity resistor   (default) generic map ( R=>Rs )   port map ( p=>a4, m=>a5      );
  Rsb:  entity resistor   (default) generic map ( R=>Rs )   port map ( p=>b4, m=>b5      );
  Rsc:  entity resistor   (default) generic map ( R=>Rs )   port map ( p=>c4, m=>c5      );
  -- PMSG EMF model:
  es_a  == -omega_r*psi_PM*sin(theta_r);
  es_b  == -omega_r*psi_PM*sin(theta_r - math_2_pi/3.0);
  es_c  == -omega_r*psi_PM*sin(theta_r + math_2_pi/3.0);
  es_alphabeta == abc_alphabeta( es_a, es_b, es_c );
  -- Mechanical model:
  if domain = quiescent_domain use
     omega_r == omega0;
  else
     (J/p)*omega_r'dot + (b/p)*omega_r == m_e - m_L;
  end use;
  theta_r == omega_r'integ;
  p_e  == is_a*es_a + is_b*es_b + is_c*es_c;
  m_e*(omega_r/p)       == is_a*es_a + is_b*es_b + is_c*es_c;
  omega_m == omega_r/p;
```

Figure D.1: *VHDL-AMS implementation of the RLC filter and PMSG circuit model.*

```
-- EEMF OBSERVER:
        v    := v_t;
        i     := i_s;

        -- Part A:
        i_tilde   := i - i_hat_prev;
        y     := y_prev + (Ts/2.0)*( i_tilde + i_tilde_prev );
        e     := (-1.0)*( K_i*y + K_p*i_tilde );

        e_s    := e;
        theta_r_est  := arctan( e(1), e(2) );

        -- Part B:
        x     := v - e - Rs*i_hat_prev;
        y_hat    := y_hat_prev + (Ts/2.0)*( x + x_prev );

        -- Delayed values:
        i_hat_prev   := ( (1.0/(Ls))*y_hat(1), (1.0/(Ls))*y_hat(2) );
        i_tilde_prev  := i_tilde;
        y_prev    := y;
        y_hat_prev   := y_hat;
        x_prev    := x;
```

Figure D.2: *VHDL-AMS implementation of the digital EEMF observer.*

```
-- FLUX OBSERVER
            --Co-ordinate transformation of stator current:
        i_s_dq   := alphabeta_dq( i_s, theta_r_est );

            --Current model of stator flux linkage:
        phi_s_dq  := ( i_s_dq(1)*Ld + phi_PM, i_s_dq(2)*Lq );

            --Transform stator flux to stationary reference frame:
        phi_s_i  := dq_alphabeta( phi_s_dq, theta_r_est );

            --Error signal for compensator:
        err_phi  := phi_s_i - phi_s;

            --Compensation voltage
        v_comp   := v_comp + ( kp+ki*(Ts/2.0) )*err_phi + ( ki*(Ts/2.0)-kp )*err_phi_prev;

            --Voltage model (Integrator)
        x_phi  := v_t - Rs*i_s + v_comp;
        phi_s  := phi_s + (Ts/2.0)*(x_phi + x_phi_prev);

            --Stator Flux Angle
        gamma_s  := arctan( phi_s(1), phi_s(2) );

            --Inverter/Filter Flux (Assuming that is_A is close to if_A)
        phi_f  := Lf*i_f + phi_s;

            -- Estimate machine electromagnetic torque
        m_e_est  := (3.0*p/2.0)*( phi_s(1)*i_s(2) - phi_s(2)*i_s(1) );

            -- Estimate inverter torque
        m_i   := (3.0*p/2.0)*( phi_i(1)*i_f(2) - phi_i(2)*i_f(1) );

        --Previous values
        err_phi_prev := err_phi;
        x_phi_prev  := x_phi;
```

Figure D.3: *VHDL-AMS implementation of the digital flux linkage observer.*

