# Power Electronic Voltage and Frequency Control for Distributed Generation System

LUND INSTITUTE
OF TECHNOLOGY
Lund University

Johan Björnstedt
Magnus Ström

# Abstract

The use of microsources in the power grid often requires assistance of power electronics to convert the produced DC to AC. A big issue is to get the converters operating together with the power grid, without any communication between them.
In this project a control algorithm for the converters is developed and tested in a standalone microgrid containing three converters and a synchronous generator. The controller is based on droop control in order to avoid communication between the converters. The behaviour at the converters DC-side is not taken into consideration, the focus is on voltage- and frequency control at the common AC-side.
Models of converter, generator, grid and load are developed in order to carry out numerical simulations in the software MATLAB/SIMULINK. The control algorithm is then implemented in a digital signal processor, controlling each converter.



**Keywords:** Renewable energy system, Distributed generation system, Standalone system Droop control, Converter control

# Preface

This report is the result of our master thesis project carried out at the department of Industrial Electrical Engineering and Automation (IEA) at Lund Institute of Technology. It is also the final part of our Master of Science in Electrical Engineering degree.

We would like to thank the staff at IEA for their kindness and hospitality and special thanks to all the people who have contributed with their opinion to the subject. Bengt Simonsson and Getachew Darge at IEA have provided us with practical help in the laboratory. Bengts comments have made us laugh a lot of times and Getachew has shown a lot of patience with us, despite we has turned his laboratory up side down.

Most of all we would like to thank our teacher and supervisor Dr Per Karlsson, for inspiring, guiding, assisting and supporting us throughout the project. Like Bengt, Per always had a lot of clever comments for everything. We also would like to express our gratitude to Per for inviting us to become co-authors to the paper "Stability of Voltage and Frequency Control in Distributed Generation on Parallel-Connected Converters Feeding Constant Load" which will be published at the 11th European Conference on Power Electronics and Applications (EPE 2005).

Finally, we would like to thank our families for always supporting us and encouraging our studies through the years.

Lund May 2005

*Magnus Ström, Johan Björnstedt*

# Table of Contents

# 1 Introduction

Today, the presence of small-size generation technologies has changed the thinking away from large power stations and control, toward the notion that there may be advantages if generation is moved closer to the end user. Much of the energy generated today is produced by large-scale, centralized power plants using fossil fuels (coal, oil and gas), hydropower or nuclear power, with energy being transmitted and distributed over long distances to consumers. There are a number of drawbacks with such a system, such as the high level of dependence on imported fuels, environmental impact, transmission losses and the necessity for continuous upgrading and replacement of transmission and distribution facilities. In contrast, in a power system composed of distributed energy resources, much smaller amounts of energy are produced by numerous small microsources. These units can be standalone or integrated into the power grid. Technologies, for an example wind turbines, fuel cells, photovoltaic panels and storage devices are small-scale technologies that can be used on a site close to the end-user. These technologies have in most cases capacities in the 1 kW to 15 MW range.

During the last decade, distributed generation systems are rapidly gaining popularity. Distributed generation can provide standby generation (emergency power), peak shaving capability and cogeneration. It also improves the efficiency of the power grid by having multiple microsources added to the system. Distributed generation systems can also help to improve power quality and power supply flexibility, maintain system stability and reduce the transmission and distribution cost. In Sweden, approximately 8-9% of the produced hydropower is lost due to transmission and distribution losses. Annually, this power loss almost corresponds to one nuclear reactor, i.e. 500 MW. These power losses in the power system impose a large cost on society, both financial and environmental. By putting the microsources closer to the end-user a reduction in the transmission and distribution cost can be achieved.

The use of distributed generation in the power grid is not a problem-free process. Interfacing microsources with the power systems involves implications on the structure and operating procedures of both the units and the system in which they are imbedded. The main concerns focus on operating procedures, control systems, islanding, and safety issues.

## 1.1 Technologies

Due to the fact that all microsources operate at varying frequency or even delivers DC, these sources require power electronic to interface to the power grid and its loads. In almost all cases there is a DC source, which must be converted to AC at the required frequency, magnitude and phase angle. The power electronic interface introduces new control issues and new possibilities. One large class is related to the traditional cogeneration problem. A system with clusters of micro generators could be designed to operate in both island mode and as a satellite system connected to the power grid. In such systems load swings become a major issue. A load step will affect the system. In the case of a power electronics based source, the DC-link voltage will decrease in response to the increased load. For example, micro turbines and fuel cells have controllers that use the decrease in DC voltage to provide an increased energy input.

The control of converters utilized to supply power to an AC system in a distributed environment should be based on information available locally at the converter. In a system with many microsources communication of control signals between systems is impractical. This means that the converter control should be based on terminal quantities in order to

respond effectively to system changes without requiring data from other sources or loads at different locations. To prevent overloading of the microsource, it is important to ensure that the converters share load in proportion to their nominal power, without communication.

## 1.2 Objective of the Thesis

In this project, three-phase 2-level voltage source converters are connected front-to-front in order to model a small standalone microgrid. The behaviour at the converters DC-side is not considered, this project is only focusing on voltage and frequency control at the common AC-side.

The main objective of this thesis is to develop a control algorithm for the three-phase converters based perform terminal quantities. Models of the converter, grid and load are developed in order to do numerical simulations in the software MATLAB/SIMULINK. Then the control algorithm is implemented in a DSP, controlling the converter.

# 2 Three Phase Voltage Source Converter

A voltage source converter, VSC, is a self commutated converter that converts DC voltage into AC current or vice versa. The power can flow through the VSC in both directions and the converter can be considered as a controllable current source. The most common VSC topology in high power applications are 2- or 3-level converters. A basic principal of the 2-level converter used in this project is shown in Figure 2.1 [1].

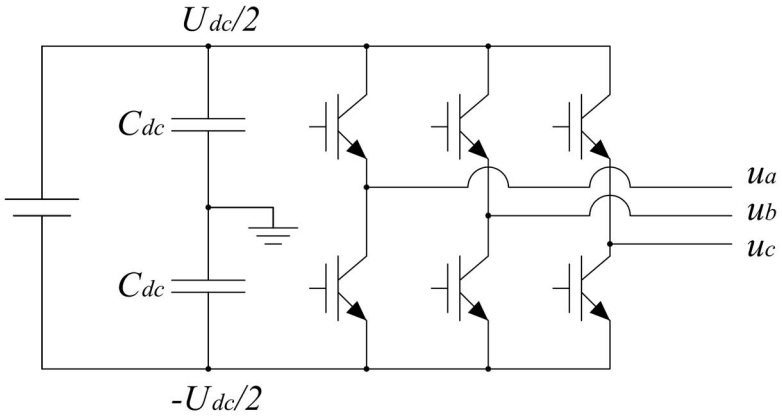


***Figure 2.1*** *A schematic principal of the 2-level converter used in this project.*

## 2.1 Semiconductors

A typical configuration of a VSC is a six pulse self commutated converter containing six power semiconductors with anti-parallel connected diodes. These power semiconductor devices can, for example be

- Gate Turn Off (GTO) Thyristors
- Metal Oxid Semiconductor Field Effect Transistors, MOSFET
- Insulated Gate Bipolar Transistors, IGBT

Typical switching frequencies for a VSC are between 2-15 kHz. The MOSFET is the preferred semiconductor for low power/low voltage applications since it has low conduction losses for low voltages, and low switching losses. The MOSFET also has a parasitic anti-parallel diode, that may be used as a built in free-wheeling diode, if its reverse recovery time is acceptable. However, the on-state resistance of a normal MOSFET has strong dependency on the blocking voltage. This makes the IGBT the preferred choice for applications in the medium/high voltage range. Both the MOSFET and the IGBT are voltage controlled which allows simple gate drive circuits. The GTO has high switching losses and requires a more complicated driver since it is current controlled. The GTO is mainly used for high power applications due to its low forward voltage drop and its high blocking voltage. This thesis is not focusing on any special semiconductor switch, although IGBT:s are used during the experiments.

## 2.2 Capacitive DC link

The DC-link capacitors, $C_{dc}$, main purpose is to store energy needed for smoothing action, which means that the voltage across it remains almost constant. In this project it is assumed that the DC-link voltage, $U_{dc}$, is constant but in reality the DC-link voltage is often load dependent.
If the DC-link is feed from a diode rectifier and the load is small, the average DC voltage is close to the peak value of the line-to-line voltage and has a ripple frequency equal to six

times the fundamental line frequency. This is due to the fact that the capacitor charges and discharges during the voltage variation. The DC-link voltage ripple will be higher when a smaller capacitor is used [5]. The pulse pattern of the output voltage of the converter are controlled by Pulse Width Modulation, PWM, which is described in the section Carrier Based Modulation.

## 2.3 Output filter

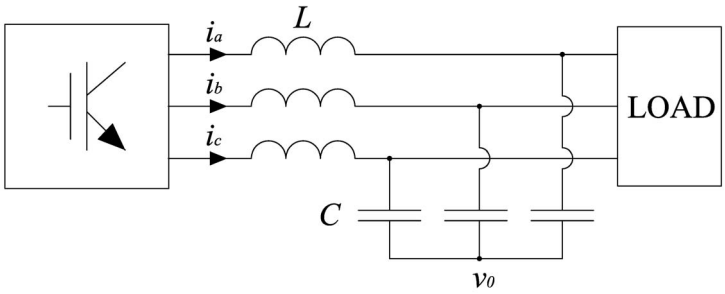


***Figure 2.2*** *Converter output filter.*

The DC-link is capacitive and therefore the output filter has to be inductive, which is also the reason why the converter acts a current source. To smooth the rough output voltage, star connected shunt capacitors are placed between the inductors and the grid. To get the same current ripple, in per unit, for all converters the inductors are chosen so that the inductance is proportional to the inverse of the nominal power of the converter. Then the capacitors of the filter are chosen so that the filters has a resonance frequency that is ten times the grid frequency, i.e. 500 Hz, this is done to avoid oscillations between the filter and the grid. For the same reason the sampling frequency of the converter should be at least 10 kHz to provide a Nyquist frequency of 5 kHz.

# 3 Converter Control

The converter is vector current controlled and the current references are determined from the active and reactive power references resulting from voltage and frequency droop control. Figure 3.1 shows the principle schematic of the control system.

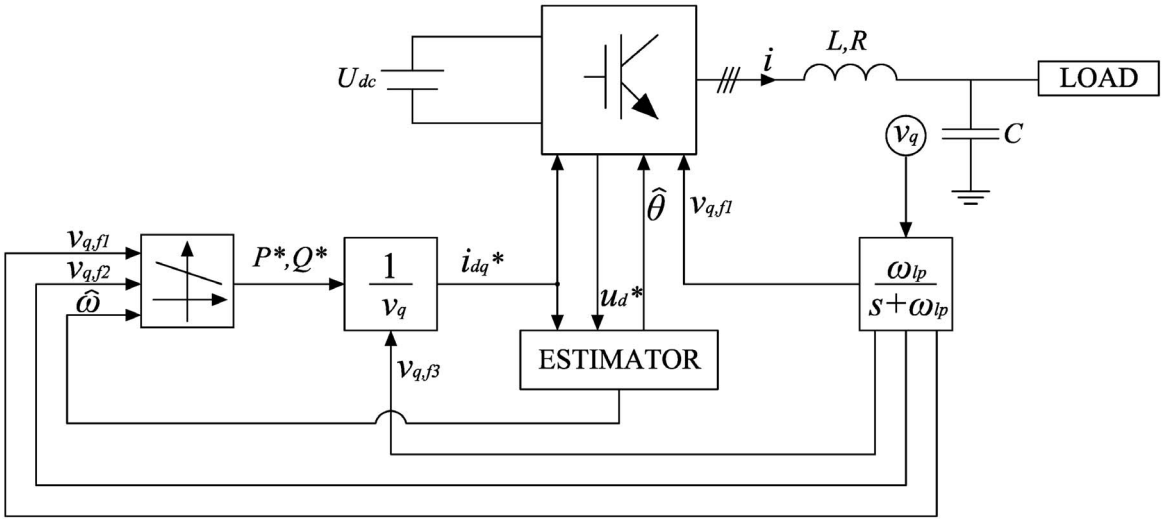


***Figure 3.1*** *Voltage source converter with control system.*

The converter is intended to control the voltage and frequency based only on instantaneous measurement at the converter terminal. When the converter is connected to a strong power grid, the grid is giving the frequency and voltage, but when several converters are connected together in a standalone system the measurements are based only on quantities produced by the converters themselves. The voltage is measured over the output filter capacitor.

## 3.1 Droop Control

In order to obtain correct load sharing between the converters, different methods can be used, for example master/slave- or droop control. Droop control has been chosen based on the criterion that communications of control signals between the converters should be necessary. Instead the converter should only use those quantities that can be measured locally at the converters. This is essential for the operation of large systems, where distance between converters makes high speed communication impractical.

To avoid overloading the converters, it is important to ensure that the converters share load proportional to their nominal power. This is achieved in conventional power systems, with multiple generators, by introducing a droop function based on the frequency of each generator. This permits each generator to adapt to changes in total load in a manner determined by its frequency droop characteristics. Similarly, a droop in the voltage with reactive power is used to ensure correct reactive load sharing. The same philosophy could be used to ensure correct distribution of total power between parallel connected converters in a standalone system.

### Converter Droop Characteristic

The converter is designed to operate with current references. To determine how the converter should be controlled the converter output filter is studied.

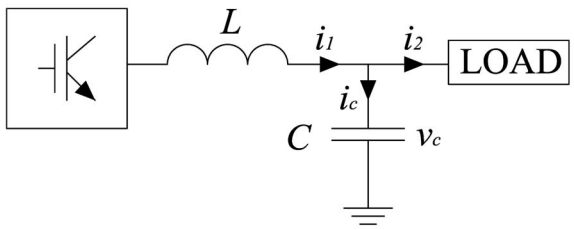


***Figure 3.2*** *Output filter.*

The current through the capacitor, in the $\alpha\beta$-frame, can be expressed as

$$i_c = C\frac{dv_c}{dt} \tag{3.1}$$

the same expression in the $dq$-frame is

$$\vec{i}_c = \omega C\vec{v}_c + C\frac{d\vec{v}_c}{dt} = \vec{i}_1 - \vec{i}_2 \tag{3.2}$$

$$i_{cd} + ji_{cq} = j\omega Cv_{cd} - \omega Cv_{cq} + C\frac{d\left(v_{cd} + jv_{cq}\right)}{dt} = \left(i_{1d} - i_{2d}\right) + j\left(i_{1q} - i_{2q}\right) \tag{3.3}$$

the real- and imaginary part of Equation (3.2) are

$$i_{cd} = -\omega Cv_{cq} + C\frac{dv_{cd}}{dt} = i_{1d} - i_{2d} \tag{3.4}$$

$$i_{cq} = \omega Cv_{cd} + C\frac{dv_{cq}}{dt} = i_{1q} - i_{2q} \tag{3.5}$$

By assuming that the $q$-axis is aligned with the grid voltage vector in the $dq$ synchronous reference frame, the grid voltage only has a $q$ component, see Appendix A. When $i_{2d}$ increases without increasing $i_{1d}$, and given by definition that $v_{cd}$ is zero, Equation (3.4) gives that $\omega$ has to increase. In the same manner Equation (3.5) for increasing $i_{2q}$ gives that the voltage decreases. From these investigations two droop controllers can be derived as shown in Figure 3.6.

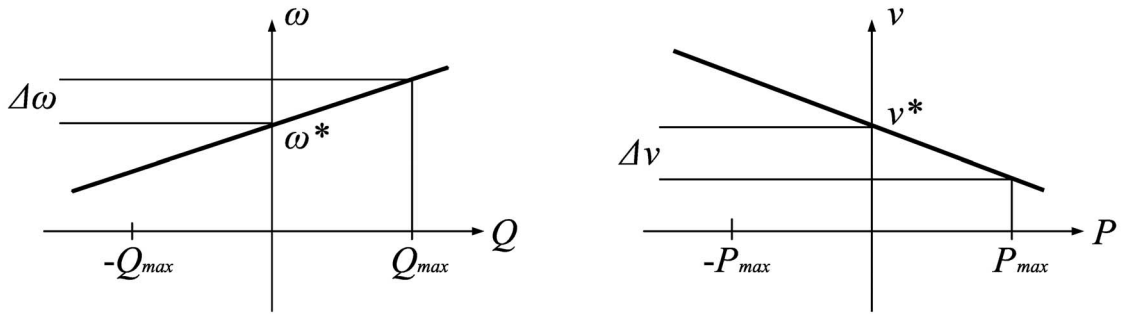


***Figure 3.3*** *Converter droop characteristics.*

The converter droop characteristics are then described by

$$Q^* = -K_\omega\left(\omega^* - \omega\right) \tag{3.6}$$

$$P^* = K_v\left(v^* - v\right) \tag{3.7}$$

The droop functions derived in Figure 3.3 is used to control the converter if it is connected directly to a load or in a grid with only converters. However, when converters are connected to a strong grid and are intended to share the load in proportion to their nominal power, this control algorithm is not appropriate. Therefore the grid behaviour is investigated to develop a control method for such a case.

### Grid Characteristic

For a system operating in sinusoidal-steady-state, the apparent power, *S*, flowing into a line, see Figure 3.4a, can be described as, [2]

$$\bar{S} = P + jQ = \bar{U}\cdot\bar{I}^* = \bar{U}\left(\frac{\bar{U}-\bar{V}}{\bar{Z}}\right)^* = U\left(\frac{U - Ve^{j\delta}}{Ze^{-j\theta}}\right) = \frac{U^2}{Z}e^{j\theta} - \frac{UV}{Z}e^{j(\theta+\delta)} \tag{3.8}$$

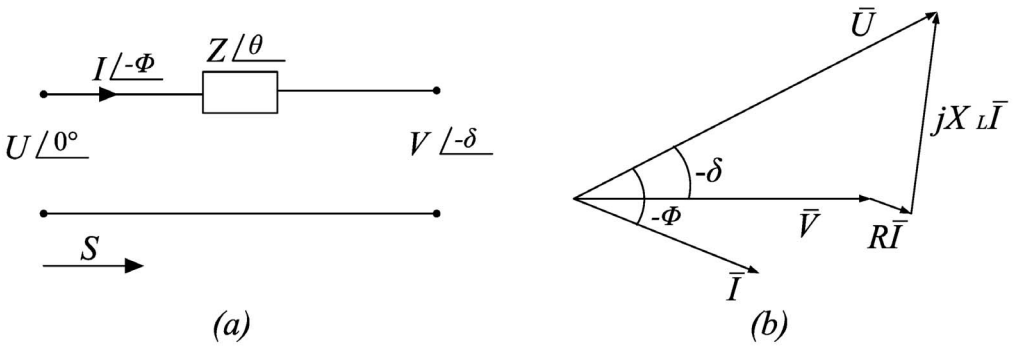


***Figure 3.4*** *a) Power flowing through a line, b) phase diagram.*

Thus, the active and reactive power flowing into the line is

$$P = \frac{U^2}{Z}\cos(\theta) - \frac{UV}{Z}\cos(\theta+\delta) \tag{3.9}$$

and

$$Q = \frac{U}{Z}\sin(\theta) - \frac{UV}{Z}\sin(\theta+\delta) \tag{3.10}$$

With the impedance $Ze^{j\theta} = R + jX$ , the equations above can be rewritten as,

$$P = \frac{U}{R^2+X^2}\left[R(U - V\cos(\delta)) + XV\sin(\delta)\right] \tag{3.11}$$

and

$$Q = \frac{U}{R^2+X^2}\left[-RV\sin(\delta) + X(U - V\cos(\delta))\right] \tag{3.12}$$

which leads to

$$V\sin(\delta) = \frac{XP - RQ}{U} \tag{3.13}$$

$$U - V\cos(\delta) = \frac{RP + XQ}{U} \tag{3.14}$$

For overhead lines the resistance, $R$, can be neglected because $X>>R$. Also if the power angle is small, then $sin(\delta) \approx \delta$ and $cos(\delta) \approx 1$. From these approximations Equation (3.13) and (3.14) are rewritten

$$\delta \cong \frac{XP}{UV} \tag{3.15}$$

$$U - V \cong \frac{XQ}{U} \tag{3.16}$$

These equations show that the power angle depends primarily on active power, $P$, while the voltage difference depends primarily on reactive power, $Q$. Since the power angle depends on the frequency, the active power can be controlled by controlling the frequency.

