NTNU – Trondheim
Norwegian University of
Science and Technology

Investigating G-parameters Modelling of Power Electronc Devices

Sigurd Hofsmo Jakobsen

Master of Energy and Environmental Engineering
Submission date: June 2013
Supervisor: Marta Molinas, ELKRAFT
Co-supervisor: Santiago Sanchez Acevedo, ELKRAFT

Norwegian University of Science and Technology
Department of Electric Power Engineering

Problem Description

The work will consist of developing a methodology to model power electronic converters as a multiport model, where the multiport model is described by g-parameters. The models are to be verified by simulations and implemented to a generic controlled converter of the same topology for validation purposes. A stability analysis will be performed using the g-parameters as basis, both by Nyquist's criterion and simulations

Samandrag

I takt med eit stadig aukande folketal på jorda, aukar også energibehovet. Det er også ei brei internasjonal semje om at utviklinga må vera berekraftig. Store kolkraftverk og andre særskild forureinande kraftverk, må derfor bytas ut med reinare kraftkjelder. Noko som må skje på likt med utbyggjinga av ny kraft, for å møta det aukande energibehovet. Sentralt i denne utviklinga er fornybar energi, både i form av tradisjonell vasskraft, geotermisk varme og distribuerte system. Særleg vindparkar og solcellepanel har auka i omfang, desse må koplast saman med resten av elektrisitetsnettet ved hjelp av kraftelektronikk. Det foreligg også planar om å byggja høgspenninglikestraumsnett, som også vil trengja kraftelektronikkomformarar. Det er derfor åpenbart at gode modellar av kraftelektronikkomformarar vil verta viktige. I denne avhandlinga vil ein målingsbasert modell verta undersøkt. Styrken til denne framgangsmåten ligg i, at ein ikkje treng å veta noko om styringssystemet eller dei indre parameterane til omformaren.

Fyrst vert ein teroi og metodedel presentert, her vert teorien og metoden grundig forklart. Saman med eit stabilitetskriterium for modellen, og eit forslag til eit nytt likningssett for å rekna ut modellen. Metoden vart fyrst testa ut på ein vekselrettar, som vart utvikla ved hjelp av Simulink. Dette gav ein nyttig plattform for å testa ut ulike aspekt ved metoden, i tillegg til at det er lettare å verifisera modellen ved hjelp av simuleringar. For at metoden skal ha ein praktisk relevans, så må det sjølvsagt også vera mogleg å bruka han på verkelege omformarar. Ein testbenk bestående av ein likestraumsomformar vart utvikla for å testa metoden i eit praktisk tilfelle.

Summary

In a world with an ever increasing population, the demand for energy is constantly on the rise. Most countries also agree, that a sustainable development should be ensured. To accomplish this, a higher penetration of renewables in the present energy system is needed. Unlike traditional power production like thermal and hydro power plants, many of the renewable energy production units can not produce electricity at the desired frequency without power electronic devices. HVDC grids have also been suggested, as a mean to electrically connect geographically distant areas, to reduce the problem of local weather systems influencing the output of many renewable sources. It should thus be quite evident, that good models of power electronic devices are needed, to analyse the system's behaviour, and in particular its stability.

In this report a black-box measurement based modelling technique will be investigated, with its biggest advantage being, that no a priori knowledge of the device under investigation is needed. The thesis starts off with an introduction to the relevant theory, and an introduction to the methodology. In the theory part a suggestion for a new equation set to calculate the model and a stability criterion for the model is presented. The new equation set for calculating the model is also proven to be equivalent with the old one. In the rest of the thesis the focus is on developing a test bench for obtaining the model. Following a step by step approach, starting with simulations on a VSI developed in Simulink, to get acquainted with the methodology. Before a test bench was developed for testing the methodology on dc-dc converters, ultimately leading to tests conducted on a physical VSI.

Preface

This thesis is the final work of the Master programme in Electrical Power Engineering at the Norwegian University of Science and Technology (NTNU).

I would like to thank my supervisor professor Marta Molinas for guidance and help with the thesis, and for setting me in contact with Sintef, which kindly let me use their laboratory. I would also like to thank my co-supervisor Santiago Sanchez Acevedo, for his technical guidance, and all the work he spent helping conducting experiments in the NTNU and Sintef laboratories. The people who borrowed me laboratory equipment, and spent numerous hours in the lab helping using said equipment and conducting the experiments Amir Hayati Soloot, Jon Are Wold Suul and Salvatore D'Arco, I would also like to thank.

At last I would like to thank all my friends and colleagues here at NTNU, who have helped me finishing the thesis, through discussions, technical guidance, and in general being supportive.

Contents

List of Figures

List of Tables

Terms Used in the Text

Subscripts and Superscripts Used in the Text

d	Real axis of the Park transformation
q	Imaginary axis of the Park transformation
0	Zero sequence component
m	Measured data
p	Perturbation
ref	Reference value
i	Refers to the input side of the model
o	Refers to the output side of the model
base	Base value
(p.u)	Per unit value
ph-n	Phase to neutral
ll	Line to line
rms	Root mean square

Parameters Used in the Text

List of Abbreviations

AC	Alternating Current
DC	Direct Current
IFBW	Intermediate Frequency Bandwidth
VSI	Voltage Source Inverter
VSC	Voltage Source Converter
PLL	Phase Locked Loop
PCC	Point of Common Coupling
PI	Proportional Integral
Nan	Not a Number
Inf	Infinite

Notation

Bold fonts like **v** represent vectors, whereas normal fonts like v represent scalars. The following notation represent small signal values and average values respectively $\tilde{v}$ and $\bar{v}$. As is common in electrical engineering the imaginary number is represented by j.

Chapter 1

Introduction

1.1 Backround

Through the Kyoto protocol most countries are obliged to reduce their CO_2 emissions[1]. Besides actions to reduce the emissions in the industrialized world, more renewables have to be integrated in to the energy mix. In many aspects renewables differ from traditional power production units like thermal plants and hydropower plants, which quite easily can produce power at the desired frequency. This is not the case for photo-voltaic, where dc-current is produced, or wind turbines, where it is desirable to vary the rotational speed to compensate the varying wind speeds. As the power system requires a given frequency, power electronic converters are needed. Renewables are also by nature a more fluctuating energy source, compared to traditional power production units. The fluctuating nature can be compensated by long distance power transmission, overreaching the geographical distribution of weather systems, utilizing HVDC transmission[2]. Other means of overcoming the fluctuations are proposed in [3] and [4], where a HVDC grid in the North Sea is proposed, together with offshore windfarms. HVDC lines also possess other advantageous features, like reactive power control, asynchronous interconnections and stabilization in power systems[5]. The backbone of all these HVDC systems, regardless of the switching devices, are power electronic devices. Power electronic converters, however, introduce new properties to the power system[6], and further research is needed to study these effects. This increases the need for reliable, and easy obtainable models of power electronic converters.

Well established methods, like average state modelling, require good knowledge of the converter in question, which is not always available. Especially detailed data of the control system can be hard to obtain. However,

these data are crucial to get a correct converter model, as the dynamics are heavily dependent on the control system. Amongst the approaches, which are being investigated by researchers to overcome these problems, are measurement based techniques. One popular approach is to separate the system investigated into a source and load system, the stability can then be assessed by applying Nyquist's criterion on the measured admittances and impedances [7] [8]. Different approaches to obtaining the admittances and impedances exist, and in [9][10] it is described how to obtain the admittances and impedances by doing current injections at various frequencies, and recording the responses. It is also possible to obtain the impedances by looking at a system's response to load steps, as proposed in [11].

Similar to the impedance based modelling, there also exist measurement based behavioural models of power electronic converters. Where one aims at, describing the input output behaviour of a converter, by using a given set of equations. The advantage being, that obtained models can be used in simulations in the same way as average based models. The difference being, that measurement behavioural models can be obtained through black-box modelling. Black-box modelling refers to a modelling technique, where no a priori knowledge of the device is known, and the measured data is tried fitted to a given model [12].

One class of black box-modelling techniques assumes the model to be a multiport system. Within this field work on so called multiport models is presented for spacecraft power systems, in [13] and [14], where it is shown that spacecraft power systems can be simulated using this method. The advantage being that each converter, which consists of multiple elements can be represented by a relatively simple circuit. Further work within this field was done in [15], where a black-box approach was used to model open and closed loop dc-dc converters. The model chosen was the g-parameters two-port model, also referred to as a two-port hybrid model. Both the methodology and the verification is presented. The methodology was further improved in [16] and [17], where a decoupling matrix was proposed for dc-dc converters and general converters respectively. The advantage of the decoupling matrix is, that the behavioural model can be obtained, while the converter is operating at its nominal operating point. Further examples and verifications are provided in [18], along with [19], where a stability criterion also is provided.

1.2 Outline

Continuing the work on g-parameters modelling using the decoupling proposed in [17], this thesis aims at investigating different aspects of the mod-

elling. In chapter 2 a theoretical introduction to the methodology is given. To further familiarize the reader with the subject, and to present the equations and theory relevant to the modelling. Having the theoretical background in place, chapter 3 tries to shed some light on the issue of implementing the theory in a general methodology. Subjects covered are data collection, disturbance injection, model fitting, and how to implement the model in Simulink. The goal is that the educated reader should be able, to obtain and run g-parameter models, after reading chapters 2 and 3. The thesis continues in a step by step approach, with the ultimate goal of verifying the methodology for a physical VSI. VSIs are especially interesting in the context of modelling, as they have both a dc- and ac-side. Different aspects related to perturbation injection and data collection, for both dc and ac, can thus be investigated. To gain insight in the challenges faced in the methodology, it was decided to start off with a VSI implemented in Simulink. This approach poses numerous advantages, like full control over the design and easy integration between measurements and post processing in Matlab. Keeping this in mind the complete control system of the Simulink implemented VSI is presented in chapter 4. The approach specific to the modelling of the VSI, along with the results are given in chapter 5. Mastering the modelling of a VSI in Simulink, it is time to move on to a physical system. Although insight in the issues related to the modelling was gained from the Simulink model, it was decided to start of with a simple open loop boost converter. Giving valuable experience in the practical aspects of perturbation injection and data collection. The results and discussion of the modelling of an open and closed loop dc-dc converter are presented in chapters 6 and 7. Thereafter the methodology is applied on a physical VSI, utilizing the knowledge gained through the previous chapters. Finally the work is concluded in chapter 9 and further work is suggested in chapter 10

Chapter 2

Analytical Basis for the Modelling Approach

In this chapter a brief analytical basis for two-port circuits in general and the chosen model in particular is presented.

2.1 General Two-Port Circuits

A two port circuit is a circuit model with two terminal pairs as seen in figure 2.1. Each terminal pair is referred to as a port. The four restrictions on a two-port circuit is given below according to [20].

- No energy can be stored within the circuit.
- There can be no independent sources within the circuit.
- The current into a port must equal the current out, meaning $\mathbf{i}_i = \mathbf{i}'_i$ and $\mathbf{i}_o = \mathbf{i}'_o$.
- All external connections must be made to either the input or the output port.

Mathematically a two-port circuit can be described by two equations, and it is possible to construct 6 such different sets of equations. In this project the set of equations referred to as "g-parameters" are being used, which will be further explained in the next section.

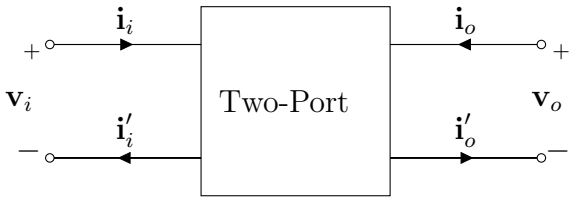

Figure 2.1: A general two-port circuit

2.2 The Model Investigated in this Report

This report is an investigation of the methodology proposed in [17], thus this report will be complying as close as possible to the notation and equations introduced, in this report. The model investigated models converters as two, three or four port models, depending on the number of phases in the input and output of the system modelled. For each side one port is needed for every phase, three phases are converted to dq-coordinates, neglecting the zero sequence. In other words an ac-ac converter will be modelled as a four port model, and a dc-dc converter as a two-port model. In this report all these models will be referred to as a g-parameters models.

The model is quite general, as it is presented here, and can be applied to ac-ac, ac-dc, dc-ac and dc-dc converters. It is a so called small signal model, meaning that the converter under investigation is operating close to its nominal values during the measurements. Only deviating from the nominal values, by a small value, due to small perturbations injected in to the system. Thus the system values can be seen as the superposition of a steady state value, and a small signal value, as seen in equation (2.1).

$$\begin{bmatrix} \mathbf{i}_i \\ \mathbf{i}_o \end{bmatrix} = \begin{bmatrix} \mathbf{I}_i + \tilde{\mathbf{i}}_i \\ \mathbf{I}_o + \tilde{\mathbf{i}}_o \end{bmatrix} \tag{2.1}$$

Having presented the small signal values, the rest of the model's equations can safely be presented.

$$\begin{bmatrix} \tilde{\mathbf{v}}_o \\ \tilde{\mathbf{i}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{G}_o(s) & \mathbf{Z}_o(s) \\ \mathbf{Y}_i(s) & \mathbf{H}_i(s) \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{v}}_i \\ \tilde{\mathbf{i}}_o \end{bmatrix} \tag{2.2}$$

where

$$\tilde{\mathbf{v}}_o = \begin{bmatrix} \tilde{v}_{od} \\ \tilde{v}_{oq} \end{bmatrix}, \tilde{\mathbf{v}}_i = \begin{bmatrix} \tilde{v}_{id} \\ \tilde{v}_{iq} \end{bmatrix}, \tilde{\mathbf{i}}_o = \begin{bmatrix} \tilde{i}_{od} \\ \tilde{i}_{oq} \end{bmatrix}, \tilde{\mathbf{i}}_i = \begin{bmatrix} \tilde{i}_{id} \\ \tilde{i}_{iq} \end{bmatrix} \tag{2.3}$$

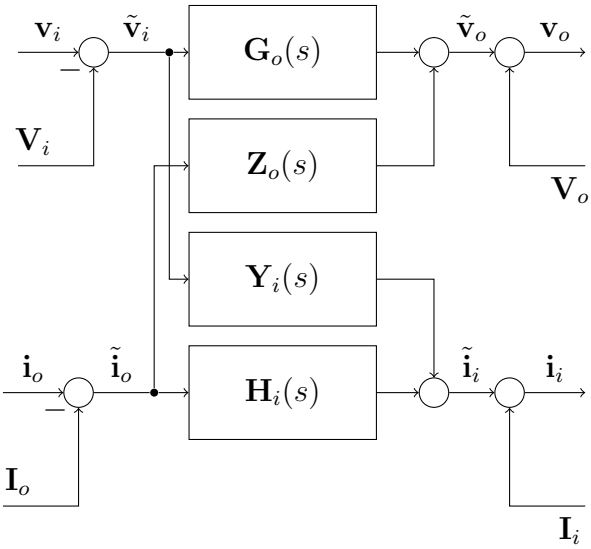

Figure 2.2: Schematic drawing of the model

and

$$\mathbf{G}_o = \begin{bmatrix} G_{dd} & G_{dq} \\ G_{qd} & G_{qq} \end{bmatrix}, \mathbf{Z}_o = \begin{bmatrix} Z_{dd} & Z_{dq} \\ Z_{qd} & Z_{qq} \end{bmatrix}$$
$$\mathbf{Y}_i = \begin{bmatrix} Y_{dd} & Y_{dq} \\ Y_{qd} & Y_{qq} \end{bmatrix}, \mathbf{H}_i = \begin{bmatrix} H_{dd} & H_{dq} \\ H_{qd} & H_{qq} \end{bmatrix} \tag{2.4}$$

The G-parameters are referred to as audio susceptibility $\mathbf{G}_o$, output impedances $\mathbf{Z}_o$, input admittance $\mathbf{Y}_i$, and back current gain $\mathbf{H}_i$. For simulations the mathematical equations will be represented by a block diagram in simulink. Substituting equation (2.1) into (2.2), gives the following block diagram presented in figure 2.2

As can be seen from (2.2), the model describes the system as a superposition, assuming the system to be linear. One can thus measure the transfer functions directly by setting the appropriate values to zero, which for a circuit means a short or an open circuit. This approach however, would not give satisfactory results. The model suggested is a small signal model, and applying shorts or opening terminals, would bring the converter far away from its operating point. In [17] it is suggested to define a set of measured transfer functions, which through substitution yields equation (2.5).

$$\begin{bmatrix} \mathbf{G}_o & \mathbf{Z}_o \\ \mathbf{Y}_i & \mathbf{H}_i \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{om} & \mathbf{Z}_{om} \\ \mathbf{Y}_{im} & \mathbf{H}_{im} \end{bmatrix} \cdot \mathbf{D}^{-1} \tag{2.5}$$

where the decoupling matrix is given by (2.6).

$$\mathbf{D} = \begin{bmatrix} 1 & TV_{dqm} & TR_{ddm} & TR_{dqm} \\ TV_{qdm} & 1 & TR_{qdm} & TR_{qqm} \\ TG_{ddm} & TG_{dqm} & 1 & TI_{dqm} \\ TG_{qdm} & TG_{qqm} & TI_{qdm} & 1 \end{bmatrix} \tag{2.6}$$

In the equations given above the measured transfer functions are denoted with a subscripted m. They are obtained through four independent measurements, which are given below.

Measurement I

$$\begin{aligned} G_{ddm} &= \frac{\tilde{v}_{od}^{I}}{\tilde{v}_{id}^{I}}, G_{qdm} = \frac{\tilde{v}_{oq}^{I}}{\tilde{v}_{id}^{I}}, Y_{ddm} = \frac{\tilde{i}_{id}^{I}}{\tilde{v}_{id}^{I}}, Y_{qdm} = \frac{\tilde{i}_{iq}^{I}}{\tilde{v}_{id}^{I}}, \\ TG_{ddm} &= \frac{\tilde{i}_{od}^{I}}{\tilde{v}_{id}^{I}}, TG_{qdm} = \frac{\tilde{i}_{oq}^{I}}{\tilde{v}_{id}^{I}}, TV_{qdm} = \frac{\tilde{v}_{iq}^{I}}{\tilde{v}_{id}^{I}} \end{aligned} \tag{2.7}$$

Measurement II

$$\begin{aligned} G_{dqm} &= \frac{\tilde{v}_{od}^{II}}{\tilde{v}_{iq}^{II}}, G_{qqm} = \frac{\tilde{v}_{oq}^{II}}{\tilde{v}_{iq}^{II}}, Y_{dqm} = \frac{\tilde{i}_{id}^{II}}{\tilde{v}_{iq}^{II}}, Y_{qqm} = \frac{\tilde{i}_{iq}^{II}}{\tilde{v}_{iq}^{II}}, \\ TG_{dqm} &= \frac{\tilde{i}_{od}^{II}}{\tilde{v}_{iq}^{II}}, TG_{qqm} = \frac{\tilde{i}_{oq}^{II}}{\tilde{v}_{iq}^{II}}, TV_{dqm} = \frac{\tilde{v}_{id}^{II}}{\tilde{v}_{iq}^{II}} \end{aligned} \tag{2.8}$$

Measurement III

$$\begin{aligned} Z_{ddm} &= \frac{\tilde{v}_{od}^{III}}{\tilde{i}_{od}^{III}}, Z_{qdm} = \frac{\tilde{v}_{oq}^{III}}{\tilde{i}_{od}^{III}}, H_{ddm} = \frac{\tilde{i}_{id}^{III}}{\tilde{i}_{od}^{III}}, H_{qdm} = \frac{\tilde{i}_{iq}^{III}}{\tilde{i}_{od}^{III}}, \\ TR_{ddm} &= \frac{\tilde{v}_{id}^{III}}{\tilde{i}_{od}^{III}}, TR_{qdm} = \frac{\tilde{v}_{iq}^{III}}{\tilde{i}_{od}^{III}}, TI_{qdm} = \frac{\tilde{i}_{oq}^{III}}{\tilde{i}_{od}^{III}} \end{aligned} \tag{2.9}$$

Measurement IV

$$\begin{aligned} Z_{dqm} &= \frac{\tilde{v}_{od}^{IV}}{\tilde{i}_{oq}^{IV}}, Z_{qqm} = \frac{\tilde{v}_{oq}^{IV}}{\tilde{i}_{oq}^{IV}}, H_{dqm} = \frac{\tilde{i}_{id}^{IV}}{\tilde{i}_{oq}^{IV}}, H_{qqm} = \frac{\tilde{i}_{iq}^{IV}}{\tilde{i}_{oq}^{IV}}, \\ TR_{dqm} &= \frac{\tilde{v}_{id}^{IV}}{\tilde{i}_{oq}^{IV}}, TR_{qqm} = \frac{\tilde{v}_{iq}^{III}}{\tilde{i}_{oq}^{III}}, TI_{dqm} = \frac{\tilde{i}_{od}^{IV}}{\tilde{i}_{oq}^{IV}} \end{aligned} \tag{2.10}$$

2.3 Proposal for an Alternative Decoupling Formulation

Although the above stated decoupling matrix is useful, and as will be shown later, especially if one has a device, which can directly measure transfer functions. It might in some cases be useful to use another formulation. The logic behind the derivation will be given here, and is based upon basic linear algebra. Inspecting equation (2.4), one will see that there is 16 g-parameters transfer functions. Linear system theory gives us that 16 linearly independent equations are needed to identify these 16 g-parameters. From this we conclude that we need four measurement sets, which should be independent, and conducted such that the g-parameters equations are valid. Denoting the measurement sets I, II, III and IV, it can be shown that the following has to be valid.

$$\begin{bmatrix} \tilde{\mathbf{v}}_o^I & \tilde{\mathbf{v}}_o^{II} & \tilde{\mathbf{v}}_o^{III} & \tilde{\mathbf{v}}_o^{IV} \\ \tilde{\mathbf{i}}_i^I & \tilde{\mathbf{i}}_i^{II} & \tilde{\mathbf{i}}_i^{III} & \tilde{\mathbf{i}}_i^{IV} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_o(s) & \mathbf{Z}_o(s) \\ \mathbf{Y}_i(s) & \mathbf{H}_i(s) \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{v}}_i^I & \tilde{\mathbf{v}}_i^{II} & \tilde{\mathbf{v}}_i^{III} & \tilde{\mathbf{v}}_i^{IV} \\ \tilde{\mathbf{i}}_o^I & \tilde{\mathbf{i}}_o^{II} & \tilde{\mathbf{i}}_o^{III} & \tilde{\mathbf{i}}_o^{IV} \end{bmatrix} \tag{2.11}$$

The g-parameters can then easily be identified by the following equation.

$$\begin{bmatrix} \mathbf{G}_o(s) & \mathbf{Z}_o(s) \\ \mathbf{Y}_i(s) & \mathbf{H}_i(s) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{v}}_o^I & \tilde{\mathbf{v}}_o^{II} & \tilde{\mathbf{v}}_o^{III} & \tilde{\mathbf{v}}_o^{IV} \\ \tilde{\mathbf{i}}_i^I & \tilde{\mathbf{i}}_i^{II} & \tilde{\mathbf{i}}_i^{III} & \tilde{\mathbf{i}}_i^{IV} \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{v}}_i^I & \tilde{\mathbf{v}}_i^{II} & \tilde{\mathbf{v}}_i^{III} & \tilde{\mathbf{v}}_i^{IV} \\ \tilde{\mathbf{i}}_o^I & \tilde{\mathbf{i}}_o^{II} & \tilde{\mathbf{i}}_o^{III} & \tilde{\mathbf{i}}_o^{IV} \end{bmatrix}^{-1} \tag{2.12}$$

As can be seen this decoupling technique removes the need of introducing the terminated transfer functions, and gives the unterminated transfer functions directly from the measured quantities. It will also be shown that the two techniques are equivalent. The proof goes as follows. Multiplying equation (2.5) by D^{-1}, one gets.

$$\begin{bmatrix} \mathbf{G}_{om} & \mathbf{Z}_{om} \\ \mathbf{Y}_{im} & \mathbf{H}_{im} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_o & \mathbf{Z}_o \\ \mathbf{Y}_i & \mathbf{H}_i \end{bmatrix} \cdot \mathbf{D} \tag{2.13}$$

Then one has to define the following vector

$$\mathbf{m} = \begin{bmatrix} \tilde{v}_{id}^I \\ \tilde{v}_{iq}^{II} \\ \tilde{i}_{od}^{III} \\ \tilde{i}_{oq}^{IV} \end{bmatrix} \tag{2.14}$$

Multiplying equation (2.13) by (2.14) and the identity matrix gives.

