VSC-HVDC System Modeling and Validation

ROBERT ROGERSTEN

KTH Electrical Engineering

Master's Degree Project
Stockholm, Sweden 2014

XR-EE-EPS 2014:003

Abstract

The performance of traditionally used converter control strategies depends on the ac system conditions. Particularly, the ac system strength is a limiting factor for converter controls to perform well. This work is motivated by the rapid proliferation of high-voltage dc technology, which enables different control structures to be simulated using different simulation tools. The first part of this thesis investigates a generic converter control model on different simulation platforms and implementations. The second part exploits the above-mentioned limiting factor for converter controls to perform well.

In this thesis, a control implementation is tailored to replicate the behavior of another control implementation on different software. In order to ensure that software implementations not become alienated from reality, comparisons to a manufacturer's black-box are also carried out. Furthermore, this thesis demonstrates how many converter stations are linked together by a dc transmission network. A previously proposed control method, power-synchronization control, has been demonstrated to perform well on a point-to-point high-voltage direct current link in connection to a weak ac system. In this thesis, the potential of power-synchronization control is demonstrated in a multi-terminal dc grid with one very weak ac system connection.

Acknowledgments

I would like to thank Dr. Luigi Vanfretti and PhD student Wei Li for making this project possible. I also wish to thank Dr. Lidong Zhang and Dr. Pinaki Mitra for their helpful support during this project.

Contents

1 Introduction **4**
- 1.1 Background and Motivation 4
  - 1.1.1 Control Models on Different Simulation Platforms . . 5
  - 1.1.2 Controller Performance Comparison 5
  - 1.1.3 Power-Synchronization Control 5
- 1.2 Problem Definition and Objectives 6
- 1.3 Limitations 7
- 1.4 General Overview 7

2 High-Voltage Direct-Current Transmission and Control **9**
- 2.1 Conversion Methods 9
  - 2.1.1 Voltage Source Converter Systems 9
  - 2.1.2 Classical HVDC Systems 10
- 2.2 Modeling of Voltage Source Converters 10
- 2.3 VSC-HVDC Transmission 11
  - 2.3.1 Point-to-Point VSC-HVDC Systems 12
  - 2.3.2 Multi-Terminal VSC-HVDC Systems 12
- 2.4 Control Methods for VSC-HVDC Systems 13
  - 2.4.1 Vector-Current Control 13
  - 2.4.2 Power-Synchronization Control 17
  - 2.4.3 Droop Control 19

3 Implementation and Design in PSCAD **22**
- 3.1 Implementation Approach 22
- 3.2 PSCAD Terminology 23
- 3.3 Graphical Implementation in PSCAD 24
  - 3.3.1 The Average Value Model 24
  - 3.3.2 The Control Module 26
  - 3.3.3 Multi-Terminal VSC-HVDC System in PSCAD 32
- 3.4 Control Implementation in PSCAD using Code Generation . 34
  - 3.4.1 Simulink C Code Generation 34
  - 3.4.2 Fortran Integration with C code 36
  - 3.4.3 Linking a library to PSCAD 38

3.4.4 Overall Integration for the Code Implementation . . . 40
3.4.5 Scalability of the Proposed Implementation 44

4 Controller Performance Comparisons and Analysis 46
4.1 Methodology 47
4.1.1 Numerical Comparisons 47
4.1.2 Fault Impedance and System Configuration 48
4.2 Controller Performance Comparisons 48
4.3 Controller Performance Analysis and Results 51
4.4 Black-Box Model Comparisons 52

5 Power-Synchronization Control Analysis 55
5.0.1 The DC Grid Test System 55
5.1 Interconnection of a Weak AC System 58
5.1.1 Estimation of the Short-Circuit Ratio 58
5.1.2 Step Response of the Active-Power Controller 58
5.1.3 Three-Phase Fault at the Bus of Converter A 59

6 Conclusions and Further Work 63
6.1 Model Design and Controller Comparisons 63
6.1.1 Graphical Implementation 63
6.1.2 Code Implementation 64
6.1.3 Further Work 64
6.1.4 Black-Box Model Comparison 65
6.2 Power-Synchronization Control 65

7 Bibliography 66

A Voltage Plots During Faults 69

B Graphical Implementation 73
B.1 Step of the Active- and Reactive-Power Controller 73
B.2 Three-Phase Faults (Fault Scenarios 3 to 5) 78
B.3 Single-Phase Faults (Fault Scenarios 6 to 8) 85

C Code Implementation 92
C.1 Step of the Active- and Reactive-Power Controller 92
C.2 Three-Phase Faults (Fault Scenarios 3 to 5) 97
C.3 Single-Phase Faults (Fault Scenarios 6 to 8) 104

D Black-Box Model Comparisons 111
D.1 Step of the Active- and Reactive-Power Controller 111
D.2 Three-Phase Faults at Rectifier Side (Fault Scenarios 3 to 5) 114
D.3 Single-Phase Faults at Rectifier Side (Fault Scenarios 6 to 8) 118
D.4 Three-Phase Faults at Inverter Side (Fault Scenarios 3 to 5) . 122

D.5 Single-Phase Faults at Inverter Side (Fault Scenarios 6 to 8) . 126

Chapter 1

Introduction

The revival of direct current (dc) for long-distance power transmission began in 1954 when ASEA, a predecessor of ABB, linked the island of Gotland to mainland Sweden with high-voltage direct-current (HVDC) lines. However, the history of dc began before that. Thomas Edison lost the war of dc against alternating current (ac) and in the 1890s it became clear that ac was more efficient at transmitting electricity over long distances. However, dc has staged a roaring comeback during recent years because of several reasons. One reason is the increase in electricity consumption throughout the world. In Europe alone, total electricity consumption has increased by 32.8% from 1990 to 2007. As a result, the European high-voltage alternating-current (HVAC) grid is operating very close to its limits.

Today, there are many HVDC projects in operation worldwide. Power transmission using dc allows for transporting large amounts of electricity across long distances, exceeding the transport capacity of ac lines. Classical HVDC technology utilizes thyristors for power conversion. However, recently, a new more advantageous HVDC technology based on voltage source converters (VSC) has emerged [1, 2]. The VSC technology utilizes gate-turn-off thyristors (GTOs) or in most industrial cases insulated gate bipolar transistors (IGBTs) for switching. For example can VSC-HVDC systems address issues regarding power transmission, asynchronous network interconnections, and stability support.

1.1 Background and Motivation

In this section, the background and motivation of the project are discussed. Simulation results from generic control models often differ between simulation platforms. Therefore, part of this work is motivated by the need to investigate such differences and to facilitate the transfer between simulation platforms. Another motivation for this project is the increasing need to compare control models for VSC-HVDC systems used in academic research to control models

from industry.

This thesis will discuss the commonly used control method referred to as *vector-current control*. It will also discuss the alternative control method referred to as *power-synchronization control*. Power-synchronization control has been implemented on control models for comparison to the traditionally used vector-current control.

1.1.1 Control Models on Different Simulation Platforms

In this thesis, a generic control model is developed based on the Cigre generic control model. Cigre is a working group that deals with HVDC transmission and power electronics for use in transmission and distribution networks. The Cigre working group has developed a generic control model in Simulink, which will be used as a reference model in this work. This thesis investigate the generic control model on different simulation platforms and implementations. Therefore, the control model has been developed in PSCAD[1]. Furthermore, the C code has been extracted from the Simulink model for implementation in PSCAD[2]. Using this approach, the controls in PSCAD can be tailored to replicate the behavior of Simulink very well.

1.1.2 Controller Performance Comparison

As already discussed, the dc grid developments and applications have made rapid progress during recent years. Among with the technology advances there exist many commercial and trade secret reasons from industry regarding VSC-HVDC converter stations. Particularly, there are plentiful questions regarding the control of VSC-HVDC converters. There is also a crucial need of realistic data regarding controller performance. Without consistent comparison and validation with the realistic data used in industry, the academic research could become alienated from reality. This motivates the comparison between generic control models and realistic results from industry carried out in this thesis.

1.1.3 Power-Synchronization Control

The theory of power-synchronization control is detailed in [3]. The potential of power-synchronization is revealed when the VSC is connected to a weak ac system. As discussed in [4], a description of the strength of the ac system

[1] In the past two sentences both PSCAD and Simulink (an extension of Matlab) has been mentioned. PSCAD and Simulink are graphical tools that can be used for simulation of electric power systems, for more information see chapter 3

[2] To distinguish between these implementations, the traditional way to build a model will be referred to as the graphical implementation, while the other approach will be referred to as the C code implementation.

relative to the power rating of the HVDC link can be given by the short-circuit ratio (SCR). Thus, if the calculated SCR is low the ac system is weak. It is discussed in [3] that the conventional thyristor based HVDC system cannot work properly if the SCR is low.

In contrast to the conventional thyristor based HVDC system, the VSC-HVDC system can produce its own voltage waveform independent of the ac system, which means that a VSC-HVDC system has the potential to be connected to very weak ac systems. However, with the traditional vector-current control the potential of the VSC is not fully utilized [3, 5, 6, 7]. Vector-current control inherits poor performance for weak ac system connections.

Motivated by this fact, power-synchronization control is implemented as an alternative control method for comparison of control performance on a system with very low SCR. Thus far, power-synchronization control has only been applied to point-to-point interconnections [6]. Therefore, power-synchronization control is implemented in a multi-terminal VSC-HVDC system in this thesis. This work is also presented in [8].

1.2 Problem Definition and Objectives

In this thesis, a control model for VSC-HVDC systems is implemented in PSCAD based on the Cigre generic control model developed in Simulink. The Simulink model will be considered as a reference for the generic control model. Therefore, comparisons to the Simulink model are performed in order to ensure that deviations are not too large. Further, comparisons are performed with a manufacturer's black-box model. These comparisons together with an analysis of how power-synchronization control behaves in a multi-terminal dc system will form this thesis. Basically, the objectives will be:

1) Compare and analyze the two control methods, vector-current control, and power-synchronization control in a multi-terminal dc system.
2) Compare the graphical implementation and the C code implementation to the Simulink model.
3) Compare the best of these implementations to the manufacturer's black-box model[3].

Working against these objectives led to two chapters that presents the results: (i) controller performance comparisons and analysis, and (ii) power-synchronization control analysis. Figure 1.1 illustrates this by a flowchart. The grey box in the bottom left of figure 1.1 represents how objective 1) is accomplished, while the grey box in the bottom right represents how objective 3) is accomplished. The two grey boxes in the middle corresponds

[3]The best here will be the C code implementation, it will be explained later that this implementation has almost a perfect match to the Simulink model.

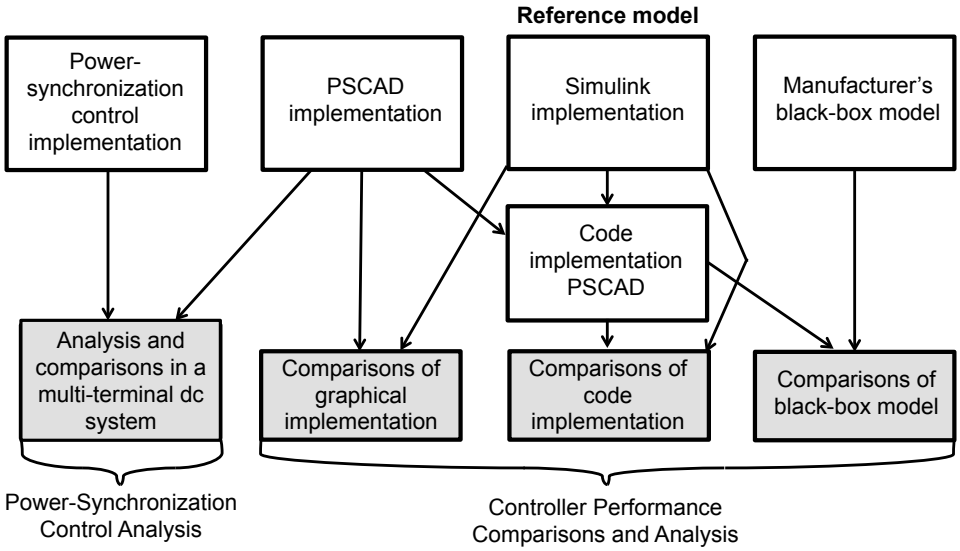


Figure 1.1: Flowchart of different model implementations, which led to the main results in this thesis. The grey box in the bottom left yields the chapter called power-synchronization control analysis. The other three grey boxes compare different implementations according to the flow chart. These comparisons will yield the chapter called controller performance comparisons and analysis.

to objective 2). Note that the grey boxes leads to the two final chapters presented in this thesis.

1.3 Limitations

This project will focus on the VSC-HVDC systems and the control methods used for such systems. The controls of a VSC-HVDC system can be divided into *lower level controls* and *upper level controls*. The upper level control system calculates three phase voltage references and feeds them into the lower level control system. The lower level control system will not be discussed in this thesis. Thus, the emphasis will be on upper level control of VSC-HVDC systems.

1.4 General Overview

This thesis will be structured as follows. Chapter 2 introduces HVDC transmission and the control strategies used for VSC-HVDC systems. First the traditional HVDC technology and the VSC technology are discussed. The point-to-point model and the HVDC grid are introduced. Further, the control strategies used in this thesis are discussed.

Chapter 3 explains how the model has been developed and implemented

in PSCAD. Chapter 3 also discusses general terminology used when working in PSCAD. This terminology is useful when discussing different type of objects implemented in PSCAD. The generic control model is constructed in two ways, graphically and by the use of C code. The first part of chapter 3 explains how to construct the model graphically, while the last part shows how to construct the model using C code extracted from the Simulink model.

Chapter 4 demonstrates the performance of vector-current control in detail. Several comparisons are carried out between software implementations. Also, comparisons with a manufacturer's black-box model are carried out.

Chapter 5 will discuss the results from simulations using power-synchronization control. Particularly, these simulations are carried out in a multi-terminal VSC-HVDC system.

Finally, chapter 6 draws some conclusions and discusses further work.

Chapter 2

High-Voltage Direct-Current Transmission and Control

A variety of technical, economical and environmental reasons are forcing the traditionally ac power transmission development to rethink. An important factor is the need of increased power carrying capability of transmission lines. Another major factor is the restriction that two interconnected ac systems need to be in synchronism [9]. This chapter discusses the conversion methods to overcome these issues and how they can be controlled using different strategies. The discussion should comprise all theory used in this thesis

2.1 Conversion Methods

This work will focus on HVDC systems based on voltage source converters (VSC), which are one out of two existing power conversion methods. Normally, the designation ac–dc power conversion is used for the processes of rectification and inversion. Two major benefits with these two processes are the improved controllability and the removal of synchronous constraints between two connection points. In this section, the two different methods used for electric power conversion are described.

2.1.1 Voltage Source Converter Systems

As previously explained, the VSC-HVDC technology has been an area of growing interest during recent years. Therefore, this thesis focuses on VSC-HVDC systems. A VSC has a voltage source connected on the dc side in the form of a large capacitor appropriately charged to maintain the required voltage. A constraint imposed on the circuit of a power converter is that one side needs to be inductive and another capacitive to prevent a loop consisting of voltage sources [9]. On a VSC, the ac side has an inductance connected, which has two purposes: first, it stabilizes the ac current and second, it

enables the control of active and reactive output power from the voltage source converter [9]. A VSC requires self-commutating switches such as gate turn off thyristors (GTO) or insulated-gate bipolar transistors (IGBT), which has a turn-on and turn-off capability so the position and frequency of the on and off switching instants can be altered to provide a specific voltage and current waveform [9, 3].

2.1.2 Classical HVDC Systems

If, in contrast to the VSC, a large smoothing reactor is placed on the dc side, pulses of constant direct current flow through the switching devices into the ac side. The same constraint as before is imposed on the circuit, therefore, capacitors is needed on the ac side. This arrangement is commonly referred to as a current source converter (CSC). A CSC utilizes thyristor valves for switching purposes. In contrast to the switches of a VSC, reversed line voltages can only turn off the thyristor valve. Therefore, it relies on the natural current zeros created by the external circuit for the transfer of current from switch to switch [9]. When the source is the ac system voltage, the converter is said to be line-commutated (LCC). Converters based on thyristor valves are called line-commutated converters (LCCs), or current-source converters (CSCs). Some common problems with the LCC technology found in [9] and [3] are:

- The converter always consumes reactive power.
- There is often an occurrence of commutation failures at the inverter station.
- There is a lack of waveform quality, normally in the form of current harmonic content.

2.2 Modeling of Voltage Source Converters

It is possible to model a VSC, either in detail or by using an average value model (AVM). If the VSC is modeled in detail all semiconductor components such as IGBTs can be included as a single unit represented in the model. To perform a simulation using a detailed model compared to an AVM can be rather time consuming; therefore, an AVM is used in this work. The AVM replicates the average response of the converter by using controlled sources and switching or averaged functions. An AVM model is proposed in [10]; the PSCAD model is based on the same type of AVM model. The PSCAD model utilizes only the fundamental frequency component, which is less time consuming, particularly when dealing with large dc grids. The proposed model in [10] includes a more detailed AVM with controlled sources that includes the harmonic content from the modulation control (or switching functions) on the ac voltage waveforms.

Normally, in an AVM model the relationship between the ac and dc circuits are (this is discussed in [11]):

$$v_i = \frac{m_i U_{\text{dc}}}{2}, \quad I_c = \frac{P}{U_{\text{dc}}} = \frac{1}{2}(m_1 i_1 + m_2 i_2 + m_3 i_3), \tag{2.1}$$

where m_i is the modulation index and v_i the phase voltage of phase i. In order to describe the AVM model the modulation index is defined according to [10] as

$$m_i = 2\frac{v_i^{ref}}{U_{\text{dc}}}, \tag{2.2}$$

where v_i^{ref} is the reference voltage of phase i and U_{dc} is the dc voltage.

In addition to (2.1), it is desirable to model the losses at the converter station. Therefore, consider the power flow P on the ac side, the power flow P_{dc} on the dc side, and the converter losses P_L. The power flow relationship will be

$$P = P_{\text{dc}} + P_L. \tag{2.3}$$

If (2.3) is divided with U_{dc} it follows that

$$I_c = \frac{P}{U_{\text{dc}}},\ I_{\text{dc}} = \frac{P_{\text{dc}}}{U_{\text{dc}}},\ I_L = \frac{P_L}{U_{\text{dc}}} \implies I_c = I_{\text{dc}} + I_L, \tag{2.4}$$

and the dc current in the converter will be

$$I_{\text{dc}} = I_c - I_L, \tag{2.5}$$

where the converter losses is modeled by the current

$$I_L = \frac{P_L}{U_{\text{dc}}} = R\frac{I_c^2}{U_{\text{dc}}}. \tag{2.6}$$

The resistance R is the equivalent resistance of the converter representing both switching and resistive losses.

2.3 VSC-HVDC Transmission

The use of HVDC technology has traditionally been limited to point-to-point interconnections. However, there is an increasing interest of more interconnection points constituting to a dc grid. This is because of technological advances in power electronics and VSC systems [2] but also due to grid integration challenges from remotely located generation sites [11]. Since there are two types of HVDC conversion methods, two types of dc grids are possible. However, this work will only consider the VSC type. Next, the point-to-point VSC-HVDC system and the multi-terminal VSC-HVDC system are discussed.

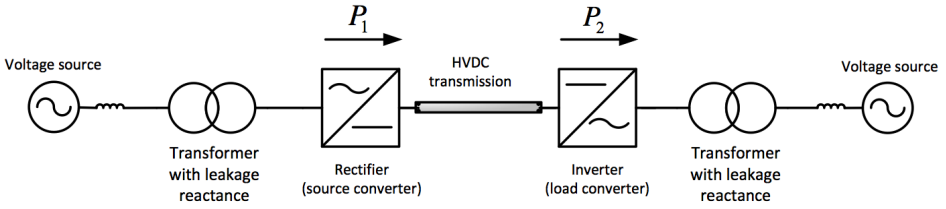


Figure 2.1: A point-to-point connection between two ac systems. The power flow is from the rectifier station towards the inverter station. The transformers have a leakage reactance that is necessary for the VSC to work properly.

2.3.1 Point-to-Point VSC-HVDC Systems

The scheme of a HVDC point-to-point station is shown in figure 2.1. The point-to-point station is mainly used to transmit active power from one ac network to another ac network. The power flow is from the rectifier towards the inverter. The study of a point-to-point terminal gives substantial information on the characteristics of a multi-terminal VSC-HVDC system. In the point-to-point terminal connection there will be a power loss P_{L_1} in the rectifier station and a power loss P_{L_2} in the inverter station; there will also be a power loss $P_{L,\mathrm{dc}}$ in the HVDC transmission line connecting the stations. Therefore, the power flow at the rectifier station P_1 will be different from the power flow at the inverter station P_2. The relationship can be written as

$$0 = P_1 + P_2 + P_{L_1} + P_{L_2} + P_{L,DC}. \tag{2.7}$$

In order to get the desired amount of power that is transferred between systems, one converter station will control the active power and another will control the dc voltage [11].

2.3.2 Multi-Terminal VSC-HVDC Systems

A multi-terminal VSC-HVDC system contains of three or more converter stations that are linked together by a dc transmission network. Multi-terminal VSC-HVDC systems is a relatively new research field, which has been attracting increasing attention since the turn of the century [11]. A transmission network that includes many interconnected points and transfer power across great distances is often referred to as a “super grid”. Recently, discussions about a multi-terminal VSC-HVDC system that constitute to an European dc super grid have emerged. An European dc super grid could be embedded in a conventional ac grid and provide a strong backbone to existing ac networks, along with all the benefits of VSC-HVDC technology (e.g., the ability to address issues of power transmission, asynchronous network interconnections, and stability support).

Within a dc grid there can at most be one converter station assigned to constant dc voltage control mode, the other converter stations are assigned to active power control mode. If there are two converters with constant dc voltage control, there will be hunting phenomena [11]. The converter station assigned to dc voltage control mode is often referred to as the dc slack-bus [12]. The dc slack-bus needs to ensure keeping the dc voltage constant and compensating for all power unbalances in the dc grid including losses. An extension of (2.7) for N converter stations can be expressed as

$$0 = \sum_{n=1}^{N} P_i + \sum_{n=1}^{N} P_{L_i} + P_{L,\mathrm{dc}}, \tag{2.8}$$

where P_i is the active power flow at the converter bridge of station i, P_{L_i} is the losses in converter station i, and $P_{L,\mathrm{dc}}$ is the dc line losses. Because of the dc slack-bus configuration at one station and active power control configuration at other stations there will not be any steady state power deviations in the power controlled at the VSC-HVDC terminals.

Another approach that is susceptible to steady state power deviations in all of the VSC- HVDC terminals is the dc voltage droop control [11, 13]. This approach is preferred in comparison to the above method for several reasons. Droop control will be further discussed in the next section.

2.4 Control Methods for VSC-HVDC Systems

This section will discuss control methods for VSC-HVDC systems. Classical HVDC systems and its controls have not been studied in this thesis. As for all-round education, the implementation of an automatic voltage stabilizer for classical HVDC can be read about in [14].

The control strategies discussed next are vector-current control, power-synchronization control, and droop control. For the controls to work properly, the three-phase currents and voltages are transformed to d and q axes. Basically, the fundamental current and voltages become dc components. Therefore, PI-controllers can be used to reduce steady state errors. The final step of both control strategies (vector-current control and power-synchronization control) is to transform the d and q voltages to three-phase quantities and let the VSC realize them into line voltages.

2.4.1 Vector-Current Control

Vector-current control has been successfully applied on many real-life VSC-HVDC links. There exists many design approaches for the vector-current control strategy. In this section, a common design approach of vector-current control, often referred to as Diagonal Internal Model Control (DIMC), is discussed. The Internal Model Control (IMC) and DIMC principle are

discussed in [15] and [3]. Vector-current control has initially been applied on variable speed drives as illustrated in [15]. Another common design approach for vector-current control is the deadbeat-current control design. This design approach is detailed in [16]. However, the deadbeat-current design approach can only be realized by digital implementations.

Next, the design procedure is described. The relationship between the converter current, $\mathbf{i}_{dq} = \begin{pmatrix} i_d & i_q \end{pmatrix}^T$, the bridge ac voltage, $\mathbf{v}_{dq} = \begin{pmatrix} v_d & v_q \end{pmatrix}^T$, and the ac line voltage, $\mathbf{u}_{dq} = \begin{pmatrix} u_d & u_q \end{pmatrix}^T$, in the dq-plane is

$$\mathbf{v}_{dq} = \mathbf{u}_{dq} + \omega_1 L \begin{pmatrix} i_q \\ -i_d \end{pmatrix} - L\frac{d\mathbf{i}_{dq}}{dt} - r\mathbf{i}_{dq}, \tag{2.9}$$

where ω_1 is the angular frequency of the ac system, L is the leakage inductance of the transformer, and r is the interconnecting resistance. The resistance r in high voltage applications is usually small and therefore neglected. Therefore, an approximation of each element of $\mathbf{v}_{dq}$ in (2.9) gives

$$\begin{aligned} v_d &= u_d + \omega_1 L i_q - L\frac{di_d}{dt}, \\ v_q &= u_q - \omega_1 L i_d - L\frac{di_q}{dt}. \end{aligned} \tag{2.10}$$

The system is obviously coupled. Therefore, a decoupler is added to an inner feedback loop of the system. By letting

$$\begin{aligned} v_d &= u_d + v'_d + \omega_1 L i_q, \\ v_q &= u_q + v'_q - \omega_1 L i_d, \end{aligned} \tag{2.11}$$

the decoupled system becomes

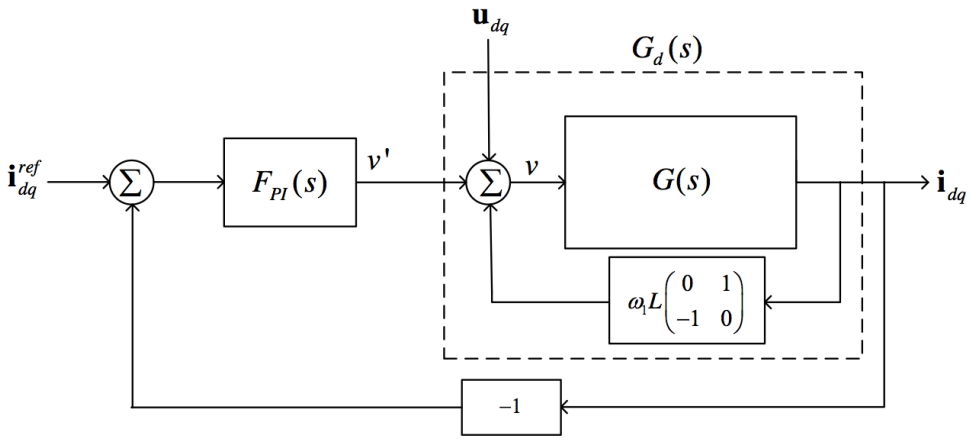
$$\begin{aligned} v'_d &= -L\frac{di_d}{dt}, \\ v'_q &= -L\frac{di_q}{dt}. \end{aligned} \tag{2.12}$$

The transfer function from $\mathbf{v}'_{dq}$ to $\mathbf{i}_{dq}$ is therefore given by

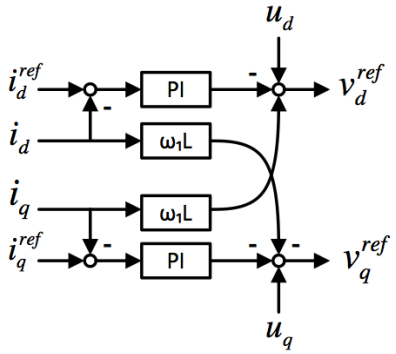
$$G_d(s) = \begin{pmatrix} -\frac{1}{sL} & 0 \\ 0 & -\frac{1}{sL} \end{pmatrix}. \tag{2.13}$$

The transfer function in (2.13) is decoupled and can therefore be controlled with a diagonal PI-controller, which means that the d and q components can be controlled independently as two single-variable systems. An illustration of the decoupling is shown in figure 2.2. The PI-controller can be expressed as

$$F_{\mathrm{PI}}(s) = -\begin{pmatrix} k_p + \frac{k_i}{s} & 0 \\ 0 & k_p + \frac{k_i}{s} \end{pmatrix}. \tag{2.14}$$



(a) Block-diagram of the feed-back loop. $F_{\mathrm{PI}}(s)$ is the diagonal PI-controller and $G(s)$ is the system transfer function.



(b) Detailed diagram that illustrates the separate PI-controllers and the reference voltages. The separate PI-controllers are given by $\mathrm{PI} = k_p + \frac{k_i}{s}$.

Figure 2.2: The inner control loop of vector-current control. The inner control loop has an inner decoupling. Basically, figure **(a)** and **(b)** are two ways to represent the same thing. However, figure **(a)** gives a better overview of the diagonal transfer function $G_d(s)$.

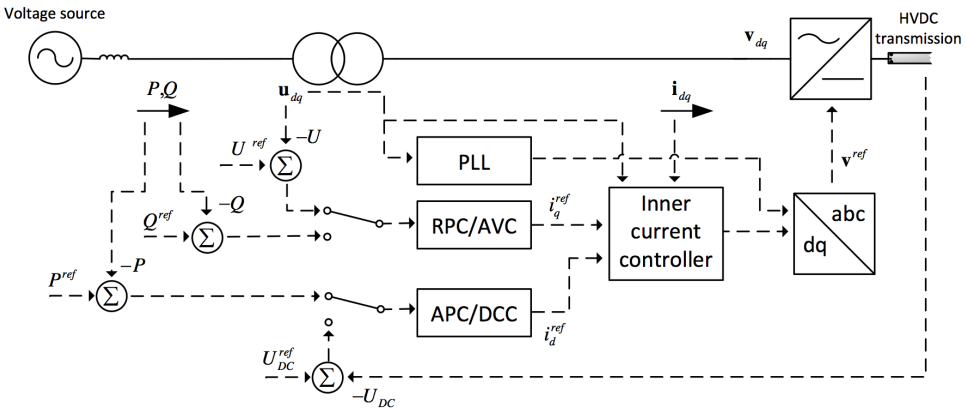


Figure 2.3: Main circuit including the control block diagram for vector-current control. The blocks include the phase-locked loop (PLL), reactive-power controller (RPC), alternating-voltage controller (AVC), active-power controller (APC) and dc-controller (DCC).

The decoupled system has a negative transfer function; therefore, the PI-controller is implemented with a minus sign. On some implementations, a low pass filter $H_{\mathrm{LP}}(s)$ is added to the control law to improve the disturbance rejection. Assuming $\mathbf{v}_{dq}^{\mathrm{ref}} = \mathbf{v}_{dq}$ yields the following control law

$$\mathbf{v}_{dq}^{\mathrm{ref}} = \begin{pmatrix} u_d \\ u_q \end{pmatrix} + F_{\mathrm{PI}}(s) \begin{pmatrix} i_d^{\mathrm{ref}} - i_d \\ i_q^{\mathrm{ref}} - i_q \end{pmatrix} + \omega_1 L \begin{pmatrix} i_q \\ -i_d \end{pmatrix}, \tag{2.15}$$

where the references i_d^{ref} and i_q^{ref} are given by an *outer control loop*. In a traditional control design, the current control in (2.15) is referred to as the *inner control loop*. The vector-current control strategy is illustrated in figure 2.3.

Outer Loop Converter Control

The outer control loop feeds the reference current to the inner control loop in order to maintain an adequate reference voltage for the VSC. Depending on the mode of operation, the reference i_d^{ref} is used to control the active power or direct voltage. In the same way, the reference i_q^{ref} is used to control the reactive power or ac voltage. There exist several ways to calculate the reference currents. In this work, an integral controller with feed-forward is used. The reference currents if the active- and reactive powers are controlled are calculated as

$$\mathbf{i}_{dq}^{\mathrm{ref}} = \frac{1}{V} \begin{pmatrix} P_{\mathrm{ref}} + \frac{k_i}{s}(P_{\mathrm{ref}} - P) \\ -Q_{\mathrm{ref}} - \frac{k_i}{s}(Q_{\mathrm{ref}} - Q) \end{pmatrix}, \tag{2.16}$$

where $V = |\mathbf{v}_{dq}| = \sqrt{v_d^2 + v_q^2}$ is the voltage magnitude at the converter bridge.

Recall that $P = v_d i_d + v_q i_q$ and $Q = v_q i_d - v_d i_q$. If the voltage at the converter bridge is aligned with the d axis and kept at 1 pu (i.e., $v_q = 0$ and $v_d = 1\,\text{pu}$), the currents can be expressed as $i_d = P/v_d = P$ and $i_q = -Q/v_d = -Q$. Therefore, a simpler controller instead of (2.16) could be

$$\mathbf{i}_{dq}^{\text{ref}} = \begin{pmatrix} P_{\text{ref}} \\ -Q_{\text{ref}} \end{pmatrix}. \tag{2.17}$$

However, the integral action from the controller in (2.16) helps to compensate for too abrupt power changes when the reference is changed. Also, the controller (2.17) needs the voltage constant and at its nominal value. Therefore, the controller from (2.16) is a preferable choice.

As an alternative to the traditional PI-controller, [17] explored the properties of an IP-controller and showed some advantages compared to the PI-controller for implementations on dc drives. Figure 2.4 illustrates the difference between a PI-controller and an IP-controller. The developed PSCAD model utilizes an IP-controller. The reference currents for direct- and alternating voltages in an IP-controller are calculated as

$$\mathbf{i}_{dq}^{\text{ref}} = \begin{pmatrix} \frac{k_i}{s}(U_{\text{dc}}^{\text{ref}} - U_{\text{dc}}) - k_p U_{\text{dc}} \\ \frac{k_i}{s}(U^{\text{ref}} - U) - k_p U \end{pmatrix}, \tag{2.18}$$

where $U = |\mathbf{u}_{dq}| = \sqrt{u_d^2 + u_q^2}$ is the voltage magnitude at the primary side of transformer and U_{dc} is the direct voltage. It is also possible to control the voltage at the ac converter bridge instead of primary side of transformer by replacing U with V. In order to improve disturbance rejection a low pass filter $H_{\text{LP}}(s)$ can be added for the controllers in (2.16) and (2.18).

In addition to (2.16) and (2.18), there exist two more control modes. If the system is configured to control the active power and the ac voltage the reference currents are calculated as

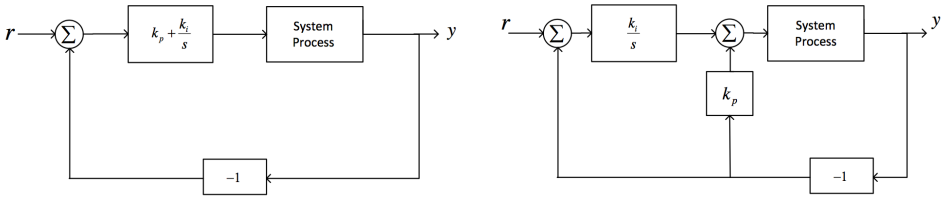
$$\mathbf{i}_{dq}^{\text{ref}} = \begin{pmatrix} \frac{1}{V}[P_{\text{ref}} + \frac{k_i}{s}(P_{\text{ref}} - P)] \\ \frac{k_i}{s}(U^{\text{ref}} - U) - k_p U \end{pmatrix}, \tag{2.19}$$

and if the system is configured to control the direct voltage and the reactive power the reference currents are calculated as

$$\mathbf{i}_{dq}^{\text{ref}} = \begin{pmatrix} \frac{k_i}{s}(U_{\text{dc}}^{\text{ref}} - U_{\text{dc}}) - k_p U_{\text{dc}} \\ \frac{1}{V}[-Q_{\text{ref}} - \frac{k_i}{s}(Q_{\text{ref}} - Q)] \end{pmatrix}. \tag{2.20}$$

2.4.2 Power-Synchronization Control

As already mentioned, there exists many real-life VSC-HVDC links where vector-current control has been successfully applied. However, a major



(a) PI-controller **(b)** IP-controller

Figure 2.4: Block diagram to illustrate the difference between the PI- and IP-controller.

drawback with vector-current control is the difficulty to adapt VSC-HVDC transmission on more challenging ac system conditions. A VSC-HVDC link connected to a weak ac system inherits poor performance. Therefore, the alternative control method referred to as power-synchronization control is introduced. Power-synchronization control is proposed in [18] as a control method for grid connected VSC. Furthermore, [6] shows how two weak ac systems can be successfully interconnected using the control method. In addition, [19] shows a successful implementation of power-synchronization control on low inertia systems. Also, a case study of an offshore wind integration to a weak grid using power-synchronization control is presented in [20]. Furthermore, an analysis of the stability limitations for this control method is detailed in [21].

For a traditional control design, a PLL is used to synchronize the VSC with the ac system. An alternative way is to use a power-synchronization loop (PSL), which instead synchronizes the VSC with the ac system by active-power control. This is one of the key concepts of power-synchronization control. By using power as a way of synchronization, the VSC avoids the instability caused by a standard PLL in a weak ac system connection [5, 7].

The PSL is proposed in [3] as[1]

$$\theta_v = -\frac{k_p}{s}(P_{\text{ref}} - P), \tag{2.21}$$

where θ_v is the synchronization input to the VSC, which means that $\omega t = \omega_{\text{ref}} t + \theta_v$. The VSC synchronizes with the ac system through the active-power control loop in (2.21), the operation is similar to a synchronous machine. Basically, the power synchronization control emulates a synchronous machine. Therefore, no requirements on the short-circuit capacity are imposed on the system and the VSC terminal can give a weak ac system strong voltage support, just like a normal synchronous machine does [6].

[1] In this work, power-synchronization control has been implemented on a model which has the current defined in an opposite direction compared to the work in [18]. Therefore, (2.21) will differ by a minus sign from the commonly used equation.