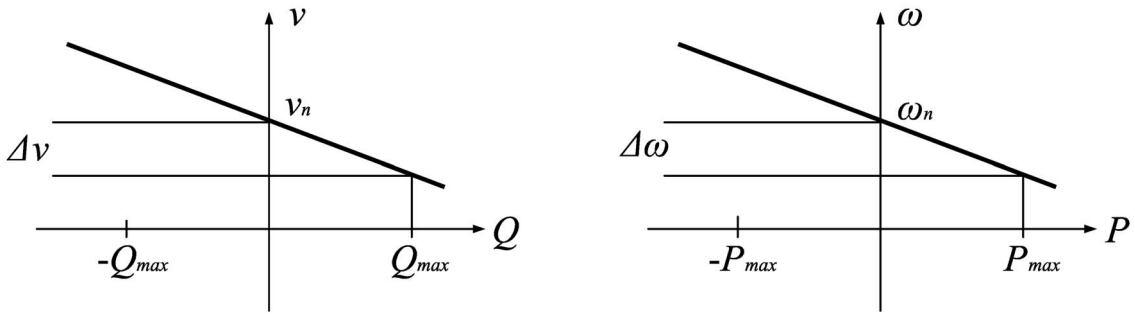


***Figure 3.5*** *Grid droop characteristics.*

Thus, by adjusting $P$ and $Q$ independently, frequency and voltage amplitude of the power grid is changed. These conclusions form the basis for droop control with active and reactive power

$$Q^* = K_v(v_n - v) \tag{3.17}$$

$$P^* = K_\omega(\omega_n - \omega) \tag{3.18}$$

where $K$ is the gain, $\omega_n$ and $v_n$ are the nominal frequency and the nominal grid voltage, respectively, $P^*$ and $Q^*$ are the references for active and reactive power respective of the converter.

These droop functions are commonly used to control synchronous generators in the power system and involves load sharing based on locally measured voltage and frequency. To make the converter behave as a synchronous generator the droop functions in Figure 3.3 and Figure 3.5 are combined.

## Converter Control

In order to make the converter behave as a synchronous generator a control method containing two steps is developed. Such a system is shown in Figure 3.6, where all four droop functions, derived above, are combined. Then the slow grid droop functions are giving the references for the fast converter droop functions. This means that the fast droop functions in Figure 3.3 are slowly shifted vertically by the droop functions in Figure 3.5 and the desired behaviour is achieved.

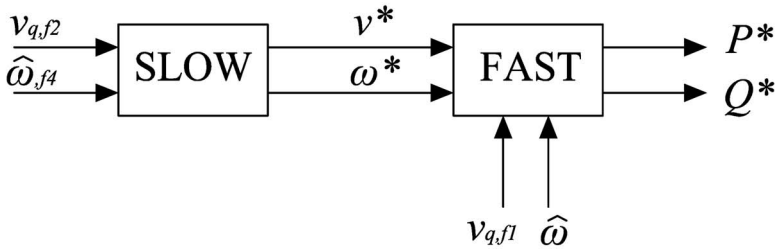


***Figure 3.6*** *The complete droop controller.*

To derive the equations for the entire system Equation (3.17) is combined with (3.6) and Equation (3.18) with (3.7)

$$Q^* = K_v\left(v_n - v\right) = -K_\omega\left(\omega^* - \omega\right) \tag{3.19}$$

$$P^* = K_\omega\left(\omega_n - \omega\right) = K_v\left(v^* - v\right) \tag{3.20}$$

the references for the fast droops are then calculated as

$$v^* = v_{q,f2} + \frac{K_\omega}{K_v}\left(\omega_n - \omega_{f4}\right) \tag{3.21}$$

$$\omega^* = \omega_{f4} - \frac{K_v}{K_\omega}\left(v_{qn} - v_{q,f2}\right) \tag{3.22}$$

where the indexes *f* indicates different filter bandwidths. The measured voltage is very rough and is therefore filtered with first order LP-filters with low bandwidths. The different cut-off frequencies are chosen to achieve the behaviour with fast and slow droops. When the references, $\omega^*$ and $v^*$, are calculated, $P^*$ and $Q^*$ are obtained according to the fast droop functions

$$Q^* = -K_\omega\left(\omega^* - \omega\right) \tag{3.23}$$

$$P^* = K_v\left(v_q^* - v_{q,f1}\right) \tag{3.24}$$

In order to obtain correct load sharing, according to the converters nominal power $S_n$, the gain parameters are chosen as

$$K_\omega = \frac{S_n}{\omega_n \cdot \Delta_\omega} \tag{3.25}$$

$$K_v = \frac{S_n}{v_n \cdot \Delta_v} \tag{3.26}$$

where $\Delta_\omega$ and $\Delta_v$ are the maximum deviation in frequency and voltage, respectively.

The converter is based on a current controller so the determined active- and reactive power references need to be recalculated to current references. Thus,

$$i_d^* = \frac{Q^*}{v_{q,f3}} \tag{3.27}$$

$$i_q^* = \frac{P^*}{v_{q,f3}} \tag{3.28}$$

## 3.2 Frequency and Phase Estimation

To make the droop control feasible, the frequency in the grid is estimated. An error signal can be obtained in the following form

$$\varepsilon = K \sin\left(m\left(\theta - \hat{\theta}\right)\right) \tag{3.29}$$

where $\theta$ is the actual voltage vector position, $\hat{\theta}$ is the estimated position, $K$ is a gain parameter and $m$ describes the rotation polarity of the vector. In cases where the rotation polarity is unknown $m = 2$ otherwise $m = 1$.

The signal $\varepsilon$ can be used to drive the frequency and angle estimation to their true values, using the following algorithm

$$\dot{\hat{\omega}} = \gamma_1 \varepsilon \tag{3.30}$$

$$\dot{\hat{\theta}} = \hat{\omega} + \gamma_2 \varepsilon \tag{3.31}$$

the gain parameters are $\gamma_1 = \rho_\omega^2 / mK$ and $\gamma_2 = 2\rho_\omega / mK$. The estimated angle is obtained as the integral of the estimated frequency corrected by the term $\gamma_2\varepsilon$. Under the condition that the error angle is small, i.e. $\theta \approx \hat{\theta}$, Equation (3.30) and (3.31) can be linearized as

$$\dot{\hat{\omega}} = \gamma_1'\left(\theta - \hat{\theta}\right) \tag{3.32}$$

$$\dot{\hat{\theta}} = \hat{\omega} + \gamma_2'\left(\theta - \hat{\theta}\right) \tag{3.33}$$

where $\gamma_1' = \gamma_1 mK$ and $\gamma_2' = \gamma_2 mK$. In order to obtain robustness and avoid oscillations both poles are placed at $s = -\rho$ where $\rho$ is a positive constant [3]. For complete small signal analysis and pole placement, see [4].

### Discrete Time Implementation

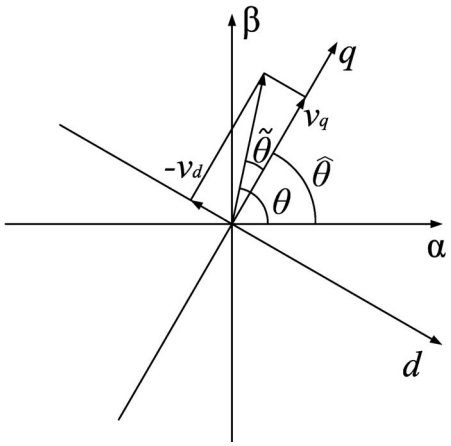


***Figure 3.7*** *Error angle in αβ-frame and dq-frame.*

From Figure 3.7 it is seen that the position error angle can be expressed as

$$\tilde{\theta} = \left(\theta - \hat{\theta}\right) \approx \frac{-v_d}{v_q} = \frac{-\left(u_d^* - Ri_d^* + \hat{\omega}Li_q^*\right)}{v_q} \tag{3.34}$$

Instead of using measured voltage and current components, their reference values are used in order to reduce noise. This does not introduce any errors, provided that the current controller gives no static error.
Finally, the frequency and angle can be described from Equations (3.32) and (3.33) as

$$\dot{\hat{\omega}} = \frac{\rho_\omega^2}{v_q}\left(-u_d^* + Ri_d^* - \hat{\omega}Li_q^*\right) \tag{3.35}$$

$$\dot{\hat{\theta}} = \hat{\omega} + \frac{2\rho_\omega}{v_q}\left(-u_d^* + Ri_d^* - \hat{\omega}Li_q^*\right) \tag{3.36}$$

Transformation from the continuous time domain to the discrete time domain is performed by applying forward Euler method on Equation (3.35) and (3.36)

$$\hat{\omega}_{k+1} = \hat{\omega}_k + \frac{\rho_\omega^2 T_s}{v_q}\left(-u_{d,k}^* + Ri_{d,k}^* - \hat{\omega}_k Li_{q,k}^*\right) \tag{3.37}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \hat{\omega}_k T_s + \frac{2\rho_\omega T_s}{v_q}\left(-u_{d,k}^* + Ri_{d,k}^* - \hat{\omega}_k Li_{q,k}^*\right) \tag{3.38}$$

where $T_s$ is the sampling period and $k$ is the sample number (at time $t = kT_s$) and $\rho$ is the estimator bandwidth. By using Equation (3.37) and (3.38) the frequency and angle are estimated. The frequency is then used in the droop controller to calculate references and the angle is used in the current controller to align the voltage vectors of the converter with the vectors of the grid.

## 3.3 Filter Design

To suppress noise and avoid transients in the measured voltage signal, it has to be low-pass filtered. As can be seen from Figure 3.1 the filtered voltage signal is used for calculations in both the fast and the slow droops for converting power reference to current reference and in the converter. In order to make the system stable different cut-off frequencies has ben calculated from the small signal models, see [4].

Consider the Laplace transfer function of a first order low-pass filter, with the cut-off frequency $\omega_0$

$$H(s) = \frac{Y(s)}{X(s)} = \frac{\omega_0}{s + \omega_0} \tag{3.39}$$

which relates the filtered signal *Y(s)* to the measured *X(s)*. This has the following time domain equivalent

$$\frac{dy(t)}{dt} + \omega_0 y(t) = \omega_0 x(t) \tag{3.40}$$

Now, the differential equation can be descretized using explicit Euler method

$$\frac{y_k - y_{k-1}}{T_s} + \omega_0 y_k = \omega_0 x_k \tag{3.41}$$

where $T_s$ is the sampling interval. Simplification and rearrangement gives

$$y_k = (x_k - y_k)\omega_0 T_s + y_{k-1} \tag{3.42}$$

Graphically this equation is illustrated as

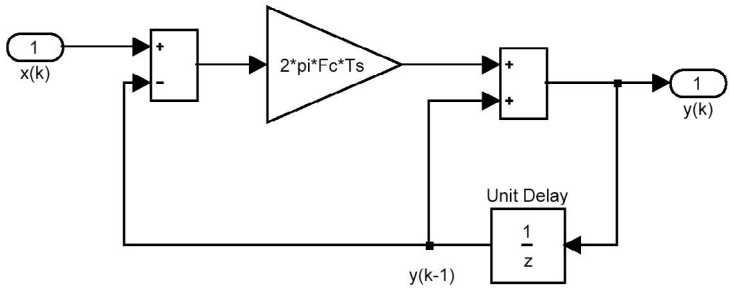


***Figure 3.8*** *Block diagram of the proposed first order low-pass filter.*

## 3.4 Vector Current Control

In this section a current controller algorithm for the converter will be derived. The logical switching states and modulation problems will also be studied.

### System Equations

The simplified circuit of a grid connected converter is displayed in Figure 3.9. The grid and the converter are modelled as three-phase voltage sources with inductors, one in each phase, connected in series between them. The converter phase voltages are denoted as $u_a(t)$, $u_b(t)$ and $u_c(t)$. The phase voltages of the grid are denoted $v_a(t)$, $v_b(t)$ and $v_c(t)$. The phase current

through the inductance are denoted as $i_a(t)$, $i_b(t)$ and $i_c(t)$. The equivalent inductance and resistance of the inductor are denoted as $L$ and $R$, respectively.

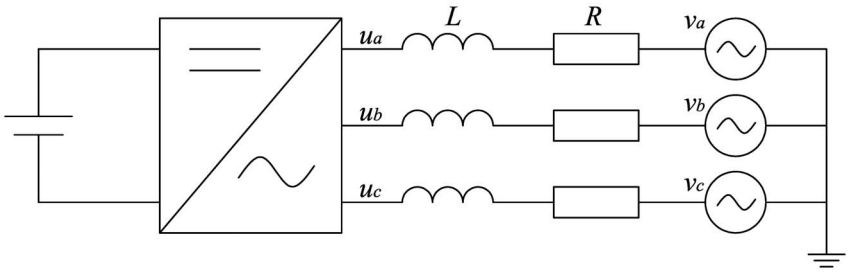


***Figure 3.9*** *Simplified circuit of a grid connected converter.*

With Kirchhoffs voltage law applied on the circuit in Figure 3.9, the three-phase system differential equations are

$$u_a(t) = v_a(t) + i_a(t)R + L\frac{di_a(t)}{dt} \tag{3.43}$$

$$u_b(t) = v_b(t) + i_b(t)R + L\frac{di_b(t)}{dt} \tag{3.44}$$

$$u_c(t) = v_c(t) + i_c(t)R + L\frac{di_c(t)}{dt} \tag{3.45}$$

The grid voltages equal

$$v_a(t) = \sqrt{\frac{2}{3}}V\cos(\omega t) \tag{3.46}$$

$$v_b(t) = \sqrt{\frac{2}{3}}V\cos\left(\omega t - \frac{2\pi}{3}\right) \tag{3.47}$$

$$v_c(t) = \sqrt{\frac{2}{3}}V\cos\left(\omega t - \frac{4\pi}{3}\right) \tag{3.48}$$

where $V$ is the phase to phase rms voltage, and $\omega$ is the grid angular frequency.

In the $\alpha\beta$-frame, the three-phase system in Equations (3.43) to (3.45) becomes

$$u_\alpha(t) = v_\alpha(t) + i_\alpha(t)R + L\frac{di_\alpha(t)}{dt} \tag{3.49}$$

$$u_\beta(t) = v_\beta(t) + i_\beta(t)R + L\frac{di_\beta(t)}{dt} \tag{3.50}$$

which also can be written in vector notation

$$\vec{u}^{(\alpha\beta)}(t) = L\frac{d\vec{i}^{(\alpha\beta)}(t)}{dt} + \vec{i}^{(\alpha\beta)}(t)R + \vec{v}^{(\alpha\beta)}(t) \tag{3.51}$$

Equation (3.51) is transformed into the *dq*-frame by using the *αβ* to *dq* transformation giving

$$\vec{u}^{(dq)}(t) = L\frac{d\vec{i}^{(dq)}(t)}{dt} + j\omega L\vec{i}^{(dq)}(t) + \vec{i}^{(dq)}(t)R + \vec{v}^{(dq)}(t) \tag{3.52}$$

If this vector expression is divided into its components, i.e. real and imaginary parts, the resulting expressions become

$$\begin{cases} u_d(t) = L\dfrac{di_d(t)}{dt} - \omega L i_q(t) + i_d(t)R + v_d(t) \\ u_q(t) = L\dfrac{di_q(t)}{dt} + \omega L i_d(t) + i_q(t)R + v_q(t) \end{cases} \tag{3.53}$$

Note the cross-coupling term between $u_d$ and $u_q$.

### Proportional Integral Controller

The first step is to use backward Euler approximation for the current derivatives in Equation (3.53). Thus,

$$\begin{cases} \dfrac{di_d}{dt} \approx \dfrac{i_{d,k} - i_{d,k-1}}{T_s} \\ \dfrac{di_q}{dt} \approx \dfrac{i_{q,k} - i_{q,k-1}}{T_s} \end{cases} \tag{3.54}$$

Furthermore, linear current variation during one sample period [*k-1,k*] is assumed in P-controller, yielding that the current can be approximated with their average levels

$$\begin{cases} i_{d,k} \approx \dfrac{i_{d,k} + i_{d,k-1}}{2} \\ i_{q,k} \approx \dfrac{i_{q,k} + i_{q,k-1}}{2} \end{cases} \tag{3.55}$$

Since the sampling frequency usually is several times faster than the dynamics of the grid, the voltage can be assumed to be constant within one sample period, i.e.

$$\begin{cases} v_{d,k} \approx v_{d,k-1} \\ v_{q,k} \approx v_{q,k-1} \end{cases} \tag{3.56}$$

These assumptions lead to the following results

$$\begin{cases} u_{d,k} = \dfrac{L}{T_s}\left(i_{d,k} - i_{d,k-1}\right) - \omega L\dfrac{i_{q,k} + i_{q,k-1}}{2} + R\dfrac{i_{d,k} + i_{d,k-1}}{2} + v_{d,k-1} \\ u_{q,k} = \dfrac{L}{T_s}\left(i_{q,k} - i_{q,k-1}\right) + \omega L\dfrac{i_{d,k} + i_{d,k-1}}{2} + R\dfrac{i_{q,k} + i_{q,k-1}}{2} + v_{q,k-1} \end{cases} \tag{3.57}$$

In order to achieve high dynamic performance, dead-beat gain is used in the controller. With dead-beat gain the entire current error is eliminated in one sampling interval, which means

that the current at the sample instant $k$ equal the current reference at the sample instant $k$-$1$. Thus

$$\begin{cases} i_{d,k} = i^*_{d,k-1} \\ i_{q,k} = i^*_{q,k-1} \end{cases} \tag{3.58}$$

which results in

$$\begin{cases} u_{d,k+1} = L\dfrac{i^*_{d,k} - i_{d,k}}{T_s} - \omega L\dfrac{i^*_{q,k} + i_{q,k}}{2} + R\dfrac{i^*_{d,k} + i_{d,k}}{2} + v_{d,k} \\ u_{q,k+1} = L\dfrac{i^*_{q,k} - i_{q,k}}{T_s} + \omega L\dfrac{i^*_{d,k} + i_{d,k}}{2} + R\dfrac{i^*_{d,k} + i_{q,k}}{2} + v_{q,k} \end{cases} \tag{3.59}$$

Simplifying and rearranging the last equations give a P-controller according to

$$\begin{cases} u^*_{d,k+1} = \left(\dfrac{L}{T_s} + \dfrac{R}{2}\right)\left(i^*_{d,k} - i_{d,k}\right) + Ri_{d,k} - \omega L\dfrac{i^*_{q,k} + i_{q,k}}{2} + v_{d,k} \\ u^*_{q,k+1} = \left(\dfrac{L}{T_s} + \dfrac{R}{2}\right)\left(i^*_{q,k} - i_{q,k}\right) + Ri_{q,k} + \omega L\dfrac{i^*_{d,k} + i_{d,k}}{2} + v_{q,k} \end{cases} \tag{3.60}$$

The resistive voltage drop term, corresponds to a steady-state error, can be interpreted as an integral part assuming that the current equals the sum of all the previous current errors, i.e.

$$\begin{cases} i_{d,k} = \displaystyle\sum_{n=0}^{k-1}\left(i^*_d(n) - i_d(n)\right) \\ i_{q,k} = \displaystyle\sum_{n=0}^{k-1}\left(i^*_q(n) - i_q(n)\right) \end{cases} \tag{3.61}$$

This results in a PI-controller

$$\begin{cases} u^*_{d,k+1} = K_p\left(\left(i^*_{d,k} - i_{d,k}\right) + \dfrac{1}{T_i}\displaystyle\sum_{n=0}^{k-1}\left(i^*_d(n) - i_d(n)\right)\right) - K_c\dfrac{i^*_{q,k} + i_{q,k}}{2} + v_{d,k} \\ u^*_{q,k+1} = K_p\left(\left(i^*_{q,k} - i_{q,k}\right) + \dfrac{1}{T_i}\displaystyle\sum_{n=0}^{k-1}\left(i^*_q(n) - i_q(n)\right)\right) + K_c\dfrac{i^*_{d,k} + i_{d,k}}{2} + v_{q,k} \end{cases} \tag{3.62}$$

where

$$\begin{aligned} K_p &= \left(\frac{L}{T_s} + \frac{R}{2}\right) \\ T_i &= R\Big/\left(\frac{L}{T_s} + \frac{R}{2}\right) = 1\Big/\left(\frac{L}{RT_s} + \frac{1}{2}\right) \\ K_c &= \omega L \end{aligned} \tag{3.63}$$

The reference voltage is then transformed from *dq*-frame to *αβ*-frame. Then the *αβ* reference voltage is transformed to three-phase voltage references.