$$\begin{bmatrix} G_{ddm} & G_{dqm} & Z_{ddm} & Z_{dqm} \\ G_{qdm} & G_{qqm} & Z_{qdm} & Z_{qqm} \\ Y_{ddm} & Y_{dqm} & H_{ddm} & H_{dqm} \\ Y_{qdm} & Y_{qqm} & H_{qdm} & H_{qqm} \end{bmatrix} \cdot \begin{bmatrix} \tilde{v}_{id}^{I} & 0 & 0 & 0 \\ 0 & \tilde{v}_{iq}^{II} & 0 & 0 \\ 0 & 0 & \tilde{i}_{od}^{III} & 0 \\ 0 & 0 & 0 & \tilde{i}_{oq}^{IV} \end{bmatrix} =$$

$$\begin{bmatrix} G_{dd} & G_{dq} & Z_{dd} & Z_{dq} \\ G_{qd} & G_{qq} & Z_{qd} & Z_{qq} \\ Y_{dd} & Y_{dq} & H_{dd} & H_{dq} \\ Y_{qd} & Y_{qq} & H_{qd} & H_{qq} \end{bmatrix} \cdot \begin{bmatrix} 1 & TV_{dqm} & TR_{ddm} & TR_{dqm} \\ TV_{qdm} & 1 & TR_{qdm} & TR_{qqm} \\ TG_{ddm} & TG_{dqm} & 1 & TI_{dqm} \\ TG_{qdm} & TG_{qqm} & TI_{qdm} & 1 \end{bmatrix} \cdot \begin{bmatrix} \tilde{v}_{id}^{I} & 0 & 0 & 0 \\ 0 & \tilde{v}_{iq}^{II} & 0 & 0 \\ 0 & 0 & \tilde{i}_{od}^{III} & 0 \\ 0 & 0 & 0 & \tilde{i}_{oq}^{IV} \end{bmatrix} =$$

$$\begin{bmatrix} \tilde{v}_{od}^{I} & \tilde{v}_{od}^{II} & \tilde{v}_{od}^{III} & \tilde{v}_{od}^{IV} \\ \tilde{v}_{oq}^{I} & \tilde{v}_{oq}^{II} & \tilde{v}_{oq}^{III} & \tilde{v}_{oq}^{IV} \\ \tilde{i}_{id}^{I} & \tilde{i}_{id}^{II} & \tilde{i}_{id}^{III} & \tilde{i}_{id}^{IV} \\ \tilde{i}_{iq}^{I} & \tilde{i}_{iq}^{II} & \tilde{i}_{iq}^{III} & \tilde{i}_{iq}^{IV} \end{bmatrix} = \begin{bmatrix} G_{dd} & G_{dq} & Z_{dd} & Z_{dq} \\ G_{qd} & G_{qq} & Z_{qd} & Z_{qq} \\ Y_{dd} & Y_{dq} & H_{dd} & H_{dq} \\ Y_{qd} & Y_{qq} & H_{qd} & H_{qq} \end{bmatrix} \cdot \begin{bmatrix} \tilde{v}_{id}^{I} & \tilde{v}_{id}^{II} & \tilde{v}_{id}^{III} & \tilde{v}_{id}^{IV} \\ \tilde{v}_{iq}^{I} & \tilde{v}_{iq}^{II} & \tilde{v}_{iq}^{III} & \tilde{v}_{iq}^{IV} \\ \tilde{i}_{od}^{I} & \tilde{i}_{od}^{II} & \tilde{i}_{od}^{III} & \tilde{i}_{od}^{IV} \\ \tilde{i}_{oq}^{I} & \tilde{i}_{oq}^{II} & \tilde{i}_{oq}^{III} & \tilde{i}_{oq}^{IV} \end{bmatrix} \tag{2.15}$$

Which is the same as (2.12). Removing the terminated transfer functions from the decoupling, also makes it easier to see the relation to other measurement based techniques. For instance, the output impedance model in[9] can be calculated by.

$$\begin{bmatrix} \mathbf{Z}_{dd}(s) & \mathbf{Z}_{dq}(s) \\ \mathbf{Z}_{qd}(s) & \mathbf{Z}_{qq}(s) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{v}}_{od}^{III} & \tilde{\mathbf{v}}_{od}^{IV} \\ \tilde{\mathbf{v}}_{od}^{III} & \tilde{\mathbf{v}}_{oq}^{IV} \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{i}}_{od}^{III} & \tilde{\mathbf{i}}_{od}^{IV} \\ \tilde{\mathbf{i}}_{oq}^{III} & \tilde{\mathbf{i}}_{oq}^{IV} \end{bmatrix}^{-1} \tag{2.16}$$

2.4 Stability Analysis on the Two-Port Model

To do stability analysis on the two-port model, the simple equivalent circuit of the two-port model presented in figure 2.3 will be used to derive expressions for the input and output voltages. The presentation of the circuit for the q-phase is omitted, since it is symmetrical. The part of the circuit representing the model is enclosed within the rectangle, whereas the input source and its impedance, is to the left of the rectangle, and the load is to the right of the rectangle. For the input side one can observe that it is dependent on the output side current. Thus to ensure stability on the input side, the output side current has to be stable. A similar argument can be made for the output side, giving the conclusion that the input side expression should be made for the voltage. Then simple circuit theory gives the four equations for stability analysis.

$$\tilde{v}_{id} = (\tilde{v}_{sd} - \tilde{v}_{iq} Y_{dq} Z_{id} - \tilde{i}_{od} H_{dd} Z_{id} - \tilde{i}_{oq} H_{dq} Z_{id}) \cdot \frac{1}{1 + Y_{dd} Z_{id}} \tag{2.17}$$

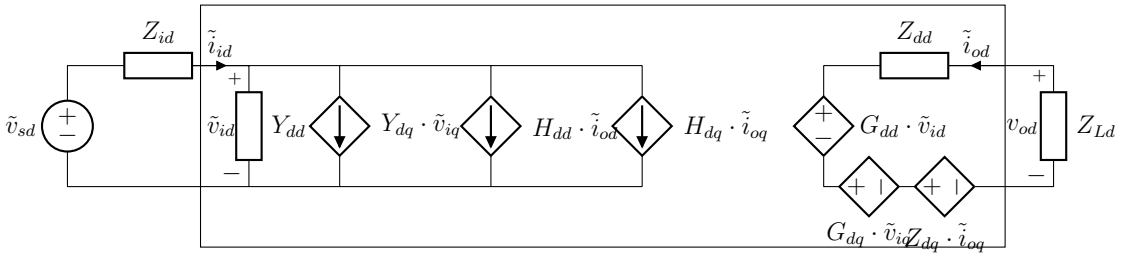

Figure 2.3: Equivalent circuit of the two-port model, for the d-phase

$$\tilde{v}_{iq} = (\tilde{v}_{sq} - \tilde{v}_{id}Y_{qd}Z_{iq} - \tilde{i}_{oq}H_{qq}Z_{iq} - \tilde{i}_{od}H_{qd}Z_{iq}) \cdot \frac{1}{1 + Y_{qq}Z_{iq}} \tag{2.18}$$

$$\tilde{i}_{od} = -\frac{\tilde{v}_{id}G_{dd} + \tilde{v}_{iq}G_{dq} + \tilde{i}_{oq}Z_{dq}}{Z_{Ld}} \cdot \frac{1}{1 + Z_{dd}/Z_{Ld}} \tag{2.19}$$

$$\tilde{i}_{oq} = -\frac{\tilde{v}_{iq}G_{qq} + \tilde{v}_{id}G_{qd} + \tilde{i}_{od}Z_{qd}}{Z_{Lq}} \cdot \frac{1}{1 + Z_{qq}/Z_{Lq}} \tag{2.20}$$

2.4.1 Stability of the Input Voltage Equations

To investigate the stability of equations (2.17) and (2.18), a closer look at equation (2.17) will be provided. As the equations are essentially the same, and the same reasoning can be used on both of them. The criterions for both the equations, will however be stated. In the first factor of this equation, the transfer functions of interest are Y_{dq}, H_{dd} and H_{dq}. In Chapter 3 it will be shown, how these transfer functions are obtained from state space models, which again are obtained from measured data. These transfer functions will be asymptotically input output stable, if the real part of the eigenvalues of the state space models are less than zero[21]. Meaning that the first factor of equation (2.17) is stable for any finite output current and any finite Z_{id}, if the eigenvalues of Y_{dq}, H_{dd} and H_{dq} are in the left half plane. Looking at the second factor of the equation, one can recognize that it resembles the transfer function of a closed loop negative feedback system[22]. For such a transfer function, the stability can be given by Nyquist's criterion. To summarize, the stability is given by the following criteria

Stability of Equation (2.17)

1. The eigenvalues of Y_{dq} are in the left half plane.

2. The eigenvalues of H_{dd} are in the left half plane.
3. The eigenvalues of H_{dq} are in the left half plane.
4. $Y_{dd}Z_{id}$ satisfies Nyquist's criterion.

Stability of Equation (2.18)

1. The eigenvalues of Y_{qd} are in the left half plane.
2. The eigenvalues of H_{qq} are in the left half plane.
3. The eigenvalues of H_{qd} are in the left half plane.
4. $Y_{qq}Z_{iq}$ satisfies Nyquist's criterion.

2.4.2 Stability of the Output Current Equations

The reasoning behind these stability criteria is the same as for the previous ones. It is just important to pay attention to the fact that, the Nyquist criterion is given as the relation between two impedances. Instead of the product of an admittance, and an impedance.

1. The eigenvalues of G_{dd} are in the left half plane.
2. The eigenvalues of G_{dq} are in the left half plane.
3. The eigenvalues of Z_{dq} are in the left half plane.
4. Z_{dd}/Z_{Ld} satisfies Nyquist's criterion.

Stability of Equation (2.20)

1. The eigenvalues of G_{qq} are in the left half plane.
2. The eigenvalues of G_{qd} are in the left half plane.
3. The eigenvalues of Z_{dq} are in the left half plane.
4. Z_{qq}/Z_{Lq} satisfies Nyquist's criterion.

For each of the four equations above, there are four criteria required for stability. In each of the cases, if one of the three first criteria are not met, the model will not be stable, regardless of what it is connected to. The fourth criteria is particularly useful, since it indicates what the model can be connected to. For instance if one is analysing a micro-grid, by representing each converter in the grid as a two-port model, it is of interest to see if the models can be connected.

Chapter 3

General Implementation of the G-parmeters Modelling

Knowing the analytical background is not everything in engineering, a methodology based upon the theory is also needed. One of the main selling points of the chosen modelling approach is the generality. In theory it can model all converter topologies meeting the two-port circuit requirements. In this chapter a general description of the methodology used for obtaining g-parameters models are described.

3.1 Data Collection

Although the converter topologies might differ, the method for injecting perturbations can be generalized. The idea is basically to let the converter run at a stable set point, and then inject a perturbation. The perturbation can be injected either in series or parallel, depending on what is the most practical. For a dc-terminal the implementation is straight forward, however for an ac-terminal one needs to make some choices. Two general approaches were used in this paper, both which allows for injecting in parallel or series.

The first approach considers the system in dq-coordinates, and aims at injecting the perturbations purely along one axis. In the case of an ac-ac converter it means, that during the first measurement set the perturbation will be injected into the input d-axis, into the input q-axis during the second measurement set, into the output d-axis during the third measurement set and into the output q-axis during the fourth and last measurement set. The first and fourth step in this procedure is demonstrated in figures 3.1 and 3.2. There are different means of injecting disturbances purely into on axis, one of which is presented in [11]. Where it is shown how one can identify

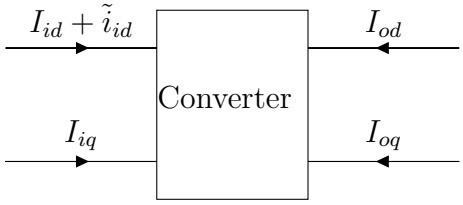

Figure 3.1: Perturbation injected at the d-phase of the input

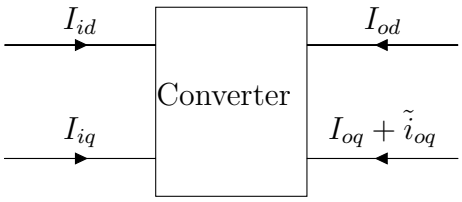

Figure 3.2: Perturbation injected at the q-phase of the output

the output impedance of a VSI, by applying load steps and measuring the transient response. The output impedance can then be identified, by for instance using the Matlab System Identification Toolbox. However, more work is needed to be able to use the decoupling procedures in this report with the mentioned technique.

In this report it was chosen to take advantage of the converter's control system. Assuming it to have an accessible PLL, which can be used to synchronize the perturbations with one of the axes. The obvious drawbacks of this approach are the complexity and the need of a PLL, and access to the converter's control system. A simple Simulink subsystem implementation of this approach is presented in figure 3.3. Although simple to implement in software, the approach is rather complicated to implement in hardware.

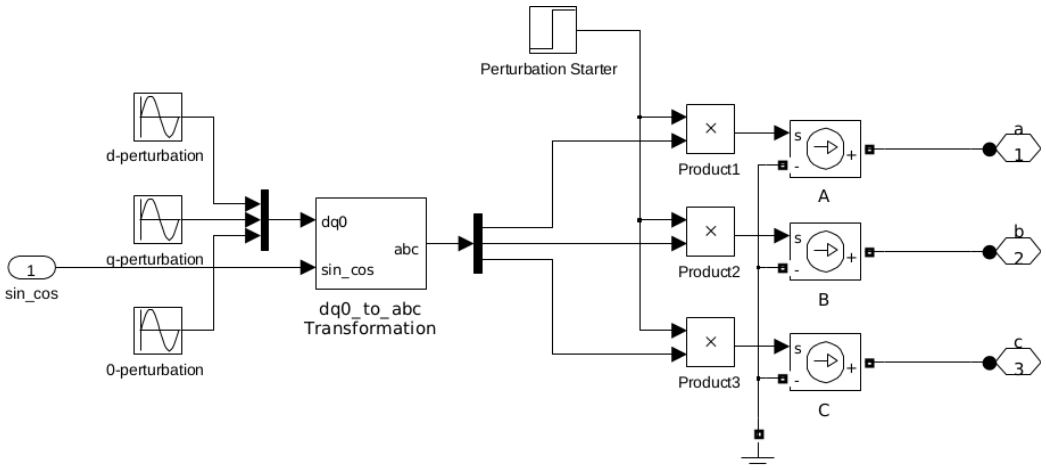

Figure 3.3: Simulink subsystem used to inject perturbations

The second technique for injecting perturbations into the ac-side used in this report, is the one proposed in [9]. Where line to line current injection is proposed. The idea behind this technique, is that instead of injecting in

the d- and q-axis separately, the independent measurements are obtained by, first injecting a positive sequence perturbation, and then a negative sequence perturbation, as shown in equations (3.1) and (3.2). Where I_m is the amplitude of the injected current. Equation (3.1) corresponds to injecting a positive sequence perturbation into the system, and (3.2) corresponds to injecting a negative sequence perturbation.

$$i_p^I = I_m cos((\omega_e + \omega_p)t) \tag{3.1}$$

$$i_p^{II} = I_m cos(|-\omega_e + \omega_p|t) \tag{3.2}$$

In a similar fashion it is also possible to inject voltages in series, where the equations for the injected voltages are.

$$v_p^I = V_m cos((\omega_e + \omega_p)t) \tag{3.3}$$

$$v_p^{II} = V_m cos(|-\omega_e + \omega_p|t) \tag{3.4}$$

The advantages of this scheme are many, first and foremost the complexity is reduced, furthermore the amount of required equipment is also reduced. In figures 3.4 and 3.5, the injection technique is presented for both current and voltage injection.

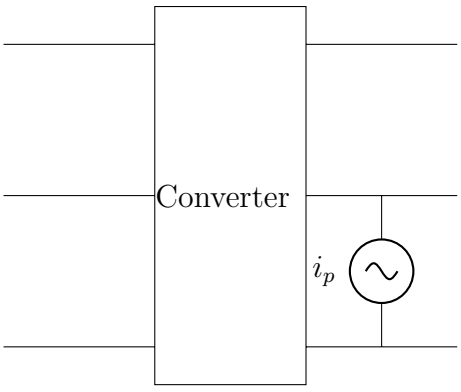

Figure 3.4: Unbalanced current injection

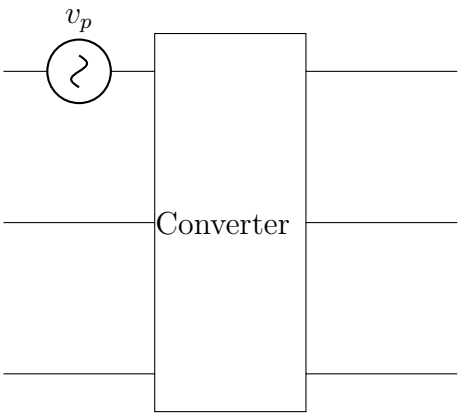

Figure 3.5: Unbalanced voltage injection

Depending on the measurement equipment available, some choices might have to be done. There exist network analysers, which handle most of the issues soon to be discussed. They are however expensive, and for the purpose of this report, simpler and cheaper solutions is sufficient. When planning a

test, even with network analysers, one will have to decide which frequencies to record. Both equipment for generating perturbations and recording the responses have lower and upper frequency boundaries. To decide the frequency of the first perturbation to be injected, the lower frequency of the available equipment is a good starting point. For frequencies above half of the switching frequency, aliasing effects should be considered[23]. Hence only frequencies below half of the switching frequency will be considered in this paper, giving the upper boundary of the perturbations' frequency.

One might also need to consider which frequencies to record, and the sampling frequency to use. If one is in possession of measurement equipment, which directly measures transfer functions, it will normally handle these issues automatically. If one, however, is measuring the voltages and currents in the time domain, one should make sure that a suitable number of samples and sample times are chosen. This is due to the fact that the data in the time domain will have to be fourier transformed into the frequency domain. For instance if one is using Matlab for doing the fourier transform, the following definition of the DFT is used[24].

$$y = \sum_{k=0}^{N-1} x \cdot e^{-j2\pi k/N} \tag{3.5}$$

Where N is the number of samples. Defining the sampling time T_s as the time between every sample and $f_s = 1/T_s$, one can predict the spacing between the samples in the fourier response. This relation is given in (3.6), and is referred to as the frequency resolution.

$$df = \frac{f_s}{N} \tag{3.6}$$

3.2 The Model Construction

After all the coupled transfer functions are measured, two steps remain before the final model is obtained. Namely calculating the decoupled transfer functions, and fitting a model to the measured data. Obtaining the decoupled transfer functions is easily accomplished with equation (2.5) or (2.12). For the fitting of the transfer functions, the matlab provided function fitfrd is used, which tries to fit a state space model to a given set of measurements. To determine the order of the model, a heuristic error and trial approach is used. In figure 3.6 the crucial steps to the approach are shown. Fitted models with varying orders are plotted in the same figure as the measured data, referred to as the object in the figure. Then the model with the best fit

can be chosen. To further help choosing the correct order, the Hankel values are also plotted. States with considerably lower Hankel values compared to the others can often be neglected.

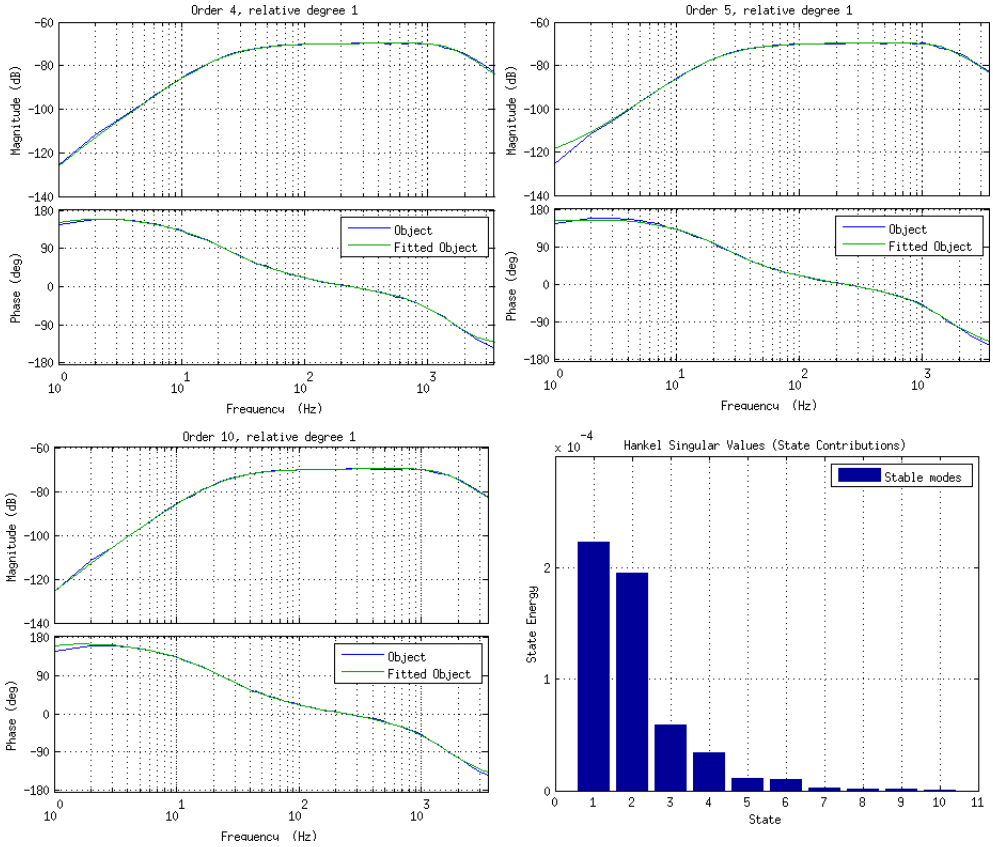

Figure 3.6: Plots used for creating model transfer functions.

In some cases it might be difficult to fit transfer functions to the measured data due to noise in the measurements. To post processing techniques are proposed to to deal with this. Before calculating the g-parameters the measured data can be filtered using an intermediate frequency and a low pass filter, or by using a separate bandpass filter suitable for each of the frequencies in the measurement sets. If the g-parameters are still noisy. A moving average filter can be applied to the g-parameters after they have been calculated.

3.3 Simulink Implementation

To be useful the model has to somehow be implemented in software. In this report the approach is to implement it using the toolbox SimPowerSystems, which is part of the Simscape Simulink toolbox. The first step was to implement the block diagram of figure 2.2 in Simulink. In figure 3.7 the VSI implementation of this block diagram is shown, which illustrates important aspects of the implementation. For instance it shows how to make the block diagram, for both an ac-side and a dc-side. Starting with the input variables to the left, one can see that the steady state values are subtracted. Followed by gains, which might be useful if the transfer functions are in per unit. The next elements in the figure are the transfer functions. The implementation of the transfer functions are illustrated by the implementation of the output impedance presented in figure 3.8. It shows an example on how one can work with vectorized signals in Simulink, to get a quite straightforward implementation of the transfer function. Moving to the right of the transfer functions in figure 3.7, there are addition blocks and gains. The addition blocks come directly from the equations, whereas the gains are to go from per unit till SI-units if desired. For the output voltage, one can see that a constant is added to the d-phase. The added constant is the reference voltage, if needed the same can be done for the q-phase. The reference could also have been added, at this point to the input current. It was, however chosen to add it by using a current source, as can be seen in figure 3.9.

Figure 3.9 shows how the g-parameters block diagram can be connected to electrical sources, to facilitate connection with other circuit elements. The solution shown here is valid for converters working in islanded mode, since the frequency is constant and provided by the sources to the right in the picture. The gain block in the figure is just to get the right sign on the current.

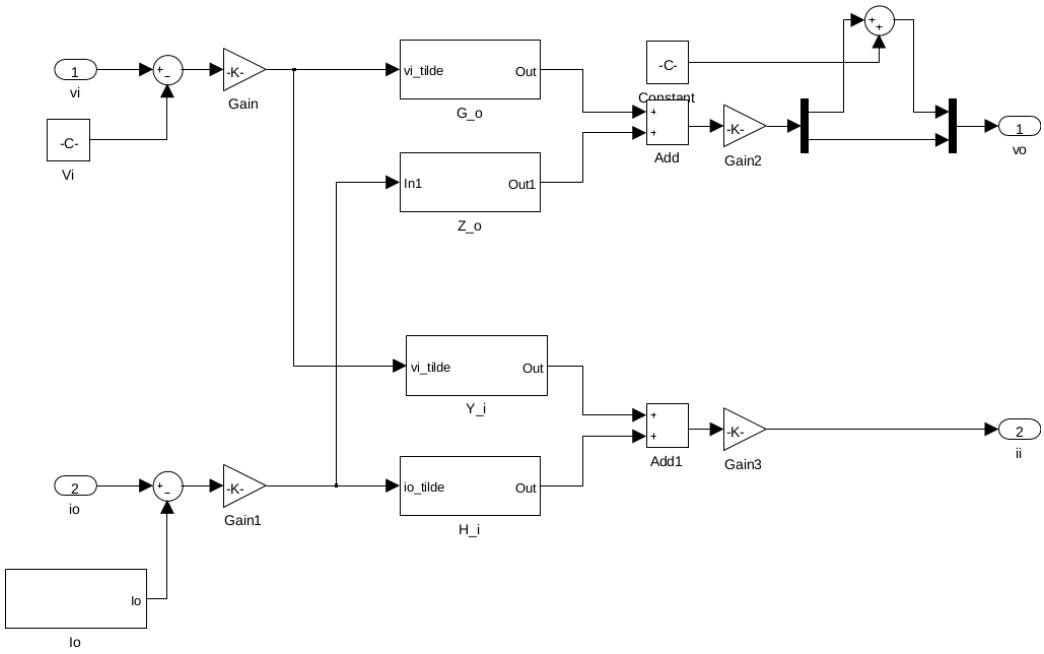

Figure 3.7: Simulink block diagram for a VSI

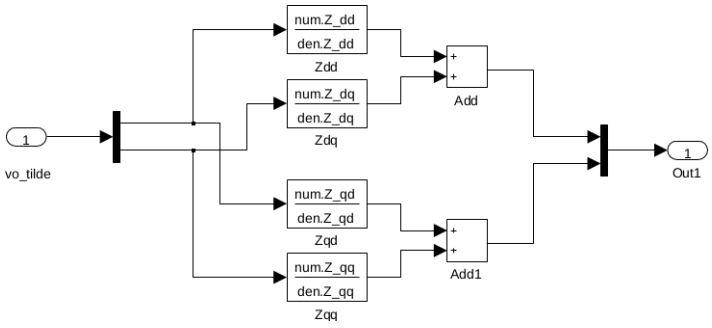

Figure 3.8: Implementation of the output impedance in simulink

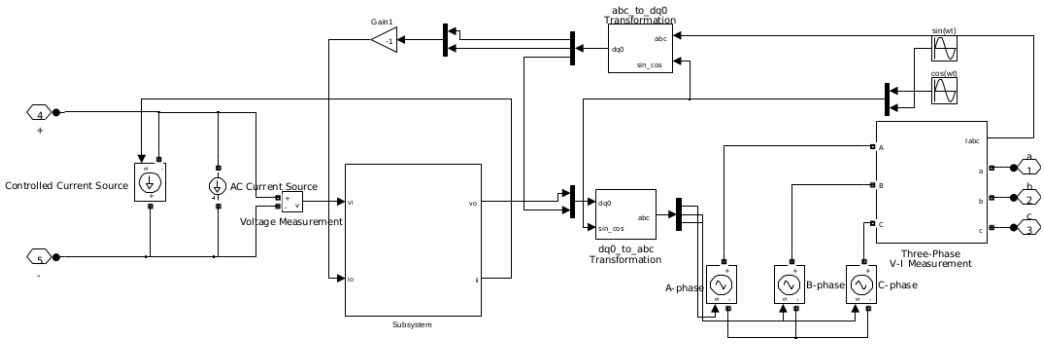

Figure 3.9: Connecting the VSI block diagram to an electrical circuit

Chapter 4

Construction of a Voltage Source Inverter in Simulink

To test the g-parameters modelling approach, a voltage source inverter, working in islanded mode, was constructed in Simulink. Although the g-parameters method is supposed to be a so called black-box modelling technique, it might be useful to have complete knowledge of the system being investigated, to gain further insight into the methodology.

