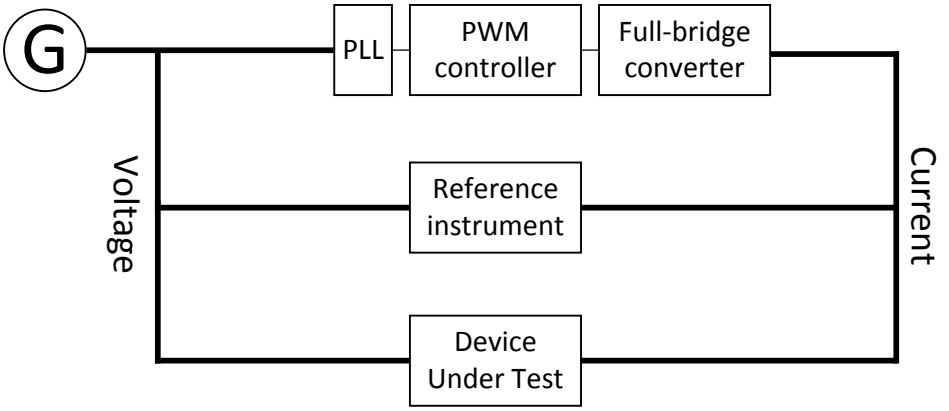
CHALMERS



Construction of a low-ripple inverter with accurate phase control for calibration of measurement equipment

Master of Science Thesis in Electric Power Engineering

ADAM EINARSSON
ALI ASGARI VAND

Department of Energy and Environment
Division of Electric Power Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2014

Construction of a low-ripple inverter with accurate phase control for calibration of measurement equipment

ADAM EINARSSON
ALI ASGARI VAND

Department of Energy and Environment
Division of Electric Power Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2014

Construction of a low-ripple inverter with accurate phase control for calibration of measurement equipment
ADAM EINARSSON
ALI ASGARI VAND

Department of Energy and Environment
Division of Electric Power Engineering
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone +46 (0)31–772 1000

Cover:
Overview of a calibration system with a phase-locked loop (PLL) for accurate phase control between current and voltage branch.

Chalmers Bibliotek, Reproservice
Göteborg, Sweden 2014

Construction of a low-ripple inverter with accurate phase control for calibration of measurement equipment

ADAM EINARSSON
ALI ASGARI VAND
Department of Energy and Environment
Division of Electric Power Engineering
Chalmers University of Technology

Abstract

For the design of an electric power system, knowledge of the parameters of all the components is essential. Especially the transformers are an important part of this. It is therefore important that the measurement equipment for determining these values is accurate. In no-load operation, the transformer mainly consumes reactive power. A small error in the measurement of the power angle therefore gives a large error in the active power calculation. The Technical Research Institute of Sweden, SP, performs calibrations of these measurement systems. Calibrations are done by measuring a reference voltage and generating a current in a secondary circuit with a controllable phase shift. The voltage and current make up a virtual power and are measured by the reference instrument and by the equipment to be calibrated.

This thesis describes the work of constructing a single-phase full-bridge converter for inverter operation with accurate phase control of the current in the secondary circuit relative to the sinusoidal reference voltage. A CompactRIO™ system from National Instruments is used to control the converter through pulse-width modulation (PWM). An analogue-to-digital converter (ADC) module is used to sample the reference signal and a digital I/O module outputs the PWM signals. A phase-locked loop (PLL) algorithm estimates the phase of the sampled reference voltage to use in the generation of the PWM signals. Three different PLLs have been implemented with LabVIEW™ code on the FPGA chip of the CompactRIO™ system. The three PLLs are the inverse Park PLL (IP-PLL) based on the synchronous reference frame, the enhanced PLL (E-PLL) based on the gradient descent method and the KF-PLL based on the Kalman filter. Both the E-PLL and the KF-PLL are also based on models for estimating specific signal components. With the basic configurations, the IP-PLL shows superior performance in simulations with dc offset and low-order harmonic in the input signal. This is due to its low-pass filters. While the standard deviation in the phase error of the IP-PLL was 50 μrad it was 350 μrad for the E-PLL and 570 μrad for the KF-PLL with a dc offset at 1.8 % of the fundamental amplitude in the input signal. With a third harmonic at 10 % of the fundamental amplitude, the standard deviations of the phase errors were 50.3 μrad, 776 μrad and 889 μrad for the IP-PLL, the E-PLL and the KF-PLL respectively. By extending the E-PLL and the KF-PLL to include estimation of dc-offset and harmonics, these steady-state errors are completely eliminated. However, especially the KF-PLL but also the E-PLL, is much more complex and harder to tune than the IP-PLL. When implemented on CompactRIO™ the KF-PLL including estimation of dc offset got the best results with a standard deviation in the phase error of 5.1 μrad compared to 49.2 μrad for the IP-PLL with minimized cut-off frequency

of the low-pass filters for a dc offset of 0.41 % of the fundamental amplitude in the input signal.

It is shown that the phase error of the complete system can be kept below a standard deviation of 600 μrad, with a stable reference voltage. The pattern of the phase error however has an oscillatory shape with a frequency of about 0.12 Hz. This is thought to originate from a drift in the clock of the ADC module relative to the clock of the FPGA in the CompactRIO™ system. This must be further investigated. Additionally, the amplitude accuracy of the fundamental frequency is found to have a standard deviation of about 63 ppm, with a dc-source specified to be accurate to 50 ppm.

Index terms: PWM, full-bridge converter, IGBT, CompactRIO™, Kalman filter, PLL.

Acknowledgements

This work has been carried out at the Department of Energy and Environment at Chalmers University of Technology in Göteborg and The Technical Research Institute of Sweden, SP, in Borås from January to June of 2014. We would like to thank Anders Lindskog, Alf-Peter Elg, Stefan Svensson and Allan Bergman at SP for helping out with measurements, supply of equipment, help with LabVIEW™ code and discussion about the problem. Many thanks are directed towards Robert Karlsson at Chalmers for indispensable help with the practical issues of constructing a converter. We are also thankful towards our examiner at Chalmers, Stefan Lundberg, for giving good feedback on the work and the report and for good discussions about problems.

Adam Einarsson, Ali Asgari Vand
Göteborg, Sweden, 2014

Contents

Abstract iii

Acknowledgements v

Contents vii

List of Abbreviations ix

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Previous work . . . 2
- 1.3 Purpose/Aim . . . 3

2 Technical background 5
- 2.1 Phase tracking . . . 5
  - 2.1.1 The general PLL algorithm . . . 5
  - 2.1.2 The synchronous reference frame PLL . . . 6
  - 2.1.3 The enhanced PLL . . . 9
  - 2.1.4 The Kalman filter . . . 11
- 2.2 Single-phase full-bridge converter . . . 16
  - 2.2.1 Pulse-width modulation . . . 16
- 2.3 The insulated-gate bipolar transistor . . . 20
  - 2.3.1 Conduction losses . . . 20
  - 2.3.2 Switching losses . . . 22
  - 2.3.3 Thermal considerations . . . 24
  - 2.3.4 Gate driver . . . 25
- 2.4 CompactRIO™ . . . 26
  - 2.4.1 The field programmable gate array . . . 26
  - 2.4.2 LabVIEW™ . . . 27

3 Simulations 29
- 3.1 Phase tracking . . . 29
  - 3.1.1 The synchronous reference frame PLL . . . 29
  - 3.1.2 The enhanced PLL . . . 34
  - 3.1.3 The Kalman filter PLL . . . 34
  - 3.1.4 Comparison . . . 39
- 3.2 Inverter . . . 47

3.2.1 Load measurements . . . 47
3.2.2 Simulations . . . 50

4 Design **59**
4.1 Real time control system implementation . . . 59
4.1.1 User interface . . . 61
4.1.2 FPGA implementation . . . 61
4.2 Inverter . . . 64
4.2.1 Selection of dc-link capacitors . . . 66
4.2.2 Selection of IGBT module . . . 67
4.2.3 Selection of gate driver . . . 69
4.2.4 Selection of heat sink . . . 71
4.2.5 CompactRIO™-to-gate driver interface . . . 72
4.2.6 Assembly and gate driver settings . . . 72

5 Tests and measurements **75**
5.1 Phase tracking of the implemented PLLs on CompactRIO™ . . . 75
5.1.1 The inverse Park PLL . . . 75
5.1.2 The enhanced PLL . . . 77
5.1.3 The Kalman filter PLL . . . 77
5.2 Inverter . . . 80
5.2.1 Gate driver . . . 80
5.2.2 Switching induced voltage spikes . . . 83
5.2.3 Losses . . . 83
5.2.4 Harmonics attenuation . . . 87
5.3 Phase and amplitude jitter . . . 87

6 Conclusions and future work **93**
6.1 Conclusions . . . 93
6.2 Future work . . . 95

References **97**

A MATLAB® code **103**
A.1 The Kalman filter . . . 103

B Kalman filter gain factor coefficients **105**

C LabVIEW™ code **107**
C.1 RT controller . . . 107
C.2 Host computer . . . 107
C.3 FPGA . . . 108
C.3.1 PWM controller . . . 111
C.3.2 The inverse Park PLL . . . 112
C.3.3 The enhanced PLL . . . 113
C.3.4 The Kalman filter PLL . . . 113

List of Abbreviations

ADC	*Analogue-to-digital converter* - Device that samples an analogue signal for use in a digital system.
DSWM	*Digital Sampling Watt Meter* - Measurement instrument with highly accurate synchronized sampling of multiple channels. Used as reference instrument in the calibration system.
DUT	*Device Under Test* - Measurement equipment that is calibrated with the calibration system.
E-PLL	*Enhanced PLL* - PLL algorithm based on the gradient descent method with separate branches for estimation of phase and amplitude.
ESR	*Equivalent series resistance* - Resistance in series with capacitance of a capacitor.
FIFO	*First In First Out* - Memory buffer where the last read data value is the first that was written to the buffer. Used in the CompactRIO™ system for transferring data from the FPGA to the RT controller.
FPGA	*Field programmable gate array* - Digital integrated circuit with reconfigurable gate logic. Good for parallel execution. Used in the CompactRIO™ system for optimally synchronized tasks.
IGBT	*Insulated-gate bipolar transistor* - Transistor that is common in high power applications.
IP-PLL	*Inverse Park PLL* - Single-phase SRF-PLL with OSG method based on the inverse Park transformation.
KF	*Kalman filter* - Model based, recursive algorithm for estimation of measured process subjected to noise.
KF-PLL	*Kalman filter PLL* - Kalman filter used for phase estimation with a PLL on the output for frequency adaptive operation.
OSG	*Orthogonal signal generation* - Method for generating an orthogonal system from a single-phase source for use with the SRF-PLL.
PLL	*Phase-locked loop* - Device used for synchronization between separate systems. Can be used for phase and frequency estimation.
PWM	*Pulse-width modulation* - Technique for control of transistor switches in converter where a reference waveform is compared to a triangular carrier wave.

RT *Real-time* - Controller system of CompactRIO™ used as interface between the FPGA and the host computer.

SNR *Signal-to-noise ratio* - Measure of noise in a signal. Calculated as the logarithm of the ratio between the power of the desired signal and the unwanted noise for expression in decibels.

SRF-PLL *Synchronous reference frame PLL* - PLL algorithm based on the Park transformation. Most common in three-phase systems but can also be used in single-phase systems with OSG.

TD-PLL *Transport delay PLL* - Single-phase SRF-PLL with OSG method based on delay of the input signal.

THD *Total harmonic distortion* - Measure of the harmonics content of a waveform. Calculated as the ratio between the RMS values of the fundamental component and all higher-order harmonics.

VI *Virtual Instrument* - A set of LabVIEW™ code. Contains a block diagram with related front panel. Can be divided in to subVIs with their own front panels.

Chapter 1

Introduction

1.1 Background

The production and consumption of electric power in today's society is larger than ever and is expected to grow significantly in the coming years. Especially renewable energy sources contribute to this [1] [2]. The electric power from renewables is harvested from e.g. water, solar or wind energy and transported to the customer via transmission and distribution networks. One of the key components in these networks is the transformer. Knowing the electric parameters of the transformer is thus essential for the designer of a power system network.

Even when the transformer is not transmitting any active power to the consumers it still consumes both active and reactive power of its own. The reactive power is much larger than the active power in no-load situations, meaning that the power factor is low and the phase shift between the current and voltage is large. Thus, a slight change in the phase shift results in a small change in reactive power but a large change in the active power consumption. It is therefore important that measurements of these parameters are performed with high accuracy.

The Technical Research Institute of Sweden, SP, is performing calibration of measurement equipment for different companies. Such equipment is e.g. used by manufacturers of transformers to find out no-load losses among other parameters. At calibration, the Device Under Test (DUT) and the reference instrument are both subjected to the same voltage and current. The current is generated with different phase shift, with respect to the voltage, to represent different power factors of transformers. For good calibration, amplitude, frequency and power factor must be very stable, as small variations may give large errors in power estimates.

Often the calibrations are performed at the customer's site, using their voltage source as reference to the calibration system. One problem is that this voltage may be noisy with respect to phase, amplitude and frequency. SP wishes to further enhance an existing calibration system to better withstand these problems and to make it easier to work with. This system should be both very accurate and portable for proper and easy use.

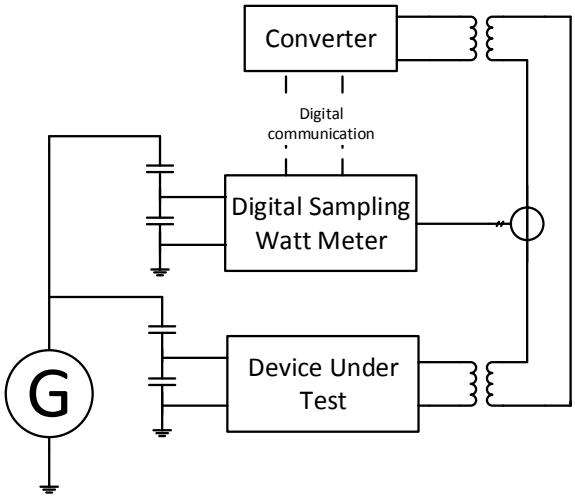


Figure 1.1 Overview of the previous calibration system setup.

1.2 Previous work

The previous calibration system as set up for calibration is shown in Figure 1.1 [3]. The system consists of the Digital Sampling Watt Meter (DSWM) which is the reference instrument, the converter and voltage and current transformers. A generated voltage is coupled to the DUT and to the DSWM in one circuit. This voltage acts as a reference for the generation of the current by the converter. This current is measured by both devices in a second circuit, thus is a virtual power measured.

During a calibration sequence, the phase shift of the current in the second circuit is locked relative to the reference voltage. The virtual power is measured by both the DSWM and the DUT and the results are saved. The duration may vary. The data of the DSWM is then analyzed to see if the current and voltage was stable during the measurement sequence. The DUT can then be calibrated based on the measurement results compared to those of the DSWM.

The converter is controlled through pulse-width modulation (PWM). The phase control is done via pulses sent by the DSWM each period of the reference for triggering a new sine wave with a set phase delay. The frequency of this wave is at a fixed value of 50 or 60 Hz. The trigger pulses are generated by a phase-locked loop (PLL) circuit. The phase delay is set in steps of 1.4 mrad with an accuracy of approximately 79 μrad. The ratings of the converter are 600 V and 75 A. The output is connected to a transformer that handles 250 V and 21.6 A on the primary side and 1.8 V and 3 kA on its secondary side. There is no feedback system regulating the output of the PWM converter.

The system has been tested both in lab and at a customer's site. With a stable voltage source, the phase angle control was found to be accurate down to a standard deviation of 262 μrad with currents of a few hundred amperes. With the grid voltage as reference, the stability of the phase angle control was ten times worse. At currents close to 3 kA the phase accuracy decreased but the current magnitude was accurate to below 0.1 % or 1000 ppm. At the customer's site a rotating machine with very stable output was used as a voltage source. The results were slightly worse than the lab tests with 349 and 698 μrad accuracy at 20 and 40 kV respectively. The inaccuracy of the phase angle control comes from jitter in the PLL circuit and the clock frequency which limits the timing accuracy of the PWM

pattern [3].

1.3 Purpose/Aim

The aim of this thesis work is to construct a single-phase current converter with a phase that is controlled relative to a reference voltage. The phase should be accurate within a standard deviation of 20-50 μrad. The magnitude should be up to 2 kA with a maximum standard deviation of 20-50 ppm. The control system should be designed on a CompactRIO™ system from National Instruments.

Chapter 2

Technical background

2.1 Phase tracking

To have a stable phase shift between the generated current and the reference voltage, the phase of the voltage must be known. A phase tracking system must therefore be used in between the measurement of the voltage and the generation of the converter control signals.

2.1.1 The general PLL algorithm

Phase-tracking is commonly performed with a phase-locked loop (PLL). A PLL is a device used for synchronization between separate systems. The general PLL algorithm is shown in Figure 2.1 [4]. The phase detector block produces an output proportional to the phase error between the input and the estimated output signals. It is the variation in the construction of this block that is the most common difference between different PLL algorithms [5]. The phase error may vary quickly due to actual or measurement disturbances. These perturbations are rejected by the low-pass filter, also called loop filter. The voltage controlled oscillator (VCO) then generates an output sinusoidal proportional to the filtered input which is fed back to the second input of the phase detector. When the phase-error is zero, so is the output of the low-pass filter and the VCO operates at its center frequency. In steady state the PLL is referred to as being in synchronized or locked mode [4] [6].

The standard implementation of this algorithm is to have a multiplier as phase detector and an integrator as the VCO. If the input and output are at the nominal frequency, the

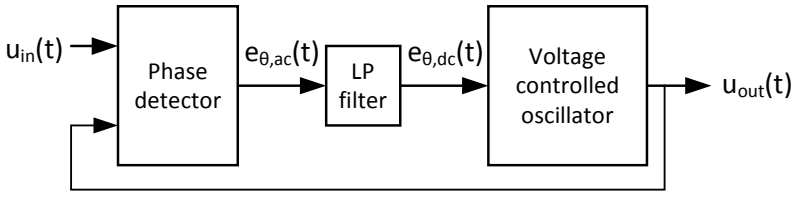


Figure 2.1 Block diagram of the general PLL algorithm.

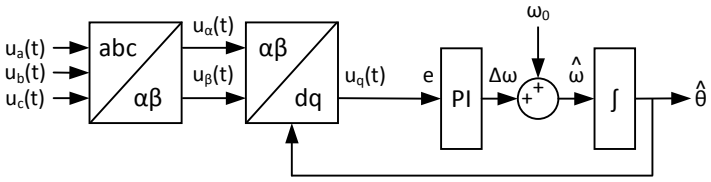


Figure 2.2 Block diagram of a synchronous reference frame PLL (SRF-PLL) circuit.

input to the low-pass filter will be [6]

$$\begin{aligned} e_{\theta,ac}(t) &= \sin(\omega_0 t + \theta_{in}) \sin(\omega_0 t + \theta_{out}) \\ &= \frac{1}{2}\Big(\cos(\theta_{in} - \theta_{out}) - \cos(2\omega_0 t + \theta_{in} + \theta_{out}) \Big) \end{aligned} \tag{2.1}$$

The double-frequency ripple is the main draw-back of this simple structure. Even if the phase-lock is exact, it will exist. However, for systems where fast response is not required, this can be practically solved by a narrow bandwidth of the low-pass filter [7].

2.1.2 The synchronous reference frame PLL

For the control of power converters connected to three-phase systems the synchronous reference frame PLL (SRF-PLL) is the most popular design [5]. A block diagram of this circuit is seen in Figure 2.2. The origin of the name is found in the first block where the three-phase system is transformed to the synchronous reference frame via the Clarke transformation. This step is defined as

$$\begin{bmatrix} u_\alpha(t) \\ u_\beta(t) \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} u_a(t) \\ u_b(t) \\ u_c(t) \end{bmatrix} \tag{2.2}$$

This is followed by the Park transformation,

$$\begin{bmatrix} u_d(t) \\ u_q(t) \end{bmatrix} = \begin{bmatrix} \cos\hat{\theta} & \sin\hat{\theta} \\ -\sin\hat{\theta} & \cos\hat{\theta} \end{bmatrix} \begin{bmatrix} u_\alpha(t) \\ u_\beta(t) \end{bmatrix} \tag{2.3}$$

where $\hat{\theta}$ is the estimated angle feedback [5].

In comparison with the general PLL algorithm; the Park transformation corresponds to the phase detector, the proportional and integral (PI) controller to the low-pass filter and the single integrator together with the added nominal frequency to the VCO. The input to the PI-controller may alternatively to the q-component be the angle or the d-component. This depends on the application of the estimated phase. However, at least one of these should be zero in locked mode.

In the Park transformation, the estimated angle is subtracted from the true angle. If the angles are equal, the output q-component is zero. For normal operation, assuming unity voltage amplitude, the block diagram of Figure 2.2 may be linearized into that of Figure 2.3. This is true since for small estimation errors [8]

$$\begin{aligned} sin(\theta - \hat{\theta}) &\approx \theta - \hat{\theta} \\ cos(\theta - \hat{\theta}) &\approx 1 \end{aligned} \tag{2.4}$$

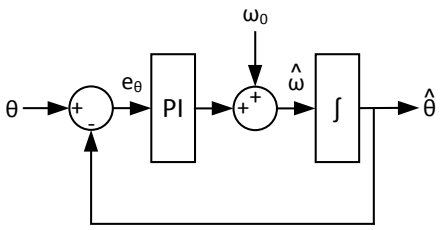


Figure 2.3 Linearized block diagram of the SRF-PLL.

The continuous time transfer function is then found to be

$$H_{SRF-PLL}(s) = \frac{\hat{\theta}}{\theta} = \frac{K_P s + K_I}{s^2 + K_P s + K_I} \tag{2.5}$$

where K_P and K_I are the proportional and integral gains of the PI controller respectively. This type of second order transfer function with one zero is commonly written on the form

$$H(s) = \frac{2\xi\omega_n s + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{2.6}$$

where

$$\omega_n = \sqrt{K_I} \tag{2.7}$$

is the natural frequency and

$$\xi = \frac{K_P}{2\sqrt{K_I}} \tag{2.8}$$

is the damping factor. ω_n effectively tells the speed of the system response. This value should be low for a small bandwidth which gives good noise rejection. ξ tells how well damped the system is. The poles are found as

$$s = -\xi\omega_n \pm \sqrt{(\xi\omega_n)^2 - \omega_n^2} \tag{2.9}$$

It is thus seen that for $\xi < 1$ the system will have complex poles and thus an oscillatory step response. For $\xi = 0$ the output will oscillate at ω_n [9]. A value of around 0.7 gives the fastest settling time for a tolerance band of 5 % [8] [10] [11]. Figure 2.4 shows the bode diagrams and step responses for some different values of ξ with $\omega_n = 2\pi 100$ rad/s. Increasing ω_n moves the bode diagram curves to the right on the frequency axis and squeezes the step responses to the left on the time axis.

The SRF-PLL is also used in single-phase applications. In these systems the initial three phase-to-$\alpha\beta$ transformation is removed. However, an orthogonal component to the input signal must still exist for proper Park transformation in the next step. This is called orthogonal signal generation (OSG) [8]. Two popular methods for this are presented. Unity amplitude is assumed for the input signal.

OSG - The transport delay

The simplest OSG method is the transport delay [12]. This is implemented simply by passing the input signal through a filter with unity gain and delay equal to 90° for the

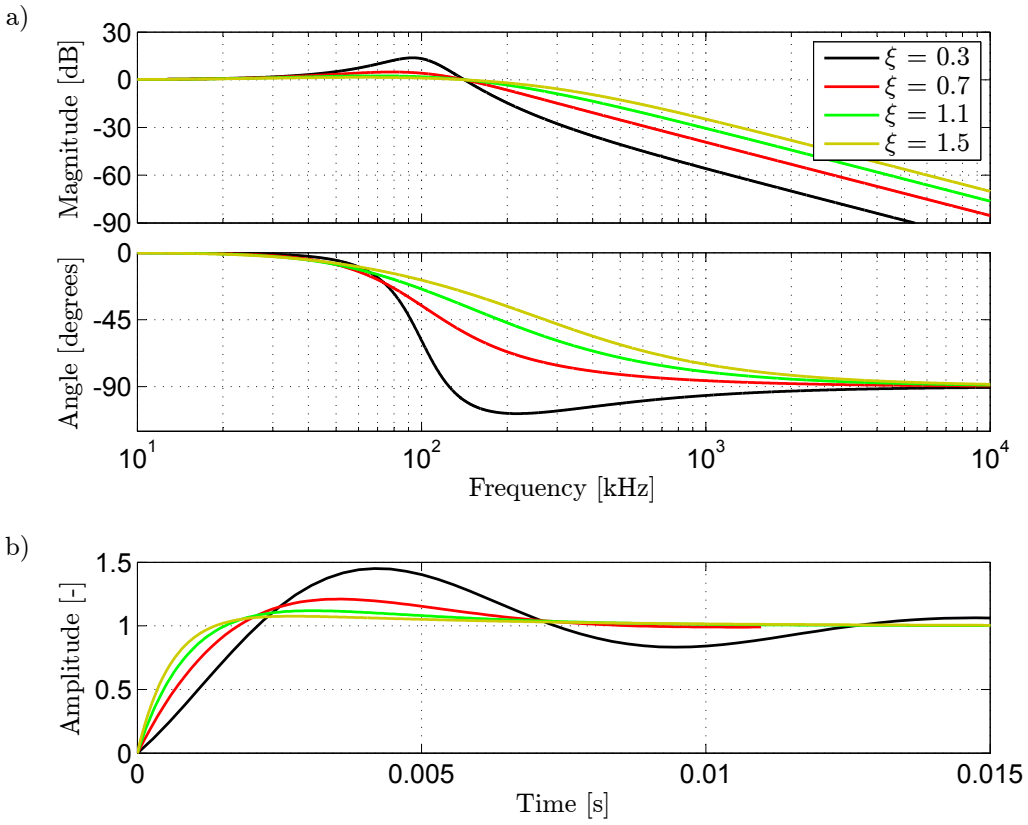


Figure 2.4 Characteristics of second order transfer functions with one zero for different values of ξ with fixed $\omega_n = 2\pi 100$ rad/s. *a)* Bode diagram showing amplitude and phase spectrum. *b)* Step response.

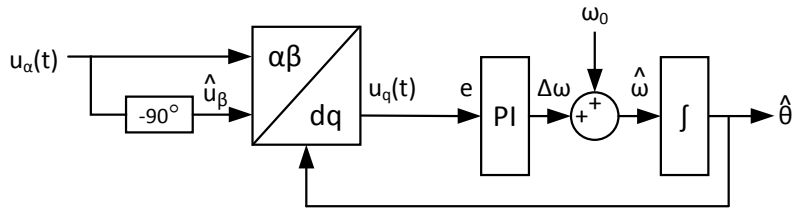


Figure 2.5 Block diagram of the TD-PLL with transport delay as OSG-method.

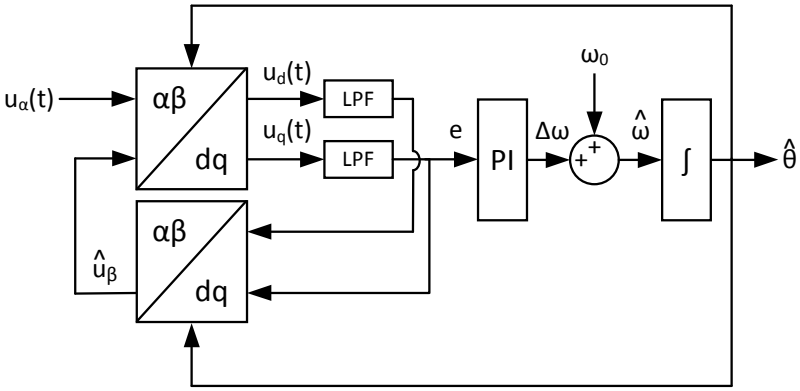


Figure 2.6 Block diagram of the IP-PLL with the inverse Park transformation as OSG-method.

fundamental frequency. The transport delay PLL (TD-PLL) is shown in Figure 2.5. The output of the delay filter is fed as the β-component to the regular SRF-PLL and the unaltered input signal acts as the α-component. For proper orthogonality, the frequency must be stable.

OSG - The inverse Park transformation

The inverse Park transformation PLL (IP-PLL) is widely used and the most well-known of the single-phase SRF-PLL algorithms [8] [7]. The block diagram is shown in Figure 2.6. The linearized model looks like that of Figure 2.3 with a low-pass filter added before the PI controller. The low-pass filters can be of first order, described by

$$LPF(s) = \frac{\omega_c}{s + \omega_c} \tag{2.10}$$

where ω_c is the cut-off frequency. These are required to avoid an algebraic loop but also helps in noise rejection [7].

2.1.3 The enhanced PLL

Additional to the branch for frequency and phase estimation, the enhanced PLL (E-PLL) consists of a loop for estimation of the amplitude. The E-PLL works as a filter to remove the double-frequency ripple. The block diagram is shown in Figure 2.7 where it is seen that three parameters must be set. Two integral gains for each branch and one proportional gain for the phase estimation [7].

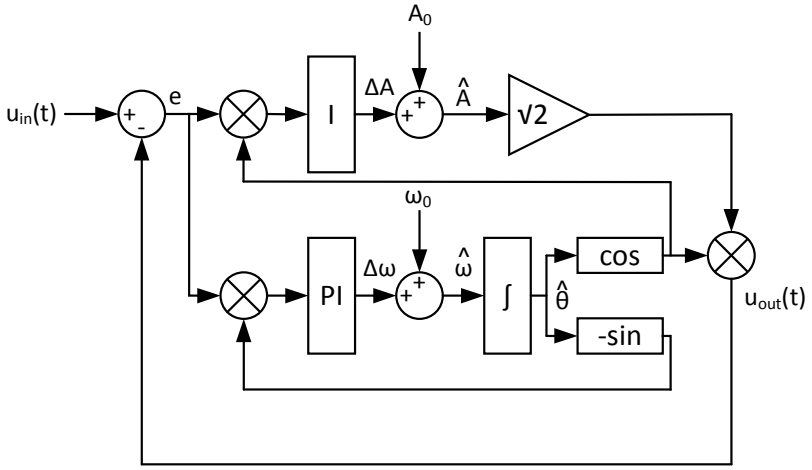


Figure 2.7 Block diagram of the E-PLL.

The algorithm is based on the gradient descent method for minimization of an error defined as

$$e = u_{in}(t) - u_{out}(t) = u_{in}(t) - \sqrt{2}\hat{A}\cos(\hat{\omega}t + \hat{\phi}) \tag{2.11}$$

where $\hat{\omega}t + \hat{\phi}$ is the estimated phase $\hat{\theta}$. The estimation parameters $\hat{A}$, $\hat{\omega}$ and $\hat{\phi}$ are put in to an estimation vector $\boldsymbol{P}$. The error index function is then defined as

$$J(t, \boldsymbol{P})^2 = e(t, \boldsymbol{P})^2 \tag{2.12}$$

By differentiating $J(t, \boldsymbol{P})$ with respect to $\boldsymbol{P}$, the minimum is approached by following the negative direction. From this, the differential equations for each estimation parameter are obtained. These are found to be

$$\begin{cases} \frac{\partial \hat{A}}{\partial t} & = K_{I,A} e \cos\hat{\theta} \\ \frac{\partial \hat{\omega}}{\partial t} & = -K_{I,\omega} e \sin\hat{\theta} \\ \frac{\partial \hat{\theta}}{\partial t} & = \hat{\omega} - K_{P,\omega} e \sin\hat{\theta} \end{cases} \tag{2.13}$$

where $K_{I,A}$ is the integral gain of the amplitude estimation branch and $K_{I,\omega}$ and $K_{P,\omega}$ are the integral and proportional gains of the phase estimation branch respectively. This is how the E-PLL algorithm is implemented [7]. Figure 2.7 shows this with the addition of the nominal frequency outside the PI controller. This is however equivalent to adding the nominal frequency before the proportional contribution of the PI controller as (2.13) indicates. Additionally, a third branch may be added for estimation of dc offset. This consists of a simple integral gain of the error with the output added to the estimated signal [13]. Additional amplitude and phase estimation branches may also be added for estimation of specific harmonics [14].

The transfer function for the phase estimation can be found to be

$$H_{E-PLL}(s) = \frac{A_0(\frac{K_{P,\omega}}{2}s + \frac{K_{I,\omega}}{2})}{s^2 + A_0\frac{K_{P,\omega}}{2}s + A_0\frac{K_{I,\omega}}{2}} \tag{2.14}$$

which, by rewriting according to (2.6) gives

$$\omega_n = \sqrt{\frac{A_0 K_{I,\omega}}{2}} \tag{2.15}$$

$$\xi = K_{P,\omega}\sqrt{\frac{A_0}{8K_{I,\omega}}} \tag{2.16}$$

This is the same type of transfer function as for the SRF-PLL.