### Anti Windup

If a control error *e(t)* remains for a longer time it is likely that the integral part becomes too large and introduces windup and saturates the control signal. Mathematically, the saturation can be seen as a nonlinear reduction in gain

$$\vec{u} = \begin{cases} u_{\max} & u \geq u_{\max} \\ u & u_{\min} \leq u \leq u_{\max} \\ u_{\min} & u \leq u_{\min} \end{cases} \tag{3.64}$$

In such a case the integral part of the controller must be limited. This is done by giving the integral part the value of the output control signal subtracted with the proportional part. It can best be described by the following code,

```
I_temp = K*(Ts/Ti)*(i_ref - i) + I;
u_ref = K*(i_ref - i) + I;

if(u_ref > u_max)
{
  u_ref = u_max;
  I = u_max - K*(i_ref - i);
}
else if(u_ref < u_min)
{
  u_ref = u_min;
  I = u_min - K*(i_ref - i);
}
else
{
  I = I_temp;
}
```

### Carrier Based Modulation

By comparing a triangular carrier wave with the three voltage references $u_a^*$, $u_b^*$ and $u_c^*$, the logic signals defining the switching states of the power transistors are generated. The references are sine waves and this method is called sinus modulation. The limit is reached when the reference signal and the carrier wave are equal, called $m$=1 in power electronics where $m$ is the modulation index. For the 2-level converter used in this project, it can be shown that there are eight possible combinations for the switching states, six active and two zero vectors, as shown in Figure 3.10 [1],[5]. The six active voltage vectors are $\sqrt{2/3}\,U_{dc}$ long and divide the frame into six sectors. The tips of the active voltage vectors form a hexagon. The converter can only provide voltage vectors that are within this hexagon. This gives that the maximum length of the voltage reference vector $u_{\max}^*$ is $1/\sqrt{2}\,U_{dc}$, see Figure 3.10. From the states it is clear that only one state, i.e. one switch, is changed at a time. This is advantage for the switches since it gives a low switching frequency.

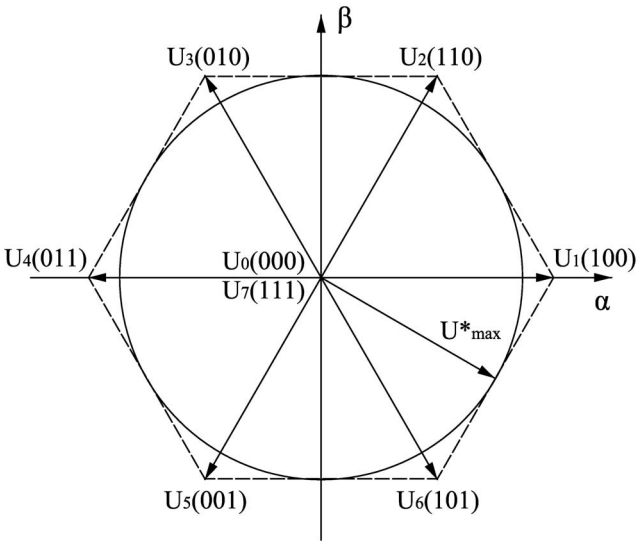


***Figure 3.10*** *Voltage vectors for the three phase 2-level VSC.*

The weak aspect of sinus modulation is that the maximum DC-link voltage is not utilized in an optimal way. Consider the three phases where one phase is at its maximum and in the same moment the other two phases are at their half negative $U_{dc}$, see Figure 3.11. The maximum phase potential will then be $U_{dc}/2$, and thus also the maximum phase voltage of the load. This means that the maximum phase-to-phase voltage is less than $U_{dc}$, which indicates that sinusoidal modulation does not take full advantage of the DC-link voltage. Since there is no common ground, there are two independent references in the $\alpha\beta$-frame but there are three phase voltage references. The third degree of freedom is used to control the potential $v_0$ to optimize the modulation. The potential can only be influenced with a zero sequence signal, thus an arbitrary signal $u_z^*$ can be subtracted from the phase potential references. The adjusted references becomes

$$\begin{cases} u_a^* = u_a^* - u_z^* \\ u_b^* = u_b^* - u_z^* \\ u_c^* = u_c^* - u_z^* \end{cases} \tag{3.65}$$

To maximize the use of the DC-link voltage the signal $u_z^*$ are used. If $u_z^*$ is selected according to Equation (3.66), the phase references will be symmetric, i.e. the most positive and the most negative of the three phase references will have the same instantaneous level.

$$u_z^* = \frac{\max\left(u_a^*, u_b^*, u_c^*\right) + \min\left(u_a^*, u_b^*, u_c^*\right)}{2} \tag{3.66}$$

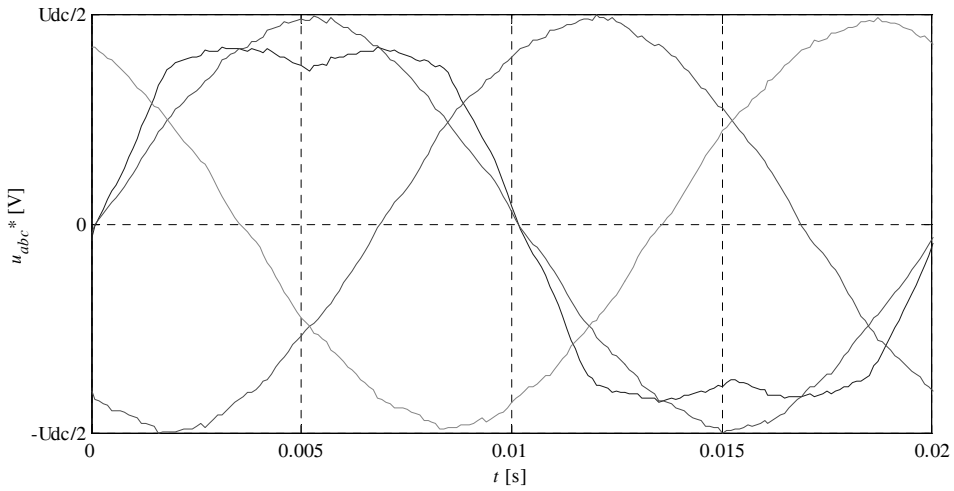


***Figure 3.11*** *Sinusoidal modulation with modified sine.*

The phase voltages and the phase potentials are no longer equal since the phase potentials contain the symmetric signal $u_z^*$. The phase voltages of the load are still sinusoidal, since the phase-to-phase voltage references are sinusoidal, but the phase potential references are not.

Figure 3.12 shows the generation of the signals when using triangular PWM. The voltage references vary slowly compared to the triangular carrier wave if the pulse ratio between the switching frequency and the fundamental voltage references is sufficiently high. The high pulse ratio results in that the voltage harmonics that the PWM converter produces only appear near multiples of the carrier frequency i.e. the switching frequency. The LC-filter is absorbing most of the PWM switching harmonics and prevent the harmonics from getting to the utility grid or the load.

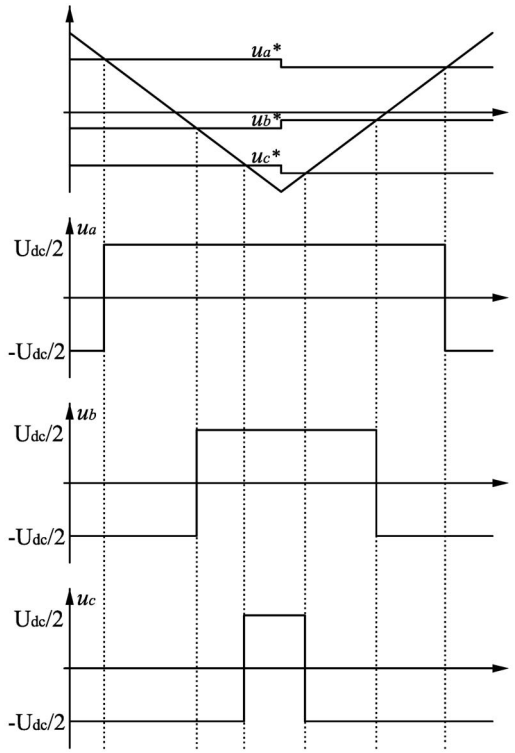


***Figure 3.12*** *Detailed description of the control signal generation.*

The PWM duty cycle is proportional to the amplitude of the triangular carrier wave and can be calculated by

$$PWM_{Duty\ cycle} = \frac{u^*}{U_{dc}} + \frac{1}{2} \tag{3.67}$$

# 4 Simulations

To be able to perform numerical simulations of a complete microgrid, models of the converters with controllers, generator and grid have been created in MATLAB\SIMULINK.

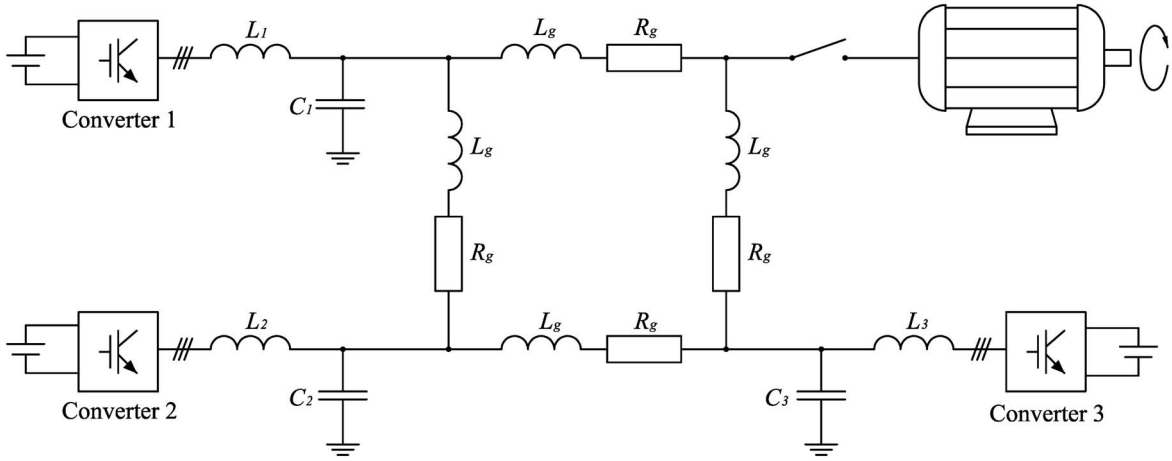


***Figure 4.1*** *Converters and generator connected in a microgrid.*

The modelled converters and generator have the following parameters.

***Table 4.1*** *Converter and generator parameters.*

| | Nominal power | Output filter |
|---|---|---|
| Converter 1: | $S_n$ = 4.5 kVA | $L_1$ = 5 mH<br>$R_{L1}$ = 0.1 Ω<br>$C_1$ = 20 μF<br>$ESR_1$ = 0.02 Ω |
| Converter 2: | $S_n$ = 3 kVA | $L_2$ = 7.5 mH<br>$R_{L2}$ = 0.1 Ω<br>$C_2$ = 14 μF<br>$ESR_2$ = 0.02 Ω |
| Converter 3: | $S_n$ = 7.5 kVA | $L_3$ = 3 mH<br>$R_{L3}$ = 0.05 Ω<br>$C_3$ = 35 μF<br>$ESR_3$ = 0.02 Ω |
| Generator: | $S_n$ = 2 kVA | |

The DC-link voltage is 270 V and the phase-to-phase voltage on the AC side is 145 V. The gains for the droop functions are determined from the maximum deviation in frequency and voltage, set to $\Delta_\omega$=0.005 and $\Delta_v$=0.04. The current controller gain is set to 0.2 times the deadbeat gain to get a stable controller. The estimator bandwidth and the filter cut-off frequency are set to

***Table 4.2*** *Filter coefficients.*

| | Cut-off frequency |
|---|---|
| $v_{q,f1}$ | 4.0 Hz |
| $v_{q,f2}$ | 1.0 Hz |
| $v_{q,f3}$ | 0.5 Hz |
| $v_{q,f4}$ | 5.0 Hz |
| $\rho_{\omega}$ | 5.356 Hz |

## 4.1 Simulink Models

To build models of the converter output filters and the grid, differential equations are used. The inductor in the output filter can be described by the equation

$$v_L = L\frac{di_L}{dt} + Ri_L \tag{4.1}$$

then the equation is integrated to get the voltage as input parameter and the current as output parameter

$$i_L = \frac{1}{L}\int v_L - Ri_L \tag{4.2}$$

For the capacitor the corresponding equations are

$$i_C = C\frac{dv_C}{dt} - Ri_C \tag{4.3}$$

$$v_C = \frac{1}{C}\int i_C + Ri_C \tag{4.4}$$

These equations are then implemented in SIMULINK to obtain a model of the output filter.

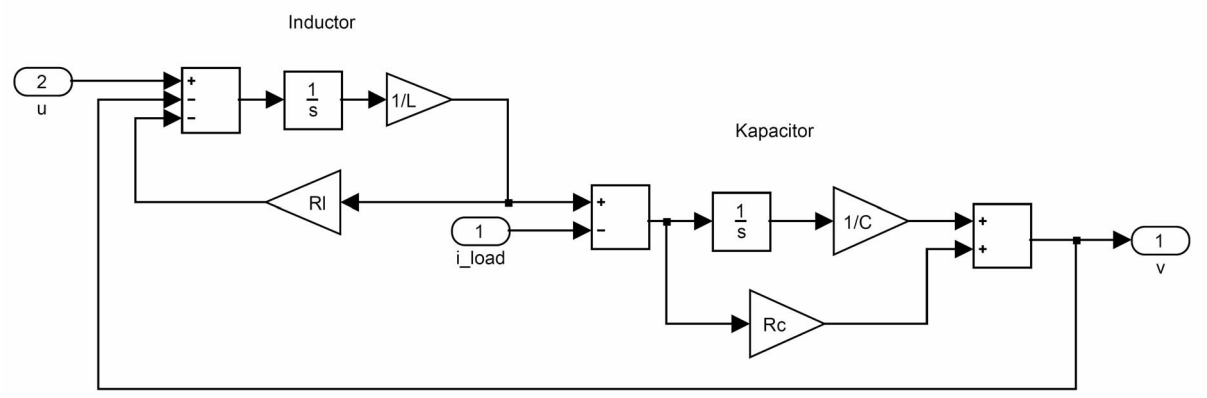


***Figure 4.2*** *SIMULINK model of converter output filter.*

In the same manner the grid model is designed with an inductance in series with a resistance. The generator is modelled with an emf in series with an inductor and a resistor.

## 4.2 Dynamic Properties of Active and Reactive Load Changes

The microgrid model shown in Figure 4.1 is designed in SIMULINK. The grid parameters are chosen to $L_g$=0.3mH and $R_g$=0.1Ω.

### Standalone Converter System

First a grid with only converters is studied. Due to the fact that converter 1 and 2 together has the same total nominal power as converter 3, converter 3 is chosen to act as load and the other two acting as sources. In the simulation, the power consumed by converter 3 is 1875 W active power and 1875 VAr reactive power, this corresponds to 0.25 p.u. for converter 3.
During the first second, the converters are operating at no load. Each converter then consumes the reactive power produced by the capacitor in their own output filters. After one second a step in active power is made by converter 3 and after two seconds there is a step in reactive power.