4.1 The Control System's Structure

A cascaded system consisting of three control loops were constructed. The cascade consists of an inner current controller and an outer voltage controller. Since a vector oriented control is being used, Park's transformation needs to be applied. To do Park's transformation the rotational speed and position of the rotating abc-frame is needed, which can be obtained by using a PLL. The PLL is the third control loop in the control system, unlike the two others the one provided by the Simulink toolbox SimPowerSystems is being used. The elements and values considered by the control system are shown in figure 4.1.

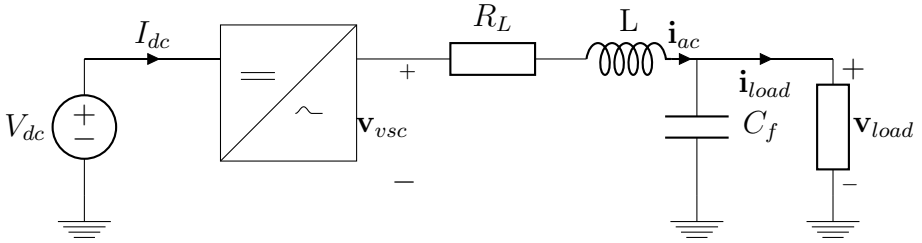

Figure 4.1: System configuration

4.2 The Inner Current Control Loop

The inner current control controls the current $\mathbf{i}_{ac}$, and its dynamics in the dq-fram are described by (4.1).

$$\mathbf{v}_{vsc} = \mathbf{i}_{ac}R_L + L\frac{d\mathbf{i}_{ac}}{dt} + j\omega L\mathbf{i}_{ac} + \mathbf{v}_{load} \tag{4.1}$$

Equation (4.1) contains not only the current, but also the voltage. Further complicating the matter, is the fact that it contains cross-couplings between the d and q axis. An easier controllable system can be obtained through considering the two last terms of (4.1) as disturbances.

4.2.1 The Inner Current Controller's Transfer Functions

Before starting with the tuning of the inner control loop, its transfer functions are presented and explained. In figure 4.2 the system is presented graphically in a block diagram, visualizing the mathematical connection between the transfer functions.

The Transfer Function of the PWM

The PWM is modelled as a first order element with a time constant of $T_a = T_s/2$ [25][26], where T_s is the switching time of the PWM, giving the following transfer function.

$$h_{PWM}(s) = \frac{1}{1 + sT_a} \tag{4.2}$$

The Feed Forward Transfer Functions

The load voltage and cross-coupling terms are considered as disturbances acting on the system, through the transfer functions $h_{v1}(s)$ and $h_{v2}(s)$. The

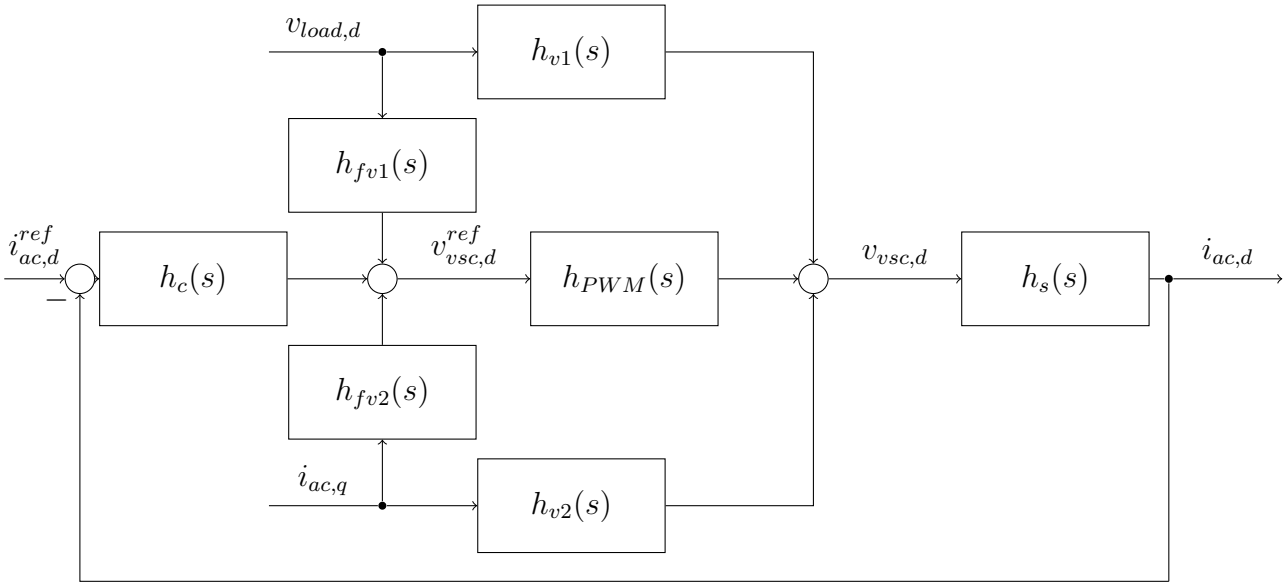

Figure 4.2: The control loop for the inner current control

goal is to choose the feed forward transfer functions, so that the disturbances do not affect the system[21]. This can be achieved by choosing feed forward functions as given in (4.3).

$$h_{fv}(s) = -\frac{h_v(s)}{h_{PWM}(s)} \tag{4.3}$$

For simplicity reasons the ideal static feed forward function is chosen, which is defined by (4.4). This corresponds to the feed forward transfer function eliminating the influence of the disturbance at steady state.

$$h_{fvs} = h_{fv}(0) \tag{4.4}$$

This choice gives the following feed forward transfer functions for the controller.

$$h_{fv1}(s) = -1 \tag{4.5}$$

and

$$h_{fv2}(s) = -j\omega L \tag{4.6}$$

The System Transfer Function

Assuming the feed forward transfer functions to eliminate the effects of the feed forward terms on the system, the system to be controlled is given by

equation (4.7), for both the d and q axis.

$$\mathbf{v}_{vsc} = \mathbf{i}_{ac}R_L + L\frac{d\mathbf{i}_{ac}}{dt} \tag{4.7}$$

From this the transfer function for the system in the laplace domain can be identified as.

$$h_s(s) = \frac{1}{R_L}\frac{1}{1+s\tau} \tag{4.8}$$

where

$$\tau = \frac{L}{R_L} \tag{4.9}$$

The System in Per Unit

It is common practise to represent VSI converter control systems in per unit. To make it easier to compare the controls system in this report with other systems, the per unit system is introduced. For the per unit system the following power invariant base values are used.

$$V_{base} = V_{ph-n} \tag{4.10}$$

$$I_{base} = \frac{2}{3}\frac{S_{base}}{V_{base}} \tag{4.11}$$

Where S_{base} is the three phase power. For the dc-side the following base values are defined, which gives a power invariant per unit system.

$$V_{dc,base} = 2V_{base} \tag{4.12}$$

$$I_{dc,base} = \frac{3}{4}I_{base} \tag{4.13}$$

$$Z_{base} = \frac{V_{base}}{I_{base}}, L_{base} = \frac{Z_{base}}{\omega_{base}} = \frac{V_{base}}{I_{base}\omega_{base}}, C_{base} = \frac{1}{Z_{base}\omega_{base}} = \frac{I_{base}}{V_{base}\omega_{base}} \tag{4.14}$$

The system and cross-coupling decoupling transfer functions, become as following in per unit.

$$h(s) = \frac{1}{R_{L(p.u.)}}\frac{1}{1+s\tau_{(p.u.)}} \tag{4.15}$$

$$h_{fv2} = -j\omega_{(p.u.)}L_{(p.u.)} \tag{4.16}$$

where

$$\tau_{(p.u.)} = \frac{\tau}{\omega_{base}} \tag{4.17}$$

4.2.2 Tuning of the Inner Current Controller

For the controller a PI-regulator is chosen, and with the assumption of the system being perfectly decoupled, the following open loop transfer function is obtained.

$$G_{ol}(s) = K_p \frac{1+sT_i}{sT_i} \frac{1}{1+sT_a} \frac{1}{R_{L(p.u.)}} \frac{1}{1+s\tau_{(p.u.)}} \tag{4.18}$$

The parameters for the controller are chosen according to modulus optimum, as in [26]. First the integral time is chosen such that the biggest time constant in the system is eliminated. As the PWM should be faster than the dynamics of the system, the largest time constant can be identified as $\tau_{(p.u.)}$, and T_i is thus chosen according to (4.19).

$$T_i = \tau_{(p.u.)} \tag{4.19}$$

The controller's proportionality constant is given by equation (4.20)[26].

$$K_p = \frac{\tau_{(p.u.)}}{2R_{L(p.u.)}T_a} \tag{4.20}$$

4.3 The Outer Voltage Controller

The purpose of the outer voltage controller is to control the voltage at the load. Its dynamics in the rotating dq-frame are given by equation (4.21).

$$\begin{aligned} \mathbf{i}_{ac} &= C_f \frac{d\mathbf{v}_{load}}{dt} && \text{System} \\ &+ j\omega L\mathbf{v}_{load} + \mathbf{i}_{load} && \text{Feed forward terms} \end{aligned} \tag{4.21}$$

The structure of the control system is principally similar to that of the inner current controller, a PI-controller, a first order block representing the inner controller and a block representing the system. As can be seen from equation (4.21), there is a cross-coupling and a term which can used as feed forward. The feed forward transfer functions are chosen like in the previous section. The inner current controller is modelled as a first order transfer function with a time constant $T_{eq} = 2 * T_a$[27].

4.3.1 Choice of Capacitor

As the control system will be a cascaded system, some restrictions are put on the choice of the capacitor. It is desirable to select it, such that the

dynamics of the outer loop are much slower than those of the inner. This is accomplished by looking at the output inductance and capacitor of the system as a low pass filter. Then one merely needs to choose a capacitor giving a filter frequency well below the switching frequency. The filter frequency is selected according to (4.22), and the capacitor is chosen according to (4.23).

$$\omega_c < \frac{1}{10}\omega_{sw} \tag{4.22}$$

Where ω_c is the angular frequency of the filter given by (4.23) and ω_{sw} is the angular frequency of the PWM.

$$\omega_c = \frac{1}{\sqrt{LC_f}} \tag{4.23}$$

4.3.2 Tuning of the Outer Current Controller

The control loop for the d-axis in per unit is shown in figure 4.3. From this the open loop transfer circuit can easily be identified as.

$$G_{ol,v} = Kv_p \frac{1 + sTv_i}{s^2 Tv_i T_c} \frac{1}{1 + sT_{eq}} \tag{4.24}$$

where

$$T_c = \frac{C_{f(p.u.)}}{\omega_{base}} \tag{4.25}$$

Here the modulus optimum criterion can not be utilized, since this will result in a system with a double integration, resulting in a system at the stability limit. A possible tuning strategy is to use the symmetrical optimum tuning criterion, as described in[27]. This gives the following expressions for the PI-Controller parameters.

$$Tv_i = a^2 T_{eq} \text{ and } Kv_p = \frac{T_c}{aT_{eq}} \tag{4.26}$$

Where a is a tuning parameter.

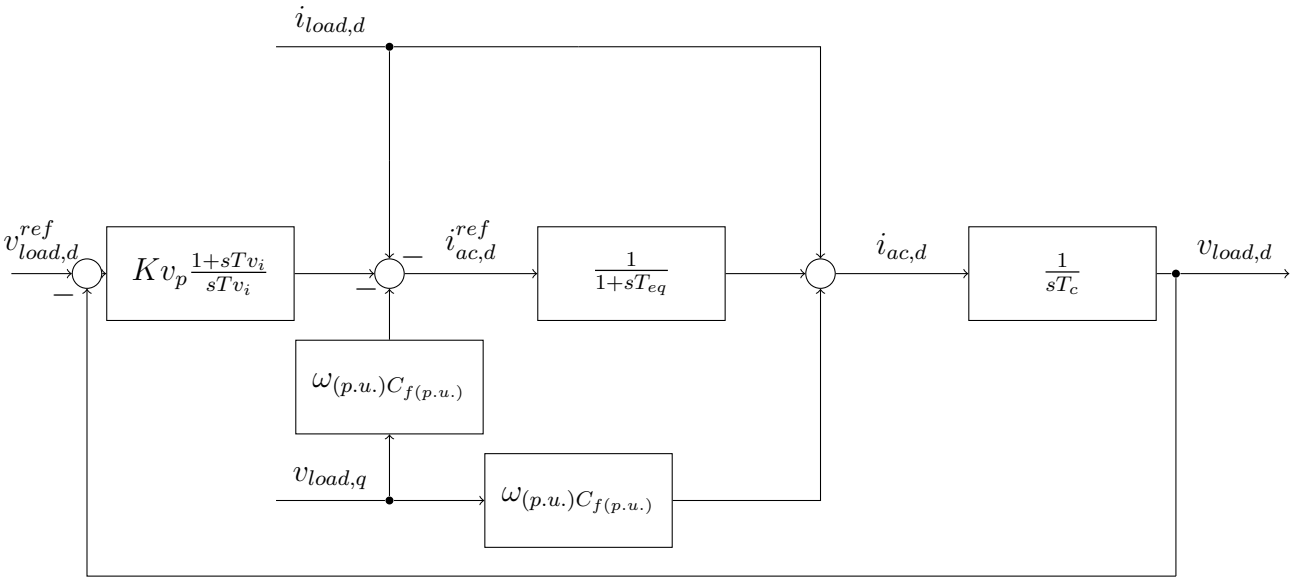

Figure 4.3: The control loop for the outer voltage control

4.4 System Data

Data for the system used in simulink will be presented here. The presentation of the system data is split into two parts. The first part describes elements, which are not modelled by the g-parameters model. The second part on the other hand describes elements, which should be modelled by the g-parameters model.

4.4.1 Common System Data

The common system data is related to the load and the source. The source side consists of a dc voltage source and a resistor, whereas the load side consists of only a load.

Table 4.1: Common System Data

V_{dc}	659.73
R_{dc}	$142.22m\Omega$
$V_{load,rms,ll}$	$400V$
P_{load}(three phase)	$30kW$
Grid frequency	$50Hz$

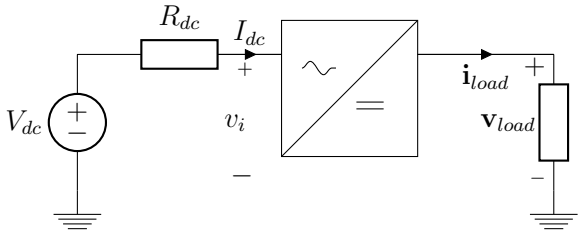

Figure 4.4: System considered

4.4.2 Actual VSI Specific System Data

The actual VSI is described by its control system, the output filter and the modelling of the switches. For the switches the Simulink block universal bridge provided by the library SimPowerSystems was used. For the universal bridge block, ideal switches was chosen, and the rest of the values were left at their respective default values. One more Simulink provided element worth mentioning is the 3-phase PLL. The PLL was set to the grid frequency, and the default gains with automatic gain control was used.

Table 4.2: Output Filter

R_L	$266.67m\Omega$
L	$2.55mH$
C_f	$20.3\mu F$

Table 4.3: Universal Bridge

Switching frequency	$7kHz$
Snubber Resistance	$100k\Omega$
Snubber Capacitance	$infinite$
On Resistance	$1m\Omega$

Table 4.4: Inner Current Controller

K_p	3.34
T_i	$9.55E-3$

Table 4.5: Outer Voltage Controller

K_p	0.38
T_i	$5.71E-4$

Chapter 5

Modelling the Simulink Voltage Source Inverter

The first test of the methodology described in chapter 3 was conducted on the voltage source inverter developed in the previous chapter. The chapter aims at giving an aid in how to implement the general theory and methodology for a specific converter. Along with the goal of verifying the modelling, and to provide insight for further modelling and use of the model.

5.1 The G-Parameters Equations for a VSI

In chapter 2 the general g-parameters equations are given. For a VSI not all the transfer functions are needed, the g-parameters for a VSI are given here for reference.

$$\begin{bmatrix} \tilde{\mathbf{v}}_o \\ \tilde{\mathbf{i}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{G}_o(s) & \mathbf{Z}_o(s) \\ \mathbf{Y}_i(s) & \mathbf{H}_i(s) \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{v}}_i \\ \tilde{\mathbf{i}}_o \end{bmatrix} \tag{5.1}$$

where

$$\tilde{\mathbf{v}}_o = \begin{bmatrix} \tilde{v}_{od} \\ \tilde{v}_{oq} \end{bmatrix}, \tilde{\mathbf{v}}_i = \begin{bmatrix} \tilde{v}_{id} \end{bmatrix}, \tilde{\mathbf{i}}_o = \begin{bmatrix} \tilde{i}_{od} \\ \tilde{i}_{oq} \end{bmatrix}, \tilde{\mathbf{i}}_i = \begin{bmatrix} \tilde{i}_{id} \end{bmatrix} \tag{5.2}$$

and

$$\begin{aligned} \mathbf{G}_o &= \begin{bmatrix} G_{dd} \\ G_{qd} \end{bmatrix}, \mathbf{Z}_o = \begin{bmatrix} Z_{dd} & Z_{dq} \\ Z_{qd} & Z_{qq} \end{bmatrix} \\ \mathbf{Y}_i &= \begin{bmatrix} Y_{dd} \end{bmatrix}, \mathbf{H}_i = \begin{bmatrix} H_{dd} & H_{dq} \end{bmatrix} \end{aligned} \tag{5.3}$$

This corresponds to measurement set I, III and IV from chapter 2, which gives the following decoupling matrix.

$$\mathbf{D} = \begin{bmatrix} 1 & TR_{ddm} & TR_{dqm} \\ TG_{ddm} & 1 & TI_{dqm} \\ TG_{qdm} & TI_{qdm} & 1 \end{bmatrix} \tag{5.4}$$

5.2 Data Collection

The circuit used for obtaining the data is shown in figure 5.1. As can be seen from the figure there are two ac-sources. Their task is to inject perturbations into the system, thus creating small deviations from the steady state operating point, from which the small signal model can be created. The best results were obtained with a voltage disturbance at the dc-side. The resistance R_{dc} was added on the dc-side to make current injection possible. If one looks at measurement set I, from chapter 2, one sees that everything is divided by $\tilde{v}_{id}$. This small signal value will always be zero, if one does current injection in parallel with an ideal source. On the ac-side the voltage is the control variable, and thus it made more sense to inject a current to the ac-side. The amplitude of the injected disturbances were chosen to be $0.05_{(p.u.)}$ in all cases. With smaller disturbances it was difficult to distinguish the injected disturbances from the system's noise.

To get the frequency response, one needs to inject disturbances at varying frequencies. In this case the frequencies ranged from 1 Hz to the half of the VSI's switching frequency. To get a reasonable frequency spread, the distance between the frequencies were increased logarithmically. And a total number of 47 injections were injected, for each measurement set.

The measurements were converted to the frequency domain by the fast fourier transform provided by Matlab. To reduce the noise in the frequency response, the perturbation injections and data recordings were done after the system had become stable. Before starting to iterate and inject perturbations at different frequencies, the frequency response of a base case with no perturbations was recorded. By doing this the small signal values could be obtained by subtracting the base case measurement from the measurements with a perturbation, as seen in (5.5)

$$\tilde{x}(f_p) = x_{base}(f_p) - x_m(f_p) \tag{5.5}$$

Where $\tilde{x}$ is the collected small signal value and x_{base} was the base case, x_m was the measured value and f_p was the frequency, at which the perturbation was injected. For every frequency first a perturbation was injected into the

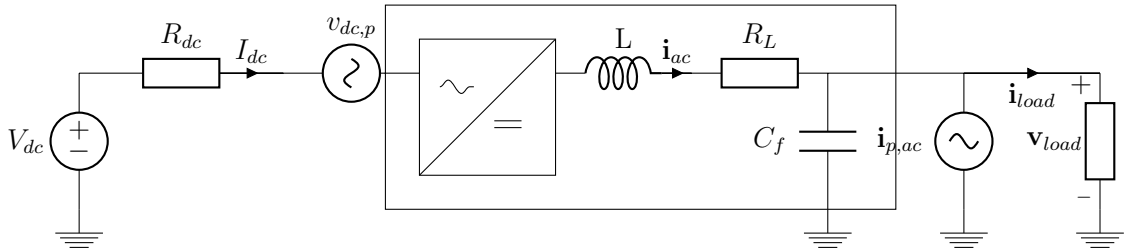

Figure 5.1: System for obtaining the data

dc side and data was collected, before the same wass done to the ac side. The inductance and capacitor at the output of the VSI converter were also assumed to be a part of the model. This is highlighted in figure 5.1 by a rectangle enclosing the part of the circuit to be modelled.

5.3 Discussion on the Perturbation Injection

For the injection scheme to work, one needs to be able to inject a disturbance at a certain frequency, and later extract the response at the same frequency. It is also preferable that the magnitude of the disturbance is greater than that of the noise, which already is in the system. As one can see from figures 5.2

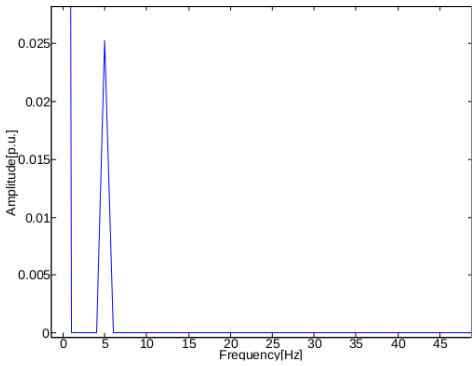

Figure 5.2: $\tilde{v}_{id}$ due to 5Hz perturbation at the dc-side injected in series

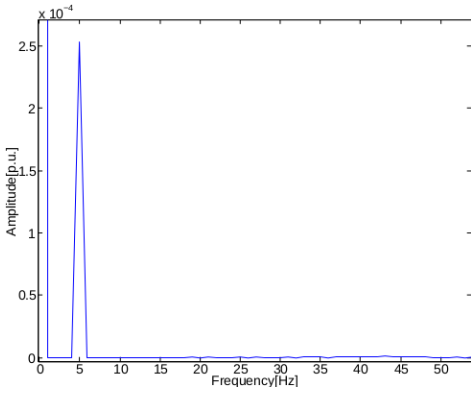

Figure 5.3: $\tilde{v}_{id}$ due to 5Hz perturbation at the dc-side injected in parallel

and 5.3, the perturbation can easily be seen as a peak in the dc-voltage. Also worth noting is that the peak in the current injection case is one hundred times smaller than that of the voltage injection case. This indicates that

the resistor at the dc-side is very small compared to the resistance in the path to the right of the injection source. In other words virtually all of the perturbation current goes to the left, and the resulting voltage peak is equal to the perturbation current times the resistance. In this case where the perturbation current is 0.05(p.u.), the resistance is 0.01(p.u.) and assuming the whole current to take the path through the resistor, it results in a fourier response of $0.05 * 0.01/2 = 2.5 \cdot 10^{-4}$. One can thus conclude that if one wants to get the same results by using current injection at the dc-side, as with voltage injection, one needs to inject a current which is 100 times as high as the voltage in per units.