An inner loop for converter control is not a necessary condition when power-synchronization control is used because the power is controlled directly from the PSL and the alternating-voltage is controlled directly by the VSC voltage. The work in [18] propose an integral controller that will maintain a voltage of 1 pu with the alternating-voltage controller (AVC). As an alternative, the IP-controller is implemented as

$$\Delta V = \begin{pmatrix} \frac{k_i}{s}(U_{\text{ref}} - U) - k_p U \\ 0 \end{pmatrix}. \tag{2.22}$$

It is possible to set $\mathbf{v}_{dq}^{\text{ref}} = \Delta V$ if the inner loop of the converter control is not used. However, there mainly exists two benefits in keeping the inner loop in (2.15): (i) it provides damping effects to poorly damped resonances in ac systems, and (ii), it limits the valve current of the converter during severe ac system faults.

As previously discussed, the VSC needs to synchronize to the ac system either by the PLL or the PSL. In vector-current control mode, the converter control use the PLL. In power-synchronization control mode, the converter control use the PSL. In order to keep the same inner loop for both control strategies, the trick for power-synchronization control mode is to arrange a current reference control so that $\mathbf{i}_{dq}^{\text{ref}} = \begin{pmatrix} i_d^{\text{ref}} & i_q^{\text{ref}} \end{pmatrix}^T$ in (2.15) yields a voltage vector control law. This is achieved by

$$\mathbf{i}_{dq}^{\text{ref}} = \frac{1}{k_p}[\Delta V + H_{\text{HP}}(s) \begin{pmatrix} i_d \\ i_q \end{pmatrix} - \begin{pmatrix} u_d \\ u_q \end{pmatrix} - \omega_1 L \begin{pmatrix} i_q \\ -i_d \end{pmatrix}] + \begin{pmatrix} i_d \\ i_q \end{pmatrix}, \tag{2.23}$$

where $H_{\text{HP}}(s)$ is a high-pass filter for damping purposes and ΔV is given by the AVC in (2.22).

Figure 2.5 illustrates how power-synchronization control can be implemented.

2.4.3 Droop Control

A thorough discussion about how droop control is implemented can be read about in [11] and [13]. If a direct-voltage controller is implemented according to (2.18), the relationship between direct voltage and active power will be as shown in figure 2.6 (this is discussed in [13]). In the same way, if an active-power controller is implemented according to (2.16), the relationship will be as in figure 2.7. As discussed in chapter 2, a terminal with direct-voltage control will be referred to as the dc slack-bus station and the other stations as active-power controller stations. In such configuration, there will not be any steady state power deviations in the power controlled at the VSC-HVDC terminals; the active power flow will be according to (2.8). If instead the system adopts direct-voltage droop control the terminals will be susceptible to steady state power deviations. If there is an *unaccounted*

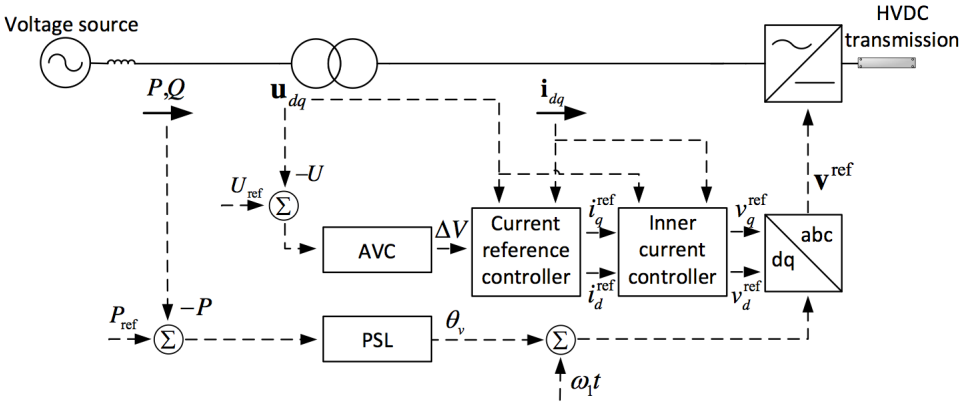


Figure 2.5: Main circuit including the control block diagram for power-synchronization control. The blocks include the alternating voltage controller (AVC) and the power-synchronization loop (PSL).

power deviation in the system because of dc line voltage drops, dc line power losses, or converter power losses, it is often necessary for more than one converter to be susceptible to steady state power deviations. As explained in [11, 13], instead of using the input $e = U_{\mathrm{dc}}^{\mathrm{ref}} - U_{\mathrm{dc}}$ as in (2.18), it is possible to use

$$e = P^{\mathrm{ref}} - P - \frac{U_{\mathrm{dc}}^{\mathrm{ref}} - U_{\mathrm{dc}}}{\rho}, \tag{2.24}$$

where ρ is the droop gain. When e approaches zero, the dc bus voltage relationship will be

$$e \to 0 \quad \Longrightarrow \quad U_{\mathrm{dc}} = U_{\mathrm{dc}}^{\mathrm{ref}} + \rho(P - P^{\mathrm{ref}}). \tag{2.25}$$

Therefore, the terminals that adopts direct-voltage droop control will be susceptible to steady state power deviations. Figure 2.8 shows the relationship between direct voltage and active power in a terminal that adopts direct-voltage droop control.

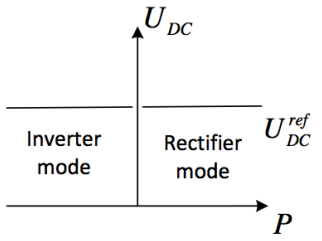


Figure 2.6: Relationship between direct voltage and active power in a VSC-HVDC system that adapts direct-voltage control.

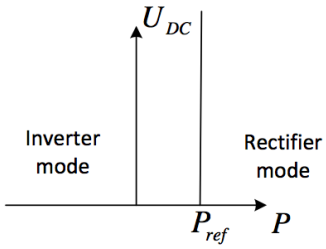


Figure 2.7: Relationship between direct voltage and active power in a VSC-HVDC system that adapts active-power control.

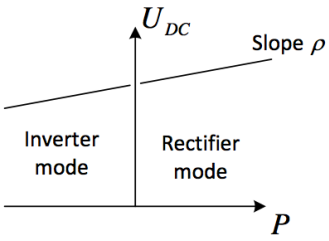


Figure 2.8: Relationship between direct voltage and active power in a VSC-HVDC system that adapts dc droop control. The relationship will be according to (2.24) and therefore the curve has slope ρ.

Chapter 3

Implementation and Design in PSCAD

In this work, Simulink (an extension of Matlab) and PSCAD have been used to simulate VSC-HVDC systems. Both Simulink and PSCAD are graphical tools with a wide variety of built-in functions that can be assembled into complete systems. Particularly, Matlab Simulink contains the SimPowerSystems toolbox, which is a physical modeling tool for electric power systems. With SimPowerSystems and PSCAD, models for an entire electric power system can be built just as it would be assembled from physical components. The PSCAD model developed in this work is based on a Cigre generic control model developed in Simulink. This section will describe the structure of the PSCAD model.

3.1 Implementation Approach

There exist several ways to implement the proposed control strategies in PSCAD. The PSCAD control structure shall replicate the control structure from the Simulink model. Therefore, three suggestions for the PSCAD implementation are:

1) Synchronize Simulink to PSCAD by the built in PSCAD interface to Simulink.

2) Implement the model graphically.

3) Generate C code from the Simulink model and let PSCAD integrate with the C code during run-time.

Alternative 1) utilize a user-defined block in PSCAD, which specify the necessary inputs and outputs, to interface with a Matlab/Simulink file. In order to ensure correct results, PSCAD proceeds only after the Simulink module simulation is completed each time step. In other words, PSCAD calls Simulink, which runs a whole simulation and then returns the result to

PSCAD. A demonstration of this implementation on HVDC systems can be read about in [22], where the authors discovered that a simulation duration of 2 s, took 30.5 s in PSCAD, 72.2 s in Simulink, and 12503.58 s when the PSCAD interface to Simulink was used. To avoid too lengthy simulations alternative 1) has been left out in this thesis.

Alternative 2) was considered to be the easiest way to implement both controls strategies. Therefore, the VSC-HVDC system was built up using this approach. However, for simulations using vector-current control it is necessary that the models behaviors are similar. The author discovered that an implementation using alternative 3) yields a much more similar behavior. Therefore, vector-current control has also been implemented using alternative 3).

When the model was implemented using C code, each control component was generated piece-by-piece. This approach was preferable in order to facilitate debugging. The first approach was to generate all of the code at once for implementation. However, after several failures the author started from the graphical implementation and replaced each component piece-by-piece, which ensured a functional model after each component exchange. Therefore, only the components that behave different between simulation platforms have been exchanged from the graphical implementation.

Most of the blocks in Simulink have support for C code generation. A complete list of blocks that has support for C code generation can be found in the help section of Matlab. After code generation, the logic operator block encountered some troubles when all of the code was implemented at once. Within the Simulink model the logic operator outputs an integer (1 for true and 0 for false), which is multiplicated with a real control signal. Within the generated C code this multiplication resulted in an integer value because the integers (1 or 0) from the logic operator became declared as integers in the generated C code. This issue can either be because of wrong settings or because of lack of support. However, it is possible to remove this block before the generation is performed without lack of functionality. This block is considered to behave in the same way no matter which simulation platform that is used.

3.2 PSCAD Terminology

The terminology used when working in PSCAD is introduced in this section. Mainly four different terms will be used: components, definitions, instances, and modules.

Components

A component or block is basically a graphical representation of a device. A component is the basic building block to construct models with PSCAD.

Components have inputs and outputs that can be linked together with other components to form larger systems.

Modules

A module is a combination of other components with their own canvas. In contrast to modules, regular components normally consist of a hard-coded script. Modules can also contain other modules within their canvas.

Definitions

All components or modules are defined by a definition. Every aspect of the component or module is defined in the definition. This can include graphical appearance, connection nodes, input parameters and model code. Only one definition can exist for every unique component or module.

Instances

An instance is a graphical copy of the definition, and is normally what is seen and manipulated by the user. Each instance can have different parameter settings from other instances based on the same definition. All components and modules have a single definition, from which many instances can be created. However, any design changes to a component definition will affect all instances.

3.3 Graphical Implementation in PSCAD

This section describes how the VSC-HVDC system is constructed graphically in PSCAD.

3.3.1 The Average Value Model

The converter in the PSCAD model is modeled with an average value model (AVM). The theory of the AVM is discussed in chapter 2. The dc side current I_{dc} is calculated with (2.5). The dc side current is composed of I_c and I_L, which are calculated according to (2.1) and (2.6). The calculation of I_c and I_L in the PSCAD model is illustrated in figure 3.1. This calculation is performed within the module called subsystem shown in figure 3.2. The module called subsystem in figure 3.2 injects the currents I_c and I_L into the dc grid with opposite directions according to (2.5).

The ac side voltage is calculated according to (2.1) in the bottom left of figure 3.2. The phase voltages are generated by three single-phase voltage sources.

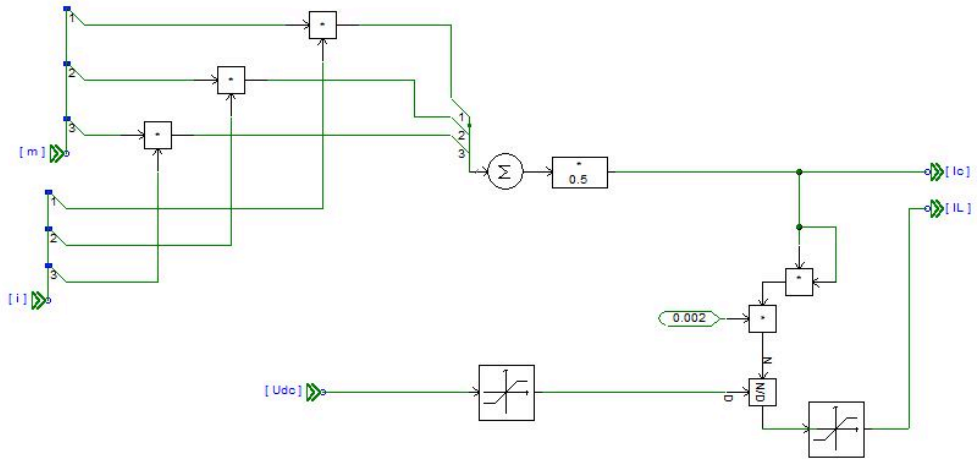


Figure 3.1: Calculation of the current I_c and the current I_L. The dc current I_{dc} is composed of I_c and I_L. The resistance that models the converter losses is set to $R = 0.002$.

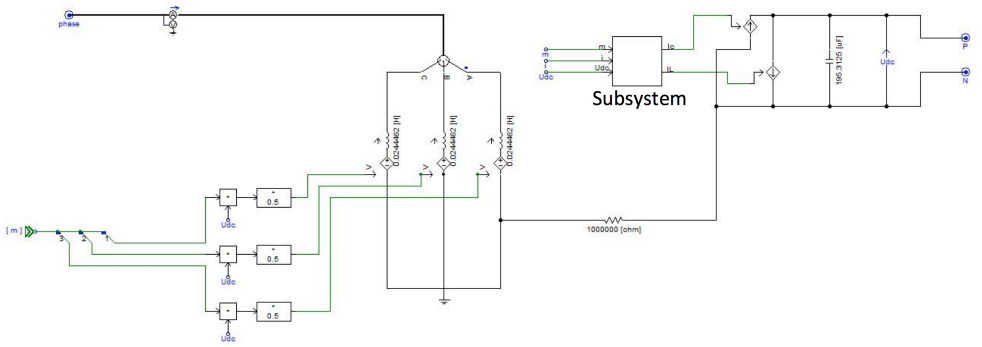


Figure 3.2: The calculation of the outputs from the module called subsystem is shown in figure 3.1. The currents I_c and I_L are injected in the dc grid with opposite directions according to (2.5). The phase voltages are calculated with (2.1) and generated by three single-phase voltage sources.

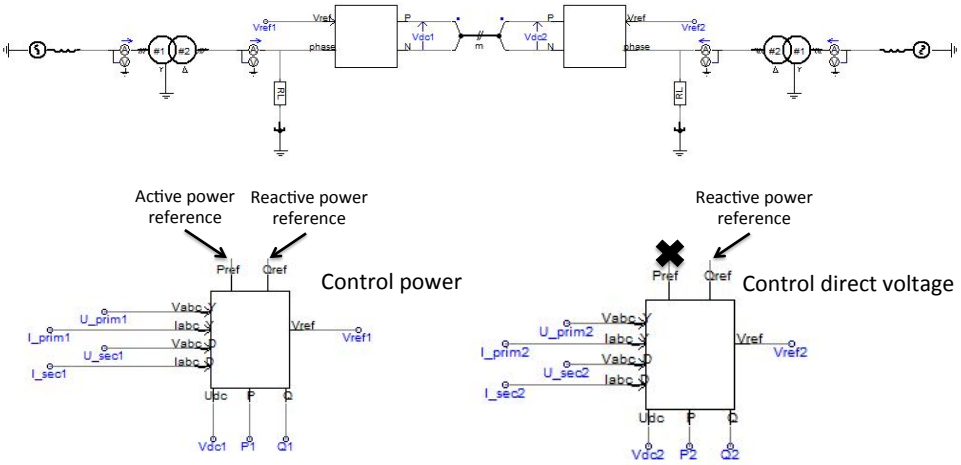


Figure 3.3: Point-to-point connection, one side controls the active power and the other side controls the direct voltage. It is possible to configure the control module in a certain mode of operation by changing its parameters accordingly. The active- and reactive power references are controlled trough inputs to the module. The direct voltage reference is set by a parameter within the module.

3.3.2 The Control Module

The control is implemented as a module in PSCAD. The module has the voltages and currents on both side of the transformer as input and the modulation index $m_i,\ i \in 1, 2, 3$ as output. It is possible to configure the module in a certain mode of operation. For example, in a point-to-point system, one instance may be configured to control the active power and another instance may be configured to control the direct voltage. This is accomplished by changing the parameters of the instances accordingly. An illustration of this is shown in figure 3.3.

The inside of the module is shown in figure 3.4. Both vector-current control and power-synchronization control are included in the control module. Next, a description of each module within the control module are described.

Signal Calculation

After the three phase quantities have been transformed to $\alpha\beta-$coordinates, the power, voltages, and currents are calculated in the signal calculation module as shown in figure 3.5. The active- and reactive power are calculated with the voltage and current in the $\alpha\beta-$coordinates. Both the active- and reactive power is filtered through a low pass filter. As is shown in figure 3.5 the ac voltage magnitude is calculated as

$$|\mathbf{v}_{\alpha\beta}| = \sqrt{v_\alpha^2 + v_\beta^2}. \tag{3.1}$$

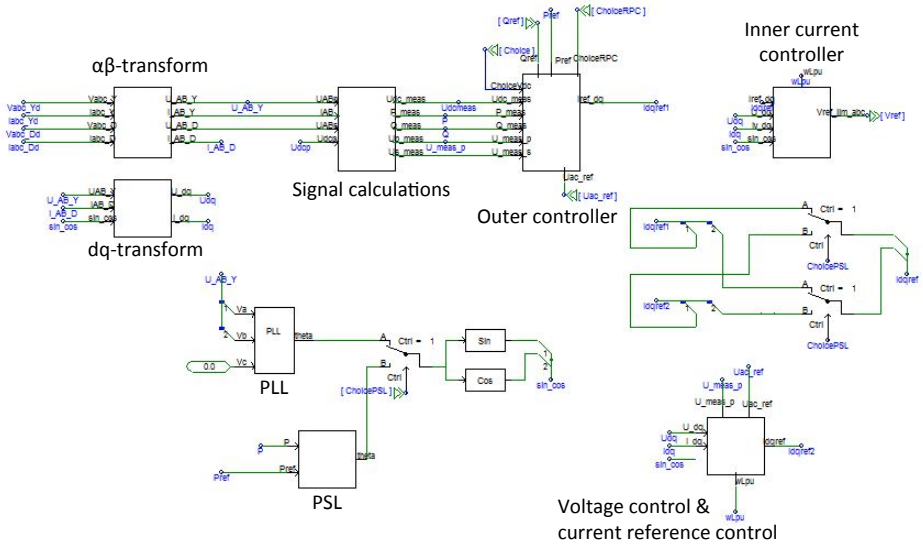


Figure 3.4: Inside of the control module, which includes an $\alpha\beta-$transform, $dq-$transform, signal calculation module, outer controller module, inner current controller module, PLL module, PSL module, voltage control and current reference control module.

Note that both the alternating and direct voltages are also filtered through a low pass filter.

Inner Current Controller

The theory of the inner current controller is detailed in chapter 2. The inside of the module is shown in figure 3.6. It includes a decoupling that is necessary according to (2.11). It also includes the two PI-controllers according to (2.14). Furthermore, it includes a limitation of the amplitude and a transformation back from the d and q axes to three phase quantities. The PI-controller is shown in figure 3.7. It includes a proportional gain k_p and an integral with gain k_i and also an anti-windup functionality.

The Outer Control Loop

As already discussed in chapter 2, the outer controller calculates a reference current to be fed to the inner current controller. The system can operate in either vector-current control mode or power-synchronization control mode. In either way, both control modes use the inner current controller. However, only if the system is set to vector-current control mode, the outer controller described in figure 3.8 is used. In power-synchronization control mode the

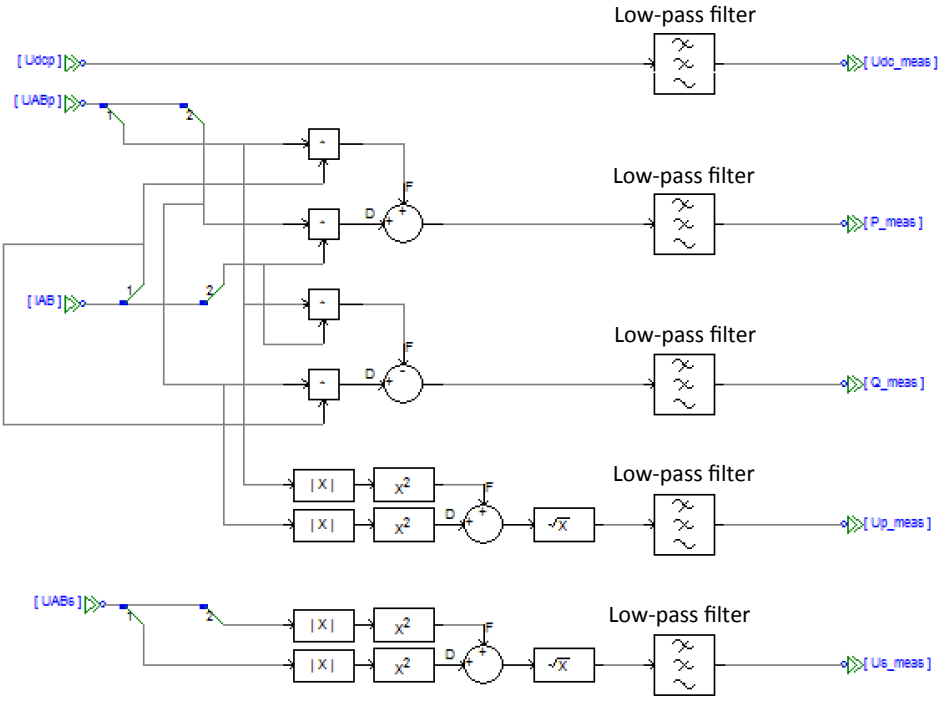


Figure 3.5: Inside of the signal calculation module. The active- and reactive power is calculated and filtered through a low pass filter. The ac voltage magnitudes are also calculated and filtered through a low pass filter. The direct voltage is filtered through a low pass filter.

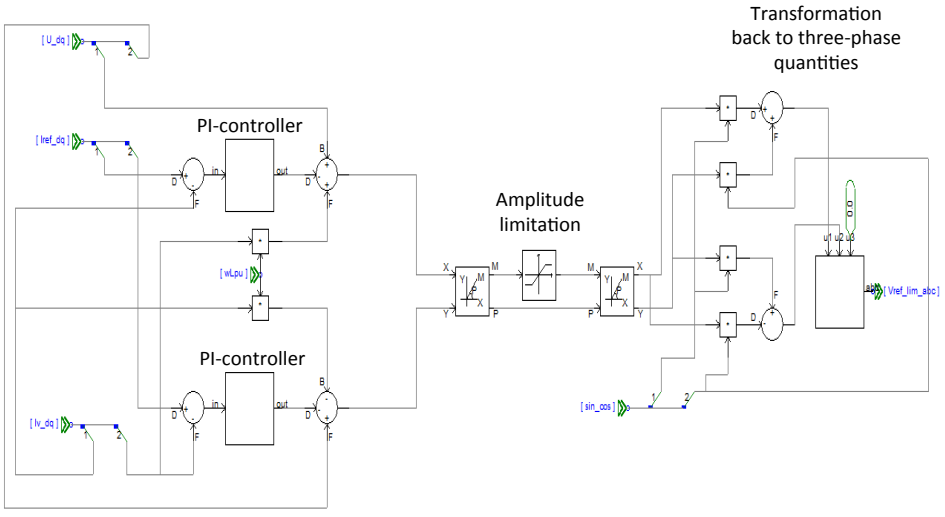


Figure 3.6: Inside of the inner current control module. The module includes a decoupling, PI-controllers, amplitude limitation, and transformation from d and q axes to three phase quantities. The PI-controller is shown in figure 3.7.

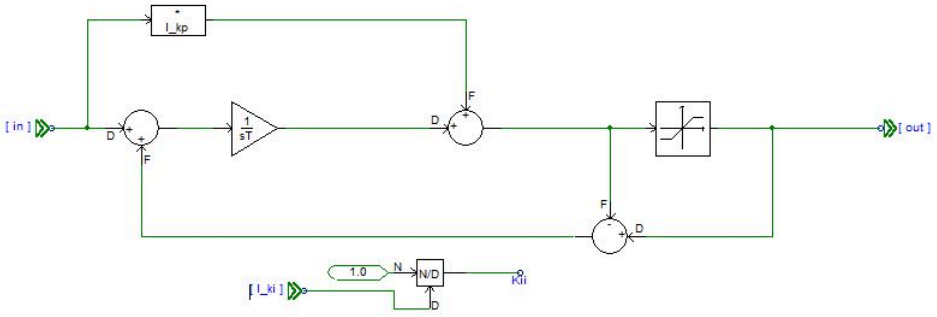


Figure 3.7: Inside of the PI-controller module. The module includes a proportional gain k_p and an integral with gain k_i. It also has an anti-windup functionality.

current reference is calculated within the current reference module described in subsequent sections.

The outer control module includes two integral controllers with feed-forward as in (2.16). The implementation of the integral controller with feed-forward is shown in figure 3.9. It also includes an anti-windup functionality. One of the two integral controllers is for the active-power control and the other one is for the reactive-power control. In addition, both the active-power control and the reactive-power control has a voltage control override that ensures an acceptable voltage level for both the ac and direct voltages. The voltage control override implementation is shown in figure 3.10.

The system can switch between active-power control and direct-voltage control. It can also switch between reactive-power control and ac voltage control. Both the direct voltage control and the ac voltage control use IP-controllers as in (2.18). The implementation of the IP-controller is shown in figure 3.11.

Furthermore, the droop control functionality, explained in chapter 2, is implemented together with the direct-voltage control. If desired, it is possible to switch of the droop control by changing the parameters of the control module accordingly.

The PLL and PSL

Depending on the mode of operation, an instance of the control module has the possibility to switch between vector-current control and power-synchronization control. The PLL is implemented by using the standard PSCAD PLL component. The PSL is implemented as shown in figure 3.12 according to (2.21). The PSL includes an integral with gain k_i and a voltage controlled oscillator (VCO).

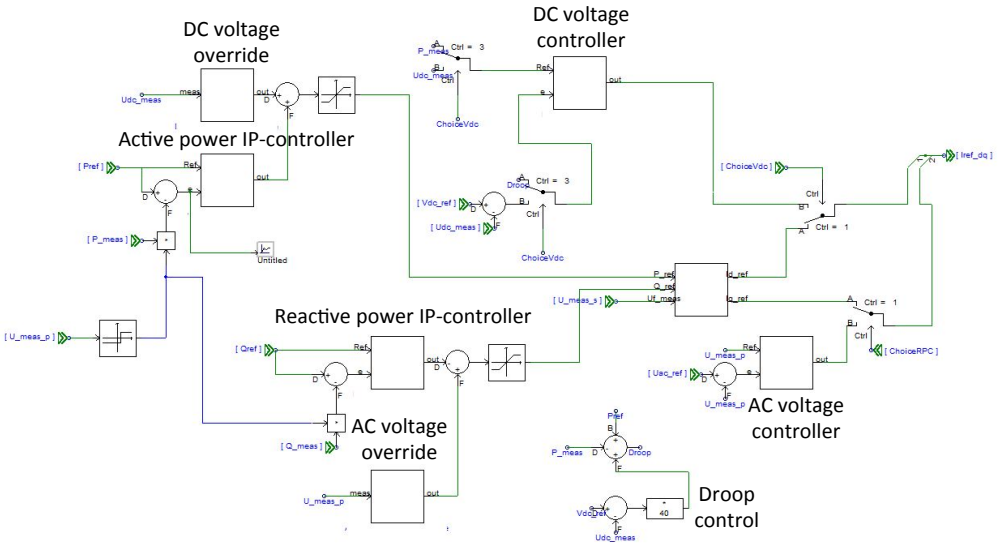


Figure 3.8: Inside of the outer control loop module. The module includes two integral controllers with feed-forward, voltage control overrides, dc- and ac voltage controllers.

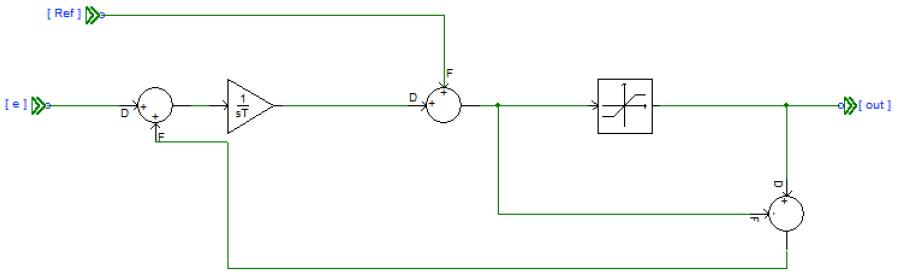


Figure 3.9: Implementation of the integral controller with feed-forward.

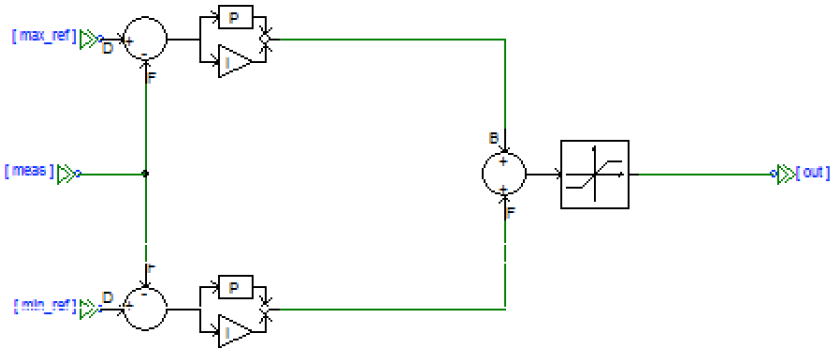


Figure 3.10: Voltage control override for the active- and reactive power. The voltage control override is not active as long as the voltage is within an acceptable level.

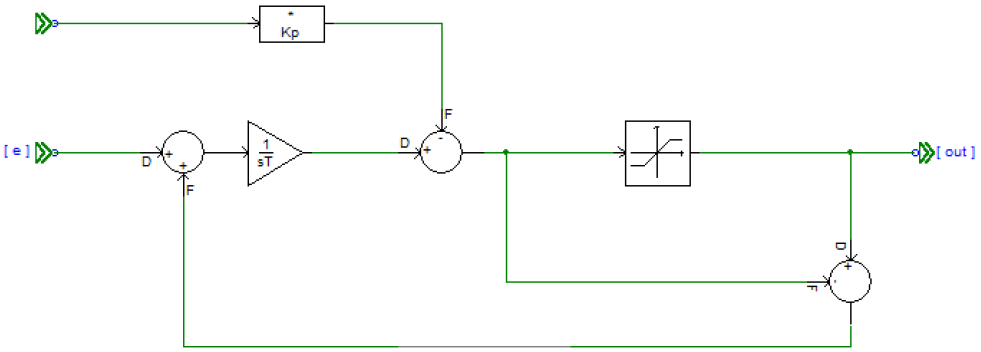


Figure 3.11: Implementation of the IP-controller.

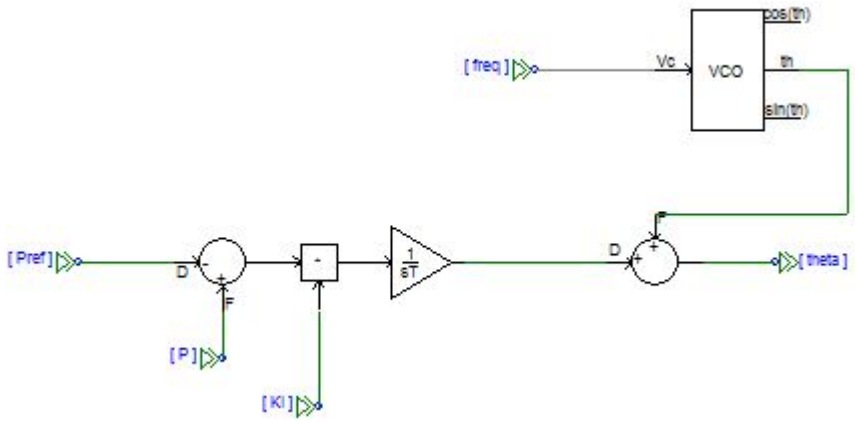


Figure 3.12: Inside of the PSL module. The PSL includes an integral with gain k_i and a voltage controlled oscillator.

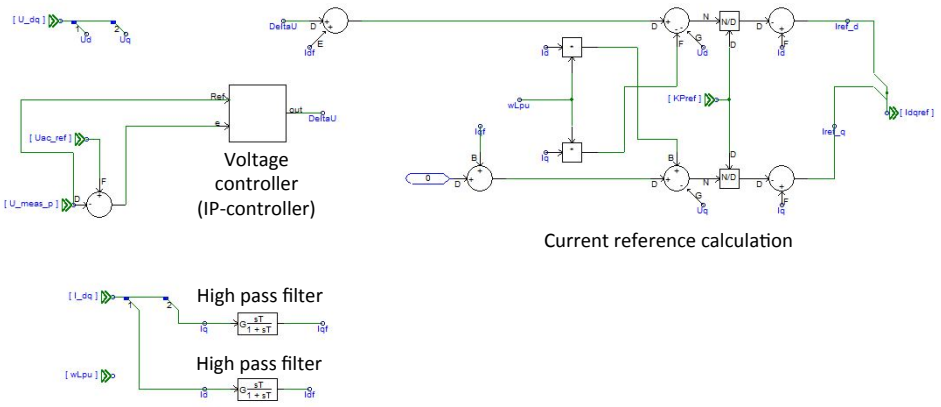


Figure 3.13: Inside of the module that calculates the current reference. The module includes a voltage controller and a calculation of the current reference.

Voltage Control and Current Reference Control

An implementation of the voltage control and current reference control used in power-synchronization control is shown in figure 3.13. The voltage control utilize an IP-controller to calculate ΔV in (2.18). It also includes a filter function and a calculation of the current reference according to (2.23).

3.3.3 Multi-Terminal VSC-HVDC System in PSCAD

In this thesis, a 9-terminal system has been built. The 9-terminal system is shown in figure 3.14. The 9-terminal system has been used for comparison of the two control strategies, vector-current control and power-synchronization control. Particularly, the control strategies have been evaluated on a HVDC link with low SCR. The results from this comparison is presented in chapter 5. The implementation of power-synchronization in a multi-terminal system is of specific interest since power-synchronization control has only been applied on point-to-point interconnections in the past [6].

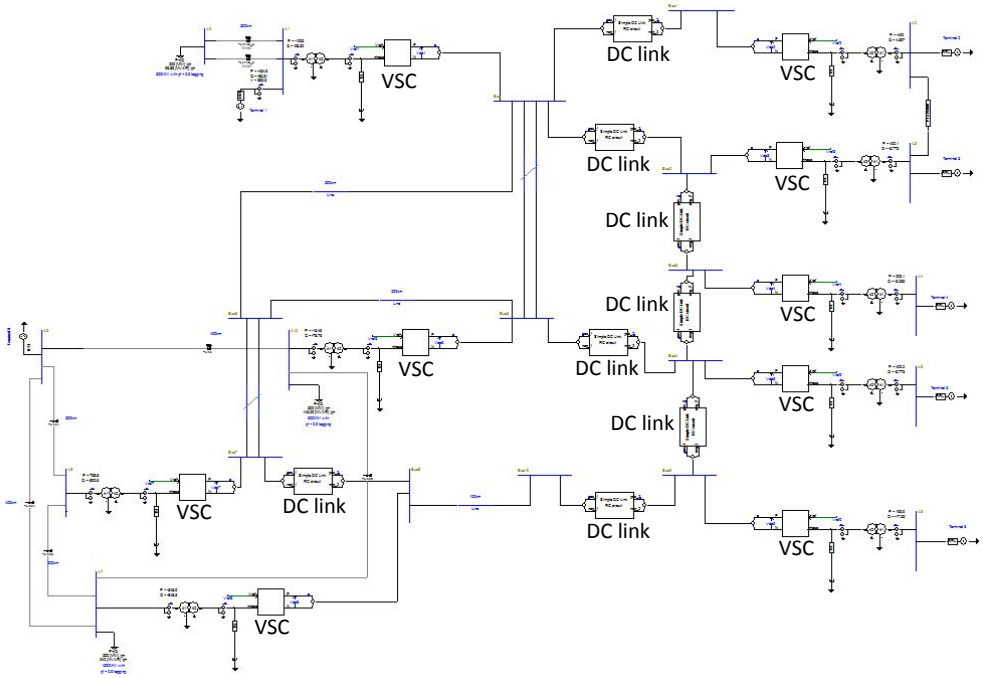


Figure 3.14: The multi-terminal VSC-HVDC system. The system consist of 9 terminals, 7 voltage sources, and 3 loads. The system will be further explained in chapter 5.

3.4 Control Implementation in PSCAD using Code Generation

This section explains an alternative way to implement the controls in PSCAD. Using this approach, the controls in PSCAD can be tailored to replicate the behavior of Simulink very well.

As explained in section 3.1 (Implementation Approach), the implementation was handled by exchanging piece-by-piece from the graphical implementation. Figure 3.15 illustrates the order of how the controllers were generated and implemented. First, the outer controller was generated, second the inner controller was generated, and finally the PLL was generated. Basically, everything in figure 3.8 was generated in the first step. In the second step, everything in figure 3.6 was generated except the limiter and transformation back to three-phase quantities. In the final step, the PLL from figure 3.4 was generated. Also, four of the low-pass filter blocks from figure 3.5 were generated on each side of the point-to-point model. Only the low-pass filters, which were in use, were generated. The side that controls the direct voltage does not need to filter the active power and the side that controls the active power does not need to filter the direct voltage. Therefore, eight low-pass filters were generated in total. The model was tested to ensure functionality between each time the code was generated and implemented.

3.4.1 Simulink C Code Generation

In computing, C is a general-purpose programming language. The controls from the generic Simulink model can be readily generated in C code. This can be achieved using the built in Simulink Coder in Matlab. The Simulink Coder generates C code from Simulink block diagrams, Stateflow charts, and Matlab functions. After the C code has been generated, it is possible to run and interact with the code outside Matlab and Simulink.