```
-- Speed estimation:
        -- Unwrap angle:
        if theta_r_est > theta_r_est_prev + (math_pi/2.0) then -- Direction dependent
            omega_r_est   := 0.995*omega_r_est - 0.005*(1.0/Ts)*(theta_r_est - (theta_r_est_prev + math_2_pi) );
        else
            omega_r_est   := 0.995*omega_r_est - 0.005*(1.0/Ts)*(theta_r_est - theta_r_est_prev);
        end if;

        -- Digital filter:
        ome_r1     := 0.995*ome_r1 + 0.005*omega_r_est;
```

Figure D.4: *VHDL-AMS implementation of the digital derivative-filter speed estimation.*

```
-- Speed Controller
        err_omega   := omega_ref - ome_r1;
            --Calculate next torque command:
        m_e_ref_spd   := m_e_ref_spd + (kp_spd + (Ts/2.0)*ki_spd)*err_omega + ((Ts/2.0)*ki_spd - kp_spd)*err_omega_prev;
            --Delayed values:
        err_omega_prev  := err_omega;
```

Figure D.5: *VHDL-AMS implementation of the digital speed controller.*

```
-- DTC:
            --Torque reference:
        if now >= 0 ms and now < 10 ms then
        -- m_e_ref   := m_e_ref_spd;
        -- m_e_ref   := -( 0.0185*ome_r1*ome_r1 - 1.575*ome_r1 );
            m_e_ref   := 0.0;
        elsif now >= 10 ms and now < 160 ms then
            m_e_ref   := 500.0;
        elsif now >= 160 ms and now < 310 ms then
            m_e_ref   := -500.0;
        else
            m_e_ref   := 0.0;
        end if;

        abs_psi_s   := abs(psi_s);
        abs_psi_i   := abs(psi_i);
            --Co-ordinate transformation of currents:
        i_xy     := alphabeta_dq( i_f, gamma_s - math_pi/2.0 );

        err_psi_dtc   := psi_ref - abs_psi_i;
        err_m_dtc   := m_e_ref - m_i;
            --Controller integration:
        x_psi_dtc   := x_psi_dtc_prev + (Ts/2.0)*(err_psi_dtc + err_psi_dtc_prev);
        x_m_dtc    := x_m_dtc_prev + (Ts/2.0)*(err_m_dtc + err_m_dtc_prev);
            --Control voltage generation:
        u_xy(1)    := kp_psi*err_psi_dtc + ki_psi*x_psi_dtc + Rf*i_xy(1);
        u_xy(2)    := kp_m*err_m_dtc + ki_m*x_m_dtc + Rf*i_xy(2);
            --Transform voltages to stationary frame for SVM:
        u_s     := dq_alphabeta( u_xy, gamma_s - math_pi/2.0);
            --Delayed values
        x_psi_dtc_prev  := x_psi_dtc;
        err_psi_dtc_prev  := err_psi_dtc;
        x_m_dtc_prev  := x_m_dtc;
        err_m_dtc_prev   := err_m_dtc;
        m_i_prev   := m_i;
        theta_r_est_prev := theta_r_est;
        p_e_est_prev  := p_e_est;
        omega_r_est_prev := omega_r_est;
```

Figure D.6: *VHDL-AMS implementation of digital DTC.*

```
-- Switch Control
        -- Reference voltage
        if now < 0.0e-3 then
            v_ref   := (3.0/2.0)*( -Vm*sin(omega0*now), Vm*cos(omega0*now) );
            u_s_prev := u_s;
        else
            --v_ref  := u_s;
            v_ref  := u_s_prev; -- One sample delay to simulate PWM voltage generation
            u_s_prev := u_s;
        end if;

        -- Space Vector Modulation (Generation of DEPWM duty cycles)
        dc      := svm( v_ref, Vdc );

        -- Limit duty cycles (generated by SVM)
        if dc(1) > 0.99 then
            dc(1) := 0.99;
        elsif dc(1) < 0.01 then
            dc(1) := 0.01;
        end if;

        if dc(2) > 0.99 then
            dc(2) := 0.99;
        elsif dc(2) < 0.01 then
            dc(2) := 0.01;
        end if;

        if dc(3) > 0.99 then
            dc(3) := 0.99;
        elsif dc(3) < 0.01 then
            dc(3) := 0.01;
        end if;