As the d-axis is the axis where the real power is transferred, the dc-side and the d-axis should be highly coupled. In figures 5.4 and 5.5 this is illustrated. One can also, see that the perturbation going through the converter is damped, which is most likely due to the filter at the output of the converter. Investigating the ac-side's response to perturbations injected at the ac-side

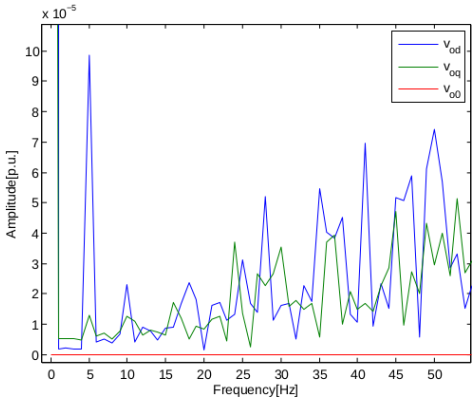

Figure 5.4: $\tilde{\mathbf{v}}_o$ due to a 5Hz perturbation at the dc-side injected in series

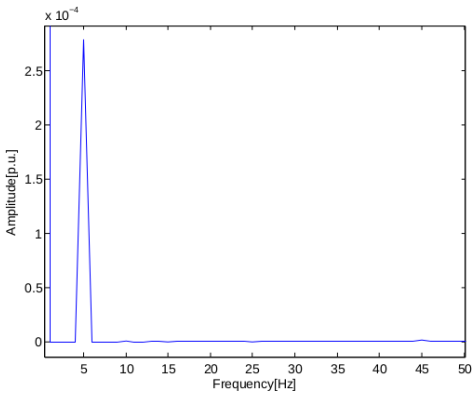

Figure 5.5: $\tilde{v}_{id}$ due to a 5Hz perturbation in the d-phase injected in parallel

also reveals interesting information. For instance in the control system there is an ideal feed forward transfer function responsible for decoupling the d and q axis. Based upon the reasoning from the construction of the control system, the decoupling should work best for low frequencies. This result can easily be seen in figures 5.6 and 5.7.

As explained in chapter 3, two means of injecting perturbations are used in this report. One where the injections are injected exclusively along first the d- and then the q-axis, which were used to obtain the model of the Simulink VSI. And one where a positive sequence perturbation is injected, followed by

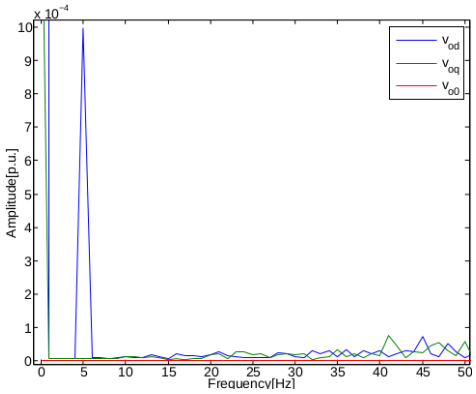

Figure 5.6: $\tilde{\mathbf{v}}_o$ due to a 5Hz perturbation in the d-phase injected in parallel

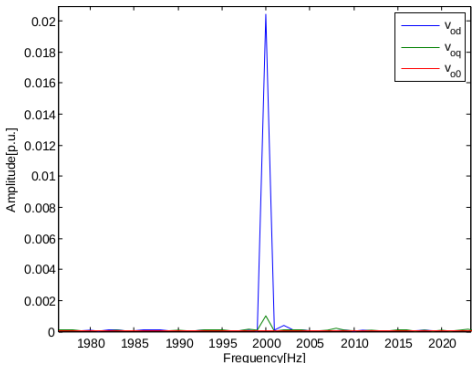

Figure 5.7: $\tilde{\mathbf{v}}_o$ due to a 2000Hz perturbation in the d-phase injected in parallel

a negative sequence perturbation. The technique used to obtain the model of the Simulink VSI, might be straightforward to implement in Simulink, however the same can not be said for practical purposes. To compare the results of the two techniques the Simulink VSI was used. The comparison is primarily done by comparing the terminated transfer functions.

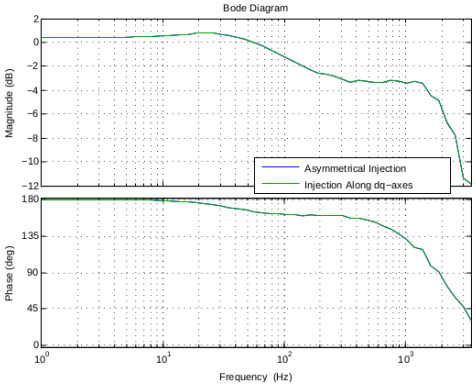

Figure 5.8: Terminated Y_{dd} obtained through different injection schemes

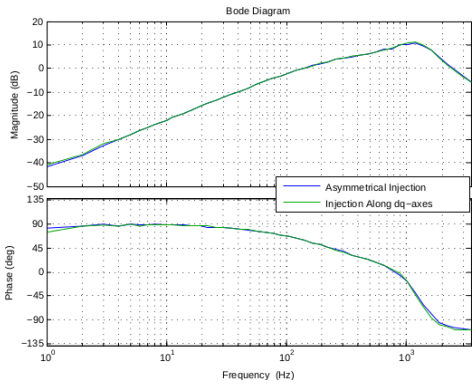

Figure 5.9: Terminated Z_{qq} obtained through different injection schemes

As figures 5.8 and 5.9 show, the two techniques yield virtually the same results for transfer functions along one axis, and on the same side of the converter. This is also confirmed in figures 5.10 and 5.11, except for a small

difference in H_{ddm} at higher frequencies. The picture changes when inspecting transfer functions between two axes, as can be seen in figures 5.12 and 5.13. In this case the transfer functions obtained, by line to line injection are of a significantly higher amplitude. Consequently the curves are also smoother, as they are not as influenced by the noise present in the system. The phases are also higher. The reason for this significant difference, is due to the fact that the line to line injection technique injects a current in both the d and q axis. Inspecting figures 5.14 and 5.15, one can see that the amplitudes in the d and q axis are the same. Which gives amplitudes of Z_{dq} and Z_{qd} equal to that of Z_{dd} and Z_{qq}. Yet it remains to confirm that the methods yield the same results, after applying the decoupling. This is confirmed in figures 5.17 and 5.16, where one can see that the curves are following each other in amplitude and phase, although the noise is somewhat different.

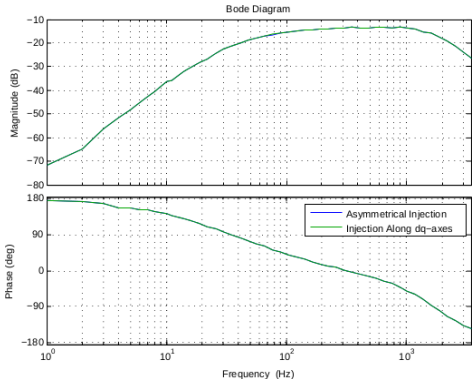

Figure 5.10: Terminated G_{dd} obtained through different injection schemes

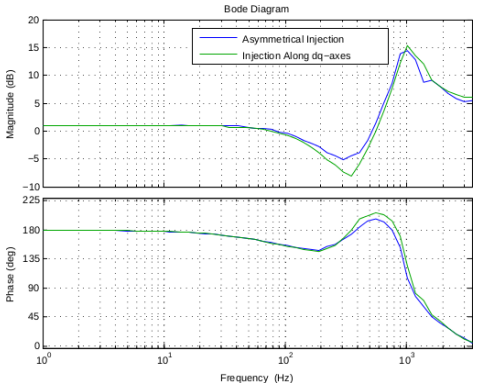

Figure 5.11: Terminated H_{dd} obtained through different injection schemes

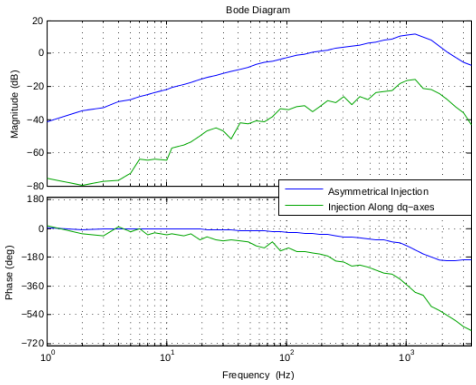

Figure 5.12: Terminated Z_{dq} obtained through different injection schemes

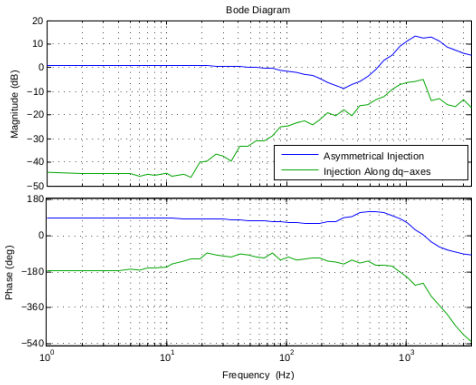

Figure 5.13: Terminated H_{dq} obtained through different injection schemes

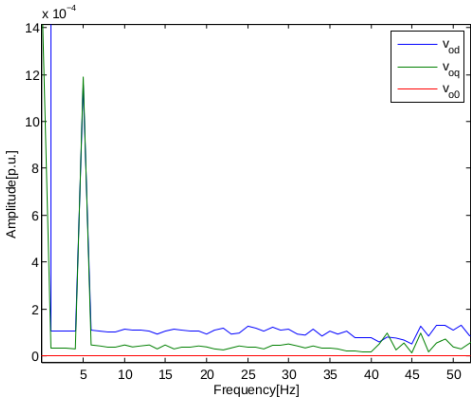

Figure 5.14: $\tilde{\mathbf{v}}_o$ due to a 5Hz positive sequence perturbation at the ac-side

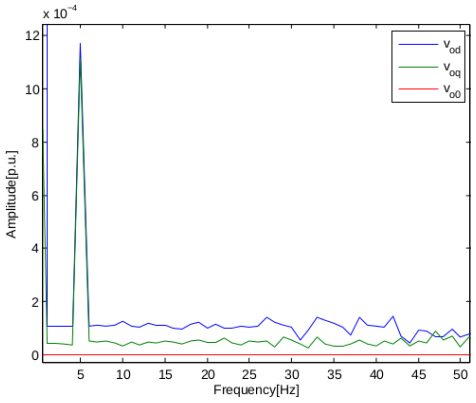

Figure 5.15: $\tilde{\mathbf{v}}_o$ due to a 5Hz negative sequence perturbation at the ac-side

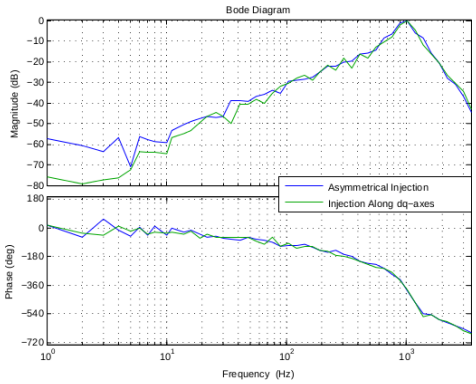

Figure 5.16: Unterminated Z_{dq} obtained through different injection schemes

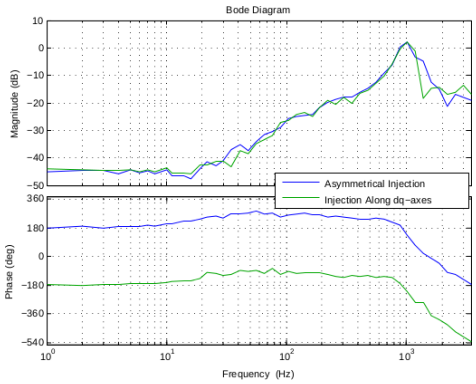

Figure 5.17: Unterminated H_{dq} obtained through different injection schemes

5.4 Comparison Between the Terminated and Unterminated Transfer Functions

After using Matlab and Simulink to obtain the terminated transfer functions, the unterminated transfer functions have to be calculated. It is of interest to investigate what the decoupling does for various reasons. For instance if one could do with only the terminated transfer functions, one measurement set at the ac-side would suffice. The terminated transfer functions are also interesting to inspect, as they give valuable information on how the perturbation propagates through the converter. The terminated transfer functions presented were obtained, by perturbating the d and q axis independently.

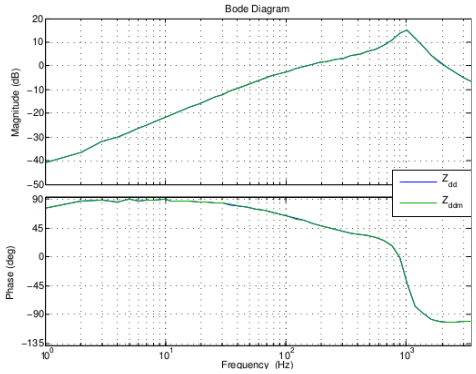

Figure 5.18: The terminated and unterminated Z_{dd}

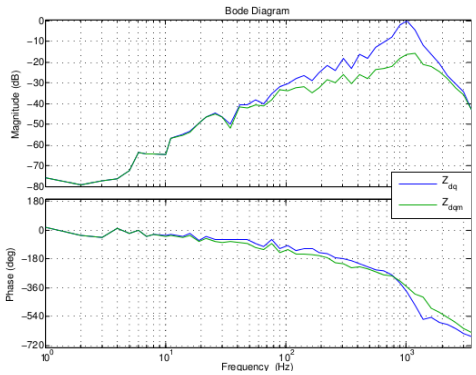

Figure 5.19: The terminated and unterminated Z_{dq}

As figures 5.18 and 5.19 show, the difference between the terminated and unterminated transfer functions are small. For Z_{dq} there is, however, some difference for higher frequencies. It is also worth noting that Z_{dq} is noisier than Z_{dd}, this is a quite general trend. On one hand the transfer functions purely along one axis are smooth, whereas on the other hand the transfer functions between two axis are noisy. This trend corresponds well with the findings from last section. Where it was shown indications on a low coupling between the q and d axis.

One more interesting thing to note is the two cases, for the transfer functions along one axis, where the terminated and unterminated transfer functions differ. Namely G_{dd} and Y_{dd}, they are both from the same set of measurements. This is the set where the deviation of the input voltage is the common denominator of all the transfer functions, which might indicate that the input source's dynamics are influencing the measurements. Another

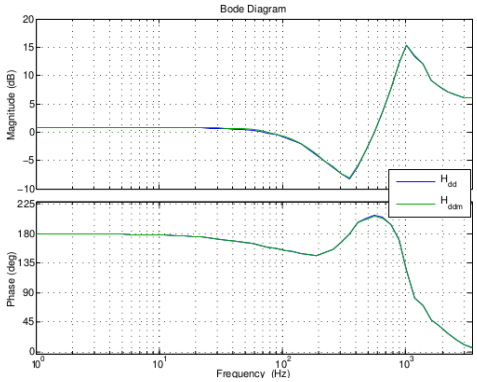

Figure 5.20: The terminated and unterminated H_{dd}

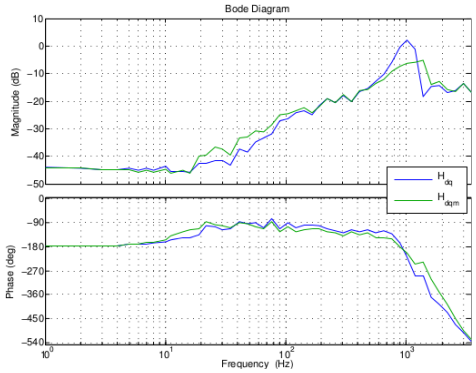

Figure 5.21: The terminated and unterminated H_{dq}

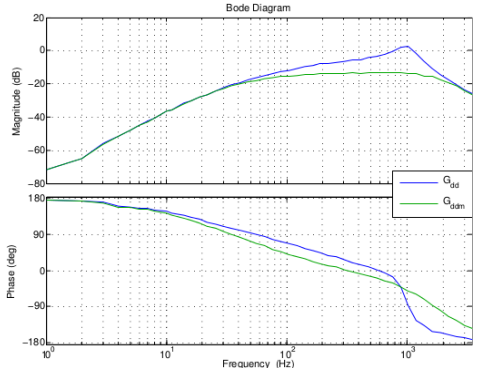

Figure 5.22: The terminated and unterminated G_{dd}

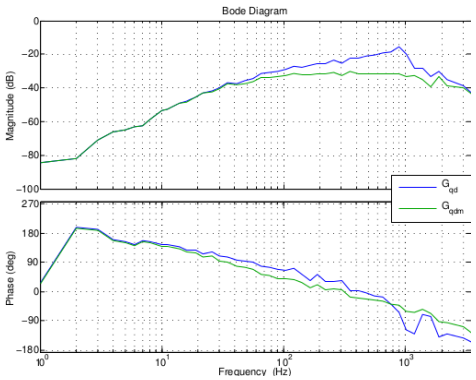

Figure 5.23: The terminated and unterminated G_{qd}

trend is that, where the terminated and unterminated transfers differs, the amplitude of the terminated transfer function is always higher. This might indicate that the source and load are damping the deviations.

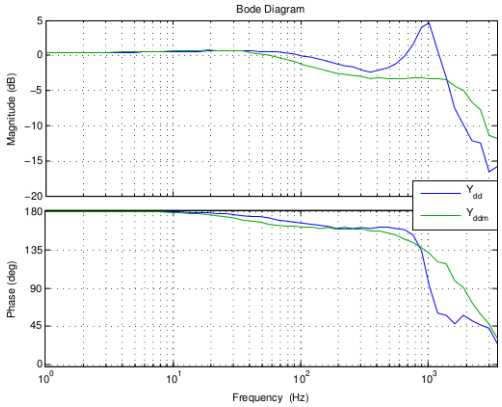

Figure 5.24: The terminated and unterminated Y_{dd}

5.5 Fitting of the Transfer functions

The next step in obtaining the model is to fit transfer functions to the measured transfer functions. Applying the methodology described in chapter 3, it was found that all the transfer functions could be approximated by fourth order functions. For the smoother curves like 5.25 the fit is almost perfect. It

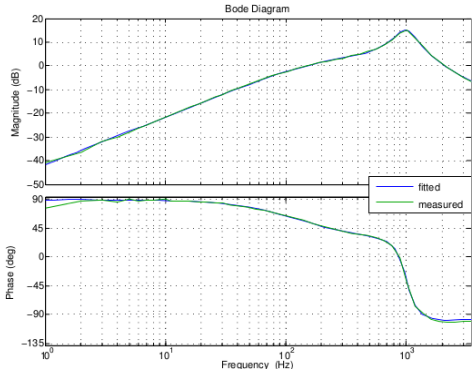

Figure 5.25: The measured and fitted Z_{dd}

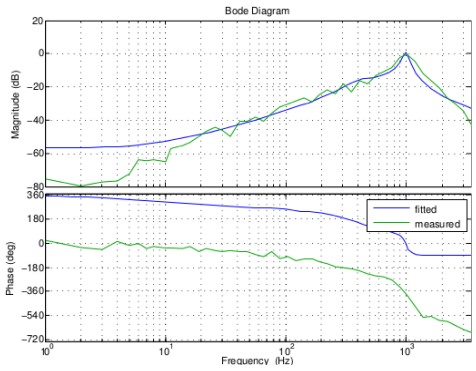

Figure 5.26: The measured and fitted Z_{dq}

should come as no surpise, that the same is not the case for the noisier curves like 5.26. The impact of this misfit, should however, not be too big. As the transfer functions where the misfit is the biggest, are the transfer functions between the axes. And the system is designed, such that there should be as little coupling as possible between the axes.

5.6 Verifying the Model

This far it has been shown that it is possible to measure the g-parameters of a VSI. And then fit transfer functions to the measured functions. However, there are more questions to be answered. For instance, how well does the measured transfer functions describe the model? In figures 5.27 and

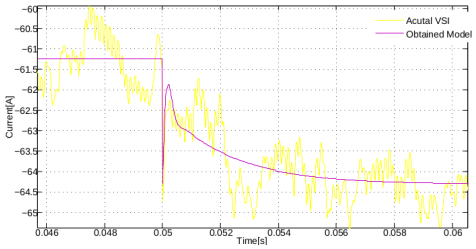

Figure 5.27: The output current response to a 0.05(p.u.) load increase

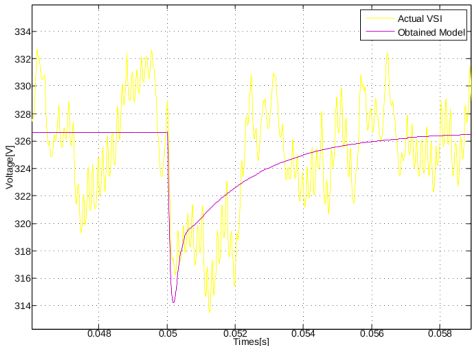

Figure 5.28: The output voltage response to a 0.05(p.u.) load increase

5.28, the output's voltage and current response to a 0.05(p.u.) load step increase is presented, for both the actual VSI implemented in Simulink, and the g-parameters model implemented in Simulink. The curves are both quantitatively and qualitatively similar. The biggest difference being that the g-parameters model doesn't contain any noise.

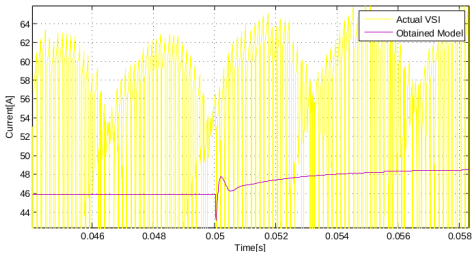

Figure 5.29: The input current response to a 0.05(p.u.) load increase

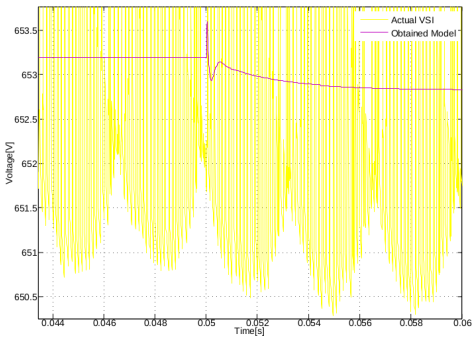

Figure 5.30: The output voltage response to a 0.05(p.u.) load increase

From figure 5.29 and 5.30 it is evident that the curves are quantitatively similar. The reason the curves can not be the same qualitatively is that the g-parameters model does not contain any switches. It is thus incapable of displaying the same discontinuous behaviour as the real circuit.

Looking at the transient curve trajectories of both the voltages and currents, one can see that part of the modelling's appeal lays in the prediction of steady state conditions after a transient, and the prediction of peaks occurring during the transients. To check the model's accuracy at predicting steady state values, the output and input powers are plotted against an increasing load step. The results are presented in figure 5.31. On one hand it

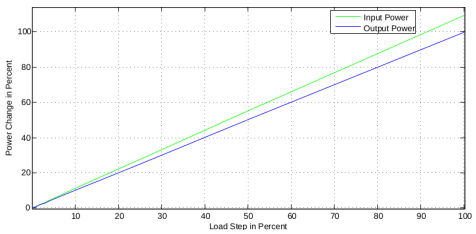

Figure 5.31: The model's input and output powers plotted against load steps

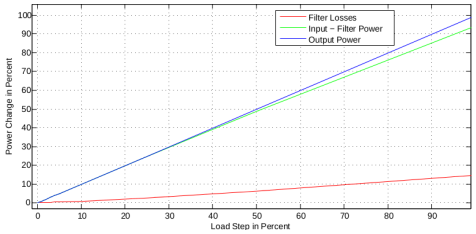

Figure 5.32: The model's input, filter and output powers plotted against load steps

can be seen that the power change at the output is almost identical to the load step. On the other hand the input power curve is to steep, giving quite big errors for larger load steps. The reason for the deviation in the input power, is partly due to the resistance in the output filter. More precisely, if the output power is $30kW$ the input power has to be $P_{in} = P_{out} + \frac{3}{2}R_L \cdot i_{od}^2$, assuming the on resistance of the switches to be negligible. In the simulations done in this part it, was assumed that the the input circuit only had to deliver power to the output, not to the filter. The input circuit was thus set to operate at $P_{in} = P_{out}$. However, when a load step is applied, the input power has to increase more than the output power, because of the filter resistance. As can be seen in figure 5.32, subtracting the estimated power dissipation in the filter from the input power, yield almost the same curve as the output power. There is still some deviation for higher load steps, indicating that the modelling estimates a too low value for the filter resistance.

To give an indication on the models ability to correctly predict the converters transient behaviour, the peak of the input and output voltages during transients are plotted in figures 5.33 and 5.34. Both the predicted peak voltages and currents are quite accurate for load step up till 10%, just like the

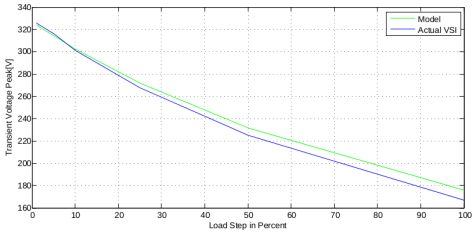

Figure 5.33: The output voltage's transient peaks

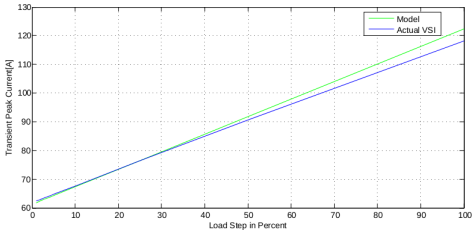

Figure 5.34: The output current's transient peaks

powers.

As the models behaviour closely matches that of the real converter in dq-coordinates, one can assume it to behave similar in abc-coordinates as well. The catch is that some more components need to be added to the simulink model, to make it produce the results in abc-coordinates. It is thus useful to also investigate the g-parameters model's behaviour in abc-coordinates. As can be seen in figures 5.35 and 5.36, the general curve shape is the same. The biggest difference, being that the model does not contain any ripple. Which is to be expected, as it was measured up to half of the switching frequency, and the ripples are mostly due to the switching.