2.1.4 The Kalman filter

The Kalman filter (KF) is a model based recursive algorithm which estimates the behavior of a measured process. Both the process itself and the measurements of it may be subjected to noise which is also considered in the model. Model estimates are compared with measurements and new estimates are calculated depending on the user-defined correlation between these quantities. This relies on the principle of conditional distributions, that knowing the outcome of one variable changes the probability of one or several other [15] [16].

Conditional normal distributions

The probability density function for a continuous variable is a function from which the probability of the variable to fall within a certain range is found through the integral over that range. It is defined such that

$$\int_{-\infty}^{\infty} p(x)\mathrm{d}x = 1 \tag{2.17}$$

where x is the continuous variable. The likelihood of X taking any value is 1 meaning that it will absolutely happen.

The normal distribution, also called the Gaussian distribution, is a specific kind of probability density function that commonly occurs in nature [16]. It is defined as

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.18}$$

where σ is the standard deviation and μ is the expected value. σ^2 is called the variance of the distribution [17]. These parameters are illustrated in Figure 2.8. μ is the most probable value to occur and the probability of X to be within the shaded area is approximately 68.3 %.

The probability density function of a two-dimensional random variable $\boldsymbol{Z}$ with a normal distribution is given by

$$p(\boldsymbol{Z}) = \frac{1}{\sqrt{4\pi^2 det(\boldsymbol{Q})}} e^{-\frac{1}{2}(\boldsymbol{Z}-\boldsymbol{\mu})^T \boldsymbol{Q}^{-1}(\boldsymbol{Z}-\boldsymbol{\mu})} \tag{2.19}$$

where μ is the 2-by-1 expected value vector and Q is the symmetric 2-by-2 covariance matrix. The diagonal elements of the covariance matrix holds the variances of the variables. The off-diagonal elements holds the cross-covariance between the two variables [15] [16]. This is defined as the product of the respective standard deviations and the correlation

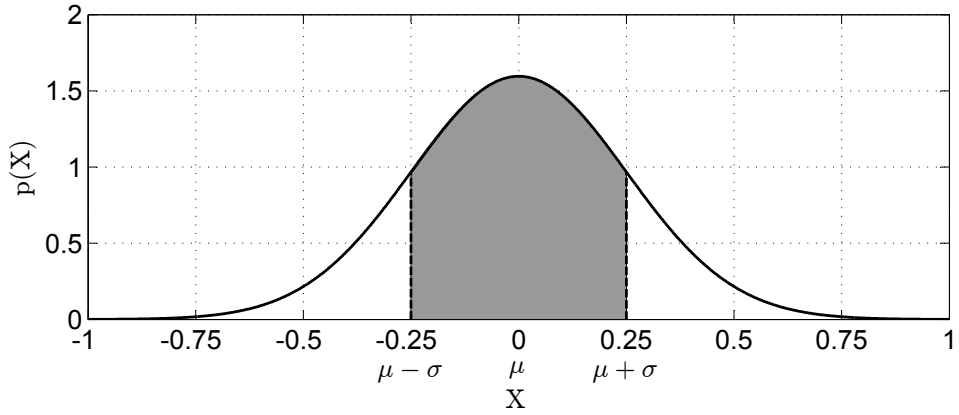


Figure 2.8 Probability density function of a normal distribution for the variable X with μ = 0 and σ = 0.25.

coefficient, denoted r, which is in the range -1 to 1. -1 indicates maximum negative linear correlation, 0 indicates no correlation and 1 indicates maximum positive linear correlation [17] [18]. As the distribution is defined only through the expected value and the covariance matrix, a shorthand notation is

$$\boldsymbol{Z} \sim \ N(\boldsymbol{\mu}, \boldsymbol{Q}) \tag{2.20}$$

Examples of two different joint normal distributions with linear correlation between two random variables are shown in *a)* and *b)* of Figure 2.9. The probability is illustrated by contour lines with black as minimum and white as maximum. If the two variables are independent of each other the distribution will look like in *a)*. In *c)* are the two distributions of $\boldsymbol{X}$ given that $\boldsymbol{Y}$ is 0.490 and -0.766 as shown in *a)*. The distributions overlap perfectly which indicates that the knowledge of $\boldsymbol{Y}$ does not affect the probability of $\boldsymbol{X}$. In other words, there is no correlation between the two variables. The skewed distribution of *b)* indicates that the two variables have a positive correlation. The correlation coefficient r = 0.875. The correlation is seen in *c)* where two distributions of $\boldsymbol{X}$ given that $\boldsymbol{Y}$ is 0.490 and -0.766 as shown in *b)* are presented. It is seen that the expected value of $\boldsymbol{X}$ is different [15] [16].

The joint distribution between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is denoted as

$$\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \right) \tag{2.21}$$

where Q_{11} and Q_{22} are the variances of $\boldsymbol{X}$ $\boldsymbol{Y}$ respectively and Q_{12} is the cross-covariance. The conditional distribution of $\boldsymbol{X}$, knowing that $\boldsymbol{Y}$ = y is found as

$$p_{\boldsymbol{X}|\boldsymbol{Y}}(x|\boldsymbol{Y} = y) = \frac{p_{X,Y}(x, y)}{p_Y(y)} \tag{2.22}$$

The shorthand notation for this can be shown to be

$$p_{\boldsymbol{X}|\boldsymbol{Y}}(x|\boldsymbol{Y} = y) \sim \ N(\mu_x + Q_{12}Q_{22}^{-1}(y - \mu_y), Q_{11} - Q_{12}Q_{22}^{-1}Q_{12}^T) \tag{2.23}$$

which directly shows how the cross-covariance parameter Q_{12} affects the conditional distribution [15] [16].

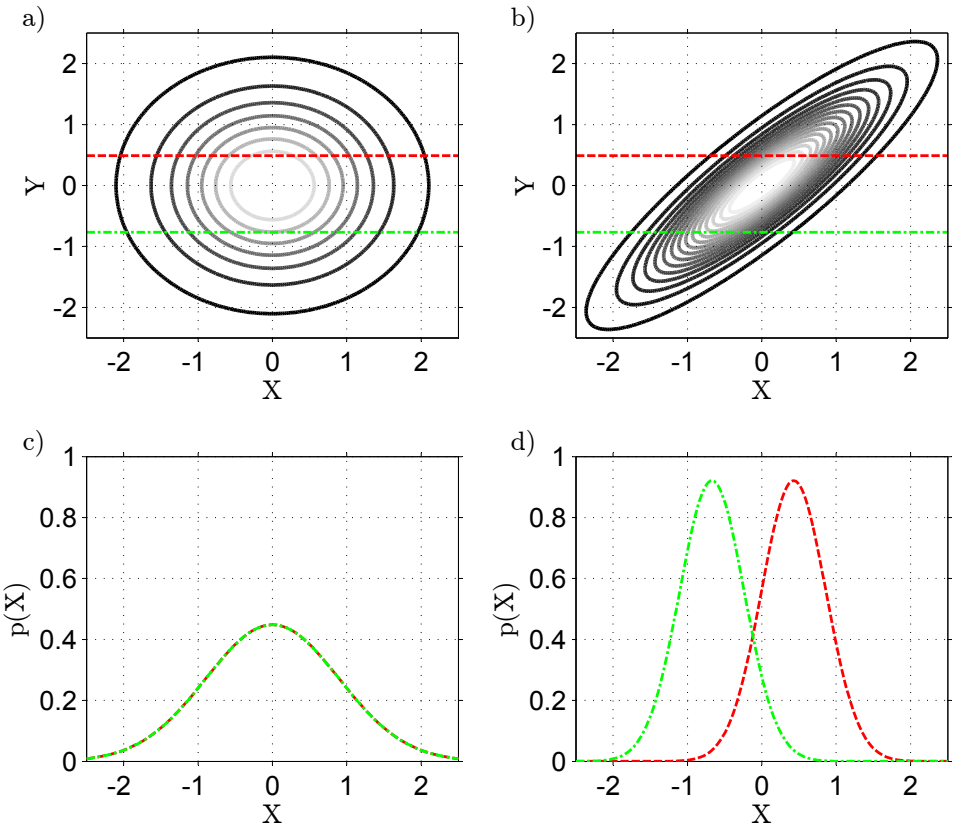


Figure 2.9 Joint normal distributions between two random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ with variance $\sigma^2 = 0.8$. *a)* Joint distribution when $\boldsymbol{X}$ and $\boldsymbol{Y}$ are uncorrelated. *b)* Joint distribution when $\boldsymbol{X}$ and $\boldsymbol{Y}$ are correlated with correlation coefficient $r = 0.875$. *c)* Two conditional distributions of $\boldsymbol{X}$ knowing $\boldsymbol{Y}$ is 0.490 and -0.766 of the distribution of *a)*. *d)* Two conditional distributions of $\boldsymbol{X}$ knowing $\boldsymbol{Y}$ is 0.490 and -0.766 of the distribution of *b)*.

Kalman filter equations

Consider a linear discrete time state space model described as

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{v}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{w}(k) \end{aligned} \tag{2.24}$$

where $\boldsymbol{x}$ is the n-by-1 state vector, $\boldsymbol{A}$ is the n-by-n transition matrix, $\boldsymbol{y}$ is the p-by-1 output vector, $\boldsymbol{C}$ is the p-by-n measurement matrix and $\boldsymbol{w}$ and $\boldsymbol{v}$ are vectors representing the state and measurement noise respectively. The noise is assumed to be normally distributed white noise as

$$\begin{bmatrix} \boldsymbol{v}(k) \\ \boldsymbol{w}(k) \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{Q} & 0 \\ 0 & \boldsymbol{R} \end{bmatrix} \right) \tag{2.25}$$

where the off-diagonal zeros shows that the two noise components are uncorrelated. Assuming a normal distribution also for the state vector,

$$\boldsymbol{x}(k) \sim \ N(\hat{\boldsymbol{x}}_k, \boldsymbol{P}_k) \tag{2.26}$$

, the joint distribution between the state and the output vector can be shown to be [15]

$$\begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{y}(k) \end{bmatrix} \sim N\left(\begin{bmatrix} \hat{\boldsymbol{x}}(k) \\ \boldsymbol{C}\hat{\boldsymbol{x}}(k) \end{bmatrix}, \begin{bmatrix} \boldsymbol{P}_k & \boldsymbol{P}_k\boldsymbol{C}^T \\ \boldsymbol{C}\boldsymbol{P}_k & \boldsymbol{C}\boldsymbol{P}_k\boldsymbol{C}^T + \boldsymbol{R} \end{bmatrix} \right) \tag{2.27}$$

The aim of the Kalman filter is to, given measurements of $\boldsymbol{y}$, update the estimation of the model to adaptively fit it to the true events. This can be written as

$$p_{\boldsymbol{X}|\boldsymbol{Y}}(x(k)|\boldsymbol{Y} = y(k)) \sim N(\hat{\boldsymbol{x}}_k^+, \hat{\boldsymbol{P}}_k^+) \tag{2.28}$$

where $\hat{\boldsymbol{x}}_k^+$ and $\hat{\boldsymbol{P}}_k^+$ are the updates of the expected value and the covariance respectively, based on (2.23). The expected value update is then found as

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k + \boldsymbol{P}_k\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{P}_k\boldsymbol{C}^T + \boldsymbol{R})^{-1}(\boldsymbol{y}(k) - \boldsymbol{C}\hat{\boldsymbol{x}}_k) \tag{2.29}$$

The update of the covariance is

$$\boldsymbol{P}_k^+ = \boldsymbol{P}_k - \boldsymbol{P}_k\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{P}_k\boldsymbol{C}^T + \boldsymbol{R})_k^{-1}\boldsymbol{C}\boldsymbol{P}_k \tag{2.30}$$

The predicted state is then

$$\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{A}\hat{\boldsymbol{x}}_k^+ \tag{2.31}$$

and the predicted covariance is

$$\boldsymbol{P}_{k+1} = \boldsymbol{A}\boldsymbol{P}_k^+\boldsymbol{A}^T + \boldsymbol{Q} \tag{2.32}$$

Equations (2.29) through (2.32) constitute the recursive algorithm of the Kalman filter. The equations are sequentially executed in a loop. The adaptation process must be initiated by some expected value $\hat{\boldsymbol{x}}_{\boldsymbol{0}}$ and covariance $\boldsymbol{P}_0$ commonly set very high to indicate a large initial uncertainty [16]. The values of the covariance matrices for the process and measurement noise, shown in (2.25), sets the steady state function of the filter. Usually it is hard to estimate these and they are therefore often tuned to optimize the filter performance. One important aspect to consider is the relative magnitude between the two. If the magnitude of $\boldsymbol{R}$ is greater than that of $\boldsymbol{Q}$, the measurements are given less weight compared to the model and the other way around [15].

Model of single-phase system

A single-frequency sinusoidal signal without dc offset can be described as

$$S_k = A_k \sin(\omega_k t_k + \theta_k) \tag{2.33}$$

where A_k is the amplitude, ω_k is the angular frequency, t_k is the time, θ_k is the phase displacement and subscript k denotes the sample index. An orthogonal system can then be defined as

$$x_{1,k} = A_k \sin(\omega_k t_k + \theta_k) \tag{2.34}$$

$$x_{2,k} = A_k \cos(\omega_k t_k + \theta_k) \tag{2.35}$$

Consider the sample time T_s so short that the next values of the amplitude, frequency and the phase displacement have changed negligibly. Then,

$$\begin{aligned} x_{1,k+1} &= A_{1,k+1} \sin(\omega_k t_k + \omega_k T_s + \theta_{k+1}) \\ &= A_{1,k+1}[\sin(\omega_k t_k + \theta_{k+1}) \cos(\omega_k T_s) + \cos(\omega_k t_k + \theta_{k+1}) \sin(\omega_k T_s)] \\ &= x_{1,k} \cos(\omega_k T_s) + x_{2,k} \sin(\omega_k T_s) \end{aligned} \tag{2.36}$$

$$\begin{aligned} x_{2,k+1} &= A_{1,k+1} \cos(\omega_k t_k + \omega_k T_s + \theta_{k+1}) \\ &= A_{1,k+1}[\cos(\omega_k t_k + \theta_{k+1}) \cos(\omega_k T_s) - \sin(\omega_k t_k + \theta_{k+1}) \sin(\omega_k T_s)] \\ &= -x_{1,k} \sin(\omega_k T_s) + x_{2,k} \cos(\omega_k T_s) \end{aligned} \tag{2.37}$$

This can be written in state space form as

$$\begin{aligned} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} &= \begin{bmatrix} \cos(\omega_k T_s) & \sin(\omega_k T_s) \\ -\sin(\omega_k T_s) & \cos(\omega_k T_s) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \boldsymbol{v}_k \\ \boldsymbol{y}_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \boldsymbol{w}_k \end{aligned} \tag{2.38}$$

where the vector $\boldsymbol{v}_k$ represents process noise and w_k measurement noise as presented in (2.25). The process noise can be sudden phase jumps, frequency variations or other deviations from the ideal model [19] [20] [21].

Summary

By sampling a single-phase signal, the future values can be predicted with the state-space model of (2.38) and the KF equations (2.29) through (2.32). The algorithm is initiated by a first guess plus an estimation of the covariance matrix. This is usually set high to indicate large uncertainty about the initial state of the measured process. The system adapts its estimates to the process and the values of the covariance matrix are minimized. This gives a problem for when the system suddenly changes, such as for a phase jump. With a small-valued covariance matrix, the adaptation process will be slow. This can be solved by resetting the covariance matrix when the error gets to high [20] [22].

The required matrices with sizes and explanations are presented in Table 2.1. The index k indicates that the matrix is updated at each sampling instance. Additionally, the frequency, ω_k, and the sampling time, T_s, must be set for the model. The presented values are only valid for tracking of the fundamental component. Additional states can be added for estimation of dc offset [21] and harmonics [19] [20] [22]. By estimating the frequency of the output, the model can be updated and the system adapt to a non-nominal state [19].

Table 2.1 Summary of the matrices required for estimation of the fundamental component in a single-phase input with the Kalman filter.

Matrix	Symbol	Size	Explanation
Measurement	$\boldsymbol{y}_k$	1x1	One measurement channel
State estimation	$\hat{\boldsymbol{x}}_k$	2x1	Two states
State transition	$\boldsymbol{A}$	2x2	Two states
State covariance	$\boldsymbol{P}_k$	2x2	Two states
Process covariance	$\boldsymbol{Q}$	2x2	Two states
Measurement covariance	$\boldsymbol{R}$	1x1	One measurement channel

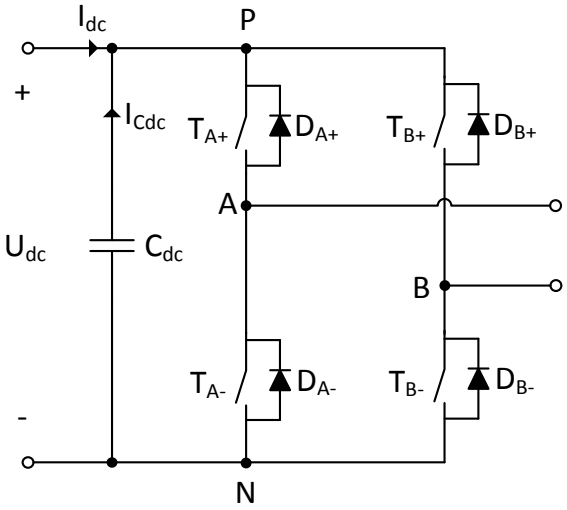


Figure 2.10 Schematic diagram of the single-phase full-bridge converter with a capacitor on the dc-link for reducing voltage ripple.

2.2 Single-phase full-bridge converter

The schematic diagram of the single-phase full-bridge converter is shown in Figure 2.10. A capacitor is used on the dc-link for reducing voltage ripple. The output voltage pattern is controlled by switching the four transistors. Each vertical pair is called a phase leg and a horizontal pair is called an arm. The freewheeling diodes gives paths for the load current when the load current and voltage have different polarities. Commonly, the input is a dc voltage. The converter may be used for both ac and dc outputs, all controlled by the switching pattern. The peak output voltage is equal to the input voltage.

2.2.1 Pulse-width modulation

A popular control scheme for the full-bridge converter is pulse-width modulation (PWM). In this strategy, a reference waveform is compared with a triangular shaped control voltage. The output is positive when the reference value is larger than the control voltage and zero otherwise. For generation of an ac output, the reference is a sinusoidal waveform and two base parameters for this comparison are the modulation index and the frequency index.

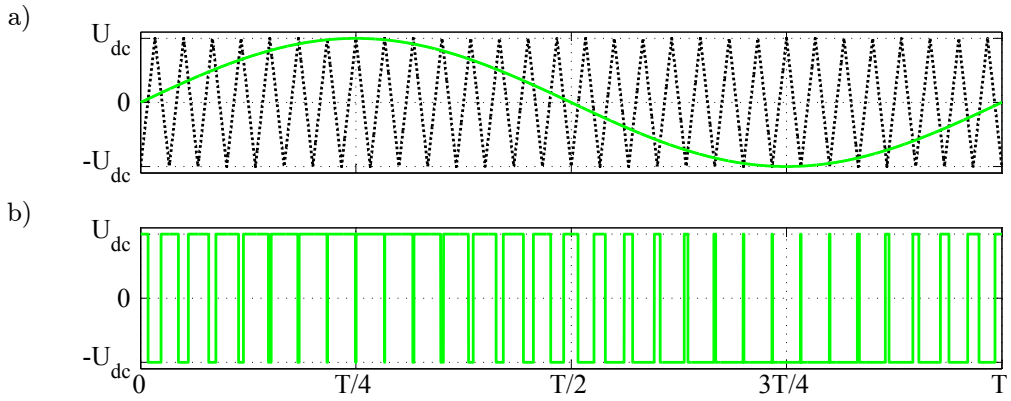


Figure 2.11 Bipolar PWM switching patterns for unity modulation index and frequency index of 30. *a)* Comparison of sinusoidal references to triangular carrier wave. *b)* Output voltage.

The modulation index is defined as

$$m_a = \frac{\hat{U}_{ref}}{\hat{U}_{ctrl}} \tag{2.39}$$

where $\hat{U}_{ref}$ and $\hat{U}_{ctrl}$ are the peak amplitudes of the reference and control voltage respectively. $\hat{U}_{ctrl}$ is equal to the input voltage of the converter. The frequency index is defined as

$$m_f = \frac{f_{ref}}{f_{ctrl}} \tag{2.40}$$

where f_{ref} and f_{ctrl} are the frequencies of the reference and control voltage respectively. Two common PWM strategies are the bipolar and the unipolar switching.

The bipolar PWM is shown in Figure 2.11 for generation of a sinusoidal output with a unity modulation index and a frequency index of 30. *a)* shows comparison between the reference and the control voltage. *b)* shows the resulting output voltage, U_{AB}, with reference to Figure 2.10. In the bipolar PWM, the diagonal transistors are switched in pairs. When the reference is higher than the control voltage, transistors T_{A+} and T_{B-} are on and T_{A-} and T_{B+} are off. This gives an output voltage of U_{dc}, as can be seen in Figure 2.11. When the control voltage is highest the opposite pair is on, giving an output of $-U_{dc}$.

The unipolar PWM is shown in Figure 2.12, also for generation of a sinusoidal output, with unity modulation index and a frequency index of 30. The comparison is made in *a)*, the voltages of the two phase legs are shown in *b)* and *c)* respectively and the resulting voltage over the load is shown in *d)*. The difference to the bipolar case is that an inverted reference is added. Each reference controls one phase leg. The top transistors are on and the lower are off when the reference is higher and the other way around when it is lower.

Figure 2.13 shows the amplitude spectra of the output voltage for bipolar PWM in *a)* and unipolar PWM in *b)*. The leftmost bar is the fundamental voltage which is equal to the reference. The advantage of the unipolar PWM is seen as that the first harmonics appears around $2m_f$ compared to around m_f for the bipolar case and therefore are easier to filter out. A low-pass filter must be present on the output to retrieve the reference waveform [23].

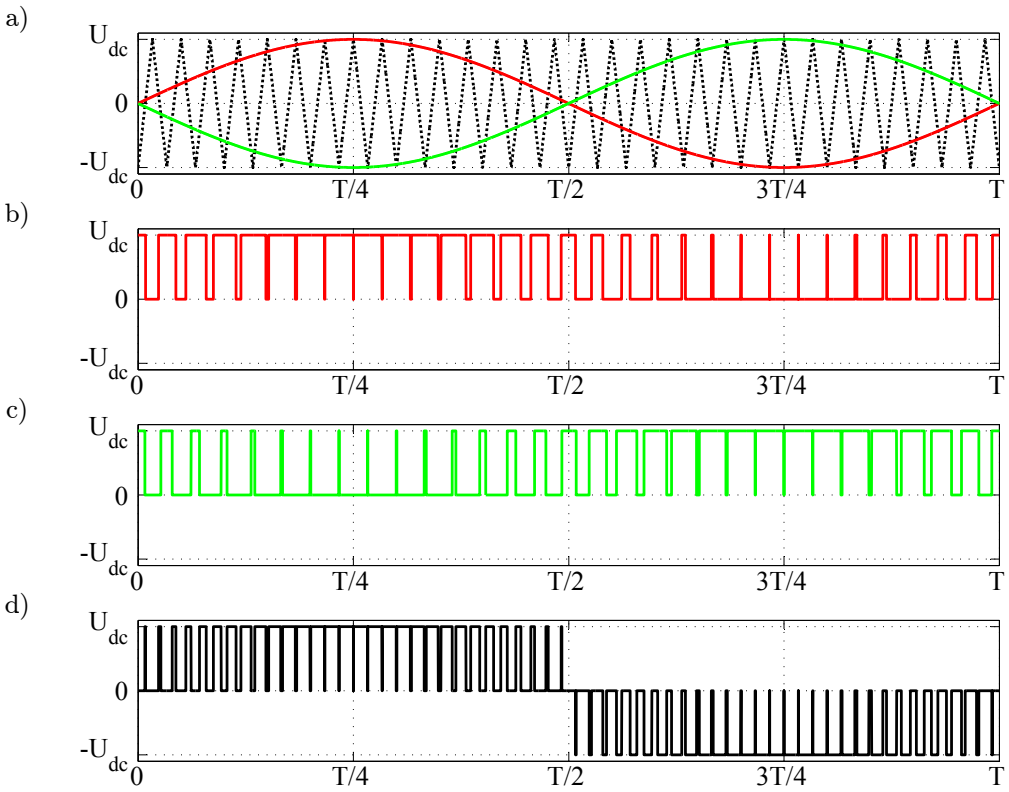


Figure 2.12 Unipolar PWM switching patterns for unity modulation index and frequency index of 30. *a)* Comparison of sinusoidal references to triangular carrier wave. *b)* PWM signals for the first phase leg. *c)* PWM signals for the second phase leg. *d)* Output voltage.

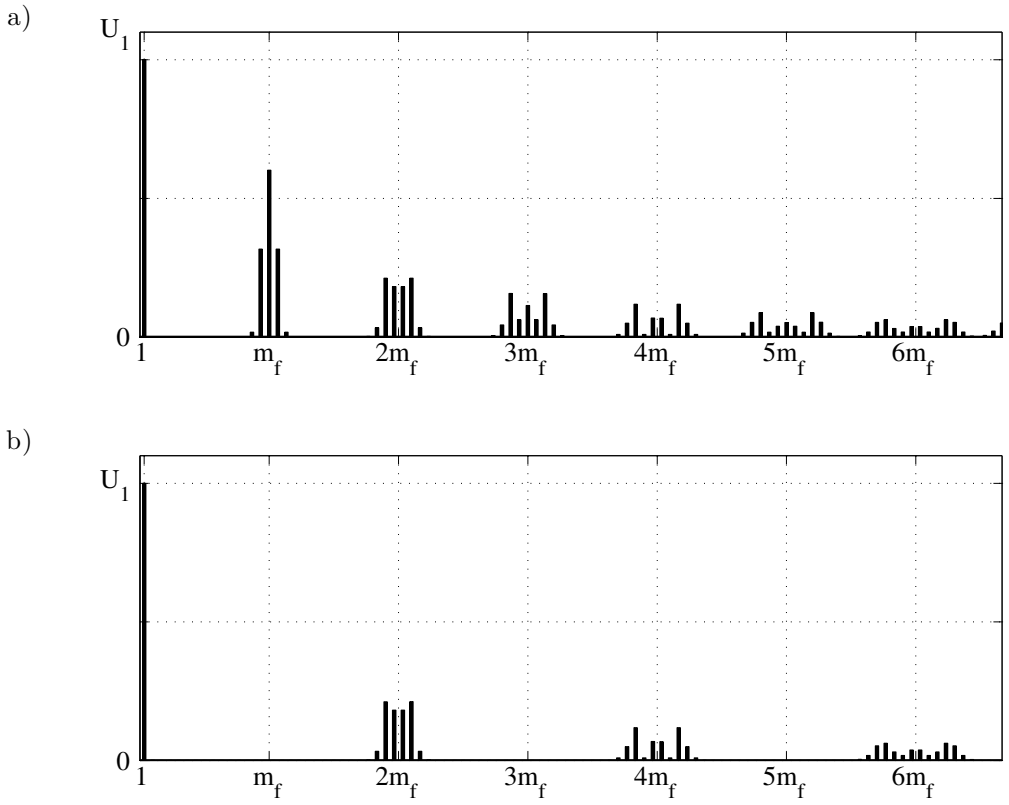


Figure 2.13 PWM output voltage amplitude spectra for unity modulation index and frequency index of 30. *a)* Bipolar PWM. *b)* Unipolar PWM.

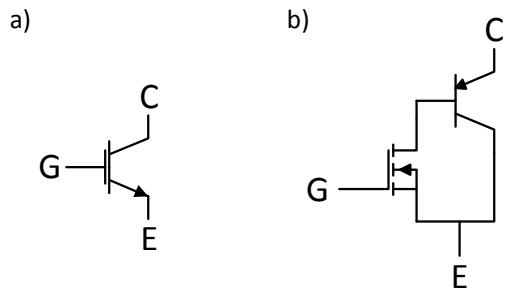


Figure 2.14 a) Circuit symbol of the IGBT. *b)* Equivalent circuit of the IGBT as composed by the MOSFET to the left and the BJT on top.

2.3 The insulated-gate bipolar transistor

The insulated-gate bipolar transistor (IGBT) is the device most commonly used in new high-voltage applications [24]. The circuit symbol is shown in Figure 2.14 *a)*, where C stands for collector, G for gate and E for emitter. The pin names come from the two older transistor types that the IGBT is a combination of. These are the bi-junctional transistor (BJT) and the metal-oxide field-effect transistor (MOSFET). This is illustrated in an equivalent circuit in Figure 2.14 *b)*. The IGBT combines the low conduction losses of the BJT with the insulated gate of the MOSFET. The insulated gate means that less current is required during switching than for the BJT. The MOSFET and the IGBT are voltage-controlled devices while the BJT is current-controlled. The IGBT is replacing the MOSFET in new high-voltage applications [24].

The IGBT has two significant drawbacks as compared to the MOSFET. First, the IGBT has longer switching times. This is because of longer current fall time which is inherited from the BJT part. Secondly, it has no intrinsic anti-parallel diode [24]. This is required in inverter bridges for driving reactive loads as discussed in Chapter 2.2. There are IGBT modules with these diodes built-in to the same package. The power diode is presented additional to the IGBT in the following subsections.

2.3.1 Conduction losses

Figure 2.15 shows typical current-voltage characteristics of the IGBT. It is seen in *a)* that the amount of current that is passed through the collector is decided by the voltage between the gate and emitter. When in full conduction mode, the IGBT operates in the linear region to the left. This can be modeled as a constant voltage drop, V_{C0}, plus a resistive part, $R_{on,\,IGBT}I_C$. The resistive part is dependent on the value of V_{GE} which sets the slope of the linear region. Figure *b)* shows $V_{GE(th)}$ which is the threshold value below which there is no collector current flowing. There is a maximum negative voltage that can be withstood before the IGBT breaks down and the current rushes [23] [24].

Figure 2.16 shows the typical current-voltage characteristics of the power diode. Also this can be simplified as a constant voltage drop, V_{F0}, and a resistive part, $R_{on,\,diode}I_F$. Same as for the IGBT, the power diode has a maximum negative voltage it can withstand before failure [23] [24].

Both the voltage drop and the resistance can be considered constant through the short

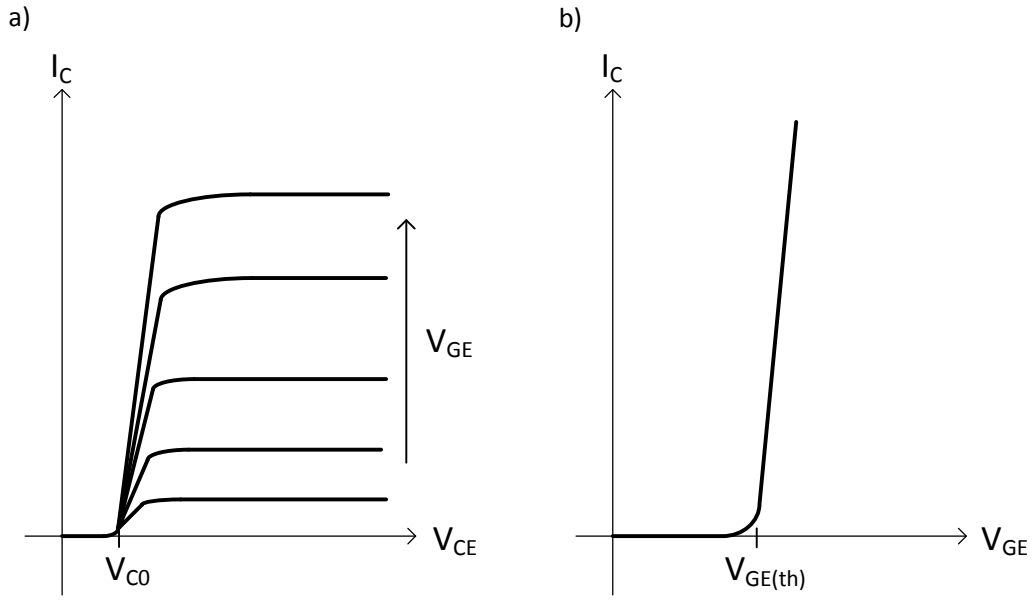


Figure 2.15 Typical current-voltage characteristics of the IGBT. *a)* Collector current vs collector-emitter voltage. *b)* Collector current vs gate-emitter voltage.

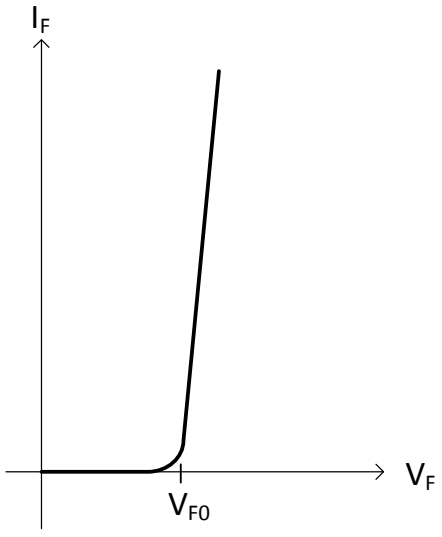


Figure 2.16 Typical current-voltage characteristics of the power diode.