        -- DEPWM:
            -- Calculate delay times
        t_delta_a    := Ts*(1.0-dc(1))*unit_time/2.0;
        t_hi_a     := dc(1)*Ts*unit_time;

        t_delta_b    := Ts*(1.0-dc(2))*unit_time/2.0;
        t_hi_b     := dc(2)*Ts*unit_time;

        t_delta_c    := Ts*(1.0-dc(3))*unit_time/2.0;
        t_hi_c     := dc(3)*Ts*unit_time;

            -- Switch ( With dead time )
        a_top_ctrl    <= '0', '1' after t_delta_a+Tdead, '0' after t_delta_a+t_hi_a;
            a_bottom_ctrl  <= '1', '0' after t_delta_a, '1' after t_delta_a+t_hi_a+Tdead;

        b_top_ctrl   <= '0', '1' after t_delta_b+Tdead, '0' after t_delta_b+t_hi_b;
            b_bottom_ctrl  <= '1', '0' after t_delta_b, '1' after t_delta_b+t_hi_b+Tdead;

        c_top_ctrl    <= '0', '1' after t_delta_c+Tdead, '0' after t_delta_c+t_hi_c;
            c_bottom_ctrl  <= '1', '0' after t_delta_c, '1' after t_delta_c+t_hi_c+Tdead;
```

Figure D.7: *VHDL-AMS implementation of SVM and DEPWM switching signal generation.*

```
--IGBT inverter bridge (two-level inverter bridge)
library ieee;
use ieee.electrical_systems.all;

entity inverter_bridge is
    port ( terminal p, m : electrical;
           terminal a, b, c : electrical;
           signal ctrl_a_top, ctrl_a_bottom, ctrl_b_top, ctrl_b_bottom, ctrl_c_top, ctrl_c_bottom : in bit );
end;

architecture default of inverter_bridge is
begin
    sa1: entity igbt (behav) port map ( p, a, ctrl_a_top );
    da1: entity d (behav) port map ( a, p );
    sa2: entity igbt (behav) port map ( a, m, ctrl_a_bottom );
    da2: entity d (behav) port map ( m, a );

    sb1: entity igbt (behav) port map ( p, b, ctrl_b_top );
    db1: entity d (behav) port map ( b, p );
    sb2: entity igbt (behav) port map ( b, m, ctrl_b_bottom );
    db2: entity d (behav) port map ( m, b );

    sc1: entity igbt (behav) port map ( p, c, ctrl_c_top );
    dc1: entity d (behav) port map ( c, p );
    sc2: entity igbt (behav) port map ( c, m, ctrl_c_bottom );
    dc2: entity d (behav) port map ( m, c );
end;
```

Figure D.8: *VHDL-AMS implementation of the two-level IGBT converter circuit model.*

```
-- ------------------------------------------------------------------------
-- Copyright (c) 2004 by ANSOFT Corp. All rights reserved. --
-- ------------------------------------------------------------------------

LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
USE IEEE.MATH_REAL.ALL;

ENTITY d IS
    GENERIC (
        ISAT : CURRENT := 1.0e-12; -- Saturation current
        VT : VOLTAGE := 35.0e-3; -- Thermal voltage
        RR : RESISTANCE := 100.0e3; -- Reverse resistance
        VF : VOLTAGE := 0.8; -- Forward voltage
        RB : RESISTANCE := 1.0e-3 -- Bulk Resistance
);
    PORT (TERMINAL p,m : ELECTRICAL);
END ENTITY d;

ARCHITECTURE behav OF d IS
    QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
    IF (v >= 0.0) USE
        i == ISAT * ((exp(v/VT)) - 1.0);
    ELSE
        i == v/RR;
    END USE;
END ARCHITECTURE behav;
```

Figure D.9: *VHDL-AMS implementation of the diode model.*