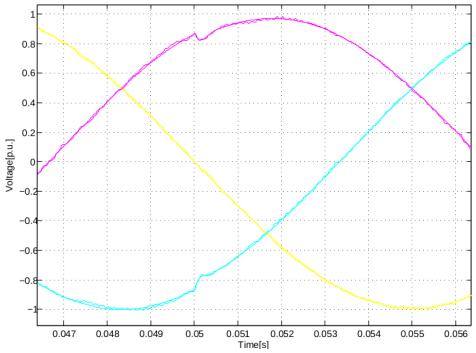

Figure 5.35: The g-parameters model's and the actual VSI's output voltage's response to a $10_{p.u.}$ load step

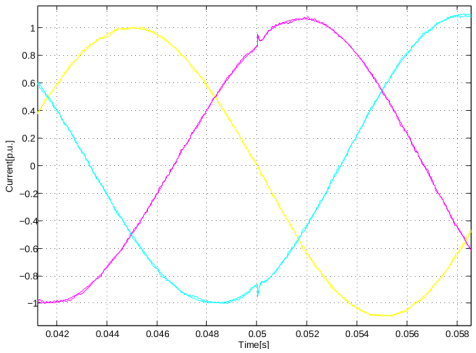

Figure 5.36: The g-parameters model's and the actual VSI's output current's response to a $10_{p.u.}$ load step

5.7 Testing the Model's Stability Criterion

From the general stability criterion stated in chapter 2, the following criterion can be found for the VSI model.

$$\tilde{v}_{id} = (\tilde{v}_{sd} - \tilde{i}_{od} H_{dd} Z_{id} - \tilde{i}_{oq} H_{dq} Z_{id}) \cdot \frac{1}{1 + Y_{dd} Z_{id}} \tag{5.6}$$

$$\tilde{i}_{od} = -\frac{\tilde{v}_{id} G_{dd} + \tilde{i}_{oq} Z_{dq}}{Z_{Ld}} \cdot \frac{1}{1 + Z_{dd}/Z_{Ld}} \tag{5.7}$$

$$\tilde{i}_{oq} = -\frac{\tilde{v}_{id} G_{qd} + \tilde{i}_{od} Z_{qd}}{Z_{Lq}} \cdot \frac{1}{1 + Z_{qq}/Z_{Lq}} \tag{5.8}$$

5.7.1 The Input Voltage Stability

In figure 5.37 the Nyquist plot of $Y_{dd} \cdot R_{dc}$ is given. From the figure one can see that, if the plot stretched 100 times further in to the left half plane, the system would be unstable, which is confirmed in figure 5.38. Giving the conclusion that a resistance of $100R_{dc}$ will make the system unstable. The

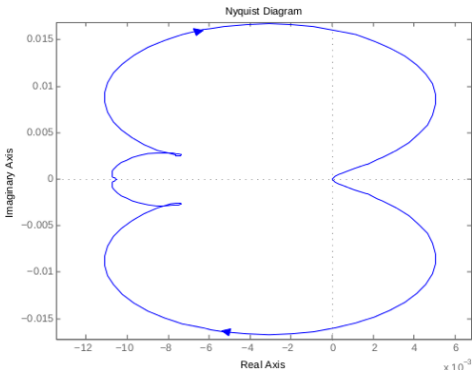

Figure 5.37: Nyquist plot of $Y_{dd} \cdot R_{dc}$

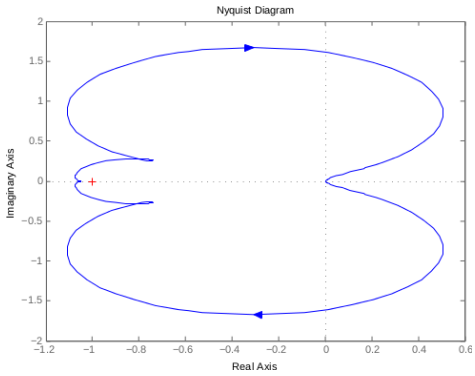

Figure 5.38: Nyquist plot of $Y_{dd} \cdot 100R_{dc}$

findings of figure 5.38 are confirmed by figures 5.39 and 5.40.

It is however, still questionable if the stability limit of the model equals that of the real VSI. For instance if one inspects figure 5.41, one will see that the model is supposed to be stable, for such an input resistance. Despite of the g-parameters model being stable one can see, in figure 5.42, that the input voltage settles at an unrealistic value. This negative voltage means that power is delivered to the source, from sources internal of the converter,

which can not happen in the real VSI. In other words the model is stable, as predicted by the criterion, although it is showing an unrealistic behaviour from a physical point of view. Further confirming these findings are the output voltages of both the model and the converter presented in figure 5.43, where one can see the model's voltage stabilizing, whereas the converter becomes unstable. From these considerations, one can conclude that this stability criterion should be used to assess the model's stability, not the converter's. The stability of the converter can still be assessed through the model, but it has to be done through simulations.

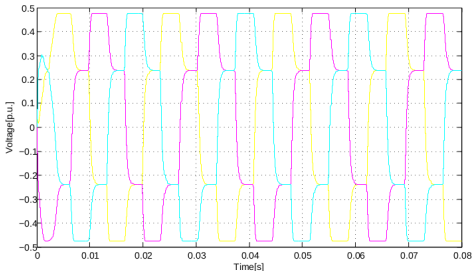

Figure 5.39: The VSI's output voltage, with $R_{dc} \cdot 100$

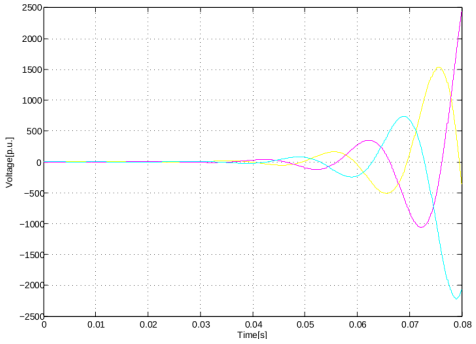

Figure 5.40: The model's output voltage, with $R_{dc} \cdot 100$

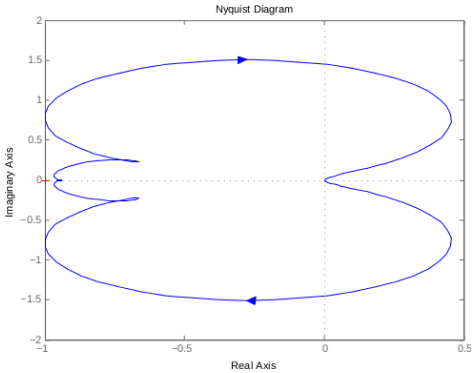

Figure 5.41: Nyquist plot of $Y_{dd} \cdot 90R_{dc}$

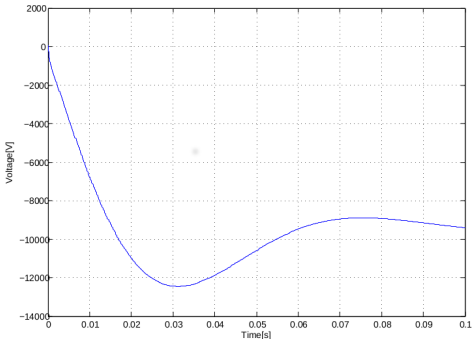

Figure 5.42: The model's input voltage, with $\cdot R_{dc} \cdot 90$

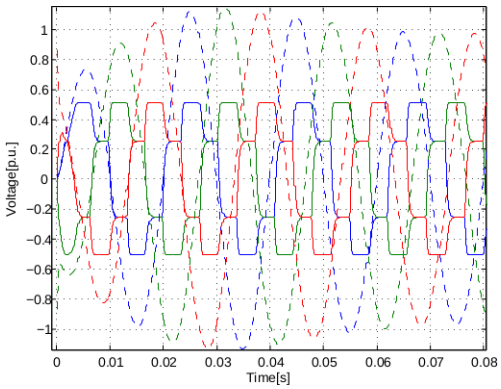

Figure 5.43: The converter and model's output voltage, with $90 \cdot R_{dc}$

5.7.2 The Output current Stability

For the output voltage stability, the criterion fails to predict the model's stability limit. As can be seen in figure 5.44 the model's voltage is unstable for a capacitive load of 5(p.u.), this is a high value, however according to figure 5.45 the model should be stable. As Z_{dd} equals Z_{qq} and the load connected is symmetrical, both of the output voltage stability criteria should yield the same results. The reason, why they fail to predict the stability might be due to some couplings, which are not considered, when the stability of the model is split up into multiple criteria.

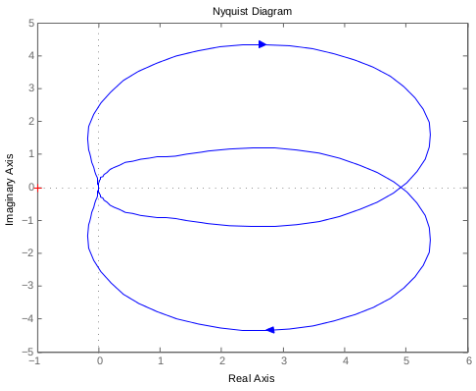

Figure 5.44: Nyquist plot of the output voltage stability criterion, for a 5(p.u.) capacitive load in series with a 1(p.u.) resistive load

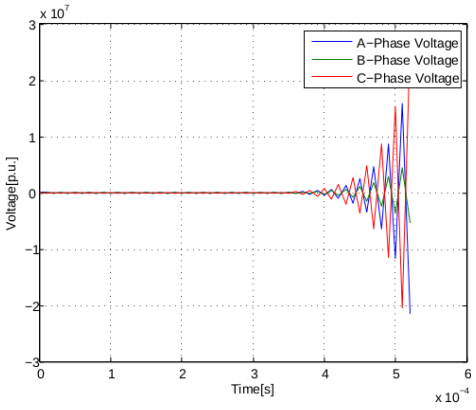

Figure 5.45: The model's output voltage, with a 5(p.u.) capacitive load in series with a 1(p.u.) resistive load

Chapter 6

Development of Test Procedure to Obtain G-Parameters for DC-DC Converters

One of the advantages of the g-parameters modelling, is the black-box modelling approach. In the previous chapter the methodology was applied to a Simulink model. Most converters of interest, however, are real physical models, in which the control system is unknown. In this chapter a measurement procedure for dc-dc converters are presented.

6.1 The G-parameters Equations for DC-DC Converters

To make it easier to follow this chapter, the relevant g-parameters equations for dc-dc converters are presented here.

$$\begin{bmatrix} \tilde{\mathbf{v}}_o \\ \tilde{\mathbf{i}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{G}_o(s) & \mathbf{Z}_o(s) \\ \mathbf{Y}_i(s) & \mathbf{H}_i(s) \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{v}}_i \\ \tilde{\mathbf{i}}_o \end{bmatrix} \tag{6.1}$$

where

$$\tilde{\mathbf{v}}_o = \left[\tilde{v}_{od}\right], \tilde{\mathbf{v}}_i = \left[\tilde{v}_{id}\right], \tilde{\mathbf{i}}_o = \left[\tilde{i}_{od}\right], \tilde{\mathbf{i}}_i = \left[\tilde{i}_{id}\right] \tag{6.2}$$

and

$$\begin{aligned} \mathbf{G}_o &= \left[G_{dd}\right], \mathbf{Z}_o = \left[Z_{dd}\right] \\ \mathbf{Y}_i &= \left[Y_{dd}\right], \mathbf{H}_i = \left[H_{dd}\right] \end{aligned} \tag{6.3}$$

This corresponds to measurement set I and III from chapter 2, which gives the following decoupling matrix.

$$\mathbf{D} = \begin{bmatrix} 1 & TR_{ddm} \\ TG_{ddm} & 1 \end{bmatrix} \tag{6.4}$$

6.2 The System Configuration

The chosen converter to model was a boost converter made for educational purposes[28][29]. It has a modular design, and can act as different types of dc-dc converters through the change of modules. The system was powered by an Instek GPC-3030DQ voltage source[30]. The load was a variable resistor with a maximum resistance of 50Ω. All the systems's component and values, describing the system's state are listed in table 6.1. The capacitor is only used for current injections, to prevent the full dc voltage to appear across the isolation transformer.

Table 6.1: System Values

Input Voltage	13.52V
Output Voltage	28.4V
Boost Converter Frequency	30kHz
Capacitor	$3300\mu F$
Load Resistance	42.2Ω

6.3 The Perturbation Injection and Measurement System

At the heart of the perturbation injection and measurement technique is the network analyser E5061B from Agilent Technologies[31]. It is used both for generating the perturbation signals and for calculating the transfer functions of interest. The signal generated by the network analyser is amplified by a power amplifier [32]. The perturbation system is separated, from the rest of the system through an isolation transformer. When doing current injection, a capacitor is used to prevent short circuiting the circuit. The input signals to the network analyser, from which it calculates the transfer functions are obtained from two types of probes. Current probes of the type Fluke 80i-110s[33] and voltage probes of the type Tektronix P5200 and P5200A[34]. The settings of the network analyser is given in table 6.2 and 6.3. The

Table 6.2: LF output

Power	$10dB$
Frequencies	$10Hz - 15kHz$
IFBW	$<= 10Hz$
Number of samples	1600

Table 6.3: R and T channel

Attenuation	$20dB$
Protection	$1M\Omega$

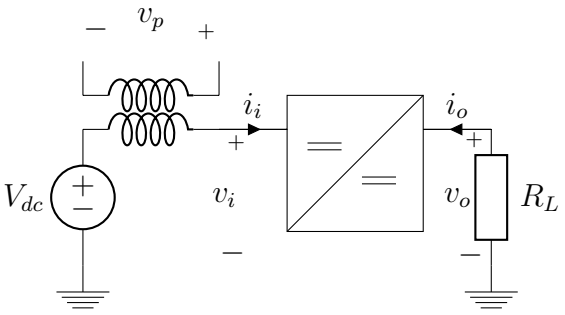

Figure 6.1: Voltage injection at the input side

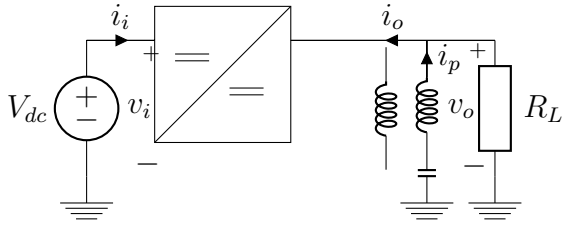

Figure 6.2: current injection at output side

amplitude of the perturbation out of the amplifier was around two volts, however it varied with the frequency.

In the previous chapter where the measurement was done in Simulink, all quantities corresponding to measurement set I, III and IV, could easily be measured. The network analyser, however, has only two inputs, making it impossible to measure all four voltage and current quantities simultaneously. It was thus decided to take advantage of the network analyser's transfer function measuring feature. That is the network analyser can measure the phase between its two inputs, and the the power of the division of its two inputs in frequency domain, which yields the measured transfer functions. As can be seen from figures 6.1 and 6.2, there are two ways of injecting perturbations, namely voltage injection in series and current injection in parallel. Although the figures show voltage injection at the input and current injection at the output, this is not a requirement. However as can be seen from measurement set I and III, half of the transfer functions will have $\tilde{v}_{id}$ in the denominator, and the rest $\tilde{i}_{od}$ in the denominator, it thus makes sense to ensure good values for these.

6.4 Comparison between Injecting in Parallel and Series

In real applications there might be restrictions on how one can connect to a circuit. In other words, one might not be able to choose freely between doing the injections in series or parallel. To investigate, if the same results can be obtained independently of the injection strategy; A comparison between the terminated output and input impedances obtained using the two strategies are presented. Where the input impedance is obtained by perturbating the input, and the output impedance by perturbating the output.

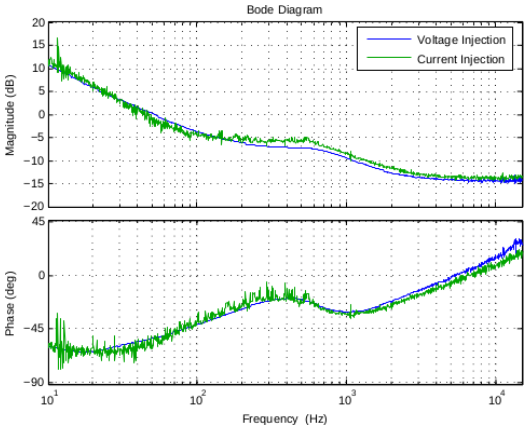

Figure 6.3: Measured input impedance

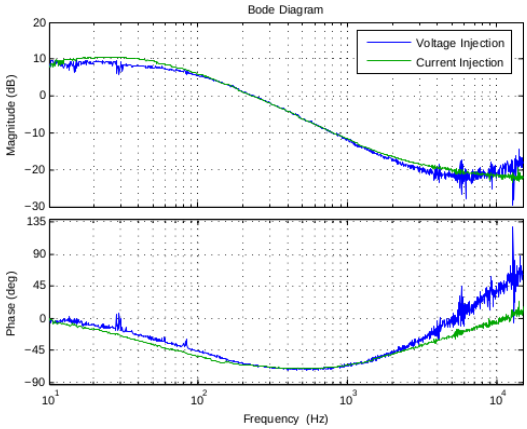

Figure 6.4: Measured output impedance

Investigating figures 6.3 and 6.4 one can observe, that doing voltage or current injections, in these cases essentially yield the same results. The reasoning is that most of the noise is of a very high frequency, and of the same amplitude. The fitting function fitfrd, will thus make approximately the same curves in both cases. However, care should be taken, if one tries to fit transfer functions using high orders. Noise in the measurements might be recognized as a valid state by the fitting function. To reduce this issue, one can smooth the transfer functions. This can be done either by using software, or the post-processing capabilities of some measurement equipment. The effect of a moving average filter applied by the network analyser used in this report, is presented in figures 6.5 and 6.6. The moving average window was set to one percent of the number of samples. To illustrate the problem, where noise is wrongly identified as valid states. An example where the input impedance is assumed to be of order 50 is presented. Although an unrealistic

and unnecessary assumption in this simple case, it is quite illustrative. In figure 6.7, one can clearly see the peaks recognized as states, in the non smoothed case.

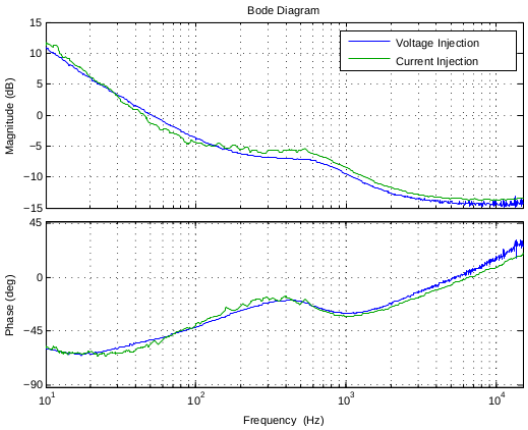

Figure 6.5: Measured input impedance, with smoothing applied to the current injected

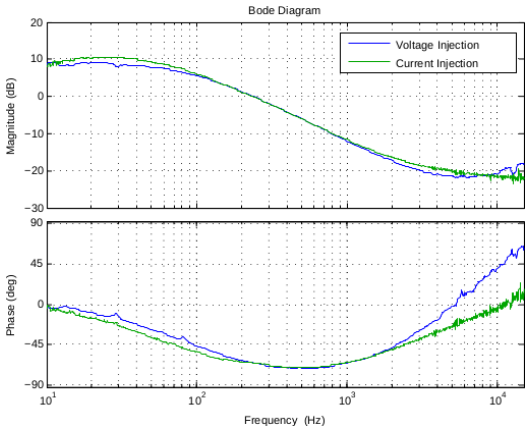

Figure 6.6: Measured output impedance, with smoothing applied to the voltage injected

Having observed virtually the same results regardless of the injection technique, some insight in the noise difference might still be useful. Observing the noisiest technique to be side dependent reveals valuable information for later design choices. When injecting at the input, current injection is the noisiest alternative, whereas it is voltage at the output. The reason for this lays most likely in the nature of the boost converter. During the testing the output of the amplifier was always 2 volts, however the magnitude of the currents and voltages, varies significantly from side to side. On the input side the current is higher than that on the output side, and for the voltage the situation is the opposite. A good perturbation is of an higher amplitude, than that of the noise already present in the system. Meaning that two volts is a significantly larger disturbance compared to the input, than to the output.

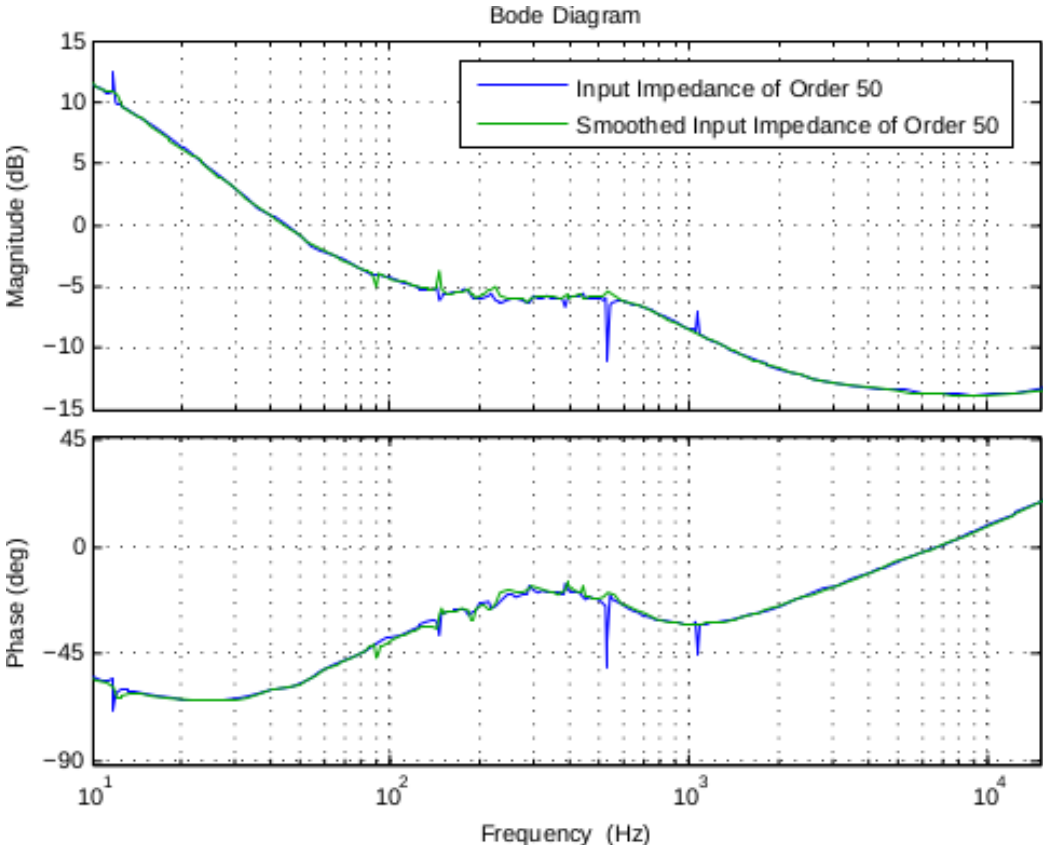

Figure 6.7: Fitted input impedances with and without smoothing

6.5 Comparison between the Terminated and Unterminated Transfer Functions

As with the VSI in Simulink it is of interest to compare the terminated and unterminated transfer functions. Figures 6.8-6.11 show many of the same tendencies as the transfers of the VSI. The transfer functions, which are between values on the same side of the converter, are smoother than those between values on both sides of the converter. It is also interesting to note, which transfer function, where the terminated deviates the most from the unterminated. Just as with the VSI, the deviation is the biggest for Y_{dd}. This might be due to the close connection to the source powering the system, which is supposed to act as a stable power supply, thus actively reducing deviations.