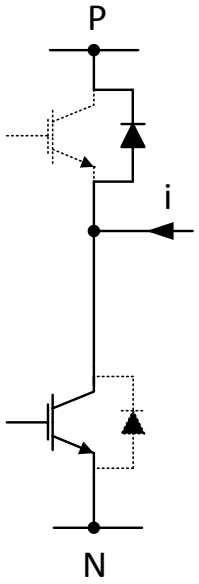


Figure 2.17 The active parts for illustration of the IGBT switching waveforms.

conduction period. The average conduction losses for the IGBT is therefore found as

$$P_{cond,\,IGBT} = \frac{1}{T}\int_0^T I_C V_{C0}\mathrm{d}t + \frac{1}{T}\int_0^T I_C R_{on,\,IGBT} I_C \mathrm{d}t = I_{C,AVG}V_{C0} + I_{C,RMS}^2 R_{on,\,IGBT} \tag{2.41}$$

where $I_{C,AVG}$ is the average current and $I_{C,RMS}$ is the RMS current through the IGBT. The conduction losses for the diode are found similarly as

$$P_{cond,\,diode} = \frac{1}{T}\int_0^T I_F V_{F0}\mathrm{d}t + \frac{1}{T}\int_0^T I_F R_{on,\,diode} I_F \mathrm{d}t = I_{F,AVG}V_{F0} + I_{F,RMS}^2 R_{on,\,diode} \tag{2.42}$$

where $I_{F,AVG}$ is the average current and $I_{F,RMS}$ is the RMS current through the diode.

2.3.2 Switching losses

Switching waveforms of the IGBT are shown in Figure 2.18. The illustrated event describes the commutation of the load current from an ideal diode to an IGBT in the same phase leg and back again. The involved components are shown with solid lines in Figure 2.17. When the IGBT is turned on by increase of the gate-emitter voltage, there is a delay until $V_{GE(th)}$ is reached before the collector current starts increasing. The current is then increased until all current has been taken over from the diode. The rate of increase is set by the gate-resistance, R_G, and the gate-emitter capacitance of the IGBT. When the load current is reached, the collector-emitter voltage starts to drop as the diode may now get reverse-biased. The drop consists of one fast MOSFET part and one slow BJT part which is called the tailing current. When the saturation level is reached, the IGBT is fully on. Now the gate-emitter voltage rises to its steady state value [23].

The turn-off sequence starts with a delay for the gate-emitter voltage to drop down to the minimum level required for maintaining the collector current. The collector-emitter

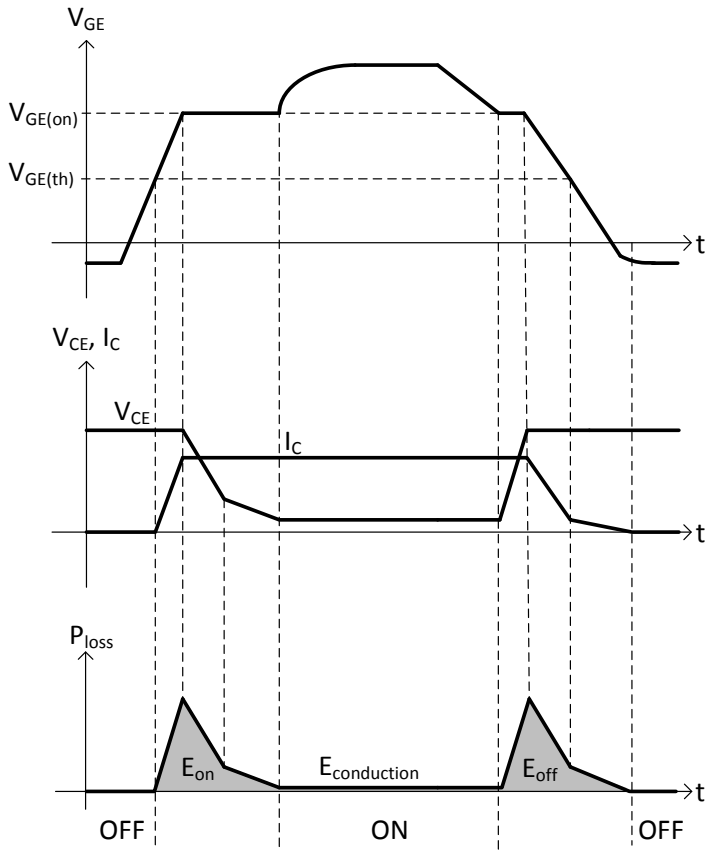


Figure 2.18 Typical switching waveforms of IGBT with inductive load.

voltage then rises so that the diode may get forward-biased and the collector-current can start to drop. The collector-current drop is also divided in to two parts, a fast MOSFET part and the tailing current which is the slow BJT part. The gate-emitter voltage is again set to a negative value in the off-state [23].

In Figure 2.19, the switching waveforms of the power diode are shown. The overshoot, V_{FP}, seen during the turn-on can be a couple tens of volts and is usually not considered in loss calculations. The turn-off losses are the significant switching losses of the power diode. These consist of Q_{rr}, the *reverse recovery* charge. This represent charges that are in excess in the diode during its on-state for minimum conduction losses. This is finalized at the negative peak, I_{rr}, and the voltage can then go negative while the current quickly goes to zero. Due to stray inductance, that is always present in a practical circuit, this will induce a voltage peak, V_{rr}. Additionally, the blocking voltage V_R falls over the diode. When the diode is used as in the full-bridge converter, the current fall rate is set by the IGBT that is taking over the current. This rate is controlled by the circuit controlling the gate-emitter voltage as explained above. V_R is in this case the dc-link voltage [23].

In datasheets, the switching losses are presented for specific parameter values. These usually include the collector-emitter voltage, collector current, gate resistance, junction temperature and what kind of load that was used. To estimate the losses for other scenarios, scaling of the datasheet values must be performed. For the IGBT, when used for sinusoidal PWM and calculating the average switching losses over one reference period, the scaling

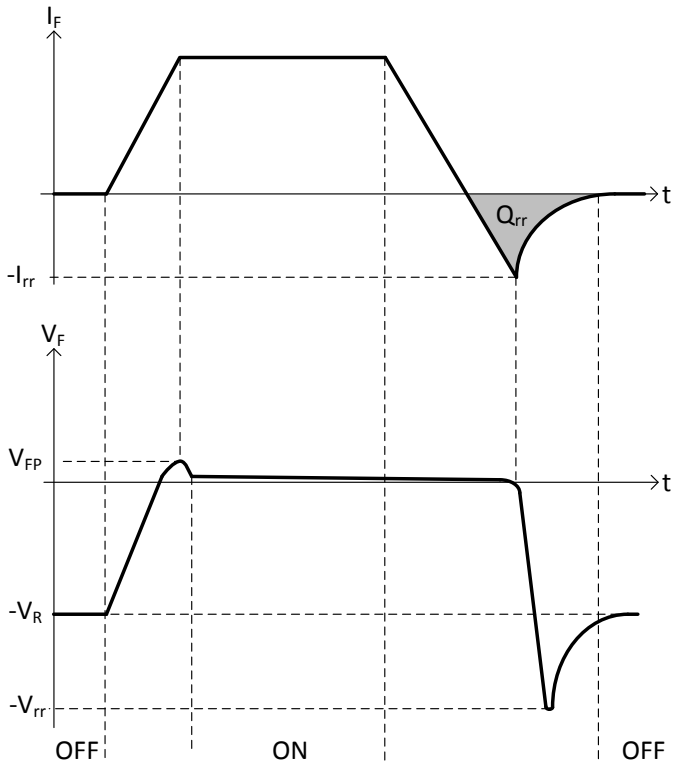


Figure 2.19 Typical switching waveforms of power diode.

can be done as

$$E_{sw,AVG,\,IGBT} = E_{sw,\,IGBT,ds}\frac{1}{\pi}\frac{\hat{I}_L}{\hat{I}_{L,ds}}\left(\frac{V_{cc}}{V_{cc,ds}}\right)^{K_{v,\,IGBT}}(1 + K_{t,\,IGBT}(T_j - T_{j,ds})) \tag{2.43}$$

where subscript ds indicates datasheet values related to the specified switching losses $E_{sw,\,IGBT,ds}$, $\hat{I}_L$ is the peak load current, V_{cc} is the dc-link voltage, $K_{v,\,IGBT}$ regulates the voltage dependency and $K_{t,\,IGBT}$ the temperature dependency. $K_{v,\,IGBT}$ can be between 1.3 and 1.4 and $K_{t,\,IGBT}$ is 0.003. The scaling of the diode switching losses, for calculation of average of switching losses during one sinusoidal PWM reference period, are found similarly as

$$E_{sw,AVG,\,diode} = E_{rr,\,diode,ds}\left(\frac{1}{\pi}\frac{\hat{I}_L}{\hat{I}_{L,ds}}\right)^{K_{i,\,diode}}\left(\frac{V_{cc}}{V_{cc,ds}}\right)^{K_{v,\,diode}}(1 + K_{t,\,diode}(T_j - T_{j,ds})) \tag{2.44}$$

where subscript ds indicates datasheet values related to the specified switching losses $E_{rr,\,diode,ds}$, $\hat{I}_L$ is the peak load current, V_{cc} is the dc-link voltage, $K_{i,\,diode}$ regulates the current dependency, $K_{v,\,diode}$ the voltage dependency and $K_{t,\,diode}$ the temperature dependency. $K_{v,\,diode}$ and $K_{i,\,diode}$ are both 0.6 and $K_{t,\,diode}$ is 0.006 [25].

2.3.3 Thermal considerations

The losses of the IGBT and the diode will generate heat that must be transported away from the silicon chip and the case. The chip temperature should normally be below 125 °C for

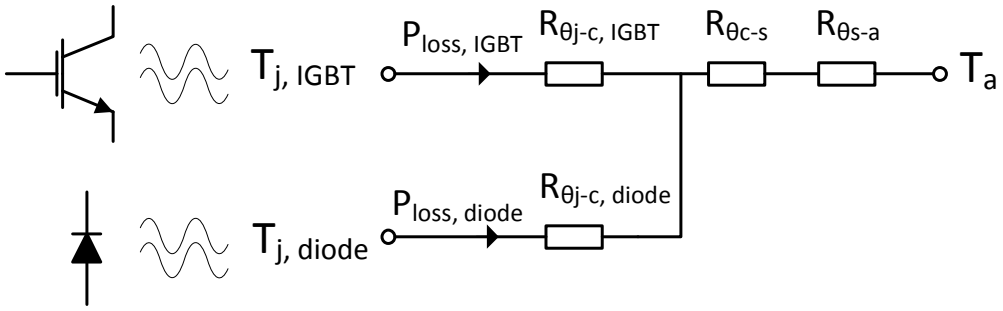


Figure 2.20 Thermal circuit for a case of one IGBT with anti-parallel diode mounted on a heat sink.

proper function but some devices can handle up to 175 °C. The design of the cooling system should aim at keeping the temperature below the breakdown value of the device [23] [26].

The chip temperature is determined by the thermal resistance between the junction and the ambient, the ambient temperature and the losses in the chip. For a case of one IGBT with an anti-parallel diode mounted on a heat sink, a thermal circuit may be drawn as shown in Figure 2.20. This shows the power flow from the high-temperature IGBT and diode junctions, $T_{j,\,IGBT}$ and $T_{j,\,diode}$, to the ambient, T_a. The thermal resistances are divided into three parts. $R_{\theta j-c}$ is the *junction-to-case* resistance, $R_{\theta c-s}$ is the *case-to-sink* resistance and $R_{\theta s-a}$ is the *sink-to-ambient* resistance. These have the units of K/W [23] [26]. Solving the resulting junction temperature for the IGBT gives

$$T_{j,\,IGBT} = T_a + (P_{loss,\,IGBT} + P_{loss,\,diode})R_{\theta s-a} + (P_{loss,\,IGBT} + P_{loss,\,diode})R_{\theta c-s} + P_{loss,\,IGBT}R_{\theta j-c,\,IGBT} \tag{2.45}$$

and for the diode

$$T_{j,\,diode} = T_a + (P_{loss,\,IGBT} + P_{loss,\,diode})R_{\theta s-a} + (P_{loss,\,IGBT} + P_{loss,\,diode})R_{\theta c-s} + P_{loss,\,diode}R_{\theta j-c,\,diode} \tag{2.46}$$

With the values of $R_{\theta j-c}$ and $R_{\theta c-s}$ as given by the manufacturer, the proper heat sink should be chosen for keeping the junction temperatures at the design value. For some applications only a natural convection-cooled heat sink is sufficient, while force-cooled by fan or water flow may be necessary in other. Some kind of thermal grease must usually be applied between the case and the heat sink for obtaining the thermal *case-to-sink* resistance provided by the manufacturer [23].

2.3.4 Gate driver

The gate driver circuit is the interface between the control system and the IGBT. The control system is usually not designed for high output power. Therefore, a gate driver circuit must be used for amplification of the control signal. Following are important aspects to consider for the gate driver circuit to an IGBT [23] [27].

- **Maximum output voltage:** The gate emitter voltage must be high enough for the wanted collector current to be met for the specific IGBT.

- **Maximum switching frequency:** The switching frequency should be high for minimum current ripple through the IGBT.
- **Power requirement:** The IGBT requires a certain amount of power depending on switching frequency and gate-emitter voltages. The power requirement can be calculated as

$$P_G = Q_G(V_{GE(on)} - V_{GE(off)})f_{sw} \tag{2.47}$$

  where Q_G is the gate charge, $V_{GE(on)}$ is the gate-emitter voltage when IGBT is turned on, $V_{GE(off)}$ is the gate-emitter voltage when the IGBT is turned off and f_{sw} is the switching frequency.
- **Average current requirement:** The average current is decided by the amount of charge that must be fed to and taken from the gate-emitter capacitance each turn-on and turn-off. This is found as

$$I_{G,AVG} = Q_G f_{sw} \tag{2.48}$$

- **Peak current requirement:** When the gate-emitter capacitance is uncharged during turn-on and fully-charged during turn-off, the gate current peaks. The peak current is found as

$$I_{G,peak} = \frac{V_{GE(on)} - V_{GE(off)}}{R_{G,min}} \tag{2.49}$$

  where $R_{G,min}$ is the minimum gate resistance during turn-on and turn-off .

2.4 CompactRIO™

CompactRIO™ is a control and acquisition system from National Instruments (NI). An example of it can be seen in Figure 2.21. It consists of a real-time (RT) controller embedded in a chassis for connection of reconfigurable I/O modules. The RT controller has a special operating system (OS) for running programs with high reliability without interruptions and delays common on a standard OS. The interface between the RT controller and the I/O modules is a field programmable gate array (FPGA). This is used for maximum reliability regarding execution speed and synchronization. The graphical programming language LabVIEW™, also from NI, can be used for programming both the FPGA and the RT controller. Available I/O modules are voltage meters, digital I/O, serial communication links and temperature measurement units among others [28] [29].

2.4.1 The field programmable gate array

The FPGA is a digital integrated circuit. It contains blocks of logic gates with configurable functions and connections. Differing from regular processing units, the internal configuration of the FPGA can be changed by the user, hence the term *field programmable*. The FPGA can be of either one-time programmable type or a type that can be reprogrammed indefinitely. Other famous processing units are the programmable logic device (PLD) and the application specific integrated circuit (ASIC). The hardware of both the PLD and the ASIC is predetermined by the manufacturer. The user can make changes in the functions of

Figure 2.21 CompactRIO™ system with eight I/O modules. Courtesy of National Instruments.

the PLD but is much more limited than with the FPGA. The ASIC is a device tailored for a specific purpose. This makes its performance optimized but without reconfigurability [30].

One particular feature that distinguishes the FPGA from other digital circuits is the parallelism inherent from the array structure. Commonly, programs are written with a sequential flow of execution. By performing independent operations in parallel with eachother, much time can be saved [30].

The fixed-point data type

Fixed-point is a way of representing numbers in digital systems. The fixed-point data type is defined by its word and integer length in number of bits. The word length is the total number of bits and the integer length is the number of bits reserved for the integer part which may use one bit reserved for the sign. The remaining bits constitute the decimal value. The decimal point is thus fixed. This is different from the floating-point representation where there is no fixed division between integer and decimal size. In general, the fixed-point data type has lower precision than the floating-point. Most FPGAs however have to be programmed with the fixed-point data type since floating-point operations are not supported. Handling floating-point processing often requires additional cost and less timing accuracy [31].

In LabVIEW, a signed fixed-point number represented by a total of 32 bits out of which 16 are reserved for the integer part, is denoted as $<\pm,32,16>$. The minimum word length is 1 bit and the maximum is 64 bits [31].

2.4.2 LabVIEW™

LabVIEW™ is a graphical programming language in which blocks of various functions are connected by threads to form the program. The code forms pathways for the data to flow and the execution of one block is immediate on the arrival of data to all of its inputs. Multiple operations can therefore be executed at the same time. This differs from regular text-based programming languages were the execution is sequential.

A LabVIEW™ program is called a virtual instrument (VI). A VI is divided into two

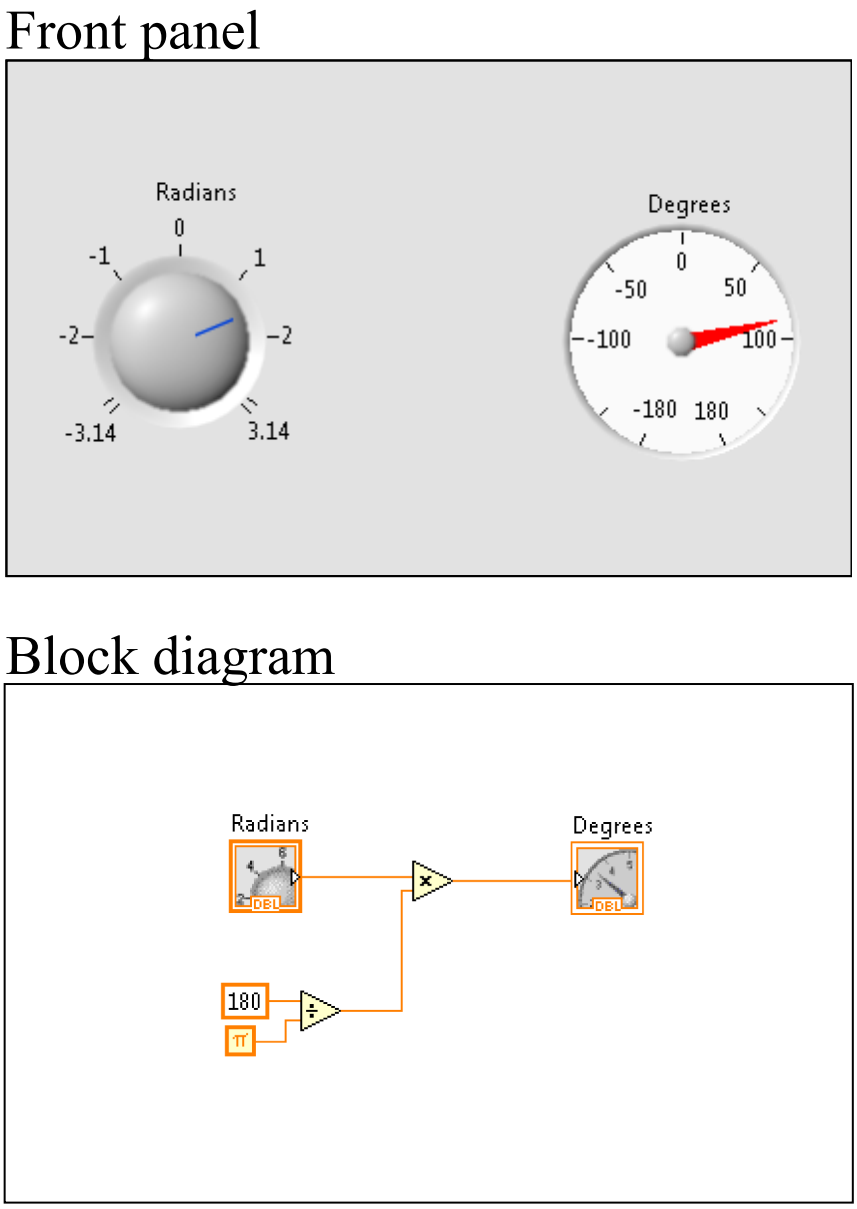


Figure 2.22 Example of LabVIEW™ code showing two front panel objects and how they are linked together in the block diagram.

parts, one front panel that constitute the graphical interface to the user and one block diagram in which the front panel objects are linked with other blocks. The front panel can contain controls and indicators. An example is seen in Figure 2.22. On the front panel, a dial can be turned to change a value in radians and the gauge shows the corresponding value in degrees. In the block diagram, the knob and gauge are connected through the multiplication with the radians-to-degrees conversion ratio. There are many different front panel objects such as charts, slides and buttons [29] [32].

In addition to operational blocks, LabVIEW™ includes common programming structures such as while and for loops. Similar to subroutines in text-based programming languages, a VI can be divided into subVIs. The front panels of the subVIs are not normally seen during execution but interaction is made with the block diagram. VIs can be executed on a computer for simulations or for interactions with measurement and control hardware connected to a computer or in a standalone unit such as CompactRIO™ [29] [32].

Chapter 3

Simulations

3.1 Phase tracking

The phase tracking system should give a reference sinusoidal wave to the inverter control system for PWM generation. The system needs not to be fast but very accurate. The standard deviation of the phase should be 20-50 μrad. If a voltage-sag or a phase-jump would occure the calibration sequence is ruined even if the phase is immediately updated accordingly. The system should stabilize within a couple of seconds and be stable for up to 20 seconds. The tuning of the algorithm parameters has been aimed at reaching steady state within 10 seconds.

3.1.1 The synchronous reference frame PLL

The SRF-PLL with ideal inputs was used for tuning the PI controll parameters. The TD-PLL, the IP-PLL and the E-PLL all have the same second order transfer functions with one zero as the SRF-PLL. Therefore, the same values of ξ and ω_n can be used for these PLLs, as was found to be a suitable tuning for the SRF-PLL.

The SRF-PLL algorithm was built up in SIMULINK® as shown in Figure 3.1 for continuous time simulations. The implementation of the Park transformation block is shown in Figure 3.2. It can be seen in Figure 3.1 that the angle of the dq-system is used, rather than the q-component, as the error signal to the PI controller. Both should be zero in steady state. However, the value of the q-component is dependent on the input amplitude whilst the angle is not. Using the angle thus works as a normalization of the error. This is important since the derivation of the transfer function (2.5) assumes unity input. Since

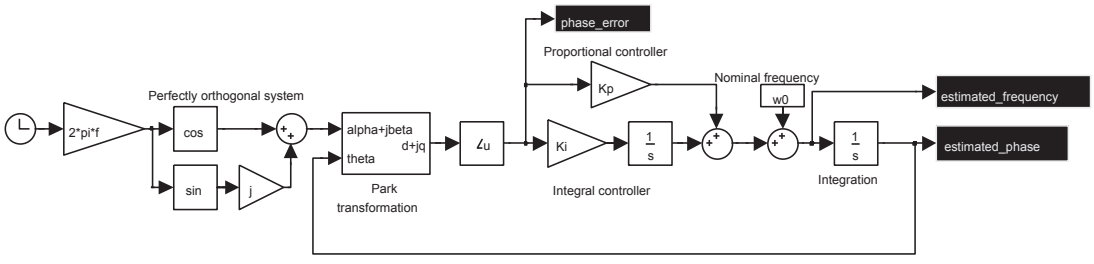


Figure 3.1 Block diagram of the SRF-PLL with ideal inputs as implemented in SIMULINK® for continuous time simulations.

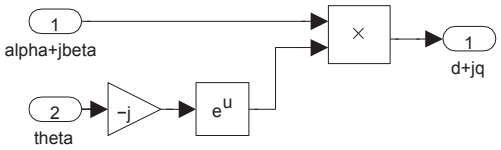


Figure 3.2 Park transformation implemented in SIMULINK®

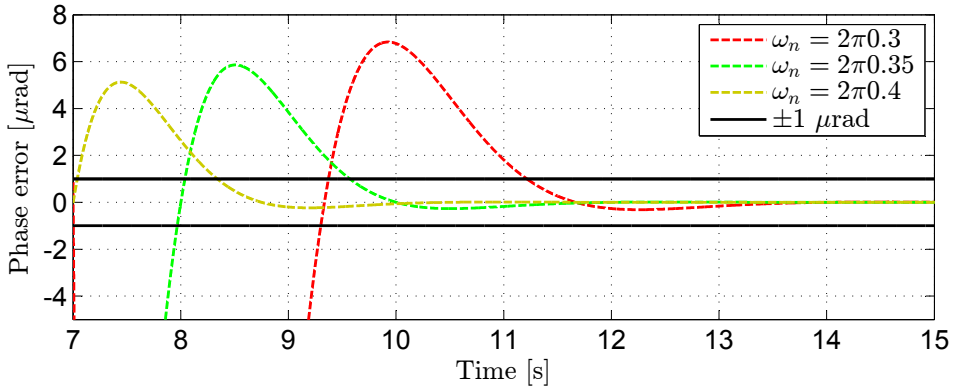


Figure 3.3 Test of different ω_n to find minimum required bandwidth for stabilization within 10 seconds for the SRF-PLL.

(2.5) can be rewritten as (2.6), the PI control parameters for the SRF-PLL can be found as

$$K_P = 2\xi\omega_n \tag{3.1}$$

$$K_I = \omega_n^2 \tag{3.2}$$

where $\xi \approx 0.7$ gives the best transient response and the bandwidth ω_n should be chosen for preferred response time and noise rejection. To find a suitable value for ω_n a noise-free signal at 50 Hz with unity amplitude was fed in to the SRF-PLL with $\xi = 0.7$ for different values of ω_n. The nominal frequency of the PLL was set to 49 Hz, for covering incorrect initial guesses. An interval around 10 s was investigated to find out if the system had settled. The results from the simulation with different ω_n are shown in Figure 3.3. The tuning requirement was chosen as that the error had reached below $|1|$ μrad, without going out of the bounds again. It is seen that at least $\omega_n = 2\pi 0.35$ rad/s is required to obtain a settling time of maximum 10s. With a more accurately known frequency a smaller value may be chosen.

In the final system, the reference waveform will be sampled and the algorithm will be implemented on a digital system. Therefore, the PLL algorithms must be discretized. The integration blocks were discretized with the forward Euler method such that

$$y(k+1) = x(k)T_s + y(k) \tag{3.3}$$

where T_s is the step size, $x(k)$ is the input and $y(k)$ is the output of the integrator. The step size is the inverse of the sample frequency, F_s. It is seen that the output is proportional to the step size which is the inverse of the sample frequency. The forward Euler method will in general give underestimates when the input is increasing and overestimates when it is decreasing [33]. The sample frequency will be in the kHz to tenths of kHz range. No significant error was seen between a 1 kHz sampling frequency and the continuous model.

There are therefore no significant errors for higher frequencies as these approach the continuous system. This is due to the low bandwidth of the PI controller. If the bandwidth instead is set to $2\pi100$ rad/s, the discrete system has a larger overshoot and longer settling time at a frequency step.

The transport delay PLL

The TD-PLL is the easiest implemented OSG method. The original algorithm uses a fixed delay. Consequently, if the input frequency deviates from the nominal frequency, the α- and the β- components will not be orthogonal. This will result in double-frequency oscillations in the input to the PI controller. This is illustrated in Figure 3.4 for a step from 50 to 51 Hz with $\xi = 0.7$, $\omega_n = 2\pi0.35$ rad/s and a sample frequency of 10 kHz. The phase error oscillates with a magnitude of 0.015 rad. With a step to 50.01 Hz the magnitude is 150 μrad. The estimated frequency could of course be fed back for update of the delay. The number of samples required for a 90° delay is found as

$$N_D = \frac{F_s}{4f_{in}} \tag{3.4}$$

where F_s is the sample frequency and f_{in} is the frequency of the input signal. N_D must be an integer and is therefore rounded off. An increased sample frequency gives a better frequency resolution. However, even for a sample frequency as high as 500 kHz, 50 and 50.01 Hz would be rounded to the same value of N_D. Therefore, the TD-PLL is not appropriate for the calibration system.

The inverse Park PLL

The IP-PLL algorithm implemented for continuous time simulations in Simulink is shown in Figure 3.5. For the IP-PLL algorithm, a cut-off frequency, ω_c, must be chosen for the low-pass filters. With $\xi = 0.7$, $\omega_n = 2\pi0.35$ rad/s and $\omega_0 = 2\pi49$ rad/s the value of ω_c was varied to find the minimum value that gives steady state within 10 seconds. A value for ω_c of $2\pi20$ rad/s was found suitable. With a higher bandwidth of the PI controller, the cut-off frequency can be lowered.

Since the low-pass filters main function is to avoid an algebraic loop, they can be replaced by delay blocks in the discrete implementation. However, for noise rejection they are still necessary. The continuous filter function was discretized with the zero-order hold (ZOH) function. The ZOH function holds the value of the continuous signal between each sample. The transformation looks as

$$H(s) = \frac{\omega_c}{s + \omega_c} \Rightarrow H(z) = \frac{b}{z + a} \tag{3.5}$$

where a and b are the coefficients of the discretized filter. With $\omega_c = 2\pi20$ rad/s and $F_s = 10$ kHz the transformation yields $b \approx 0.01249$ and $a \approx$ -0.9875 [34]. Comparison of the continuous and discretized filters is shown in Figure 3.6. No significant error was seen in the outputs of the discretized compared to the continuous model.

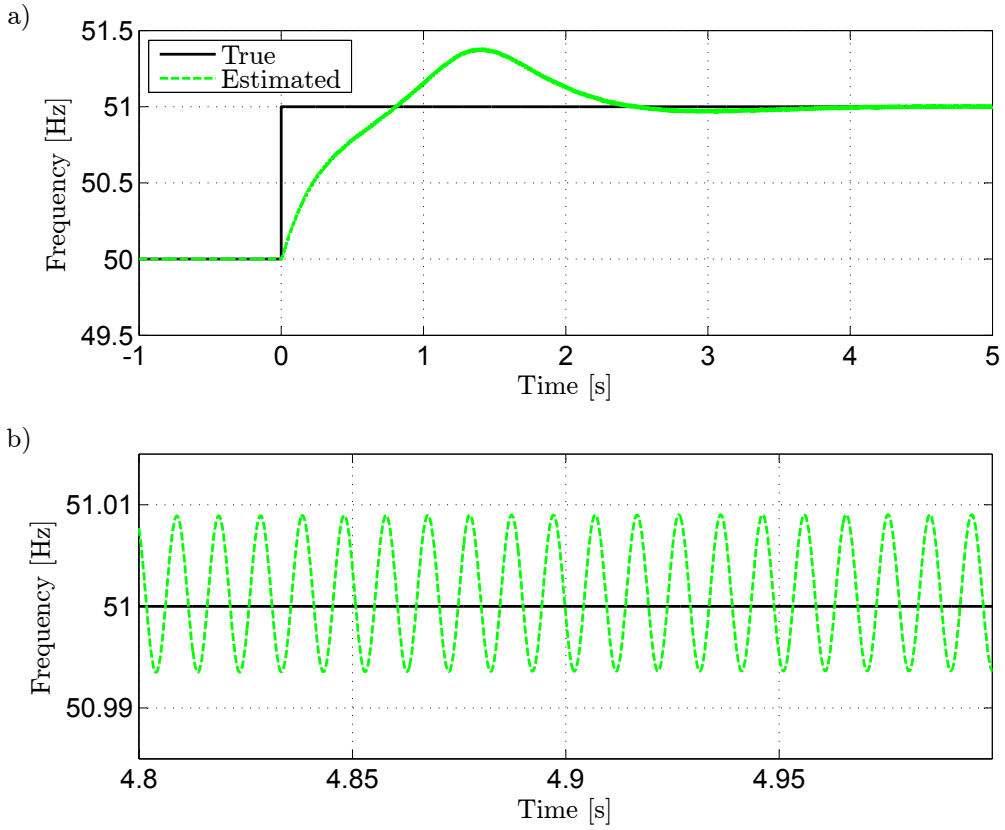


Figure 3.4 Frequency step with the TD-PLL for a constant delay input. *a)* Transient state. *b)* Close-up of oscillations in estimated frequency.

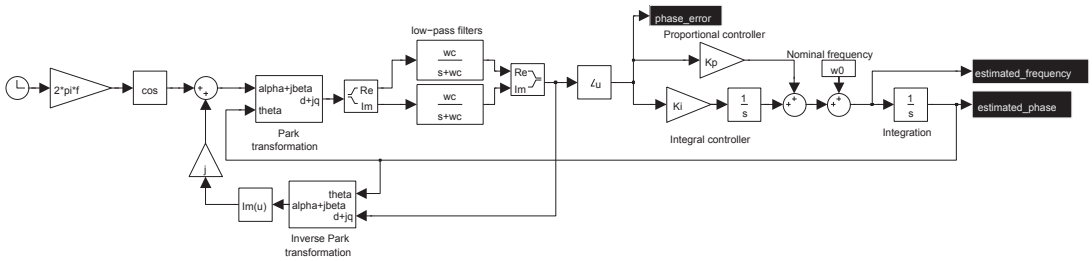


Figure 3.5 Block diagram of the IP-PLL as implemented in SIMULINK® for continuous time simulations.