In this work, PSCAD has been used to execute the C code generated from Simulink. PSCAD executes the C code on a fixed time-step interval. Therefore, it has been suitable to use the Embedded Coder, which extends the Simulink Coder in Matlab. When the Embedded Coder is used, the Simulink model is configured so it maintains a constant (fixed) step size and also so it applies a fixed-step integration technique for computing the state derivative of the model. The fixed-step size used in this work is 20 ms. Some essential files generated from the Embedded Coder are shown in table 3.1.

As previously explained, the controller behavior depends on the values of the control parameters. It is possible to modify these values even after the controller has been generated in C code. Modifying the `ModelName_data.c` file changes the parameter values for the controller.

One of the most important files generated from Simulink is the file that contains the step function (called `ModelName.c` in table 3.1). The step

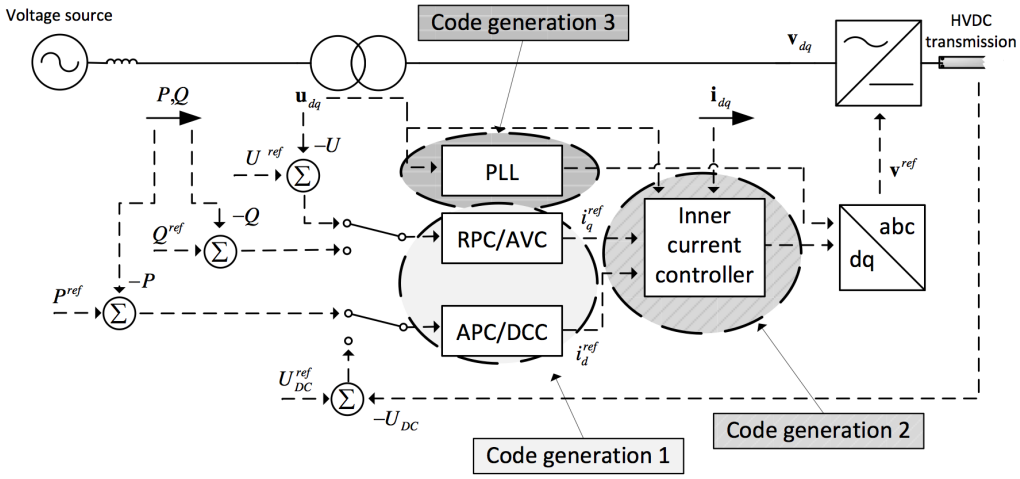


Figure 3.15: In order to facilitate debugging, the code was generated in three main steps. First, the outer controller was generated. Second, the inner controller was generated. Finally, the PLL was generated. Also the low pass filters were generated in C code, the low pass filters are not illustrated in this figure.

function should be executed each fixed time step during run-time in PSCAD.

The files generated from the Embedded Coder in Simulink contain a set of global variables along with the model step, initialize, and terminate functions. The global variables `ModelName_U` and `ModelName_Y` corresponds to the input and output structure of the model, respectively. These can be used to set the inputs prior to each execution of the step function and to receive the outputs after each execution of the step function. Therefore, by the use of a main execution file (`ert_main.c`), the inputs are set, then the step function is executed, and finally the outputs are set.

Code Generation and Execution of the Inner Control

The procedure to generate C code from Simulink is similar for any system. Therefore, the procedure is only illustrated for the inner control. This corresponds to code generation 2 from figure 3.15. The C code of the outer control loop and the inner control loop are generated and implemented separately. The inner control loop of the Simulink model is shown in figure 3.16. The inner control loop has 6 inputs and 2 outputs. The inputs are here called U_d, U_q, i_d, i_q, i_d^{ref}, and i_q^{ref}. The outputs are called d and q and represent the voltages that will be transformed to three-phase quantities and converted into line voltages by the VSC. As recently explained, the generated C code for the controller needs to be executed every 20 ms in PSCAD. Figure 3.17 illustrates how the file, which is executed every 20 ms, can look like. Values are assigned for the inputs, the step function is executed, and values are assigned for the outputs. The code in figure 3.17 needs to be manually

ert_main.c	Main file to execute step function
ModelName.c	C file that contains the controller
ModelName_data.c	C file that assigns values to data structures
ModelName.h	Header file that defines data structures
ModelName_private.h	Header file that defines data structures
ModelName_types.h	Header file that defines the model data structure
rtGetInf.c, rtGetNaN.c, rtwtypes.c, rtGetInf.h rtGetNaN.h, rtwtypes.h	Other C files and header files that are generated for the inner controller. These files serve a general purpose and can be used by other models.

Table 3.1: Files generated from the Simulink Coder.

written by the user.

3.4.2 Fortran Integration with C code

Fortran is a general-purpose programming language that is especially suited for numeric computations and scientific computing. Fortran can call existing code that is written in another language. This is commonly referred to as mixed-language programming. In mixed-language programming, a routine written in Fortran can call a function written in C code. Mixed language programming between these languages is relatively straightforward because of some key similarities between the languages.

PSCAD is a graphical front-end to EMTDC (Electromagnetic Transients including DC) for creating models and analyzing results. EMTDC solves differential equations in the time domain and calculates the solutions based on a fixed time step. The blocks in PSCAD are actually Fortran code, which call for EMTDC code library to combine them into an executable file. Running this file, runs the simulation and the results can be picked up by PSCAD.

Figure 3.18 illustrates the Fortran code that can call the C procedure in figure 3.17. In C code, the arguments are passed either by value (passing a variable) or by reference (passing a pointer to a variable). It is necessary to distinguish between these two when Fortran communicates with C. In contrast to C, all arguments are passed by reference by default when a procedure is called in Fortran. However, the code shown in figure 3.18 anyway declares the arguments as reference attributes. If an argument shall be passed as value, the arguments should be declared so in the code. Failure to do this will result in the values being treated as references and the program will show incorrect results or run-time error.

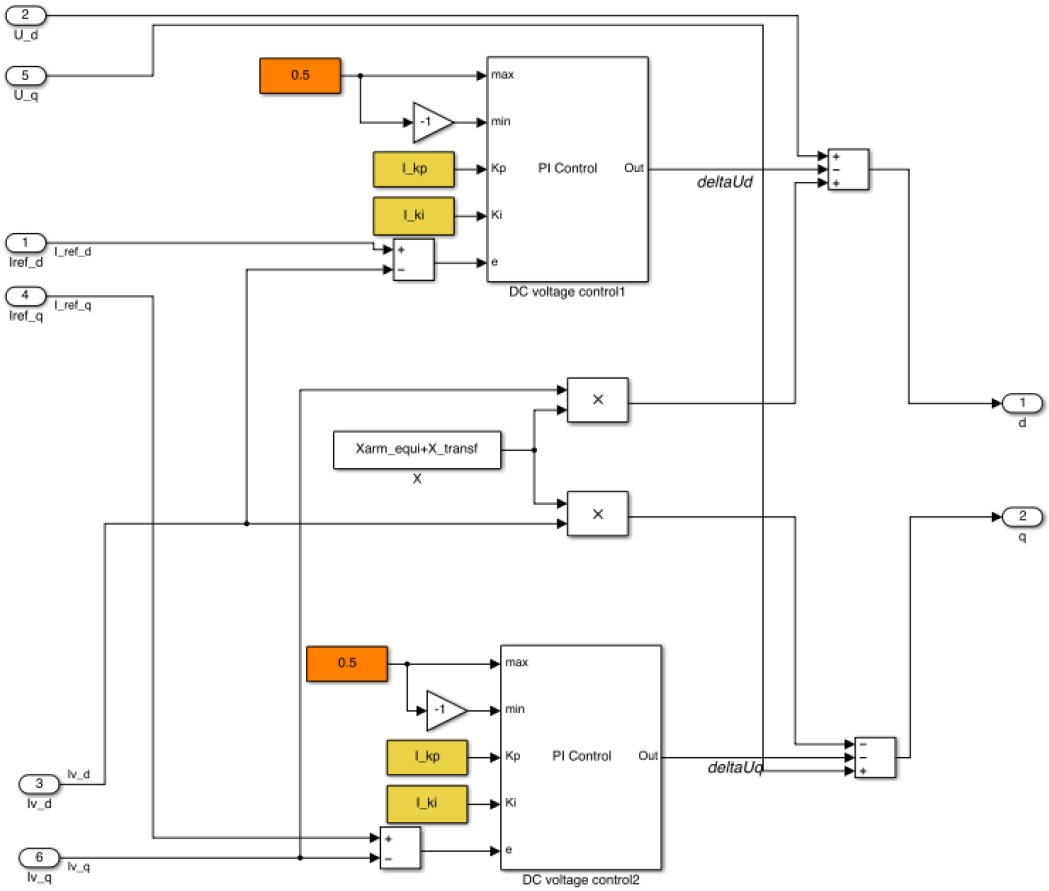


Figure 3.16: Simulink block diagram of the inner control loop. The inner control loop contains two PI-controllers for each of the i_d and i_q currents.

```
#include <stdio.h>
#include "ModelName.h"
#include "rtwtypes.h"
void rt_onestep(double* Ud, double* Uq, double* d,
                double* q, double* Ivq, double* Ivd,
                double* Id_ref,double* Iq_ref)
{
ModelName_U.I_ref_q=*Iq_ref;
ModelName_U.I_ref_d=*Id_ref;
ModelName_U.Iv_d=*Ivd;
ModelName_U.Iv_q=*Ivq;
ModelName_U.U_q=*Uq;
ModelName_U.U_d=*Ud;
ModelName_step();
*Id=ModelName_Y.d;
*Iq=ModelName_Y.q;
}
```

Figure 3.17: The main function, which is manually written in C code. The code assign values to the inputs and outputs along with the execution of the step function.

3.4.3 Linking a library to PSCAD

In PSCAD, components can be custom designed by Fortran code. The component wizard is used in order to create such component. This section describes how to create a component in PSCAD, which can be linked to the C code generated from Simulink. Therefore, the component will inherit the same functionality as the desired controller block in Simulink. The component can be created with a single line of Fortran code calling a Fortran subroutine such as in figure 3.18. The code contained in the component for calling the subroutine is listed in figure 3.19.

Additionally library (.lib) and object (.obj) files can be linked in PSCAD. Therefore, all C and Fortran files are compiled into one library file that is linked to PSCAD. The procedure to compile all files is lengthy. Therefore, this is achieved with a bash script. The following text of this chapter describes how to compile the inner and outer controller of a point-to-point model.

Unfortunately, it is not possible to use multiple instances of the components that can call a C function in PSCAD. Therefore, two inner controllers, and two outer controllers are generated from Simulink. In the subsequent text, these are called inn1, inn2, out1, and out2. Simulink automatically creates a folder named `ModelName_ert_shrlib_rtw` for each controller. ModelName here represents the controller name (e.g. inn1, inn2, out1, and out2).

```
1     SUBROUTINE FUN(Ud,Uq,d,q,Ivq,Ivd,Id_ref,Iq_ref)
2     REAL Ud,Uq,d,q,Ivq,Ivd,Id_ref,Iq_ref
3     INTERFACE
4        SUBROUTINE RT_ONESTEP (Ud,Uq,d,q,Ivq,Ivd,Id_ref,Iq_ref)
5           !DEC$ ATTRIBUTES C :: RT_ONESTEP
6           !DEC$ ATTRIBUTES REFERENCE :: Ud
7           !DEC$ ATTRIBUTES REFERENCE :: Uq
8           !DEC$ ATTRIBUTES REFERENCE :: d
9           !DEC$ ATTRIBUTES REFERENCE :: q
10          !DEC$ ATTRIBUTES REFERENCE :: Ivq
11          !DEC$ ATTRIBUTES REFERENCE :: Ivd
12          !DEC$ ATTRIBUTES REFERENCE :: Id_ref
13          !DEC$ ATTRIBUTES REFERENCE :: Iq_ref
14          REAL Ud,Uq,d,q,Ivq,Ivd,Id_ref,Iq_ref
15       END SUBROUTINE
16    END INTERFACE
17    CALL RT_ONESTEP(Ud,Uq,d,q,Ivq,Ivd,Id_ref,Iq_ref)
18    END
```

Figure 3.18: Fortran code which calls the C function in figure 3.17. Note the call of the C procedure on line 17.

```
1  CALL FUN($Ud,$Uq,$d,$q,$Ivq,$Ivd,$Id_ref,$Iq_ref)
```

Figure 3.19: Single line Fortran code in the PSCAD module which calls the subroutine in Fig 3.18.

The folder contains all C code files listed in figure 3.1. Further, the bash script is placed in a manually created folder called `interface`, placed in the same directory as automatically generated folders from Simulink. The bash script is listed in figure 3.20. The main bash script executes one additionally bash script for each controller. These additionally bash scripts are located in each of the Simulink generated folders. The bash script for controller inn1 is listed in figure 3.21. In the main bash script shown in figure 3.20, some C code files are compiled on row 25 and 26. Further, some Fortran files are compiled on row 29 and 30. The files compiled here are listed in the text files `clist.txt` and `flist.txt`, respectively. These files can be used in common by all controllers, i.e., the files listed in the bottom of table. 3.1. The Fortran files contains the Fortran code described in figure 3.18, one Fortran file is needed for each controller. Finally, everything is put together into a `control.lib` file. This is achieved on line 33 in the main bash script. The `control.lib` file can easily be linked in PSCAD. After the C and Fortran files have been linked, the modules in PSCAD will execute the corresponding C function.

For the tests performed in chapter 4, also the PLL and low pass filters have been implemented using C code. This was done in order to mimic the behavior of the Simulink model in more detail.

3.4.4 Overall Integration for the Code Implementation

The previous sections describe how to generate the C code of the controllers from Simulink and implement them in PSCAD. To illustrate the whole process, a flowchart is shown in figure 3.22. The upper left of the flowchart shows how the code is automatically generated form Simulink. The automatically generated files here are those shown in table 3.1. The `ert_main.c` file in table 3.1 needs to be manually written or modified by the user. Therefore, the `ert_main.c` file is shown in the upper right of the flowchart, along with the manually written Fortran files, which has the structure explained in figure 3.18.

The process to merge the automatically generated and manually written files into a `control.lib` is lengthy. Therefore, the files are merged together with a bash script as was shown in figure 3.20. Bottom of the flowchart illustrates how the `control.lib` file finally is linked together with PSCAD during run-time.

The procedures within the grey box in figure 3.22 needs to be manually written or modified by the user. This is the most time consuming and complicated part regarding the C code implementation. Particularly, the procedure needs to be repeated for each component that calls any C code because multiple instances are not supported. There exists a possibility to write a script that automatically writes the Fortran and C code needed for each module. However, this is considered to be out of the scope for this

```
@echo off

echo Compiling control library

cd ..

cd inn1_ert_shrlib_rtw
call makefile.bat
cd ..

cd inn2_ert_shrlib_rtw
call makefile.bat
cd ..

cd out1_ert_shrlib_rtw
call makefile.bat
cd ..

cd out2_ert_shrlib_rtw
call makefile.bat
cd ..

cd interface

cl -O2 -c -I"C:\Program Files (x86)\
Microsoft Visual Studio 9.0\VC\Include" @clist.txt
if not %errorlevel% == 0 goto error

ifort -O2 -c -free -I"C:\Program Files (x86)\
Microsoft Visual Studio 9.0\VC\Include" @flist.txt
if not %errorlevel% == 0 goto error

lib /OUT:control.lib @objlist.txt
if not %errorlevel% == 0 goto error


goto end

:error
pause
:end
```

Figure 3.20: Main batch script to compile all Fortran and C files into a common library file.

```
@echo off

echo Compiling inn1_ert_shrlib_rtw..

cl -O2 -c -I"C:\Program Files (x86)\
Microsoft Visual Studio 9.0\VC\Include" @filelist.txt
if not %errorlevel% == 0 goto error

lib /OUT:inn1.lib @objlist.txt
if not %errorlevel% == 0 goto error

goto end
:error
pause
:end
```

Figure 3.21: Additionally bash script within each of the generated folders from Simulink. The script shown is for controller inn1.

thesis. A similar script is illustrated in [23]; the script automatically writes code, which can be used with Simulink and hardware components.

The grey box can be considered as a customized interface in order to get the C code implementation to work. This interface is rather difficult to build and maintain. It also lacks generality. There exists a standardized interface to be used in computer simulations, called the functional mock-up interface (FMI) [24]. This tool is an independent standard to support both model exchange and co-simulation of dynamic models using a combination of xml files and compiled C code. With such interface there is no need for any manual or automatic script writing. Unfortunately, PSCAD lacks support for this interface. Hopefully, the future will see more of standardized interfaces that work between a broader set of simulation platforms.

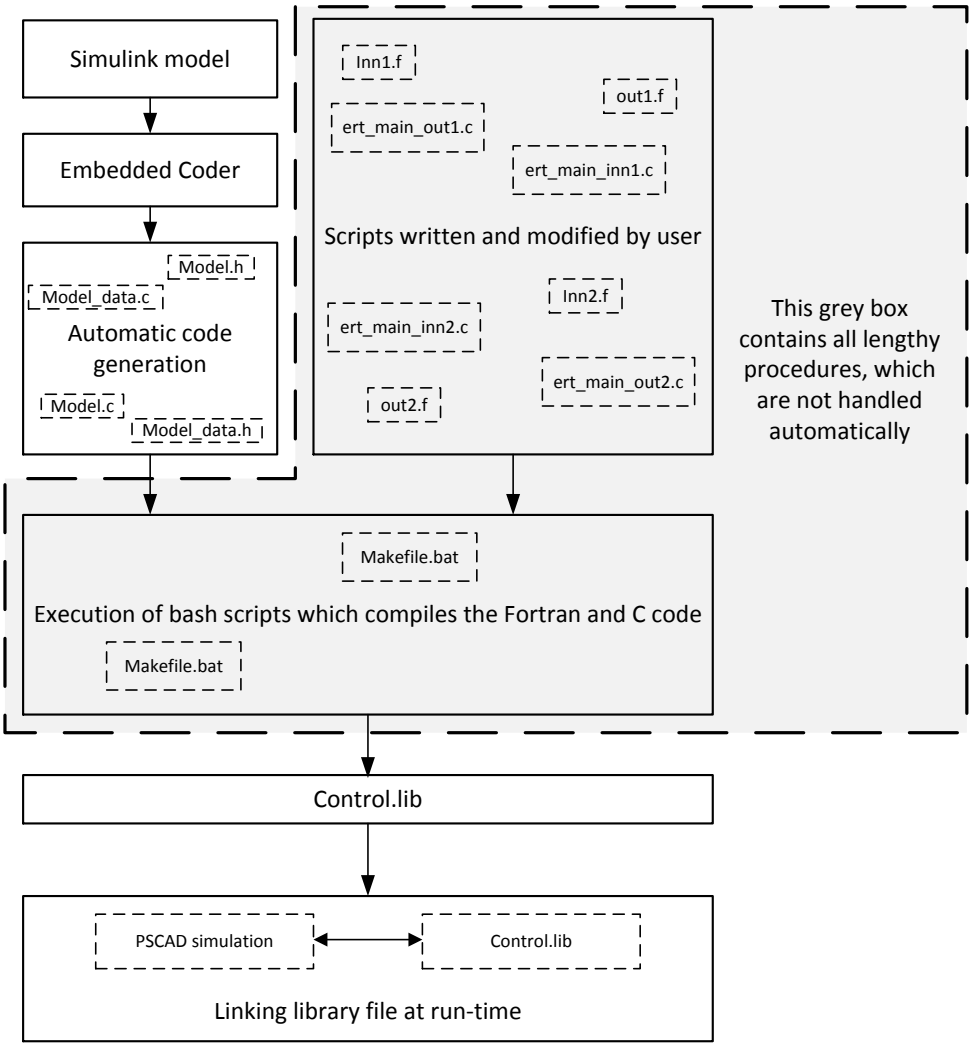


Figure 3.22: The flowchart describes how the `control.lib` file is generated and linked as a library file in PSCAD during run-time. The grey box contains all procedures (script writing and compilation), which are not handled automatically. The other boxes are more or less automated.

3.4.5 Scalability of the Proposed Implementation

Scalability is the ability of a computer application or product to function well as it is changed in volume or size [25]. In other words, scalability is the concept of a system to accommodate an increasing number of elements or objects. Therefore, scalability is often a desirable attribute for almost any system.

In a multi-terminal VSC-HVDC system, scalability becomes an important issue when the number of converters increases. In PSCAD, it is normally possible to use multiple instances of modules. Therefore, it is easy to extend a point-to-point system to any number of terminals linked together by a dc grid. A multi-terminal VSC-HVDC system can be built by a copy and paste procedure. Normally, the limiting factor will be the increase in simulation time when the system increase in complexity.

One of the most major drawbacks with the C code implementation in PSCAD is the lack of possibility to use multiple instances of modules. This constraint forces the user to go through all the relevant steps in the previous sections for each terminal. This is a rather lengthy procedure compared to the simple copy and paste procedure. How lengthy this procedure will be depends on the number of inputs and outputs of the system, because each signal needs some lines of code when the Fortran and C code scripts are made.

On the other hand, if the system in Simulink is complex, only has a few set of inputs and outputs, and only a few instances of the same module will be used. Then, this approach may be very appealing. An illustration of when the approach should be good or bad is shown in figure 3.24 and figure 3.23. The illustration in figure 3.24 is mainly because of the time it takes to write the C and Fortran code for each module. If this is done automatically, the focus will be on figure 3.23. There will be a trade-off between the number of modules, the total number of inputs and outputs, and the complexity of the system within the module.

As previously discussed, there exist no script that automatically writes the Fortran and C code for the customized interface shown in figure 3.22. Therefore, with n modules, i inputs, and j outputs, there will be $n(i+j)/2$ more steps to consider in the Fortran and C code in comparison to one module with one input and one output.

The example in section 3.4.1 illustrates how the inner controller is extracted in C code. The inner controller has in total 8 inputs and outputs while the complexity is rather simple. The inner controller consists of two PI-controllers and an additional decoupling term. Therefore, it can be built graphically rather quickly in PSCAD. The author has found that it is easier to construct such module graphically. Particularly, it will facilitate to construct modules graphically if several modules are used in the project.

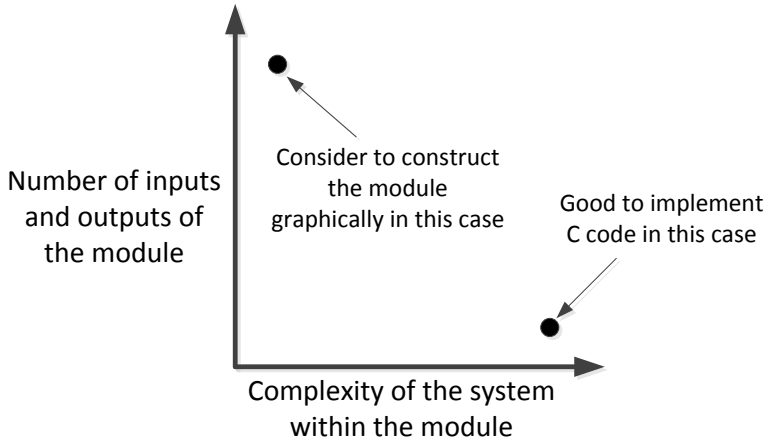


Figure 3.23: Illustration of a good and bad scenario when to use C code from Simulink to construct a module. Note that the y axis denotes the total number of inputs and outputs.

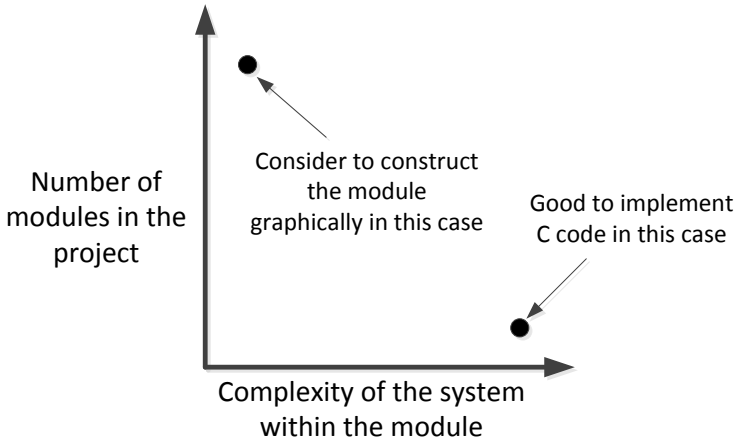


Figure 3.24: Illustration of a good and bad scenario when to use modules of C code to build up a project. In contrast to figure 3.23, note that the y axis denotes the number of modules within the whole project.

Chapter 4

Controller Performance Comparisons and Analysis

This chapter compare and analyze the different software implementations of vector-current control. The Simulink model will be considered as a reference for these comparisons. Therefore, comparisons to the Simulink model are first performed in order to ensure that deviations are not too large. Further, comparisons are performed with the manufacturer's black-box model. The controllers are compared using the eight test scenarios presented in figure 4.1. Comparisons with the Simulink model are performed on the rectifier side, while comparisons with the manufacturer's black-box model are performed on both rectifier and inverter side. The comparisons are performed on a point-to-point link.

1) Step of the active-power controller by 30%.
2) Step of the reactive-power controller by 30%.
3) Three-phase fault at the primary side of transformer during 100 ms with 10% remaining voltage.
4) Three-phase fault at the primary side of transformer during 100 ms with 30% remaining voltage.
5) Three-phase fault at the primary side of transformer during 100 ms with 70% remaining voltage.
6) Single-phase fault at the primary side of transformer during 100 ms with 10% remaining voltage.
7) Single-phase fault at the primary side of transformer during 100 ms with 30% remaining voltage.
8) Single-phase fault at the primary side of transformer during 100 ms with 70% remaining voltage.

Figure 4.1: Eight test scenarios used for controller performance comparisons.

4.1 Methodology

The comparisons are performed both graphically and numerically. This section describes the methodology of how the numerical comparisons are performed. It also describes how the system was configured during the test scenarios in figure 4.1.

4.1.1 Numerical Comparisons

For the numerical comparison, the root mean square (RMS) is used. The numerical comparisons are always performed between two software or controller implementations. It is the active- and reactive power that is compared between implementations. Therefore, the RMS value is calculated according to

$$X_{\text{RMS}} = \sqrt{\frac{1}{n}[(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2]}, \tag{4.1}$$

where x_i is the discrete measurement point at time t_i for software/controller implementation (i), and y_i is the discrete measurement point at time t_i for software/controller implementation (ii). That is, if $x(t)$ and $y(t)$ are continues functions of the active- or reactive power, then $x(t_i) = x_i$ and $y(t_i) = y_i$, where $t_1 < \ldots < t_i < \ldots < t_{n-1} < t_n$. The RMS value is calculated on an interval of 1.5 s divided into 15000 measurement points (i.e.,

$t_{i+1} - t_i = 0.0001$). The interval is spread over a duration of the particular test scenario starting 0.5 s before the test is performed. In other words, if a fault is injected at $t = 1$ s, then $t_1 = 0.5$ s and $t_{15000} = 1.9999$ s.

4.1.2 Fault Impedance and System Configuration

The generators voltage is set to $U = 275\angle 0°$ kV and they operate at 50 Hz. Each generator has an impedance of $Z_{\text{gen}} = 16.9\angle 85° \, \Omega$, which yields a short-circuit capacity of approximately $S_{\text{ac}} = 4500\angle 85°$ MVA. The HVDC link is set to 300 MW. It is possible to calculate a fault impedance Z_f that yields a remaining voltage of 10%, 30%, or 70% for each fault scenario. Let n denote the fraction of remaining voltage during fault (i.e., $n = 0.1 \Rightarrow 10\%$ remaining voltage, $n = 0.3 \Rightarrow 30\%$ remaining voltage, and $n = 0.7 \Rightarrow 70\%$ remaining voltage). The calculation become as follows.

$$I_f(Z_{\text{gen}} + Z_f) = U \Longrightarrow \begin{cases} I_f Z_{\text{gen}} = nU \\ I_f Z_f = (1-n)U \end{cases} \tag{4.2}$$

It follows that

$$\left.\begin{array}{l} Z_{\text{gen}} = \frac{|U|^2}{S_{\text{ac}}^*} \\ \frac{Z_{\text{gen}}}{Z_f} = \frac{n}{1-n} \end{array}\right\} \Longrightarrow Z_f = \frac{|U|^2}{S_{\text{ac}}^*}\frac{(n-1)}{n} = \frac{|U|^2}{|S_{\text{ac}}|}\frac{(n-1)}{n}\angle 85°, \tag{4.3}$$

where S_{ac}^* denotes the complex conjugate of S_{ac}. Using the fault impedance Z_f from equation 4.3 yields the voltage curves in appendix A. The voltage curves of each fault scenario are very similar between systems. Therefore, appendix A only demonstrates the plots from Simulink.

4.2 Controller Performance Comparisons

Before the comparison between the PSCAD model and the manufacturer's black-box model, the comparisons between the two PSCAD implementations and Simulink model are performed. The parameters used in the Simulink and PSCAD models for this comparisons are listed in table 4.1. The eight test scenarios from figure 4.1 are performed. Figure 4.2 illustrates a comparison of the active power during a three-phase fault with 10% remaining voltage. For the same test scenario, the reactive power is shown in figure 4.3. All plots from comparisons between the graphical implementation and the Simulink model are shown in appendix B, while all plots from comparisons between the C code implementation and the Simulink model are shown in appendix C. Further, table 4.2 and 4.3 lists the RMS measurements calculated according to (4.1). In addition to active- and reactive power measurements, appendix B and C also illustrates how the current $|\mathbf{i}_{dq}|$ and the reference currents i_d^{ref} and i_q^{ref} are influenced during each test scenario. Table 4.2 and 4.3 also list in which figure from appendix the power and current measurements are found.

Controller	Parameter value
Inner current loop k_p	0.4
Inner current loop k_i	85
Direct voltage control k_p	6
Direct voltage control k_i	200
Outer control loop active power k_i	15
Outer control loop reactive power k_i	15
Voltage control override ac k_p	10
Voltage control override ac k_i	3
Voltage control override dc k_p	10
Voltage control override dc k_i	3

Table 4.1: Parameters used when comparing the Simulink model to the PSCAD models.

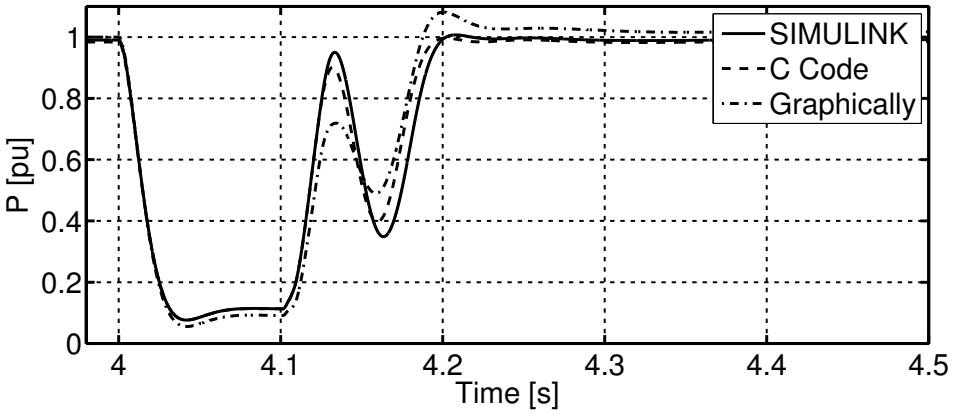


Figure 4.2: Illustration of the active power during a three-phase fault with 10% remaining voltage. This corresponds to scenario 3) from figure 4.1.

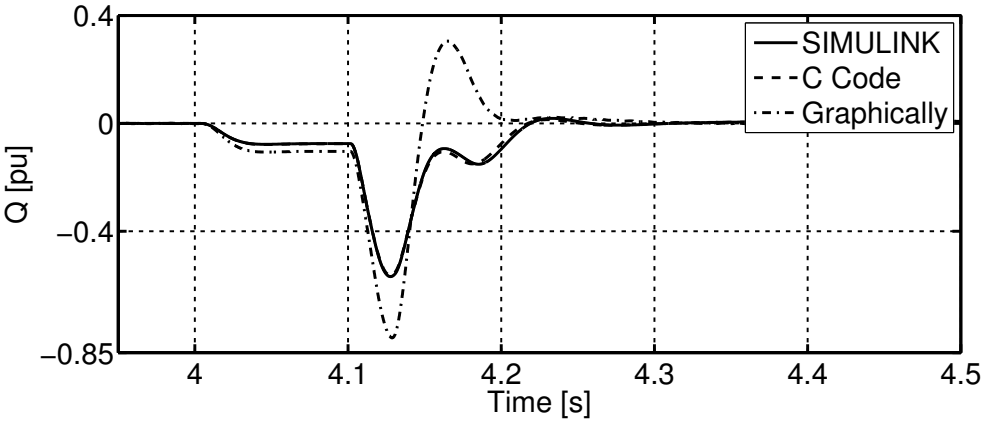


Figure 4.3: Illustration of the reactive power during a three-phase fault with 10% remaining voltage. This corresponds to scenario 3) from figure 4.1.

Test scenario	1	2	3	4
P_{RMS}	0.0030	0.0144	0.0435	0.0386
Q_{RMS}	0.0143	0.0026	0.0628	0.0573
Power measurement in figure	B.1	B.3	B.5	B.7
Current measurement in figure	B.2	B.4	B.6	B.8
Test scenario	5	6	7	8
P_{RMS}	0.0421	0.0557	0.0542	0.0208
Q_{RMS}	0.0513	0.0587	0.0555	0.0081
Power measurement in figure	B.9	B.11	B.13	B.15
Current measurement in figure	B.10	B.12	B.14	B.16

Table 4.2: RMS calculations according to (4.1). The calculations regards the graphical implementation compared to the Simulink model. The RMS values are calculated for the active- and reactive power. Also, the references to relevant figures in appendix are listed.

Test scenario	1	2	3	4
P_{RMS}	0.0076	0.0003	0.0198	0.0125
Q_{RMS}	0.0003	0.0002	0.0031	0.0020
Power measurement in figure	C.1	C.3	C.5	C.7
Current measurement in figure	C.2	C.4	C.6	C.8
Test scenario	5	6	7	8
P_{RMS}	0.0061	0.0061	0.0062	0.0063
Q_{RMS}	0.0003	0.0019	0.0011	0.0004
Power measurement in figure	C.9	C.11	C.13	C.15
Current measurement in figure	C.10	C.12	C.14	C.16

Table 4.3: RMS calculations according to (4.1). The calculations regards the C code implementation compared to the Simulink model. The RMS values are calculated for the active- and reactive power. Also, the references to relevant figures in appendix are listed.

4.3 Controller Performance Analysis and Results

To this end, no statement has been made on how well the curves match to the Simulink model for each PSCAD implementation. A glance at figure 4.2 and 4.3 gives a hint about that the C code implementation provides the best match. The conclusion from figures in appendix and from RMS calculations are that the curves match much better for the C code implementation. The means (here denoted $\overline{X}$) of all active- and reactive power RMS measurements are significantly lower for the C code implementation compared to the graphical implementation. The means for the graphical implementation are

$$\overline{P}^{\text{gr}}_{\text{RMS}} = 0.0340,$$

$$\overline{Q}^{\text{gr}}_{\text{RMS}} = 0.0388.$$

Further, the means for the C code implementation are

$$\overline{P}^{\text{cc}}_{\text{RMS}} = 0.0081,$$

$$\overline{Q}^{\text{cc}}_{\text{RMS}} = 0.0012.$$

It follows that the active power match is

$$\frac{\overline{P}^{\text{gr}}_{\text{RMS}}}{\overline{P}^{\text{cc}}_{\text{RMS}}} \approx 4$$

times better for the C code implementation. The reactive power match is

$$\frac{\overline{Q}^{\text{gr}}_{\text{RMS}}}{\overline{Q}^{\text{cc}}_{\text{RMS}}} \approx 33$$

times better. Note that these calculations are only representative for the specific test scenarios in figure 4.1, here tested on the rectifier side. Other scenarios might turn out different. In addition to the active- and reactive power measurements, the variables i_d^{ref}, i_q^{ref}, and $|\mathbf{i}_{dq}|$ have almost no difference for the C code implementation. This can be seen from the plots in appendix C. Therefore, the C code implementation is used for comparisons with the manufacturer's black-box model.

4.4 Black-Box Model Comparisons

The comparisons are performed between the C code implementation and a manufacturer's black-box model in PSCAD. The black-box protects any intellectual property invested in the design by the manufacturer. Therefore, the tests are performed without any knowledge about the control implementation within the black-box. For example, components like shunt reactors and dc link capacitors are unknown. Also, control strategies and control tuning are unknown. The manufacturer's black-box model might also take more things into consideration (e.g., it is not modeled with an AVM).

The parameters used for the C code implementation are listed in table 4.4. These are tuned in order to have a better match to the manufacturer's black-box model. All eight test scenarios from figure 4.1 are performed on the rectifier side. Further, test scenarios 3 to 8 are performed on the inverter side. Figure 4.4 illustrates a comparison of the active power during a single-phase fault with 10% remaining voltage at rectifier side. For the same test scenario, the reactive power is shown in figure 4.5. Appendix D presents all figures from comparisons.