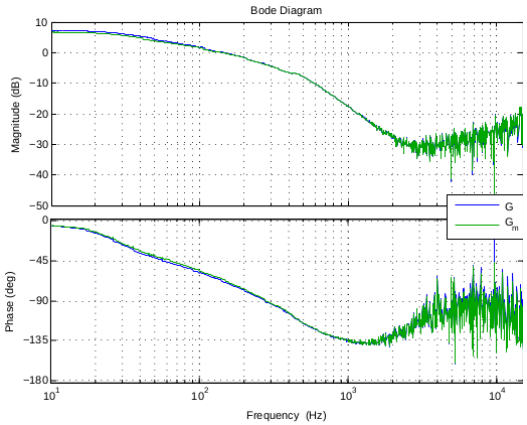

Figure 6.8: The terminated and unterminated G_{dd}

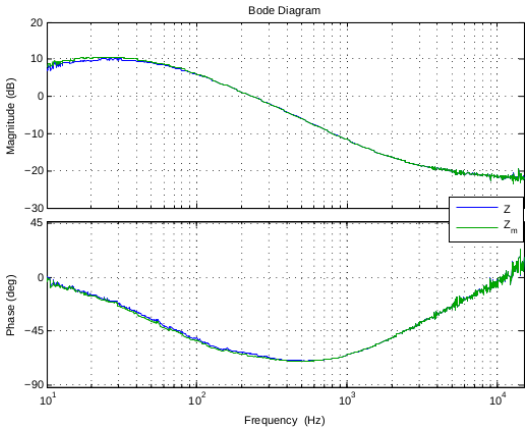

Figure 6.9: The terminated and unterminated Z_{dd}

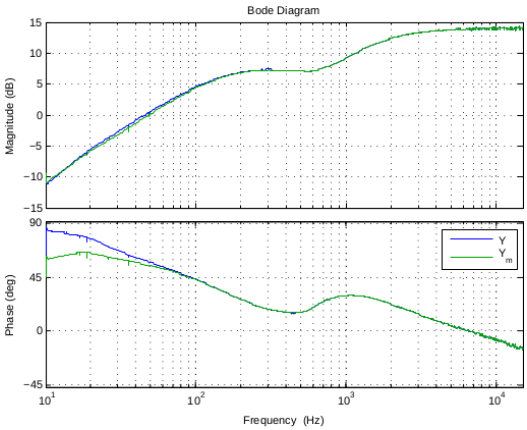

Figure 6.10: The terminated and unterminated Y_{dd}

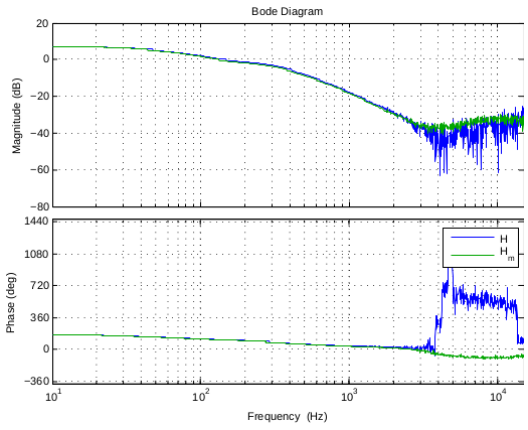

Figure 6.11: The terminated and unterminated H_{dd}

6.6 Fitting of the Transfer Functions

By using the methodology described in chapter 3, the reduced order of all the transfer functions where found to be one. Except for the order of the input admittance, which was found to be a third order transfer function. All of the transfer functions had better fits at higher orders, however the hankel values revealed an order of one to be sufficient. This is illustrated for the output impedance in figures 6.12 and 6.13. Inspecting figures 6.12 and 6.16

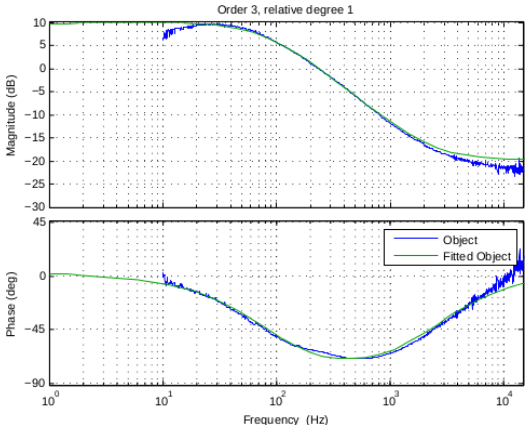

Figure 6.12: The measured and fitted Z_{dd}, as a third order transfer function

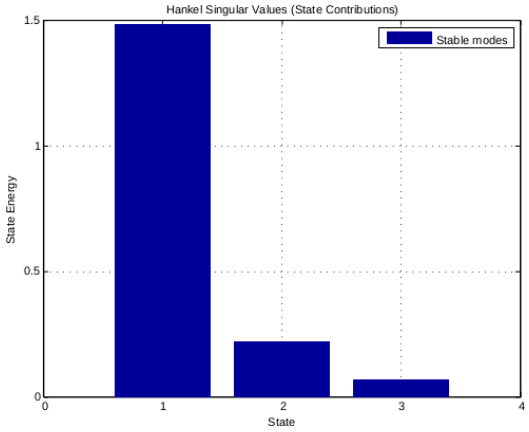

Figure 6.13: The Hankel values of Z_{dd}, as a third order transfer function

it is evident that the third order transfer function is a better fit. However the Hankel values show that the other states can be omitted. One might ask why not more states were omitted from the input admittance, the reason is revealed by inspecting the Hankel values presented in figure 6.14. In this figure one can see that the state energy in the third and second state is the same order of magnitude as the first state, meaning that it should not be omitted.

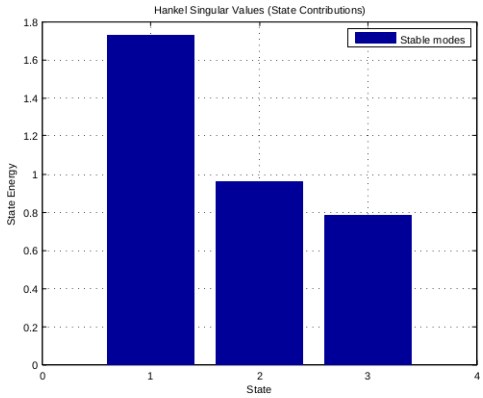

Figure 6.14: The Hanker values of Y_{dd}, as a third order transfer function

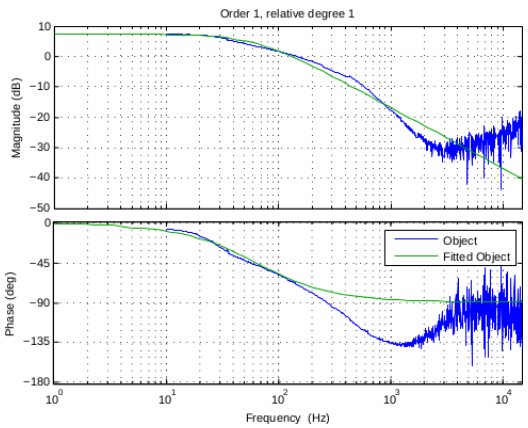

Figure 6.15: The measured and fitted G_{dd}

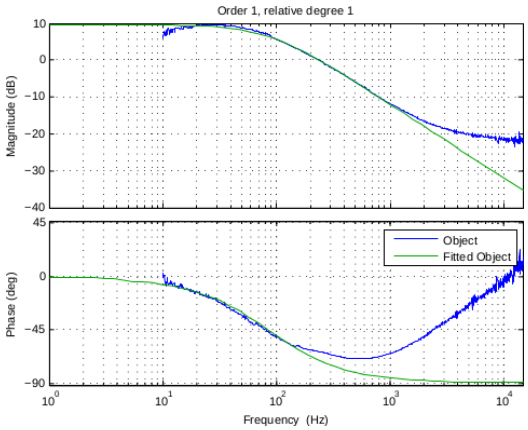

Figure 6.16: The measured and fitted Z_{dd}

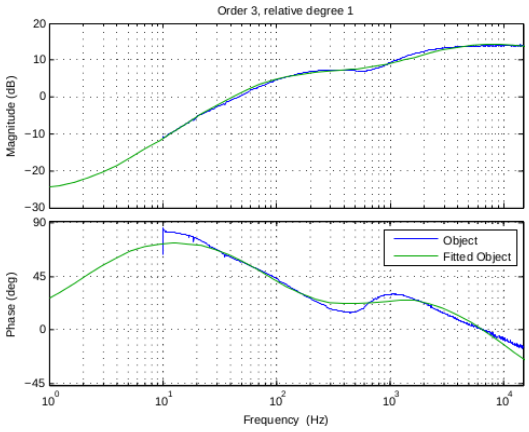

Figure 6.17: The measured and fitted Y_{dd}

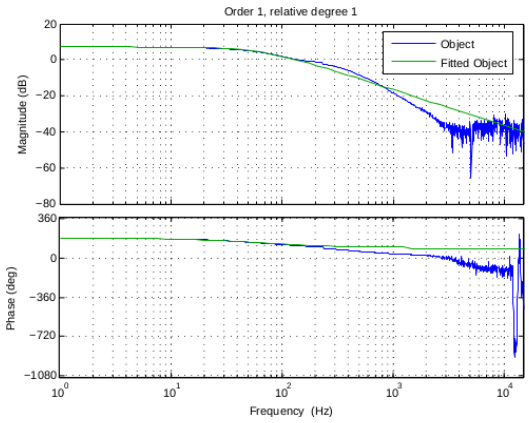

Figure 6.18: The measured and fitted H_{dd}

6.7 Verifying the Obtained Model

Although the goal of this chapter was to develop a simple test bench for obtaining the transfer functions for dc-dc converters. It still has to be confirmed, that the correct transfer functions were obtained. To test this, a load step was made by connecting a resistor in shunt. The step response was recorded by a Tektronix TDS2014C oscilloscope [35]. The system's state and parameters were all calculated by taking the average of the measured curves in the oscilloscope, and are presented in table 6.4 and 6.5. To synchronize the measured and simulated data, the time vector of the simulated data was shifted to align with the measurements. The alignment was done by inspecting figure 6.22.

Table 6.4: System State Before the Load Step

Input Voltage	$13.5968V$
Input Current	$1.9787A$
Output Voltage	$28.4183V$
Output Current	$-0.7031A$
Load Resistance	40.4206Ω

Table 6.5: System State After the Load Step

Input Voltage	$13.5986ViV$
Input Current	$2.1242A$
Output Voltage	$28.3362V$
Output Current	$-0.7774A$
Load Resistance	36.4510Ω

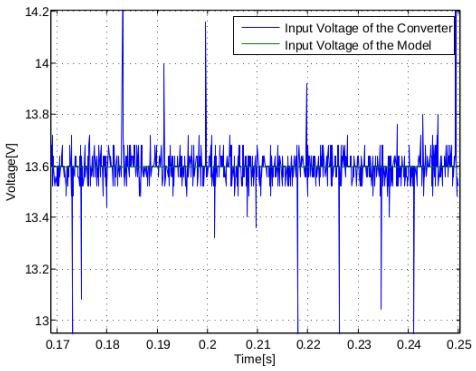

Figure 6.19: The input voltage's transient response

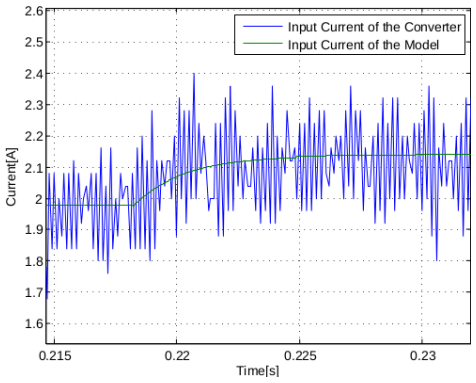

Figure 6.20: The input current's transient response

From the figures one can observe that all the curves fit quite well, except for the output voltage, which settles at a too low value. There might be different reasons for this like the fact that there is a small increase in the input voltage in the real system. Which is not possible in the Simulink

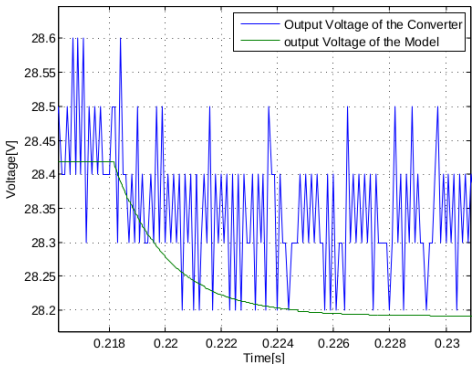

Figure 6.21: The output voltage's transient response

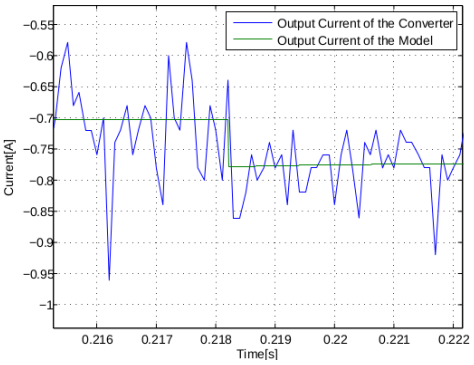

Figure 6.22: The output current's transient response

model, where an ideal source is used. The deviation might also be due to measurement errors, which the output current gives some indication on. As one can see the output current of the model and the converter match quite well. Keeping in mind that the output of the model is given by controlled voltage sources, an incorrect voltage should also give an incorrect current, unless the measured resistance is incorrect. It should also be mentioned that the system was subjected to a 10 percent increase in the load. And that the output voltage's deviation is less than one percent. Furthermore it should be taken into account, that the system is modelled by only first order transfer functions, as the input admittance has to be inactive in the configuration used.

Chapter 7

Applying the G-Parameters Modelling to a Closed Loop Boost Converter

Having a test-bench and procedure in place from the previous chapter, it was decided to move on to a more realistic case. The same converter and measurement system was used, with the major change being the introduction of a simple control system.

7.1 System Configuration and Measurement Configuration

Both the system configuration and the measurement configuration was unchanged from the previous chapter, except for the introduction of a simple control system. The control was implemented using LabVIEW and the data acquisition device NI USB-6216[36]. The control system consisted of a PI-regulator controlling the output current.

7.2 Comparison between the Terminated and Unterminated Transfer Functions

Figures 7.1-7.4 show the terminated and unterminated transfer functions of the closed loop boost converter. It can be seen that the amount of noise in the measurements are quite small, except for the output impedance measurement. The general curve shape of the output impedance, can however, still

be recognized. Also worth noting is the peaks at all the measurements between 20 and 30 Hz. These peaks were not present in the open loop system, and are most likely caused by the dynamics of the controller.

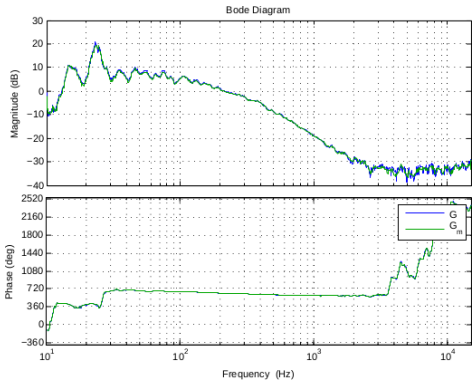

Figure 7.1: The terminated and unterminated audio susceptibility

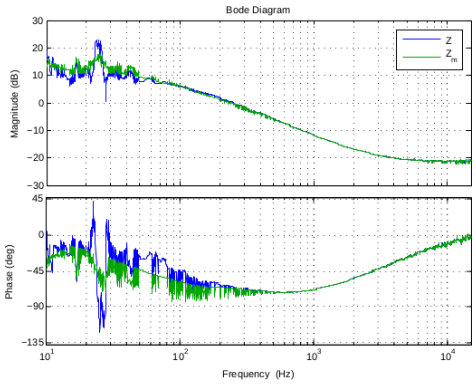

Figure 7.2: The terminated and unterminated output impedance

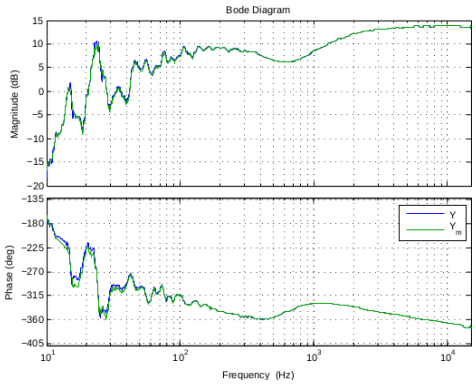

Figure 7.3: The terminated and unterminated input admittance

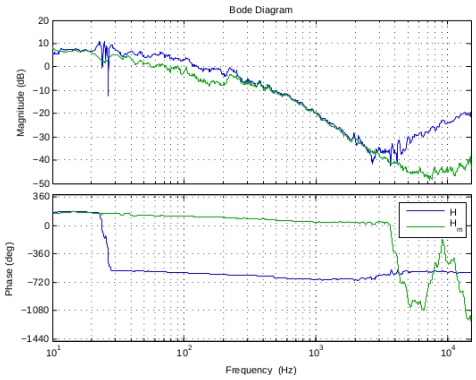

Figure 7.4: The terminated and unterminated back current gain

7.3 Fitting of the Transfer Functions

To complete the modelling transfer functions have to be fitted to the measured data. In the case of the open loop boost converter, it was found that the transfer functions could be approximated by first order transfer functions, except for the input admittance, which could be approximated by a

third order transfer function. In this case, the introduction of a control system increases the required order of the transfer functions. As with the converter in open loop it was found that the input admittance was of the highest order, in this case it could be approximated by a sixth order transfer function. The Audio susceptibility and the output impedance could both be approximated by a third order transfer function, and the back current gain could be approximated by a second order transfer function. It should also be noted, that all the transfer functions contain the peak, between 20Hz and 30Hz.

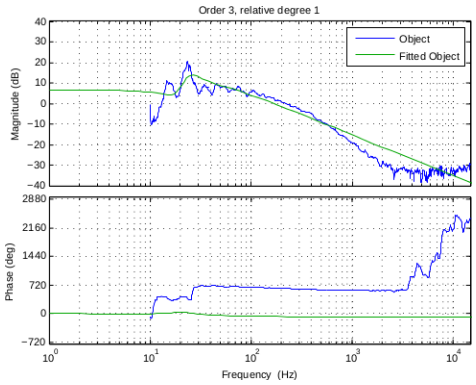

Figure 7.5: fitting of the audio susceptibility

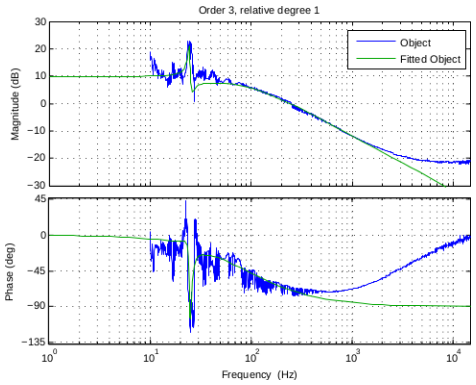

Figure 7.6: Fitting of the output impedance

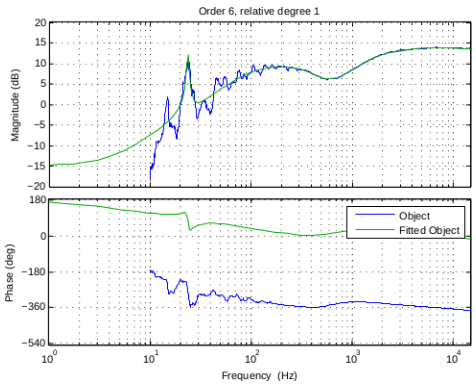

Figure 7.7: Fitting of the input admittance

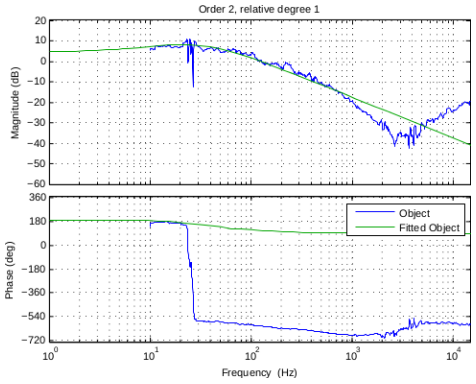

Figure 7.8: Fitting of the output impedance

7.4 Comparison between the Open and Closed Loop Transfer Functions

Inspecting figures 7.9-7.12 one will see that the general curve shapes of the open and closed loop transfer functions are similar, except for the peak in the closed loop ones. The amplitudes are also not always the same.

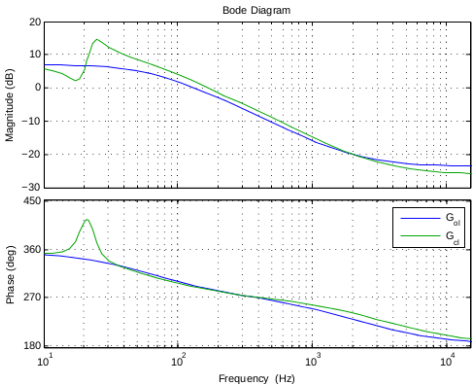

Figure 7.9: The open and closed loop audio susceptibility

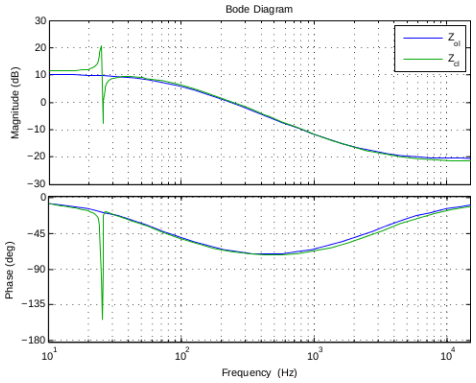

Figure 7.10: The open and closed loop output impedance

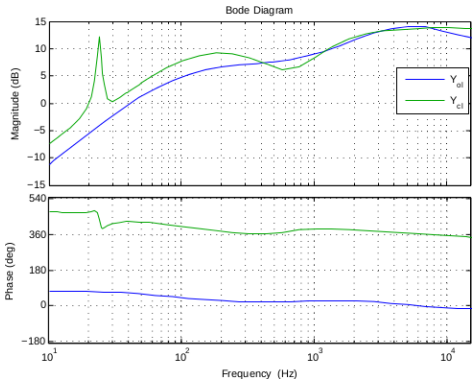

Figure 7.11: The open and closed loop input admittance

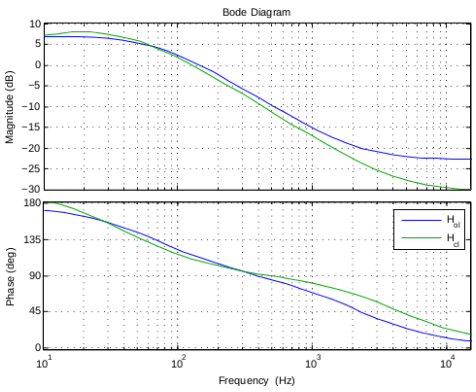

Figure 7.12: The open and closed loop output impedance

Chapter 8

Modelling of a Physical VSI

In this chapter the test bench developed for dc-dc converters will be tested on a VSI.

8.1 The System Configuration

The VSI used for the test was a current controlled VSI in the Sintef lab, using a PI controller implemented in an Opal-RT device. The DC-link was supplied from the ac-grid through an six pulse rectifier. The ac-side was connected to a load in parallel to the ac-grid.

Table 8.1: System Parameters

Filter inductance	$2mH$
Filter capacitor	$50\mu F$
DC capacitor	$3300\mu F$
Load	31Ω
DC voltage	$400V$
AC voltage	$163V$
AC current	$5A$

Table 8.2: PI-controller parameters

k_p	0.5
k_i	65

8.2 The Measurement Configuration

Since the Agilent network analyser only has two inputs, it can not be used on three-phase systems. Instead two Tektronix DPO 4054 oscilloscopes[37] were used. To ensure synchronized measurements, the trigger signal of one of the scopes were used as the trigger input for the other. It was assumed

that the time the trigger signal uses to propagate from one scope to the other is negligible. The sampling frequency of the scopes was set to 100 kS/s, and 100000 samples were collected. The injections were done as line to line injections, in the same fashion as the parallel injection described for dc-dc converters, with a μF ac-capacitor instead of a dc-capacitor. For each measurement set 24 injections were made, where the frequencies to be collected were distributed logarithmically.

8.3 The Measurements

This section is divided into two subsections. One presenting the transfer functions obtained using data filtered by a bandpass filter. The other presenting the transfer functions, after being subjected to a moving average filter.

8.3.1 Post Processing Using a Bandpass Filter

As the modelling could be done quite easily with the help of a network analyser on dc-dc converters, it was decided to do a measurement set using the same network analyser used on the boost converters. The network analyser does not have enough inputs to measure the whole g-parameters model, however it can measure the converters input impedance, which is defined as.

$$Z_i = \frac{\tilde{v}_{id}}{\tilde{i}_{id}} \tag{8.1}$$

In figure 8.1 a comparison between the input impedance calculated from the oscilloscope data, from the filtered oscilloscope data, and the one measured by the network analyser is presented. It is evident from the figure that both methods measure the same impedance. However, the amount of noise in the impedance measured using the network analyser is significantly lower than the other.

The reason for this noise difference is probably the built in low pass filter of the network analyser. To emulate this behaviour each measurement set from the oscilloscope were filtered using a bandpass filter. The filter was designed using the Matlab function design, provided by the signal processing toolbox. A butterworth filter was used with the configuration given in table 8.3. This solution is not as good as using intermediate frequencies and a lowpass filter, however in this case no higher frequencies than $5kHz$ are of any interest. And as can be observed in figures 8.2 and 8.3, it is first at the second highest frequency the filter introduces any attenuation to the

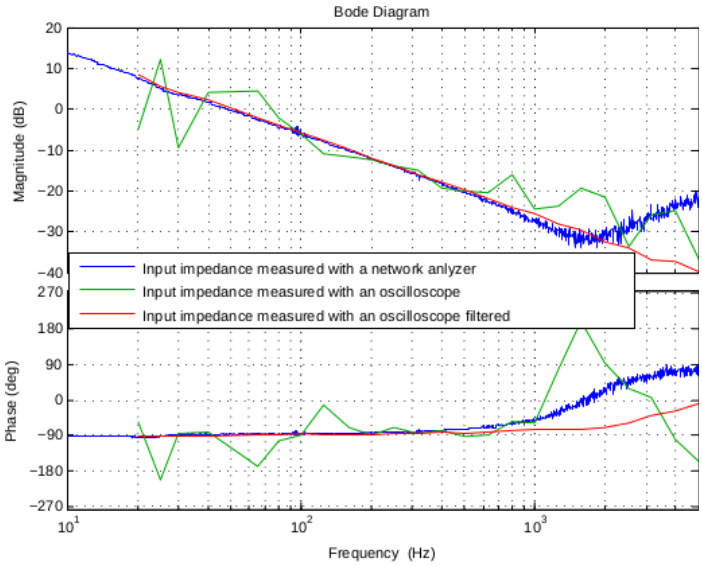

Figure 8.1: Comparison of measurements done with a network analyser and an oscilloscope

Table 8.3: Configuration of the Passband Filter

Passband bandwidth	$10Hz$
Transition band bandwidth	$20Hz$
Passband ripple allowed	$1dB$
Stopband attenuation	$60dB$

frequency of interest. Besides the amplitude of the injection increases with increasing amplitudes, which should counteract the attenuation.