```
-- ------------------------------------------------------------------------
-- Copyright (c) 2004 by ANSOFT Corp. All rights reserved. --
-- ------------------------------------------------------------------------

LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
USE IEEE.MATH_REAL.ALL;

ENTITY IGBT IS
    GENERIC (
        ISAT : CURRENT := 1.0e-12; -- Saturation current
        VT : VOLTAGE := 35.0e-3; -- Thermal voltage
        RR : RESISTANCE := 100.0e3; -- Reverse resistance
        VF : VOLTAGE := 0.8; -- Forward voltage
        RB : RESISTANCE := 1.0e-3 -- Bulk Resistance
);
    PORT (TERMINAL C,E : ELECTRICAL;
          CTRL : BIT := '0'); -- Control Signal;
END ENTITY IGBT;

ARCHITECTURE behav OF IGBT IS
    QUANTITY v ACROSS i THROUGH C TO E;
    SIGNAL sw_on : BOOLEAN := FALSE;
BEGIN
    --sw_on <= CTRL'ABOVE(0.0);
    IF (v >= 0.0) AND (ctrl = '1') USE
        i == ISAT * ((exp(v/VT)) - 1.0);
    ELSE
        i == v/RR;
    END USE;
END ARCHITECTURE behav;
```

Figure D.10: *VHDL-AMS implementation of the IGBT model.*

```
--Clock
library ieee;
use ieee.math_real.all;

entity clock is
    generic ( f0 : real := 10.0e3 );
    port ( clk : out bit := '0' );
end;

architecture default of clock is
    constant unit_time : time := 1.0 sec;
    constant T0 : real := (1.0/f0);
    signal clk_s : bit := '0';
begin
    process (clk_s) is
    begin
        if clk_s'last_value = '0' then
            clk_s <= '1', '0' after 0.5*T0*unit_time;
        elsif clk_s'last_value = '1' then
            clk_s <= '0', '1' after 0.5*T0*unit_time;
        end if;
    end process;
clk <= clk_s;
end default;
```

Figure D.11: *VHDL-AMS implementation of the digital clock.*

```
1  library ieee;
2  use ieee.math_real.all;
3  use work.header.all;
4
5  package transformations is
6      function abc_alphabeta( xa, xb, xc : real ) return real_vector;
7      function alphabeta_abc( xs : real_vector ) return real_vector;
8      function alphabeta_dq( xs : space_vector ; theta : real ) return space_vector;
9      function dq_alphabeta( xdq : space_vector ; theta : real ) return space_vector;
10 end transformations;
11
12 package body transformations is
13     function abc_alphabeta( xa, xb, xc : real ) return real_vector is
14         variable x : space_vector;
15     begin
16         x(1) := (2.0/3.0)*( xa - xb*(1.0/2.0) - xc*(1.0/2.0) );
17         x(2) := (2.0/3.0)*( xb*sqrt(3.0)/2.0 - xc*sqrt(3.0)/2.0 );
18         return x;
19     end function abc_alphabeta;
20     function alphabeta_abc( xs : real_vector ) return real_vector is
21         variable x : real_vector(1 to 3);
22     begin
23         x(1) := sqrt(2.0/3.0)*( xs(1) );
24         x(2) := sqrt(2.0/3.0)*( xs(1)*(-1.0/2.0) + xs(2)*sqrt(3.0)/2.0 );
25         x(3) := sqrt(2.0/3.0)*( xs(1)*(-1.0/2.0) - xs(2)*sqrt(3.0)/2.0 );
26         return x;
27     end;
28     function alphabeta_dq( xs : space_vector ; theta : real ) return space_vector is
29         variable x : space_vector := (0.0, 0.0);
30     begin
31         x(1) := xs(1)*cos(theta) - xs(2)*sin(theta);
32         x(2) := xs(1)*sin(theta) + xs(2)*cos(theta);
33         return x;
34     end;
35     function dq_alphabeta( xdq : space_vector ; theta : real ) return space_vector is
36         variable x : space_vector := (0.0, 0.0);
37     begin
38         x(1) := xdq(1)*cos(theta) + xdq(2)*sin(theta);
39         x(2) := -xdq(1)*sin(theta) + xdq(2)*cos(theta);
40         return x;
41     end;
42 end transformations;
```

Figure D.12: *VHDL-AMS implementation of the reference frame coordinate transformations.*

Appendix E

DSP Implementation Code