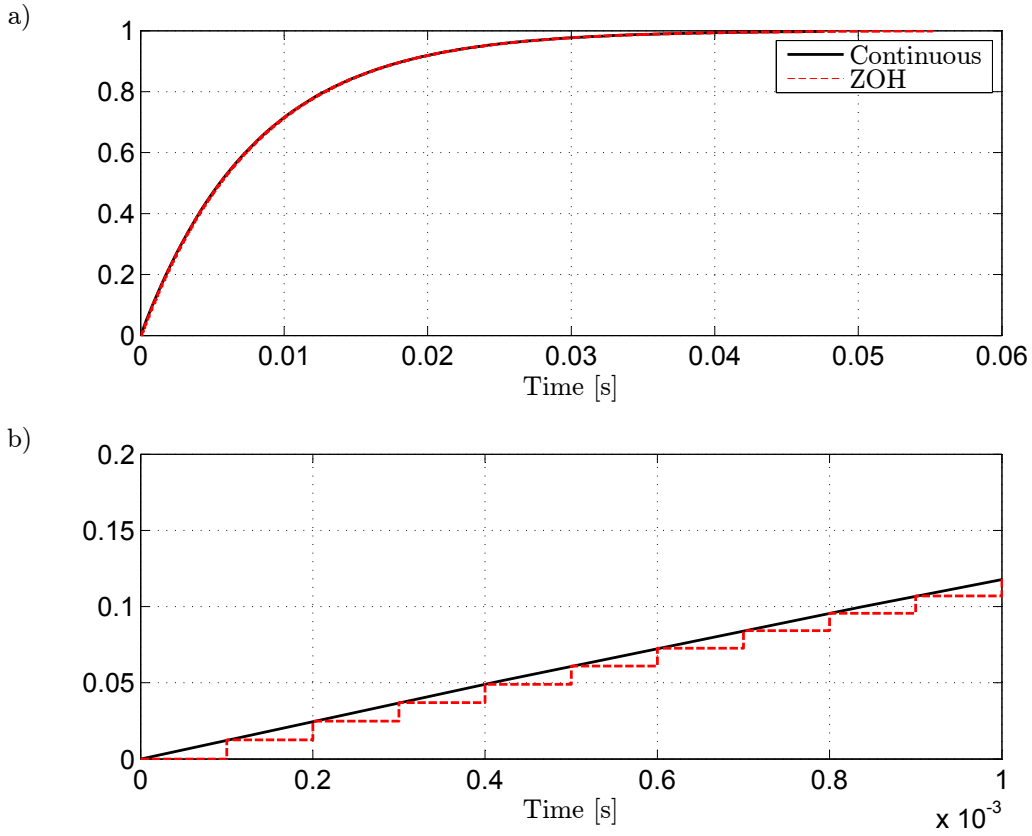


Figure 3.6 Step response of continuous and ZOH-discretized low-pass filter. *a)* Step response. *b)* Close-up of discrete steps.

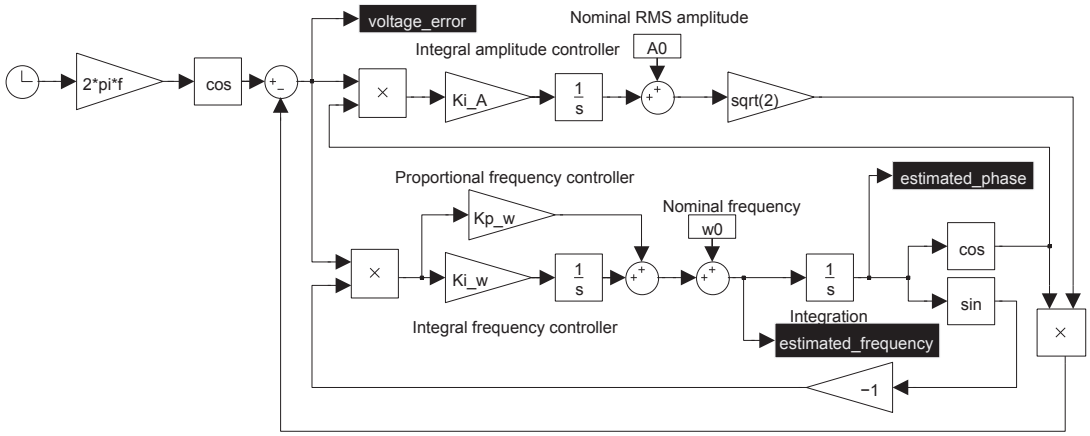


Figure 3.7 Block diagram of the E-PLL as implemented in SIMULINK® for continuous time simulations.

Table 3.1 KF phase estimation settings for tracking of noisy input signal with added harmonics as described by (3.6).

Parameter		Value
A_k		1
w_k	[rad/s]	$2\pi 50$
T_s	μs	100
$\hat{\boldsymbol{x}}_{\mathbf{0}}$		$[1\ 0]^T$
$\boldsymbol{P}_{\mathbf{0}}$		$1\boldsymbol{I}_{2x2}$
$\boldsymbol{Q}$		$0\boldsymbol{I}_{2x2}$
$\boldsymbol{R}$		$0.1^2\boldsymbol{I}_{1x1}$

3.1.2 The enhanced PLL

The E-PLL algorithm implemented for continuous time simulations in SIMULINK® is shown in Figure 3.7. As the transfer function for the phase estimation is similar to that of the SRF-PLL, the same values of ξ and ω_n are used. Additionally, a value for the integral gain of the amplitude estimation must be set. The integral gain was varied to find steady state within 10 seconds. A_0 was set 2 % below its nominal value which is the same offset used for the frequency. An integral gain for the amplitude estimation that is equal to the integral gain of the phase estimation branch proved to be suitable for this case.

3.1.3 The Kalman filter PLL

To illustrate the power of the KF algorithm, a very noisy signal with added harmonics was used as an input to the system. The input signal was described as

$$y(k) = 1\sin(2\pi 50t(k)) + \frac{1}{3}\sin(2\pi 150t(k)) + \frac{1}{5}\sin(2\pi 250t(k)) + v(k) \tag{3.6}$$

where $t(k)$ is the time vector with a time step of 100 μs and $v(k)$ is normally distributed noise with a standard deviation of 0.1. This gives a signal-to-noise ratio (SNR) for the fundamental component of 20 dB. The filter settings are presented in Table 3.1. The first three rows are the model parameters, set equal to that of the fundamental component. The initial guess, $\hat{\boldsymbol{x}}_{\mathbf{0}}$, equals to $\frac{\pi}{2}$. The initial value of the state covariance, $\boldsymbol{P}_{\mathbf{0}}$, is set high to indicate large uncertainty. The process covariance matrix, $\boldsymbol{Q}$, is set to zero. The measurement covariance matrix, $\boldsymbol{R}$, is set to equal the added noise. The values set for $\boldsymbol{Q}$ and $\boldsymbol{R}$ makes the output of the filter to mainly depend on the model as compared to the measurement. Since these are always defined by a certain magnitude times the unity matrix, only $|\boldsymbol{Q}|$ and $|\boldsymbol{R}|$ will be used further on. The results are shown in Figure 3.8. *a)* shows comparisons of waveforms for the first 100 ms. The initial guess is incorrect but after some short transients the estimation starts following the fundamental component. *b)* shows the phase error from 1 to 100 s. The error is below 100 μrad from about 35 seconds and onwards.

It seems very promising that such a low error is reached for an extremely noisy input that also has harmonics that are not even accounted for in the model. However, this was based on exact knowledge of the fundamental component frequency. Any change in the

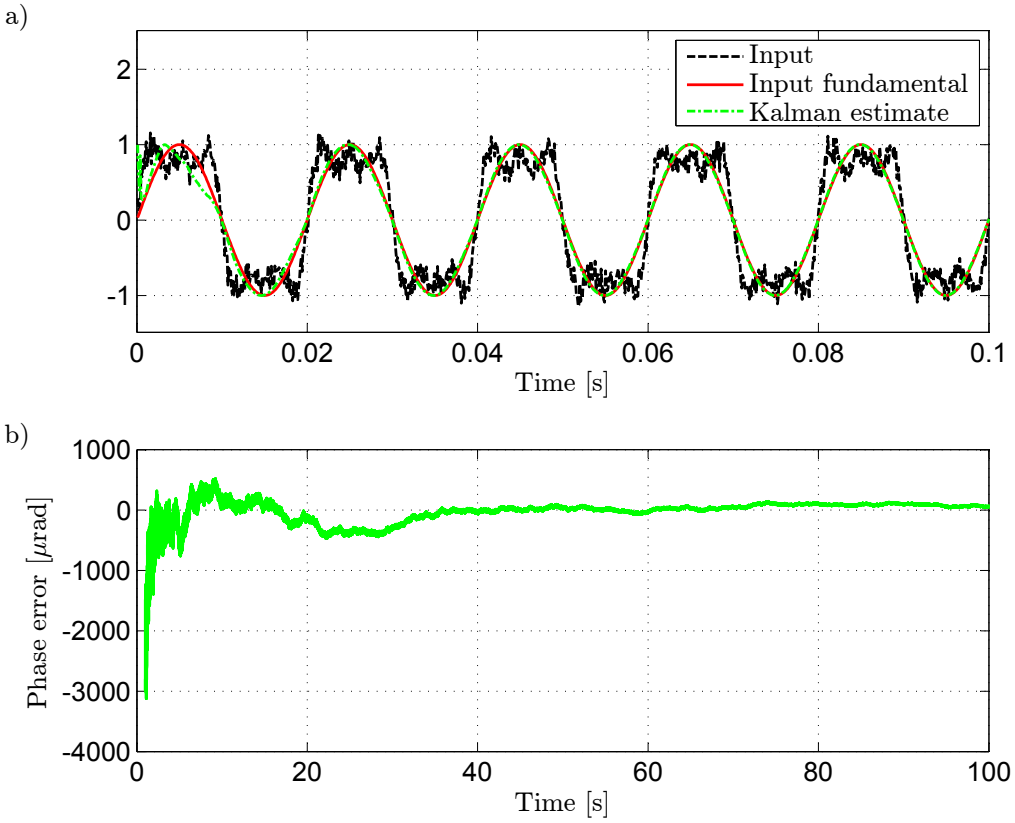


Figure 3.8 Kalman filter estimation on very noisy input signal with significant low-order harmonics and process covariance set to zero. *a)* Input, input fundamental and estimated signal in transient state. *b)* Phase error.

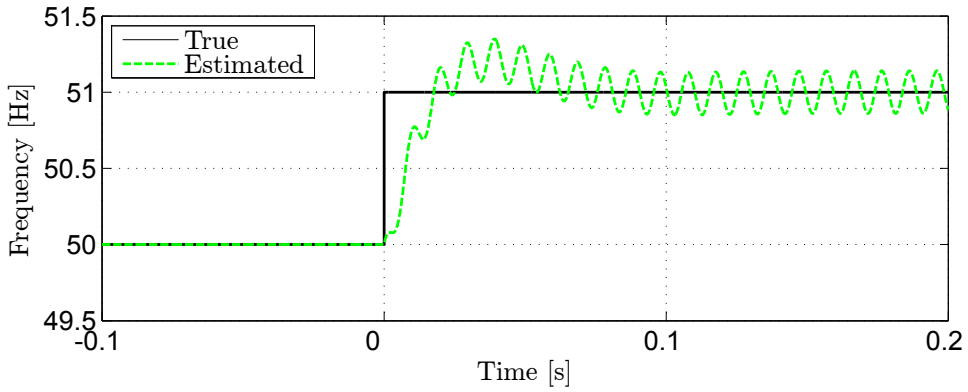


Figure 3.9 Typical behavior of the KF phase estimator for non-zero process noise covariance with input-frequency deviating from the model-frequency.

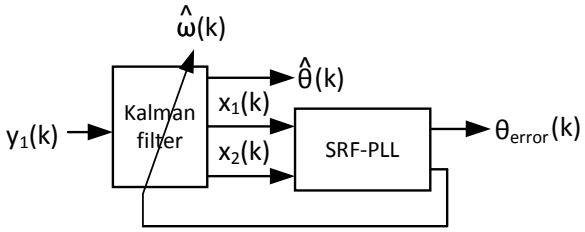


Figure 3.10 Kalman filter phase estimator with frequency update by SRF-PLL to form the KF-PLL.

input frequency gives a steadily increasing error. This is natural since the output is fixed at 50 Hz due to the magnitude of the process noise covariance matrix being zero, not allowing any deviation. Depending on the time required for phase stability and the allowed error, some frequency offset can be allowed.

Frequency update

If $|\boldsymbol{Q}|$ was increased from zero, the estimation frequency could adapt as well. Figure 3.9 shows the typical behavior of a frequency deviation between the model and the input signal with non-zero magnitude of both noise covariance matrices. The delayed step response is due to the frequency being estimated by a PLL. The input contains no noise to clearly illustrate the harmonic oscillations that occur in the estimated frequency. The magnitude of the oscillations increases with the magnitude of the deviation. It is thus clear that if the model frequency was updated, the oscillations would diminish. As mentioned in Chapter 2.1.4, this can be solved by using a PLL feedback structure on the KF output.

Figure 3.10 illustrates the concept of model-frequency update by using the SRF-PLL and forming the KF-PLL. The SRF-PLL suits well since the output from the KF phase estimator consists of two orthogonal components. This also adds the benefit that an estimation of the phase error can be taken from the Park transformation block. Also the estimated phase could be taken from the SRF-PLL but as this is a second estimation, it is less accurate than calculating the phase from the orthogonal components. Figure 3.11 shows the real phase error and the phase error estimated by the SRF-PLL for a noisy input. It can be seen

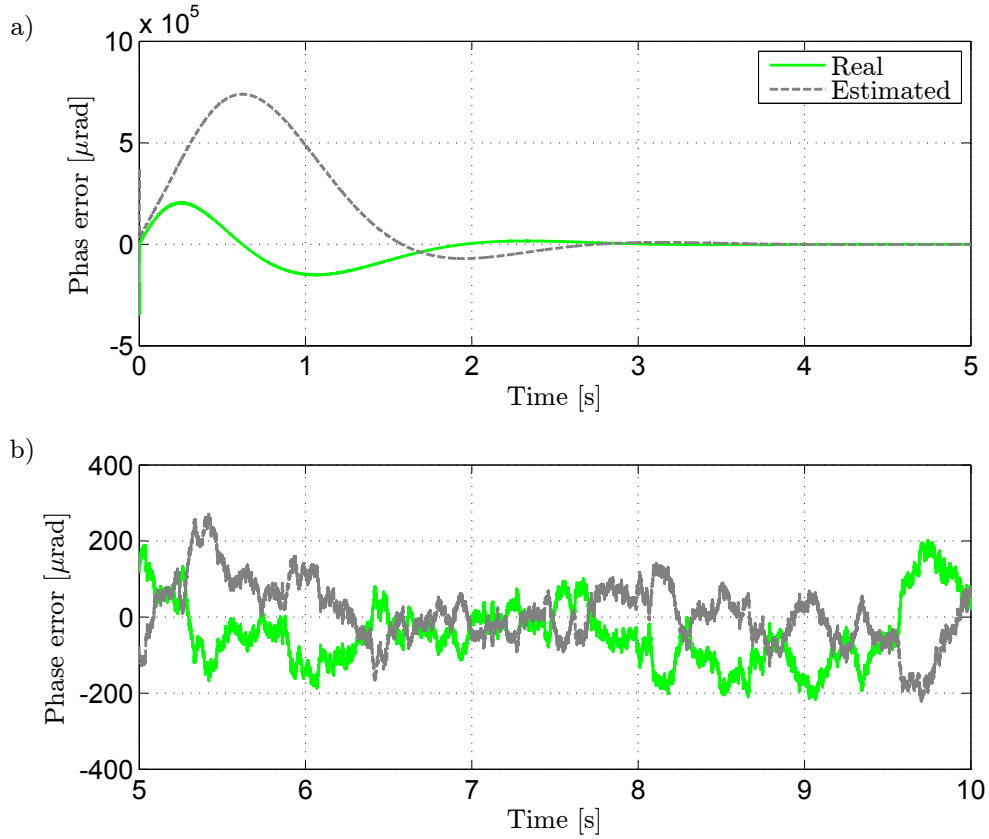


Figure 3.11 Real phase error and phase error of SRF-PLL in the KF-PLL with a noisy input. *a)* Transient state. *b)* Steady state

that for the transient part, the error of the SRF-PLL is a delayed and amplified version of the real error. The steady state error is a mirror of the real error. The estimated error is smaller in magnitude but still comparable to the real value. The code used for simulation in MATLAB® is found in Appendix A.1.

If the frequency update is used with $|\boldsymbol{Q}|$ set to zero, the system becomes unstable. At a frequency step, the frequency does change but only to oscillate closely around the model frequency. This suggests that it should work for small frequency deviations. However, the transients of the frequency estimation pulls the system out of stability.

Tuning of parameters

Tuning of the KF algorithm is the hardest to perform among the investigated phase tracking systems. In addition to the initial values and covariance matrices of the KF, the damping and the bandwidth must be set for the frequency updating SRF-PLL. The ideal model parameters may be chosen from the nominal values of the input signal and the time step is decided by the sample frequency.

The contribution from the error between the estimate and the measurement is set by a gain factor, seen in (2.29). The gain factor is multiplied with the error and added to the prediction from the previous iteration. The gain factor is

$$\boldsymbol{G}_k = \boldsymbol{P}_k\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{P}_k\boldsymbol{C}^T + \boldsymbol{R})^{-1} \tag{3.7}$$

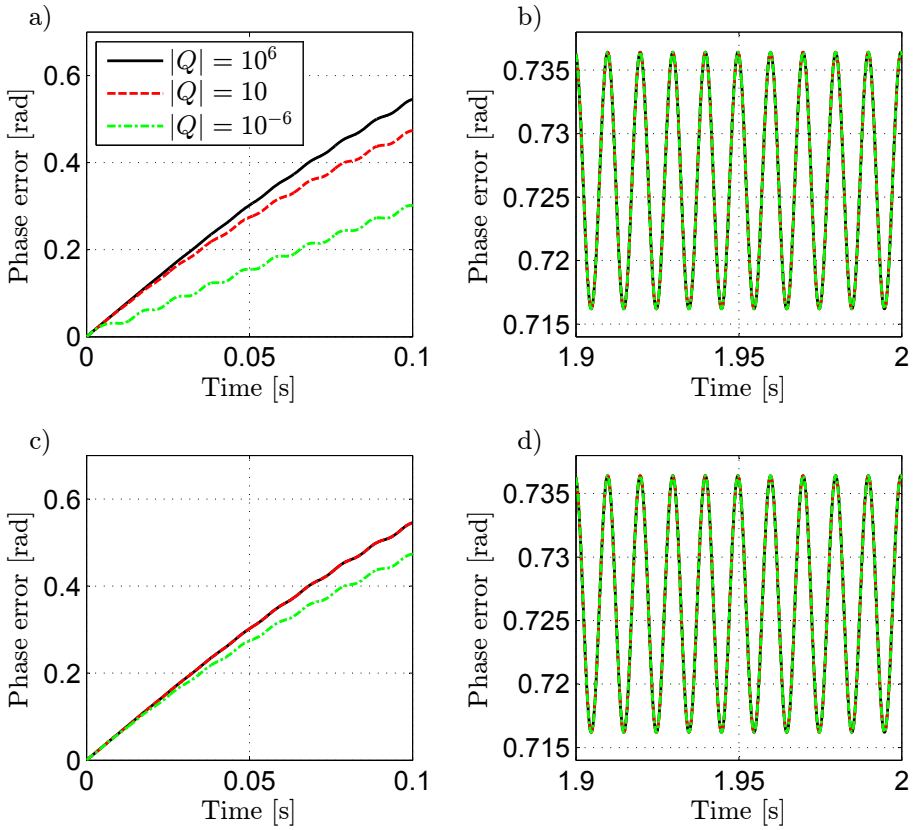


Figure 3.12 Behavior of the Kalman filter for different values of $|\boldsymbol{Q}|$ as related to $|\boldsymbol{P}_0|$. The covariance ratio, $|\boldsymbol{Q}|/|\boldsymbol{R}|$ is constant and $\omega_0 = 49$ πrad/s with an input frequency at 50 Hz. *a)* Transient state for $|\boldsymbol{P}_0| = 10^3$. *b)* Steady state for $|\boldsymbol{P}_0| = 10^3$. *c)* Transient state for $|\boldsymbol{P}_0| = 10^{-3}$. *d)* Steady state for $|\boldsymbol{P}_0| = 10^{-3}$.

It is seen that if $|\boldsymbol{R}|$ is decreased, the gain is increased. An increased gain will result in a bigger error contribution to the estimated state. According to (2.32), a decreased $|\boldsymbol{Q}|$ will decrease the state covariance matrix, $\boldsymbol{P}$, which will decrease the gain. By varying the respective values but keeping the ratio, the steady state results are found to be equal. This is shown in Figure 3.12, without the PLL and with a 1 Hz offset between input and output frequency, for three different values of $\boldsymbol{Q}$ with a fixed ratio of $|\boldsymbol{Q}|/|\boldsymbol{R}|$. However, the value of $|\boldsymbol{Q}|$ relative to the initial state covariance $\boldsymbol{P_0}$ is seen to have importance for the transient state. The transient states are shown in *a)* for $|\boldsymbol{P}_0| = 10^3$ and *c)* for $|\boldsymbol{P}_0| = 10^{-3}$ and the results are seen to be different. The steady states are shown in *b)* for $|\boldsymbol{P}_0| = 10^3$ and *d)* for $|\boldsymbol{P}_0| = 10^{-3}$ and the results are seen to be the same.

To find appropriate values of $|\boldsymbol{Q}|$ and $|\boldsymbol{R}|$, the PLL was kept disconnected. The model frequency was set to 49 Hz and the sample frequency to 10 kHz. With the input at 50 Hz, the output frequency must oscillate around 50 Hz for the frequency update to work. The aim was then to find the minimum ratio $|\boldsymbol{Q}|/|\boldsymbol{R}|$ for this to happen. This minimizes the steady state gain and thus maximizes the noise rejection. At the same time the time to steady state is extended.

Steady state was defined as a phase error below $|1|$ μrad as was done for the other algorithms. The covariance ratio of 10^{-7} was found to be the lowest possible value. Steady

Table 3.2 Summary of the phase estimation algorithms with settings of parameters obtained in the respective tuning procedures.

		IP-PLL	**E-PLL**	**KF-PLL**
ξ		0.7	0.7	0.7
ω_n	[rad/s]	$2\pi 0.35$	$2\pi 0.35$	$2\pi 0.36$
ω_c	[rad/s]	$2\pi 20$		
$K_{I,A}$			$K_{I,\omega}$	
$\lvert\boldsymbol{Q}\rvert$				10^{-6}
$\lvert\boldsymbol{R}\rvert$				1

state was met well before the 10 s limit. For test together with the PLL, $|\boldsymbol{P}_0|$ was set to 10, $|\boldsymbol{Q}|$ to 10^{-6}, ξ of the PLL to 0.7 and then the PLL bandwidth was tuned for reaching steady state right within 10 s. The nominal frequency of the PLL was set to 50 Hz. The covariance ratio of 10^{-7} was proved too low for finding steady state within 10 s. It had to be increased one decade to 10^{-6} to work with the PLL. $\omega_n = 2\pi 0.36$ rad/s of the PLL proved to be the lowest possible bandwidth for steady state within 10 seconds, for this selection of covariance ratio.

3.1.4 Comparison

Out of a total of four different phase tracking algorithms, three seemed suitable for the calibration system. The simplest algorithm, the TD-PLL, proved to give phase errors at 150 μrad at deviations of only 0.01 Hz from a nominal frequency of 50 Hz and it was therefore disregarded. The IP-PLL, the E-PLL and the KF-PLL are however promising. All algorithms have been tuned for minimum bandwidth but still in order to reach steady state within 10 seconds. Steady state was defined as the phase error being below $|1|$ μrad, without going out of the bounds again. The initial frequencies have been set at 49 Hz and a noise-free signal at 50 Hz has been used as input. For the E-PLL, the integral gain of the amplitude estimation was tuned with the initial amplitude set to 98 % of the input. These errors might seem large and better estimations can probably be made, at least for the frequency. However, this serves as a common platform for proper comparison between the algorithms. A summary of the algorithms with settings obtained in the tuning procedure are seen in Table 3.2. It was shown that the initial guesses made for the KF-PLL are insignificant for the steady state results.

The IP-PLL is the simplest algorithm to tune. Additional to the damping factor, bandwidth and cut-off frequency of the low-pass filters must be set. The E-PLL requires choice of nominal amplitude and integral gain for the amplitude estimation. If this is set too far from the actual amplitude, the result will be bad. The KF-PLL is the hardest to tune. It was however shown that the focus can be put on the ratio between the magnitudes of the process covariance and the measurement covariance matrices, denoted $\boldsymbol{Q}$ and $\boldsymbol{R}$ respectively.

Different steady state errors have been investigated. Therefore, phase jumps, frequency jumps, amplitude drops and similar phenomena have not been considered. A sudden change like that would interrupt the calibration sequence. In addition, there will be a peak in the estimation error that in many cases extends beyond 50 μrad.

Simulations have been run for 30 seconds with the comparison performed for the last 20. Nominal values have always been unity amplitude and 50 Hz frequency. Discretized algorithms have been used with the sample frequency at 10 kHz. For the oscillating disturbances, a frequency of 0.05 Hz was used. This gives one period over the 20 seconds that are considered. Magnitudes of the errors have been chosen such that at least one of the algorithms gives errors that have a standard deviation below or very close to 50 μrad. In all figures, the best PLL is written with bold text in the legend. *a)* shows the phase errors over time and *b)* shows the error spreads. The error spreads are illustrated by the mean and standard deviation.

Noise

Noise may originate from the measurement equipment and quantization in the analogue-to-digital conversion (ADC). It can also originate from high-order harmonics present due to conducted or radiated emission.

Normally distributed noise with a standard deviation of approximately 0.22 % of the fundamental amplitude was added to a 50 Hz signal. This is equivalent to an SNR of 53 dB. Exactly the same noise pattern was used for the three algorithms. Figure 3.13 shows the results. The standard deviation of the phase errors are 48.0 μrad, 52.6 μrad and 66.0 μrad for the IP-PLL, the E-PLL and the KF-PLL respectively. The IP-PLL is slightly better than the E-PLL. The bandwidth of the PI controllers of the IP-PLL and the E-PLL are set exactly the same. The figure clearly shows the likeness between the algorithms and the effect of the low-pass filters of the IP-PLL.

Frequency variation

The frequency may vary due to varying load in the grid. Oscillations will come from when the system stabilizes in a new steady state. A pure 50 Hz sinusoidal was frequency modulated to vary with a magnitude of 4 mhz at a frequency of 0.05 Hz. The results are presented in Figure 3.14. The standard deviation of the phase errors are 1150 μrad, 811 μrad and 48.5 μrad for the IP-PLL, the E-PLL and the KF-PLL respectively. Only the performance of the KF-PLL is acceptable.

Amplitude variation

If the frequency of a generator varies, so does the induced voltage and thus the amplitude over the constant load. Amplitude variations may also result from local load variations that does not affect the frequency of a large grid. With oscillations of 20 % of the input amplitude, the algorithms perform similarly. Results are shown in Figure 3.15. The standard deviation of the phase errors are 73.2 μrad, 57.1 μrad and 50.7 μrad for the IP-PLL, the E-PLL and the KF-PLL respectively. The KF-PLL was best by a small difference.

Dc offset

Just as noise may occur from the measurement and quantization in the ADC block, an average dc error may occur. With a dc offset of 1.8 % of the input amplitude, only the IP-PLL performed well. Results are shown in Figure 3.16. The standard deviation of the

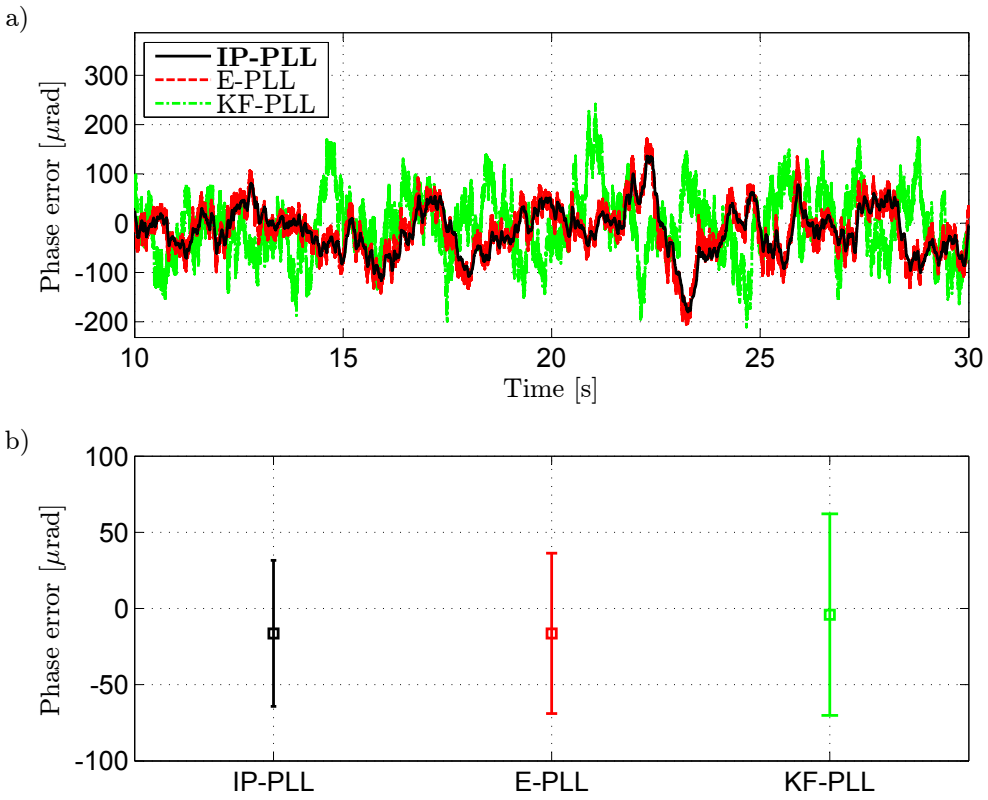


Figure 3.13 Comparison of steady state results between the IP-PLL, E-PLL and KF-PLL for an SNR = 53 dB. *a)* Phase error over time. *b)* Error spread showing mean value and standard deviation.

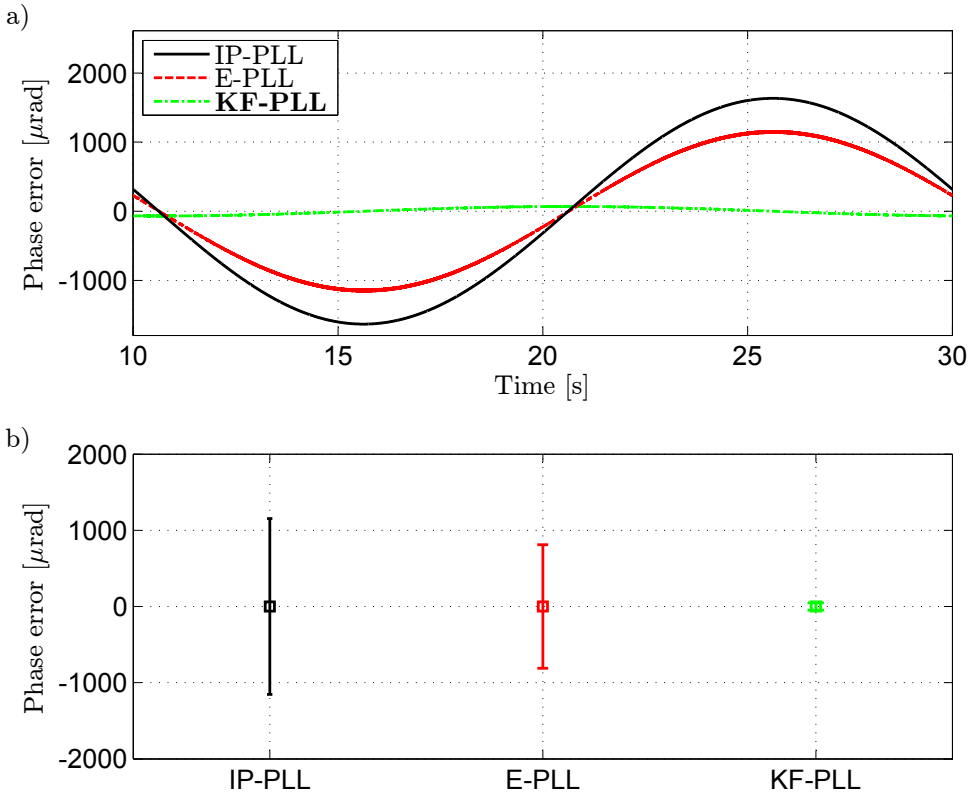


Figure 3.14 Comparison of steady state results between the IP-PLL, E-PLL and KF-PLL for frequency deviations of 7 mHz at a frequency of 50 mHz. *a)* Phase error over time. *b)* Error spread showing mean value and standard deviation.