The filtering described above was also applied to the the measurements used to obtain the g-parameters model. In figures **??** and 8.5 a comparison between two of the g-parameters functions obtained with and without filtering is given. Although the functions obtained using filtering is slightly smoother, there is still some noise left. Because of the low number of samples used, the noise might make it difficult to fit transfer functions to the measured data. Since it is difficult to distinguish if a peak is caused by noise, or by the converter's dynamics. Choosing a higher amplitude on the perturbation might also help. Recalling that the same measurement system were used as for the dc-dc converters, which were operating at a much lower power. In this case the injected disturbances might in other words be of a too low amplitude compared to the noise already present in the system. Particularly for lower frequencies, as the perturbations are injected into the system through

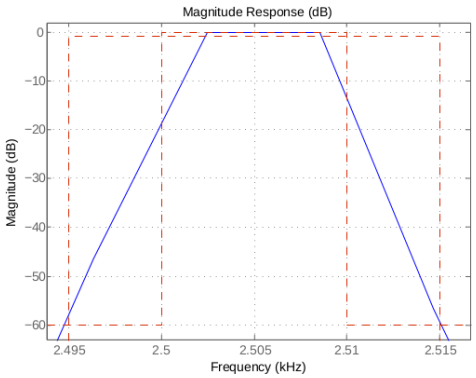

Figure 8.2: Bandpass filter centered at $2505Hz$

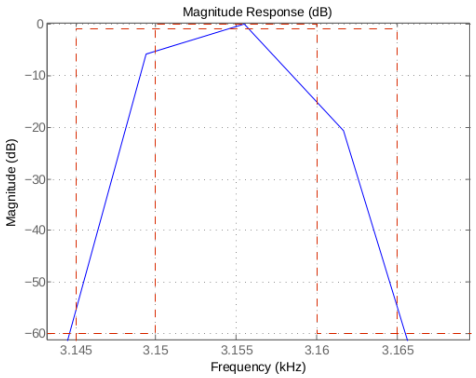

Figure 8.3: Bandpass filter centered at $3155Hz$

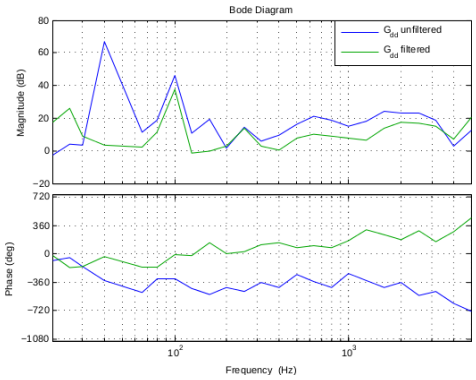

Figure 8.4: The filtered and unfiltered G_{dd}

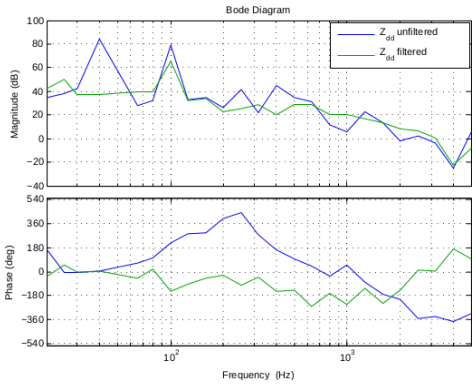

Figure 8.5: The filtered and unfiltered H_{dd}

an capacitor. In figure 8.6 the problem with the small perturbation is illustrated. For the positive sequence injection, a peak at the injection frequency of 315Hz can easily be recognized. This is, however not the case for the negative sequence injection, or the injection at the dc-side. In both these cases the amplitudes at the inspected frequency do not differ substantially from the amplitude at the surrounding frequencies. The picture is better in figure 8.7, where a peak can be recognized at the injection frequency of 795Hz in most of the cases. The exception being the output voltage along the d-axis, from the set where the injections were made at the dc-side.

Moving on to the obtained g-parameters, one can from figures 8.8 to 8.11 observe that they all contain a significant amount of noise, except for the input admittance. All the g-parameters presented here have been subjected

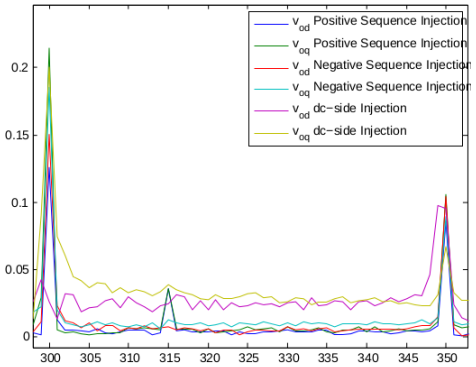

Figure 8.6: The output voltages from all three measurement sets, with an injection at 315Hz

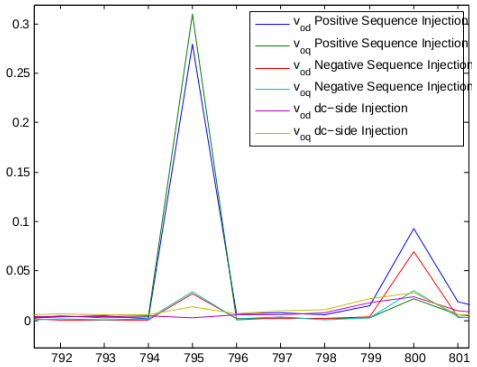

Figure 8.7: The output voltages from all three measurement sets, with an injection at 795Hz

to the filtering technique previously described. As shown in figures 8.4 and 8.5, the filtering did help. however, it might prove problematic to fit transfer functions to the data.

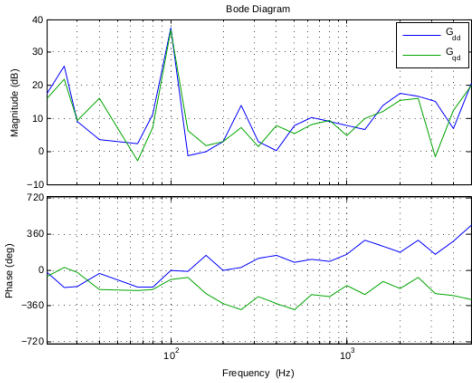

Figure 8.8: The measured audio susceptibilities

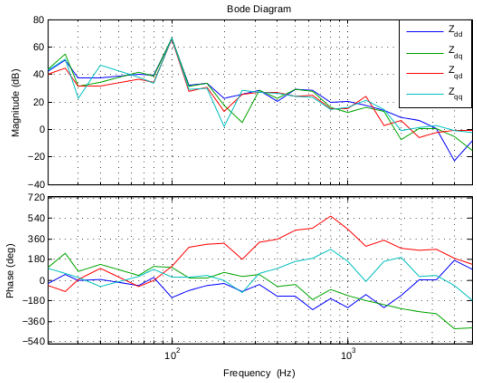

Figure 8.9: The measured output impedances

8.3.2 Post Processing using an Moving Average Filter

Due to the amount of noise still left in the g-parameters after the bandpass filtering, it was decided to use a moving average filter on the calculated g-parameters. A filter window of four was chosen. The resulting g-parameters are presented in figures 8.12 to 8.15.

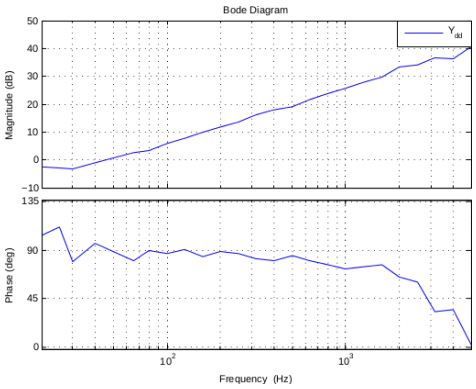

Figure 8.10: The measured input admittance

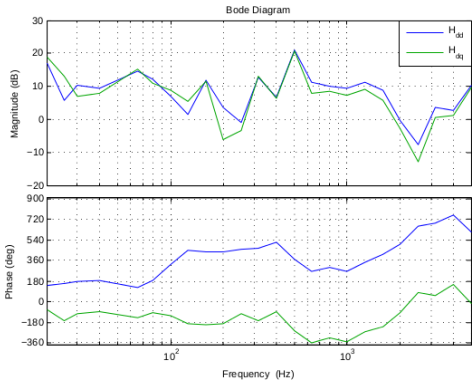

Figure 8.11: The measured back current gains

Even after applying the moving average filtering the measurements contain some noise. The reason for this is the low number of samples. A sampling window of four samples, give in this case a window size of 16.7%, which is quite high. However, as only 24 samples were collected, the distance between some of them is already quite big for a window size of four. Under these circumstances it is difficult to fit transfer functions to the measured data. For instance inspecting figure 8.16 one can see that the fitted transfer function, closely resembles the measured data. The problem is that the fitted system is not stable, as can be observed in figure 8.17. In figures 8.18 and 8.19 the opposite problem is encountered.

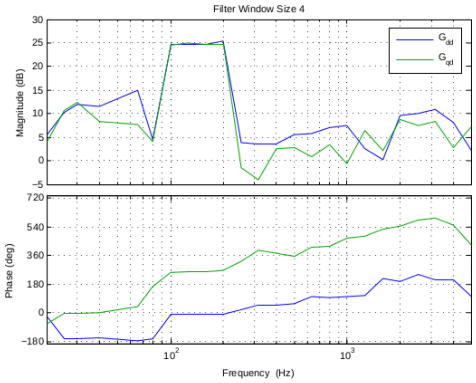

Figure 8.12: The audio susceptibilities subjected to the moving average filter

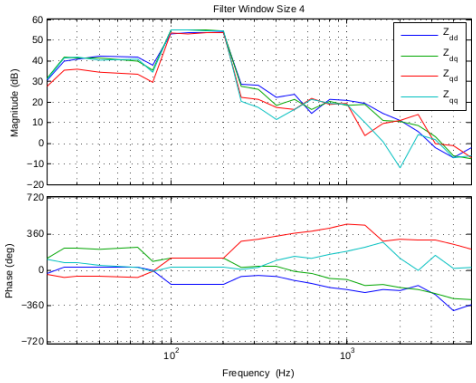

Figure 8.13: The input impedances subjected to the moving average filter

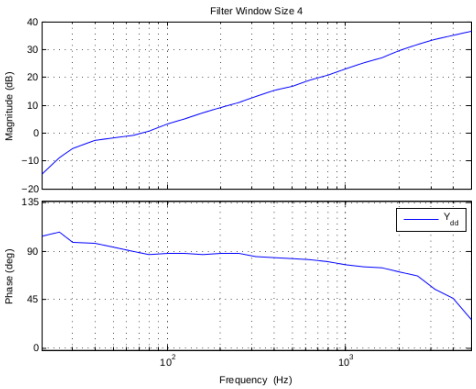

Figure 8.14: The input admittance subjected to the moving average filter

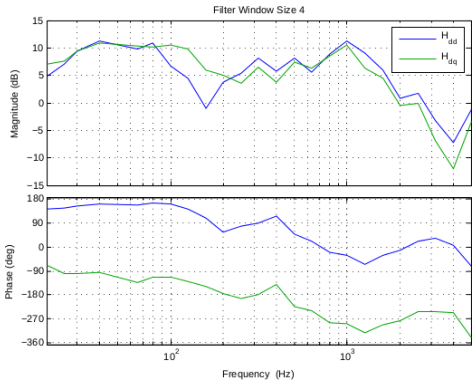

Figure 8.15: The back current gains subjected to the moving average filter

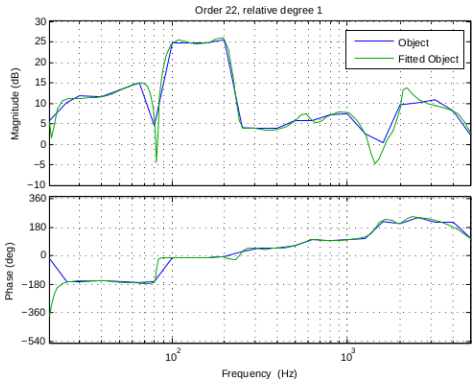

Figure 8.16: Fitted transfer function of G_{dd} assuming an order of 22

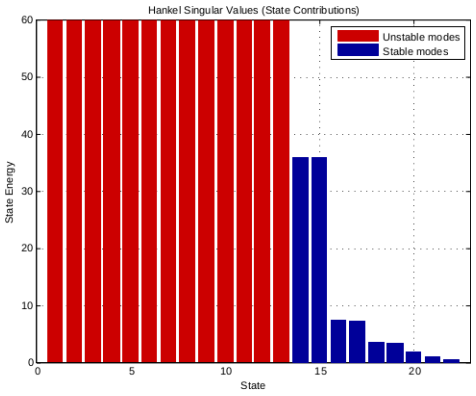

Figure 8.17: Hankel values of fitted transfer function of G_{dd} assuming an order of 22

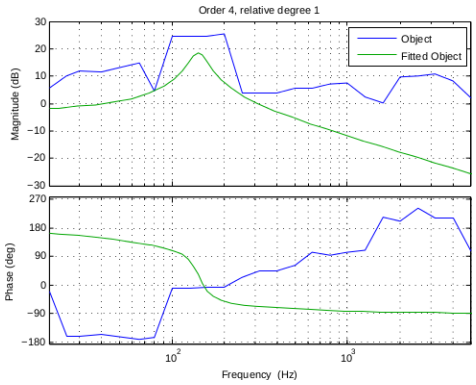

Figure 8.18: Fitted transfer function of G_{dd} assuming an order of 4

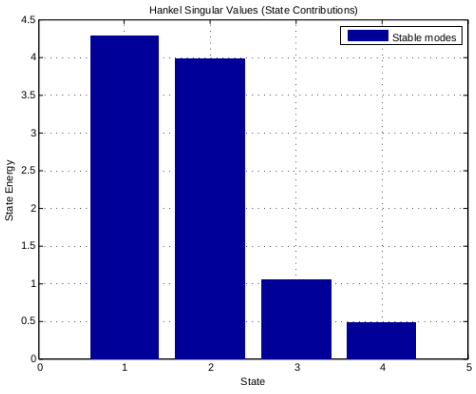

Figure 8.19: Hankel values of fitted transfer function of G_{dd} assuming an order of 4

Chapter 9

Conclusions

Throughout the chapters of this report, different aspects related to the g-parameters modelling of power electronic converters have been investigated. The main conclusions from each of the chapters containing results will be highlighted and discussed in order.

9.1 Modelling of the VSI in Simulink

The main goal of the simulations in Simulink, was to have a mean of quickly testing different approaches to obtaining the model. With the added advantage of having full control over the system design. It provided a robust and flexible platform, giving direction and aid for the investigations to come.

9.1.1 Verification of the Methodology

Although the methodology has already been verified through simulations of ac-dc converters in [17]. It was important to verify the methodology to give relevance to the rest of the report. The successful development of Simulink models, implementing the g-parameters model in both the dq and abc planes, show that the modelling can be used to simulate converters connected in grids. Further strengthening the models main appeal as an alternative to average based models. Comparing the VSI's and the model's response to load steps, revealed the performance to be quite satisfactory up till load steps around 10 percent.

9.1.2 On the Perturbation Injection

Different perturbation injection schemes were shown to yield the same results, regardless if the injections were done in series or parallel. Although the

injection technique, where the injections were injected separately into the d and q axis obtained measured transfer functions closest to the real ones. It is not the recommended choice, due to it complexity and equipment count. The recommended strategy is the technique, where one measurement is done after injecting a positive sequence perturbation, and the next one by injecting a negative sequence perturbation. The advantage of this technique being its simple implementation and low equipment count. Regarding the size of the perturbations, the main conclusion was that they should be big enough to be easily observable in the frequency domain, however not so big as to make the system unstable.

9.1.3 The Stability Criterion

The stability criterion successfully predicted the model's stability limit, using the criterion for the input voltage. The drawback being that this did not correspond with the real converter's stability limit. The stability criterion's failure to predict the unstability, which occurred when connecting a capacitive load proves that the criterion should be further investigated.

9.2 Modelling of the Boost Converters

The work on the boost converters were undertaken to gain insight into the practical issues of the modelling approach. The main concern being how to physical implement the perturbation injection schemes investigated in Simulink. It was found that the perturbations could easily be injected into the systems using a isolating transformer a signal source and a linear amplifier. If it is desirable to do the injections in parallel a capacitor should also be added. As in the investigation in Simulink it was found that the injections could be done both in series or parallel. Potential noise due to the injection strategy, can be filtered with the measurement equipment or by post processing the data in software. All in all it is an easily implementable solution, where the measurement and signal generation system could possible be implemented in Simulink or LabView.

9.3 Modelling of a Physical VSI

The input impedance of the converter was successfully obtained, using a bandpass filtering technique, showing that the measurement configuration can be used to obtain transfer functions. Using the two filtering techniques on the measured data, also gave an indication on the shape of the g-parameters.

However to obtain a fully functional model more measurements would need to be taken, particularly around the peak at 100 Hz. Also it would be of an advantage to inject perturbations of an higher amplitude, however due to the limitations of the available amplifier this was not possible.

9.4 Overall Conclusions

The G-parameters modelling show promising features for future use in simulation of converters, in which the parameters are unknown. DC-DC and Simulink converters were easily modelled. Some work is still needed to model converters also featuring at least one ac-side. However the test bench developed should give a strong basis for continuing the work towards accomplishing this.

More work is needed on the stability criterion of the model. The methodology can, however be used to obtain other models, where more investigated stability criteria exist. In light of this, the current lack of a working stability criterion, should not be considered as a major drawback.

Chapter 10

Further Work

Based upon the conclusions it is clear that further work should be done to successfully simulate a physical VSI using the proposed methodology. The first step towards accomplishing this could be to use the same test bench used in this report, however with a substantial higher number of injections. For the Simulink implemented VSI 47 injections were made, around the same number of injections should be sufficient for a physical VSI running at $10kHz$. Furthermore it would be of an advantage to develop a measurement and injection system, this could for instance be done using a DAQ device and LabVIEW. A system like this could automate the methodology described in this report, thus substantially decrease the time needed to obtain models. It could also be used to obtain other measurement based models of converters, as the same measurement sets can be used.

Efforts should also be made in developing a good stability criterion. A good starting point might be dc-dc converters, where less parameters need to be considered. Investigating such simple systems, the difference between the admittances and impedances obtained using the g-parameters method could be compared, to the ones obtained from models, where the converters are considered to be only impedances or admittances.

Finally the modelling should be used in simulations, where its performance should be compared to switching models and average models. This work would require a variety of converters to be modelled. It would thus be easier to do, if one had a measurement and injection system, as previously described. Ones again indicating that most urgency should be placed in further developing the methodology to obtain the models.

Bibliography

[1] K. Kusakana, J. Munda, and A. Jimoh, "Feasibility study of a hybrid pv-micro hydro system for rural electrification," in *AFRICON, 2009. AFRICON'09.*, pp. 1–5, IEEE, 2009.

[2] N. Ahmed, S. Norrga, H.-P. Nee, A. Haider, D. Van Hertem, L. Zhang, and L. Harnefors, "Hvdc supergrids with modular multilevel converters - the power transmission backbone of the future," in *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pp. 1–7, 2012.

[3] T. Haileselassie and K. Uhlen, "Power system security in a meshed north sea hvdc grid," *Proceedings of the IEEE*, vol. 101, no. 4, pp. 978–990, 2013.

[4] T. Haileselassie, K. Uhlen, J. Tande, and O. Anaya-Lara, "Connection scheme for north sea offshore wind integration to uk and norway: Power balancing and transient stability analysis," in *PowerTech, 2011 IEEE Trondheim*, pp. 1–5, 2011.

[5] J. Setreus and L. Bertling, "Introduction to hvdc technology for reliable electrical power systems," in *Probabilistic Methods Applied to Power Systems, 2008. PMAPS '08. Proceedings of the 10th International Conference on*, pp. 1–8, 2008.

[6] S. Sanchez Acevedo and M. Molinas, "Modeling of switching power interfaces for smart-grid stability studies," in *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on*, pp. 1 –6, dec. 2011.

[7] J. Sun, "Impedance-based stability criterion for grid-connected inverters," *Power Electronics, IEEE Transactions on*, vol. 26, no. 11, pp. 3075–3078, 2011.

[8] R. Burgos, D. Boroyevich, F. Wang, K. Karimi, and G. Francis, "On the ac stability of high power factor three-phase rectifiers," in *Energy Conversion Congress and Exposition (ECCE), 2010 IEEE*, pp. 2047–2054, 2010.

[9] J. Huang, K. Corzine, and M. Belkhayat, "Small-signal impedance measurement of power-electronics-based ac power systems using line-to-line current injection," *Power Electronics, IEEE Transactions on*, vol. 24, pp. 445 –455, feb. 2009.

[10] Y. Familiant, K. Corzine, J. Huang, and M. Belkhayat, "Ac impedance measurement techniques," in *Electric Machines and Drives, 2005 IEEE International Conference on*, pp. 1850–1857, 2005.

[11] V. Valdivia, A. Lazaro, A. Barrado, P. Zumel, C. Fernandez, and M. Sanz, "Impedance identification procedure of three-phase balanced voltage source inverters based on transient response measurements," *Power Electronics, IEEE Transactions on*, vol. 26, no. 12, pp. 3810–3816, 2011.

[12] L. Ljung, "Black-box models from input-output measurements," in *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*, vol. 1, pp. 138–146 vol.1, 2001.

[13] J. Lee, B. Cho, S. Kim, and F. Lee, "Modeling and simulation of spacecraft power systems," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 24, no. 3, pp. 295–304, 1988.

[14] B. Cho and F. Lee, "Modeling and analysis of spacecraft power systems," *Power Electronics, IEEE Transactions on*, vol. 3, no. 1, pp. 44–54, 1988.

[15] L. Amedo, R. Burgos, F. Wang, and D. Boroyevich, "Black-box terminal characterization modeling of dc-to-dc converters," in *Applied Power Electronics Conference, APEC 2007 - Twenty Second Annual IEEE*, pp. 457 –463, 25 2007-march 1 2007.

[16] I. Cvetkovic, D. Boroyevich, P. Mattavelli, F. Lee, and D. Dong, "Unterminated small-signal behavioral model of dc-dc converters," *Power Electronics, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2012.

[17] I. Cvetkovic, M. Jaksic, D. Boroyevich, P. Mattavelli, F. Lee, Z. Shen, S. Ahmed, and D. Dong, "Un-terminated, low-frequency terminal-behavioral d-q model of three-phase converters," in *Energy Conversion Congress and Exposition (ECCE), 2011 IEEE*, pp. 791 –798, sept. 2011.

[18] I. Cvetkovic, D. Boroyevich, D. Dong, P. Mattavelli, R. Burgos, M. Jaksic, G. Francis, Z. Shen, S. Ahmed, and F. Wang, "Dynamic interactions in hybrid ac/dc electronic power distribution systems," in *Power Electronics and ECCE Asia (ICPE ECCE), 2011 IEEE 8th International Conference on*, pp. 2121 –2128, 30 2011-june 3 2011.

[19] D. Boroyevich, R. Burgos, L. Arnedo, and F. Wang, "Synthesis and integration of future electronic power distribution systems," in *Power Conversion Conference - Nagoya, 2007. PCC '07*, pp. K–1–K–8, 2007.

[20] J. W. Nilsson and S. A. Riedel, *Electric Circuits*. Pearson Prentice Hall, 2008.

[21] J. Balchen, M. Fjeld, and O. Solheim, *Reguleringsteknikk*, vol. 5. Tapir, 1971.

[22] J. Sun, "Impedance-based stability criterion for grid-connected inverters," *Power Electronics, IEEE Transactions on*, vol. 26, no. 11, pp. 3075–3078, Nov.

[23] G. C. Verghese and V. Thottuvelil, "Aliasing effects in pwm power converters," in *Power Electronics Specialists Conference, 1999. PESC 99. 30th Annual IEEE*, vol. 2, pp. 1043–1049 vol.2, 1999.

[24] "Matlab dft." `http://www.mathworks.se/help/matlab/math/discrete-fourier-transform-dft.html?s_tid=doc_12b`.

[25] V. Blasko and V. Kaura, "A new mathematical model and control of a three-phase ac-dc voltage source converter," *Power Electronics, IEEE Transactions on*, vol. 12, pp. 116 –123, jan 1997.

[26] C. Bajracharya, M. Molinas, J. Suul, T. Undeland, *et al.*, "Understanding of tuning techniques of converter controllers for vsc-hvdc," in *Nordic Workshop on Power and Industrial Electronics (NORPIE/2008), June 9-11, 2008, Espoo, Finland*, Helsinki University of Technology, 2008.

[27] C. Bajracharya, "Control of vsc-hvdc for wind power," Master's thesis, Norwegian University of Science and Technology, 2008.

[28] *Schematics of the Power Electronic Board*.

[29] Department of Electrical and Computer Engineering University of Minnesota, *Power Electronics Laboratory User Manual*, September 2011.

[30] *Manual Instek GPC-3030DQ*.