```
interrupt void GEN_isr(void){
    //run_time  = run_time + Ts;
    serviceADC(GEN);
    serviceProtection();
    CheckHStemp();
    // State Observers
    EEMFObserver();
    FluxObserver();
    SpeedEstimator();

        Vm_sync   = sqrt( vs_A*vs_A + vs_B*vs_B );
        Vm_sync_fil  = 0.999*Vm_sync_fil + 0.001*Vm_sync;

        if ( ipMode == MODE_ONLINE && ( ipState >= ONLINE_STATE02 ) && ( ipState <= ONLINE_STATE08 ) ) {
            FluxController();
            if ( ( ipState >= ONLINE_STATE03 ) && ( ipState <= ONLINE_STATE08 ) ) {
                SpeedController();
            }
            TorqueController();

            Vm_ref  = sqrt( u_A*u_A + u_B*u_B );

            rValpha  = (3.0/2.0)*u_A;
            rVbeta  = (3.0/2.0)*u_B;
        }
        SpaceVectorPWM();
        //m_e_fil  = 0.9*m_e_fil + 0.1*m_e;
        m_i_fil  = 0.9*m_i_fil + 0.1*m_i;
    }
```

Figure E.1: *Code of the generator interrupt service routine, executed at each sample instance.*

```
double kp_EEMF        = 20.0;
double ki_EEMF        = 1000.0;
double kp_FLUX        = 1000;
double ki_FLUX        = 5000

void EEMFObserver() {
       // Transform to stationary reference frame (Clarke)
   vs_A  = Space_3A(Vu,Vv,Vw);
   vs_B  = Space_3B(Vu,Vv,Vw);

   is_A  = Space_3A(Iu,Iv,Iw);
   is_B  = Space_3B(Iu,Iv,Iw);

   // EEMF Observer Compensator
   erri_A  = is_A - iso_A_prev;
   erri_B  = is_B - iso_B_prev;

   yc_A  = yc_A_prev + (Ts/2.0)*( erri_A + erri_A_prev);
   yc_B  = yc_B_prev + (Ts/2.0)*( erri_B + erri_B_prev);

   es_A  = -( kp_EEMF*erri_A + ki_EEMF*yc_A );
   es_B  = -( kp_EEMF*erri_B + ki_EEMF*yc_B );

   // Calculate Rotor Angle
   theta_r  = atan2( es_A, es_B );

   // Estimate rotor speed
   //omega_r  = sqrt( es_A*es_A + es_B*es_B )/phi_m;

   // Stator Current Model
   xo_A  = vs_A - es_A - Rs*iso_A_prev;
   xo_B  = vs_B - es_B - Rs*iso_B_prev;

   yo_A  = yo_A_prev + (Ts/2.0)*( xo_A + xo_A_prev );
   yo_B  = yo_B_prev + (Ts/2.0)*( xo_B + xo_B_prev );

   iso_A_prev = (1.0/Ls)*( yo_A );
   iso_B_prev = (1.0/Ls)*( yo_B );

   // Delayed Values
   erri_A_prev = erri_A;
   erri_B_prev = erri_B;
   yc_A_prev = yc_A;
   yc_B_prev = yc_B;
   xo_A_prev = xo_A;
   xo_B_prev = xo_B;
   yo_A_prev = yo_A;
   yo_B_prev = yo_B;

   // Voltage dq transformation
   vs_d  = SpaceABd( vs_A, vs_B, theta_r );
   vs_q  = SpaceABq( vs_A, vs_B, theta_r );
}
```

Figure E.2: *DSP implementation code of the EEMF observer.*