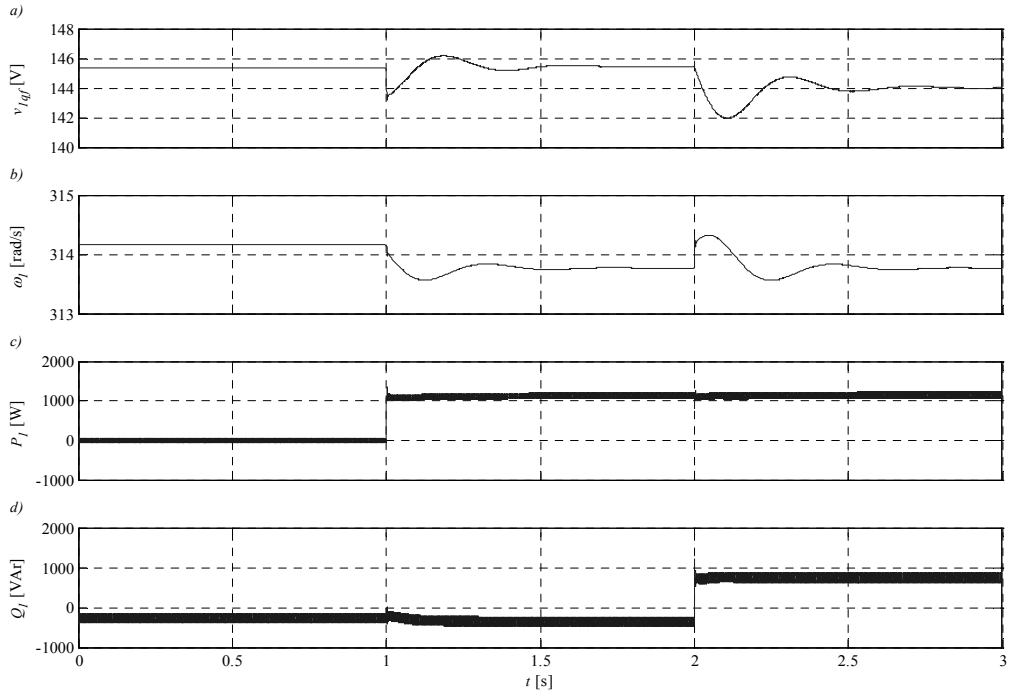


***Figure 4.3*** *Converter 1 a) output voltage, b) frequency, c) active power and d) reactive power.*

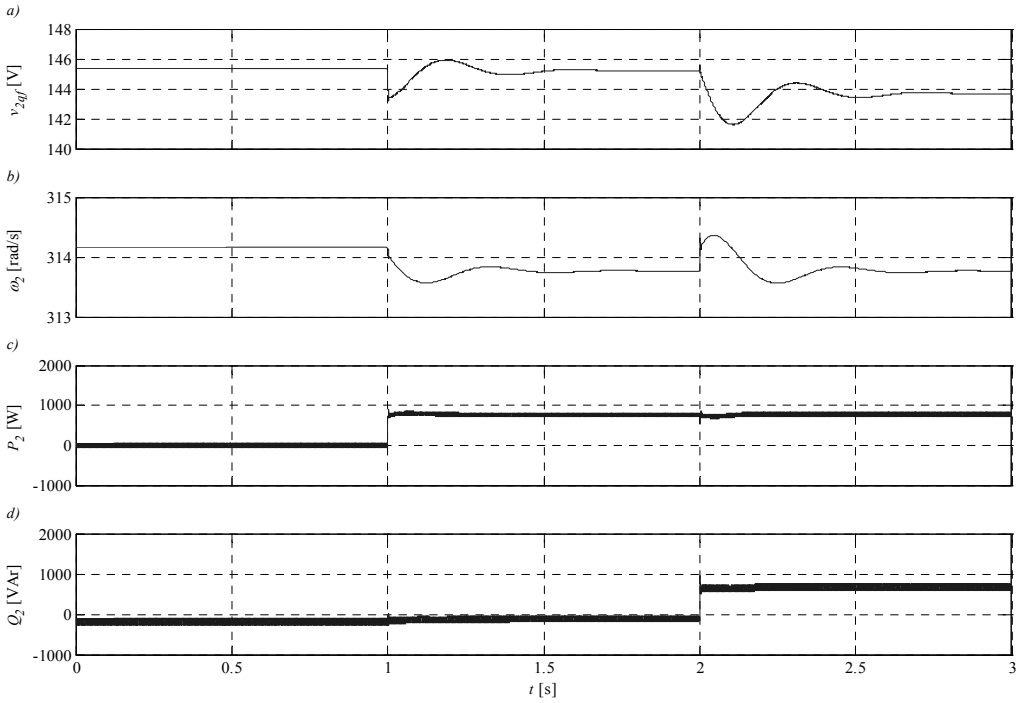


***Figure 4.4*** *Converter 2 a) output voltage, b) frequency, c) active power and d) reactive power.*

When the active power is connected the frequency drops and the sources starts to deliver active power corresponding to the frequency drop and each individual droop function. It is also obvious in Figure 4.3 and Figure 4.4 that the voltage drops at the same time but then recovers fast to the nominal voltage 145V. This is caused by the cross coupling between $i_d$ and $i_q$ in Equation (3.4) and (3.5) for the converter.
At $t$=2.0 s reactive power is connected and the voltage therefore drops. In the same manner as for active power, the cross coupling makes the frequency increase temporary and it is then stabilized at a value, determined by the active power.

Although the estimated frequency and the measured voltage are filtered with low bandwidth, the step response is very fast when the power is changed.

As seen in Figure 4.3 and Figure 4.4 the two source converters share the total load according to their nominal power. This is taken care of by the droop controller. The total reactive power produced by the converters is less than 1875 VAr because the capacitors in the output filters make a contribution to the total produced reactive power.

### Synchronous Generator

In this simulation three converters are connected together with a synchronous generator in a microgrid. A droop controller just like the slow droops controlling in the converters controls the generator. Frequency is controlled by controlling the torque feed into the generator and to regulate the voltage the field current is adjusted.

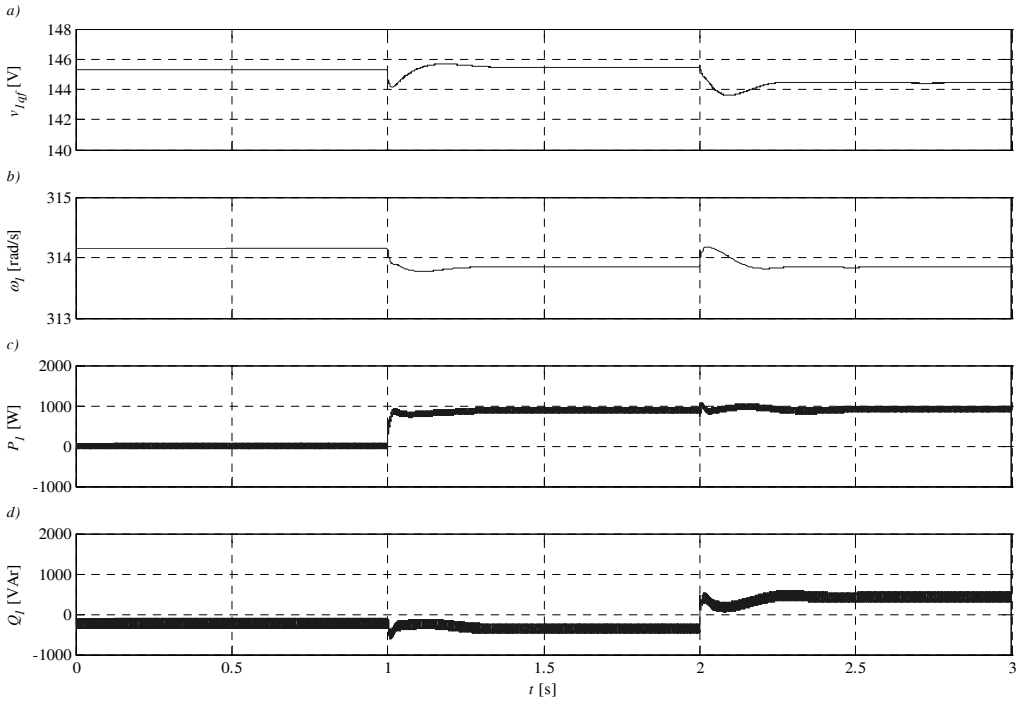


***Figure 4.5*** *Measurements from converter 1 connected in a microgrid with 3 converters and a generator. a) output voltage, b) frequency, c) active power and d) reactive power.*

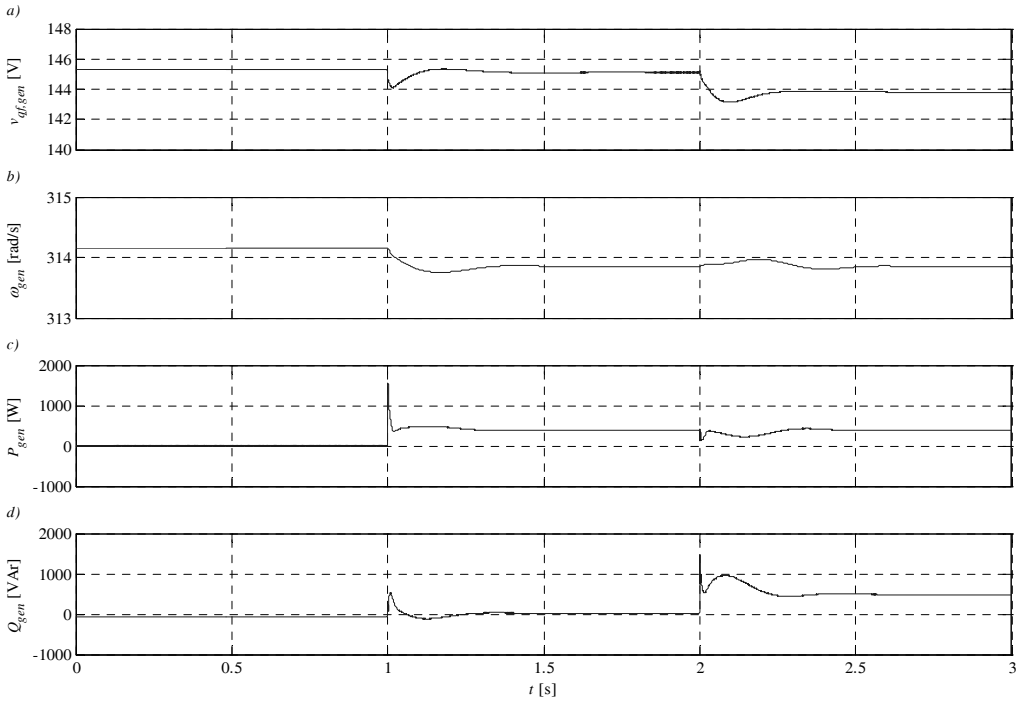


***Figure 4.6*** *Measurements from synchronous generator connected together with three converters. a) output voltage, b) frequency, c) active power and d) reactive power.*

Figure 4.5 and Figure 4.6 shows the result from converter 1 and the generator. Just like before, a load of 1875 W at $t$=1.0 s and 1875 VAr at $t$=2.0 s is connected.
The frequency is now much more stable because the generator has inertia and does not change speed fast. When rotating generators are connected in the grid it gets easier for the

converter controllers, they do not have to control the frequency alone. At the same time the voltage also gets more stable. The load sharing controlled by the droops still works well.

# 5 Experimental Verifications

To verify the simulation results the controller is implemented in a DSP-processor controlling the converters.

## 5.1 Hardware Description

The converters utilized have a maximum power of 10 kVA and is feed with DC from rotating converters, one for each converter. Each converter controller is implemented in a dSPACE digital signal processor mounted on a PCI-card, located in a PC. An interface card is connected between the DSP and the converter. The interface is designed so that switching can be turned off. Bit 11 in the digital IO turns the switching on and off. Current and voltage measurement cards convert the measured values to –10 to 10 V to be able to operate with the interface card and the DSP.

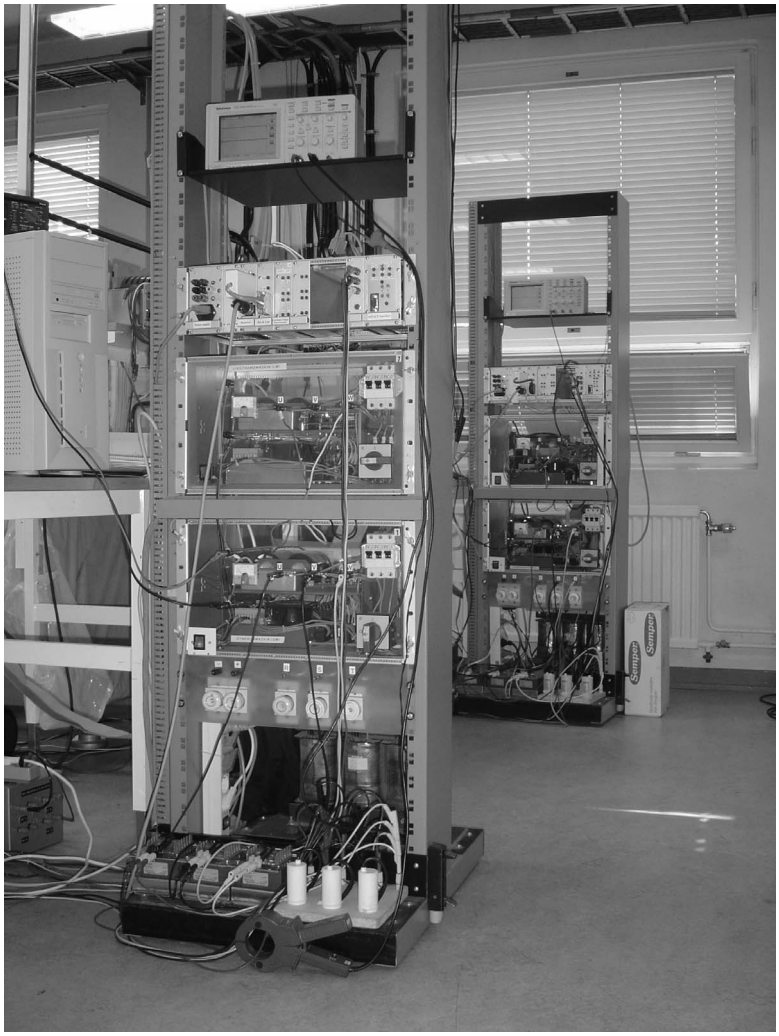

***Figure 5.1*** *Converter station 1 and 2. At the bottom of the picture the output filters with inductors and capacitors can be seen.*

In order to study the behaviour of the converters, when connected in a grid with rotating generators, a 2 kVA synchronous generator driven by a DC-machine is used, see Figure 5.2. The generator is droop controlled with the droop functions shown in Figure 3.5.

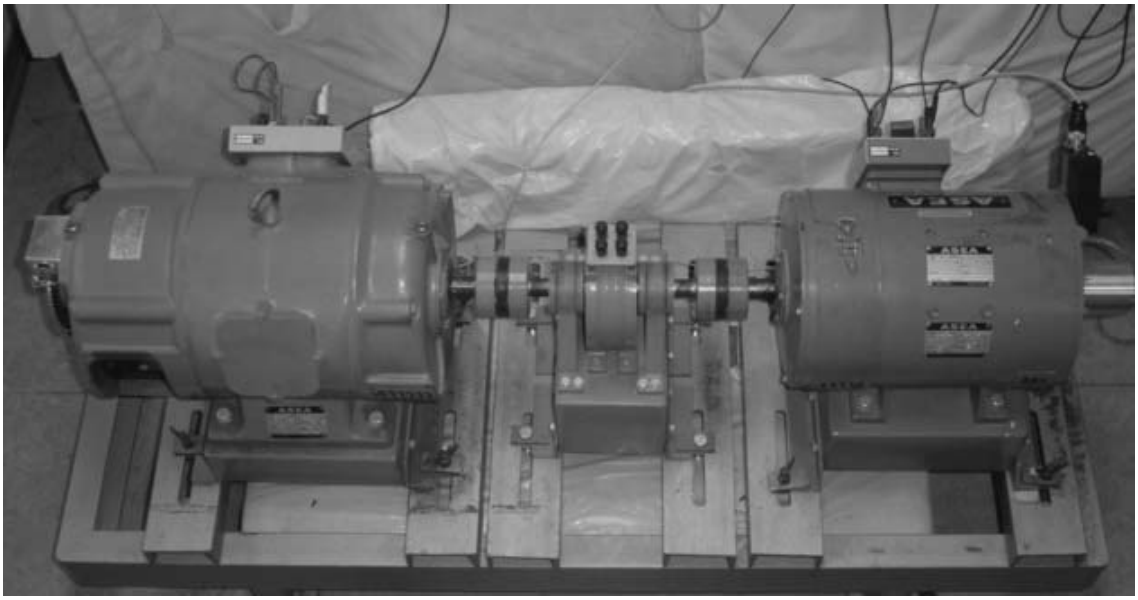

***Figure 5.2*** *To the left, the synchronous generator, driven by a DC machine.*

## 5.2 Software Description

The control algorithm for the converters is written in C-code. An advantage with the dSPACE software is that it is possible to plot and save all measured and calculated signals to the computer. Parameters can also be adjusted while the program is running. This is done from a instrumentation layout called ControlDesk. In ControlDesk different instruments, buttons etc. are put together to a layout file (`.LAY`) .
Three files are downloaded to the DSP. The C file containing the controller code, a `TRC` file with all parameters to be accessed and changed by ControlDesk and a `SDF` file containing information about the current dSPACE card. In the C file all variables that needs to be changed while running the program is declared as `volatile`. These three files, together with the layout files, are saved in an experiment. If many parameters are used it is suitable to organize them in groups. An example of a `TRC` file can look like

```
sampling_period = 1.0E-4

_floating_point_type(64,IEEE)
_integer_type(32)

group "Model"
  enable  flt
  Sn      flt

  group "Model Parameters"
    P       flt
    Q       flt
    vqf1    flt
    w_hat   flt
    K_adj   flt
  endgroup

endgroup
```

Note that there has to be an empty line before `endgroup`.

In the `SDF` file it is only the type of dSPACE board and the filename that has to be changed.

```
[RTP]
Type=DS1104
BoardName=ds1104
File=regsys.ppc
TraceFile=regsys.trc

[System]
Version=1.0
Status=Start
RTP=RTP
SystemType=SingleProcessorSystem
```

To compile the files, open a DOS window and go to the directory where the experiment is saved. Then type `down1104 regsys`, where `1104` is the type of dSPACE card and `regsys` is the experiment name.

### ControlDesk

After compilation and downloading, the parameters listed in the `TRC` file are available in ControlDesk.

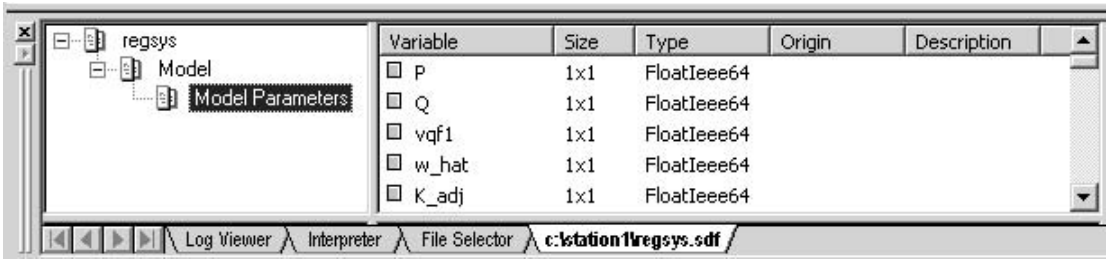


***Figure 5.3*** *Example of parameters listed in ControlDesk.*

To create a new experiment chose `File-New Experiment`, then type the name of the experiment and the working root. Open the files created before and chose `File - Add All Opened Files`, to add the files to the experiment.