[31] "Agilent technologies e5061b ena series network analyzer." http://www.home.agilent.com/en/pd-1668980-pn-E5061B/ena-series-network-analyzer?&cc=NO&lc=eng.

[32] "Harrison 6824a." http://www.hpmemory.org/present/contribution/db_instr_02.htm.

[33] *Fluke 80i-110s AC/DC Current Probe Instruction Sheet.*

[34] *High-voltage Differential Probes.*

[35] "Tektronix tds2000c series datasheet." http://www.tek.com/sites/tek.com/files/media/media/resources/TDS2000C-Digital-Storage-Oscilloscope-Datasheet-4_1.pdf.

[36] "Ni usb-6216." http://sine.ni.com/nips/cds/view/p/lang/no/nid/209147.

[37] "Mso4000 dpo4000 mixed signal oscilloscope datasheet." http://www.tek.com/datasheet/oscilloscope/mso4000-dpo4000-mso4000-dpo4000-mixed-signal-oscilloscope-datasheet.

Appendix A

Class Used to Obtain the Simulink Model

```
classdef TPMeasurement

    properties
        % The terminated g-parameters
        G_ddm
        G_qdm
        Y_ddm
        TG_ddm
        TG_qdm

        Z_ddm
        Z_qdm
        H_ddm
        TR_ddm

        Z_dqm
        Z_qqm
        H_dqm
        TR_dqm
        TI_qdm
        TI_dqm
        Terminated = 0;

        % The unterminated g-parameters
        G_dd
        G_qd
        Y_dd
        Z_dd
        Z_qd
        H_dd
        Z_dq
        Z_qq
        H_dq
        UnTerminated = 0;

        fp                  % The frequencies injected in Hz
        wp                  % The frequencies injected in rad/s

        % System Data
        ModelName           % The Model name
        MeasurementsNum     % Number of measurements
        SwitchingFreq       % The converters swithcing frequency
        ACVoltage           % The AC bus three-phase rms voltage
        ACPower             % The three phase power at the AC-bus
```

```
        DCVoltage           % The DC-link voltage
        PerturbationAmp     % Amplitude of the perturbation
        deltaT              % The simulation time step
        Fs                  % The samples frequency
        t0                  % Time where the system is considered steady
        l                   % Lenght of the samples vector
        tend                % End of simulation time
        samples             % The samples which are collected

        % Name of Simulink Output Variables
        vi = 'v1'            % Input voltage
        ii = 'i1'            % Input current
        vo = 'v2'            % Output voltage
        io = 'i2'            % Input current

        % Base Values
        ACBaseVoltage       % The AC base voltage
        ACBaseCurrent       % The AC base current
        BasePower           % The base power
        DCBaseVoltage       % The DC base voltage
        DCBaseCurrent       % The DC base current


    end

    methods
        function obj = measure_functions (obj)
            if can_simulate (obj)
                % End of simulation time
                obj.tend = obj.t0 + (obj.l-1)*obj.deltaT;

                % The samples I want to collect
                obj.samples = floor(obj.t0/obj.deltaT):floor(obj.tend/obj.deltaT);

                % The sample frequency
                obj.Fs = 1/obj.deltaT;


                %% Get data for the base case
                v_dc_p = 0;
                id = 0;
                iq = 0;
                assignin ('base', 't0', obj.t0);
                assignin ('base', 'i0', 0);
                assignin ('base', 'deltaT', obj.deltaT);
                f_p = 0;
                obj.to_workspace (v_dc_p, id, iq, f_p); % Initialize variables
                [V1m_base V2m_base I1m_base I2m_base] = obj.collect_data();

                %% Measure the g-parameters

                %% Code for the frequencies

                f_sw = obj.SwitchingFreq; % Less to write
                n = obj.MeasurementsNum;  % Less to write

                % The frequencies I want to record
                indexes = unique(double(uint64(logspace(log10(2),...
                    log10(f_sw/2*obj.l/obj.Fs + 1),n))));

                n = length(indexes);
                %indexes = double(uint64(logspace(log10(2),log10(500*l/Fs + 1),n)));
                % Frequencies actually in the fourier transform
                obj.fp = obj.Fs/obj.l*linspace(0, obj.l-1, obj.l);

                % The g-parameters also have to be initialized
                obj.G_ddm = zeros(n,1);
                obj.G_qdm = zeros(n,1);
                obj.Y_ddm = zeros(n,1);
                obj.TG_ddm = zeros(n,1);
                obj.TG_qdm = zeros(n,1);

                obj.Z_ddm = zeros(n,1);
```

```
obj.Z_qdm = zeros(n,1);
obj.H_ddm = zeros(n,1);
obj.TR_ddm = zeros(n,1);

obj.Z_dqm = zeros(n,1);
obj.Z_qqm = zeros(n,1);
obj.H_dqm = zeros(n,1);
obj.TR_dqm = zeros(n,1);
obj.TI_qdm = zeros(n,1);
obj.TI_dqm = zeros(n,1);

% To store the frequencies recorded
obj.wp = zeros(n,1);

% To index the matrices properly
m = 0;

for index = indexes
    %% Measurement set 1 from the paper
    %initialize stuff
    f_p = obj.fp(index);
    m = m + 1;
    obj.wp(m) = 2*pi*f_p;
    v_dc_p = obj.PerturbationAmp*obj.DCBaseVoltage;
    id = 0;
    iq = 0;
    obj.to_workspace (v_dc_p, id, iq, f_p);


    % This code runs the simulation
    [V1m V2m I1m I2m] = obj.collect_data ();

    temp = V1m(index)- V1m_base(index);

    obj.G_ddm(m) = (V2m(index,1) - V2m_base(index,1))/temp;
    obj.G_qdm(m) = (V2m(index,2) - V2m_base(index,2))/temp;
    obj.Y_ddm(m) = (I1m(index) - I1m_base(index))/temp;
    obj.TG_ddm(m) = (I2m(index,1) - I2m_base(index,1))/temp;
    obj.TG_qdm(m) = (I2m(index,2) - I2m_base(index,2))/temp;

    %% Measurement set 3 from the paper
    %initialize stuff
    v_dc_p = 0;
    id = obj.PerturbationAmp*obj.ACBaseCurrent;
    iq = 0;
    obj.to_workspace (v_dc_p, id, iq, f_p);
    % This code runs the simulation
    [V1m V2m I1m I2m] = obj.collect_data ();

    temp = I2m(index,1) - I2m_base(index,1);

    obj.Z_ddm(m) = (V2m(index,1) - V2m_base(index,1))/temp;
    obj.Z_qdm(m) = (V2m(index,2) - V2m_base(index,2))/temp;
    obj.H_ddm(m) = (I1m(index) - I1m_base(index))/temp;
    obj.TR_ddm(m) = (V1m(index) - V1m_base(index))/temp;
    obj.TI_qdm(m) = (I2m(index,2) -I2m_base(index,2))/temp;


    %% Measurement set 4 from the paper
    v_dc_p = 0;
    id = 0;
    iq = obj.PerturbationAmp*obj.ACBaseCurrent;
    obj.to_workspace (v_dc_p, id, iq, f_p);

    % This code runs the simulation
    [V1m V2m I1m I2m] = obj.collect_data ();

    temp = I2m(index,2)- I2m_base(index,2);

    obj.Z_dqm(m) = (V2m(index,1) - V2m_base(index,1))/temp;
    obj.Z_qqm(m) = (V2m(index,2) - V2m_base(index,2))/temp;
    obj.H_dqm(m) = (I1m(index) - I1m_base(index))/temp;
    obj.TR_dqm(m) = (V1m(index) - V1m_base(index))/temp;
```

```
            obj.TI_dqm(m) = (I2m(index,1) - I2m_base(index,2))/temp;
        end
        obj.Terminated = 1;
    end
end
function [V1m V2m I1m I2m] = collect_data (obj)
    % This code runs the simulation
    simOut = sim(obj.ModelName, 'ReturnWorkSpaceOutputs', 'on',...
        'StopTime', num2str(obj.tend),...
        'FixedStep', num2str(obj.deltaT));

    % Extract the simulation data, in the time domain
    v1m = simOut.get(obj.vi);
    v2m = simOut.get(obj.vo);
    i1m = simOut.get(obj.ii);
    i2m = simOut.get(obj.io);

    % First I find the fourier transforms with no perturbations
    V1m = fft(v1m(obj.samples));
    V2m = fft(v2m(obj.samples,:));
    I1m = fft(i1m(obj.samples));
    I2m = fft(i2m(obj.samples,:));
end
function retVal = is_AC_set (obj)
    retVal = ~isempty(obj.ACVoltage) & ~isempty(obj.ACPower);
end
function retVal = is_DC_set (obj)
    retVal = ~isempty(obj.DCVoltage) & ~isempty(obj.ACPower);
end
function retVal = is_base_values_set (obj)
    retVal = ~isempty(obj.ACBaseVoltage) & ~isempty(obj.ACBaseCurrent)...
        & ~isempty(obj.BasePower) & ~isempty(obj.DCBaseCurrent)...
        & ~isempty(obj.DCBaseVoltage);
end
function obj = calculate_bases (obj)
    if is_AC_set(obj)
        obj.BasePower = obj.ACPower;
        obj.ACBaseVoltage = sqrt(2/3)*obj.ACVoltage; % The base voltage defined as the phase to
        obj.ACBaseCurrent = 2/3*obj.BasePower/obj.ACBaseVoltage; % The base current defined as
        % The DC bases defining a power invariant base system
        obj.DCBaseVoltage = 2*obj.ACBaseVoltage;
        obj.DCBaseCurrent = 3/4*obj.ACBaseCurrent;

    elseif is_DC_set (obj)
            obj.BasePower = obj.ACPower;
            obj.DCBaseVoltage = obj.DCVoltage;
            obj.ACBaseVoltage = obj.DCBaseVoltage;
            obj.ACBaseCurrent = 2/3*obj.BasePower/obj.ACBaseVoltage;
            obj.DCBaseCurrent = 3/4*obj.DCBaseVoltage;
    end
end
function retVal = can_simulate (obj)
    retVal = ~isempty(obj.ModelName) & ~isempty(obj.MeasurementsNum)...
        & ~isempty(obj.SwitchingFreq) & ~isempty(obj.PerturbationAmp) &...
        ~isempty(obj.deltaT) & ~isempty(obj.t0) & ~isempty(obj.l);
end
function obj = unterminate (obj)
    if obj.Terminated
        for i = 1:length(obj.Z_ddm)
            D = [   1               obj.TR_ddm(i)   obj.TR_dqm(i)
                    obj.TG_ddm(i)   1               obj.TI_dqm(i)
                    obj.TG_qdm(i)   obj.TI_qdm(i)   1];

            % Matrix containing the terminated transfer functions
            terminated = [  obj.G_ddm(i)   obj.Z_ddm(i)   obj.Z_dqm(i)
                            obj.G_qdm(i)   obj.Z_qdm(i)   obj.Z_qqm(i)
                            obj.Y_ddm(i)   obj.H_ddm(i)   obj.H_dqm(i)];

            % The calculation of the unterminated transfer functions
            unterminated = terminated/D;

            %% Extract transfer functions
```

```
                    obj.G_dd(i) = unterminated(1,1);
                    obj.G_qd(i) = unterminated(2,1);
                    obj.Z_dd(i) = unterminated(1,2);
                    obj.Z_dq(i) = unterminated(1,3);
                    obj.Z_qd(i) = unterminated(2,2);
                    obj.Z_qq(i) = unterminated(2,3);
                    obj.Y_dd(i) = unterminated(3,1);
                    obj.H_dd(i) = unterminated(3,2);
                    obj.H_dq(i) = unterminated(3,3);
                end
                obj.UnTerminated = 1;
            end
        end
    end
    methods(Static)
          function to_workspace (v_dc_p, id, iq, f_p)
            assignin ('base', 'v_dc_p', v_dc_p);
            assignin ('base', 'id', id);
            assignin ('base', 'iq', iq);
            assignin ('base', 'f_p', f_p);
          end
    end
end
```

Appendix B

Class Used to Contain the GParameters Model

```
classdef TPModel

    properties
        G_dd
        G_qd
        Z_dd
        Z_dq
        Z_qd
        Z_qq
        Y_dd
        H_dd
        H_dq
    end
    methods
        function obj = TPModel ()
            prop = properties (obj);
            for i = 1:numel(prop)
                obj.(prop{i}) = TPTransfer ();
            end
        end
    end
end
```

Appendix C

Class Used to Contain the Transfer Functions in the Model

```
classdef TPTransfer
    properties
        func
        order
        rd
    end
    methods
        function obj = TPTransfer ()
            obj.func = 0;
            obj.order = 0;
            obj.rd = 0;
        end
    end
end
```

Appendix D

Function Used to Fit the Transfer Functions

```
function fitterRD1 (object, startorder, endorder,maxFreq)

P = bodeoptions;
P.frequnits = 'Hz';
P.Xlim = [20 maxFreq];
P.Grid = 'on';
while startorder <= endorder
        figure
        bode (object,P)
        hold on
        bode (fitfrd(object, startorder, 1),P)
        legend ('Object', 'Fitted Object')
        title (['Order ' int2str(startorder) ', relative degree ' int2str(1)])

    startorder = startorder + 1;
end
end
```

Appendix E

Function Used to Extract Transfer Functions from the Network Analyser's Data

```
function [fp, tf] = getBoostTF (magPath, phasePath, dBCorr)
    temp = csvread(magPath,3,0);
    fp = temp(:,1);
    mag= temp(:,2)+dBCorr; % Correct gain from measurement
    mag = 10.^(mag/20); % Convert the magnitude to absolute val
    temp = csvread(phasePath,3,0);
    phase = temp(:,2)*pi/180; % Convert to radians
    tf = mag.*exp(1i.*phase);
end
```

Appendix F

Class Used to Obtain Data from Tektronix tds2000c

```
classdef ScopeData
    properties
        recordLength;
        deltaT;
        time;
        data;
        verticalOffSet;
    end

    methods
        function obj = ScopeData (filename)
            obj.recordLength = csvread(filename, 0, 1,[0,1,0,1]);
            obj.deltaT = csvread(filename, 1,1,[1,1,1,1]);
            obj.time = csvread(filename, 0,3,[0,3,obj.recordLength-1,3]);
            obj.data = csvread(filename, 0,4,[0,4,obj.recordLength-1,4]);
            obj.verticalOffSet = csvread(filename,9,1,[9,1,9,1]);

            % Fix the problem with negative time values
            %if min (obj.time) < 0
                obj.time = obj.time - obj.time(1);
            %end
            % Fix the problem with false values
            if sum(obj.data == min(obj.data)) >= 0.1*obj.recordLength
                temp = find(obj.data == min(obj.data));
                if min(temp) || max(temp) == obj.recordLength
                    obj.data = obj.data(obj.data ~= min(obj.data));
                else
                    left = 1:(min(temp)-1);
                    right = (max(temp)+1):obj.recordLength;
                    temp = obj.data(left);
                    obj.data = obj.data(right);
                    obj.data = [obj.data' temp']';
                end

                obj.recordLength = numel(obj.data);
                obj.time = 0:obj.deltaT:(obj.recordLength-1)*obj.deltaT;
            end
        end
    end
end
```

Appendix G

Class Used to Obtain Data from a Generic Tektronix Oscilloscope

```
classdef TekTroData
    properties
        samples = 0;
        deltaT = 0;
        time = 0;
        data = 0;
        channels = 0;
    end

    methods
        function obj = TekTroData (filename, channels, samples, timePos)
            obj.channels = channels;
            obj.samples = samples;
            obj.data = csvread(filename, timePos ,0,...
                [timePos, 0, timePos+obj.samples-1, obj.channels]);
            obj.time = obj.data(:,1);
            obj.data = obj.data(:,2:channels+1);
            obj.deltaT = obj.time(2)-obj.time(1);
        end
    end
end
```

Appendix H

Matlab Script Used to Create the Measurement Sets for the Physical VSI

```
%% Experiment Definitions
fs = 50;
fp = [20 25 30 40 65 80 100 125 160 200 250 315 400 500 630 795 1000 1255 ...
    1585 1990 2505 3155 3970 5000];
fpp = fs + fp;
fpn = abs(-fs+fp);
L = 1e5;
channels = 4;

%% Initialize Stuff
Mdir = 'Experiment/VSI_Sintef/Take Two/';
Bdir = 'Experiment/VSI_Sintef/';
scope1Dir = 'M_SC1/';
scope2Dir = 'M_SC2/';

setI = GPSetSintef(fp,L);
setII = GPSetSintef(fp,L);
setIII = GPSetSintef(fp,L);
setBase = GPSetSintef(fp,L);

%% Collect the Base Case in a Hackish Way
nDir='Negative_Sequence_Output/';
filename1 = [Mdir,nDir,scope1Dir];
filename2 = [Mdir,nDir,scope2Dir];

setBase = setBase.createSet(filename1,filename2,channels);

% The hack is to assume that injecting 10Hz doesn't work. However, to be
% safe I set the 10 Hz stuff to zero
%setBase.vo(:,4);
%setBase.io(:,4);
%setBase.vi(:,4);
%setBase.io(:,4);


%% Do Stuff for the Positive Perturbation
pDir='Positive_Sequence_Output/';

filename1 = [Mdir,pDir,scope1Dir];
filename2 = [Mdir,pDir,scope2Dir];
```

```
setI = setI.createSet(filename1,filename2,channels);

%% Do Stuff for the Negative Perturbation
nDir='Negative_Sequence_Output/';

filename1 = [Mdir,nDir,scope1Dir];
filename2 = [Mdir,nDir,scope2Dir];

setII = setII.createSet(filename1,filename2,channels);

%% Do Stuff for the Input Perturbation
iDir='Input/';

filename1 = [Mdir,iDir,scope1Dir];
filename2 = [Mdir,iDir,scope2Dir];

setIII = setIII.createSet(filename1,filename2,channels);
```

Appendix I

Generic Class Used to Contain the Measurement Sets for the Physical VSI

```
classdef GPSet
    properties
        vo;
        io;
        vi;
        ii;
        fp;
        L;
    end

    methods
        function obj = GPSet(fp,L)
            n=numel(fp);
            obj.vo = zeros(2,n,L);
            obj.io = zeros(2,n,L);
            obj.vi = zeros(n,L);
            obj.ii = zeros(n,L);
            obj.fp = fp;
            obj.L = L;
        end
    end
end
```

Appendix J

Class Used to Contain the Measurement Sets for the Physical VSI

```
classdef GPSetSintef < GPSet
    methods
        function obj = GPSetSintef (fp,L)
            obj = obj@GPSet (fp,L);
        end

        function obj = createSet (obj,filename1, filename2, channels)
            % Setting up variables
            vdq = zeros(3,obj.L);
            idq = zeros(3,obj.L);

            ldir = dir(filename1);

            for i = 1:numel(obj.fp);
                temp = ldir(2+i).name;
                tempScope1 = MSO4054Data([filename1,temp], channels, obj.L);
                tempScope2 = DPO4054Data([filename2,temp], channels, obj.L);

                Vab = tempScope1.data (:,1);
                Vbc = tempScope1.data (:,2);
                Vca = -tempScope2.data (:,1);

                Ia = -tempScope2.data (:,2);
                Ib = -tempScope2.data (:,3);
                Ic = -tempScope2.data (:,4);

                Vdc = tempScope1.data (:,3);
                Idc = tempScope1.data (:,4);

                Va = 1/3*(Vab-Vca);
                Vb = 1/3*(Vbc-Vab);
                Vc = 1/3*(Vca-Vbc);

                Vabc = [Va'; Vb'; Vc'];
                Iabc = [Ia'; Ib'; Ic'];


                time = 0:tempScope1.deltaT:(obj.L-1)*tempScope1.deltaT;
                for k1=1:obj.L
                    theta=2*pi*50*time(k1);
```

```
                    P=2/3*[ cos(theta),cos(theta-2*pi/3),cos(theta+2*pi/3)
                            sin(theta),sin(theta-2*pi/3),sin(theta+2*pi/3)
                            1/2 1/2 1/2]; %tansform book
                    vdq(:,k1)=P*Vabc(:,k1); %v_dq complex number
                    idq(:,k1)=P*Iabc(:,k1); %idq_complex number
                end
                obj.vo(1,i,:) = vdq(1,:);
                obj.vo(2,i,:) = vdq(2,:);

                obj.io(1,i,:) = idq(1,:);
                obj.io(2,i,:) = idq(2,:);

                obj.vi(i,:) = Vdc;
                obj.ii(i,:) = Idc;
            end
        end
    end
end
```

Appendix K

Matlab Script to Filter the Measurements and Obtain the Model of the Physical VSI

```
Yddf = zeros(1,numel(fp));
Ydqf = zeros(1,numel(fp));
Yqdf = zeros(1,numel(fp));
Yqqf = zeros(1,numel(fp));

Zif = zeros(1,numel(fp));

GPf = TPModel ();

for i = 1:numel(fp)
    %% Base
    Fs = 1e5;
    d = fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2',fp(i)-10,fp(i)-5,...
        fp(i)+5,fp(i)+10,60,1,60,Fs);
    Hlp = design(d,'butter');
    vod=fft(filter(Hlp,setBase.vo(1,i,:)))/L);
    voq=fft(filter(Hlp,setBase.vo(2,i,:)))/L);

    iod=fft(filter(Hlp,setBase.io(1,i,:)))/L);
    ioq=fft(filter(Hlp,setBase.io(2,i,:)))/L);

    vdc=fft(filter(Hlp,setBase.vi(i,:)))/L);
    idc=fft(filter(Hlp,setBase.ii(i,:)))/L);

    VodBase = vod(fp(i));
    VoqBase = voq(fp(i));

    IodBase = iod(fp(i));
    IoqBase = ioq(fp(i));

    ViBase = vdc(fp(i));
    IiBase = idc(fp(i));

    %% Set 1
    vod=fft(filter(Hlp,setI.vo(1,i,:)))/L);
    voq=fft(filter(Hlp,setI.vo(2,i,:)))/L);

    iod=fft(filter(Hlp,setI.io(1,i,:)))/L);
    ioq=fft(filter(Hlp,setI.io(2,i,:)))/L);
```

```
    vdc=fft(filter(Hlp,setI.vi(i,:))/L);
    idc=fft(filter(Hlp,setI.ii(i,:))/L);

    VodI = vod(fp(i));
    VoqI = voq(fp(i));

    IodI = iod(fp(i));
    IoqI = ioq(fp(i));

    ViI = vdc(fp(i));
    IiI = idc(fp(i));

    %% Set 2
    vod=fft(filter(Hlp,setII.vo(1,i,:))/L);
    voq=fft(filter(Hlp,setII.vo(2,i,:))/L);

    iod=fft(filter(Hlp,setII.io(1,i,:))/L);
    ioq=fft(filter(Hlp,setII.io(2,i,:))/L);

    vdc=fft(filter(Hlp,setII.vi(i,:))/L);
    idc=fft(filter(Hlp,setII.ii(i,:))/L);

    VodII = vod(fp(i));
    VoqII = voq(fp(i));

    IodII = iod(fp(i));
    IoqII = ioq(fp(i));

    ViII = vdc(fp(i));
    IiII = idc(fp(i));

%% Set 3
    vod=fft(filter(Hlp,setIII.vo(1,i,:))/L);
    voq=fft(filter(Hlp,setIII.vo(2,i,:))/L);

    iod=fft(filter(Hlp,setIII.io(1,i,:))/L);
    ioq=fft(filter(Hlp,setIII.io(2,i,:))/L);

    vdc=fft(filter(Hlp,setIII.vi(i,:))/L);
    idc=fft(filter(Hlp,setIII.ii(i,:))/L);

    VodIII = vod(fp(i));
    VoqIII = voq(fp(i));

    IodIII = iod(fp(i));
    IoqIII = ioq(fp(i));

    ViIII = vdc(fp(i));
    IiIII = idc(fp(i));

    MOutput = [ VodI VodII VodIII;
                VoqI VoqII VoqIII;
                IiI  IiII  IiIII];%...
              %-[VodBase VodBase VodBase;
               % VoqBase VoqBase VoqBase;
                %IiBase IiBase IiBase];

    MInput =  [ ViI  ViII  ViIII;
                IodI IodII IodIII;
                IoqI IoqII IoqIII];%...
              %-[ViBase ViBase ViBase;
               % IodBase IodBase IodBase;
                %IoqBase IoqBase IoqBase];


    M = MOutput/MInput;

    GPf.G_dd.func(i) = M(1,1);
    GPf.G_qd.func(i) = M(2,1);
    GPf.Z_dd.func(i) = M(1,2);
    GPf.Z_dq.func(i) = M(1,3);
    GPf.Z_qd.func(i) = M(2,2);
    GPf.Z_qq.func(i) = M(2,3);
```

```
    GPf.Y_dd.func(i) = M(3,1);
    GPf.H_dd.func(i) = M(3,2);
    GPf.H_dq.func(i) = M(3,3);

    Zif(i) = ViIII/IiIII;

    Ym = MInput(2:3,2:3)/MOutput(2:3,2:3);

    Yddf(i) = Ym(1,1);
    Ydqf(i) = Ym(1,2);
    Yqdf(i) = Ym(2,1);
    Yqqf(i) = Ym(2,2);
end
```

Appendix L

Matlab Function to Filter the Transfer Functions Using a Moving Average

```
function GPavg = GPPostProcess (object,window,fp)

    Havg = ones(1,window)/window;

    GPavg = object;

    prop = properties(object);
    for i = 1:numel(prop)
        GPavg.(prop{i}).func = filter(Havg,1,object.(prop{i}).func);
    end
end
```