Table 4.5 lists all RMS calculations and figure references for the rectifier side, while table 4.6 lists all RMS calculations and figure references for the inverter side. The mean value calculations from the RMS values are

$$\overline{P}_{\text{RMS}}^{\text{gr}} = 0.1078,$$

$$\overline{Q}_{\text{RMS}}^{\text{gr}} = 0.0460.$$

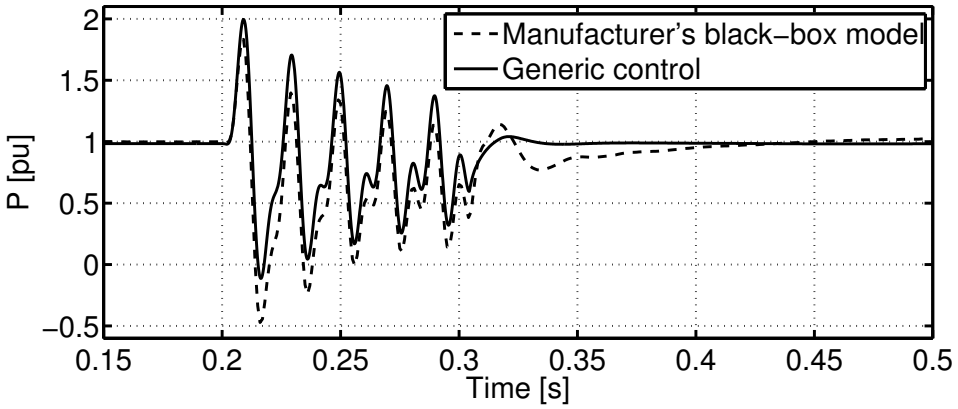


Figure 4.4: Illustration of the active power during a single-phase fault with 10% remaining voltage. This corresponds to scenario 6) from figure 4.1.

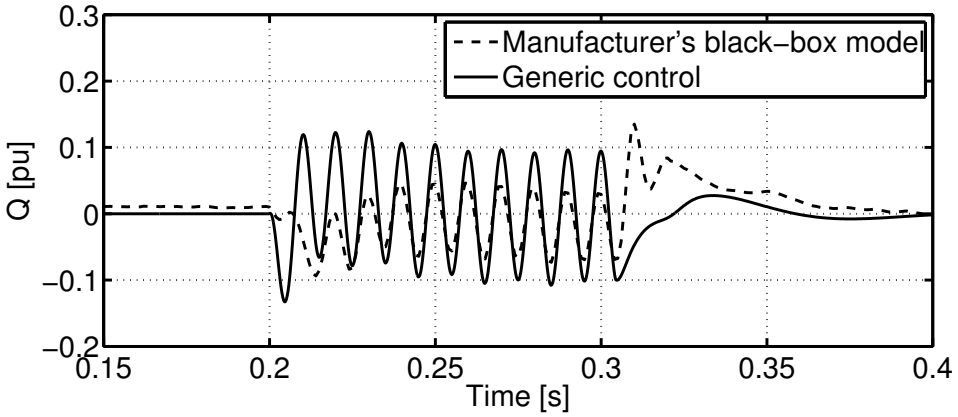


Figure 4.5: Illustration of the reactive power during a single-phase fault with 10% remaining voltage. This corresponds to scenario 6) from figure 4.1.

Controller	Parameter value
Inner current loop k_p	0.4
Inner current loop k_i	85
Direct voltage control k_p	6
Direct voltage control k_i	200
Outer control loop active power k_i	15
Outer control loop reactive power k_i	15
Voltage control override ac k_p	0.1
Voltage control override ac k_i	0.1
Voltage control override dc k_p	0.1
Voltage control override dc k_i	0.1

Table 4.4: Parameters used when the C code implementation is compared with realistic control performance results from industry.

Test scenario	1	2	3	4
P_{RMS}	0.0715	0.0028	0.1134	0.1008
Q_{RMS}	0.0110	0.0642	0.0755	0.0357
Power measurement in figure	D.1	D.2	D.3	D.4
Test scenario	5	6	7	8
P_{RMS}	0.0925	0.0719	0.0528	0.0784
Q_{RMS}	0.1048	0.0217	0.0206	0.0180
Power measurement in figure	D.5	D.6	D.7	D.8

Table 4.5: RMS calculations according to (4.1). The calculations regards the C code implementation compared to the manufacturer's black-box model. The RMS values are calculated for the active- and reactive power on the rectifier side. Also, the references to relevant figures in appendix are listed.

Test scenario	3	4	5
P_{RMS}	0.2000	0.1554	0.1308
Q_{RMS}	0.0561	0.0398	0.1254
Power measurement in figure	D.9	D.10	D.11
Test scenario	6	7	8
P_{RMS}	0.1593	0.1518	0.1274
Q_{RMS}	0.0243	0.0246	0.0217
Power measurement in figure	D.12	D.13	D.14

Table 4.6: RMS calculations according to (4.1). The calculations regards the C code implementation compared to the manufacturer's black-box model. The RMS values are calculated for the active- and reactive power on the inverter side. Also, the references to relevant figures in appendix are listed.

Chapter 5

Power-Synchronization Control Analysis

This chapter summarize the main results presented in [8]. In particular, it is demonstrated how one terminal in a dc grid is connected to a very weak ac system by using power-synchronization control. The ac system is weak because of the interconnecting lines and the conditions under which they are connected. Thus far, power-synchronization control has only been applied to point-to-point interconnections [6]. In this chapter, power-synchronization control is implemented in a multi-terminal VSC-HVDC system.

First the dc grid test system is described. Then, comparisons between control strategies are carried out.

5.0.1 The DC Grid Test System

The dc grid test system used in this work is based on the Cigre dc grid test system and therefore, it has the same topology. In this section, the dc grid test system is described. The dc grid test system used for simulations is shown in figure 5.1. A description of the components used in figure 5.1 is shown in figure 5.2. There exist nine terminals in the system, marked in the range of A to I. In this experiment, converter station A is set to both vector current control mode and power-synchronization control mode. The other converter stations are always set to vector current control mode. Simulations are performed with different references of the active-power controller for converter station A. The control mode and active-power references of all the converter stations are listed in table 5.1. Further, the line-to-line voltages and the loads connected close to each converter are listed in table 5.2. All the HVDC links have a rated power of 1000 MW and all the generators operate at 50 Hz.

As described in figure 5.2, the system has different types of power lines. Some of the power lines are modeled using the PSCAD tower transmission line models, and the others are modeled using simple pi-sections. Particularly,

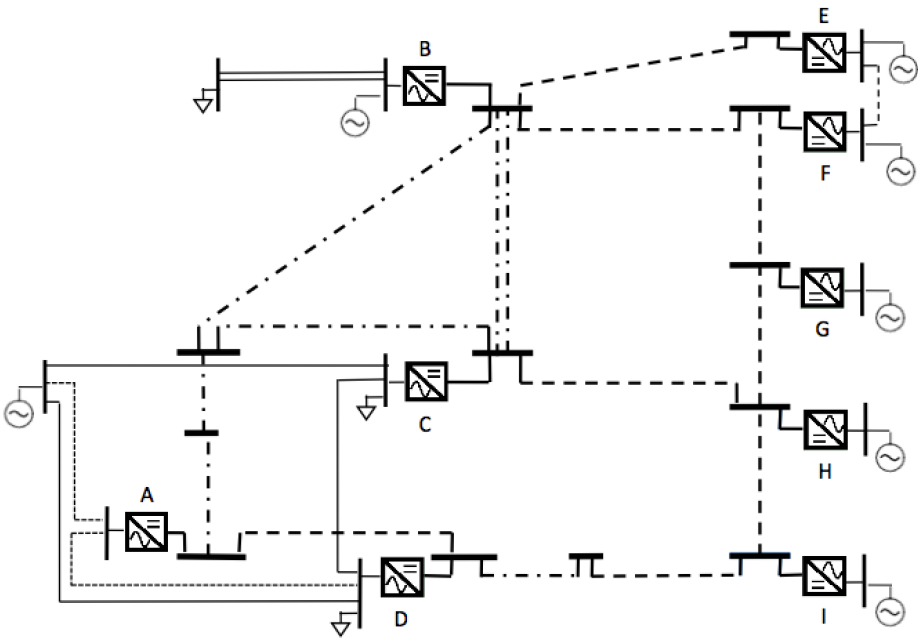


Figure 5.1: The Cigre dc grid test system used for simulations.

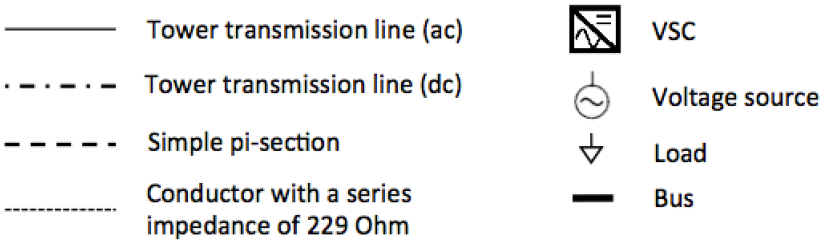


Figure 5.2: Description of the components used in figure 5.1.

converter station A is considered. Both lines that connects converter station A has a series resistance of 65 Ω and an inductance of 0.7 H. Therefore, each line has an impedance of approximately 229 Ω.

Converter station	Control mode	Active-power reference
A	Active-power	Modified by user
B	Active-power	-1 pu
C	Droop dc	0.2 pu
D	Droop dc	-0.3 pu
E	Active-power	-0.5 pu
F	Active-power	0.4 pu
G	Active-power	0.2 pu
H	Active-power	0.2 pu
I	Active-power	0.2 pu

Table 5.1: Controller data of all the converter stations

Converter station	Line-to-line voltage	Load close to converter
A	380 kV	No load
B	380 kV	600 MW, PF = 0.9 lagging
C	380 kV	900 MW, PF = 0.9 lagging
D	380 kV	1500 MW, PF = 0.9 lagging
E	155 kV	No load
F	155 kV	No load
G	155 kV	No load
H	155 kV	No load
I	155 kV	No load

Table 5.2: Voltage and load data of all the converter stations

5.1 Interconnection of a Weak AC System

As previously discussed, [4] presents a way to measure the ac system strength in terms of the short-circuit ratio (SCR). The SCR is calculated as

$$\mathrm{SCR} = \frac{S_{\mathrm{ac}}}{P_{\mathrm{dc}}}, \tag{5.1}$$

where S_{ac} is the short-circuit capacity of the ac system and P_{dc} is the rated power of the HVDC link. The strength of the ac system is from [4] defined as:

- *Strong system*, if the SCR is greater than 3.0.
- *Weak system*, if the SCR is between 2.0 and 3.0.
- *Very weak system*, if the SCR is lower than 2.0.

Consider converter station A, which has an ac bus in connection to a bus with a stiff voltage source and a bus connecting converter station D. In this section, the SCR of the ac system connecting converter station A is estimated. Furthermore, simulations during a step response of the active-power controller and during a three-phase fault are presented.

5.1.1 Estimation of the Short-Circuit Ratio

An estimation of the maximum SCR of the ac system in connection to converter station A is performed to simplify the calculations. The maximum short-circuit capacity at converter station A can be estimated as

$$S_{\mathrm{ac}}^{\mathrm{max}} = 2\frac{(380\,\mathrm{kV})^2}{229\,\Omega} \approx 1260\,\mathrm{MVA}, \tag{5.2}$$

where a multiplication by 2 is performed because there are two interconnecting lines. Thus, when the rated power of the HVDC link is 1000 MW, the maximum SCR of the ac system is not greater than

$$\mathrm{SCR} \leq \frac{S_{\mathrm{ac}}^{\mathrm{max}}}{P_{\mathrm{dc}}} = \frac{1260\,\mathrm{MVA}}{1000\,\mathrm{MW}} = 1.26. \tag{5.3}$$

5.1.2 Step Response of the Active-Power Controller

The first test of power-synchronization control is a step response of the active-power controller. Using vector current control, the authors of [5] were able to achieve a maximum power transfer of 0.4 pu when only one of the converter stations had a SCR of 1.0 in a point-to-point link. Figure 5.3 illustrates that converter station A can maintain stable operation during a step response from 0.1 pu to 0.3 pu of the active-power controller when vector current control is used. At present, the thesis author is not able to

achieve stable operation of converter station A during a step response from 0.3 pu to 0.54 pu using vector current control on converter station A.

Figure 5.4 shows that it is possible to achieve a step response from 0.3 pu to 0.7 pu using power-synchronization control in the multi-terminal VSC-HVDC system. The step response of the active-power controller is applied at $t = 0.1$ s at converter station A.

5.1.3 Three-Phase Fault at the Bus of Converter A

The second test of power-synchronization control is to apply a three-phase fault at the bus that connects converter station A to the ac grid. The active-power measurement during the fault is shown in figure 5.5. Note that converter station C and D are also influenced by the fault. The duration of the fault is 100 ms. The power transfer of converter station A is back to 0.7 pu after approximately 400 ms. The durations of time are slightly longer for converter stations C and D to return back to normal operation.

The current measurements i_d and i_q for converter stations A, C, and D during the fault are shown in figure 5.6. In order to avoid high increase in the currents, both i_d and i_q currents have a limiter.

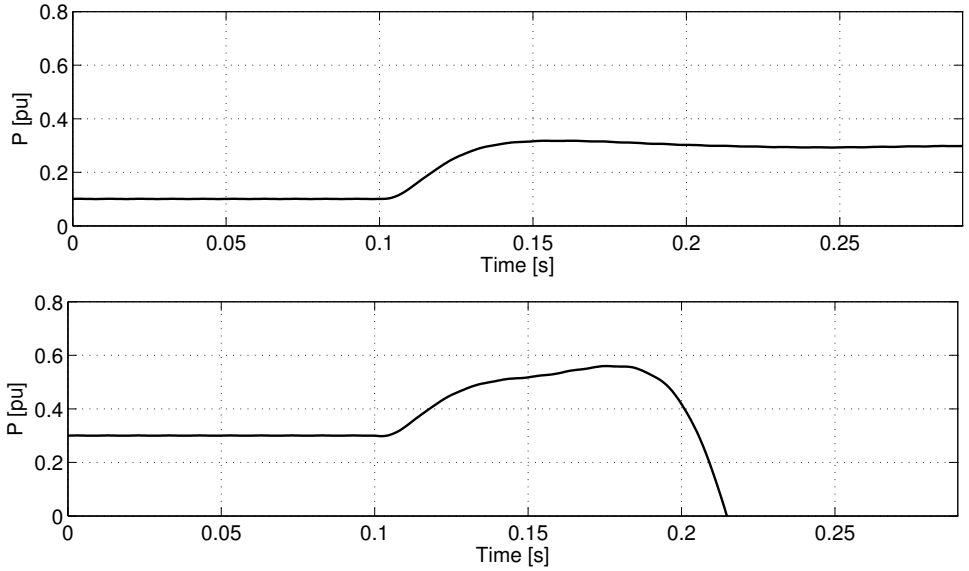


Figure 5.3: Step response of the active-power controller using vector current control. The upper figure shows a step from $P_1 = 0.1\,\text{pu}$ to $P_2 = 0.3\,\text{pu}$ at $t = 0.1\,\text{s}$. The lower figure shows a step from $P_1 = 0.3\,\text{pu}$ to $P_2 = 0.54\,\text{pu}$ at $t = 0.1\,\text{s}$. The system is not able to maintain stable operation for the latter of these step responses.

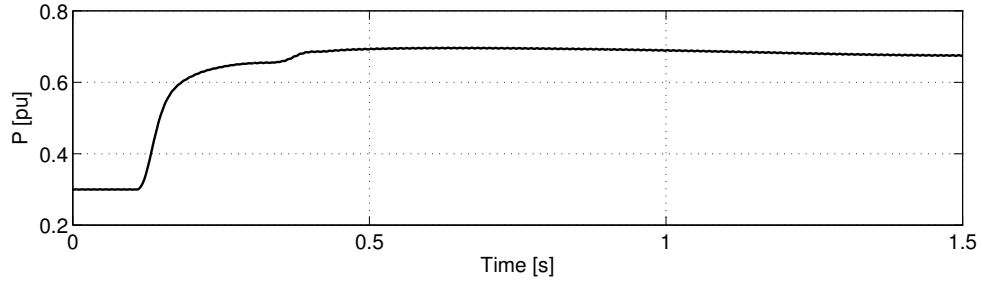


Figure 5.4: Step response of the active-power controller using power-synchronization control. The step is from $P_1 = 0.3\,\text{pu}$ to $P_2 = 0.7\,\text{pu}$ at $t = 0.1\,\text{s}$.

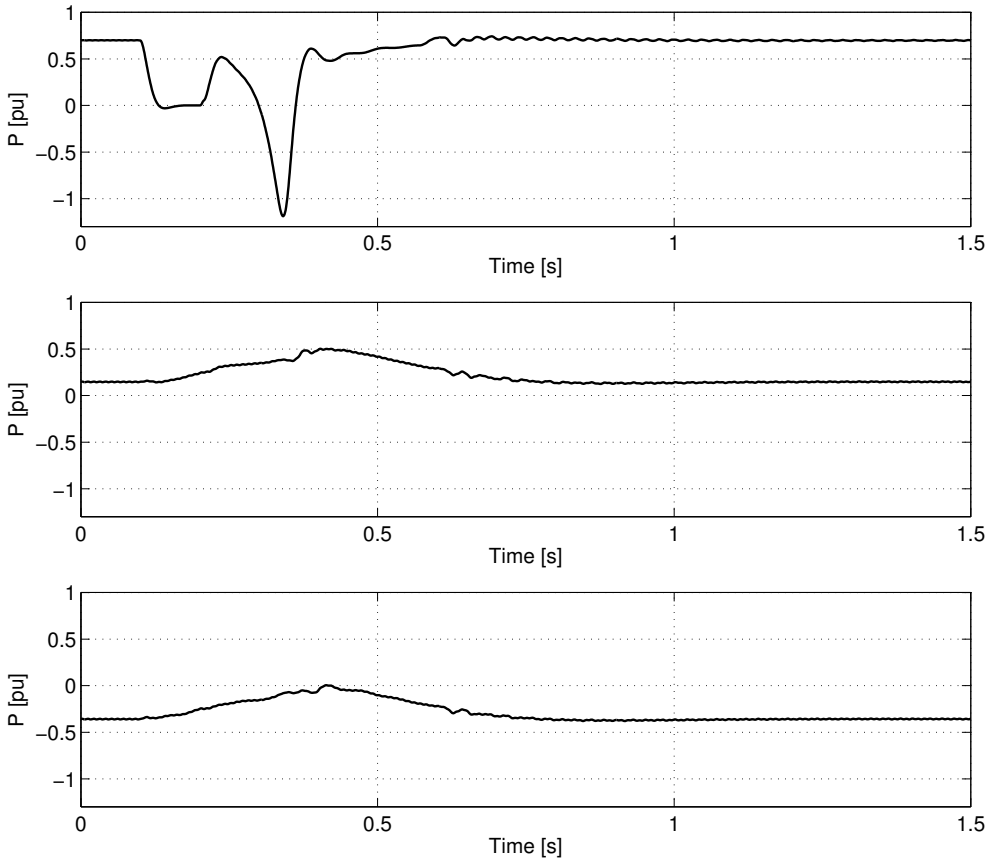


Figure 5.5: Active-power measurement at converter station A, C, and D, during a three-phase fault at the connecting bus of converter station A. The top, middle, and bottom figure show the power measurements of converter station A, C, and D, respectively. The fault is applied at $t = 0.1\,\mathrm{s}$ and has a duration of 100 ms.

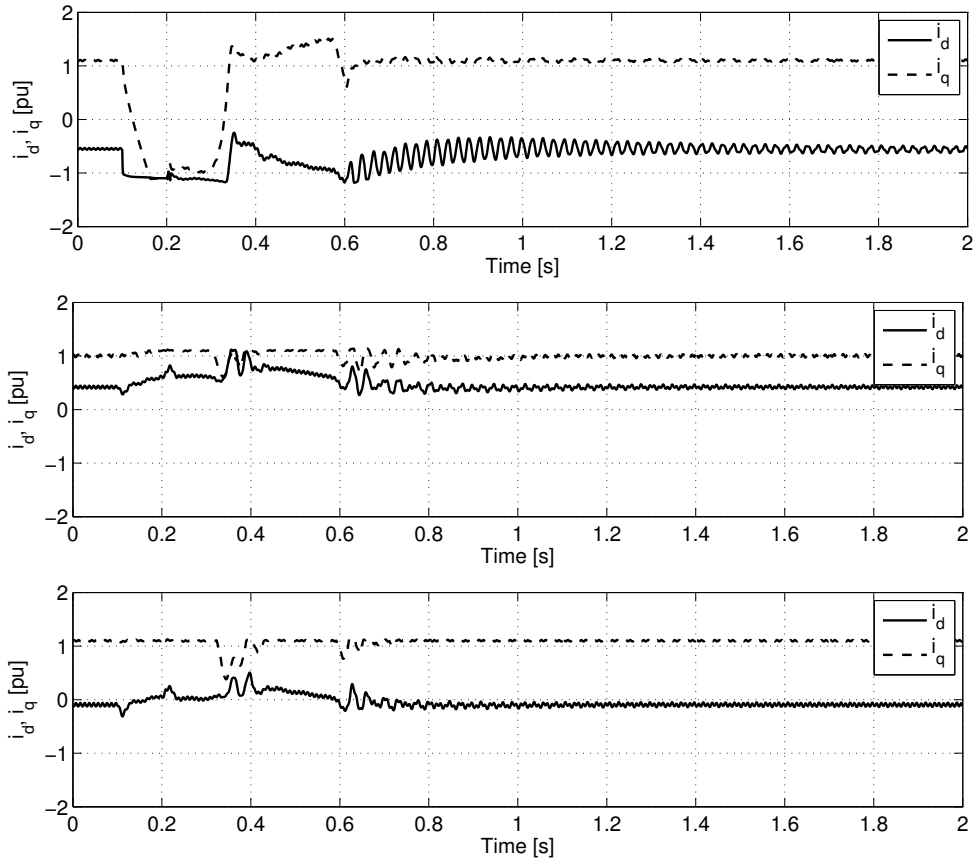


Figure 5.6: Current measurement at converter stations A, C, and D, during a three-phase fault at the connecting bus of converter station A. The top, middle, and bottom figure show the current measurements of converter stations A, C, and D, respectively. The fault is applied at $t = 0.1\,\mathrm{s}$ and has a duration of 100 ms.

Chapter 6

Conclusions and Further Work

This chapter draw some conclusions from the previous chapters. Also, some suggestions to further work are made. First, chapter 3 and chapter 4 are discussed, which mostly regards how the models were designed and the controller performance comparisons of vector-current control. Second, chapter 5 is discussed, which regards the analysis of power-synchronization control.

6.1 Model Design and Controller Comparisons

Next, some conclusions regarding the graphical implementation and the C code implementation are drawn. Also, some suggestions to further work are made.

6.1.1 Graphical Implementation

It is mostly without hardship to implement models graphically in PSCAD. A great advantage with PSCAD is the feature to use multiple instances. Once a control module and an AVM have been constructed, it is possible to copy and paste these to any number of terminals, which makes it easy to construct a multi-terminal VSC-HVDC system. One drawback with the standard PSCAD components is that it is difficult to know how they are implemented. The PLL makes up a good example on that. If a replica of the PLL in Simulink should be made in PSCAD, it is difficult to know the implementation differences. Therefore, when the model was constructed graphically in PSCAD, the model behavior did not always make a perfect match to the Simulink model. A brief summary of the pros and cons with the graphical implementation in this project follows.

Pros	Cons
- Easy to construct models this way. - Multiple instances work great when the number of terminals increase.	- Difficult to know how standard PSCAD components are implemented. - Hard to mimic the behavior of another software implementation.

6.1.2 Code Implementation

One goal for this thesis was to tailor controls in PSCAD so they replicate the behavior of controls in Simulink. First, the model was implemented graphically and several tests were performed on both the graphical implementation and the Simulink model. However, some tests showed that it was difficult to mimic the exact behavior of the Simulink model. Therefore, the controls were also implemented using C code extracted form the Simulink model. This C code was implemented in PSCAD with a nice outcome; the control behavior was very similar to the Simulink model. However, the implementation was, in comparison to the graphical implementation, rather difficult to achieve. Chapter 3 presents a customized interface that has to be manually programmed for the implementation to work. Further, PSCAD lacks functionality to implement multiple instances that runs on C code. A brief summary of the pros and cons with the C code implementation in this project follows.

Pros	Cons
- The controls can be tailored to replicate Simulink very well.	- Multiple instances does not work. - Necessary to manually program a customized interface.

6.1.3 Further Work

Chapter 3 first discusses how the model can be implemented graphically. It also discusses how the model can be implemented with C code extracted from the Simulink model. In order to get the C code implementation to work, chapter 3 described how to program a customized interface. This interface lacks generality and is rather difficult to build and maintain. The model exchange tool FMI and scripts that automatically construct the interface were also discussed in chapter 3. In this context, some suggestions to further work follow.

- Construct a script that automatically writes the code for a customized interface.
- Develop something similar to FMI or extend FMI/PSCAD features for compatibility.

6.1.4 Black-Box Model Comparison

The tests with the manufacturer's black-box model were performed without any knowledge about the control implementation within the black-box. Components like shunt reactors and dc link capacitors were unknown. Also, control strategies and the control tuning were unknown. In several test scenarios it seems like the generic control performs better than the black-box, which is considered to be unlikely. Probably, this is because of that the manufacturer's black-box model might take more things into consideration (e.g., it is not modeled with an AVM). The conclusion drawn here is that the generic control does not perform badly in comparison to other controls. Of course, more comparisons to other control implementations and other manufacturer's are needed to be sure on the correctness of this conclusion.

6.2 Power-Synchronization Control

In chapter 5, power-synchronization control was investigated in a multi-terminal VSC-HVDC system. The work demonstrated how one terminal in a dc grid was connected to a very weak ac system by using power-synchronization control. In particular, the ac system was weak because of the interconnecting lines and the conditions under which they were connected. In order to simplify the calculations, an estimation of the upper bound of the SCR was performed. In comparison to power-synchronization control, two step responses of the active-power controller were demonstrated using vector current control. Particularly, the author was not able to achieve stable operation with a power transfer of 0.54 pu using vector-current control, while power synchronization control reaches stable operation with a power transfer of 0.7 pu.

Chapter 7

Bibliography

[1] L. Zhang, L. Harnefors, and P. Rey, "Power system reliability and transfer capability improvement by VSC-HVDC (HVDC light)," in *Cigre Regional Meeting*, Tallin, Estonia, 2007.

[2] N. Flourentzou, V. Agelidis, and G. Demetriades, "VSC-Based HVDC Power Transmission Systems: An Overview," *Power Electronics, IEEE Transactions on*, vol. 24, no. 3, pp. 592–602, 2009.

[3] L. Zhang, *Modeling and Control of VSC-HVDC Links Connected to Weak AC Systems*. PhD thesis, KTH, Electrical Machines and Power Electronics, 2010. QC20100607.

[4] "IEEE Guide for Planning dc Links Terminating at ac Locations Having Low Short-Circuit Capacities," *IEEE Std 1204-1997*, pp. i–, 1997.

[5] M. Durrant, H. Werner, and K. Abbott, "Model of a VSC HVDC terminal attached to a weak ac system," in *IEEE Conference on Control Applications*, vol. 1, pp. 178–182 vol.1, 2003.

[6] L. Zhang, L. Harnefors, and H.-P. Nee, "Interconnection of Two Very Weak ac Systems by VSC-HVDC Links Using Power-Synchronization Control," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 344–355, 2011.

[7] L. Zhang and H.-P. Nee, "Multivariable feedback design of VSC-HVDC connected to weak ac systems," in *PowerTech, 2009 IEEE Bucharest*, pp. 1–8, 2009.

[8] R. Rogersten, L. Zhang, and P. Mitra, "Applying Power-Synchronization Control in a Multi-Terminal DC System," in *IEEE PES General Meeting 2014*, (Washinton DC, USA), 2014.

[9] J. Arrillaga, Y. H. Liu, N. R. Watson, and N. J. Murray, *Self-Commutating Converters for High Power Applications*. Wiley, 2009.

[10] J. Peralta, H. Saad, S. Dennetiere, J. Mahseredjian, and S. Nguefeu, "Detailed and Averaged Models for a 401-Level MMC-HVDC System," *Power Delivery, IEEE Transactions on*, vol. 27, no. 3, pp. 1501–1508, 2012.

[11] T. M. Haileselassie, *Control, Dynamics and Operation of Multi-terminal VSC-HVDC Transmission Systems*. PhD thesis, Norwegian University of Science and Technology, 2012.

[12] J. Beerten, D. Van Hertem, and R. Belmans, "VSC MTDC systems with a distributed dc voltage control - A power flow approach," in *PowerTech, 2011 IEEE Trondheim*, pp. 1–6, 2011.

[13] T. Haileselassie and K. Uhlen, "Precise control of power flow in multi-terminal VSC-HVDCs using dc voltage droop control," in *Power and Energy Society General Meeting, 2012 IEEE*, pp. 1–9, 2012.

[14] K. Weyrich, W. Kuehn, R. Leelaruji, and L. Vanfretti, "Real-Time Implementation of an Automatic Voltage Stabilizer for HVDC Control," in *IFAC 8th Power Plant and Power Systems Controls Symposium, 2012*, 2012.

[15] L. Harnefors and H.-P. Nee, "Model-Based Current Control of ac Machines Using the Internal Model Control Method," *IEEE transactions on industry applications*, vol. 34, no. 1, pp. 133–141, 1998.

[16] R. Ottersten and J. Svensson, "Vector current controlled voltage source converter-deadbeat control and saturation strategies," *Power Electronics, IEEE Transactions on*, vol. 17, no. 2, pp. 279–285, 2002.

[17] P. K. Nandam and P. Sen, "Analog and Digital Speed Control of DC Drives Using Proportional-Integral and Integral-Proportional Control Techniques," *Industrial Electronics, IEEE Transactions on*, vol. IE-34, no. 2, pp. 227–233, 1987.

[18] L. Zhang, L. Harnefors, and H.-P. Nee, "Power-Synchronization Control of Grid-Connected Voltage-Source Converters," *Power Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 809–820, 2010.

[19] L. Zhang, L. Harnefors, and H.-P. Nee, "Modeling and Control of VSC-HVDC Links Connected to Island Systems," *Power Systems, IEEE Transactions on*, vol. 26, no. 2, pp. 783–793, 2011.

[20] P. Mitra, L. Zhang, and L. Harnefors, "Offshore Wind Integration to a Weak Grid by VSC-HVDC Links Using Power-Synchronization Control: A Case Study," *Power Delivery, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.

[21] L. Zhang, H.-P. Nee, and L. Harnefors, "Analysis of Stability Limitations of a VSC-HVDC Link Using Power-Synchronization Control," *Power Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 1326–1337, 2011.

[22] M. Faruque, Y. Zhang, and V. Dinavahi, "Detailed modeling of CIGRE HVDC benchmark system using PSCAD/EMTDC and PSB/SIMULINK," *Power Delivery, IEEE Transactions on*, vol. 21, pp. 378–387, Jan 2006.

[23] R. Rogersten, H. Xu, N. Ozay, U. Topcu, and R. Murray, "Synthesis and Validation of Control Software For A Vehicular Electric Power Distribution Testbed," *accepted by AIAA Journal of Aerospace Information Systems*, 2014.

[24] "FMI." `www.fmi-standard.org/`. Retrieved 17 February 2014.

[25] A. B. Bondi, "Characteristics of Scalability and Their Impact on Performance," in *Proceedings of the 2Nd International Workshop on Software and Performance*, WOSP '00, (New York, NY, USA), pp. 195–203, ACM, 2000.

Appendix A

Voltage Plots During Faults

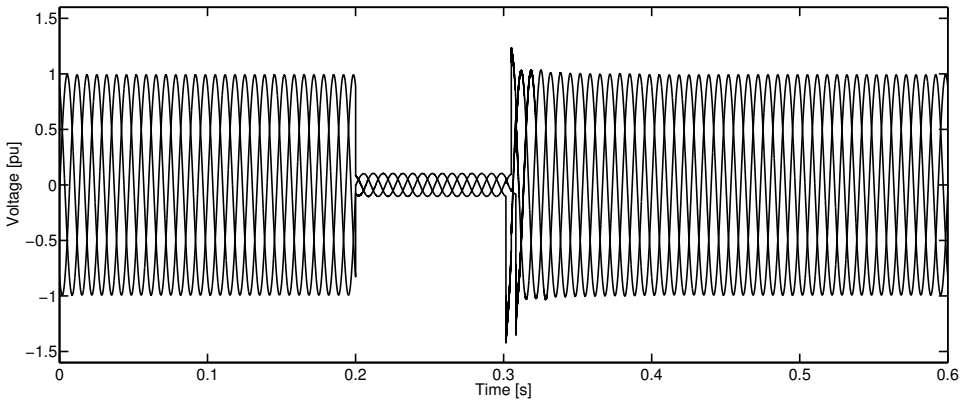


Figure A.1: Instantanous voltage mesurement during a three-phase fault at $t = 0.2\,\mathrm{s}$ with 10% remaining voltage. This voltage plot is from Simulink. The voltage plots from PSCAD are very similar and therefore not shown here.

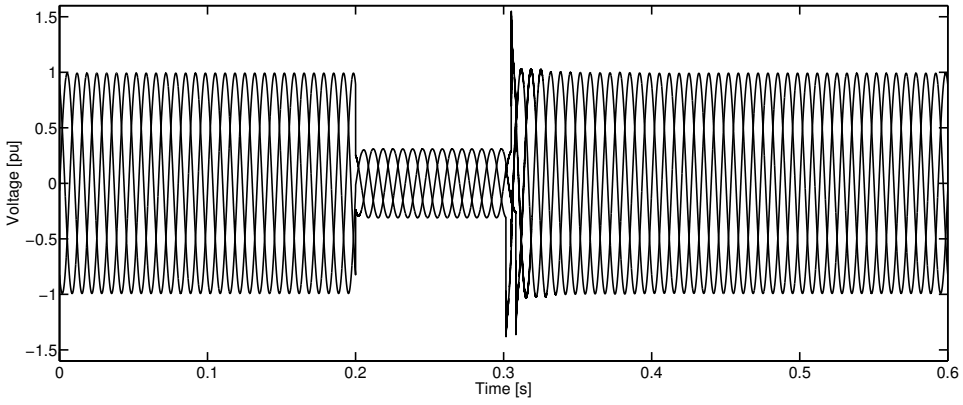


Figure A.2: Instantanous voltage mesurement during a three-phase fault at $t = 0.2\,\mathrm{s}$ with 30% remaining voltage. This voltage plot is from Simulink. The voltage plots from PSCAD are very similar and therefore not shown here.

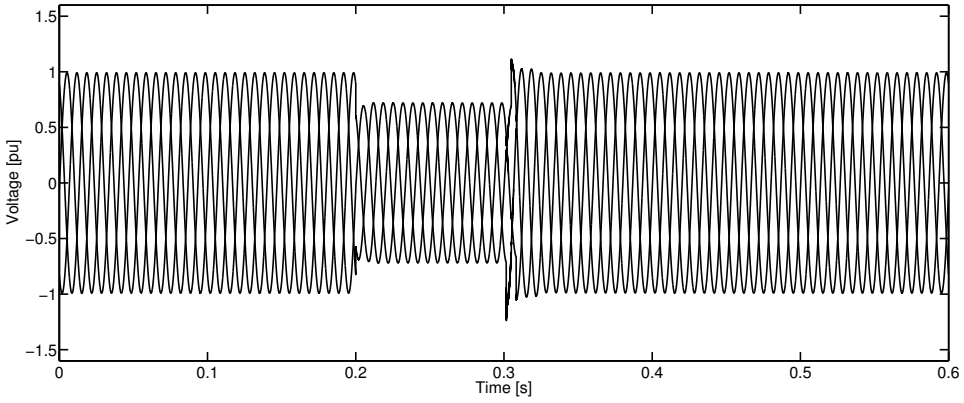


Figure A.3: Instantanous voltage mesurement during a three-phase fault at $t = 0.2\,\mathrm{s}$ with 70% remaining voltage. This voltage plot is from Simulink. The voltage plots from PSCAD are very similar and therefore not shown here.

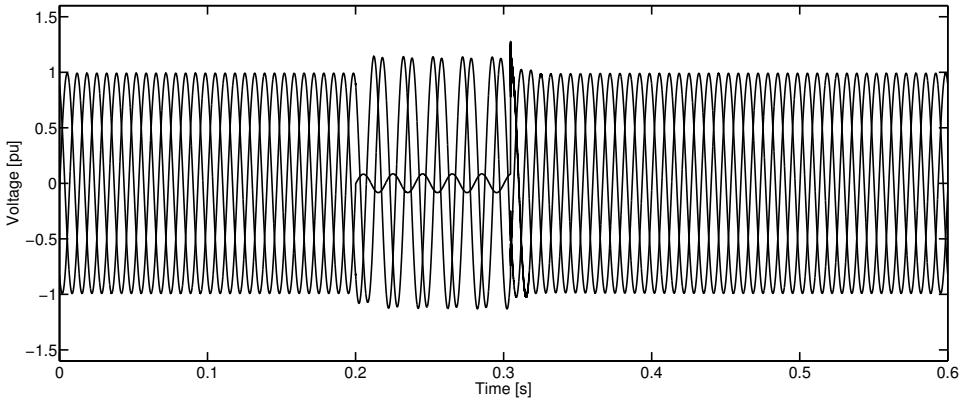


Figure A.4: Instantanous voltage mesurement during a single-phase fault at $t = 0.2\,\mathrm{s}$ with 10% remaining voltage. This voltage plot is from Simulink. The voltage plots from PSCAD are very similar and therefore not shown here.