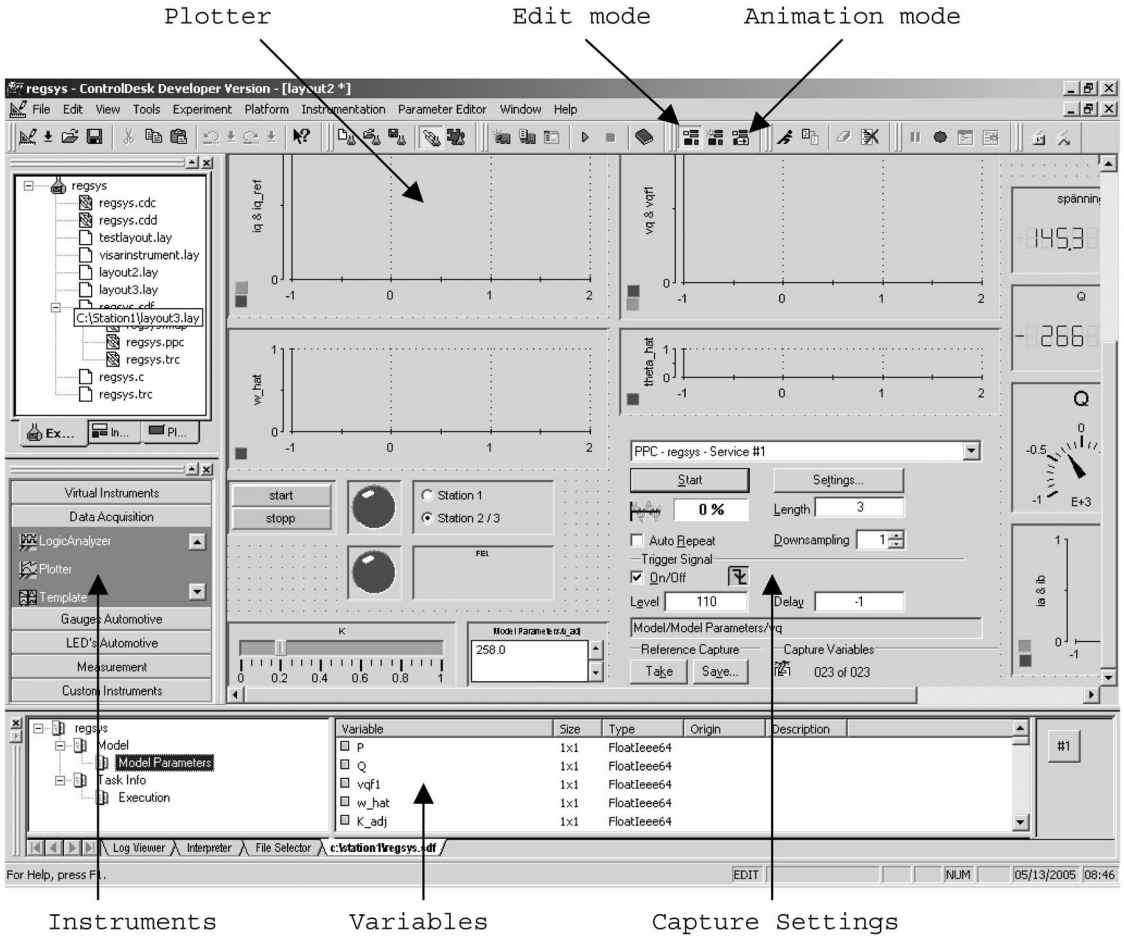


***Figure 5.4*** *Example of layout in ControlDesk.*

A new layout is created by choosing `File - New - Layout`. Then different instruments are chosen from the instrumentation list, see Figure 5.4. Instruments are inserted in the layout by clicking at them in the list and then draw a box in the layout. The instrument is then marked by a red rectangle, indicating that no signal is connected to it. Signals are connected to the instrument by dragging them from the variable list to the instrument. To do this the instrumentation panel `Edit mode` is selected. To run the program the `Animation mode` is selected.

The variables displayed in the plots can also be saved to a file. To do this the instrument `Capture settings` is chosen from the instrument list. The data capture is started manually or by a trigger variable chosen from the variable list. Different settings can be used, for example rising or falling edge, trigger level and delay. When the button `Save` is pushed a `MAT`-file is created. To use the `MAT`-file in MATLAB the following lines are useful.

```
time=double(filename.X.Data');
signal1=double(filename.Y(1).Data');
signal1description=filename.Y(1).Name
```

### C-code

SIMULINK models can be compiled and downloaded direct to dSPACE. Disadvantage with this method is that the user does not have full control over the generated code. The code gets very large and the executing time therefore very long. With the model earlier created in SIMULINK it would be impossible to run the generated code with the desired sample frequency. Therefore the controller is hand coded in C. The execution time is then about 30 μs. With a sampling time of 100 μs there is plenty of time for further calculations.

The code is structured so that all calculations are made in the PWM-interrupt. The interrupt occur once each switching period. Due to this construction the sample frequency equals the switching frequency. The sampling then occurs at the negative peak of the triangular wave, see Figure 3.12.

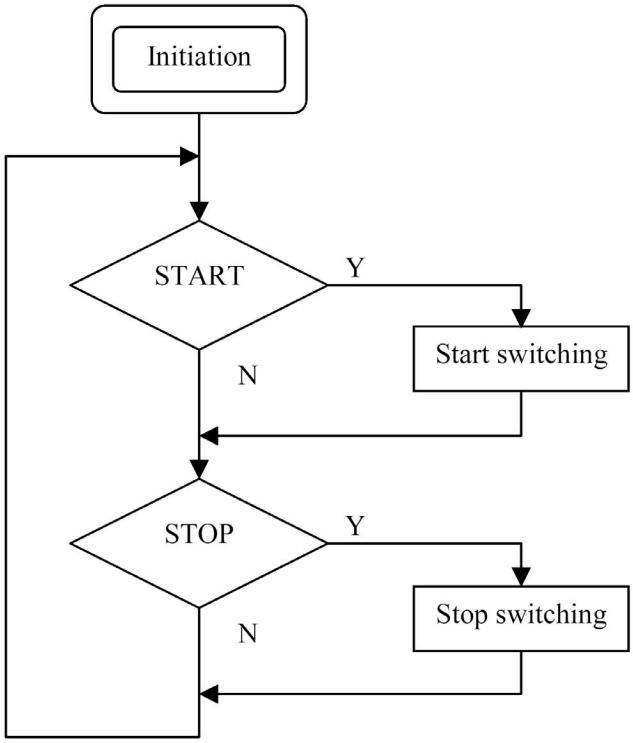


***Figure 5.5*** *Main program flow chart.*

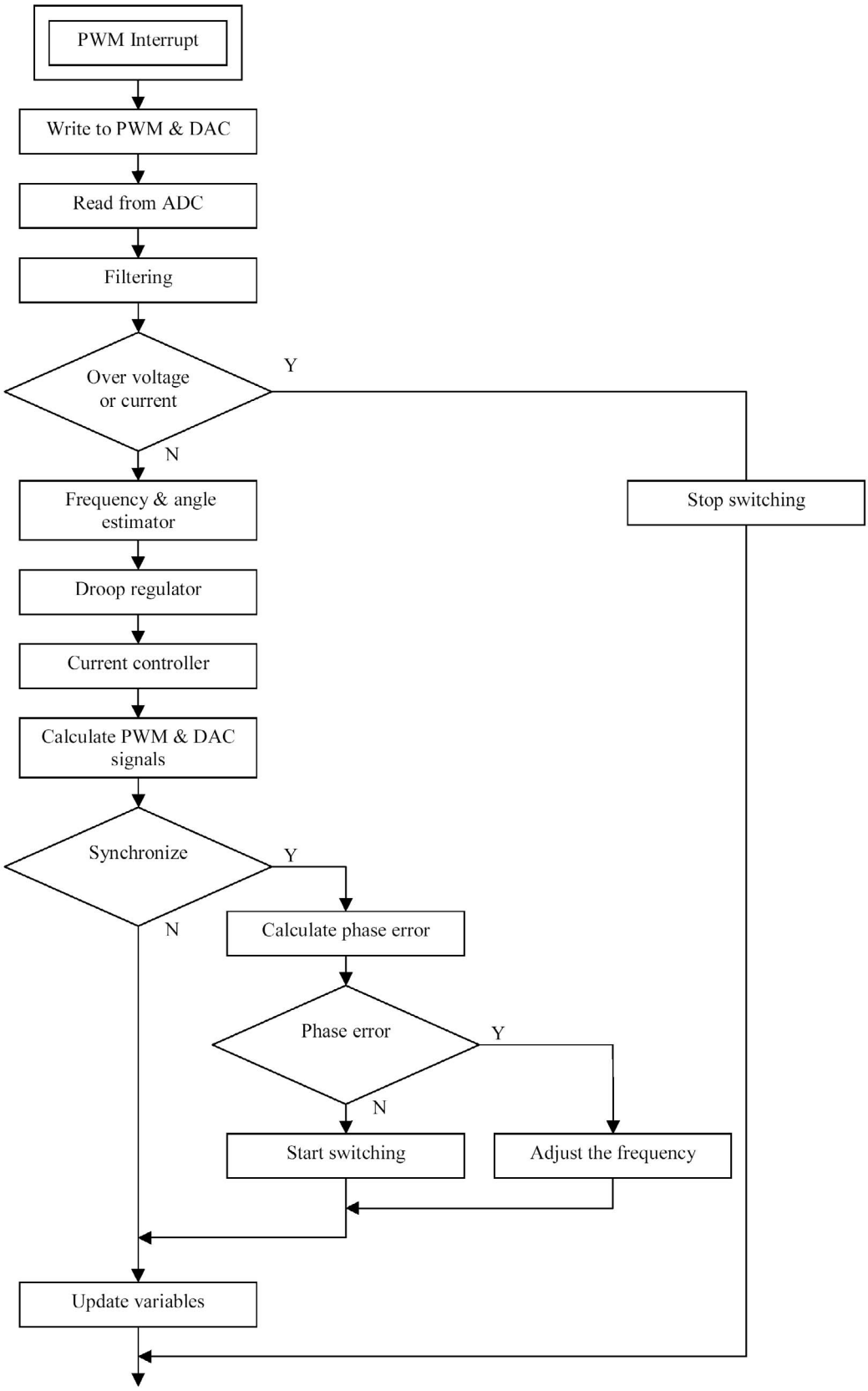


***Figure 5.6*** *PWM interrupt flow chart.*

In the beginning of the PWM-interrupt a time measuring function is started to be able to measure the execution time. Then the previously calculated PWM duty cycles are written to

the PWM and a strobe pulse is sent to the DAC to activate all the previous written values at the same time.

```
host_service(1, 0);                 /* Data acquisition service */
RTLIB_TIC_START();                  /* Start time measurement */

/* write PWM Duty cycle to slave DSP and test for error */
ds1104_slave_dsp_pwm3_duty_write(task_id, index, duty1, duty2, duty3);

/* activate the previously written DAC values synchronously */
ds1104_dac_strobe();
```

Then it is time to read the signals from the A/D-converter. In the version of dSPACE used in this project there are four muxed and four unmuxed channels. In this case only one of the muxed channels is used. All A/D-converters are started at the same time and then their values are read. When more than one of the muxed channels is used they have to be started with delay to synchronize all the read values.
The measured signals are adjusted according to the scaling factors of the measuring cards.

```
ds1104_adc_start(DS1104_ADC2|DS1104_ADC3|DS1104_ADC4|DS1104_ADC5);

ds1104_adc_read_mux(scantable, 4, u);
Udc = u[3];

ia = ds1104_adc_read_conv(2);
ib = ds1104_adc_read_conv(3);

valfa = ds1104_adc_read_conv(4);
vbeta = ds1104_adc_read_conv(5);

Udc = Udc*408;

ia = ia*25;
ib = ib*25;

valfa = valfa*v_adj;
vbeta = vbeta*v_adj;
```

The signals read from the voltage measurement card are in the $\alpha\beta$-frame. By definition the voltage over the capacitor, in the output filter, is only in the $q$ direction. Due to this fact the voltage $v_q$ can easily be calculated as the absolute value of the $\alpha\beta$-components. The phase voltages are also calculated.
Only the $a$- and $b$-phase currents are measured and assuming symmetry, the current in phase $c$ is calculated. Then the phase currents are converted to the $\alpha\beta$-frame and later on to the $dq$-frame. To do this the angle `theta_hat_old`, calculated at previous sample, is used.

```
/*--------------------Calculation of vq-----------------------------*/
vq = sqrt((valfa*valfa)+(vbeta*vbeta));

/*--------------------Vectors to abc--------------------------------*/
va = sqrt(0.66666667)*valfa;
```

```
vb = -sqrt(0.16666667)*valfa + sqrt(0.5)*vbeta;
vc = -sqrt(0.16666667)*valfa - sqrt(0.5)*vbeta;

/*--------------------Calculation of i_alfa,beta--------------------*/
ic = -ia-ib;
sym = (ia+ib+ic)/3;
ia = ia-sym;
ib = ib-sym;
ic = ic-sym;
ialfa = sqrt(3/2)*ia;
ibeta = sqrt(0.5)*(ib-ic);

/*--------------------ab to dq-------------------------------------*/
id = ialfa*sin(theta_hat_old) - ibeta*cos(theta_hat_old);
iq = ialfa*cos(theta_hat_old) + ibeta*sin(theta_hat_old);
```

Filters with different cut-off frequencies are used to filter the voltage $v_q$.

```
/*--------------------Low-pass Filter------------------------------*/

vqf1 = ((vq-vqf1_old)*w1*Ts)+vqf1_old;     //LP-filter1 fc=4Hz

vqf2 = ((vq-vqf2_old)*w2*Ts)+vqf2_old;     //LP-filter2 fc=1Hz

vqf3 = ((vq-vqf3_old)*w3*Ts)+vqf3_old;     //LP-filter3 fc=0.5Hz
```

Different kinds of protection are used. It is important to have protection so the switching does not start without any DC-link voltage, `Udc`, which would lead to short circuit like behaviour, until the DC-link capacitors are charged, if the converter is connected to the grid at the AC side. The over voltage protection is based on the unfiltered signal to be able to trip fast if an error occurs. Under-voltage is not so critical and is therefore based on a filtered voltage to avoid tripping when a large step in power is made. Then the unfiltered voltage reaches almost zero for a moment. Under-voltage protection is operating only when the switching is on, otherwise it would be impossible to start the converter.
When a fault occurs `enable` is set to zero and then the switching stops. The variable `error_type` indicates which kind of fault it is.

```
/*--------------------Over current protection----------------------*/
if(id>imax || iq>imax || id<(-imax) || iq<(-imax))
{
  enable = 0;
  error_type = 1;
}

/*--------------------Under voltage protection for Udc--------------*/
if(Udc<200)
{
  enable = 0;
  error_type = 4;
}
```

```
/*--------------------Over voltage protection----------------------*/
if(vq>vmax)
{
  enable = 0;
  error_type = 2;
}

/*--------------------Under voltage protection---------------------*/
if(vqf2<vmin && converter_on ==1)
{
  enable = 0;
  error_type = 3;
}
```

The frequency and angle estimator described in Equation (3.37) and (3.38) is then implemented. The estimated frequency is limited between 49 and 50 Hz to make the start up easier. When the converter is running the frequency never reaches these limits.
The angle is linearly increased and to avoid overflow in the processor the angle is subtracted with $2\pi$ every period.
When the switching is off, `converter_on==0`, the frequency is fixed to 50 Hz plus a term `w_adj`. `w_adj` is given by the synchronization algorithm, described later on.

```
/*--------------------Estimator-------------------------------------*/
w_hat=w_hat_old+((ra_w*ra_w*Ts)*(-ud_ref_old+(R*id_ref_old)- ↵
 (w_hat_old*L*iq_ref_old)))/Eqn;
theta_hat=theta_hat_old+w_hat_old*Ts+((2*ra_w*Ts)* ↵
 (-ud_ref_old+(R*id_ref_old)-(w_hat_old*L*iq_ref_old)))/Eqn;

if(w_hat<(2*PI*49))
w_hat = 2*PI*49;
if(w_hat>(2*PI*51))
w_hat = 2*PI*51;

if(theta_hat_old>(2*PI))
{
  theta_hat = theta_hat - (2*PI);
  theta_hat_old = theta_hat_old - (2*PI);
}

if(converter_on ==0)
{
  w_hat = (2*PI*50)+w_adj;
}

//LP-filter4 5Hz
wf4 = ((w_hat-wf4_old)*w4*Ts)+wf4_old;
```

The droop controller is implemented according to Equation (3.21), (3.22), (3.19), and (3.20). When `aut==1` the droop controller is in action, `aut==2` means that the power reference values are set to zero and the converter is operating at no load. In this case the old current references, and the integral part, of the current controller are set to zero to make the controller

start without any old values. The error from the previous values could otherwise result in large transients. The references can also be manually set. This is used when the converter is operating as load, the desired load is defined by `P_ref_man` and `Q_ref_man`.
The determined power references are recalculated to current references and these are limited to the maximum output current `imax`.

```
/*--------------------Droop controller-------------------------------*/
if(aut==1)
{
  //Slow droops
  vq_ref = vqf2_old + (Kw/Kv)*(wn-wf4_old);
  w_ref = wf4_old - (Kv/Kw)*(Eqn-vqf2_old);

  //Fast droops
  P_ref = Kv*(vq_ref-vqf1);
  Q_ref = -Kw*(w_ref-w_hat_old);
}

//Idling
else if(aut==2)
{
  P_ref = 0;
  Q_ref = 0;

  id_ref_old = 0;
  iq_ref_old = 0;
  ud_I = 0;
  uq_I = 0;
}

//Manual reference values
else
{
  P_ref = P_ref_man;
  Q_ref = Q_ref_man;
}

/*--------------------P,Q to idq-------------------------------------*/
id_ref = Q_ref/vqf3_old;
iq_ref = P_ref/vqf3_old;

//Current limitation
if(id_ref > imax)
  id_ref = imax;

if(iq_ref > imax)
  iq_ref = imax;

if(id_ref < (-imax))
  id_ref = -imax;
```

```
if(iq_ref < (-imax))
  iq_ref = -imax;
```

According to Equation (3.62) and (3.63) the current controller is implemented together with the anti-windup introduced in Equation (3.64).

```
/*--------------------Current controller----------------------------*/
ud_I_temp = K_adj*K*(Ts/Ti)*(id_ref-id) + ud_I;
uq_I_temp = K_adj*K*(Ts/Ti)*(iq_ref-iq) + uq_I;

ud_ref = K_adj*K*(id_ref-id) + ud_I - (Kc*0.5*(iq_ref+iq));
uq_ref = K_adj*K*(iq_ref-iq) + uq_I + (Kc*0.5*(id_ref+id)) + vqf1;

//Anti-windup
if(ud_ref>umax)
{
  ud_ref = umax;
  ud_I = umax - (K_adj*K*(id_ref-id));
}
else if(ud_ref<umin)
{
  ud_ref = umin;
  ud_I = umin - (K_adj*K*(id_ref-id));
}
else
{
  ud_I = ud_I_temp;
}

if(uq_ref>umax)
{
  uq_ref = umax;
  uq_I = umax - (K_adj*K*(iq_ref-iq));
}
else if(uq_ref<umin)
{
  uq_ref = umin;
  uq_I = umin - (K_adj*K*(iq_ref-iq));
}
else
{
  uq_I = uq_I_temp;
}
```

The calculated voltage references are transformed back to the $\alpha\beta$-frame and then to phase voltage refererences. Then the phase voltage references are symmetrized to be able to use as much of the DC-link voltage as possible and avoid overmodulation, see Equation (3.65) and (3.66) and Figure 3.11. The resulting phase potential references are then recalculated to PWM duty cycle. The duty cycle is defined between zero and one, where one equals half the DC-link voltage.

```
/*--------------------dq to ab-------------------------------------*/
ualfa_ref = ud_ref*sin(theta_hat_old) + uq_ref*cos(theta_hat_old);
ubeta_ref = -ud_ref*cos(theta_hat_old) + uq_ref*sin(theta_hat_old);

/*--------------------Vectors to abc-------------------------------*/
ua_ref = sqrt(0.66666667)*ualfa_ref;
ub_ref = -sqrt(0.16666667)*ualfa_ref + sqrt(0.5)*ubeta_ref;
uc_ref = -sqrt(0.16666667)*ualfa_ref - sqrt(0.5)*ubeta_ref;

/*--------------------Symmetry-------------------------------------*/
max=ua_ref;
if(ub_ref>max)
  max=ub_ref;
if(uc_ref>max)
   max=uc_ref;

min=ua_ref;
if(ub_ref<min)
  min=ub_ref;
if(uc_ref<min)
  min=uc_ref;

uz_ref=(max+min)/2;

ua_ref = ua_ref-uz_ref;
ub_ref = ub_ref-uz_ref;
uc_ref = uc_ref-uz_ref;

/*--------------------PWM duty cycle-------------------------------*/
duty1 = (ua_ref/Udc)+0.5;
duty2 = (ub_ref/Udc)+0.5;
duty3 = (uc_ref/Udc)+0.5;
```

Values are written to the D/A-converter to be ready when the strobe pulse comes at the next sampling instant.

```
/*--------------------DAC------------------------------------------*/
ds1104_dac_write(1,(valfa/300));
ds1104_dac_write(2,(ualfa_ref/300));
```

At the end of the interrupt the values calculated during this sampling interval are set as old values for the next sampling interval. Then the execution time is read.

```
/*--------------------Uppdate variables----------------------------*/
vqf1_old = vqf1;
vqf2_old = vqf2;
vqf3_old = vqf3;
w_hat_old = w_hat;
theta_hat_old = theta_hat;
wf4_old = wf4;
ud_ref_old = ud_ref;
```

```
id_ref_old = id_ref;
iq_ref_old = iq_ref;

exec_time=RTLIB_TIC_READ();
```

In the beginning of the main program initialization is performed for the DSP card, the digital IO, communication between master- and slave-DSP and the D/A-converter. For further information about initialization see dSPACE help/DS1104 RTLib Reference [13].