```
void FluxObserver() {
    // Current Model
        // Transform to rotor reference frame (Park)
    is_d  = SpaceABd(is_A, is_B, theta_r);
    is_q  = SpaceABq(is_A, is_B, theta_r);

    phis_d  = is_d*Ld + phi_m;
    phis_q  = is_q*Lq;

    phis_Ai  = SpacedqOA(phis_d, phis_q, theta_r);
    phis_Bi  = SpacedqOB(phis_d, phis_q, theta_r);
        // Transform back to stationary reference frame
    err_phi_A = phis_A - phis_Ai;
    err_phi_B = phis_B - phis_Bi;
    // Compensation voltage
    v_comp_A = v_comp_A + ( kp_FLUX+ki_FLUX*(Ts/2.0) )*err_phi_A + ( ki_FLUX*(Ts/2.0)-kp_FLUX )*err_phi_A_prev;
    v_comp_B = v_comp_B + ( kp_FLUX+ki_FLUX*(Ts/2.0) )*err_phi_B + ( ki_FLUX*(Ts/2.0)-kp_FLUX )*err_phi_B_prev;
    // Voltage model (Integrator)
    x_phi_A  = vs_A - Rs*is_A - v_comp_A;
    x_phi_B  = vs_B - Rs*is_B - v_comp_B;

    phis_A  = phis_A + (Ts/2)*(x_phi_A + x_phi_A_prev);
    phis_B  = phis_B + (Ts/2)*(x_phi_B + x_phi_B_prev);
    // Stator Flux Angle
    gamma_s  = atan2( phis_A, phis_B );
    // Inverter/Filter Flux (Assuming that is_A is close to if_A (may consider estimating if_A later...))
    phii_A  = is_A*Lf + phis_A;
    phii_B  = is_B*Lf + phis_B;
    // PMSM EM Torque Calculation
    m_e = (3*Np/2)*(phis_A*is_B - phis_B*is_A);
    // Inverter/Filter Torque
    m_i = (3*Np/2)*(phii_A*is_B - phii_B*is_A);
    // Delayed Values
    err_phi_A_prev = err_phi_A;
    err_phi_B_prev = err_phi_B;
    x_phi_A_prev = x_phi_A;
    x_phi_B_prev = x_phi_B;
}
```

Figure E.3: *DSP implementation of the flux observer.*

```
void SpeedEstimator() {
    if (theta_r > theta_r_prev + PI) {
        omega_r   = 0.995*omega_r - 0.005*(1.0/Ts)*(theta_r - (theta_r_prev+2.0*PI) );
    }
    else if (theta_r < theta_r_prev - PI) {
        omega_r   = 0.995*omega_r - 0.005*(1.0/Ts)*(theta_r - (theta_r_prev-2.0*PI) );
    }
    else {
        omega_r   = 0.995*omega_r - 0.005*(1.0/Ts)*(theta_r - theta_r_prev);
    }
    theta_r_prev  = theta_r;
    omega_r_fil = 0.997*omega_r_fil + 0.003*omega_r;
}
```

Figure E.4: *DSP implementation of the derivative-filter speed estimation.*

```
// Controller Parameters
double phi_ref  = phi_m;
double kp_PHI  = 2000.0;
double ki_PHI  = 5000.0;
double kp_M   = 0.3;
double ki_M   = 50.0;
double kp_spd  = 10.0;
double ki_spd  = 5.0;

void FluxController() {
    phi_abs   = sqrt( phii_A*phii_A + phii_B*phii_B );

    is_x    = SpaceABd(is_A, is_B, gamma_s -(PI/2.0) );
    is_y    = SpaceABq(is_A, is_B, gamma_s -(PI/2.0) );

    err_phi   = phi_ref - phi_abs;
    u_phi   = u_phi + err_phi*(kp_PHI + ki_PHI*(Ts/2.0) ) + err_phi_prev*( ki_PHI*(Ts/2.0) - kp_PHI );

    u_x    = Rf*is_x + u_phi;