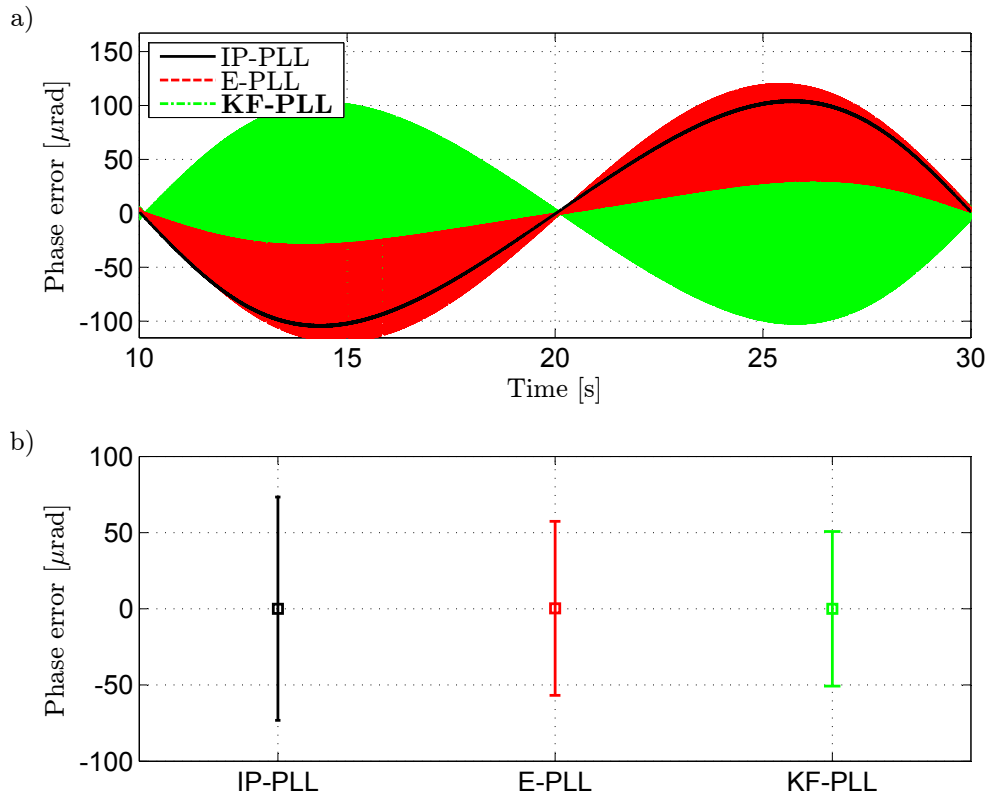


Figure 3.15 Comparison of steady state results between the IP-PLL, E-PLL and KF-PLL for amplitude deviations of 20 % at a frequency of 50 mHz. *a)* Phase error over time. *b)* Error spread showing mean value and standard deviation.

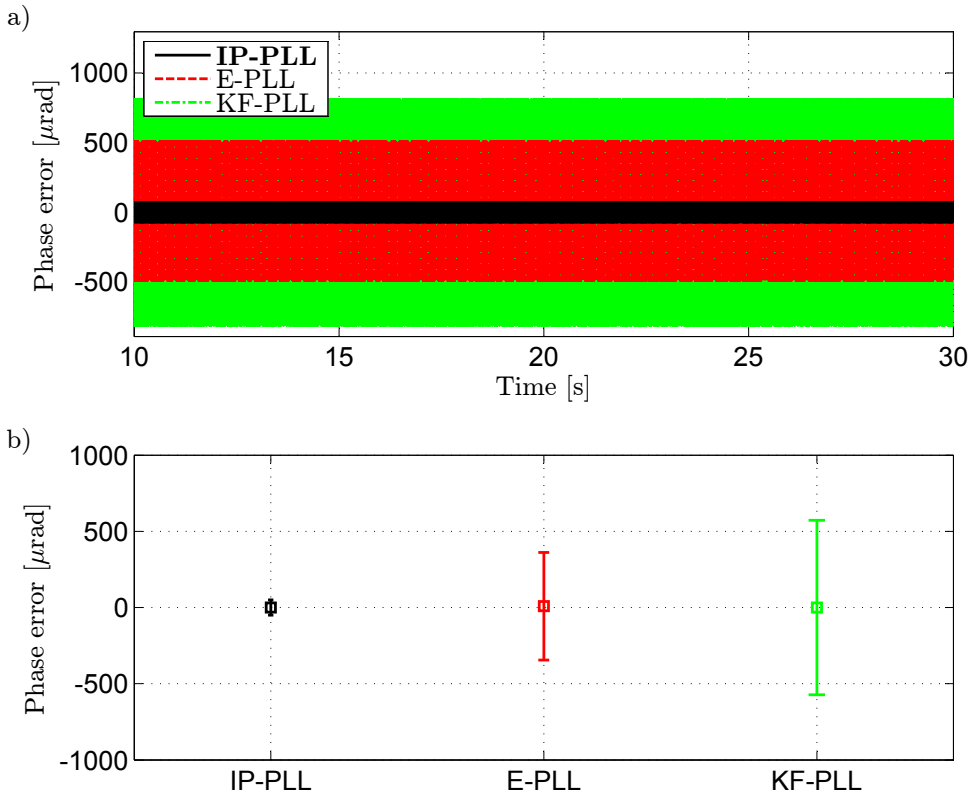


Figure 3.16 Comparison of steady state results between the IP-PLL, E-PLL and KF-PLL for a dc offset of 1.8 %. *a)* Phase error over time. *b)* Error spread showing mean value and standard deviation.

phase errors are 50.0 μrad, 353 μrad and 573 μrad for the IP-PLL, the E-PLL and the KF-PLL respectively. The dc offset produces 50 Hz oscillations in the Park transformation and these are damped out by the low-pass filters [8].

As mentioned in Chapter 2.1.3, an integrating controller can be added to the E-PLL for estimation of a dc offset in the input. Block diagram of the improved E-PLL is shown in Figure 3.17. The same solution could be added to the IP-PLL. The integral gain was tuned to the same value as for the amplitude estimation. This gives zero error with a dc offset at 100 % of the input amplitude. The signal model of the Kalman filter may be expanded to

$$\begin{aligned} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{k+1} &= \begin{bmatrix} \cos(\omega_k T_s) & \sin(\omega_k T_s) & 0 \\ -\sin(\omega_k T_s) & \cos(\omega_k T_s) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_k + \boldsymbol{v}_k \\ \boldsymbol{y}_k &= \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_k + \boldsymbol{w}_k \end{aligned} \tag{3.8}$$

where x_3 is the estimated dc component. This structure also eliminates a 100 % dc offset, without changing any other parameters.

These modifications have insignificant impact on the performance for the other errors for the KF-PLL. For the E-PLL, only the performance during amplitude variation was

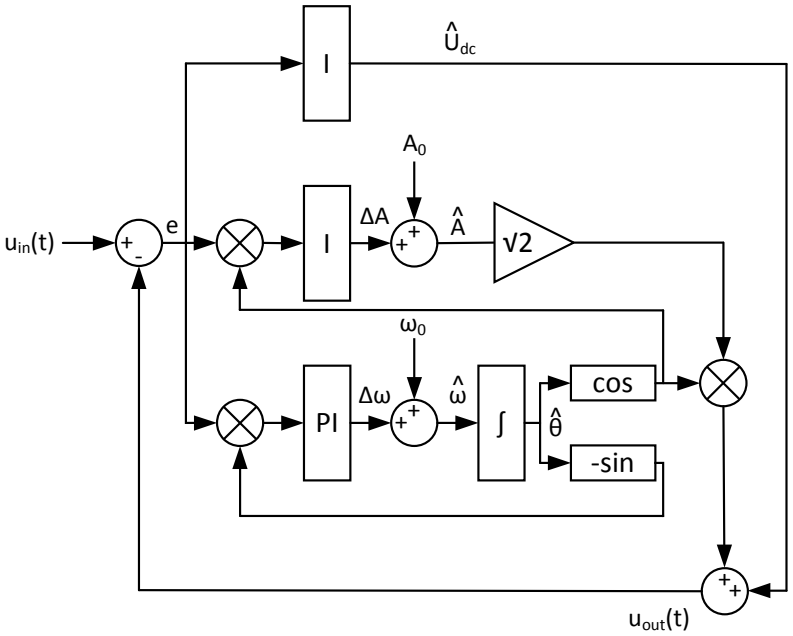


Figure 3.17 The E-PLL with added dc offset estimation.

affected. The standard deviation of the phase error increased to 260 μrad.

Low-order harmonic

Low-order harmonics may result from saturation of transformers or be coupled to the grid from diode rectifiers. A third harmonic was added to the input with an amplitude at 10 % of the fundamental component. The results are presented in Figure 3.18. The standard deviation of the phase errors are 50.3 μrad, 776 μrad and 889 μrad for the IP-PLL, the E-PLL and the KF-PLL respectively. The dc estimation were not included in neither the E-PLL nor the KF-PLL. However, with these included, there was no significant differences. It is likely that the low-pass filters of the IP-PLL are the main reasons of its superior performance. This suggests that a similar structure should be added for the other algorithms.

Low-pass filters were tested on the error signals on both the E-PLL and the KF-PLL. The low-pass filter dampened the harmonic and thus decreased the error spread. By including estimation of the harmonic in the algorithms, even a harmonic at 100 % amplitude of the fundamental could be completely eliminated. As described in Chapter 2.1.3, just another amplitude and phase estimation branch, equal to that for the fundamental, was added to the E-PLL. The nominal frequency is changed to the frequency to be tracked. The control parameters must be scaled properly with the nominal amplitude. It can be hard to estimate the amplitude of harmonics. Simulations shows that the convergence speed is sometimes prolonged for an inaccurate nominal amplitude for the E-PLL. For the KF-PLL, the algorithm is also expanded with the same components used for the fundamental but with changed nominal frequency. As the KF itself adjusts the amplitude, there is no convergence problem occurring from improper nominal settings.

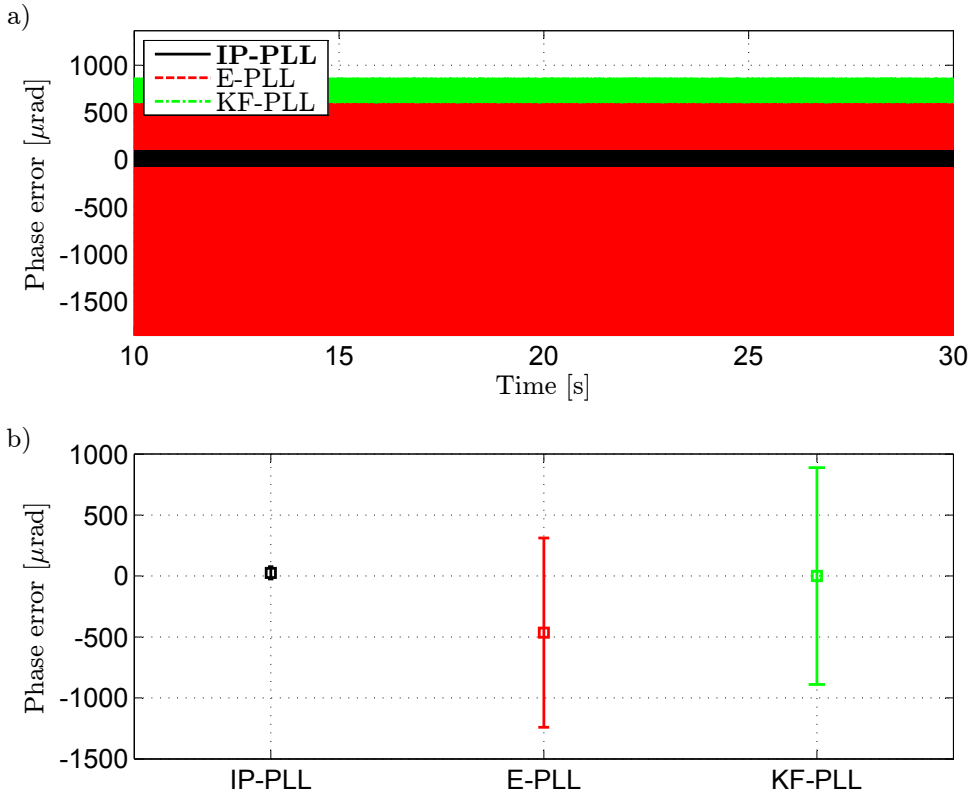


Figure 3.18 Comparison of steady state results between the IP-PLL, E-PLL and KF-PLL for a third harmonic at 10 % of fundamental amplitude. *a)* Phase error over time. *b)* Error spread showing mean value and standard deviation.

Summary

For the unmodified algorithms, targeted at the fundamental component, the IP-PLL gave the best results overall. Especially for the dc offset and the inclusion of a third harmonic. This was thought to be largely due to the low-pass filters. With dc estimation branches added for E-PLL and KF-PLL, the dc offset error was completely eliminated. However, the error for the amplitude variations were increased for the E-PLL. With low-pass filters added on the error of the E-PLL and KF-PLL, the results with the third harmonic were improved but still worse than for the IP-PLL. The errors were completely eliminated by inclusion of the harmonic in the estimation. The harmonics present in the input signal may be hard to estimate and including multiple harmonics makes the implementation complex. This function should be added with knowledge of the major harmonics in the grid were the algorithm will be used. Additionally, for the E-PLL it is important that the nominal amplitude is set close to the actual.

The KF-PLL gave the best results for the frequency oscillations. As the reason for this was not clear, there was nothing that could be copied to the other algorithms for improvement of their response for frequency oscillations. When comparing the real error with the estimated error of the SRF-PLL in the KF-PLL, only the input with frequency oscillations gave different results. For this input error, the estimated phase error is much larger than the real error. It was shown above that a system with the transfer function of the SRF-PLL does not perform well for a sinusoidal input with frequency oscillations. The KF algorithm itself adapts well to the frequency oscillations and therefore this error is present in the input to the SRF-PLL.

3.2 Inverter

The inverter must very accurately deliver an output current with a phase shift referred to the input to the PLL. The maximum current amplitude should be 2 kA and it must be accurate to 20-50 ppm. The inverter is based on the full-bridge converter circuit shown in Figure 2.10, that will generate a low current and high voltage which is connected to a transformer to boost the current. This is the same method as is used in the previous system.

3.2.1 Load measurements

At the customer's site the cables forming the load cannot always be oriented freely, depending on the installation of the DUT. This may result in relatively high inductance. Therefore, measurements were done with cables forming different loops. The same transformer as in the previous version of the calibration system was used. It has a rating of 250 V and 21.6 A on the primary side and 1.8 V and 3 kA on the secondary. The secondary side has to be wound with an external cable and the ratings are for one turn. The transformer and the secondary winding are shown in Figure 3.19. The cable was too stiff for tight fixation to the core. However, this was the same setup used with the previous calibration system. The increased winding area increases the series inductance of the complete circuit.

The measurement setup is shown schematically in Figure 3.20. R_T and L_T represent the short-circuit impedance of the transformer. R_{CL} and L_{CL} are the impedance parameters of the cable loop. A 50 Hz voltage source with variable amplitude was connected to the

Figure 3.19 Transformer secondary winding of impedance measurement.

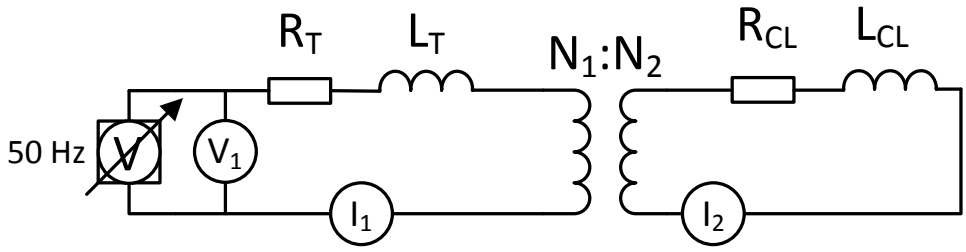


Figure 3.20 Measurement schematic for impedance estimation of different cable loops.

primary side of the transformer. The primary side voltage and current was measured with a power quality analyzer that gave values of power and harmonic distortion. At the secondary side, only the current could be measured.

The cable loops were made up of 240 mm^2 copper cable sections of 610 cm in length. Four orientations were used. Two with the cable oriented in a circle, with one and two cable sections. Two with a tight cable loop, with one and two cable sections. Figure 3.21 shows the circular orientation for one cable section. Figure 3.22 shows the tight orientation for one cable section.

Data was not available on the impedance parameters of the transformer and no separate measurements were done to estimate them. However, the resistance should have doubled when adding the second cable section and equally for the inductances of the tight cable orientations. The short-circuit impedance of the transformer could therefore be found by solving

$$\begin{cases} Z_{tight1} &= Z_T + Z_{CL,tight} \\ Z_{tight2} &= Z_T + 2Z_{CL,tight} \end{cases} \tag{3.9}$$

where Z_{tight1} and Z_{tight2} are the measured impedance values with one and two sections

Figure 3.21 Circular orientation for maximum inductance.

Figure 3.22 Tight orientation for minimum inductance.

Table 3.3 Estimated parameters of the different cable loop setups.

		Short		**Long**	
		Tight	*Circle*	*Tight*	*Circle*
R_{CL}	[mΩ]	0.85	0.85	1.71	1.71
L_{CL}	[μH]	2.3	6.0	4.6	14

of tight cable orientation respectively on the secondary side, Z_T is the transformer short-circuit impedance and $Z_{CL,tight}$ is the impedance of one section of tight cable orientation. The transformer short-circuit impedance parameters were estimated to be a resistance of 1.5 mΩ and an inductance of 1.5 mH. The estimated cable loop impedance parameters are presented in Table 3.3.

The resistance per meter was estimated to 0.14 mΩ/m from the measurements. The analytical value in room temperature is found through

$$R = \rho \frac{l}{A} \tag{3.10}$$

where ρ is the resistivity of copper, l is the length and A is the area of the conductor [35]. This gives a resistance of 0.070 mΩ/m. The cable got very hot due to the large currents. However, since the temperature coefficient for copper is only 0.00433 $^{\circ}\mathrm{K}^{-1}$, the temperature rise alone cannot solely be responsible for the mismatch. Significant resistance was probably added by the bolted connections of the cables. The inductance values were however close to analytical estimations. The inductance per meter of two parallel conductors carrying the same current in opposite directions is found as

$$L_{tight} = \frac{\mu_0}{4}\left(\ln\frac{d}{r} + \frac{1}{4}\right) \tag{3.11}$$

where d is the distance between the center of the conductors, r is the radius of the conductors and μ_0 is the permeability of free space [36]. Inputting the cable geometry gives the value of 0.38 μH/m which is exactly what the measurements gave. The value of a

single-turn circular coil can be estimated through

$$L_{circle} = \frac{\mu_0 R}{2}\left(\ln\frac{4R}{d} - \frac{1}{2}\right) \tag{3.12}$$

where R is the radius of the circle and d is the diameter of the cable [37]. This gives approximately 5.07 and 12.7 μH for the small and big cable loop respectively. These values are slightly below the results from the measurements.

In Figure 3.23, a comparison between all the four cable orientations is shown for four different turns ratios based on the estimated impedance values. The RMS values of the primary current in *a)* and the secondary current in *b)*, at 50 Hz, are shown. The rated primary side voltage of 250 V RMS was used. The limit for the primary side is the rated value of 21.6 A. The limit for the secondary side is the 2 kA level that is wanted for the calibration system. It is seen that only the small tight and loop orientations reaches 2 kA on the secondary side, but with too high primary side current.

To reach 2 kA on the secondary side, the minimum turns ratio must be

$$\begin{aligned} N_{min} &= \frac{I_s}{I_p} \\ &= \frac{2000}{21.6} \approx 92.6 \end{aligned} \tag{3.13}$$

With two of the same transformers in parallel supplying 1 kA each, the minimum turns ratio is 46.3. This is possible for all cable orientations except the large loop which is only slight below 1 kA at 46 turns. However, a longer cable might be required which would lower the current due to increased impedance. The loop does not necessarily have to be larger which would increase the inductance, but resistance will nevertheless be added. There are in total three of these transformers available. Even with additional cable sections, 2 kA should be possible to reach even if all of these are used.

3.2.2 Simulations

The transformer is the limiting component in the system, as was shown in Chapter 3.2.1. Three of these are required to reach the 2 kA output current. Therefore, a dc-link voltage of 354 V with an impedance for 91.6 A peak load current had to be used in the simulations. The simulation model was built up in PLECS, an electronic simulation package for SIMULINK®. This was done according to the circuit shown in Figure 2.10 but with an ideal, voltage-stiff dc-link. The simulations were performed in continuous time with saving of data and PWM comparison at a frequency of 1 MHz.

Currents

Figure 3.24 shows the typical currents over one output fundamental component period with an inductive load. *a)* show the collector current and forward current of an IGBT and a diode respectively. *b)* shows the total dc-link current. The dc-link current is seen to be at twice the output fundamental frequency. For calculation of the conduction losses, according to (2.41) and (2.42), the average and RMS currents must be known for the IGBT and diode respectively. These are presented in Table 3.4. The impedance has been matched to make the output current equal to the full load of the three transformers. The power factors 0.76

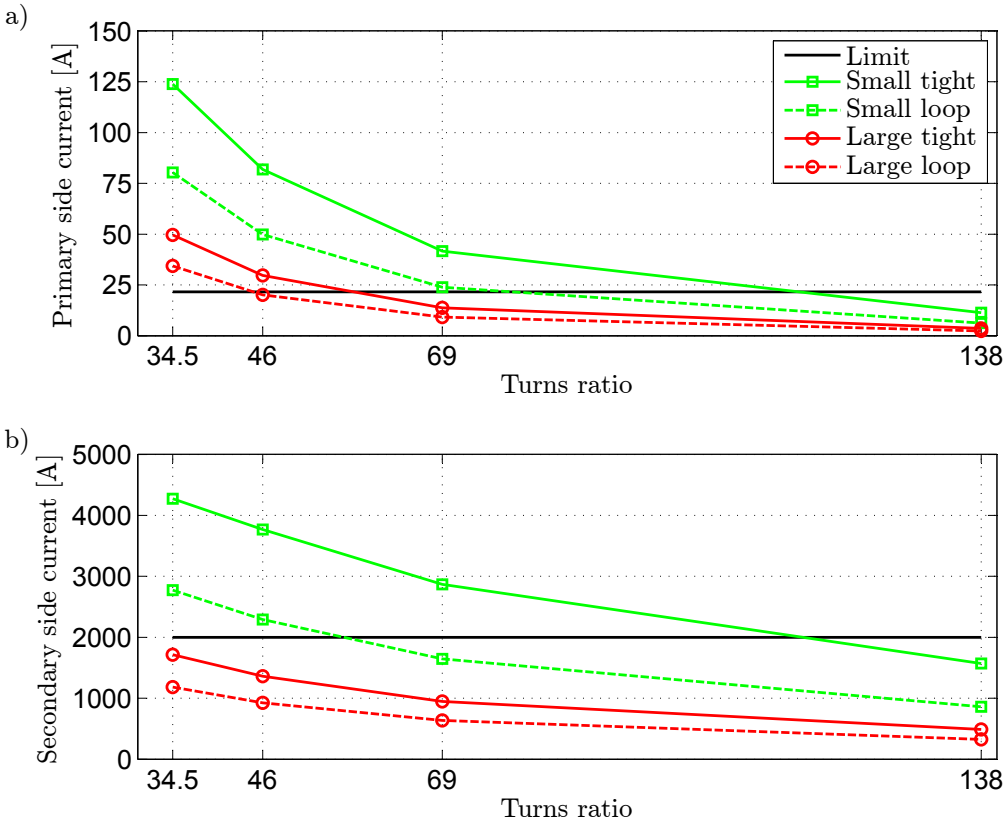


Figure 3.23 Transformer currents with the four measured cable loops for turns ratios equivalent to (from the right) one, two, three and four secondary side winding turns. *a)* Primary side current. *b)* Secondary side current.

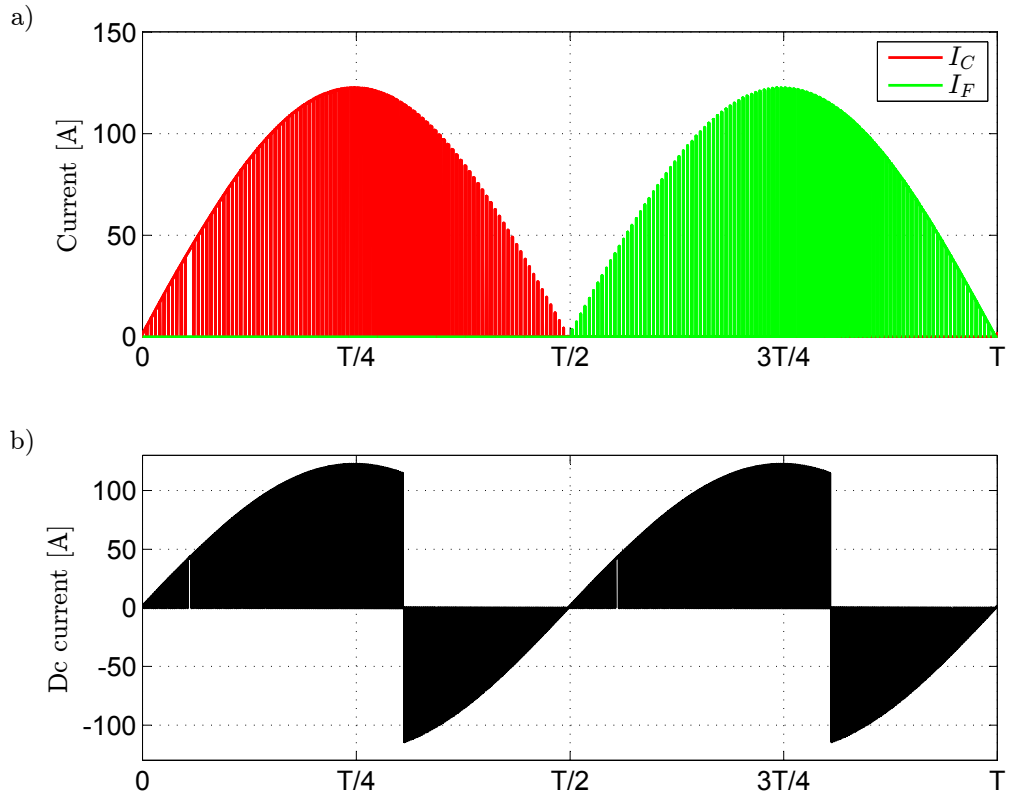


Figure 3.24 Typical shape of currents of the full-bridge converter for an inductive load. *a)* Collector current of IGBT and forward current of diode. *b)* Total dc-link current.

and 0.35 to represent the highest and lowest power factor of the measured cable loops. 0.1 and 0.9 were used as extreme points.

The values of the average and RMS of the total dc-link current, $I_{dc,tot} = I_{dc} + I_{Cdc}$, for different power factors are shown in Table 3.5. These are important for choosing appropriate dc-link capacitors [38]. It is seen that the 100 Hz component is the same for all power factors. This is natural since the load impedance does not change. With respect to the most significant frequency components, the dc current can be divided in to a dc component, a 100 Hz component and a higher frequency component related to the switching frequency. The level of the dc-component is decided by the power factor of the load and thus the active power demand [39] [40].

The harmonics content of a waveform can be quantified by the total harmonics distortion (THD) index. For currents, THD can be calculated as

$$THD = \frac{I_{harmonics}}{I_1} \tag{3.14}$$

where I_1 is the RMS value of the fundamental component and $I_{harmonics}$ is the RMS value of the higher-order harmonics. THD is calculated in the same way for voltage [23]. In the following subsections, the impact on the harmonics content on the output with different nonidealities is investigated. The power factor is always 0.35. For the ideal case, the THD of the output current is 0.15 %.

Table 3.4 Average and RMS currents of an IGBT and an anti-parallel diode for different power factors.

		Power factor			
		0.10	*0.35*	*0.76*	*0.90*
$I_{C,AVG}$	[A]	16.0	19.1	24.0	25.8
$I_{C,RMS}$	[A]	34.2	37.3	42.2	43.8
$I_{F,AVG}$	[A]	13.6	10.7	5.83	4.20
$I_{F,RMS}$	[A]	31.1	27.2	19.1	15.5

Table 3.5 Average and RMS values of the total dc-link current for different power factors.

		Power factor			
		0.10	*0.35*	*0.76*	*0.90*
$I_{dc,tot,AVG}$	[A]	1.76	6.15	13.4	15.8
$I_{dc,tot,RMS\,100Hz}$	[A]	12.4	12.4	12.4	12.4
$I_{dc,tot,RMS\,>100Hz}$	[A]	10.4	10.1	9.06	8.47

Effect of lowering modulation index

Control of the output voltage can be done by changing the reference amplitude and thus the modulation index. However, if the modulation index is low, so is also the dynamics of the PWM pattern. This is illustrated in Figure 3.25 for a modulation index of 0.2. The comparison is made in *a)*, the voltages of the two phase legs are shown in *b)* and *c)* and the resulting voltage over the load is shown in *d)*. It is seen that the pulses generated by the two references do not vary much in width and the resulting output voltage pulses are short. Additionally, the harmonics are more significant when compared with the fundamental component. This is seen in Figure 3.26 where the amplitude spectrum of the output voltage is shown. The THD is 232.1 % as compared to only 52.49 % for unity modulation index which was shown in Figure 2.13. For output voltage control, it is therefore better to regulate the dc link voltage.

Effect of dead-time

In the practical case there is both a turn-on and turn-off delay for any kind of transistor. This forces the use of so called blanking or dead-time in the generation of the PWM control signals. This is a short duration between the switching off of one transistor and the switching on of the other transistor, or the other way around, in the same phase leg. This prevents short-circuit of the source. Figure 3.27 shows the effect on the load current when implementing dead-time, for 5 and 10 μs. It is seen in *a)* both how the phase shift is increased and the waveform gets distorted as the dead-time is increased. It is also seen how, as dead-time minimizes the total conduction time, the RMS current is decreased. Introduction of dead-time gives low-order odd harmonics as shown in *b)*. The THD values are 1.77 % and 3.77 % for 5 μs and 10 μs respectively.

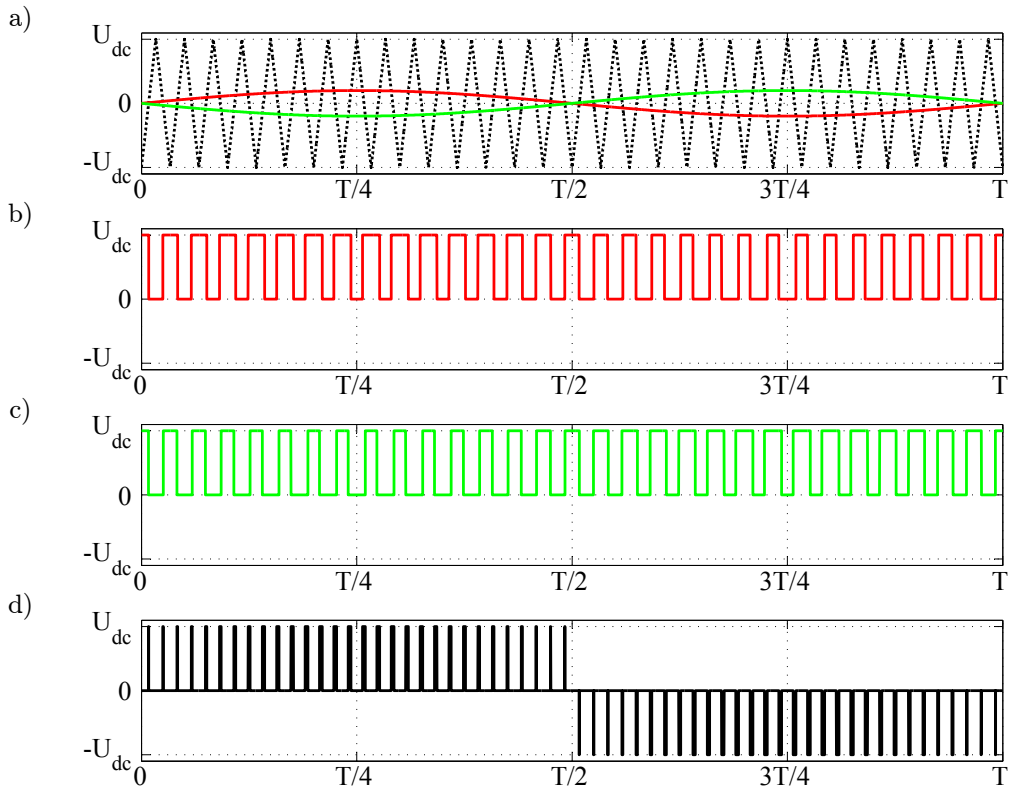


Figure 3.25 Unipolar PWM switching patterns for modulation index of 0.2 and frequency index of 30. *a)* Comparison of sinusoidal references to triangular carrier wave. *b)* PWM signals for the first phase leg. *c)*PWM signals for the second phase leg. *d)* Output voltage.