The infinite loop of the main program takes care of start and stop of the PWM interrupt and the initial values are also set here.

`station==1` should be selected at the station started first in the standalone grid, then the synchronization is inactive at startup. To be able to synchronize the other stations the zero crossings for the measured voltage `va` and the reference voltage `ua_ref` are detected. The number of samples between the two zero crossings are counted, if the number of samples equals a phase error less than 3.6 degrees, i.e. 200 μs, the switching starts. If the error is larger than 3.6 degrees the frequency of `u_ref` is adjusted. The frequency is increased in proportion to the phase error and therefore approaches 50 Hz when the error becomes smaller. The reason for choosing 3.6 degrees is that this equals two samples. A smaller error is hard to detect. The gain in the frequency adjustment is chosen to get a smooth synchronization.

```
/*--------------------Synchronization-------------------------------*/
if(fas==0 && station!=1)
{
  if(va>20)
  {
    trigg1 = 1;
  }
  if(ua_ref>20 && trigg1==1)
  {
    trigg2 = 1;
  }
  //va zero crossing
  if(trigg1==1 && va<10 && va>-10 || count!=0)
  {
    count = count+1;
  }
  //ua_zero crossing
  if(trigg2==1 && ua_ref<10 && ua_ref>-10 && count!=0)
  {
    error = count;
    count = 0;
  }
  if(error!=0)
  {
    //start the switching if the error is less than 3.6 degrees
    if(((error*Ts)/20e-3)<0.01)
    {
      fas = 1;
```

```
            on = 1;
          }
          else
          {
            w_adj = ((2*PI*error*Ts)/20e-3);
            error = 0;
            trigg1 = 0;
            trigg2 = 0;
          }
        }
      }
```

The entire code, with all initializations, is found in Appendix B.

## 5.3 Active and Reactive Load

Similar to the simulations, converter 3 is acting as load and the other two converters as sources. 1875 W active load is connected after one second and 1875 VAr reactive load after two seconds. As seen in Figure 5.7 and Figure 5.8 the curves are essentially the same as in the simulation except for the fact that the active power does not increase to the new level as fast as in the simulation. This might depend on converter 3 which cannot consume the full amount of power immediately. It could also depend on parameters that are different in the simulation and reality. Varying the parameters of the simulation have been investigated to achieve the same behaviour, without any result. The simulation result is still perfect. Another possible explanation could be the rotating converters feeding the converters with DC. They have inertia and do not change speed very fast. Another reason could be that one of the DC generators is very heavily magnetized to get the required voltage, which means it starts behave nonlinear. The long cables between the DC generator and the converter means a large resistance is inserted which could also be a reason for the difference between reality and simulations.

The active load sharing works well which is due to the fact that it depends on the frequency and the frequency is the same in the entire grid. It is more complicated when it comes to reactive load sharing. There is impedance in the grid, which means that the voltage is not the same at all nodes. This affects the droop function and leads to an error in reactive load sharing. A source located far away from the load take up less load per unit than a source close to the load. Another problem is that the droop controllers are so sensitive that a small error in the voltage measurement leads to a large error in load sharing.

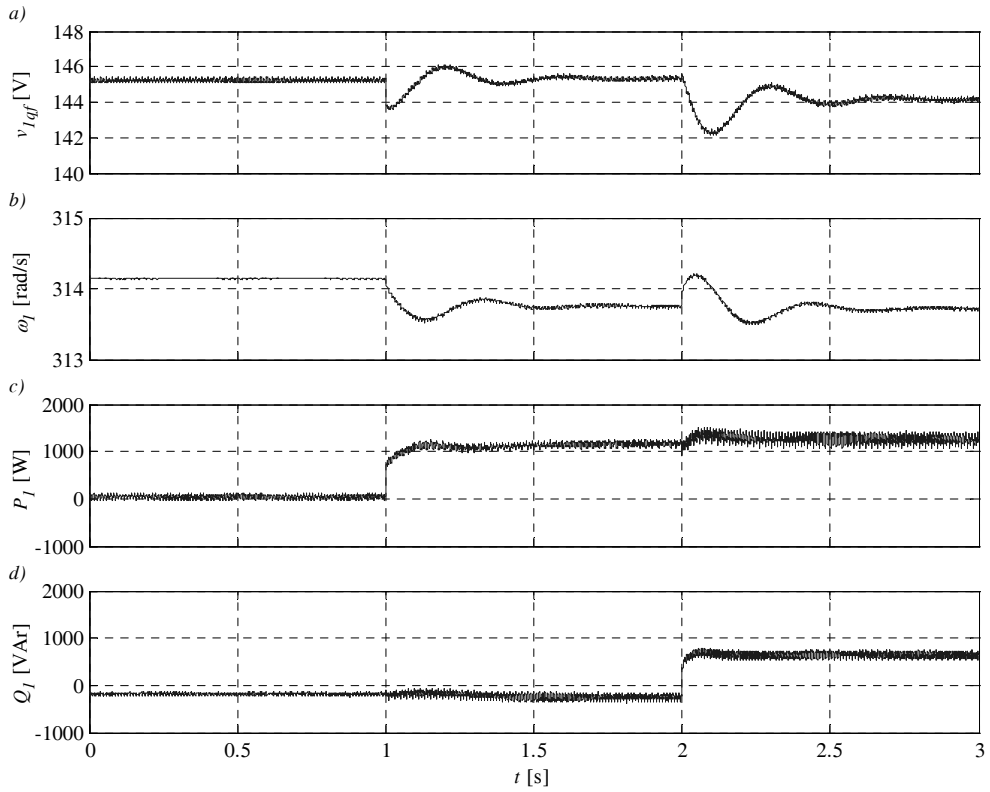


***Figure 5.7*** *Connecting first active then reactive load. Converter 1 a) output voltage, b) frequency, c) active power and d) reactive power.*

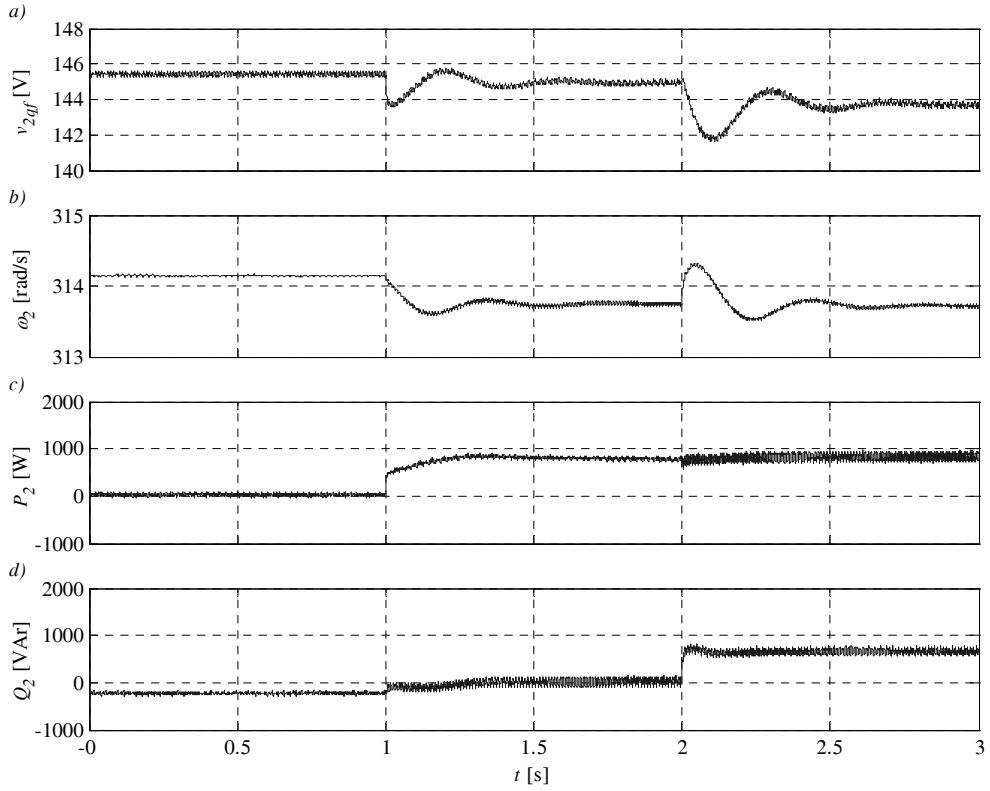


***Figure 5.8*** *Connecting first active then reactive load. Converter 2 a) output voltage, b) frequency, c) active power and d) reactive power.*

The most important thing to note is that the voltage and frequency control operates well. In less than half a second both voltage and frequency has stabilized after the step in power. Although there are some oscillations in voltage and frequency, they are well within their limits. The maximum deviations are $\Delta_v = 0.05$ and $\Delta_\omega = 0.004$ which equals a voltage between 137.8 and 152.3 V and a frequency between 312.9 and 315.4 rad/s.
The cross coupling between *P* and *Q* is obvious in Figure 5.7 and Figure 5.8.

Also the case with connecting reactive load first and then active load has been studied. In this case, see Figure 5.9, the cross coupling gets more obvious but the controller has no problem controlling both voltage and frequency.

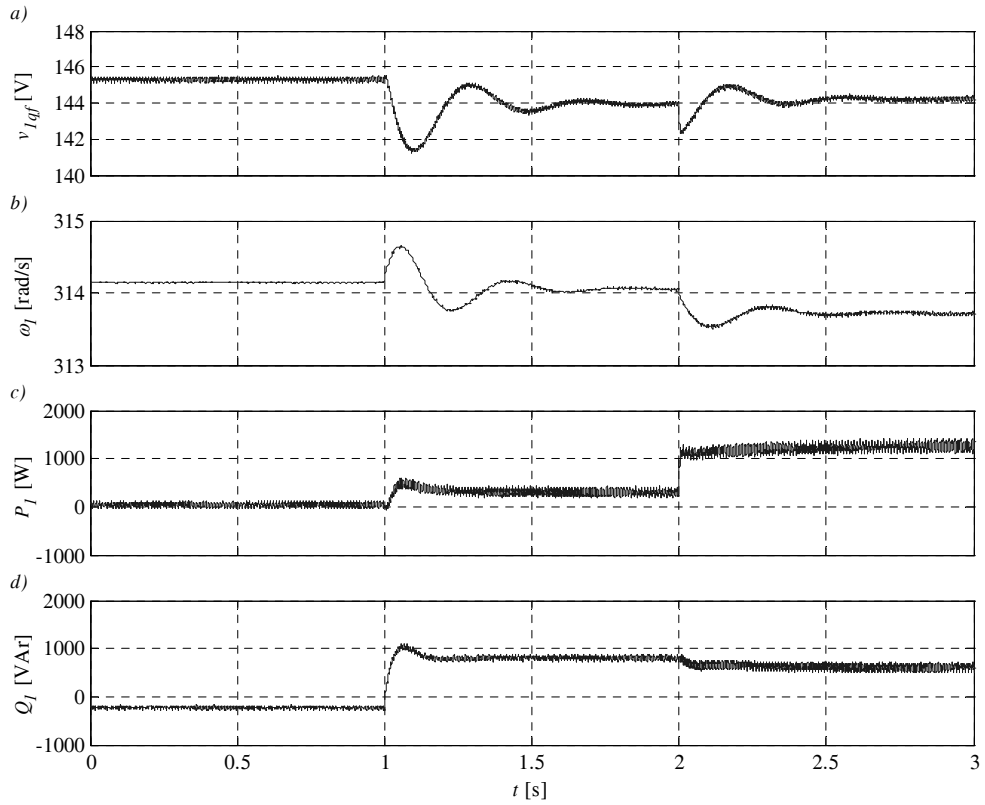


***Figure 5.9*** *Connecting first reactive then active load. Converter 1 a) output voltage, b) frequency, c) active power and d) reactive power.*

## 5.4 Disconnecting Load

In Figure 5.10 disconnecting both active and reactive power at the same time are shown. This could correspond to a circuit breaker opening, caused by an error in the grid. It could be a problem to maintain a stable voltage and frequency in such a case but with this method it is no problem. In less than one seconds the part of the grid, still in operation, has stabilized. It is very important to be able to maintain a part of the grid when another part trips.

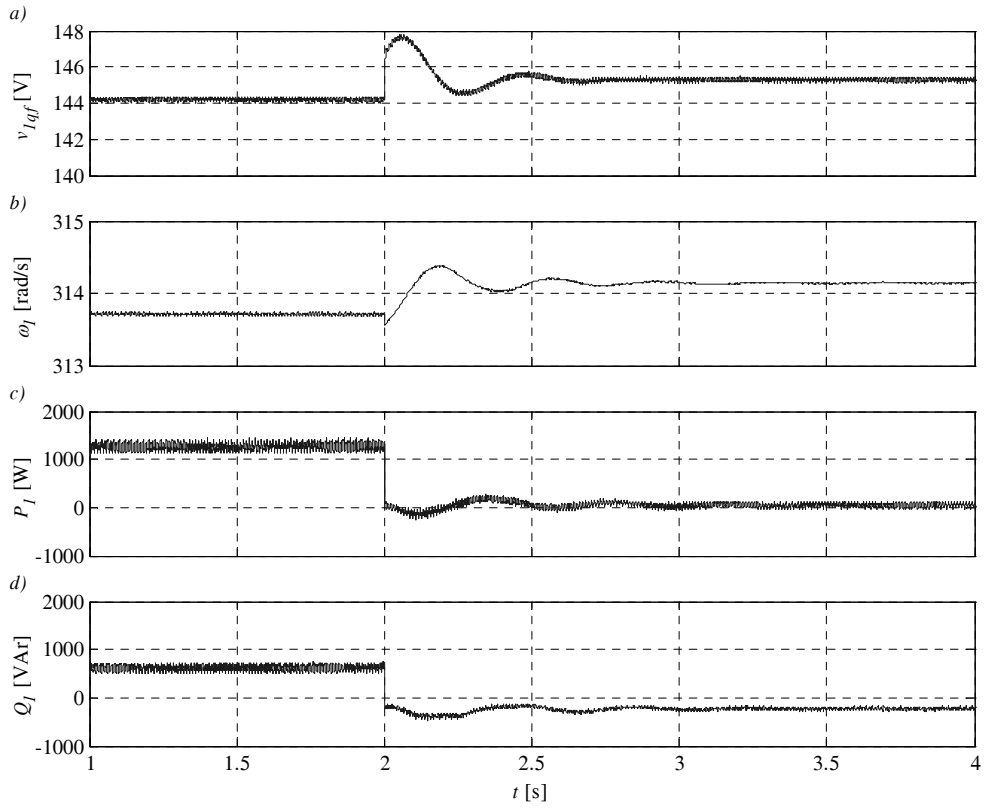


***Figure 5.10*** *Disconnecting active and reactive load at the same time. Converter 1 a) output voltage, b) frequency, c) active power and d) reactive power.*

## 5.5 Synchronous Generator

A four pole 2 kVA synchronous generator is connected together with the three converters. The generator is controlled by a droop controller. Just like before one converter is acting as load and the other two as sources. The result in Figure 5.11 is almost identical to the case without the generator. According to the simulations carried out earlier the frequency should be more stable when the generator is connected but one problem in the experiment is that the generator is located far away from the load. This means that the impedance in the cable together with the inductance in the transformer, needed to be able to connect the generator to the converters, affects the result.

A common problem with droop control is when the impedance between the sources is large. Then the voltage drop affects the droop controllers and the load sharing does not work properly. The problem is worst in the control of reactive power because it is based on the voltage. Frequency is the same in the entire grid so the control of active power is easier.

The most important thing to notice is that the converter control still works well also when a droop controlled rotating generator is connected in the grid. This means that the converters could easily operate together with the public grid, where most sources are synchronous generators.

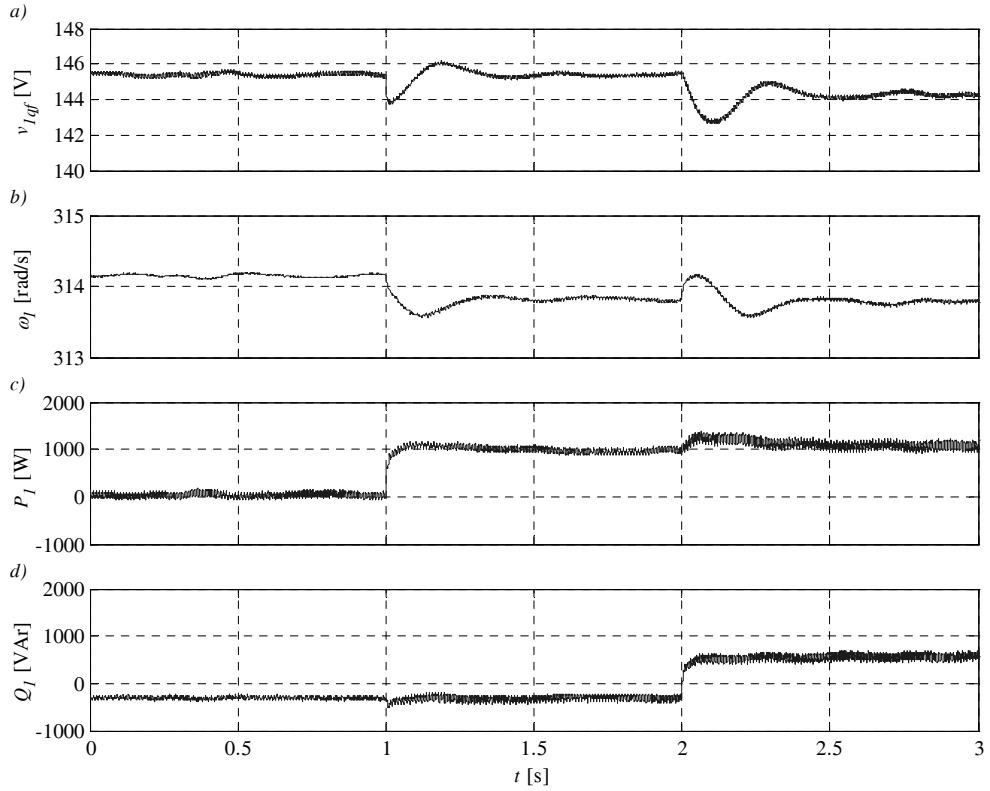


***Figure 5.11*** *Synchronous generator together with three converters in a microgrid. Converter 1 a) output voltage, b) frequency, c) active power and d) reactive power.*

## 5.6 Synchronization

To be able to connect converter to a grid, a synchronization algorithm is needed. The grid voltage is measured and the controllers of the converters are running, but with the switching turned off. The zero crossings are detected and the phase difference between the converters coordinate system and the grid coordinate system is measured. To synchronize the two coordinate systems the frequency of the converter is increased proportional to the phase error. When the phase error is less than a two samples, equal to 3.6 degrees, switching is turned on.