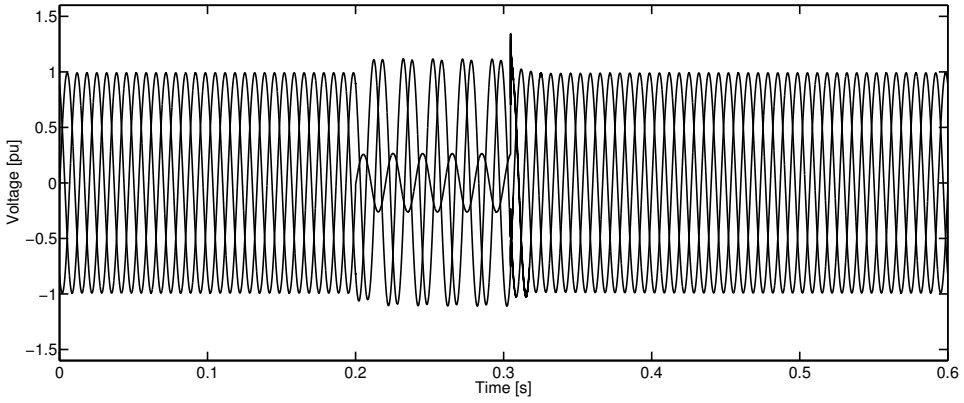


Figure A.5: Instantanous voltage mesurement during a single-phase fault at $t = 0.2\,\mathrm{s}$ with 30% remaining voltage. This voltage plot is from Simulink. The voltage plots from PSCAD are very similar and therefore not shown here.

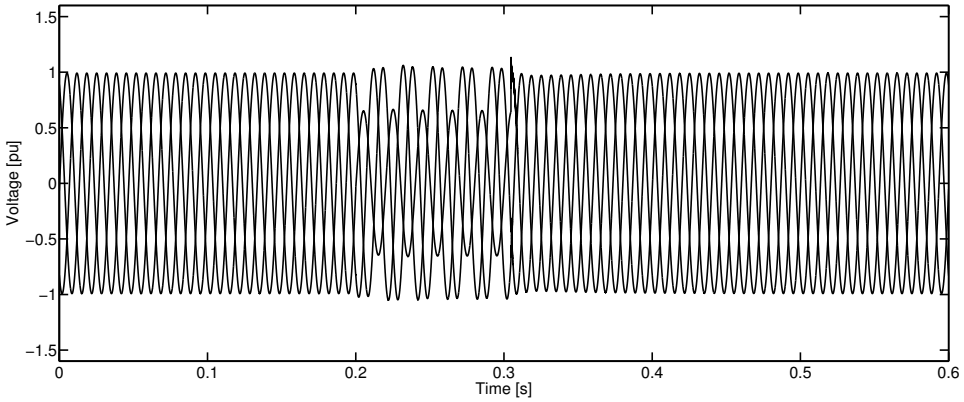


Figure A.6: Instantanous voltage mesurement during a single-phase fault at $t = 0.2\,\mathrm{s}$ with 70% remaining voltage. This voltage plot is from Simulink. The voltage plots from PSCAD are very similar and therefore not shown here.

Appendix B

Graphical Implementation

B.1 Step of the Active- and Reactive-Power Controller

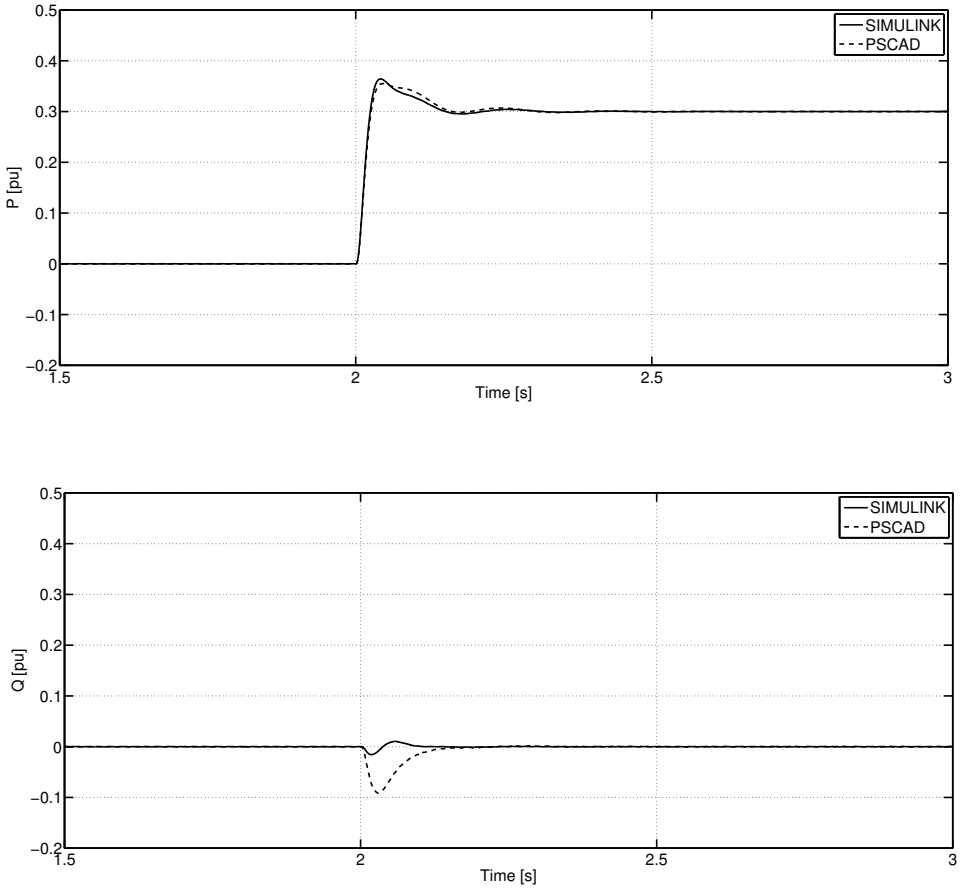


Figure B.1: Active power step change by 30% at $t = 2\,\mathrm{s}$. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

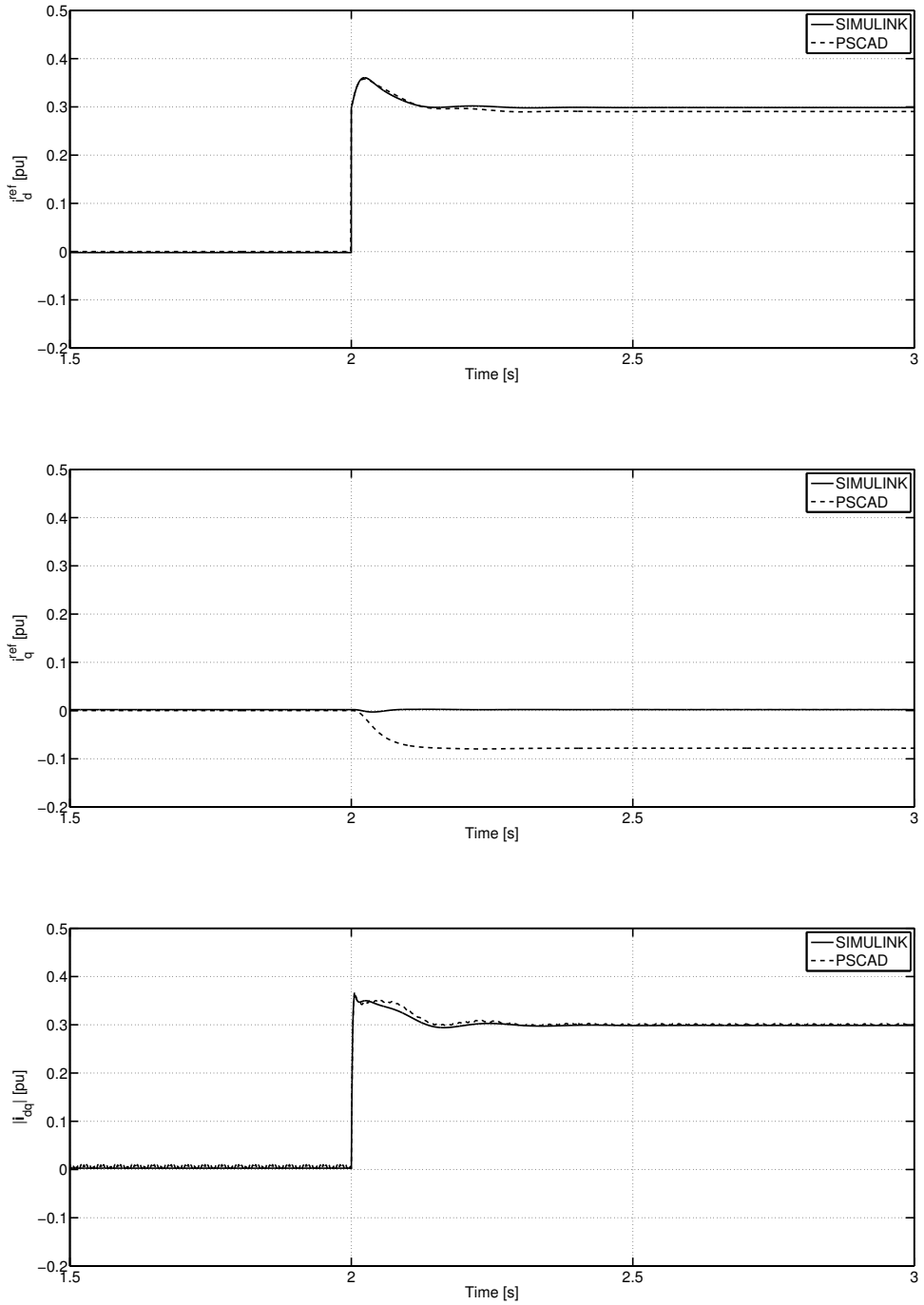


Figure B.2: The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref} during the active power step change by 30% at $t = 2\,\mathrm{s}$. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$ during the active power step change by 30% at $t = 2\,\mathrm{s}$.

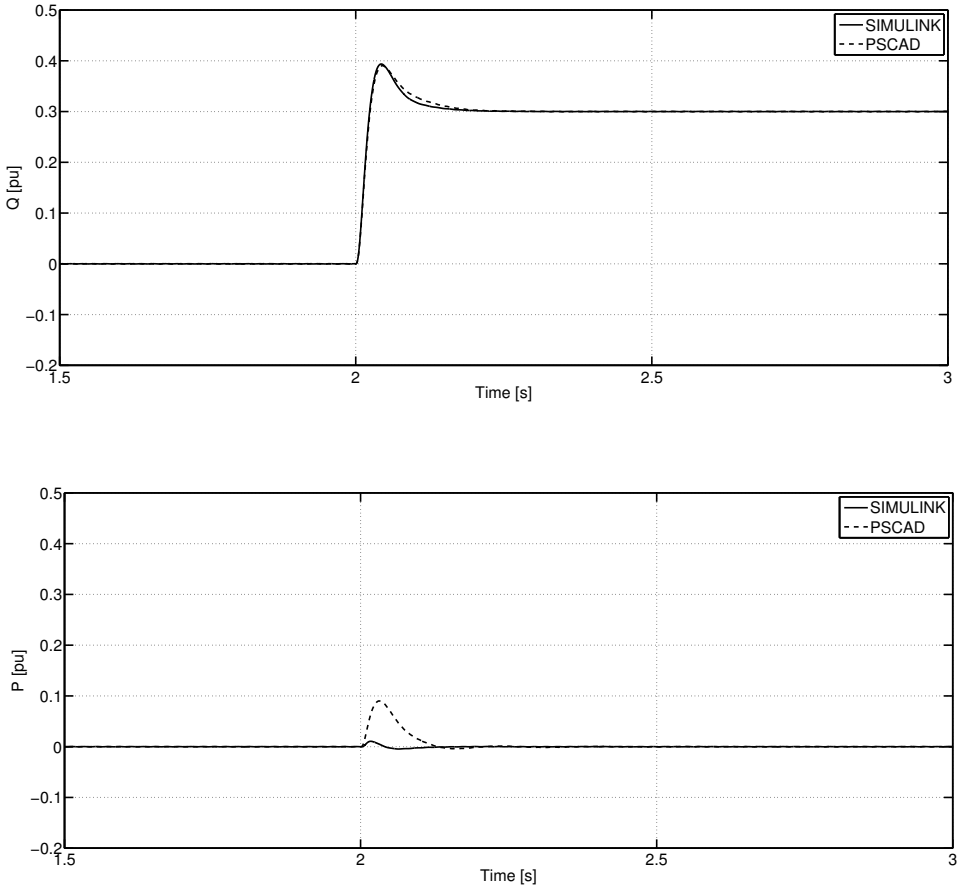


Figure B.3: Reactive power step change by 30% at $t = 2\,\mathrm{s}$. The upper figure illustrates the change in reactive power and the lower figure illustrates the change in active power due to that the system is coupled.

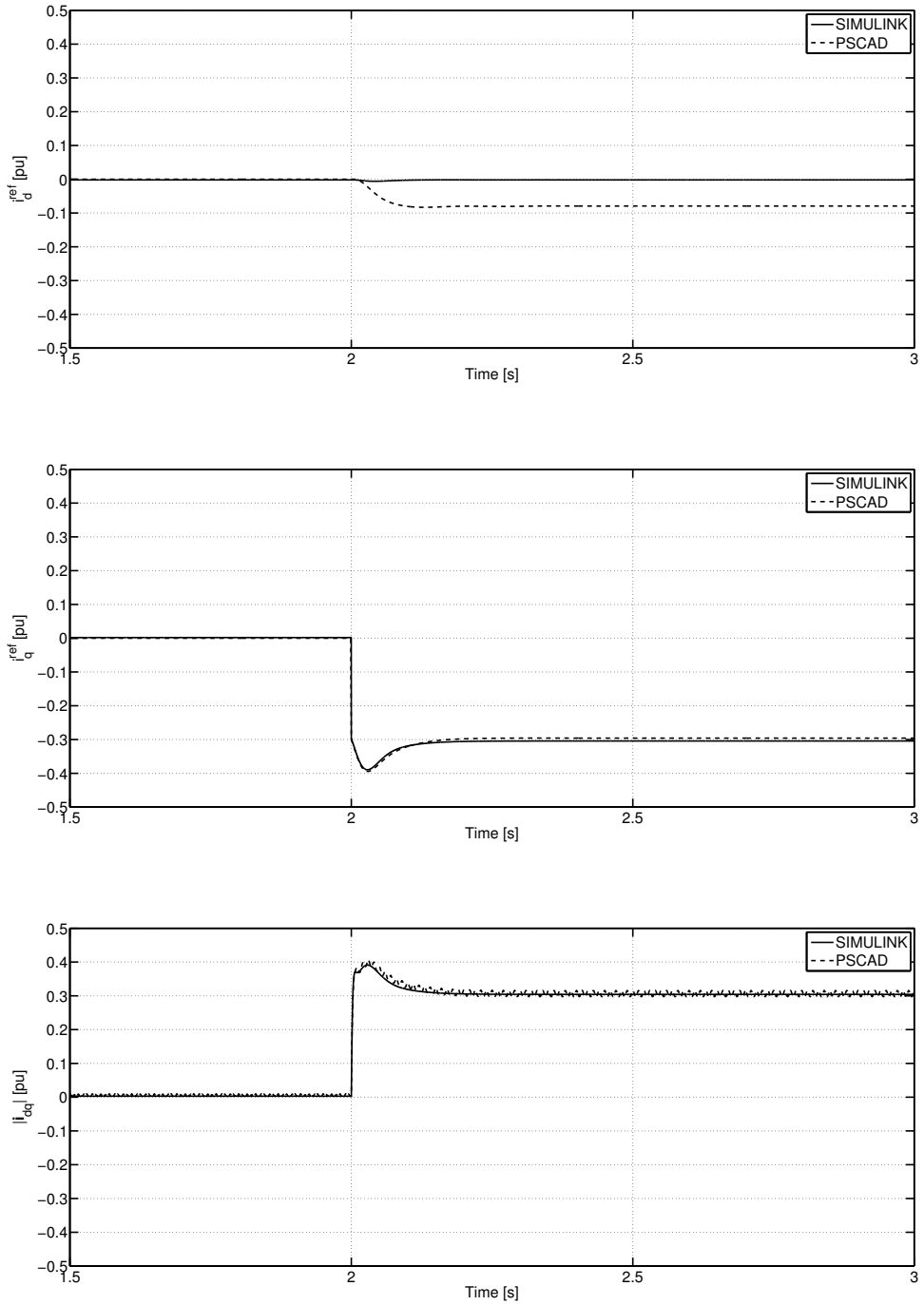


Figure B.4: The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref} during the reactive power step change by 30% at $t = 2\,\mathrm{s}$. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$ during the reactive power step change by 30% at $t = 2\,\mathrm{s}$.

B.2 Three-Phase Faults (Fault Scenarios 3 to 5)

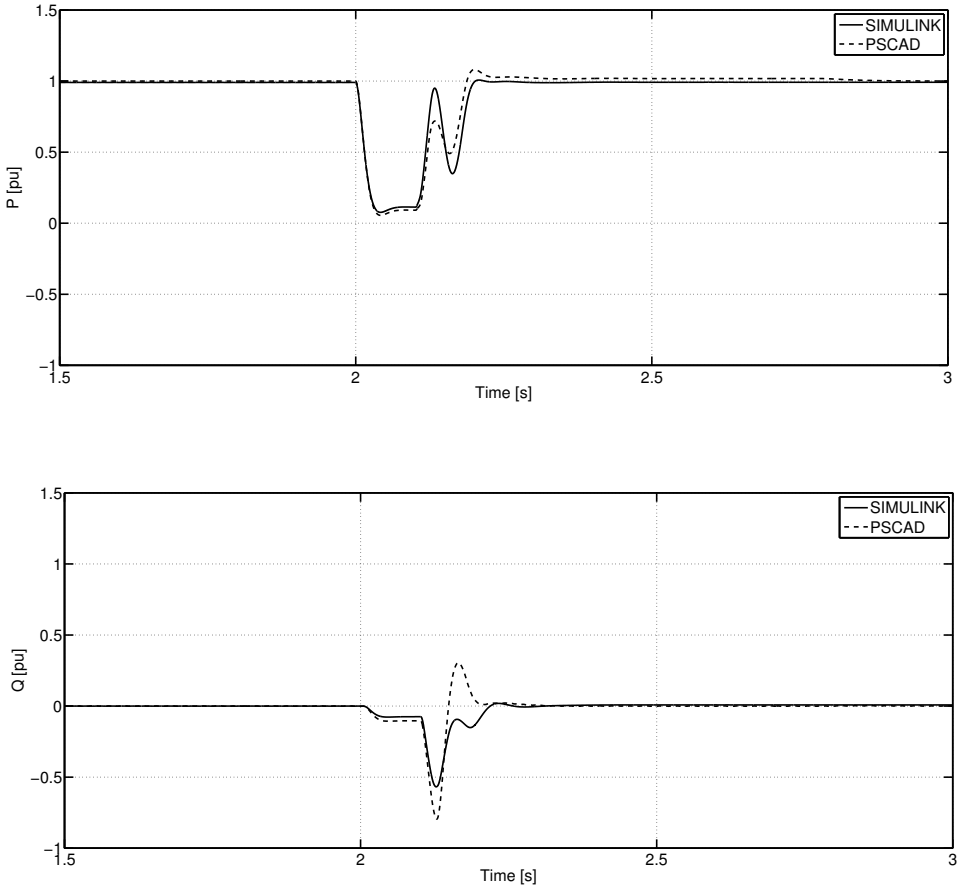


Figure B.5: Three-phase fault at $t = 2\,\mathrm{s}$ with 10% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

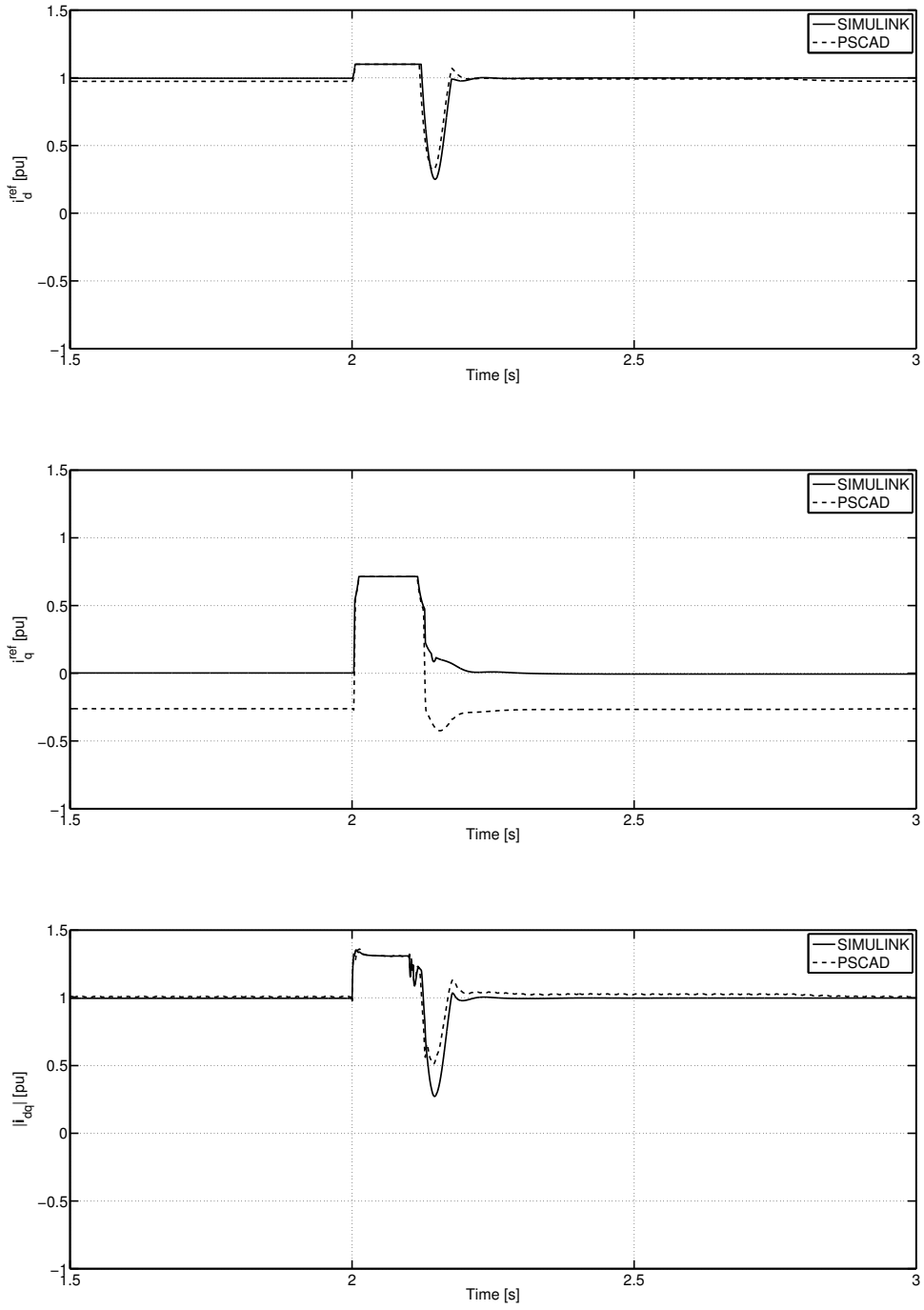


Figure B.6: Three-phase fault at $t = 2\,\mathrm{s}$ with 10% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

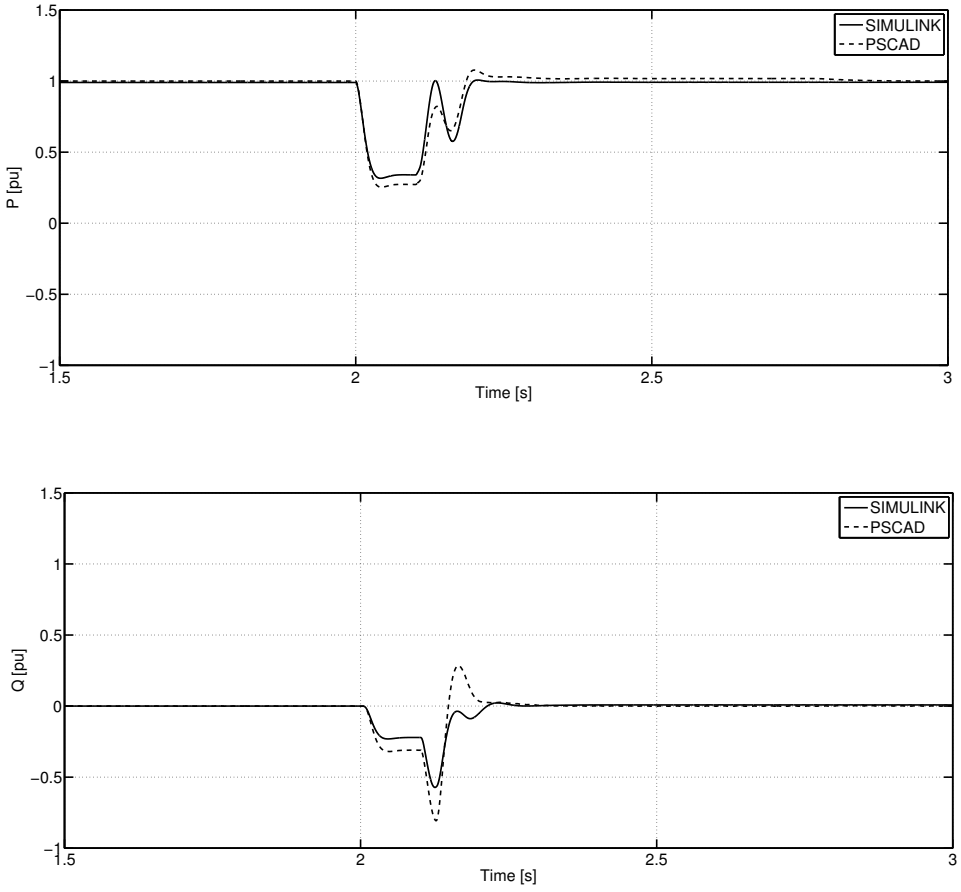


Figure B.7: Three-phase fault at $t = 2\,\mathrm{s}$ with 30% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

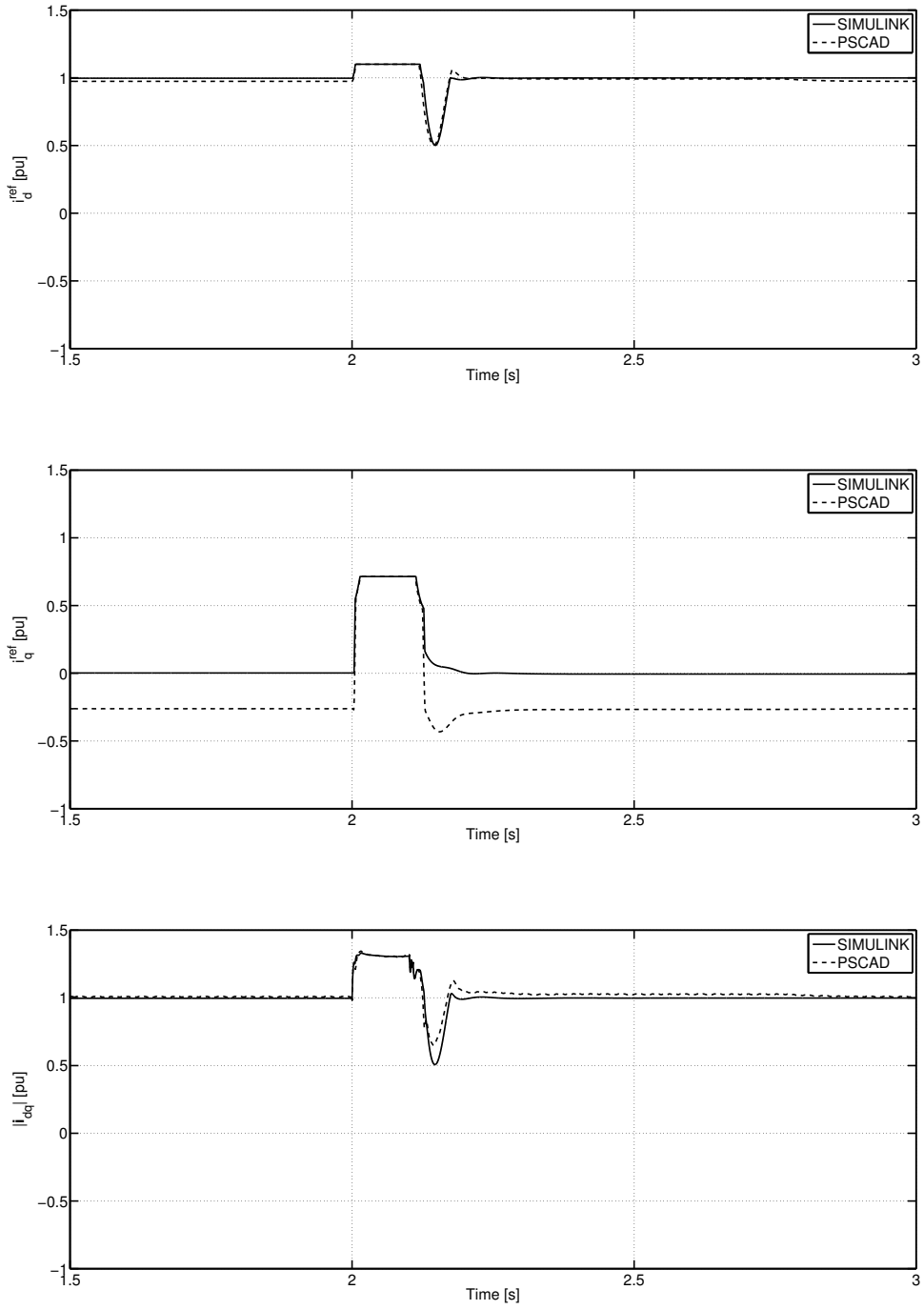


Figure B.8: Three-phase fault at $t = 2\,\mathrm{s}$ with 30% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

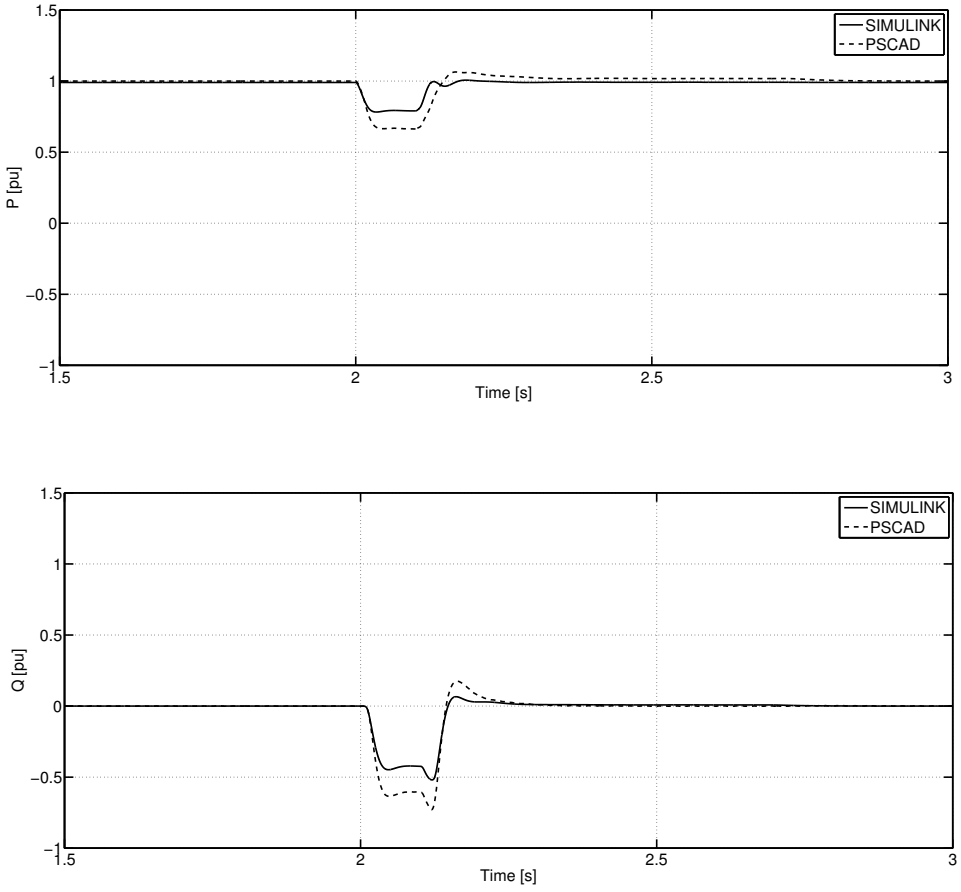


Figure B.9: Three-phase fault at $t = 2\,\mathrm{s}$ with 70% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

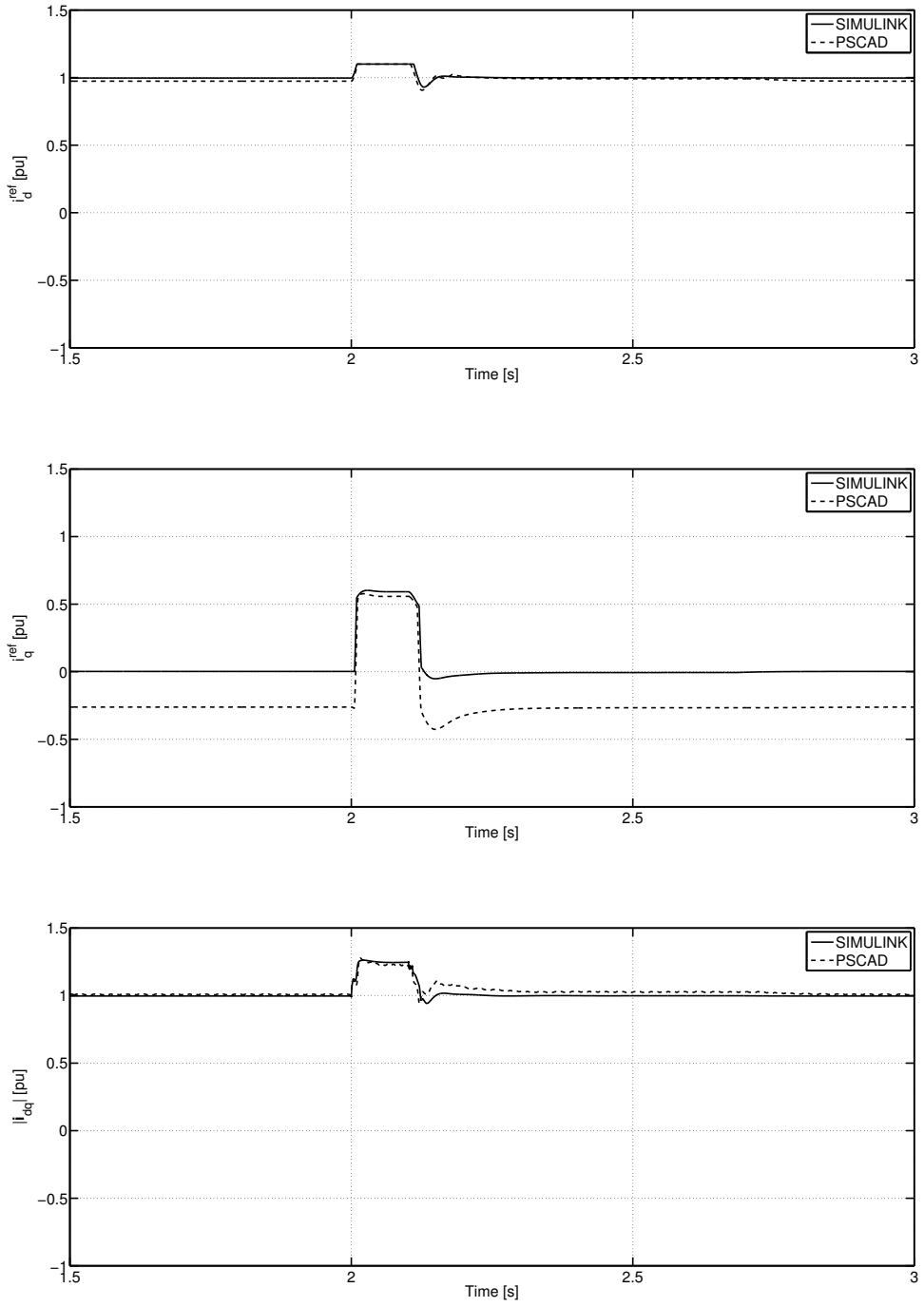


Figure B.10: Three-phase fault at $t = 2\,\mathrm{s}$ with 70% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

B.3 Single-Phase Faults (Fault Scenarios 6 to 8)