    err_phi_prev = err_phi;
}
void SpeedController() {
    err_omega  = omega_ref - omega_r_fil;

    m_e_ref_spd  = m_e_ref_spd + (kp_spd+(Ts/2.0)*ki_spd)*err_omega + ((Ts/2.0)*ki_spd-kp_spd)*err_omega_prev;

    err_omega_prev = err_omega;
}
void TorqueController() {

    if ( ipState == 2 ) {
        m_i_ref   = -( 0.0185*omega_r_fil*omega_r_fil - 1.575*omega_r_fil );
    }
    else if ( (ipState >= 3) && (ipState <= 8) ) {
        m_i_ref   = m_e_ref_spd;
    }
    else {
        m_i_ref   = 0.0;
    }

    m_i_ref   = m_e_ref_spd;

    err_m   = m_i_ref - m_i;
    u_m    = u_m + err_m*( kp_M + ki_M*(Ts/2.0) ) + err_m_prev*( ki_M*(Ts/2.0) - kp_M );

    u_y    = Rf*is_y + u_m;

    err_m_prev  = err_m;

    u_A   = Spacedq0A( u_x, u_y, gamma_s -(PI/2.0) );
    u_B   = Spacedq0B( u_x, u_y, gamma_s -(PI/2.0) );
}
```

Figure E.5: *DSP implementation code of the DTC.*

```
void OnlineState00(){ // First mode after grid sync: Pre-states
   // Check operating region
   if ( (Vm_sync_fil/phi_m) < omega_cut_in) {
      GotoState( ONLINE_STATE01 );
   }
}
void OnlineState01(){
// setLEDGenerate();    // Turn off LED that indicates generating
   if ( ( (Vm_sync_fil/phi_m) >= omega_cut_in) && ( (Vm_sync_fil/phi_m) < omega_max ) ) {
      PWM_Gen(ENABLE);    // Enable Generator PWM

      u_phi   = 0.0;   // Set starting values for DTC voltages.
      u_m     = Vm_sync_fil;

      err_omega  = 0.0;   // Set starting values for speed controller
      err_omega_prev = 0.0;
      omega_ref  = omega_cut_in; // NB!!!
      omega_r    = omega_cut_in;

      m_e_ref_spd  = -( 0.0185*omega_cut_in*omega_cut_in - 1.575*omega_cut_in ); // NB!!!
      omega_r_fil  = omega_cut_in;

      dat_idx = 0;     // start logging data

      setRelay3();
      setLEDGenerate();
      // Go to state 5
      GotoState( ONLINE_STATE05 );
      //GotoState( ONLINE_STATE02 ); // TEST FOR TORQUE STEP ONLY
   }
}
```

Figure E.6: *Code for states 0 and 1 of the gain scheduling system.*

```
void OnlineState02(){
   if ( (Vm_sync_fil/phi_m) < omega_cut_out ) { // Check if the generator is running too slowly
      // starting values for speed controller
      err_omega  = 0.0;   // Set starting values for speed controller
      err_omega_prev = 0.0;
      omega_ref  = omega_cut_out; // NB!!!
      omega_r    = omega_cut_out;
      m_e_ref_spd  = m_i_fil; // NB!!!
      omega_r_fil  = omega_cut_out;

      dat_idx = 0;     // start logging data
      GotoState( ONLINE_STATE07 ); // Go to cut out state...
   }
   if ( ((Vm_sync_fil/phi_m) >= omega_max ) && Sync_Timer == 0 ) {
      // starting values for speed controller
      err_omega  = 0.0;   // Set starting values for speed controller
      err_omega_prev = 0.0;
      omega_ref  = omega_max; // NB!!!
      omega_r    = omega_max;
      m_e_ref_spd  = m_i_fil; // NB!!!
      omega_r_fil  = omega_max;

      // save last torque command
      m_i_tmp    = m_i_fil;

      // Go to state 3
      dat_idx = 0;     // start logging data
      GotoState( ONLINE_STATE03 );
   }
}
```

Figure E.7: *Code for state 2 of the gain scheduling system.*