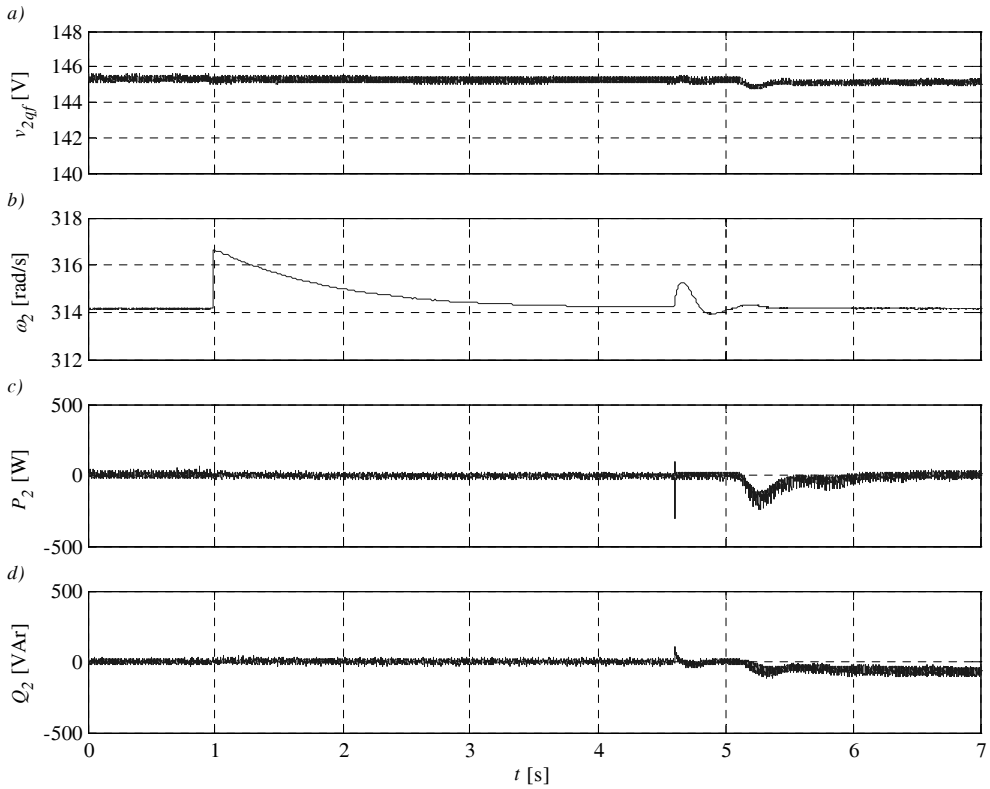


***Figure 5.12*** *Synchronisation of converter 2 a) output voltage, b) frequency, c) active power and d) reactive power.*

At $t$=1.0 s the synchronization algorithm is turned on and the frequency is adjusted to bring the two coordinate systems in phase. In about 3.6 seconds the phase error is small enough and the switching starts. The converter operates at no load for 0.5 s and then the controller is turned on and is ready to take up load. This gives a very smooth synchronization with only a small amount of power transmitted at connection.
The disadvantage of this method is that the frequency of the converter is always increased, no matter how big the phase error is, this means that in some cases the coordinate system needs to be adjusted a full period. This can take up to a few seconds, especially when a smooth synchronization is desirable.

## 5.7 Line Voltage and Current

In section 3.4, the transistors were assumed to be ideal, meaning the two switches in a half bridge could change states at the same time. In practice, switching devices are non-ideal. To avoid a short circuit of the DC-link across the bridge, a blanking time is applied. This blanking time allows each switch to become completely turned off before the other one is turned on. Blanking time is applied by the hardware and appears as a small deformation in the current at the zero crossings in Figure 5.13. The ripple in voltage and current is a result of switching and has the same frequency as the switching, i.e. 10 kHz. Despite this deformation in both voltage and current appear to be almost sinusoidal with the spectrums shown in Figure 5.14. Obviously, the harmonics are of small magnitudes, meaning the curves are very close to a 50 Hz sine wave. There is a small DC-offset in the current measuring causing the DC-component in the current spectrum. The harmonics caused by the switching are almost eliminated by the output filters. In the current spectrum, only small harmonics remains around the switching frequency, i.e. 10 kHz, see Figure 5.15.

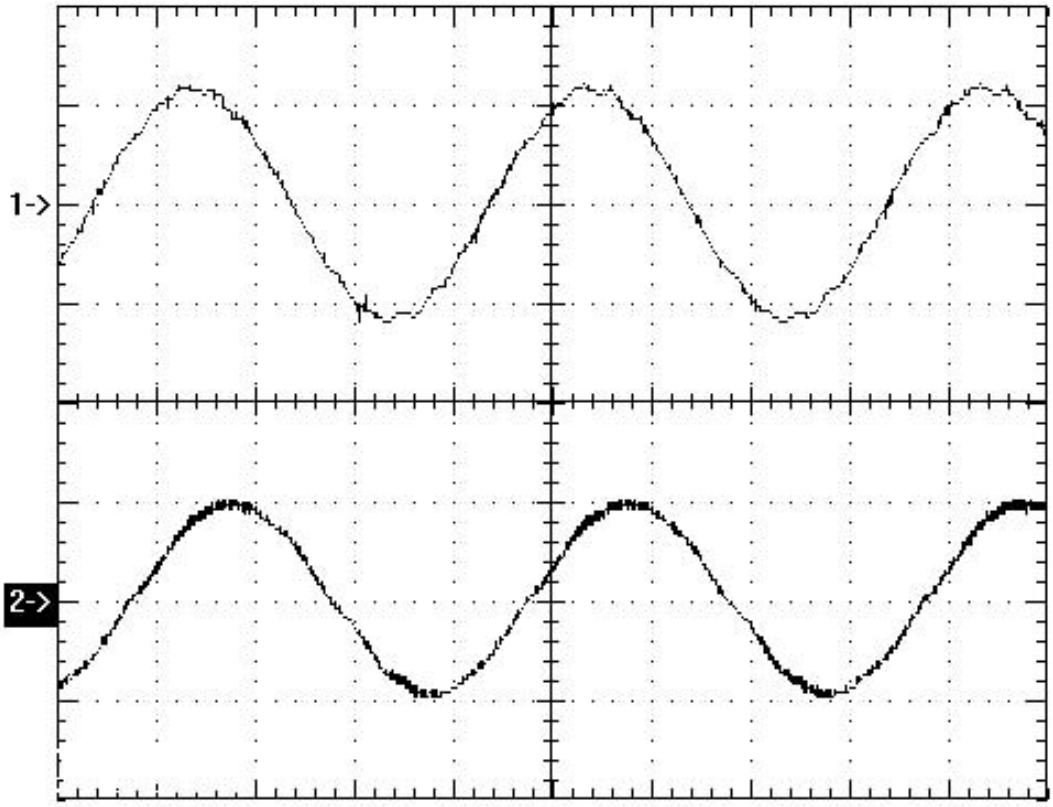


***Figure 5.13*** *1) Line to neutral voltage and 2) line current measured in a standalone grid with three converters.*

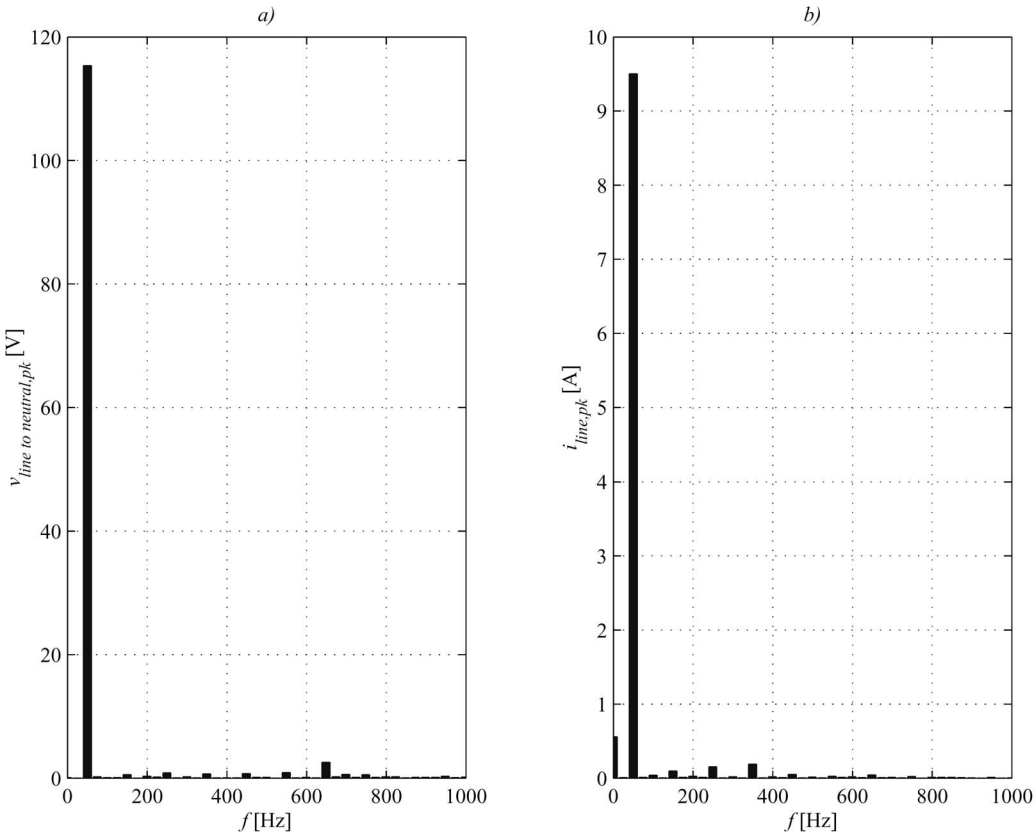


***Figure 5.14*** *Spectrum of a) line to neutral voltage and b) line current.*

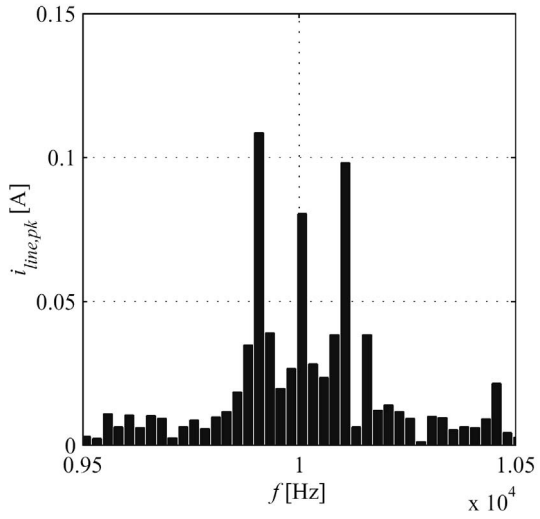


***Figure 5.15*** *Spectrum of line current centred on the switching frequency.*

When using converters, the increased level of harmonics may become a problem. The amount of harmonics as well as their spectrum depends on the type of converter applied and the type of modulation used. The PWM converter that is used generates harmonics around multiples of its switching frequency. Traditionally there have been no limits or recommendation to the harmonics at switching frequencies. Older thyristor converters generate $5^{th}$, $7^{th}$, $11^{th}$, $13^{th}$ (and so on) harmonics of the 50 Hz fundamental frequency. The total harmonic distortion (THD) of the current is usually high and above the standard limits without proper filtering. In this case with an output filter on the converter, with a cut-off frequency at 500 Hz, the multiples of the switching frequency are suppressed. This can be seen from Figure 5.15.

# 6 Conclusions

A control algorithm for parallel connected converters in a standalone grid was developed. The use of microsources connected together to a grid often needs power electronics to convert the produced DC to AC at the correct frequency and voltage. Communication is often impractical because the mobility of the sources is lost. To overcome this problem a droop controller is developed. The droop controller is controlling both frequency and voltage based on locally measured quantities. The voltage and current output from the converter is measured and the frequency is estimated from these measurements and the converter control signals. With droop control the converters take up load proportional to their nominal power without any communication. The controller is based on the fact that the frequency in the grid depends on the produced active power and the voltage depends on the reactive power.

The first case studied was a standalone microgrid with only converters. One converter was acting as load and the other two as sources. When a step in power was performed frequency and voltages decreased and the sources started to produce power. Load sharing was operating well and both frequency and voltage was stable and well within the limits. Disconnection of a load was also handled well by the converters.

Then the case with a synchronous generator, connected together with the converters, was investigated. The generator was droop controlled just like the converters. This case is more likely in a real power grid, which mostly contains rotating generators. The generator was keeping the frequency more stable, which means that the converter control becomes easier. This results in a good control and load sharing also with rotating generators.

A problem with droop control is when the distance between the sources is long and the impedance increases. Then the voltage is not the same in the entire grid and leads to an error in load sharing of reactive power. Controlling the active power is much easier due to the fact that the frequency is the same in the entire grid, when operating in steady state.

The line voltage and current appears to be almost sinusoidal according to the spectrums made, the harmonics have very low magnitudes, meaning the curves are almost sinusoidal.

# References


[1] M. Alaküla, "Power Electronic Control" *Department of Industrial Electrical Engineering and Automation, Lund Institute of Technology*, Lund, Sweden 2003.

[2] J. D. Glower, M. S. Sarma, "Power System – Analysis and Design" 3:rd edition, 2002.

[3] L. Harnefors, H. P. Nee, "A General Algorithm for Speed and Position Estimation of AC Motors", *IEEE trans. Power Electronics*, vol 47, no. 1, pp. 77-83, Feb. 2000.

[4] P. Karlsson, "Stability of Voltage and Frequency Control in Distributed Generation on Parallel-Connected Converters Feeding Constant Load" submitted to *EPE 2005*, Dresten, Germany, Sept. 12-14 2005.

[5] P. Karlsson, "Kraftelektronik", *Department of Industrial Electrical Engineering and Automation, Lund Institute of Technology*, Lund, Sweden 1998.

[6] P. Karlsson, "Modulation and Converter Control", *Department of Industrial Electrical Engineering and Automation, Lund Institute of Technology*, Lund, Sweden 2000.

[7] M.C. Chandorkar, D.M. Divan and R. Adapa, "Control of Parallel Connected Inverters in Standalone AC Supply Systems", *IEEE trans. Industry Applications*, vol. 29, no. 1, pp.136-143, Jan./Feb. 1993.

[8] R. Tirumla, N. Mohan and C. Henze, "Seamless Transfer of Grid-Connected PWM Inverters Between Utility - Interactive and Stand-Alone Modes", IEEE APEC 2002 Conf. Proc., Dallas, Texas, Mar. 10-14, 2002, pp. 1081-1086.

[9] Y. Li, M. Vilathgamuwa and P.C. Loh, "Design, analysis, and Real-Time Testing of a Controller for Multibus Microgrid System", *IEEE trans. Power Electronics*, vol. 19, no. 5 pp. 1195-1204, Sept. 2004.

[10] A. Engler, "Applicability of Droops in Low Voltage Grids", *Der Journal*, no. 1, Jan.2005.

[11] S. Wijnbergen and S.W.H. de Haan, "Power Electronic Interface with Independent Active and Reactive Power Control for Dispersed Generators to Support Grid Voltage and Frequency Stability", *EPE 2003 Conf. Proc*., Toulouse, France, Sept. 2-4, 2003, CD-ROM page 8.

[12] P. Karlsson and J. Svensson, "Voltage Control and Load Sharing in DC Distribution Systems", *EPE 2003 Conf. Proc*., Toulouse, France, Sept. 2-4, 2003, CD-ROM page 10.

[13] DS1104 R&D Controller Board, RTLib Reference, Release 3.4, *dSPACE GmbH*, May 2002.

# Appendix A – Vector Transformation

The transformations derived in this appendix are used when analyzing three phase systems dynamically. By using this transformation, the voltage and current instantaneous values can be transformed between three-phase system and $\alpha\beta$-frame, and, between $\alpha\beta$-frame and $dq$-frame. The transformations are power invariant.

### Transformations between three phase system and $\alpha\beta$-frame

The three-phase quantities, $v_1(t)$, $v_2(t)$ and $v_3(t)$ in a positive sequence system can be transformed into a vector $v_\alpha(t)+\mathrm{j}v_\beta(t)$ in the fixed two axis coordinate system, called $\alpha\beta$-frame. The vector in $\alpha\beta$-frame is defined by

$$\vec{v}(t)=v_\alpha(t)+jv_\beta(t)=K\left(v_1(t)+v_2(t)\cdot e^{j\frac{2\pi}{3}}+v_3(t)\cdot e^{j\frac{4\pi}{3}}\right) \tag{A.1}$$

where $K$ is a scaling. It is often selected equal to $\sqrt{2/3}$ to achieve power invariance. Also, the sum of the three-phase voltages will be zero when no conductor is connected to the neutral point of the three-phase system, i.e.,

$$v_1(t)+v_2(t)+v_3(t)=0 \tag{A.2}$$

Then Equation (A.1) can be expressed as a matrix equation

$$\begin{bmatrix} v_\alpha(t) \\ v_\beta(t) \end{bmatrix}=T\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} \tag{A.3}$$

and the inverse becomes

$$\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix}=T^{-1}\begin{bmatrix} v_\alpha(t) \\ v_\beta(t) \end{bmatrix} \tag{A.4}$$

where

$$T=\begin{bmatrix} \sqrt{\frac{3}{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \qquad T^{-1}=\begin{bmatrix} \sqrt{\frac{2}{3}} & 0 \\ -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \tag{A.5}$$

### Transformation between $\alpha\beta$-frame and the $dq$-frame

Let the vector $\vec{v}(t)$ rotate in the $\alpha\beta$-frame with the angular frequency $\omega(t)$ in a positive direction. Then if two perpendicular vectors, $dq$, rotating with the same angular velocity as $\vec{v}(t)$, are put into the $\alpha\beta$-frame, $\vec{v}(t)$ will appear as fixed in the $dq$-frame. The components in the $dq$-frame can be determined from Figure A.1.