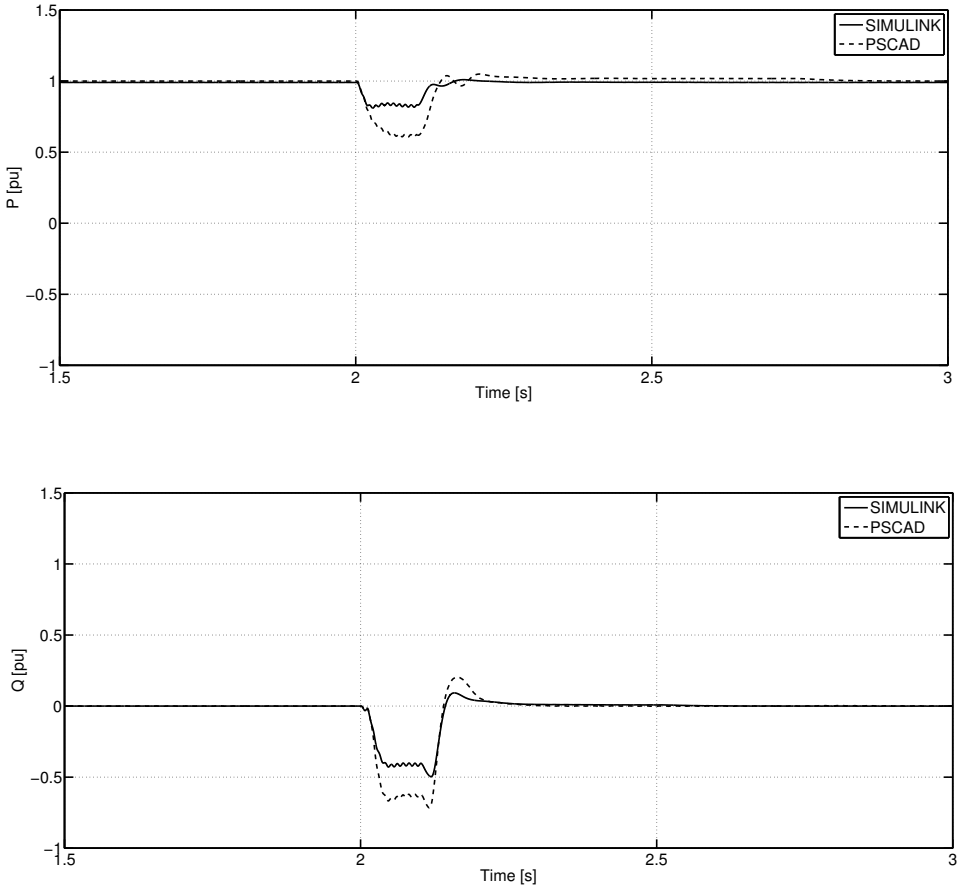


Figure B.11: Single-phase fault at $t = 2\,\mathrm{s}$ with 10% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

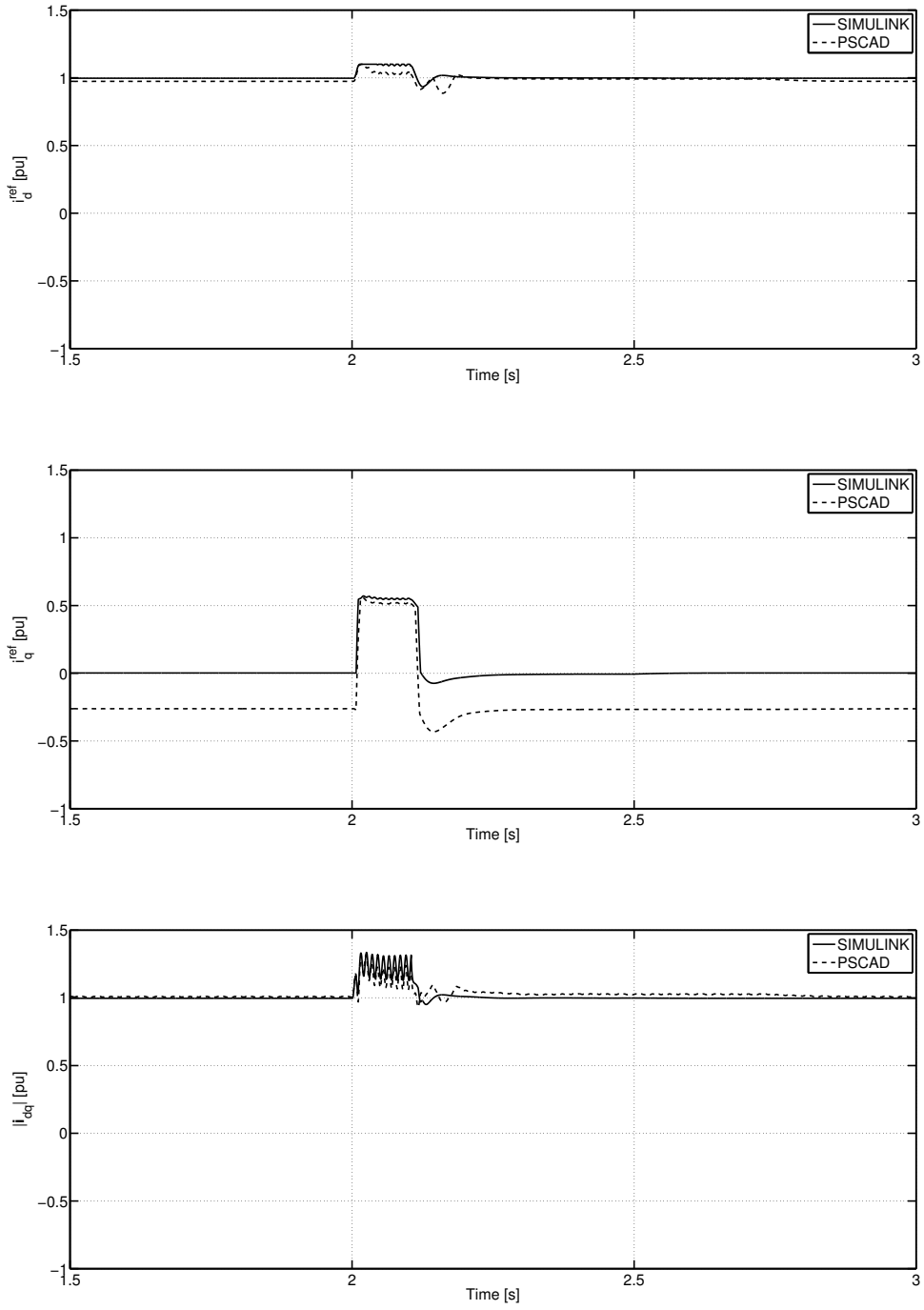


Figure B.12: Single-phase fault at $t = 2\,\mathrm{s}$ with 10% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

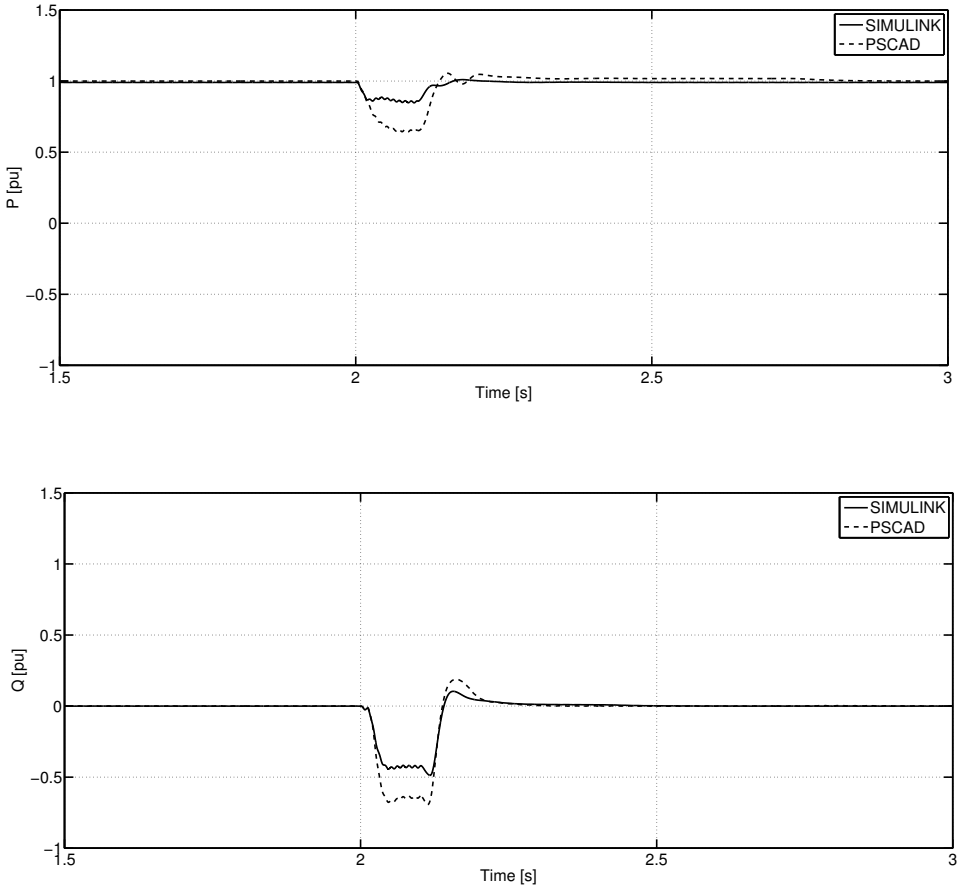


Figure B.13: Single-phase fault at $t = 2\,\mathrm{s}$ with 30% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

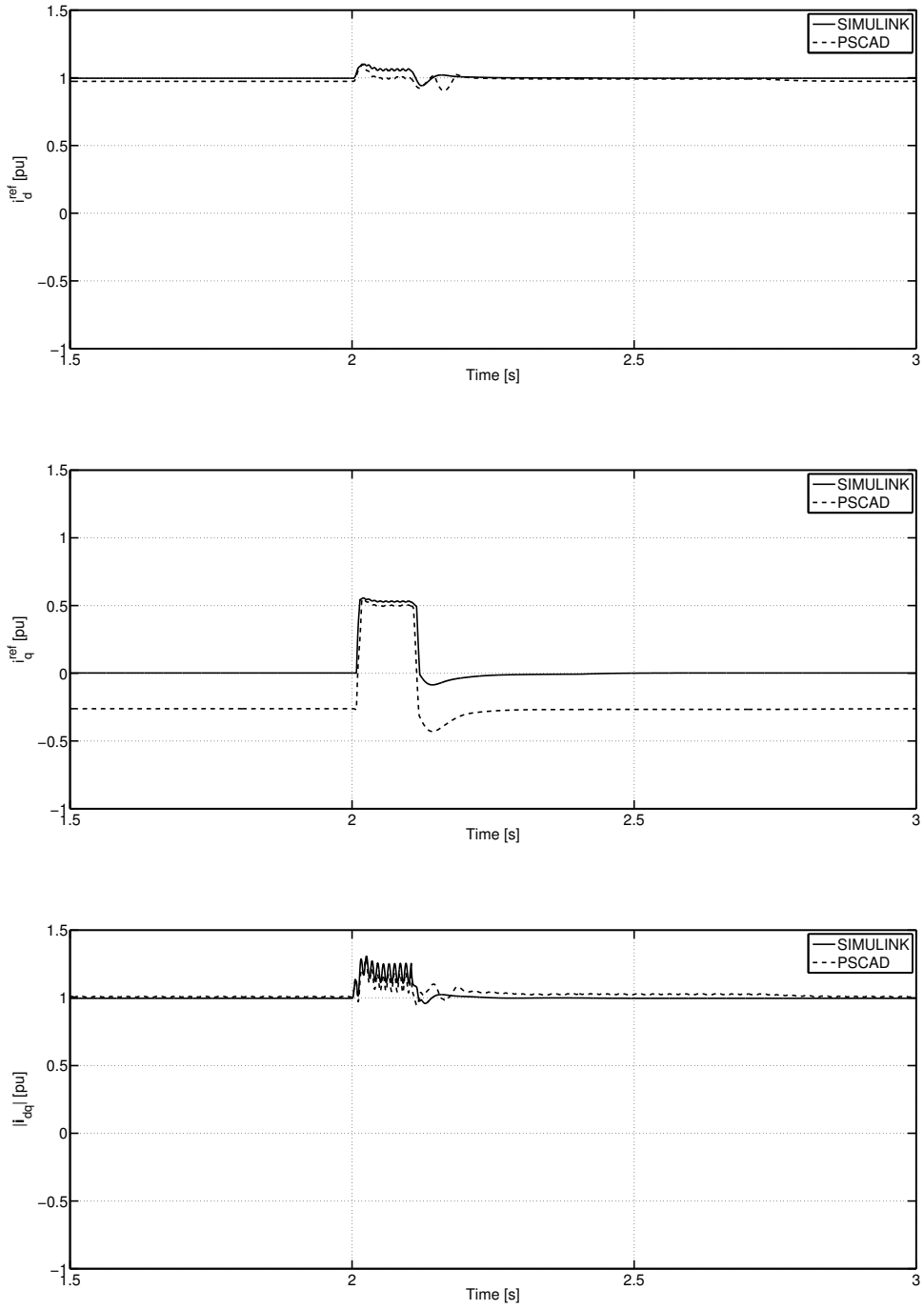


Figure B.14: Single-phase fault at $t = 2\,\mathrm{s}$ with 30% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

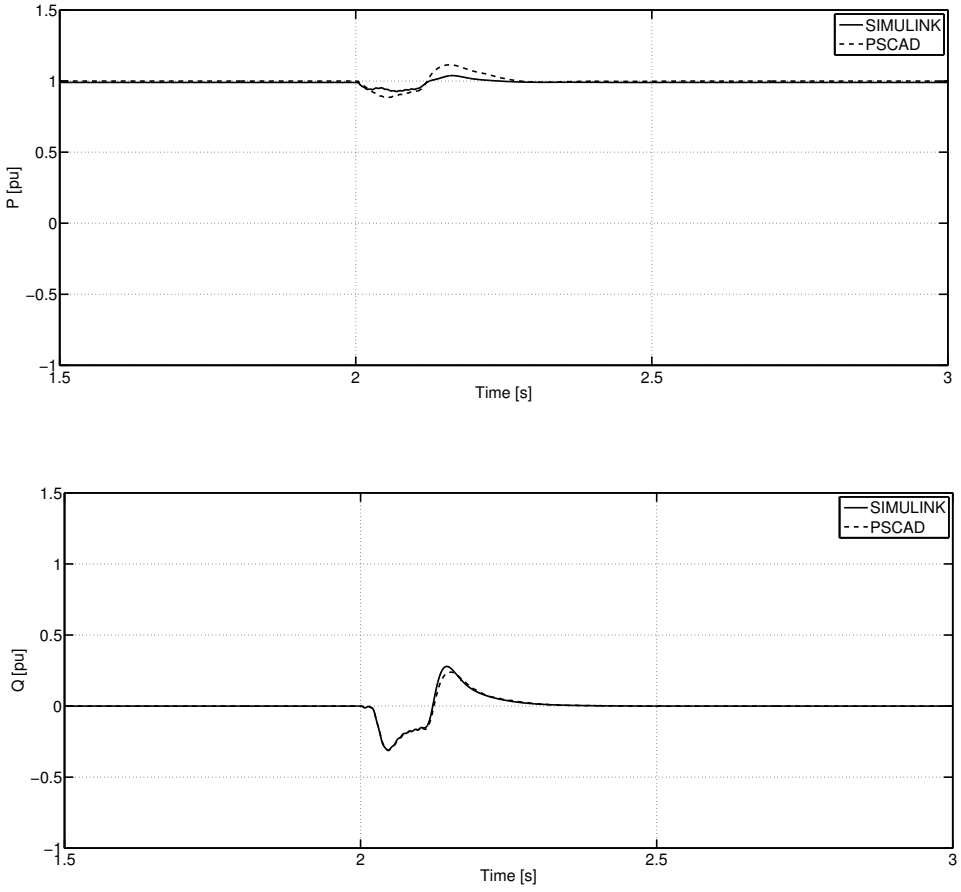


Figure B.15: Single-phase fault at $t = 2\,\mathrm{s}$ with 70% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

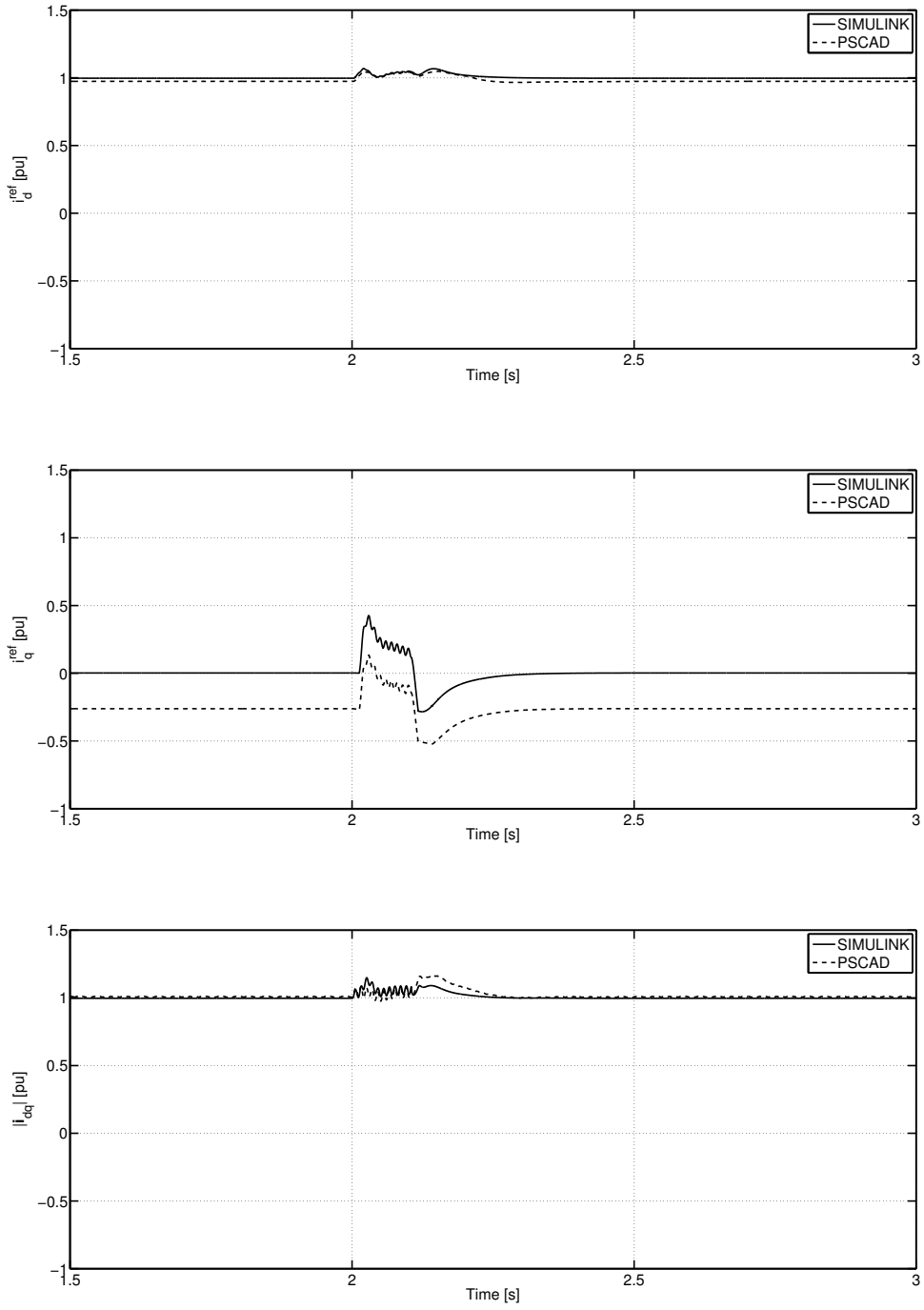


Figure B.16: Single-phase fault at $t = 2\,\mathrm{s}$ with 70% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

Appendix C

Code Implementation

C.1 Step of the Active- and Reactive-Power Controller

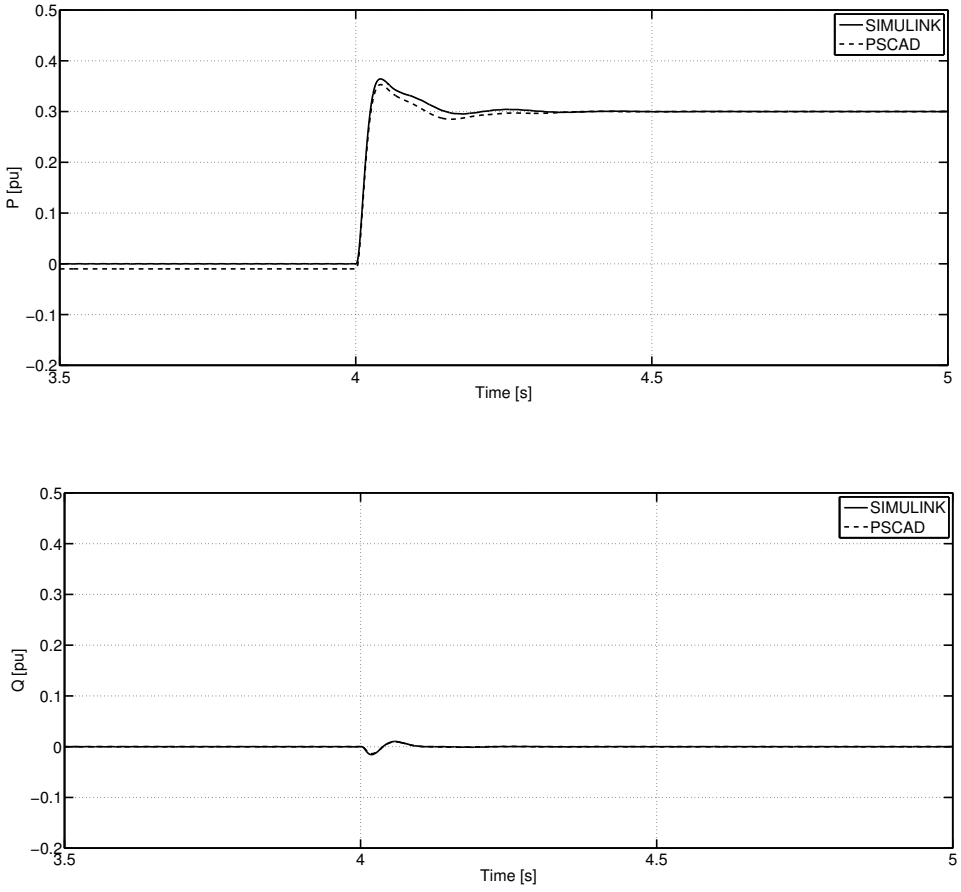


Figure C.1: Active power step change by 30% at $t = 4\,\mathrm{s}$. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

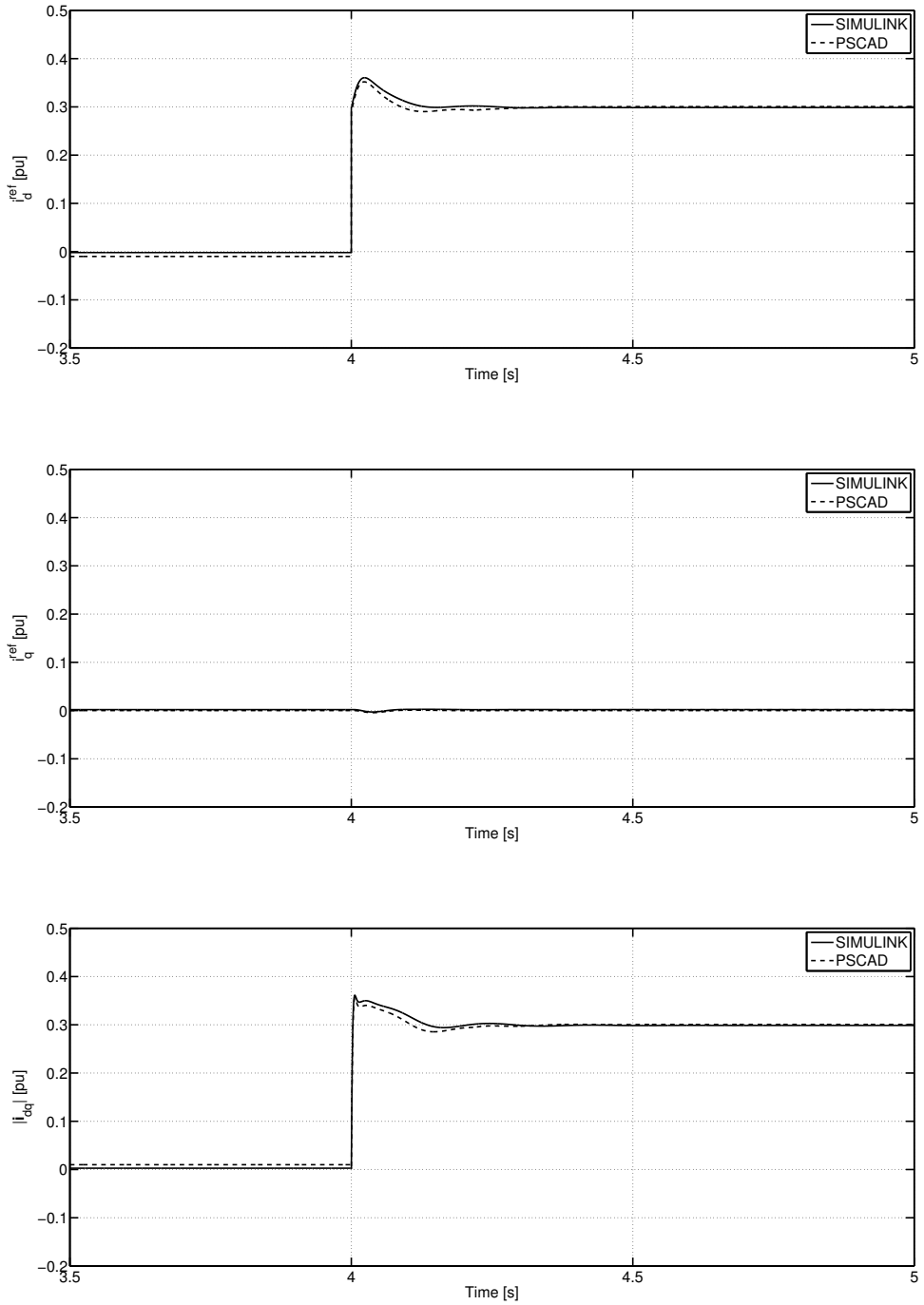


Figure C.2: The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref} during the active power step change by 30% at $t = 4\,\mathrm{s}$. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$ during the active power step change by 30% at $t = 4\,\mathrm{s}$.

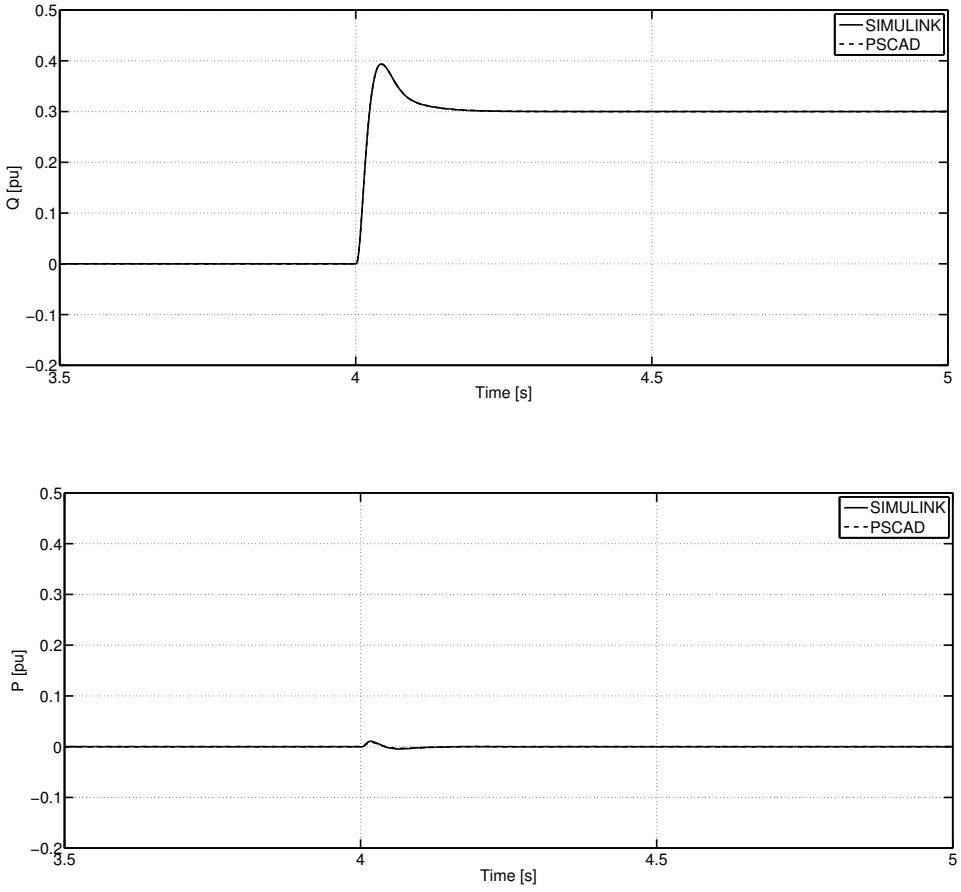


Figure C.3: Reactive power step change by 30% at $t = 4\,\mathrm{s}$. The upper figure illustrates the change in reactive power and the lower figure illustrates the change in active power due to that the system is coupled.

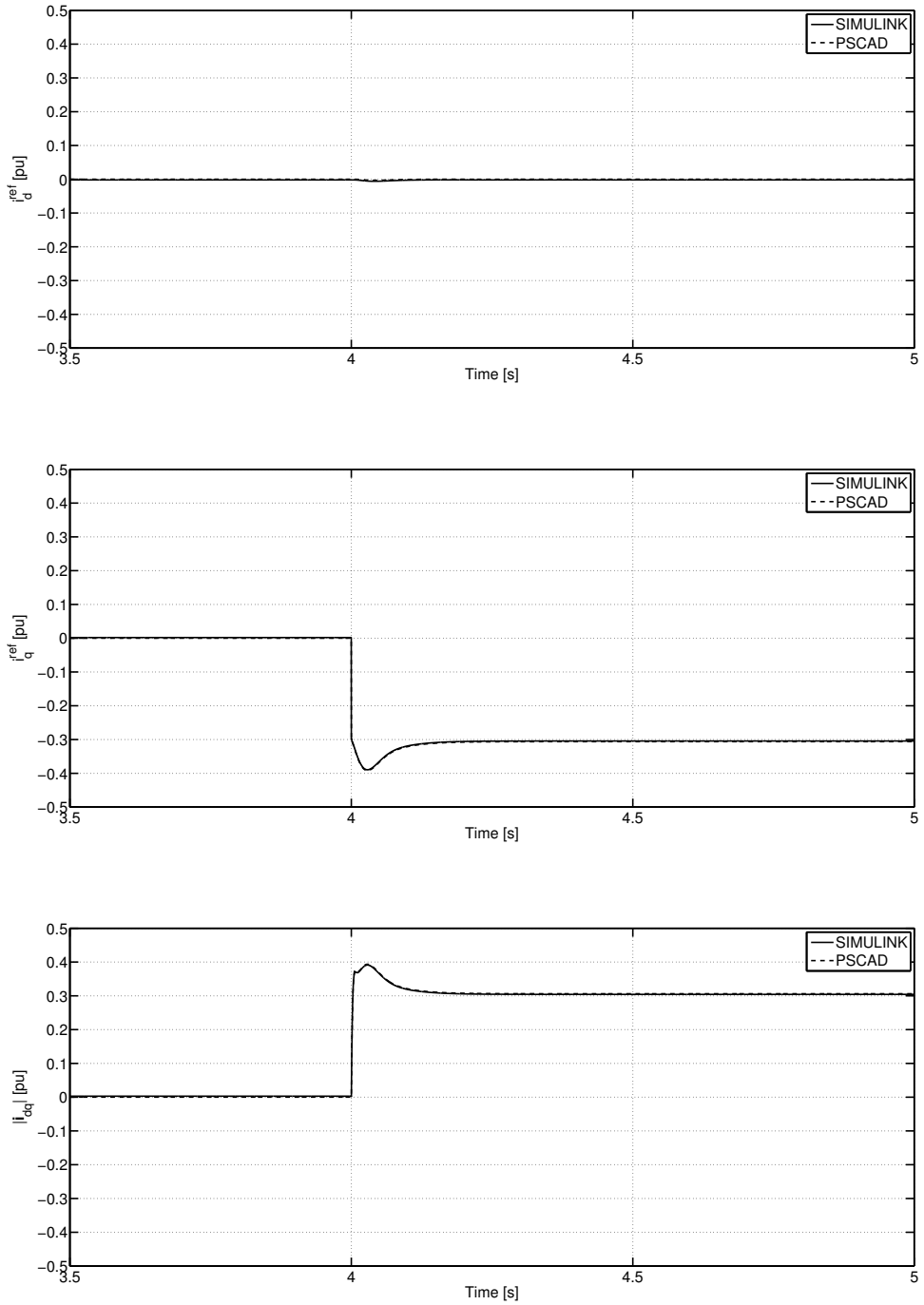


Figure C.4: The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref} during the reactive power step change by 30% at $t = 4\,\mathrm{s}$. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$ during the reactive power step change by 30% at $t = 4\,\mathrm{s}$.