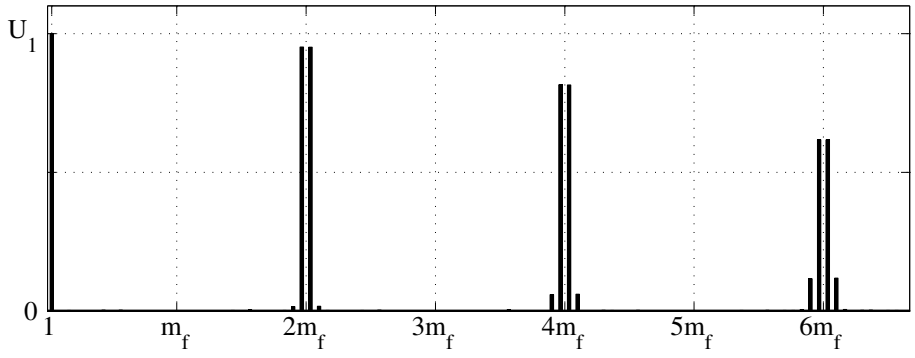


Figure 3.26 Unipolar PWM output voltage amplitude spectrum for modulation index of 0.2 and frequency index of 30.

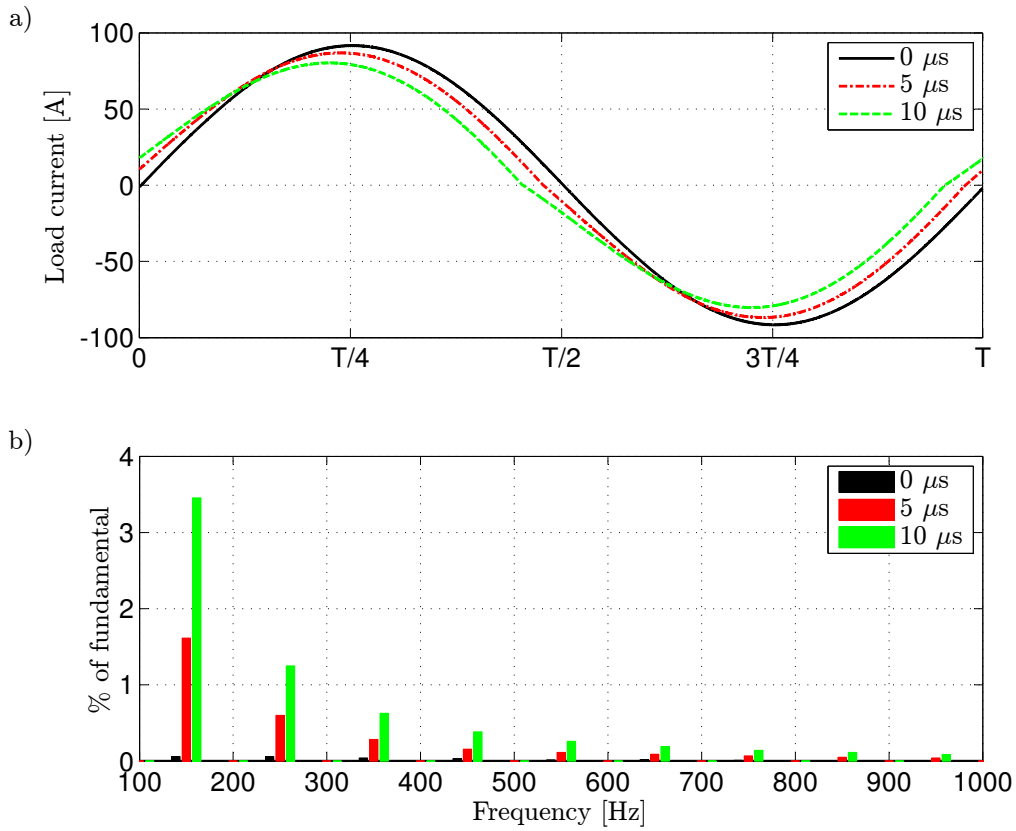


Figure 3.27 Effect of dead-time on load current. *a)* Waveforms. *b)* Amplitude spectra.

Effect of dc-link ripple

The full-bridge converter controlled for sinusoidal PWM draws a current with a second harmonic from the dc-link as was shown in Figure 3.24. This causes the same ripple in the dc voltage due to the impedance of the dc-link. If the input voltage is rectified from a three-phase source, a sixth harmonic will appear [23]. Figure 3.28 illustrates the individual effects of these harmonics. The amplitude of the oscillations were set to 1 % of the nominal voltage. Increasing the magnitude a factor 10 make the resulting current harmonics ten times as large. The fundamental component was not affected and remains at approximately 91.6 A. By changing the order of the harmonic it is seen that it is always the two adjacent harmonics that are affected in the output current. THD for the second harmonic case is 0.23 % and 0.20 % for the sixth harmonic case.

Effect of voltage drops

As described in Chapter 2.3.1, both the IGBT and the diode can be modeled as constant voltage drops with series resistances, when they conduct. The voltage drop is dependent on the current and will thus vary throughout the period. In addition, the resistance will vary with temperature. The effect of the conduction voltage drop is illustrated in Figure 3.29, where constant values have been used for both the constant voltage drop and the resistance. For the constant part, 1 V was used for both devices and 10 mΩ and 5 mΩ was used for the resistance of the IGBT and the diode respectively. From the figure it can be noticed that it primarly is the third, fifth and seventh harmonics that are increased in amplitude. It

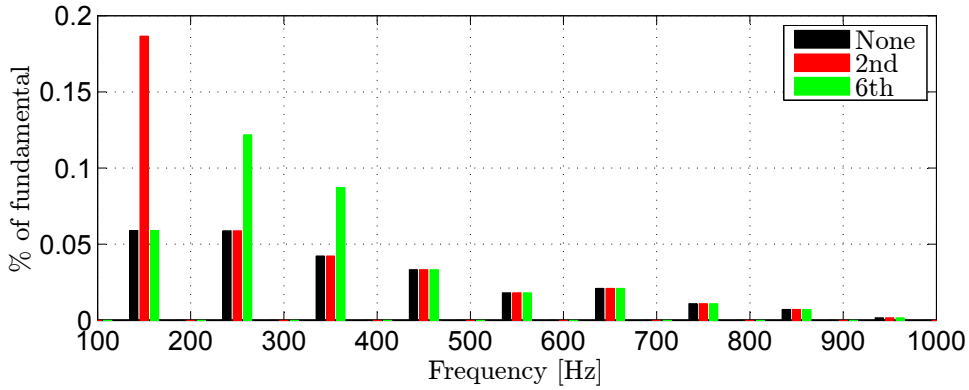


Figure 3.28 Effect of dc-link ripple of different harmonics on load current amplitude and phase spectrum.

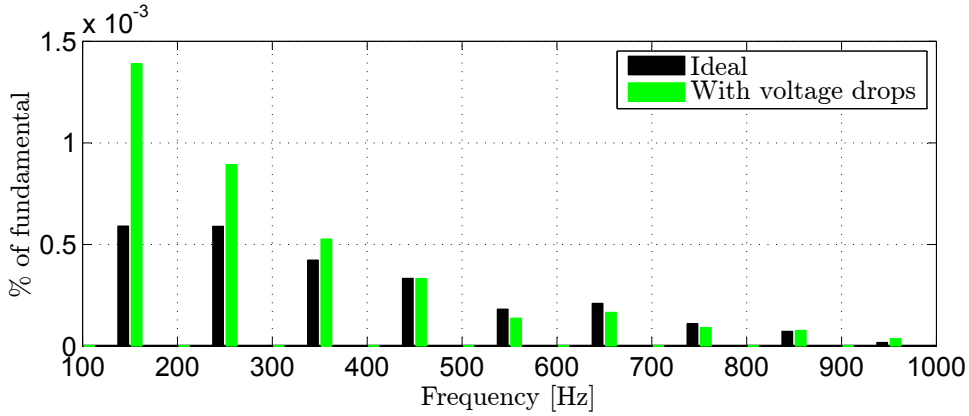


Figure 3.29 Effect of IGBT and diode voltage drops on load current amplitude and phase spectrum.

is also seen how some higher harmonics actually are decreased in amplitude. In total, it is only the odd harmonics that are affected. The THD is 0.21 % with the voltage drops.

Harmonics attenuation

The high-order harmonics are naturally attenuated by the characteristics of the cable loop which is primarily inductive. They are also easier to filter out than the low-order harmonics if a low-pass filter would be used. The low-order harmonics must however be attenuated by other means. The phase shifts of the harmonics of the output voltage can be calculated and used to improve the current waveform. This can be done by adding the inverse of the voltage harmonics to the reference [41]. By adjusting the amplitude, optimum results can be found.

Harmonics attenuation through modification of the reference was done for the third, fifth and seventh harmonic. Dead-time of 5 μs and voltage drops as presented above were added, together with a 10 % dc-link voltage ripple at 100 Hz. Lowering the dc-link voltage ripple to 1 % does not lower the resulting third harmonic as much as without the other nonidealities. The results are presented in Figure 3.30. *a)* Shows the amplitude spectra. *b)* shows the resulting modified reference waveform. The third harmonic was attenuated

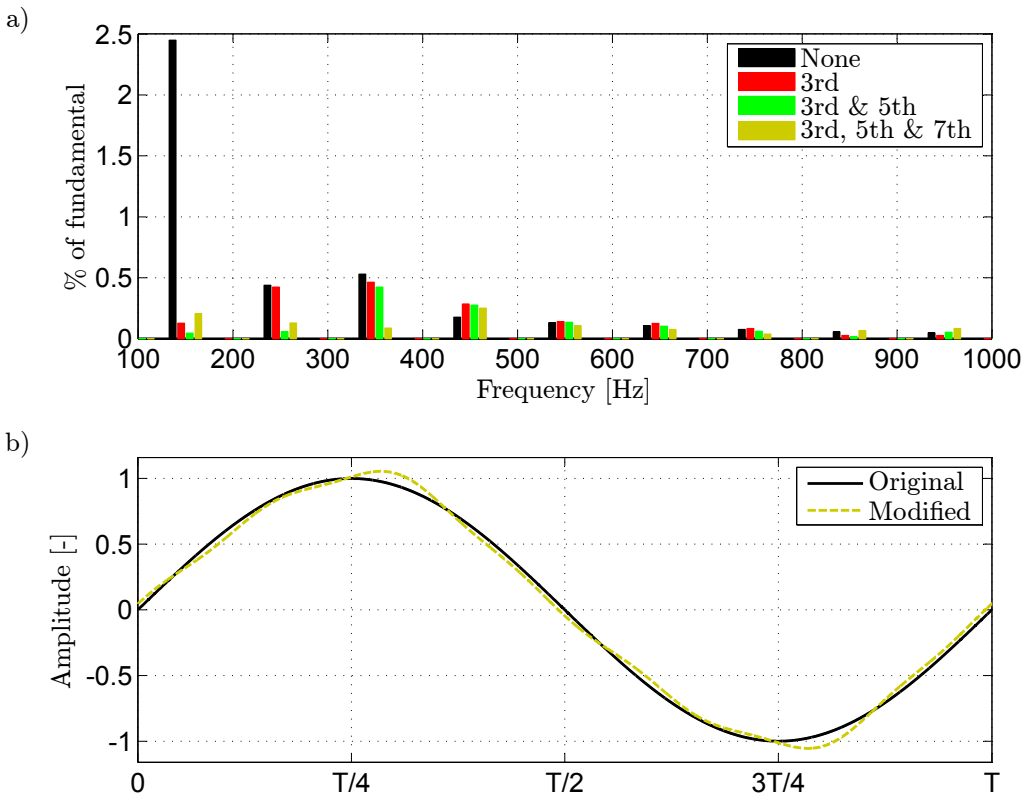


Figure 3.30 Attenuation of low-order harmonics through stepwise reference modification. *a)* Amplitude spectra for attenuation of third, fifth and seventh harmonic. *b)* Resulting reference waveform after attenuation of all three harmonics.

first, then the fifth and finally also the seventh. It can be seen that the attenuation of one harmonic in general increases the amplitude of higher-order harmonics. Additionally, the fundamental component amplitude is increased, such that it is closer to its nominal value. The third and the fifth harmonics are increased when the seventh harmonic is attenuated. The final values are however still much lower than before the modification. THD for the non-compensated case was 1.86 %. This decreased to 0.78 %, 0.59 % and finally to 0.52 % when all three harmonics had been compensated for.

Chapter 4

Design

4.1 Real time control system implementation

An overview of the complete control system is shown in Figure 4.1. The CompactRIO™ chassis NI cRIO-9076 was used for implementation of the algorithms as a real time system. This contains the RT controller, FPGA chip, module slots and communication ports. The ADC module NI 9239 is used for sampling the input signal and NI 9402 is the digital I/O module used to output the PWM signals to the IGBT gate drivers. In Tables 4.1, 4.2 and 4.3 summaries of the important specification of the chassis, digital I/O module and the ADC module respectively are shown [42] [43] [44].

The number of logic cells and RAM of the FPGA in the chassis restricts the size of the FPGA program. There are chassis with more module slots but only two modules are required for the control system. The values presented for propagation delay of the digital I/O module give a jitter of 37 ns. The ADC module has analogue prefiltering to prevent aliasing. There is no specified jitter for the ADC module. The noise data gives an SNR of 80 dB for a 1 V peak input and 100 dB for a 10 V peak input. These values are much higher than 53 dB which was the lower limit found in the simulations presented in Chapter 3.1.4.

The phase tracking algorithm is implemented, together with the PWM generation, on the FPGA chip. This is done to keep accurate synchronization between input and output of the whole system. The real-time controller is used for initiation and transportation of data between the FPGA and the user interface on the host computer. The RT controller must be started before the user interface to establish a network link. All LabVIEW™ code is found in Appendix C.

Table 4.1 Specifications of the NI cRIO-9076 CompactRIO™ chassis [42].

NI cRIO-9076 chassis		
Module slots		4
RT controller clock rate	[MHz]	400
FPGA logic cells		43661
FPGA RAM	[Kb]	2088

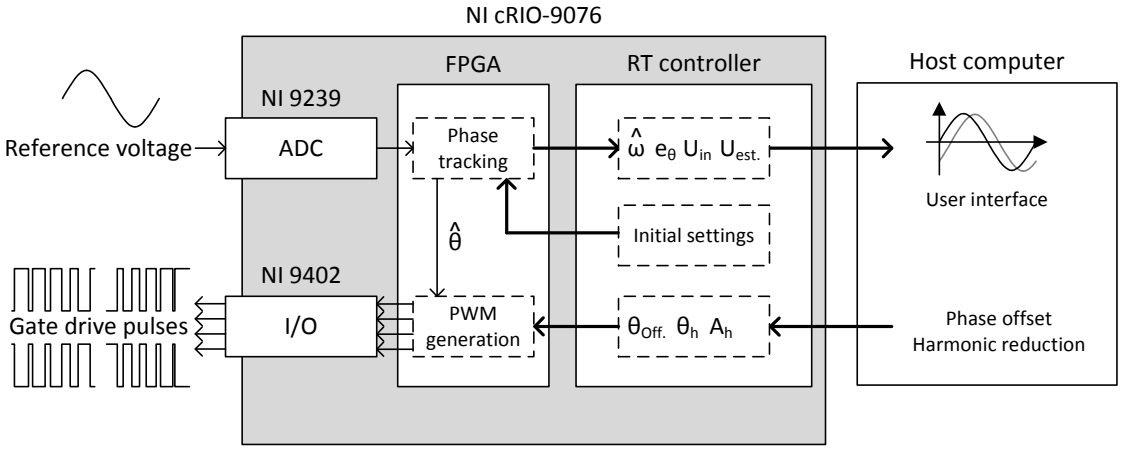


Figure 4.1 Overview of the complete control system. This contains the CompactRIO™ chassis, digital I/O module, ADC module and host computer

Table 4.2 Specifications of NI 9402 CompactRIO™ digital I/O module [44].

NI 9402 digital I/O module		
Output channels		4
Max clock frequency	[MHz]	20
Typical propagation delay	[ns]	18
Max propagation delay	[ns]	55
Output range	[V]	0–3.4

Table 4.3 Specifications of NI 9239 CompactRIO™ ADC module [43].

NI 9239 ADC module		
Input channels		4
Main clock	[MHz]	12.8
Accuracy	[ppm]	$\pm$100
Max sampling frequency	[kHz]	50
Resolution	[bit]	24
Input noise	[μV_{RMS}]	70
Input delay	[μs]	3 + 38.4/f_s

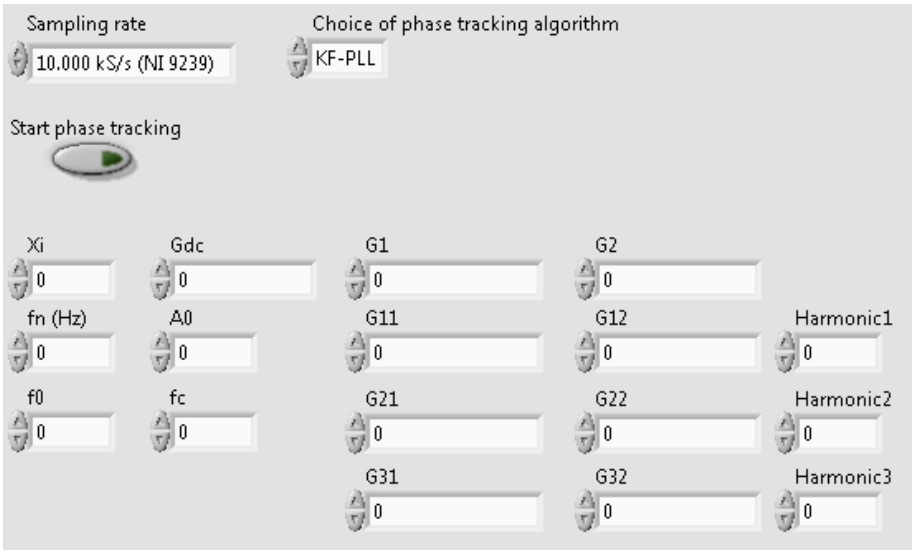


Figure 4.2 The initiation user interface of the RT controller.

4.1.1 User interface

The user interface is started by executing the VI seen in Figure 4.2 on the RT controller. The user chooses the sampling rate and phase tracking algorithm in the upper part and sets the parameters below. Xi is damping factor, fn bandwidth in Hz, $A0$ nominal amplitude of the E-PLL, fc cutoff frequency of IP-PLL low-pass filters, $f0$ nominal frequency, Gdc to $G32$ gain factors of the KF-PLL which will be explained in the subsection about the Kalman filter in Chapter 4.1.2 and Harmonic1 to Harmonic3 are settings for which harmonics that should be estimated by the KF-PLL. When *Start phase tracking* is clicked, a connection is set up to the FPGA chip, the settings are transferred and the code on the FPGA is executed. The LabVIEW™ code of the RT controller is found in Appendix C.1.

The RT controller buffers data from the FPGA and sends it out on a network link via an Ethernet connection. When both the RT controller and the FPGA is running, the main user interface, shown in Figure 4.3, of the host computer may be started. The actual tracking of the input signal can be seen in the top chart and the estimated frequency in the middle chart. The bottom chart shows the estimation error of the active PLL. The file path boxes to the left are used for setting the location for storing data to text files when the *Write to file?* button is clicked. Phase offset of the PWM reference relative to the input and the additional phases for harmonics attenuation are set via knobs in the -180° to 180° range. Amplitudes of the harmonics can be set relative to the fundamental component via slides. The stop button stops the main VI on the host computer, the RT controller and the FPGA. The LabVIEW™ code of the host computer is found in Appendix C.2.

4.1.2 FPGA implementation

The FPGA can handle fixed-point numbers up to 64 bits but also the single-precision floating-point (SGL) data type. Usage of the SGL data type gives superior accuracy to using the 24-bit fixed-point data type. Since the integer/decimal weight does not have to be specified, it is also easier to implement. It does however require more resources and will take longer time for execution of the same operation [45]. When using the fixed-point

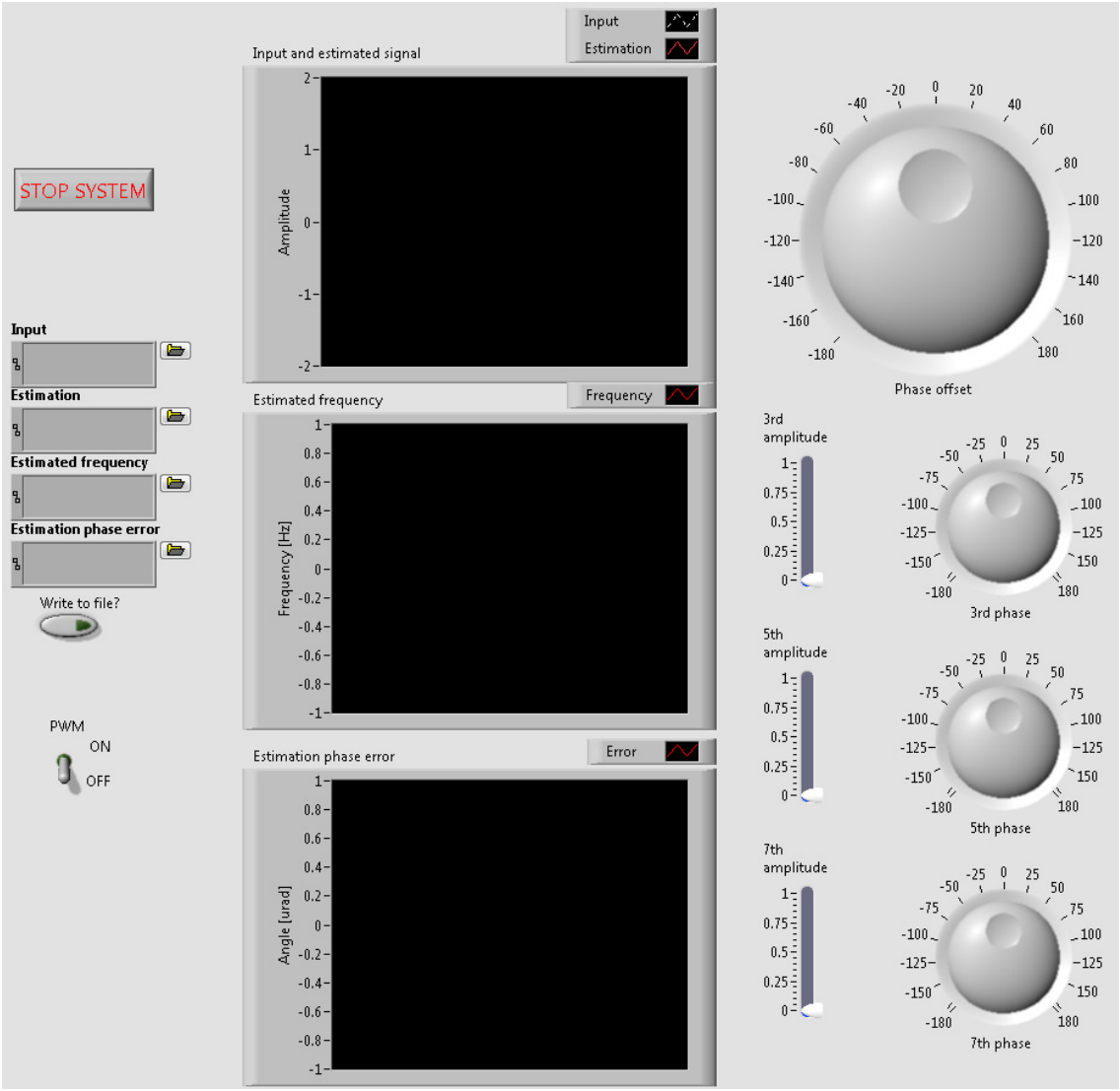


Figure 4.3 The main user interface of the host computer.

data type, arithmetic operations will change the output configuration to fit to the inputs. If a <+,32,20> number is multiplied with a <+,32,10> number, the output will be of type <+,32,30> to contain the product of the largest possible product. If many multiplications are performed, the final integer size may exceed the total word size. This will make the resolution go above 1. For the FPGA environment LabVIEW™ code, there are specific math operators that can be used with the fixed-point data type. These can be used to set the data type configuration of the output independent of the inputs. In this way a fixed representation of all the variables in the code can be used. When using this it is important to make sure that no overflow may occur anywhere.

The IP-PLL, the EP-PLL and the KF-PLL were implemented in LabVIEW™ code and simulated for comparison with the previous simulations with MATLAB®. Usage of the SGL data type gave oscillations at a few μrad in steady state, with a noise-free input. Therefore, the fixed-point data type had to be used. With a <±,64,15> fixed representation for all variables, the same accuracy as with MATLAB® was obtained. Due to limited resources of the FPGA chip, the <±,54,15> representation had to be used in the final code. With this smaller data type, the standard deviations of the phase errors were about 1.5 times highers than the results obtained with MATLAB®. The results could be improved by making the integer part smaller. On the other hand, this would lower the maximum time of operation as the integrators would saturate in a shorter time. The LabVIEW™ code for the FPGA is shown in Appendix C.3.

PWM controller

The triangular-shaped carrier wave is built in steps at 40 MHz. At each new step, the carrier wave is increased or decreased and a comparison is made with the PWM reference. The PWM reference is updated at the same speed as the sampling of the voltage reference signal. This is also the same as the carrier wave frequency. The LabVIEW™ code of the PWM controller implementation is shown in Appendix C.3.1.

The frequency of the PWM reference is likely not to be exactly 50 Hz or any other integer divisor of the sampling frequency. This will cause subharmonics in the PWM signals and subsequently in the output from the inverter. If the frequency of these harmonics are low they may cause high currents due to the inductive character of the load. This must therefore always be considered when running the system as the frequency of the input is likely to differ from time to time [23].

The inverse Park PLL

The IP-PLL was implemented with a digital low-pass filter for estimation of the fundamental component equal to that used in the MATLAB® simulations. The LabVIEW™ code of the IP-PLL implementation is shown in Appendix C.3.2.

The enhanced PLL

The E-PLL was implemented only for estimation of dc-offset and the fundamental component including phase and amplitude. Tests with an extended implementation for estimation of harmonics were done. No successful simulation or test with sampled signal was however

achieved and it was therefore not implemented in the final FPGA configuration. The LabVIEW™ code of the E-PLL implementation is shown in Appendix C.3.3.

The Kalman filter

The original KF-PLL was implemented for estimation of the fundamental component. The matrix operations related to the state variance matrix $\boldsymbol{P}$ required a lot of resources even if the off-diagonal elements were set equal and lowered the number of computations. In the static KF, without model update, the terms of the state covariance matrix $\boldsymbol{P}$ will converge to some small values, as described in Chapter 2.1.4. These values are dependent only on the value set for the process covariance matrix $\boldsymbol{Q}$ and the measurement covariance matrix $\boldsymbol{R}$. With model update through frequency feedback, noise may be added to the steady state value. Moreover, the final values will vary depending on the frequency. However, for an input without much variations there will be small variations in the steady state value for $\boldsymbol{P}$. At the end of a 20 second simulation, noise free input with estimation of only the fundamental component, the state covariance has converged to

$$\boldsymbol{P}_{converged,50\ Hz} = \begin{bmatrix} 0.00141512433465 & 0.00001591747924 \\ 0.00001591747924 & 0.00141484033866 \end{bmatrix} \tag{4.1}$$

where P_{11} and P_{22} are the in-phase and orthogonal fundamental component state variances respectively and $P_{12} = P_{21}$ is the cross-covariance between the states. Increasing the frequency to 51 Hz gives

$$\boldsymbol{P}_{converged,51\ Hz} = \begin{bmatrix} 0.00141512781391 & 0.00001560531771 \\ 0.00001560531771 & 0.00141484033866 \end{bmatrix} \tag{4.2}$$

where only small differences are noticed compared to (4.1). Figure 4.4 shows the results of using the converged values of $\boldsymbol{P}$ shown in (4.1) for a noisy input at 51 Hz, compared to the original KF-PLL. It is seen in *a)* that the transient errors are increased and the time to reach steady state is slightly delayed. The steady state results shown in *b)* are however the same and due to this, the simplification does not affect the accuracy of the estimation.

To fit more states in the FPGA implementation for improved performance, update and prediction of $\boldsymbol{P}$ was eliminated. With $\boldsymbol{P}$ constant, the gain factor $\boldsymbol{G}_k$ is constant as well. This will make the input stage of the KF-PLL to be a proportional controller. Depending on the number of included states, the steady state values of $\boldsymbol{G}_k$ varies. Values for inclusion of dc-offset and odd harmonics up to the ninth were obtained by running the static KF, without the frequency updating PLL, for 20 seconds in MATLAB with the fundamental frequency at 50 Hz. The results can be found Appendix B. With this simplification, dc-offset and estimation of three harmonics were added to the FPGA implementation. All LabVIEW™ code of KF-PLL is found in Appendix C.3.4.

4.2 Inverter

The choice of components design for the inverter were based on the supply of 250 V and 64.8 A RMS to the three parallel connected transformers as described in Chapter 3.2.1. This is equivalent to a dc-link voltage of 354 V and a peak load current of 91.6 A. Safety margins and possible expansions to the system were also considered, if the cost increase

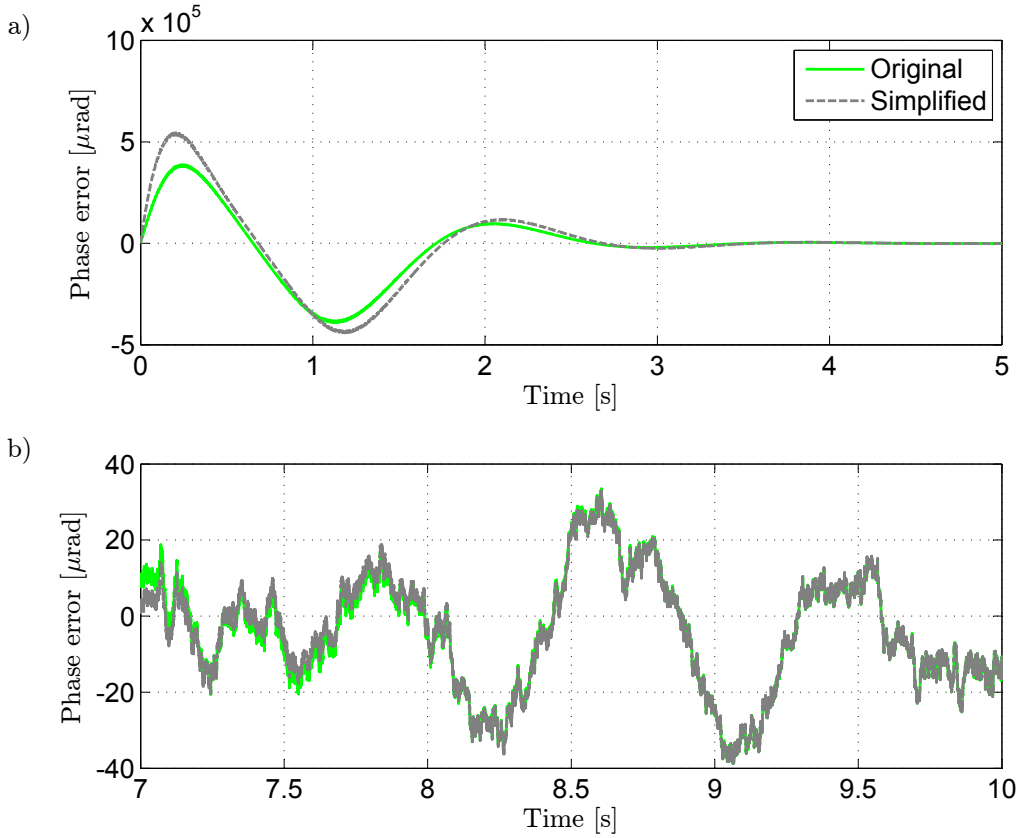


Figure 4.4 Effects of using constant $\boldsymbol{P}$ in KF-PLL for a noisy input. *a)* Transient state. *b)* Steady state.

would not be too high. Additionally, the components had to be available in reasonable time before the end of the thesis work. The average and RMS currents obtained in the simulations presented in Chapter 3.2.2 were used for loss calculations and choice of capacitors.