```
void OnlineState03(){ // maximum speed control state
    if ( ( fabs(m_i_fil) < 0.8*fabs(m_i_tmp) ) ) { // wait for load to decrease sufficiently before re-entry into
                                                   // variable speed operation.
        omega_ref  = (1.575 + sqrt(2.4806 - 0.074*(0.8*m_i_tmp)))/0.037; // inverse function of MPPT
                                                                         // to calculate temporary speed
        dat_idx = 0;     // start logging data    // reference.
        GotoState( ONLINE_STATE04 );
    }
    if ( ( fabs(m_i_fil) > m_i_max ) ) { // if max torque is reached
        clearRelay3();    // then cut out
        PWM_Gen(DISABLE);
        GotoState( ONLINE_STATE00 ); // Go to pre-states
    }
}
void OnlineState04(){
    if ( fabs(err_omega) < 2.0*PI*1.0 ) { // Wait for speed controller to reach steady state
        dat_idx = 0;     // start logging data
        GotoState( ONLINE_STATE02 );  // before returning to variable speed operation.
    }
}
void OnlineState05(){ // Intermediate state before initial MPPT kicks in (lower speed limit)
    if ( fabs(err_omega) < 2.0*PI*1.0 ) { // Wait for speed controller to reach steady state
        dat_idx = 0;     // start logging data
        GotoState( ONLINE_STATE06 );  // before going to state 6....
    }
}
```

Figure E.8: *Code for states 3 to 5 of the gain scheduling system.*

```
void OnlineState06(){
    if (m_i_fil < -2.0*( 0.0185*omega_cut_in*omega_cut_in - 1.575*omega_cut_in) ) {
        dat_idx = 0;     // start logging data
        GotoState( ONLINE_STATE02 ); // Wait for wind speed to increase more...
                                     // ..then enter MPPT region - variable speed.
    }
    if ( (m_i_fil > 50.0) && ( fabs(err_omega) < 2.0*PI*1.0 ) ) {
        err_omega  = 0.0;   // Set starting values for speed controller
        err_omega_prev = 0.0;
        omega_ref  = omega_cut_out; // NB!!!
        omega_r    = omega_cut_out;

        m_e_ref_spd  = 0.0;//-( 0.0185*omega_cut_out*omega_cut_out - 1.575*omega_cut_out ); // NB!!!
        omega_r_fil  = omega_cut_out;

        dat_idx = 0;     // start logging data
        GotoState( ONLINE_STATE08 );
    }
}
void OnlineState07(){
    if ( fabs(err_omega) < 2.0*PI*0.5 ) { // Wait for speed controller to reach steady state
        dat_idx = 0;     // start logging data
        GotoState( ONLINE_STATE08 );  // before going to state 8....
    }
}
```

Figure E.9: *Code for states 6 and 7 of the gain scheduling system.*

```
void OnlineState08(){
    if ( (m_i_fil > 70.0) && ( fabs(err_omega) < 2.0*PI*0.5 ) ) { // Wait for wind speed to decrease sufficiently
        clearRelay3();          // before re-sync to prevent limit-cycle.

        PWM_Gen(DISABLE);
        dat_idx = 0;     // start logging data
        GotoState( ONLINE_STATE01 ); // Go to pre-cut-in state.
    }
    if ( (m_i_fil < -70) && ( fabs(err_omega) < 2.0*PI*0.5 ) ) { // see if wind speed increases enough to re-enter.
        err_omega  = 0.0;   // Set starting values for speed controller
        err_omega_prev = 0.0;
        omega_ref  = omega_cut_in; // NB!!!
        omega_r    = omega_cut_in;

        m_e_ref_spd  = -70.0;  // NB!!!
        omega_r_fil  = omega_cut_in;

        dat_idx = 0;     // start logging data
        GotoState( ONLINE_STATE05 );
    }
}
```

Figure E.10: *Code for state 8 of the gain scheduling system.*