C.2 Three-Phase Faults (Fault Scenarios 3 to 5)

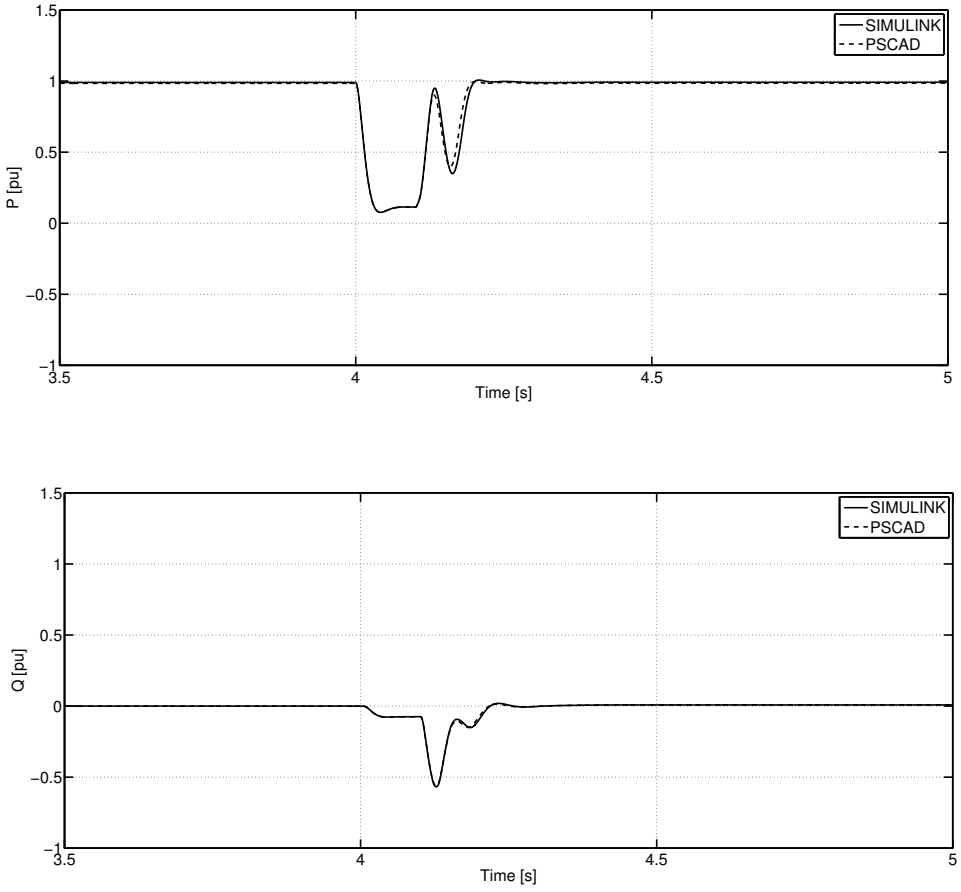


Figure C.5: Three-phase fault at $t = 4\,\mathrm{s}$ with 10% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

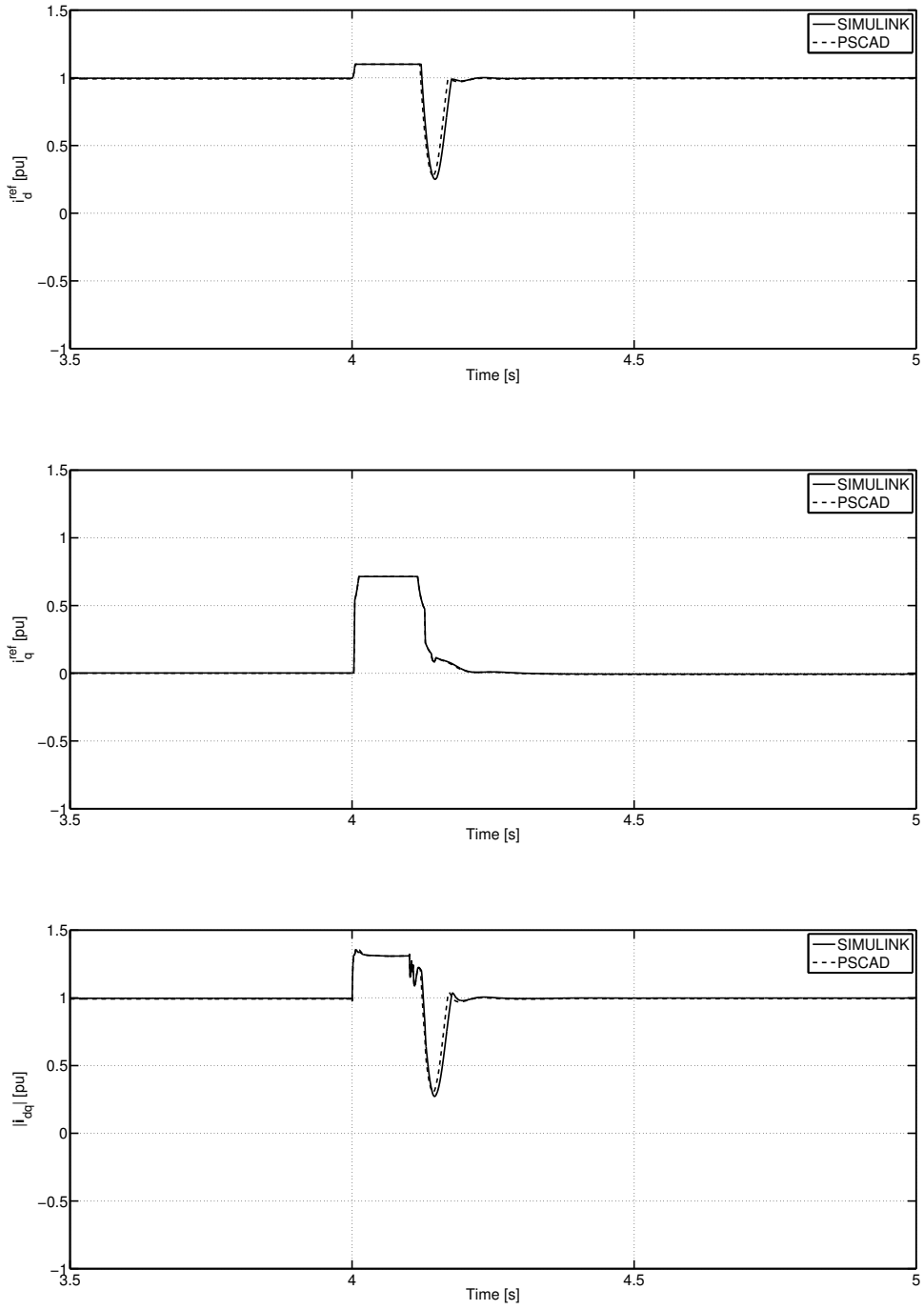


Figure C.6: Three-phase fault at $t = 4\,\mathrm{s}$ with 10% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

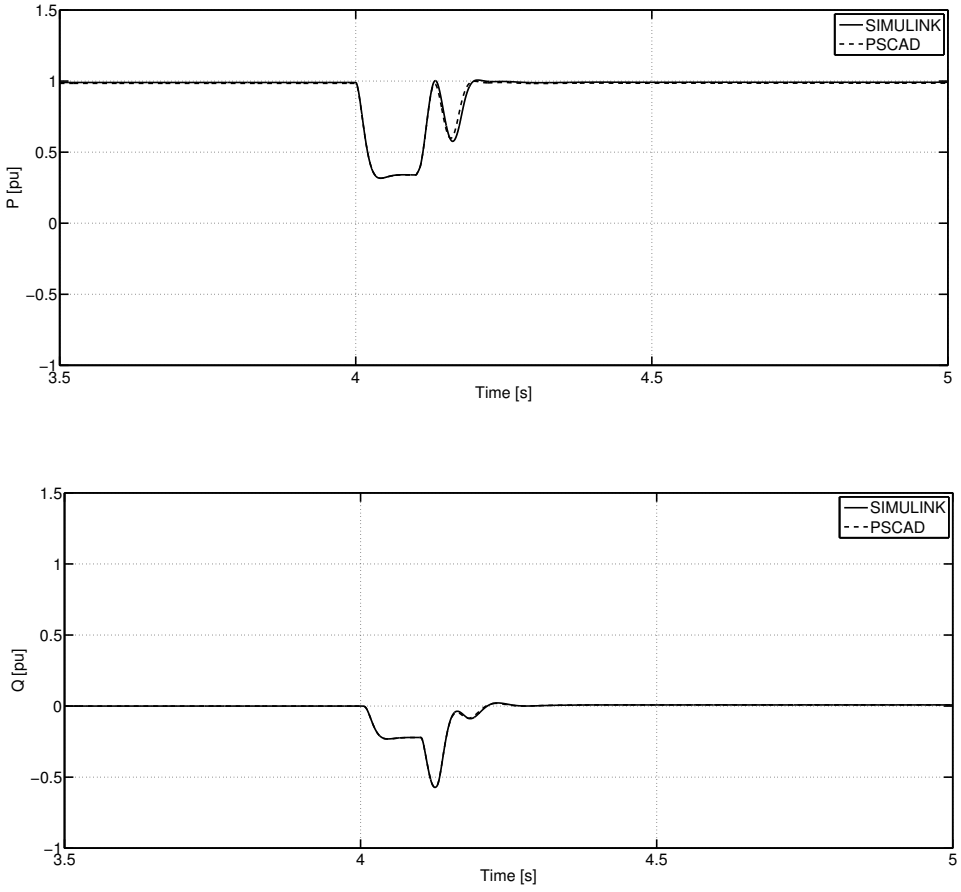


Figure C.7: Three-phase fault at $t = 4\,\mathrm{s}$ with 30% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

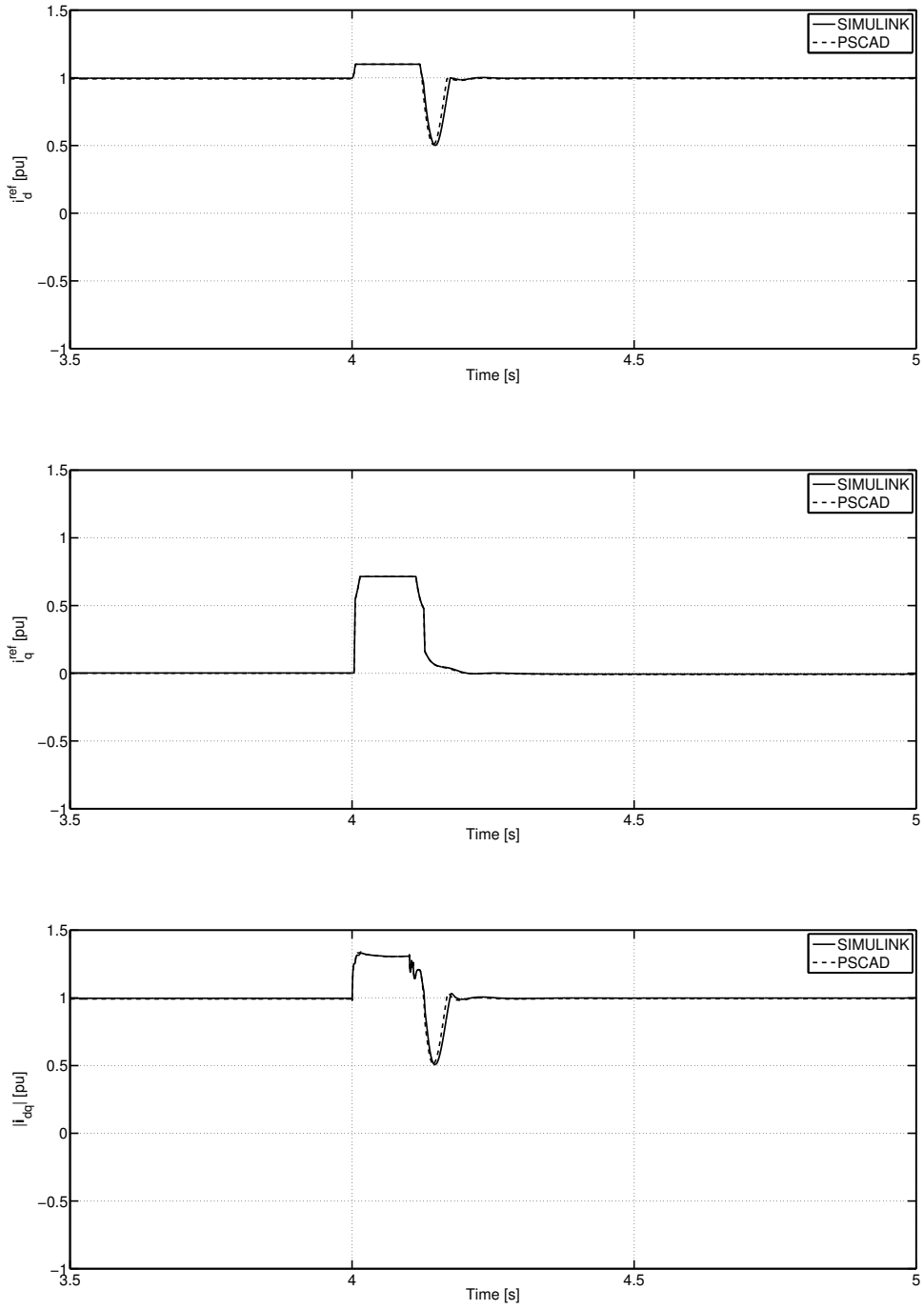


Figure C.8: Three-phase fault at $t = 4\,\mathrm{s}$ with 30% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

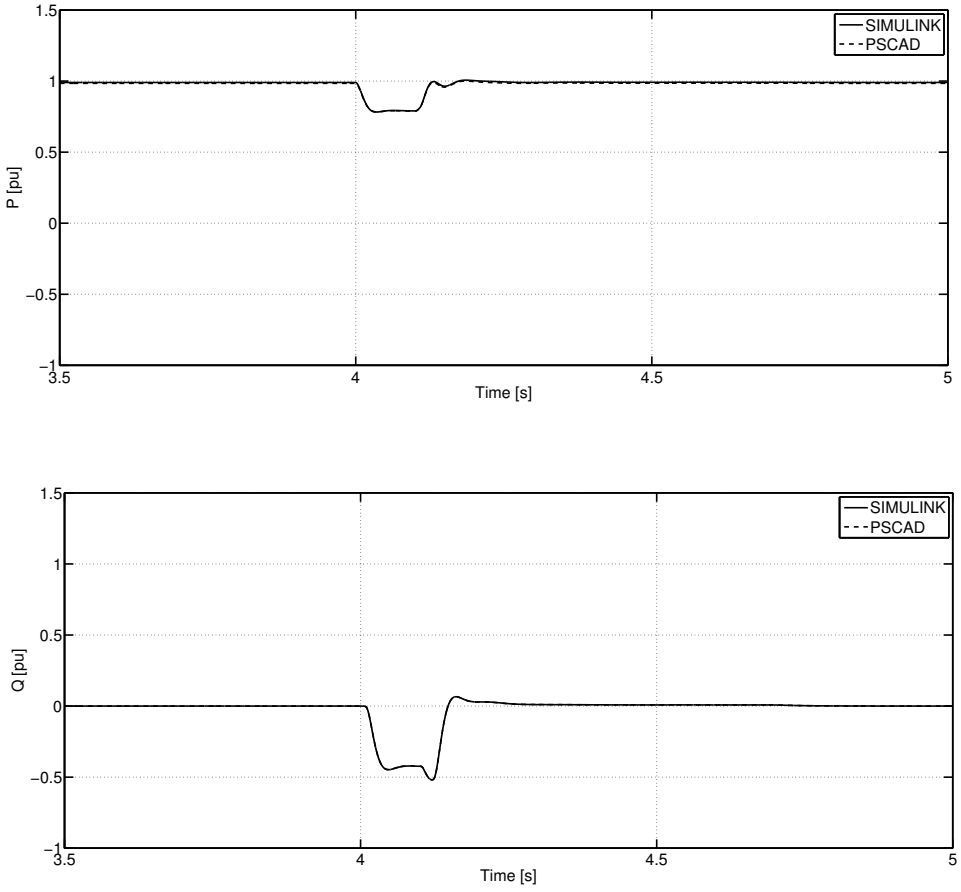


Figure C.9: Three-phase fault at $t = 4\,\mathrm{s}$ with 70% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

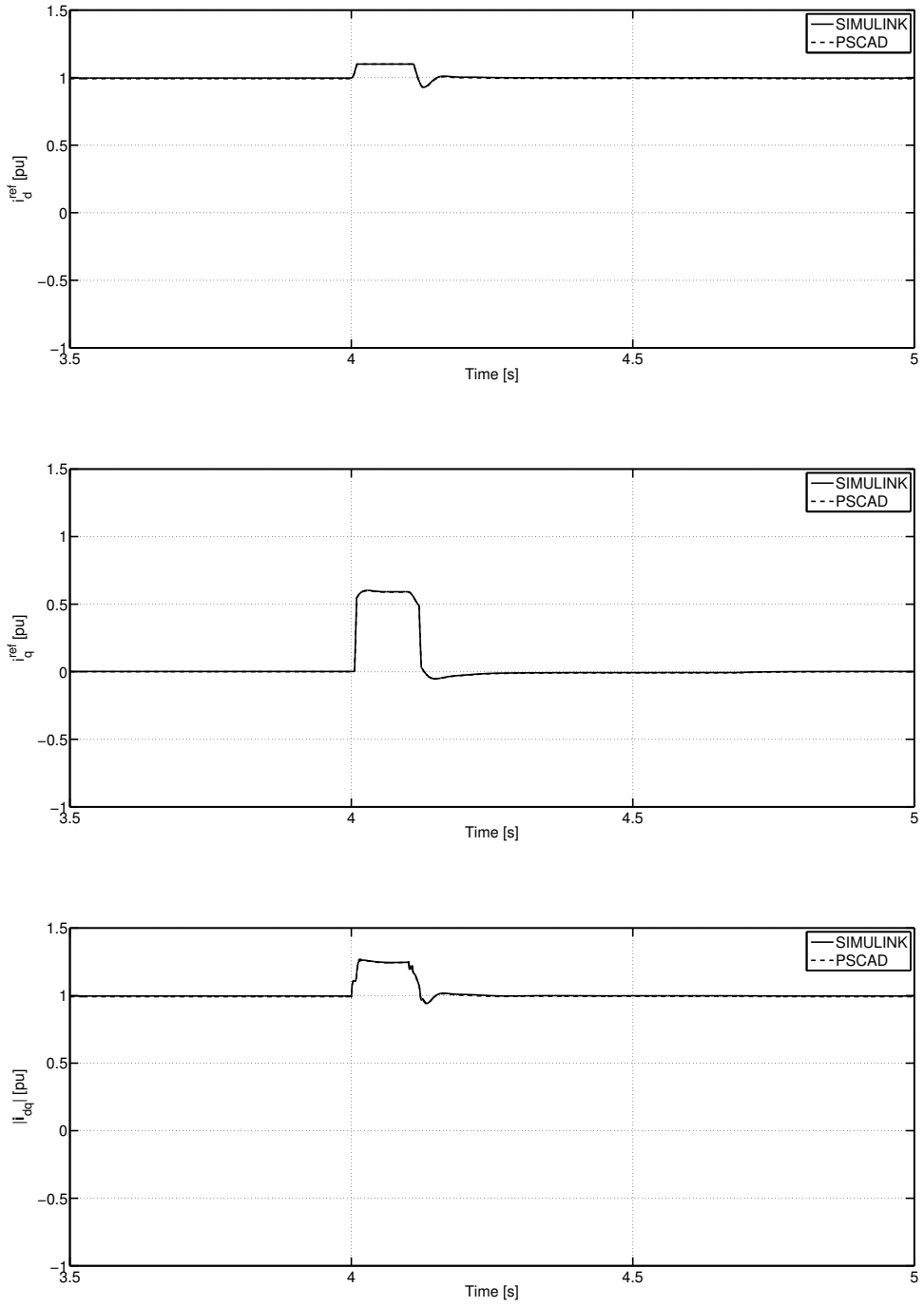


Figure C.10: Three-phase fault at $t = 4\,\mathrm{s}$ with 70% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

C.3 Single-Phase Faults (Fault Scenarios 6 to 8)

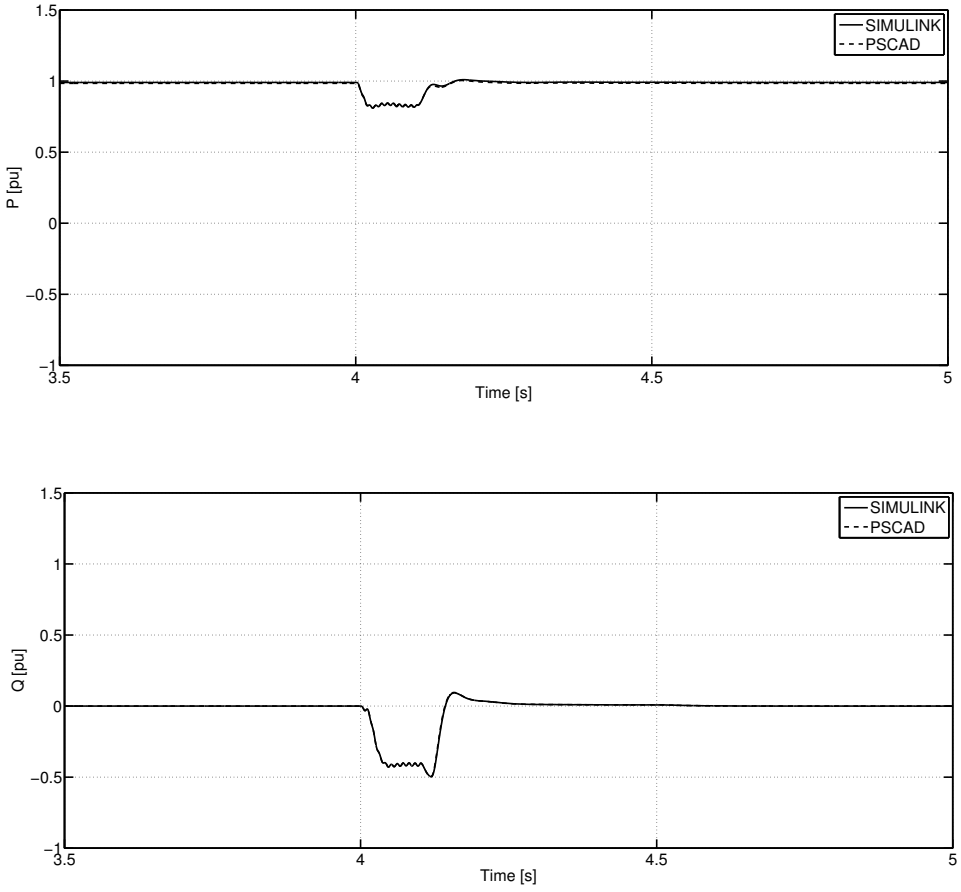


Figure C.11: Single-phase fault at $t = 4\,\mathrm{s}$ with 10% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

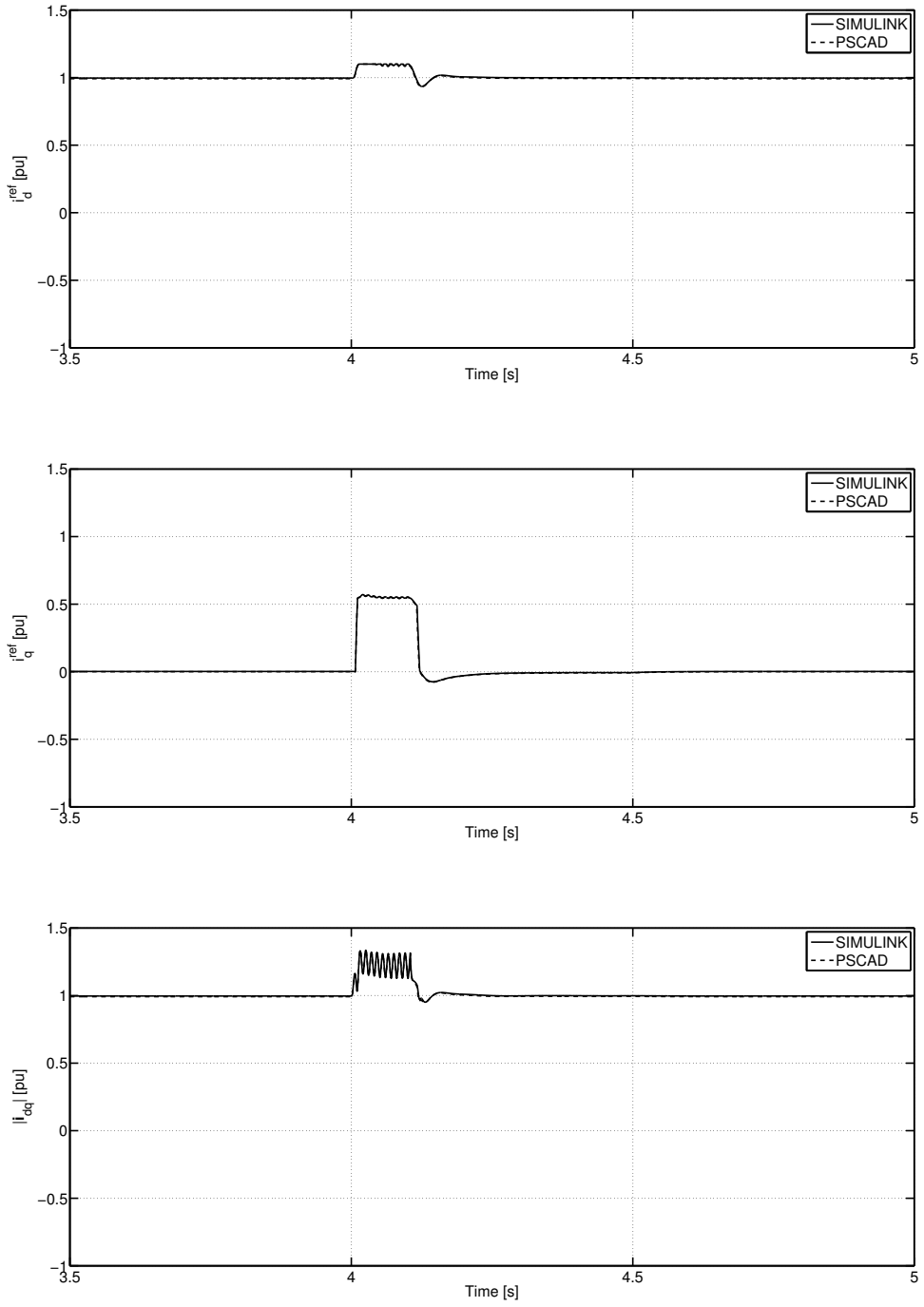


Figure C.12: Single-phase fault at $t = 4\,\mathrm{s}$ with 10% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

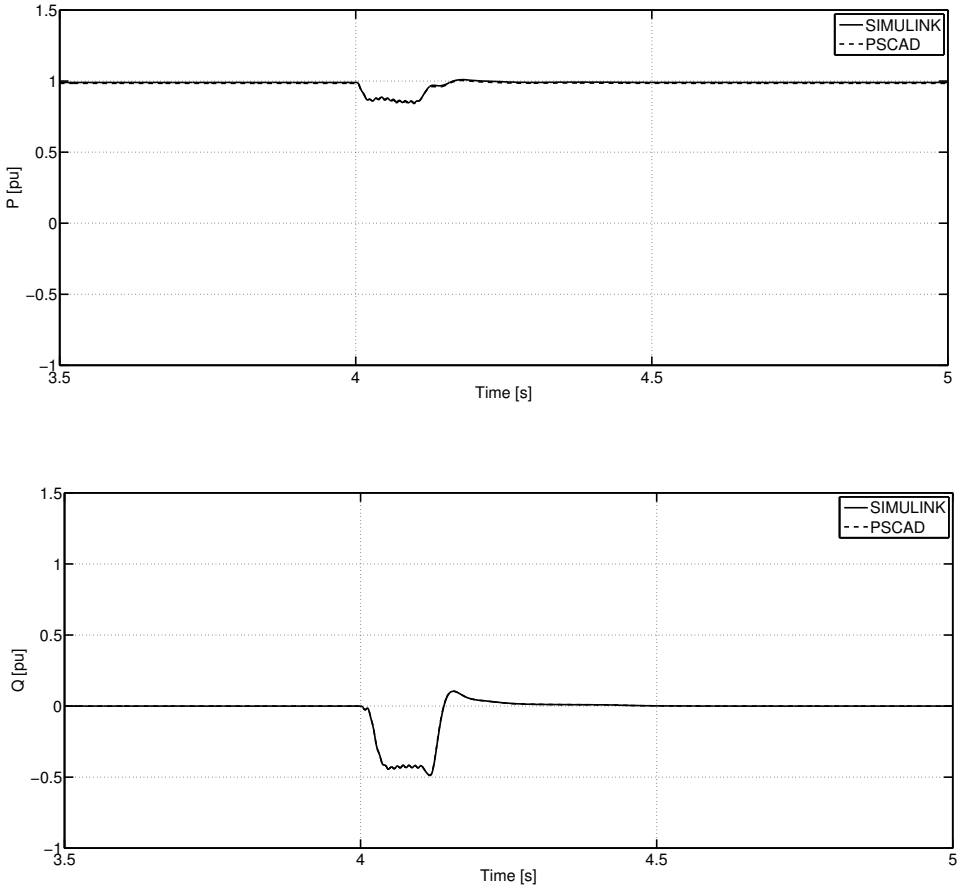


Figure C.13: Single-phase fault at $t = 4\,\mathrm{s}$ with 30% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

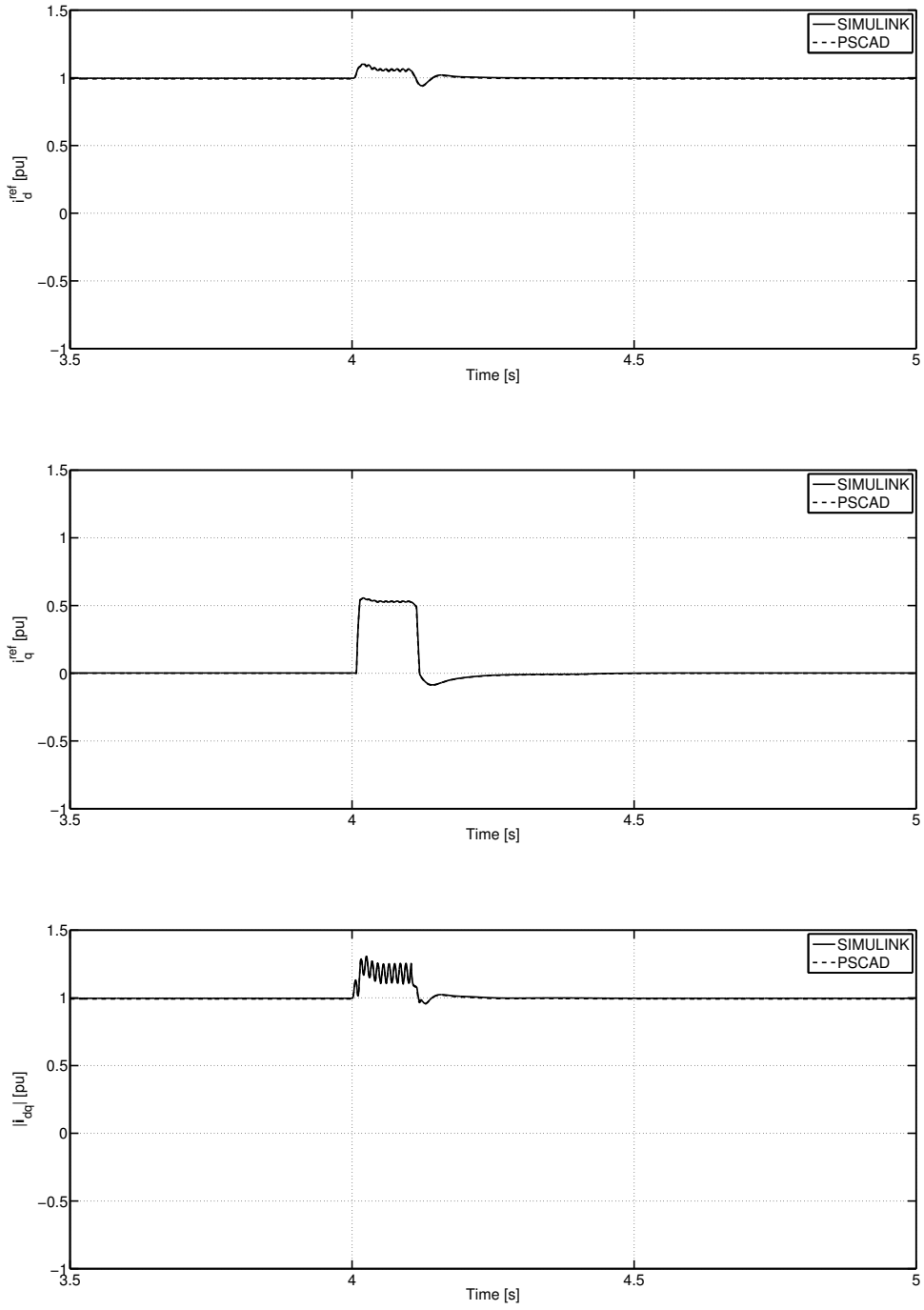


Figure C.14: Single-phase fault at $t = 4\,\mathrm{s}$ with 30% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

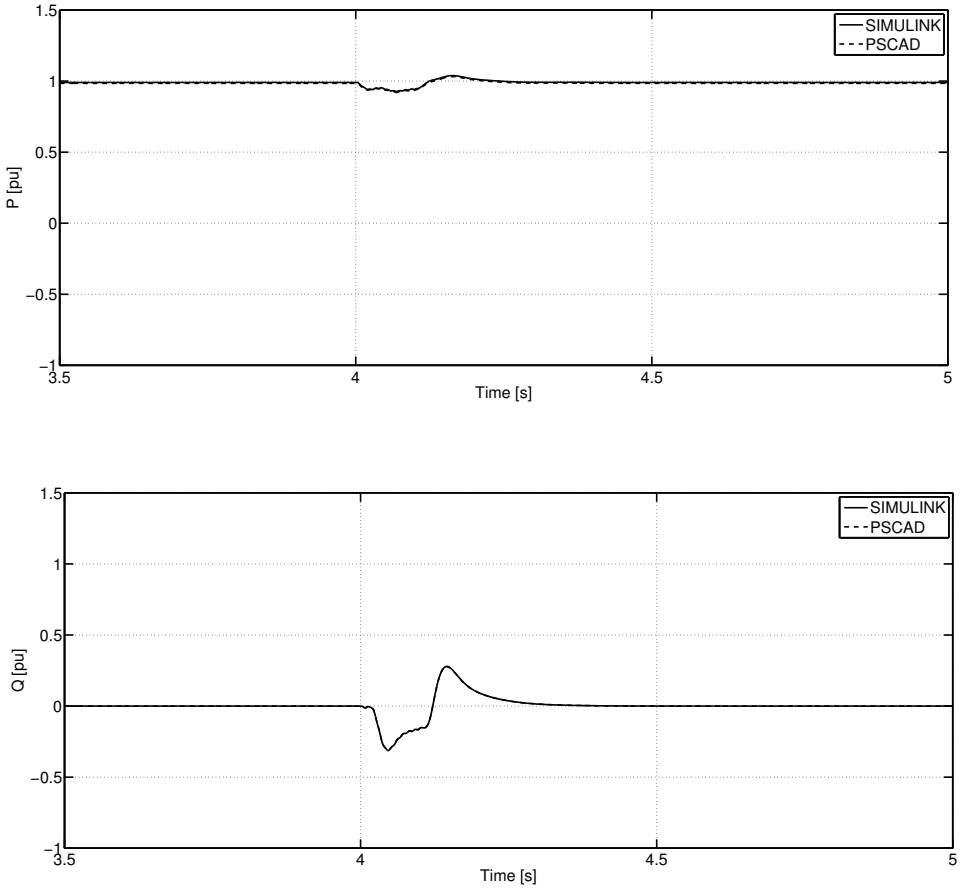


Figure C.15: Single-phase fault at $t = 4\,\mathrm{s}$ with 70% remaining voltage. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

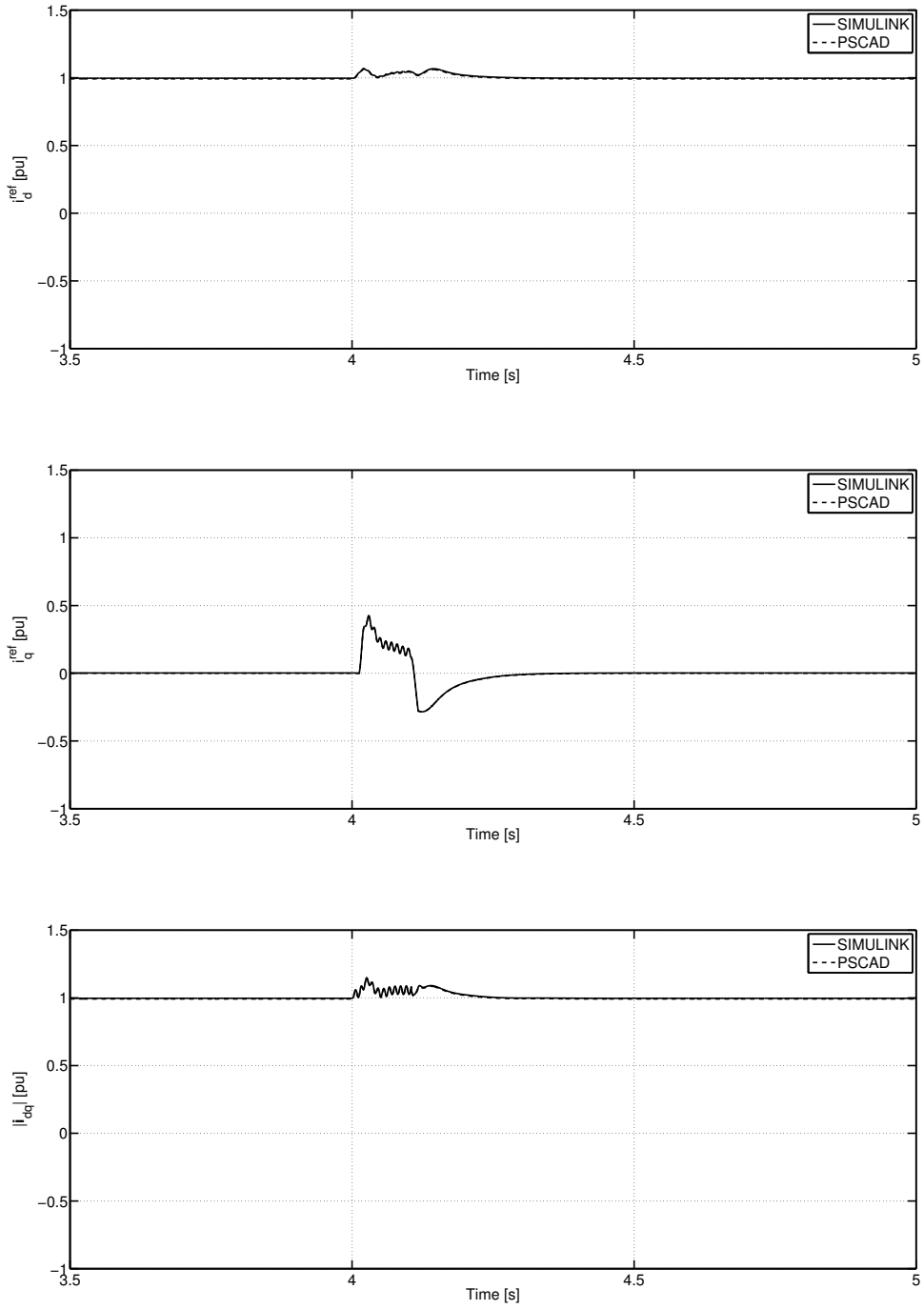


Figure C.16: Single-phase fault at $t = 4\,\mathrm{s}$ with 70% remaining voltage. The two upper figures show the change of the reference currents i_d^{ref} and i_q^{ref}. The lower figure illustrates the changes in the current $|\mathbf{i}_{dq}|$.