4.2.1 Selection of dc-link capacitors

In the worst case, three transformers are required that draws three times 21.6 A RMS to generate in total 2 kA RMS in the large cable loop. Possibly with additional cable sections but with no need for increased loop area and thus with a similar inductance. This gives the peak energy storage in the magnetic fields of

$$\begin{aligned} E_{load} &= \frac{LI^2}{2} \\ &= 2000^2 * 14 * 10^{-6} + 3 * 21.6^2 * 1.5 * 10^{-3} \approx 58.1\ J \end{aligned} \tag{4.3}$$

where the first term is the magnetic energy in the cable loop and the second term is for the short circuit impedance of the transformer. The energy in the cable loop alone is 56 J. When the current goes down, the energy will flow back to the dc-link. To prevent overvoltages, a sufficient capacitance must be used in the dc-link. If the rating of the capacitance is 400 V, its minimum capacitance must be

$$\begin{aligned} C &= 2\frac{E_{load}}{U_{dc,final}^2 - U_{dc,initial}^2} \\ &= 2\frac{58.1}{400^2 - 354^2} \approx 3.4\,mF \end{aligned} \tag{4.4}$$

where $U_{dc,initial}$ is the nominal dc-link voltage and $U_{dc,final}$ is the peak voltage due to the increased energy storage [35]. Additional voltage will appear due to the equivalent series resistance (ESR) of the capacitor. To decrease the ripple, the capacitance can be increased [23]. However, it was shown in Chapter 3.2.2 that especially the third harmonic can be significantly attenuated by modification of the reference. This harmonic is generated partially by the fundamental component power flow.

Electrolytic capacitors generally have higher energy density for the same price and volume as compared to film capacitors. The latter however have lower ESR and can thus withstand higher currents [23] [38]. Therefore, electrolytic capacitors were chosen to handle the energy storage requirement. The power flow generates a 100 Hz component in the dc-link current as was shown in Chapter 3.2.2. Film capacitors are chosen to handle the higher frequency components related to the switching.

By using multiple capacitors in parallel, the equivalent ESR is lowered and thus the power losses. Naturally, there are practical limitations, both due to space but also considering the total impedance of the dc-link. For the main energy storage, two B43456A9688M electrolytic capacitors from EPCOS™ were chosen. This has a capacitance of 6.8 mF, ESR of 18 mΩ and can handle 19 A at 80° [46]. Two of these can thus handle 38 A. According to Table 3.5 the maximum 100 Hz current is 37.2 A RMS. The total capacitance is 13.6 mF.

This makes the voltage rise to

$$U_{dc,final} = \sqrt{2\frac{E_{load}}{C} + U_{dc,initial}^2} + I_{peak} * ESR \tag{4.5}$$
$$= \sqrt{2\frac{58.1}{2 * 6.8 * 10^{-3}} + 354^2} + 91.6 * 9 * 10^{-3} \approx 366.7\,V$$

which is about 3.6 % above the nominal dc level. The no load voltage of the dc source must be set lower than the rating of the transformer to avoid saturation of the core. For the high-frequency ripple, three MKP1848622454P4 film capacitors from Vishay® were chosen. This has a capacitance of 22 μF, ESR of 6.5 mΩ and can withstand 11.3 A RMS at 10 kHz and 85° [47]. Three of these can thus handle a current of 33.9 A RMS. According to Table 3.5, the maximum RMS value of the high frequency current is 31.2 A.

4.2.2 Selection of IGBT module

The IGBTs can be bought in different packages for specific purposes, such as rectification or dc/dc conversion, also half- and full-bridges are available. A half-bridge module consists of one phase leg and two of these can be used in a full-bridge converter as the one shown in Figure 2.10. The advantage of this is that the connection between the upper and lower arm is kept very short. This minimizes the stray inductance, which gives voltage spikes during switching. A full-bridge module also minimizes the inductance of the dc-link between the two phase legs.

The maximum dc-link voltage is 354 V and the peak load current is 91.6 A for the design load as presented in Chapter 3.2.1. This is thus the minimum rating of the IGBT. It must however be considered that the rated values are absolute max values and presented during certain conditions such as specific chip temperatures and type of load. It is therefore necessary to have higher ratings than required. Additionally, the cooling system must be designed such that the maximum temperatures of the different parts of the IGBT are not exceeded for the design load.

Losses

As presented in Chapter 2.3, the total losses of the IGBT and the diode can be divided into switching and conduction losses,

$$P_{loss,\,IGBT} = P_{sw,\,IGBT} + P_{cond,\,IGBT} \tag{4.6}$$
$$P_{loss,\,diode} = P_{sw,\,diode} + P_{cond,\,diode} \tag{4.7}$$

The conduction losses are calculated based on (2.41) and (2.42) together with the simulation data of current values presented in Table 3.4. The switching losses are found by multiplying the average energy losses during one switching period with the switching frequency as

$$P_{sw,\,IGBT} = f_{sw} E_{sw,AVG,\,IGBT} \tag{4.8}$$
$$P_{sw,\,diode} = f_{sw} E_{sw,AVG,\,diode} \tag{4.9}$$

where f_{sw} is the switching frequency. The switching losses are scaled from the datasheet values. For scaling of the IGBT switching losses with (2.43), $K_{v,\,IGBT} = 1.4$ is used for

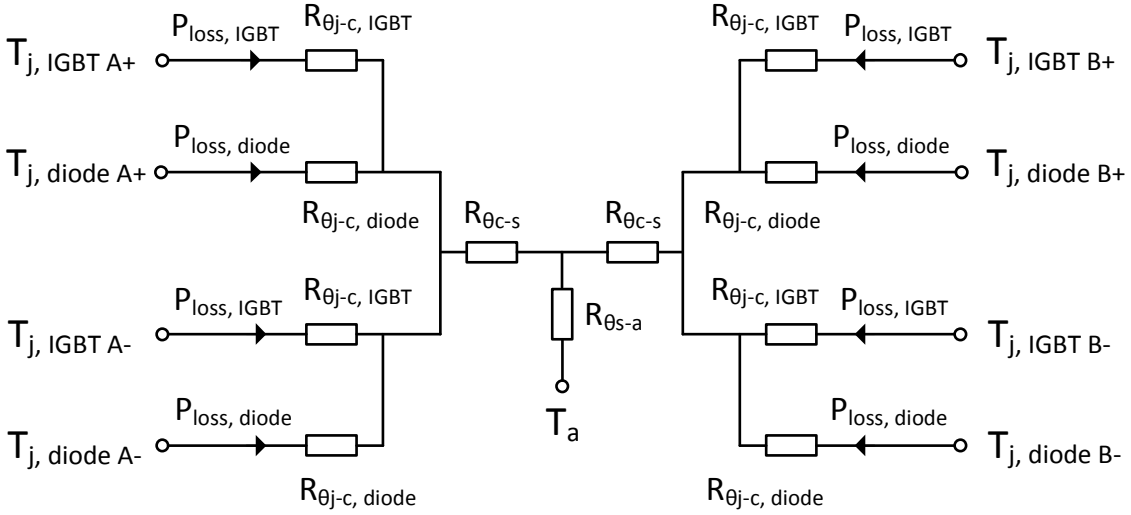


Figure 4.5 Thermal circuit for two equivalent half-bridge modules mounted on the same heat sink.

maximum losses. The diode switching losses are scaled according to (2.44). For both (2.43) and (2.44), a peak load current of 91.6 A and a dc-link voltage of 354 V is used according to the design load.

Thermal considerations

Many IGBT modules come in packages of one half-bridge that contains two serial connected IGBTs with anti-parallel diodes. Two of these are thus required to make up the full-bridge converter of Figure 2.10. To calculate the requirement on the heat sink for this setup, the thermal circuit shown in Figure 2.20 must be expanded. When mounting two equivalent modules on one common heat sink, the equivalent circuit is as shown in Figure 4.5. Solving the resulting junction temperature for any of the four IGBTs thus gives

$$T_{j,\,IGBT} = T_a + 4(P_{loss,\,IGBT} + P_{loss,\,diode})R_{\theta s-a} + 2(P_{loss,\,IGBT} + P_{loss,\,diode})R_{\theta c-s} + P_{loss,\,IGBT}R_{\theta j-c,\,IGBT} \tag{4.10}$$

and for one diode

$$T_{j,\,diode} = T_a + 4(P_{loss,\,IGBT} + P_{loss,\,diode})R_{\theta s-a} + 2(P_{loss,\,IGBT} + P_{loss,\,diode})R_{\theta c-s} + P_{loss,\,diode}R_{\theta j-c,\,diode} \tag{4.11}$$

For the same junction temperatures, different case temperature requirements are generally found for the diode and the IGBT as the losses and thermal resistances usually are different. The choice of the heat sink must be based on the lowest of these temperatures. This will make the design temperature of the other junction be below its maximum. By combining (4.10) and (4.11), the maximum *sink-to-ambient* thermal resistance is found as

$$R_{\theta s-a,max} = \frac{T_{c,min} - T_a - 2(P_{loss,\,IGBT} + P_{loss,\,diode})R_{\theta c-s}}{4(P_{loss,\,IGBT} + P_{loss,\,diode})} \tag{4.12}$$

where

$$T_{c,min} = min\Big\{T_{j,max} - P_{loss,\,IGBT} R_{\theta j-c,\,IGBT}\ ;\ T_{j,max} - P_{loss,\,diode} R_{\theta j-c,\,diode}\Big\} \tag{4.13}$$

where $T_{j,max}$ is the desired maximum junction temperature of either the IGBT or the diode. It is preferable to have a high $R_{\theta s-a,max}$ to limit the requirement of the heat sink. If a lower $R_{\theta s-a}$ than required is used, T_j is kept below $T_{j,max}$ which prolongs the lifetime of the device.

Comparison of IGBT modules

Modules from several manufacturers were investigated. However, due to their modular design and limited requirement for mechanical adaptations, products from Semikron were chosen to be used. Semikron is a German manufacturer of semiconductor devices for high power applications [48]. Only half-bridge modules were considered since no appropriate full-bridge module was found.

A module rated at 600 V and 100 A should suffice according to the specifications of the inverter. It is however good to have some safety margin. Figure 4.6 shows a comparison of three modules rated at 600 V and 100 A and, with dashed edges, two rated at 600 V and 150 A [49] [50] [51] [52] [53]. The plots show $R_{\theta s-a,max}$ for $V_{cc} = 354$ V, $\hat{I}_L = 91.6$ A with $T_{j,max} = 150$ °C and $T_a = 25$ °C. *a)* is for power factor 0.9. *b)* is for power factor 0.1. It is seen that $R_{\theta s-a,max}$ is higher for the lower power factor which means that the losses are lower. A lower power factor means that a bigger part of the current goes through the diode. The losses of the IGBT are therefore found to be larger than for the diode. For both power factors it is seen how the SKM 145GB066D is best at 10 kHz but only second best to the SKM 100GB063D at higher frequencies. This shows that the SKM 145GB066D has higher switching losses than the SKM 100GB063D and the other way around for the conduction losses.

SKM100GB063D was found to have the highest $R_{\theta s-a,max}$ overall. However, this was not available for delivery in reasonable time before the end of the thesis work. Therefore, SKM 145GB066D was chosen as being the second best. This is rated at 600 V and 150 A. A summary of parameters for this module are shown in Table 4.4. It can be seen that $T_{j,max} = 175$ °C which is 25 °C above what was used for calculations of $R_{\theta s-a,max}$. A safety margin of 25 °C is however recommended by the manufacturer [25].

4.2.3 Selection of gate driver

Also the gate driver was bought from Semikron. The main difference between the investigated drivers were the peak output current capability. The lowest recommended gate resistance of the SKM 145GB066D IGBT module is 2.8 Ω. According to (2.49), this gives a peak output current of 8.2 A considering switching of the gate voltage between -8 and 15 V.

The SKHI 23/12R gate driver board can handle 8 A peak output current for a 1200 V half-bridge IGBT module. At only slightly higher price is the SKHI 23/17R which has the same parameters but can handle 1700 V half-bridge IGBT modules. These two gate drivers were the only that had a maximum output current of 8 A and could be bought

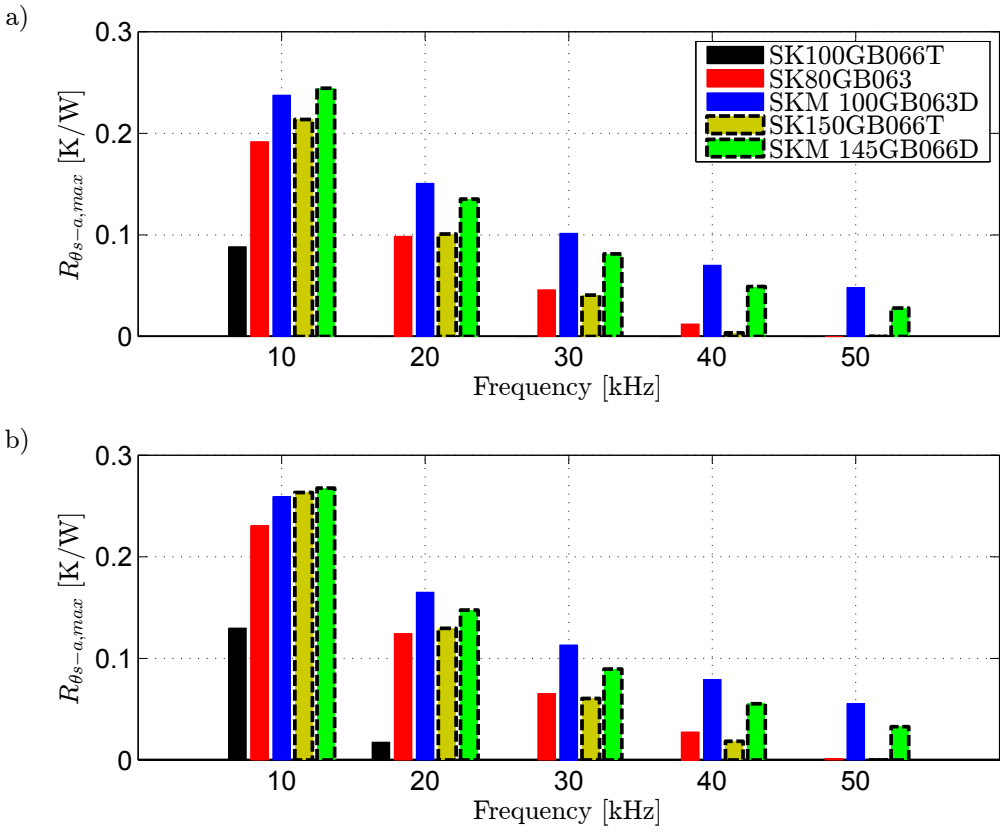


Figure 4.6 $R_{\theta s-a,max}$ for $V_{cc} = 354$ V, $\hat{I}_L = 91.6$ A with $T_{j,max} = 150$ °C and $T_a = 25$ °C for five different half-bridge IGBT modules from Semikron. All are rated at 600 V. Those with dashed edges are rated at 150 A and the others at 100 A. *a)* Power factor 0.9. *b)* Power factor 0.1.

Table 4.4 Specifications of the SKM 145GB066D IGBT half-bridge module.

SKM 145GB066D IGBT half-bridge module		
$V_{GE,max}$	[V]	20
Q_G[1]	[μC]	1.1
$V_{C0,max}$	[V]	1
$R_{on,IGBT,max}$	[mΩ]	8
$V_{F0,max}$	[V]	1
$R_{on,diode,max}$	[mΩ]	4
$E_{sw,IGBT}$[2]	[mJ]	14
E_{rr}[2]	[mJ]	3.5
$T_{c,max}$	[°C]	125
$T_{j,max}$	[°C]	175
$R_{\theta j-c,IGBT}$	[K/W]	0.3
$R_{\theta j-c,diode}$	[K/W]	0.5
$R_{\theta c-s,module}$	[K/W]	0.05

[1] *For V_{GE} in the range of -8 to 15 V.*
[2] *For $V_{cc} = 300$ V, $I_L = 150$ A and $T_j = 150$ °C.*

Table 4.5 Specifications of the SKHI 23/12R double gate drive circuit board.

SKHI 23/17R double gate driver		
Output channels		2
Galvanic isolation		Transformer
Max switching frequency	[kHz]	50
Typical propagation delay	[μs]	1.4
Output range	[V]	-8–15 V
Input high minimum[1]	[V]	9.5
Input low maximum[1]	[V]	4.8
$I_{G,AVG,max}$[2]	[mA]	$\pm$50
$I_{G,peak,max}$[2]	[A]	$\pm$8
$Q_{out/pulse,max}$	[μC]	4.2

[1] *For 15 V logic levels. 5 V levels also exist but are not recommended for noisy environments.*
[2] *Output current during either turn-on (positive) or turn-off (negative).*

in a quantity of two. Therefore, the SKHI 23/17R was chosen for possible use in other systems or upgrades of the inverter. Parameters of the chosen gate driver are presented in Table 4.5 [54].

The important aspects to be considered when selecting gate driver were presented in Chapter 2.3.4. The selected gate driver board SKHI 23/17R has a maximum output voltage rating that is higher than required for the IGBT module SKM 145GB066D. The gate charge that must be supplied is 1.1 μC as shown in Table 4.4 [53], for switching between the recommended gate voltages -8 and 15 V. These are the only voltage levels that the SKHI 23/17R uses on the outputs. The SKHI 23/17R can supply up to 4.2 μC. According to (2.48), with maximum output average current of 50 mA, the maximum switching frequency is found to be about 45 kHz. According to (2.47), the maximum power that the SKHI 23/17R can deliver is therefore found to be 1.15 W.

4.2.4 Selection of heat sink

The chosen heat sink was KL 285 P3 300 mm, also from Semikron. This was the heat sink with lowest thermal resistance that had premade tracks for fixing the IGBT modules and driver boards on. It has a $R_{\theta s-a}$ of 0.0847 K/W if used with the SKF 3-230-01 fan that also can be easily mounted on the heat sink [55]. According to Figure 4.6, this should make it possible to switch the IGBT module SKM 145GB066D at 30 kHz with a power factor of 0.1. However, this is considering and ambient temperature of 25 °C. If the ambient temperature was as high as 35 °C, 20 kHz could still be used, even at power factor 0.9. This would require a $R_{\theta s-a,max}$ of 0.12 K/W according to (4.12).

Figure 4.7 Copper plates of the dc-link.

4.2.5 CompactRIO™-to-gate driver interface

Since the NI 9402 digital I/O module can only output 3.4 V as highest but at least 9.5 V is required for the gate driver input, an amplification had to be made. The UCC27524 MOSFET driver from Texas Instruments was chosen to boost the voltage. It has a maximum propagation delay of 23 ns and jitter of 17 ns which was the minimum jitter found among the investigated drivers. One circuit can handle two signals and the output high level is equal to the supply voltage. Both channels of the UCC27524 MOSFET driver are non-inverting [56]. Two of these drivers were used, one for each gate driver board and they were supplied with 15 V.

The SKHI 23/17R board has an error signal that is active for short circuits and supply under voltage. To reset the error both the input control signals must be set to zero for at least 5 μs long [54]. Light-emitting diodes (LED) are used in in the interface box to indicate if the error signal is active. Reset can then be done by turning off and on the PWM signals in the main user interface as was shown in Figure 4.3.

4.2.6 Assembly and gate driver settings

To minimize the inductance of the dc-link, two 1 mm thick copper plates were used, one for each pole. These are seen in Figure 4.7. By placing them on top of each other with 1 mm plastic insulation in between, the area between them and thus the inductance becomes very small. This is important for reduction of overvoltages due to switching of the current. The minimum copper width is approximately 25 mm. This will make the maximum peak current density to be below 4 A/mm^2 which should not make the copper too hot [23].

The fully assembled inverter is seen in Figure 4.8. In the lower part of the figure, the interface box which contains two MOSFET drivers, one for each gate driver board, is seen with four BNC connectors. This amplifies the gate drive pulses from the digital I/O module. This also supplies the gate driver with its operational voltage of 15 V. The supply voltage is connected to the red and black terminals, either screwed on or by banana plugs. Two flat wires with the supply voltage, the PWM signals and the error feedback signals lead out to the two gate driver boards. The error-indicating LEDs are seen on the top right part of the interface box. The film snubber capacitors are soldered to small copper bars to a circuit board and mounted right between the white IGBT modules on the dc-link. Optimally, they should have been soldered directly on the copper plates. An extra insulation layer has

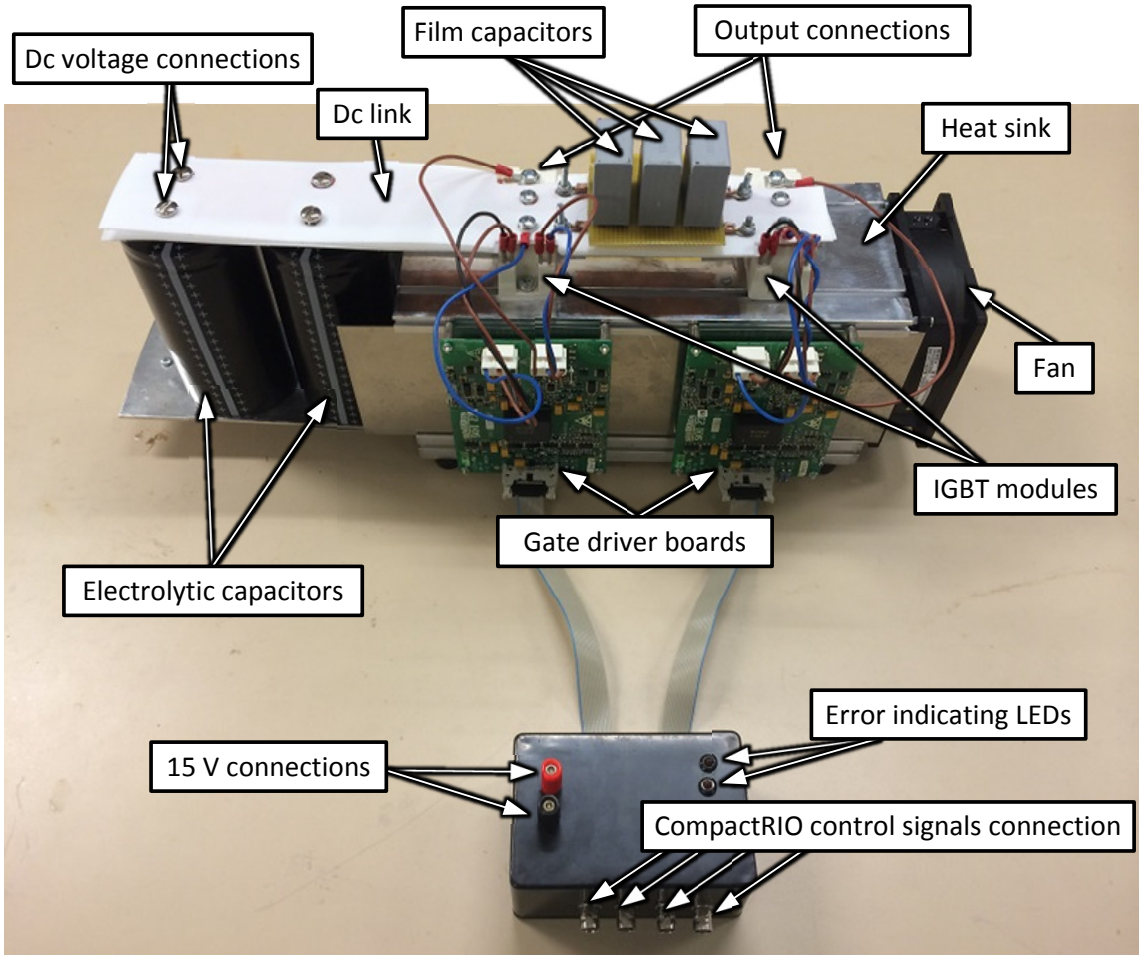


Figure 4.8 Fully assembled inverter with interface box.

been added on top of the dc-link for safety. The IGBTs are mounted on the heat sink to optimally share the space. Metal plates are used around the heat-sink to direct the air flow in it. The electrolytic capacitors are placed on a raised plate to be vertically aligned with the IGBTs. The dc supply voltage is connected on the far left. The load is connected to the two screw terminals of the IGBTs in the top of the figure. The fan is seen in the far right and is mounted to extract air out from the heat sink. This heats up the fan instead of the capacitors at the other end but it is designed to withstand this [57].

Gate driver settings

The internal gate-resistance of the SKM 145GB066D IGBT module is 2 Ω [53]. Resistance was added such that the total gate-resistance was about 3.3 Ω. This gives a current peak of about 7 A from the gate driver, according to (2.49). The SKHI 23/17R gate driver board can handle 8 A [54]. $R_G = 3.3\ \Omega$ gives fast switching but still some safety margin for the gate driver board and considering tolerance of the resistors.

The dead-time was set to approximately 2.5 μs. This is more than twice the maximum total turn-off time of the SKM 145GB066D IGBT module and should give a good safety margin. The maximum total turn-off time is about 1.2 μs, for a load current of 150 A and a gate-resistance of 26 Ω. The turn-off time decreases with both decreasing load current and gate-resistance. The dead-time is set by soldering resistors to the gate driver board. The

tolerance of these resistors adds to the inaccuracy of the set dead-time.

Chapter 5

Tests and measurements

5.1 Phase tracking of the implemented PLLs on CompactRIO™

The Wavetek 10 MHz DDS Function Generator model 29 with frequency setting in steps of 0.1 mHz was used as input to the CompactRIO™ system. The frequency stability of this function generator is 1 ppm with an accuracy of 10 ppm which corresponds to 50 mHz and 500 mHz respectively for a 50 Hz signal [58]. The frequency was set to 50.0000 Hz with no dc offset. The amplitude was set to 7.5 V. The E-PLL, the IP-PLL and both the original and simplified KF-PLL were tested. The same tuned settings as was presented in Table 3.2 were used. All the presented values are the estimated values from the algorithms based on tracking of the sampled signal from the function generator. The nominal frequencies of all the PLLs were set to 50 Hz. Since the frequency offset to the input is smaller, the time to reach steady state is shorter than the 10 seconds design limit from the tuning process. The restricted amount of FIFOs for storing data from the FPGA limited the analyzed data to the input sampled signal, the estimated signal, the estimated frequency and the estimation error.

The errors shown in the figures are the errors between the estimated signal and the sampled signal. The figures show the transient of the estimated frequency in *a)* with steady state in *b)* and transient state of the error in *c)* with steady state in *d)*. The results of the E-PLL show the voltage error instead of the phase error which is shown for the IP-PLL and KF-PLL. The errors and the estimated frequency are read from the PLLs as was shown in Figure 3.5, Figure 3.7 and Figure 3.10 for the IP-PLL, E-PLL and KF-PLL respectively.

5.1.1 The inverse Park PLL

The results of the IP-PLL are shown in Figure 5.1. As can be seen, the transient state is short but the error in steady state is large. The standard deviation of the phase error is 1120 μrad and 549 μHz for the estimated frequency. The phase error has an oscillatory shape at the fundamental frequency. This can be the results of a dc offset in the input signal to the Park transformation block [8]. The output function generator was set to contain no dc offset but a dc offset might appear from quantization error in the ADC module or inaccuracy of the function generator.

By adding a dc offset with the function generator, the ADC quantization and function generator inaccuracy was compensated for. Additionally, with the cut-off frequency of the low-pass filters lowered to 1 Hz, standard deviations of 49.2 μrad for the phase error and

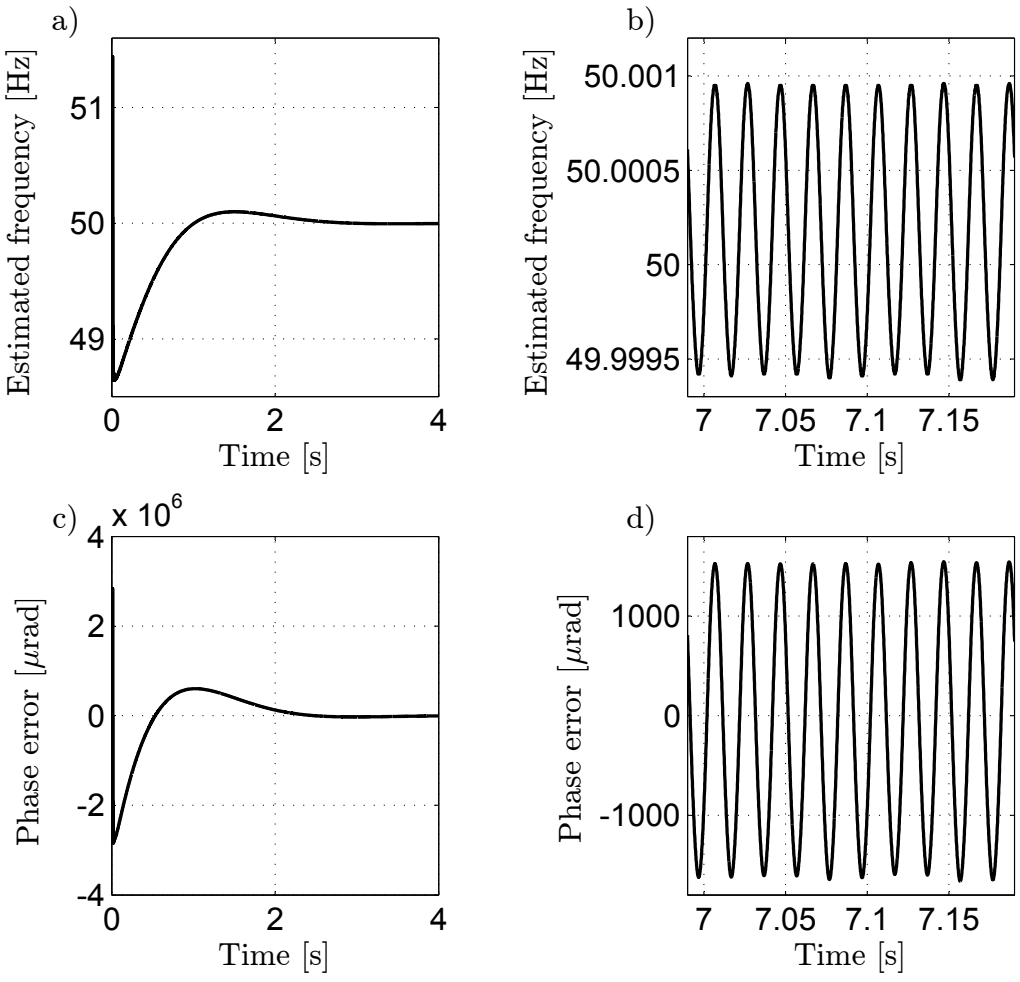


Figure 5.1 Results for CompactRIO™ implementation of the IP-PLL. *a)* Transients of frequency estimation. *b)* Steady state of frequency estimation. *c)* Transients of phase error. *d)* Steady state of phase error.

24.5 μHz for the estimated frequency was obtained. A frequency error of 24.5 μHz is lower than the specified stability of the function generator. It is not specified how the stability of the function generator is defined. However, the instability may also be smoothed out by the low bandwidth of the IP-PLL.

The added dc offset was 31 mV which is only 0.41 % of the input amplitude at 7.5 V. This is much lower than in the simulations where a dc offset of 1.8 % gave a standard deviation of 50 μrad, with the cut-off frequency of the low-pass filters at 20 Hz. The compensation of the dc offset was done with the function generator. This could however be added to the CompactRIO™ implementation and be controlled from the main user interface. This would make a manual estimator of the dc offset.

5.1.2 The enhanced PLL

Unlike the IP-PLL and the KF-PLL, the E-PLL does not estimate the true phase error directly. The error must however be proportional to the phase error since the phase error is what should be minimized. Figure 5.2 shows the results for the E-PLL with dc estimation included. The transient of the voltage error seen in *c)* is saturated only due to limited size of the FIFO used to store data from the FPGA. The standard deviation of the frequency was 8.81 mHz which is more than 3.5 times worse than what was obtained with the IP-PLL. The nominal amplitude of the E-PLL was set to 1 V. With the nominal amplitude set to 7.5 V, as the input amplitude, similar results were obtained. The performance of the E-PLL was not improved by any further adjustments.

5.1.3 The Kalman filter PLL

The phase error of the KF-PLL is from the SRF-PLL and is thus the phase error in the estimation of the estimation of the sampled signal. The phase error of the SRF-PLL is however similar to the real error as was shown in Figure 3.11.

Figure 5.3 shows the estimated frequencies and phase errors of the original KF-PLL and the simplified KF-PLL without dc estimation. The relative behavior is the same as in the simulations with the simplified algorithm giving larger errors in the transient state but same errors in steady state. The standard deviation of the steady state estimated frequency is 70.1 μHz and 139 μrad for the phase error. Different from the simulations, the KF-PLL performs better than both the IP-PLL and the E-PLL with a dc offset present in the input.

With dc estimation added to the simplified algorithm, the results are 5.1 μrad standard deviation in the phase error and 4.66 μHz for the estimated frequency. This happens in the same time span as illustrated by Figure 5.3 and is thus much faster than the response of the IP-PLL with similar error.