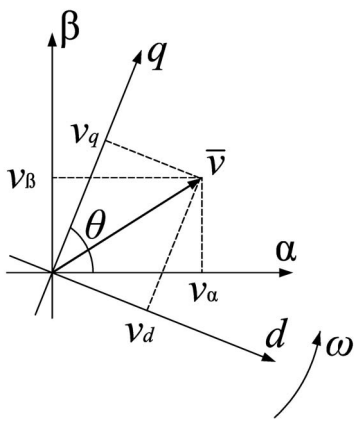


***Figure A.1*** *Relation between αβ-frame and the dq-frame.*

The transformation equation from the *αβ*-frame to the *dq*-frame becomes, in matrix form

$$\begin{bmatrix} v_d(t) \\ v_q(t) \end{bmatrix} = R_{dq\to\alpha\beta} \begin{bmatrix} v_\alpha(t) \\ v_\beta(t) \end{bmatrix} \tag{A.6}$$

and the inverse are

$$\begin{bmatrix} v_\alpha(t) \\ v_\beta(t) \end{bmatrix} = R_{\alpha\beta\to dq} \begin{bmatrix} v_d(t) \\ v_q(t) \end{bmatrix} \tag{A.7}$$

where the projections matrixes is given by

$$R_{dq\Rightarrow\alpha\beta} = \begin{bmatrix} \sin(\theta(t)) & -\cos(\theta(t)) \\ \cos(\theta(t)) & \sin(\theta(t)) \end{bmatrix} \qquad R_{\alpha\beta\Rightarrow dq} = \begin{bmatrix} \sin(\theta(t)) & \cos(\theta(t)) \\ -\cos(\theta(t)) & \sin(\theta(t)) \end{bmatrix} \tag{A.8}$$

# Appendix B – Controller C-code

```
/************************************************************************
*
* FILE:
*   regsys.c
*
* RELATED FILES:
*   Brtenv.h
*
************************************************************************/

#include <brtenv.h>                         /*basic real-time environment*/
#include <math.h>

/*----------------------------------------------------------------------*/

#define PI 3.141592654

/* variables for communication with Slave DSP */
Int16 task_id = 0;                          /* communication channel */
Int16 index = -1;                           /* slave DSP command index */

/* parameters for PWM initialization */
Float64 Ts = 1.0e-4;
Float64 deadband = 0.0;               /* deadband period */
Float64 sync_pos = 0.5;       /* position of the synch. interrupt signal */

/*cut of frequency*/
Float64 w1 = 2*PI*4;
Float64 w2 = 2*PI*1;
Float64 w3 = 2*PI*0.5;
Float64 ra_w = 2*PI*5.3558;
Float64 w4 = 2*PI*5;

/*constants*/
Float64 Eqn = 145;
Float64 wn = 2*PI*50;
Float64 delta_v = 0.04;
Float64 delta_w = 0.005;
Float64 Udc = 0;
Float64 K, Kc, Ti;

/*parameters accessed by ControlDesk*/
volatile Float64 Kw, Kv, P_ref, Q_ref, id_ref, iq_ref;

volatile Float64 R = 0.05;
volatile Float64 L = 0.005;
volatile Float64 C = 0.00002;
volatile Float64 v_adj = 258;
volatile Float64 imax = 10;
```

```
volatile Float64 Sn = 4500;
volatile Float64 K_adj = 0.2;
volatile Float64 enable = 0;
volatile Float64 aut = 1;
volatile Float64 P_ref_man = 100;
volatile Float64 Q_ref_man = 100;
volatile Float64 on = 0;
volatile Float64 station = 1;
volatile Float64 tid = 0;

/*Variables*/
Float64 duty1, duty2, duty3 ;

Float64 vq;
Float64 vqf1_old, vqf2_old, vqf3_old;
Float64 vmax, vmin;

Float64 w_hat_old, theta_hat_old, wf4_old;


Float64 sum_id_err, sum_iq_err;
Float64 id_ref_old, iq_ref_old;
Float64 ud_ref_old;

Float64 ialfa, ibeta;
Float64 id, iq;

Float64 vqf1, vqf2, vqf3;
Float64 w_hat, theta_hat, wf4;
Float64 w_adj = 0;

Float64 ud_ref, uq_ref, ualfa_ref, ubeta_ref, ua_ref, ub_ref, uc_ref;

Float64 valfa, vbeta, va, vb ,vc;

Float64 ia, ib, ic;
Float64 P, Q;
Float64 sym;

Float64 vq_ref, w_ref;
Float64 stopp = 1;

Float64 sample = 0;
Float64 error_type  = 0;
Float64 off = 1;
Float64 converter_on = 0;
Float64 fas = 0;
Float64 count = 0;
Float64 error = 0;
Float64 trigg1 = 0;
Float64 trigg2 = 0;
```

```
Float64 max, min, uz_ref, duty1_temp, duty2_temp, duty3_temp;


Float64 theta;
Float64 ud_I, uq_I, ud_I_temp, uq_I_temp, umax, umin;
Float64 Udc_test;
UInt16 scantable[4] = {1, 2, 3, 4};
Float64 u[4];

Float64 exec_time;                                    /* execution time */
/*-----------------------------------------------------------------------*/

/* interrupt service routine for PWM sync interrupt */

void PWM_sync_interrupt(void)
{

  /*---------------------------------On/Off--------------------------------*/
  if(Udc>250 && station==1 && off==1 && enable==1)
    {
      aut = 1;
      on = 1;
    }
  else if(off==1 && enable==1)
    {
      aut = 2;
    }

  if(on==1 && off==1 && enable==1)            //On
  {
    duty1 = 0.5;
    duty2 = 0.5;
    duty3 = 0.5;
    id_ref_old = 0;
    iq_ref_old = 0;
    ud_ref_old = 0;
    ud_I = 0;
    uq_I = 0;

    if(station==1)
    {
      vqf1_old = Eqn;
      vqf2_old = Eqn;
      vqf3_old = Eqn;
      w_hat_old = 2*PI*50;
      theta_hat_old = 0;
      wf4_old = 2*PI*50;
    }

    ds1104_bit_io_set(DS1104_DIO11);
    sample=0;
```

```
  tid = 0;
  error_type = 0;
  off = 0;
  converter_on = 1;
}
else if(on==0 && off==0)                    //Off
{
  ds1104_bit_io_clear(DS1104_DIO11);
  off = 1;
  converter_on = 0;
}

if(converter_on ==1)
{
  sample = sample+1;
}

if(station!=1 && sample == 5000)
{
  aut=1;
}

/*---------------------------------Startup sequence--------------------*/
if(sample < 10000)     //Change limits after one second
{
  imax = 20;
  vmax = 300;
  vmin = 0;
  K_adj = 0.20;
}
else if(sample == 10000)
{
  imax = 20;
  vmax = 250;
  vmin = 130;
  K_adj = 0.20;
}

/*-------------------------------------------------------------------*/
host_service(1, 0);                          /* Data Acquisition service */
RTLIB_TIC_START();                           /* start time measurement */

/* write PWM Duty cycle to slave DSP and test for error */
ds1104_slave_dsp_pwm3_duty_write(task_id, index, duty1, duty2, duty3);

/* activate the previously written DAC values synchronously */
ds1104_dac_strobe();

/*--------------------------------Read from ADC-----------------------*/
ds1104_adc_start(DS1104_ADC2|DS1104_ADC3|DS1104_ADC4|DS1104_ADC5);
```

```
ds1104_adc_read_mux(scantable, 4, u);
Udc = u[3];

ia = ds1104_adc_read_conv(2);
ib = ds1104_adc_read_conv(3);

valfa = ds1104_adc_read_conv(4);
vbeta = ds1104_adc_read_conv(5);

//signalanpassning
ia = ia*25;
ib = ib*25;

valfa = valfa*v_adj;
vbeta = vbeta*v_adj;

Udc = Udc*408;

/*---------------------------------Calculation of vq--------------------*/
vq = sqrt((valfa*valfa)+(vbeta*vbeta));       //voltage over capacitor

/*---------------------------------Vectors to abc----------------------*/
va = sqrt(0.66666667)*valfa;
vb = -sqrt(0.16666667)*valfa + sqrt(0.5)*vbeta;
vc = -sqrt(0.16666667)*valfa - sqrt(0.5)*vbeta;

/*---------------------------------Calculation of i_alfa,beta----------*/
ic = -ia-ib;
sym = (ia+ib+ic)/3;
ia = ia-sym;
ib = ib-sym;
ic = ic-sym;
ialfa = sqrt(3/2)*ia;
ibeta = sqrt(0.5)*(ib-ic);

/*---------------------------------ab to dq----------------------------*/
id = ialfa*sin(theta_hat_old) - ibeta*cos(theta_hat_old);
iq = ialfa*cos(theta_hat_old) + ibeta*sin(theta_hat_old);
//id = ialfa*cos(theta_hat_old) + ibeta*sin(theta_hat_old);
//iq = ialfa*sin(theta_hat_old) - ibeta*cos(theta_hat_old);

/*---------------------------------Over current protection-------------*/
if(id>imax || iq>imax || id<(-imax) || iq<(-imax))
{
  enable = 0;
  error_type = 1;
}

/*---------------------------------Under voltage protection Udc--------*/
if(Udc<200)
{
```

```
    enable = 0;
    error_type = 4;
  }

  /*---------------------------------Filtering--------------------------*/
  //LP-filter1 4Hz
  vqf1 = ((vq-vqf1_old)*w1*Ts)+vqf1_old;

  //LP-filter2 1Hz
  vqf2 = ((vq-vqf2_old)*w2*Ts)+vqf2_old;

  //LP-filter3 0.5Hz
  vqf3 = ((vq-vqf3_old)*w3*Ts)+vqf3_old;



  /*---------------------------------Over voltage protection-------------*/
  if(vq>vmax)
  {
    enable = 0;
    error_type = 2;
  }

  /*---------------------------------Under voltage protection------------*/
  if(vqf2<vmin && converter_on ==1)
  {
    enable = 0;
    error_type = 3;
  }

  /*---------------------------------Estimator--------------------------*/
  w_hat = w_hat_old + ((ra_w*ra_w*Ts)*(-ud_ref_old+(R*id_ref_old)- ↵
(w_hat_old*L*iq_ref_old)))/Eqn;
  theta_hat = theta_hat_old + w_hat_old*Ts + ((2*ra_w*Ts)* ↵
(-ud_ref_old+(R*id_ref_old)-(w_hat_old*L*iq_ref_old)))/Eqn;

  if(w_hat<(2*PI*49))
    w_hat = 2*PI*49;
  if(w_hat>(2*PI*51))
    w_hat = 2*PI*51;

  if(theta_hat_old>(2*PI))
  {
    theta_hat = theta_hat - (2*PI);
    theta_hat_old = theta_hat_old - (2*PI);
  }

  if(converter_on ==0)
  {
    w_hat = (2*PI*50)+w_adj;
  }
```

```
//LP-filter4 5Hz
wf4 = ((w_hat-wf4_old)*w4*Ts)+wf4_old;

/*---------------------------------Droops-----------------------------*/
if(aut==1)
{
//Slow droops
vq_ref = vqf2_old + (Kw/Kv)*(wn-wf4_old);
w_ref = wf4_old - (Kv/Kw)*(Eqn-vqf2_old);

//Fast droops
P_ref = Kv*(vq_ref-vqf1);
Q_ref = -Kw*(w_ref-w_hat_old);
}

//Idling
else if(aut==2)
{
  P_ref = 0;
  Q_ref = 0;

  id_ref_old = 0;
  iq_ref_old = 0;
  ud_I = 0;
  uq_I = 0;
}

//Manual reference values
else
{
  tid = tid + 1;

  P_ref = P_ref_man;

  if(tid > 10000)     //1 second
  {
    Q_ref = Q_ref_man;
  }
  else
  {
    Q_ref = 0;
 }

  if(tid == 21000)    //2.1 seconds
  {
    aut = 2;
  }
}
/*---------------------------------P,Q to idq-------------------------*/
id_ref = Q_ref/vqf3_old;
```

```
iq_ref = P_ref/vqf3_old;

//Current limitation
if(id_ref > imax)
  id_ref = imax;

if(iq_ref > imax)
  iq_ref = imax;

if(id_ref < (-imax))
  id_ref = -imax;

if(iq_ref < (-imax))
  iq_ref = -imax;

/*--------------------------------Current controller------------------*/
ud_I_temp = K_adj*K*(Ts/Ti)*(id_ref-id) + ud_I;
uq_I_temp = K_adj*K*(Ts/Ti)*(iq_ref-iq) + uq_I;

ud_ref = K_adj*K*(id_ref-id) + ud_I - (Kc*0.5*(iq_ref+iq));
uq_ref = K_adj*K*(iq_ref-iq) + uq_I + (Kc*0.5*(id_ref+id)) + vqf1;

//Anti-windup
if(ud_ref>umax)
{
  ud_ref = umax;
  ud_I = umax - (K_adj*K*(id_ref-id));
}
else if(ud_ref<umin)
{
  ud_ref = umin;
  ud_I = umin - (K_adj*K*(id_ref-id));
}
else
{
  ud_I = ud_I_temp;
}

if(uq_ref>umax)
{
  uq_ref = umax;
  uq_I = umax - (K_adj*K*(iq_ref-iq));
}
else if(uq_ref<umin)
{
  uq_ref = umin;
  uq_I = umin - (K_adj*K*(iq_ref-iq));
}
else
{
  uq_I = uq_I_temp;
```

```
}

/*---------------------------------dq to ab---------------------------*/
ualfa_ref = ud_ref*sin(theta_hat_old) + uq_ref*cos(theta_hat_old);
ubeta_ref = -ud_ref*cos(theta_hat_old) + uq_ref*sin(theta_hat_old);
//ualfa_ref = ud_ref*cos(theta_hat_old) - uq_ref*sin(theta_hat_old);
//ubeta_ref = ud_ref*sin(theta_hat_old) + uq_ref*cos(theta_hat_old);

/*---------------------------------Vectors to abc---------------------*/
ua_ref = sqrt(0.66666667)*ualfa_ref;
ub_ref = -sqrt(0.16666667)*ualfa_ref + sqrt(0.5)*ubeta_ref;
uc_ref = -sqrt(0.16666667)*ualfa_ref - sqrt(0.5)*ubeta_ref;

/*---------------------------------symmetry---------------------------*/
duty1_temp = (ua_ref/Udc)+0.5;
duty2_temp = (ub_ref/Udc)+0.5;
duty3_temp = (uc_ref/Udc)+0.5;

max=ua_ref;
if(ub_ref>max)
  max=ub_ref;
if(uc_ref>max)
  max=uc_ref;

min=ua_ref;
if(ub_ref<min)
  min=ub_ref;
if(uc_ref<min)
  min=uc_ref;

uz_ref=(max+min)/2;

ua_ref = ua_ref-uz_ref;
ub_ref = ub_ref-uz_ref;
uc_ref = uc_ref-uz_ref;

/*---------------------------------PWM duty cycle---------------------*/
duty1 = (ua_ref/Udc)+0.5;
duty2 = (ub_ref/Udc)+0.5;
duty3 = (uc_ref/Udc)+0.5;


/*---------------------------------Power calculation------------------*/
P = iq*vqf1;
Q = (id*vqf1);

/*---------------------------------DAC--------------------------------*/
ds1104_dac_write(1, (valfa/300));
ds1104_dac_write(2, (ualfa_ref/300));

/*---------------------------------Synchronization--------------------*/
```

```
if(fas==0 && station!=1)
{
  if(va>20)
  {
    trigg1 = 1;
  }
  if(ua_ref>20 && trigg1==1)
  {
    trigg2 = 1;
  }

  //va zero crossing
  if(trigg1==1 && va<10 && va>-10 || count!=0)
  {
    count = count+1;
  }

  //ua_ref zero crossing
  if(trigg2==1 && ua_ref<10 && ua_ref>-10 && count!=0)        {
    error = count;
    count = 0;
  }
  if(error!=0)
  {
    //start switching if the error is less than 3.6 degrees
    if(((error*Ts)/20e-3)<0.01)
    {
      fas = 1;
      on = 1;
    }
    else
    {
      w_adj = ((2*PI*error*Ts)/20e-3);
      error = 0;
      trigg1 = 0;
      trigg2 = 0;
    }
  }

}

/*---------------------------------Update variables--------------------*/
vqf1_old = vqf1;
vqf2_old = vqf2;
vqf3_old = vqf3;
w_hat_old = w_hat;
theta_hat_old = theta_hat;
wf4_old = wf4;
ud_ref_old = ud_ref;
id_ref_old = id_ref;
iq_ref_old = iq_ref;
```

```
  exec_time = RTLIB_TIC_READ();
}

/*-----------------------------------Main-------------------------------*/
void main(void)
{
  init();                          /* DS1104 and RTLib1104 initialization */

  /*initiering av DIO*/
  ds1104_bit_io_init(DS1104_DIO11_OUT);     //bit_IO kanal 11 out

  /* initialization of slave DSP communication */
  ds1104_slave_dsp_communication_init();

  /* init and start of 3-phase PWM generation on slave DSP */
  ds1104_slave_dsp_pwm3_init(task_id, Ts,
                             duty1, duty2, duty3,
                             deadband, sync_pos);

  //ds1104_slave_dsp_pwm3_start(task_id);

  /* registration of PWM duty cycle update command */
  ds1104_slave_dsp_pwm3_duty_write_register(task_id, &index);

  /* init D/A converter in latched mode */
  ds1104_dac_init(DS1104_DACMODE_LATCHED);

  /* initialization of PWM sync interrupt */
  ds1104_set_interrupt_vector(DS1104_INT_SLAVE_DSP_PWM,
                              (DS1104_Int_Handler_Type)
                              &PWM_sync_interrupt,
                              SAVE_REGS_ON);

  ds1104_enable_hardware_int(DS1104_INT_SLAVE_DSP_PWM);
  RTLIB_INT_ENABLE();

  /*------------------------------------------------------------------*/

  /* Background tasks */
  while(1)
  {
    if(enable==0 && stopp==0)               //stop
    {
      ds1104_bit_io_clear(DS1104_DIO11);

      ds1104_slave_dsp_pwm3_stop(task_id);
      stopp=1;
      ds1104_dac_write(1, stopp);
      ds1104_dac_strobe();
      converter_on = 0;
```

```
  fas=0;
  count = 0;
  error = 0;
  w_adj = 0;

}
else if(enable==1 && stopp==1)          //start
{
  /*-------------Constants---------------*/
  Kw = Sn/(wn*delta_w);
  Kv = Sn/(Eqn*delta_v);

  K = (L/Ts)+(R/2);
  Ti = ((L/R) + (Ts/2));
  //Ti=0;
  Kc = wn*L;


  //imax = 1.2*(Sn/Eqn); //OBS OBS OBS OBS OBS

  /*-------------Initial values----------*/
  duty1 = 0.5;
  duty2 = 0.5;
  duty3 = 0.5;
  vqf1_old = Eqn;
  vqf2_old = Eqn;
  vqf3_old = Eqn;

  w_hat_old = 2*PI*50;
  theta_hat_old = 0;
  wf4_old = 2*PI*50;

  id_ref_old = 0;
  iq_ref_old = 0;
  ud_ref_old = 0;

  ud_I = 0;
  uq_I = 0;

  umax = 0.7*Udc;
  umin = -0.7*Udc;

  on = 0;
  stopp=0;
  sample=0;
  error_type = 0;
  fas = 0;
  count = 0;
  error = 0;
  w_adj = 0;
  trigg1 = 0;
```

```
      trigg2 = 0;

      ds1104_slave_dsp_pwm3_start(task_id);
    }

    RTLIB_BACKGROUND_SERVICE();                    /* background service */
  }
}
```