Appendix D

Black-Box Model Comparisons

D.1 Step of the Active- and Reactive-Power Controller

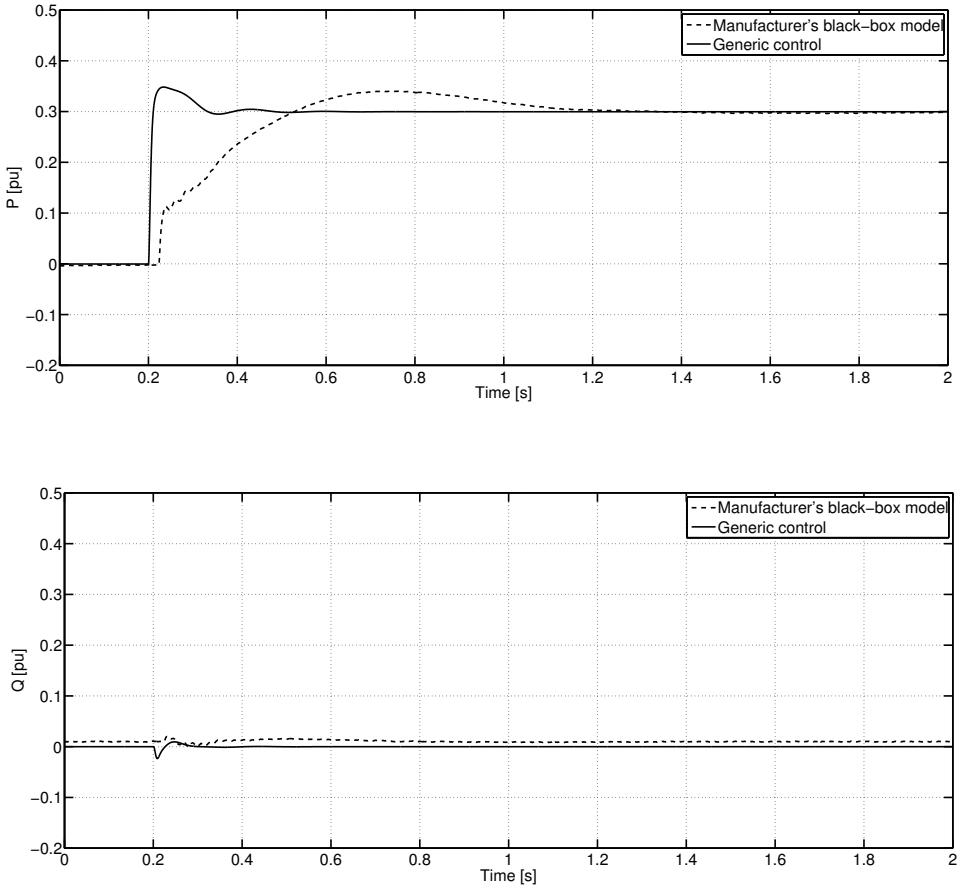


Figure D.1: Active power step change by 30% at $t = 0.2\,\mathrm{s}$. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

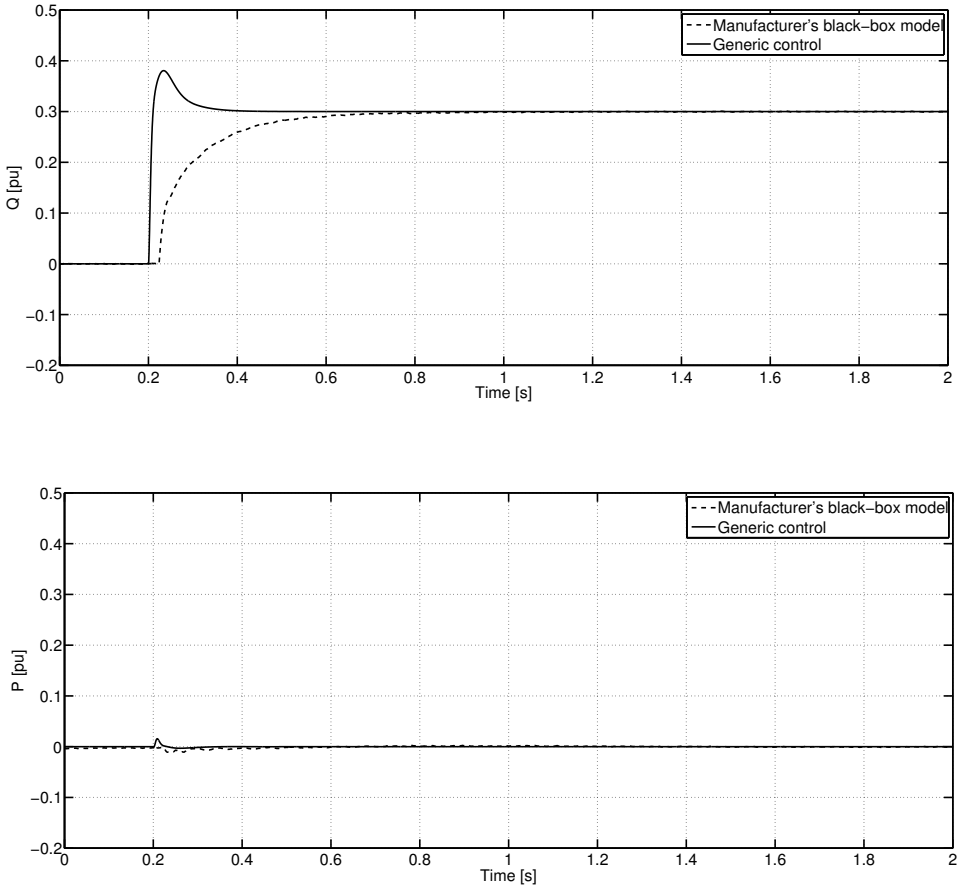


Figure D.2: Reactive power step change by 30% at $t = 0.2\,\mathrm{s}$. The upper figure illustrates the change in reactive power and the lower figure illustrates the change in active power due to that the system is coupled.

D.2 Three-Phase Faults at Rectifier Side (Fault Scenarios 3 to 5)

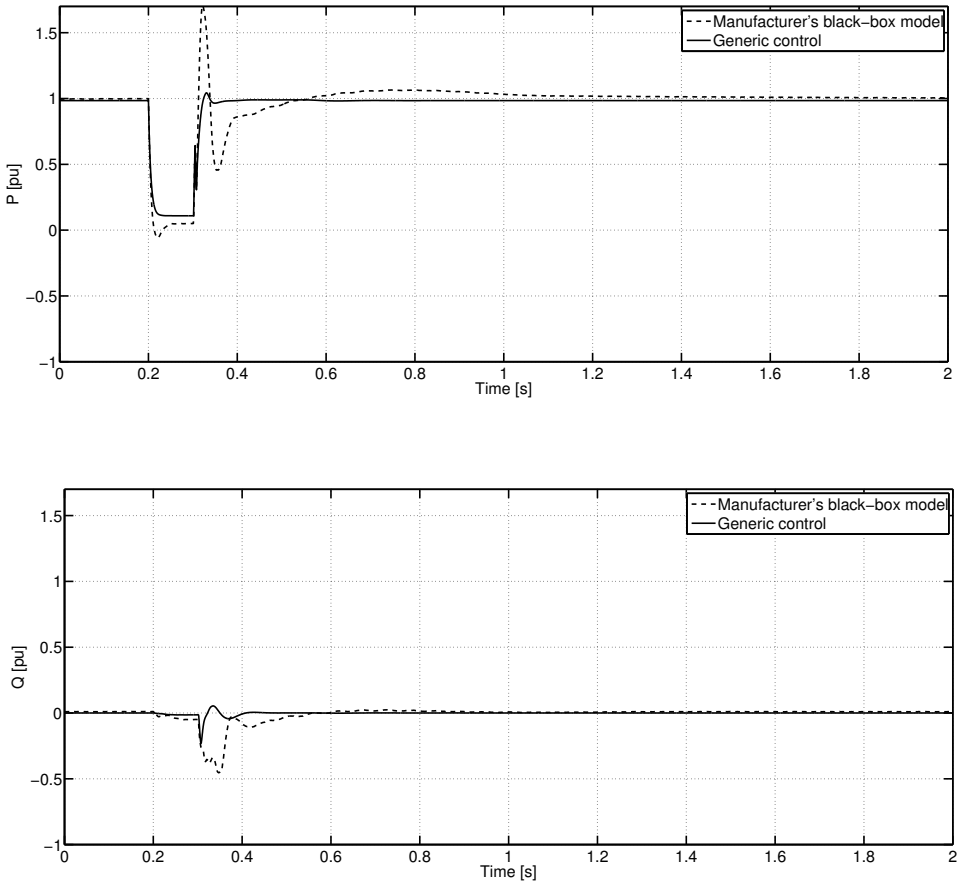


Figure D.3: Three-phase fault at $t = 0.2\,\mathrm{s}$ with 10% remaining voltage. The fault is applied on the rectifier side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

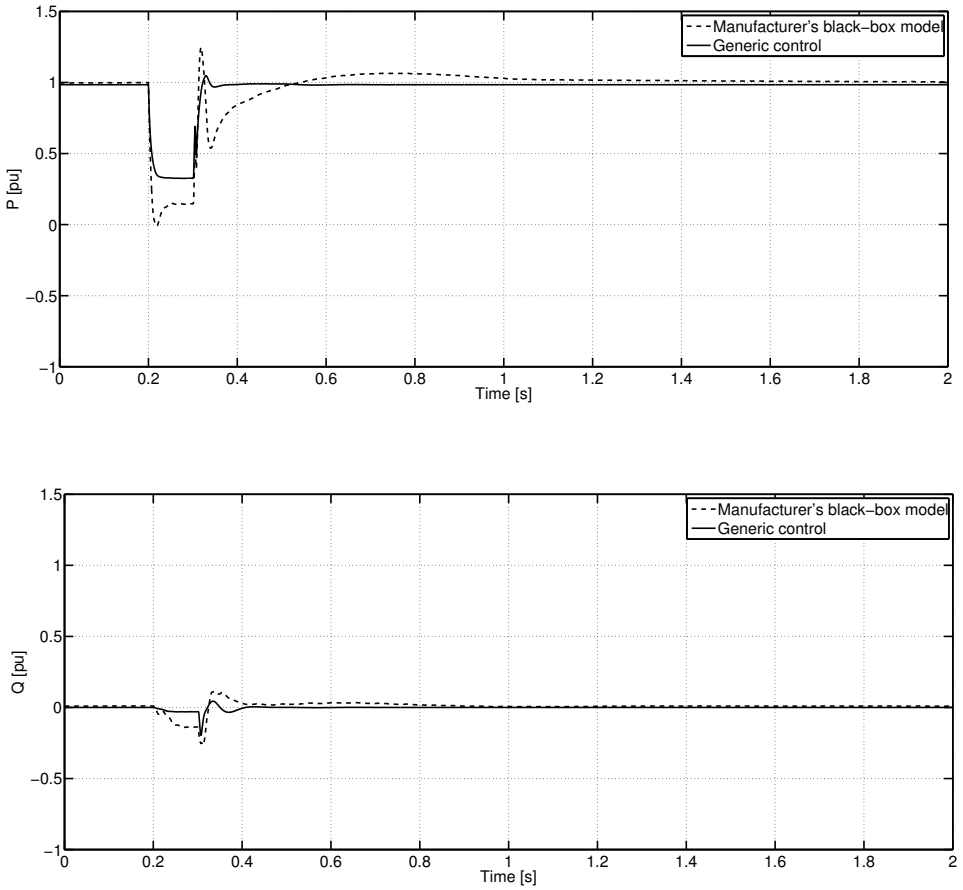


Figure D.4: Three-phase fault at $t = 0.2\,\mathrm{s}$ with 30% remaining voltage. The fault is applied on the rectifier side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

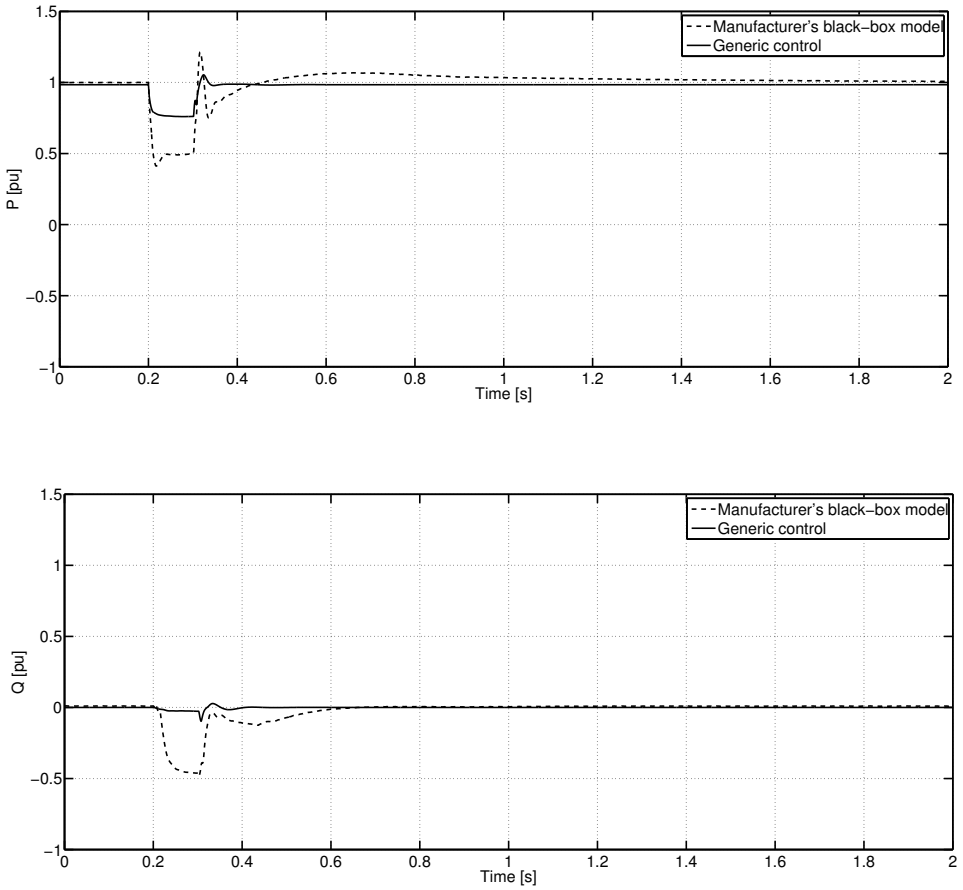


Figure D.5: Three-phase fault at $t = 0.2\,\mathrm{s}$ with 70% remaining voltage. The fault is applied on the rectifier side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

D.3 Single-Phase Faults at Rectifier Side (Fault Scenarios 6 to 8)

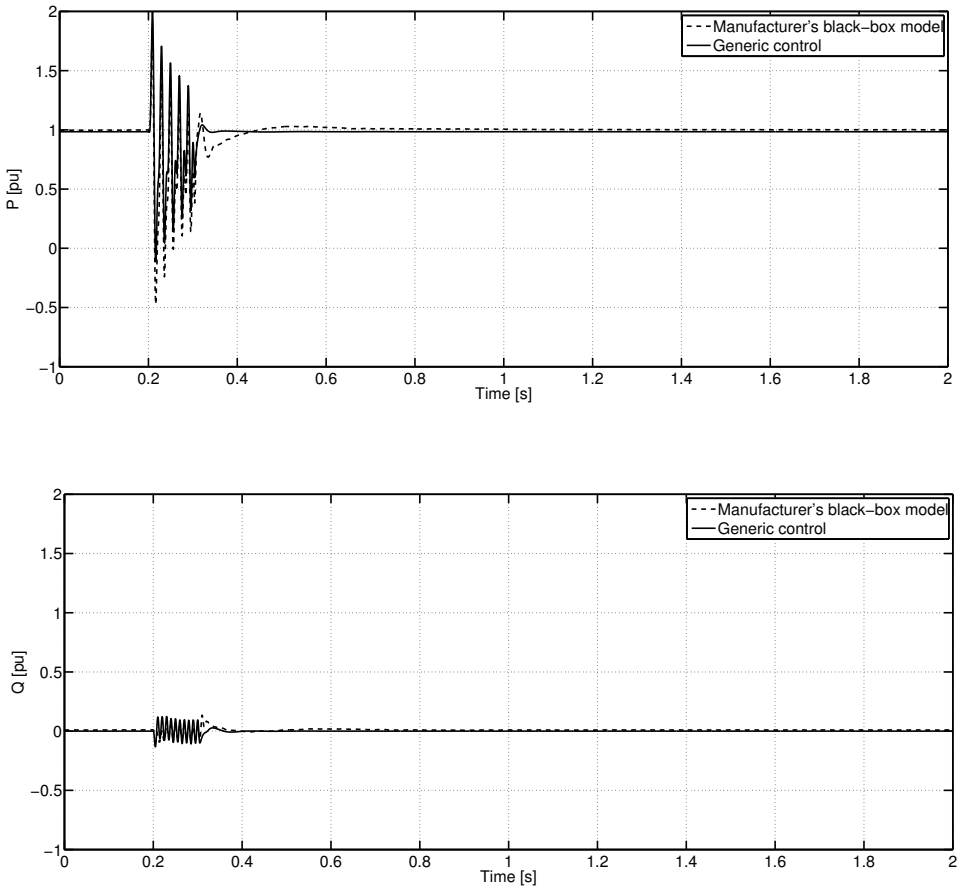


Figure D.6: Single-phase fault at $t = 0.2\,\mathrm{s}$ with 10% remaining voltage. The fault is applied on the rectifier side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

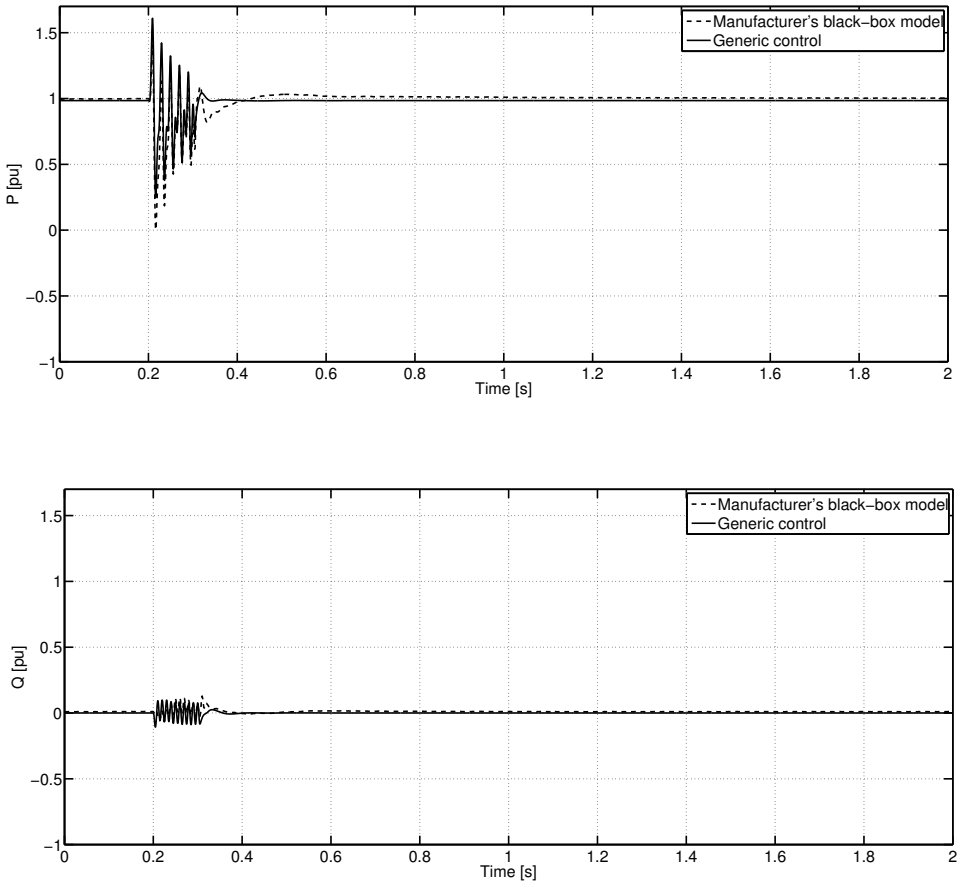


Figure D.7: Single-phase fault at $t = 0.2\,\mathrm{s}$ with 30% remaining voltage. The fault is applied on the rectifier side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

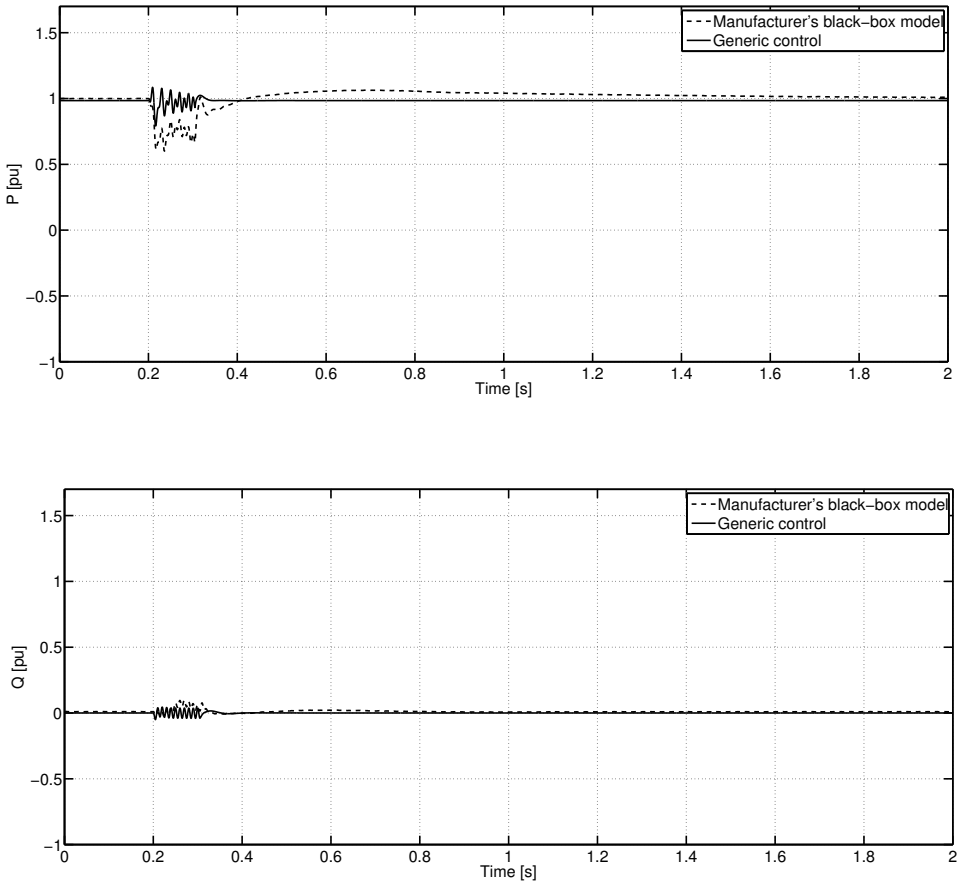


Figure D.8: Single-phase fault at $t = 0.2\,\mathrm{s}$ with 70% remaining voltage. The fault is applied on the rectifier side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

D.4 Three-Phase Faults at Inverter Side (Fault Scenarios 3 to 5)

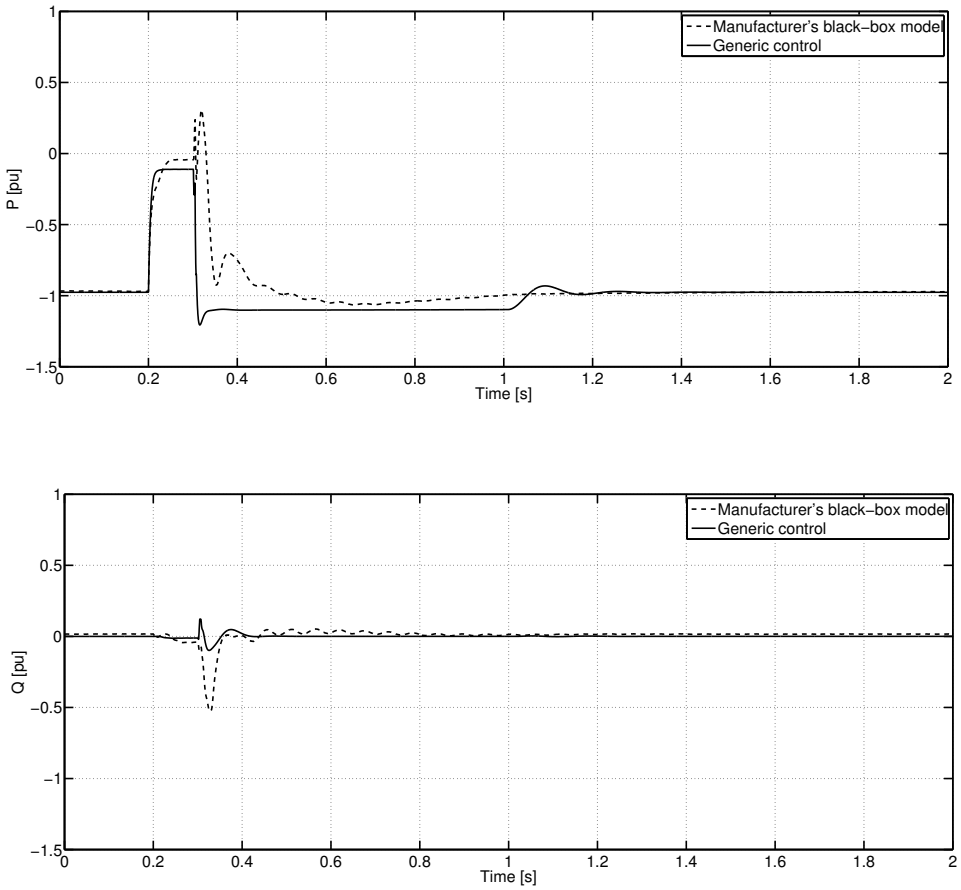


Figure D.9: Three-phase fault at $t = 0.2\,\mathrm{s}$ with 10% remaining voltage. The fault is applied on the inverter side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

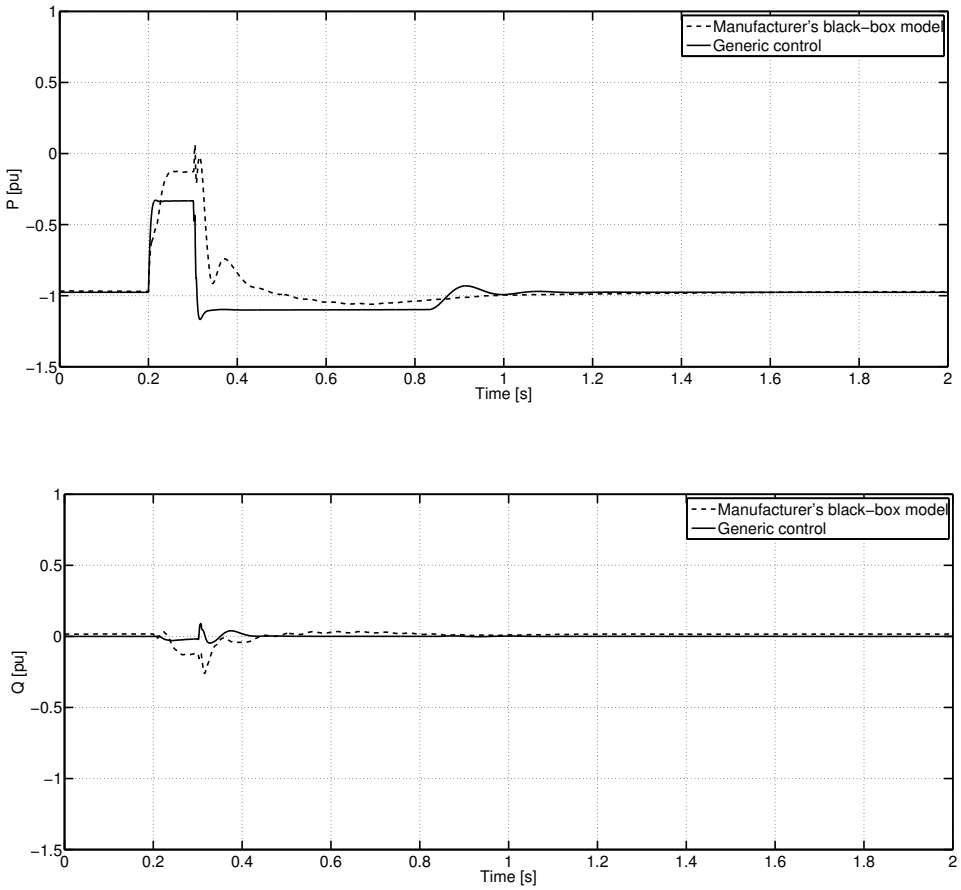


Figure D.10: Three-phase fault at $t = 0.2\,\mathrm{s}$ with 30% remaining voltage. The fault is applied on the inverter side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

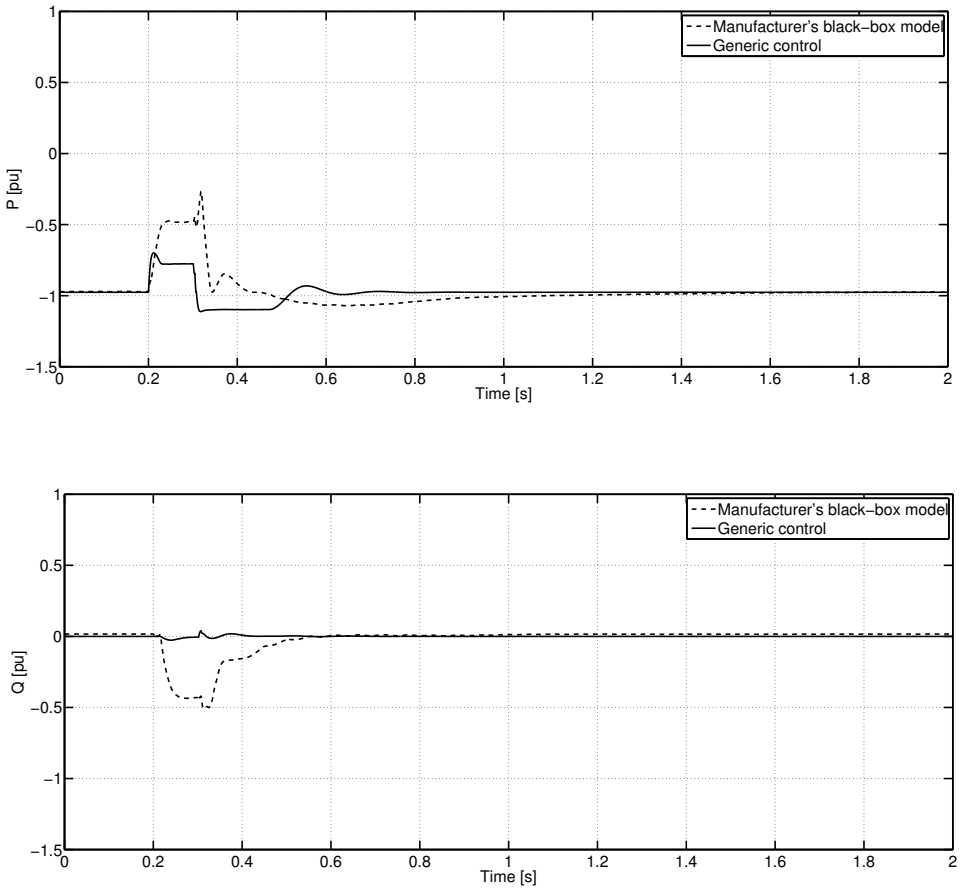


Figure D.11: Three-phase fault at $t = 0.2\,\mathrm{s}$ with 70% remaining voltage. The fault is applied on the inverter side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

D.5 Single-Phase Faults at Inverter Side (Fault Scenarios 6 to 8)

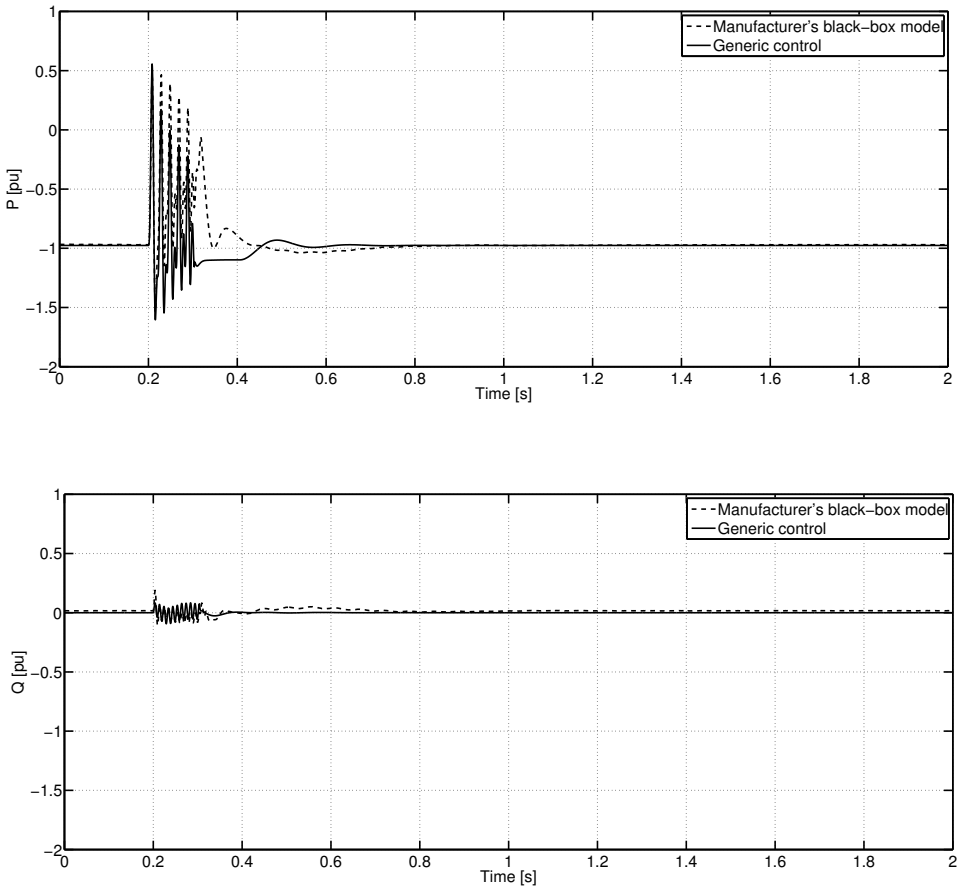


Figure D.12: Single-phase fault at $t = 0.2\,\mathrm{s}$ with 10% remaining voltage. The fault is applied on the inverter side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

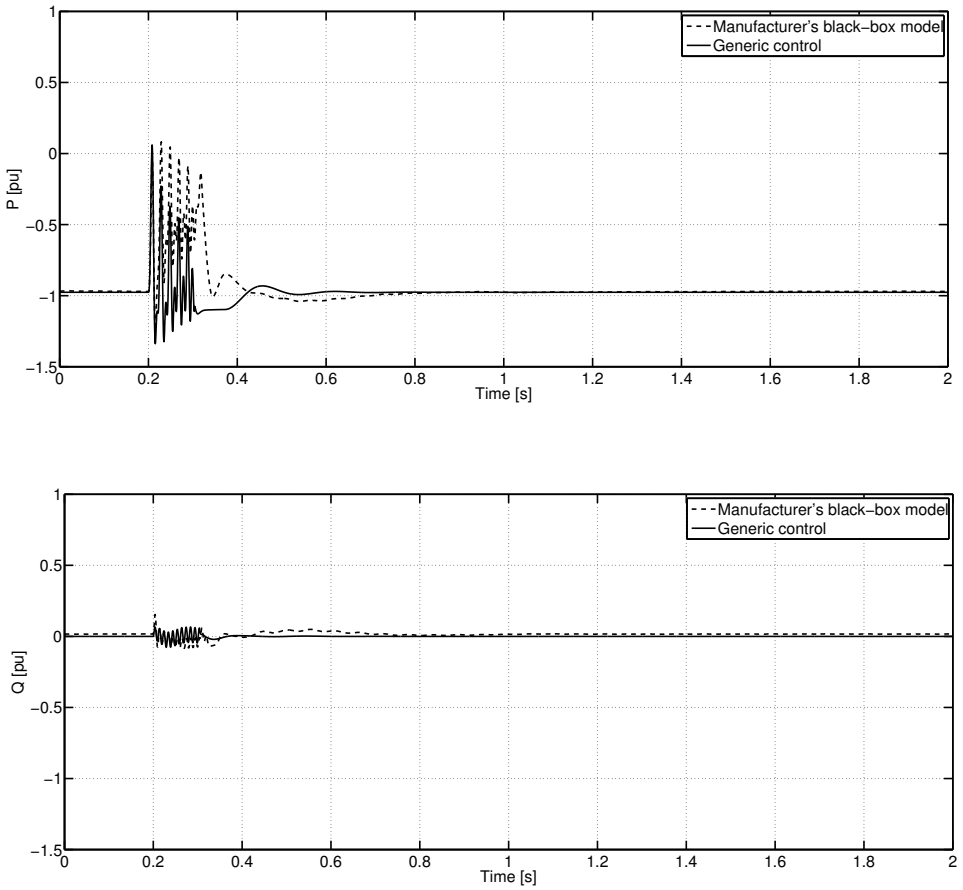


Figure D.13: Single-phase fault at $t = 0.2\,\mathrm{s}$ with 30% remaining voltage. The fault is applied on the inverter side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.

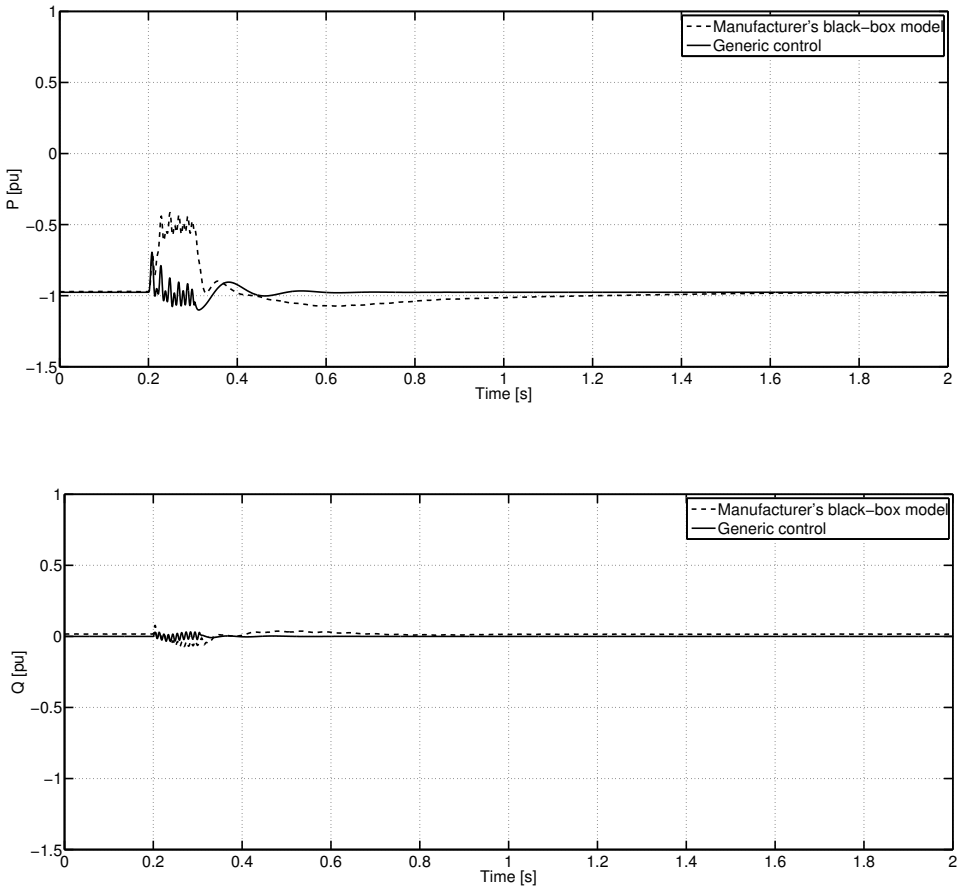


Figure D.14: Single-phase fault at $t = 0.2\,\mathrm{s}$ with 70% remaining voltage. The fault is applied on the inverter side. The upper figure illustrates the change in active power and the lower figure illustrates the change in reactive power due to that the system is coupled.