Tests were also done with the grid voltage as input. The error peaked at $\pm$1000 μrad with estimation of dc offset and third, fifth and seventh harmonic included. The error had a mainly sinusoidal shape. This gives a standard deviation of about 700 μrad which is close to a fourth of what was obtained with the previous calibration system. Comparisons between the amplitude and phase spectra of the input and estimated signal are seen in Figure 5.4. The estimation seems to be very good of the states that are included. The size of the FPGA chip was presented in Table 4.1. With a FPGA chip containing more cells and more RAM, more states could be added and the overall estimation could be significantly improved. By

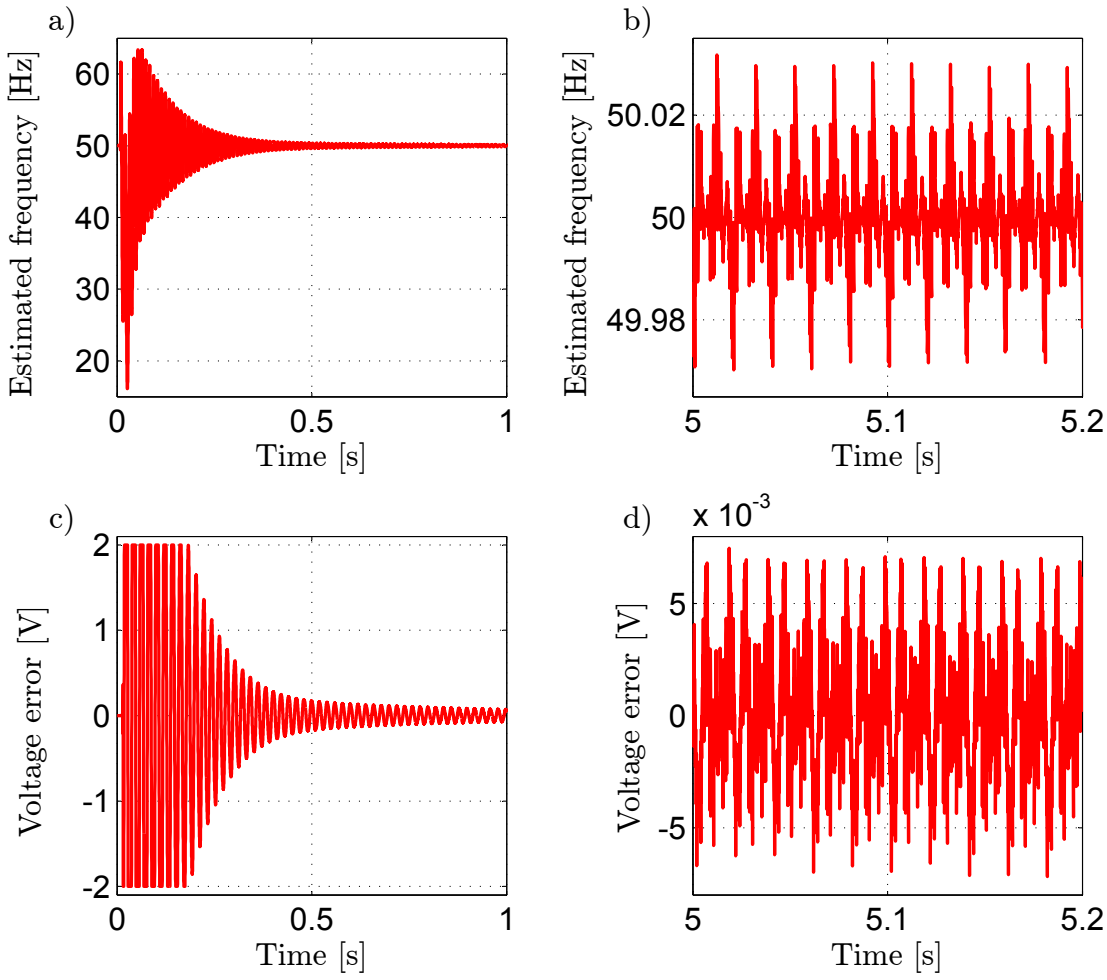


Figure 5.2 Results for the CompactRIO™ implementation of the E-PLL with dc-estimation. *a)* Transients of frequency estimation. *b)* Steady state of frequency estimation. *c)* Transients of voltage error. *d)* Steady state of voltage error.

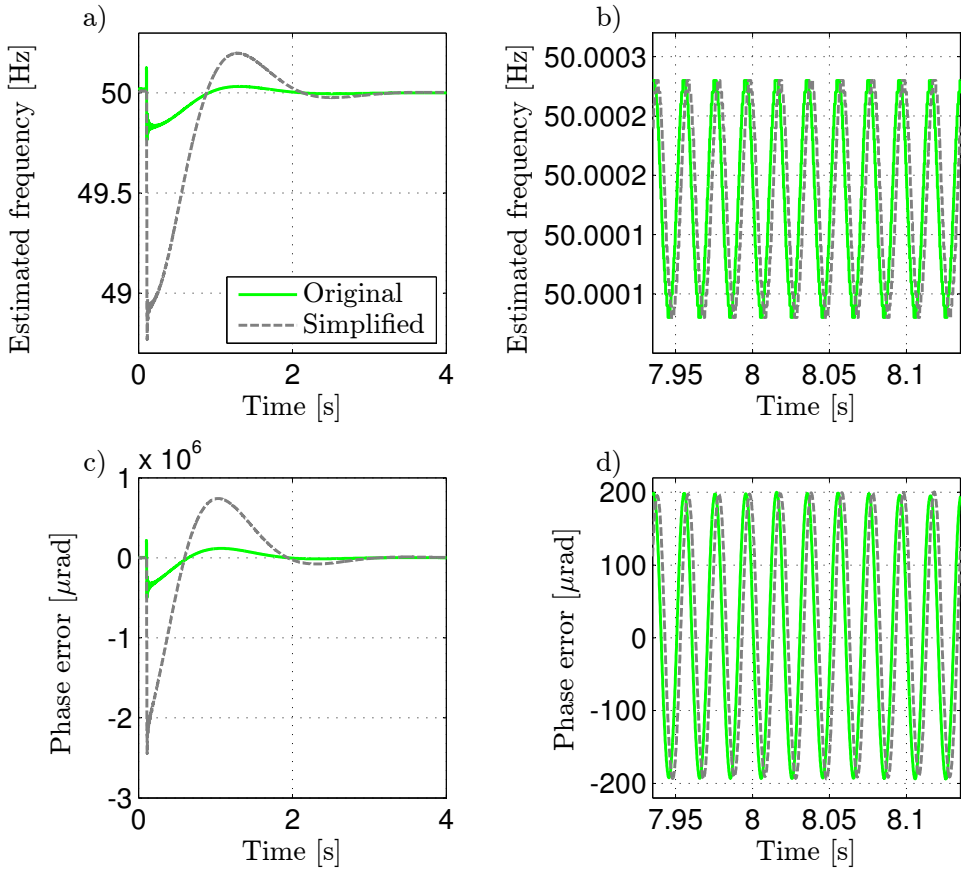


Figure 5.3 Comparison between the CompactRIO™ implementations of the original KF-PLL and the simplified KF-PLL without dc estimation. *a)* Transients of frequency estimation. *b)* Steady state of frequency estimation. *c)* Transients of phase error. *d)* Steady state of phase error.

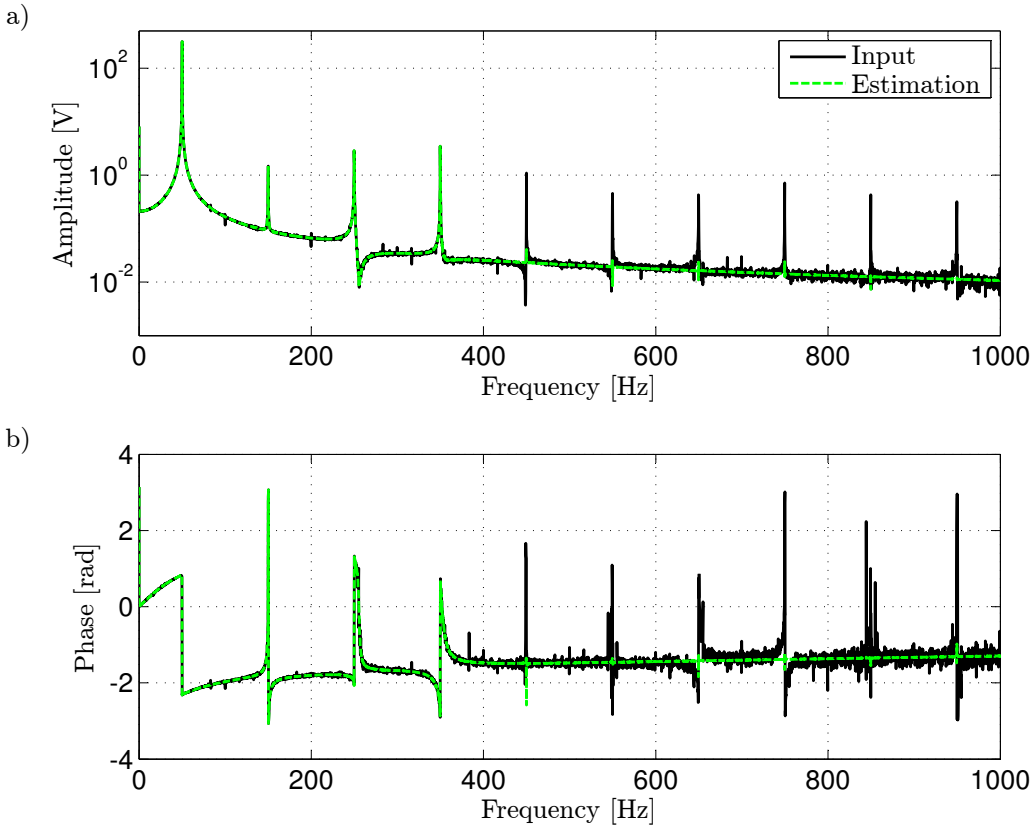


Figure 5.4 Comparison of amplitude and phase spectra between input and estimation of grid voltage with the simplified KF-PLL including estimations of the dc offset, fundamental component and the third, fifth and seventh harmonics

inclusion of a low-pass filter, additional to the harmonics estimation, the overall estimation could be improved. In the simulations it was shown that adding a low-pass filter alone does not to give good enough results for a third harmonic in the input. It should however work better dampening higher-order harmonics.

5.2 Inverter

5.2.1 Gate driver

Propagation jitter

Propagation of the PWM signals was analyzed between the CompactRIO™ output to the gate driver inputs and the gate driver outputs. Figure 5.5 shows the delay to the gate of one of the IGBTs, during turn-on and turn-off. The output from CompactRIO™ switches between 0 and 3.4 V and the voltage at the gate switches between -8 and 15 V. From the figure it can be noticed that the turn-on delay is about 3.6 μs and the turn-off delay is about 1.4 μs. The extra delay during turn-on is the dead-time of approximately 2.2 μs. No jitter in the turn-on or turn-off delays was observed to any of the IGBTs.

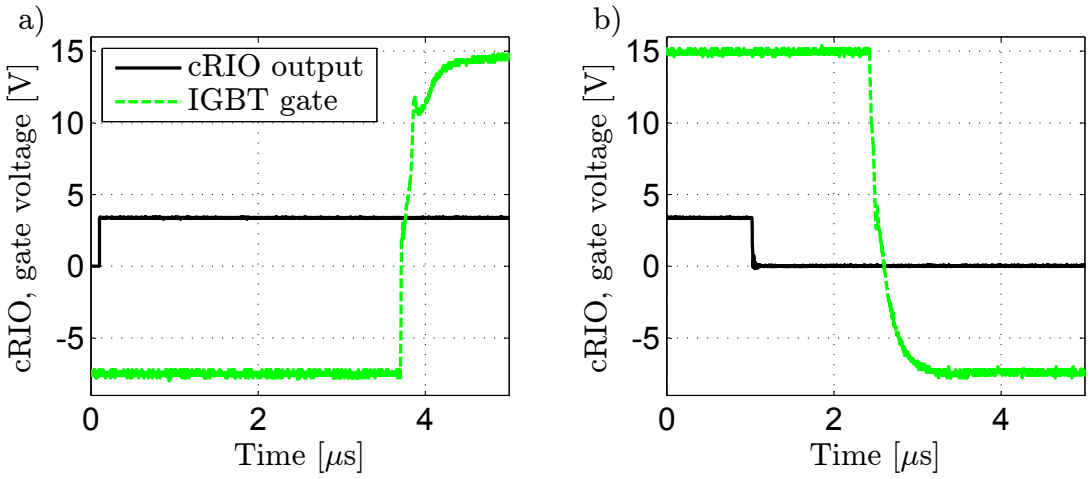


Figure 5.5 Propagation of turn-on and turn-off PWM signals from CompactRIO™ (cRIO) to the IGBT gate. *a)* Turn-on. *b)* Turn-off.

Gate signals

One period of the gate signals to the two IGBTs of one phase leg are shown in Figure 5.6, for a modulation index of 0.995, a switching frequency 10 kHz and a fundamental frequency of 50 Hz. *a)* shows the gate voltage of the top IGBT and *b)* shows the gate voltage of the bottom IGBT. It is seen how the gate driver eliminates the shortest pulses but also the shortest pulse gaps. The elimination of both the pulses and pulse gaps reduces the switching losses but increases the distortion of the output current. When the switching frequency is increased, there are more short pulses and pulse gaps as the switch period is decreased. Elimination of the pulses and pulse gaps with the same duration as for a lower switching frequency thus increases the distortion.

The pulse-width of the PWM signals can be calculated as

$$T_{on}(\theta) = T_{sw}(0.5 + \frac{m_a}{2}\sin\theta) \tag{5.1}$$

where T_{sw} is the maximum length of the switch period, the inverse of the switching frequency, and θ is the phase of the sinusoidal PWM reference. The pulse-width is equal to the switch period at the peak of the PWM reference signal and zero at its bottom. The length of the period of no pulse gaps for the top IGBT is about 0.8 ms. This equals to 14.4° for the reference frequency at 50 Hz. The pulse gaps are thus eliminated between 82.8 and 97.2° of the PWM reference. According to (5.1), since $m_a = 0.995$, the pulse gaps are eliminated when $T_{on} > 99.36$ μs. The maximum pulse-width should therefore not exceed 99.36 μs at any time to avoid elimination of pulse gaps. The modulation index must therefore be kept below

$$\begin{aligned} m_{a,max} &= 2\left(\frac{T_{on,max}(90°)}{T_{sw}} - 0.5\right) \\ &= 2\left(\frac{99.36}{100} - 0.5\right) = 0.9872 \end{aligned} \tag{5.2}$$

for a switching frequency of 10 kHz to avoid elimination of short pulse gaps.

The length of the period of no pulses to the top IGBT is about 2.3 ms. This equals to 41.4° of the 50 Hz reference. The pulses are thus eliminated between 249.3 and 290.7°

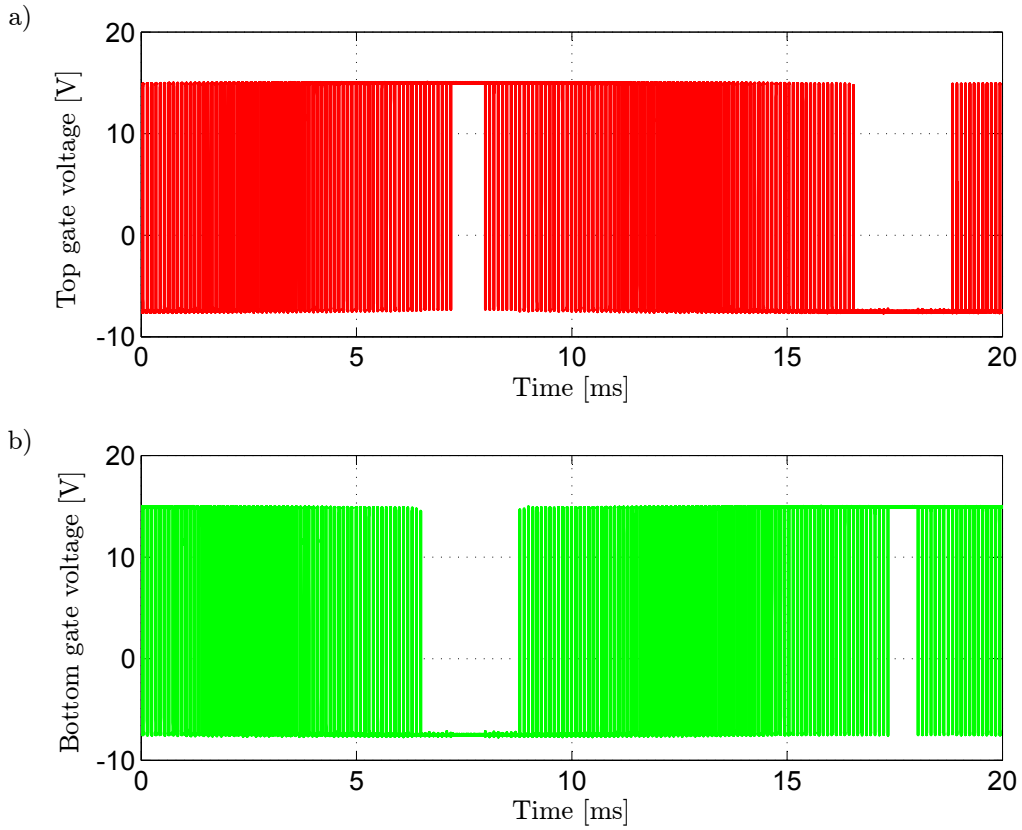


Figure 5.6 Gate signals to the two IGBTs of one phase leg for a switching frequency of 10 kHz and a fundamental frequency of 50 Hz at a modulation index of 0.995.

of the PWM reference. For $m_a = 0.995$, the pulses are eliminated when $T_{on} < 3.46$ μs according to (5.1). The modulation index must therefore be kept below

$$\begin{aligned} m_{a,max} &= -2\left(\frac{T_{on,min}(270^\circ)}{T_{sw}} - 0.5\right) \\ &= -2\left(\frac{3.460}{100} - 0.5\right) = 0.9308 \end{aligned} \tag{5.3}$$

for a switching frequency of 10 kHz to avoid elimination of short pulses. The maximum switching frequency that can be used is 45 kHz, limited by the gate driver board. To avoid pulses to be eliminated at this frequency, $m_{a,max} = 0.6886$.

In summary, the requirement on $m_{a,max}$ to avoid elimination of short pulses is seen to be limiting for avoiding distortion due to elimination of either pulses or pulse gaps. This is also clear from Figure 5.6 since the period without pulses is longer than the period without pulse gaps. Note, the calculations assume that the elimination is based purely on the duration of the pulse or pulse gap. No information about this is however found in the datasheet of the gate driver board. Additionally, lowering the modulation index increases the distortion due to increased relative amplitude of the switching harmonics as was shown in Figure 3.25 for $m_a = 0.2$. These are however easier filtered out than the low-order harmonics. For $m_a = 0.9308$, THD $= 60.72$ % and $m_a = 0.6886$ gives THD $= 92.29$ % rather than 52.43 % for $m_a = 1.000$.

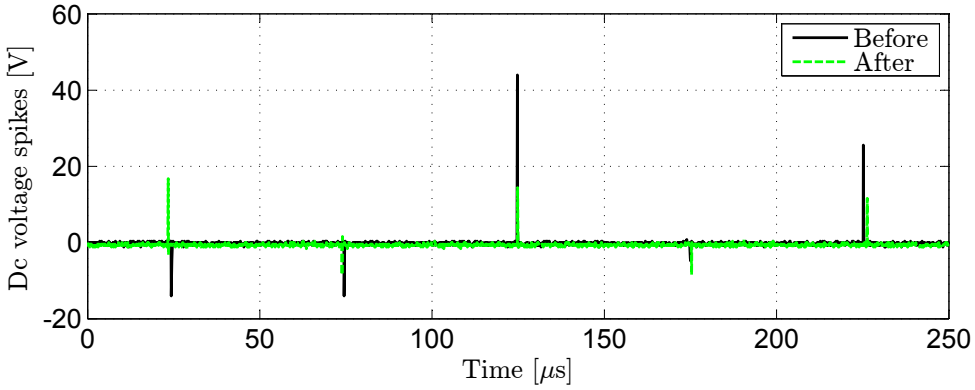


Figure 5.7 Voltage spikes at the dc-link, before and after adding 1 μF capacitors directly at the dc-terminals of the IGBT modules. Only the deviations from the mean value are shown.

5.2.2 Switching induced voltage spikes

When the current through the stray inductances of the dc-link or the IGBT modules is interrupted at the switching of an IGBT, voltage spikes occur. The spikes occuring over single IGBTs must be kept below the ratings of the IGBTs. The voltage spikes on the dc-link could reach up to 50 V as most with the original configuration of the dc-link as was shown in Figure 4.8. The operational voltage should be around 350 V and including ripple due to the power flow, can reach close up to 370 V. The rating of the film capacitors is 450 V and the rating of the electrolytic capacitors is 400 V. These ratings decrease with increasing temperature, which leads to that these spikes could be harmful [46] [47].

Attenuation of the voltage spikes was investigated. This was done by setting the dc source voltage to 150 V and measuring the voltage directly on the dc-terminals of one of the IGBT modules. The load was about 50 Ω and thus a load current of around 3 A. Figure 5.7 shows the dc-voltage deviations from the average value, before and after adding 1 μF film capacitors directly at the dc-terminals of the IGBT modules. The original film capacitors added to a total of 66μF, spread between the IGBT modules and Figure 5.8 shows the mounting of the extra 1 μF capacitor at one of the IGBT modules. The spikes reached up to 20 V as highest after the extra capacitor had been added. The results were the same for both IGBT modules.

Voltage spikes over one single IGBT were mainly slightly below 100 V but could reach up to 150 V before the addition of the extra capacitors, for the same settings as above. As the operational voltage should be around 350 V , there is still close to a 17 % margin to the rated voltage of 600 V. By adding the tight capacitance the majority of the spikes were kept below 50 V. The results were the same for all four IGBTs.

Tests were done with both higher dc-link voltages and load currents for calculation of losses which is presented in the following section. No larger spikes were however observed.

5.2.3 Losses

A 6 kW dc power supply supply capable of outputting maximum 20 A at 300 V was used for testing the inverter. The switching frequency was set to 10 kHz, the reference frequency

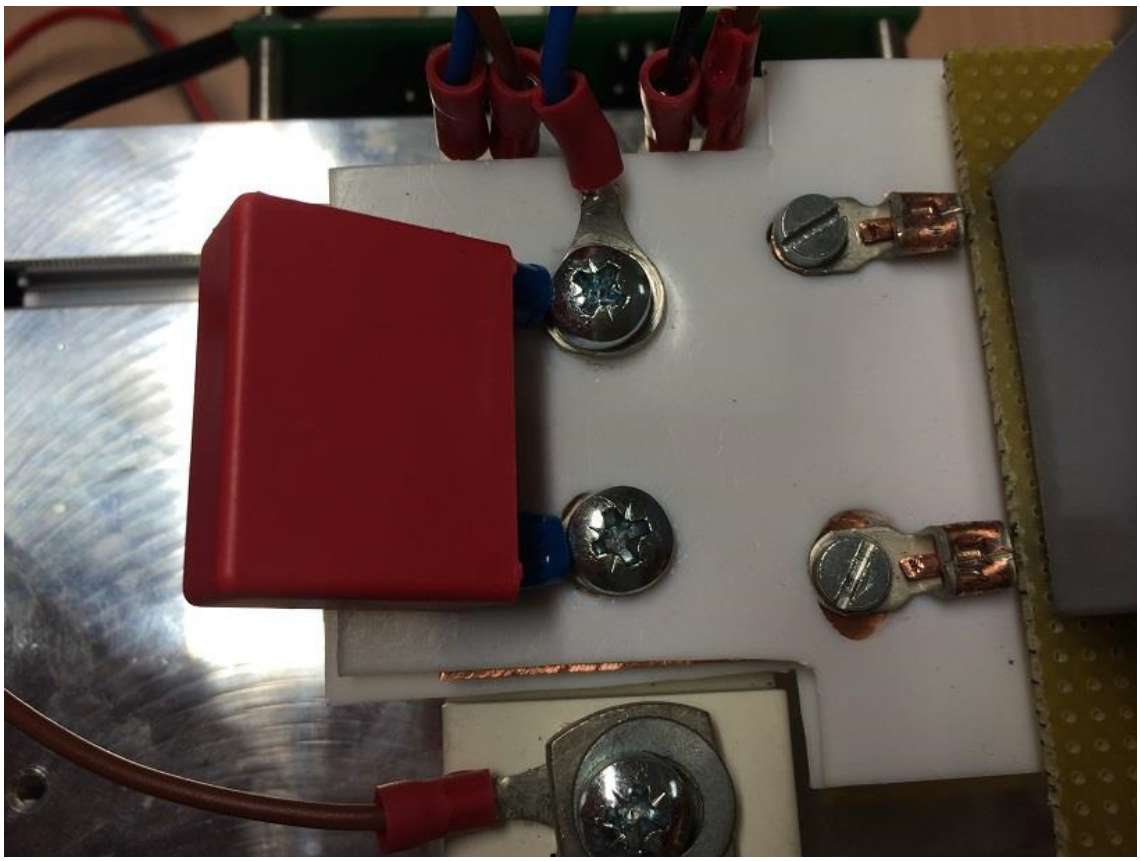


Figure 5.8 A 1 μF film capacitor mounted directly at the dc-terminals of one of the IGBT modules. The connection of the original film capacitor bank is seen to the right.

Table 5.1 Losses of the inverter for different loads.

Inverter losses							
Power factor	V_{dc}	I_{dc}	$I_{ac,RMS}$	P_{in}	P_{out}	P_{loss}	$P_{loss,est}$
	[V]	[A]	[A]	[W]	[W]	[W]	[W]
0.093	115	7.50	79.03	862.5	602.6	259.9	279
0.034	305	0.70	19.67	213.5	89.74	123.9	129
0.175	91.0	9.55	60.62	869.1	688.9	180.2	190
0.926	154	20.3	32.26	3126	3042	84.47	111

was 50 Hz and a modulation index of 0.995 was used. An additional 6.6 mF electrolytic capacitance was added to reach higher power levels in the measurements. This was due to that the dc-link voltage oscillated too much for the dc supply with only the two original 6.8 mF capacitors. Various loads were used but none to exactly match the design load. Based on the measured voltage and current, power losses were estimated in the same way as was done for choosing the IGBT module in Chapter 4.2.2.

The measurements for each load were started by increasing the dc source voltage until the maximum ripple which the dc power supply could handle occured. The input power was calculated from the dc voltage and current read on the display of the dc power supply. The dc-link voltage, IGBT voltage, inverter output current and output voltage were all sampled with an oscilloscope. Differential probes were used for the voltages and a current clamp for the current. The measurements of both the inverter output voltage and the voltage over one IGBT was performed directly on the terminals of the IGBT modules. The dc-link voltage was measured on the connection of the film capacitor bank.

A summary of the measurement results with actual and estimated losses are found in Table 5.1. As it can be seen, the estimated losses are a little higher than the actual. This can be due to that the gate driver eliminates the shortest pulse gaps, resulting in reduction of the switching losses. Additionally, the estimations were calculated based on the parameters from the datasheet of the IGBT module giving the highest losses. No significant temperature was seen on the IGBT module cases or the heat sink during any of the tests.

Figure 5.9 shows the load current in *a)*, dc-link voltage in *b)*, output voltage in *c)* and voltage over one IGBT in *d)*, for a load at power factor 0.175. It can be seen that the voltage spikes are below 20 V in the dc-link voltage and below 50 V in the voltage over the IGBT. The average dc-link voltage is 91 V and output peak current right below 86 A. This is close to the peak design current but only nearly a fourth of the design voltage. The oscillations in the dc-link voltage is seen to be at 100 Hz which is twice the fundamental frequency.

The magnitude of the oscillations in the dc-link voltage including the main voltage spikes is approximately 10 V which is a deviation of over 11 % from the average value of 91 V. The load inductance was 3 mH which gives a peak magnetic energy of approximately 11 J. Thus, with an initial voltage of 91 V, the peak voltage should be around 96 V according to (4.5). But as it can be seen in Figure 5.9, the oscillations in the dc-link voltage seem to be higher. It is hard to distinguish the ripple from the voltage spikes but additional to the capacitors, the impedance of the dc-link adds to the voltage ripple.

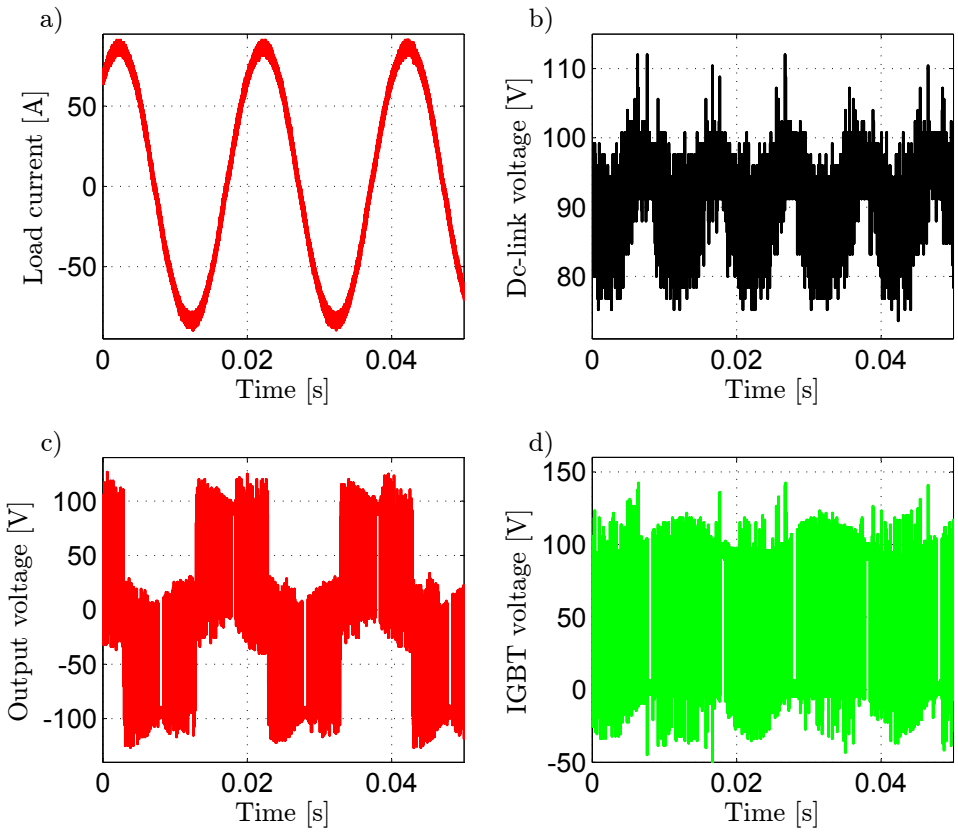


Figure 5.9 Results for a load of power factor 0.175 at 688.9 W. *a)* Load current. *b)* Dc-link voltage. *c)* Output voltage. *d)* Voltage over one IGBT.

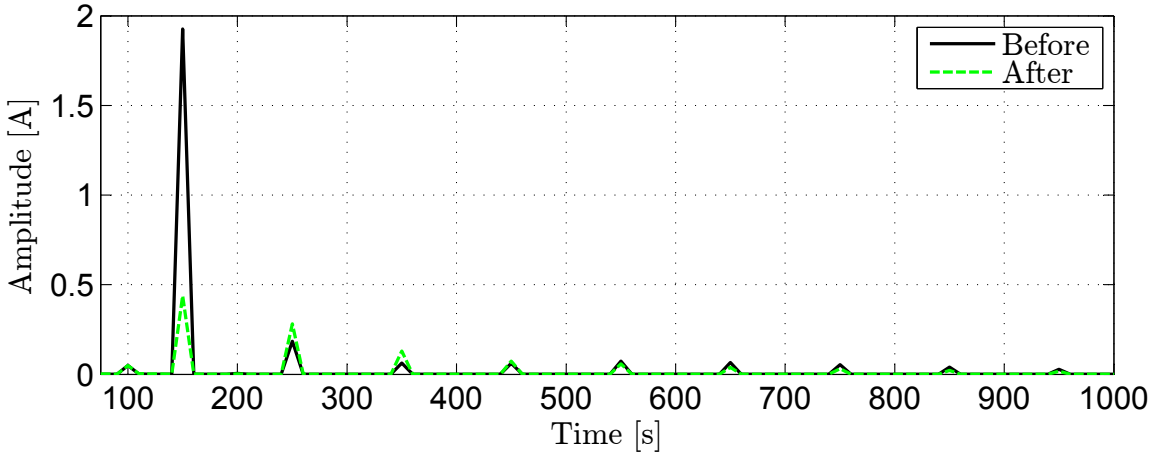


Figure 5.10 Result of modifying the reference to attenuate the third harmonic.

5.2.4 Harmonics attenuation

It was difficult to accurately find the phase difference between the fundamental component of the output voltage and the low-order harmonics of output voltage. When observing the phase spectrum, the phase shifts seemed to vary over time. This is thought to be related to problems with the FPGA and ADC clock which will be discussed in Chapter 5.3.

Only the third harmonic was successfully attenuated in the measured current. This was done by tuning the reference settings in small steps and observing the resulting amplitude spectrum with a spectrum analyzer. Figure 5.10 shows the result, with and without attenuation. As can be seen in the figure, the attenuation of the third harmonic is nearly a factor of four. The fifth and seventh harmonics were slightly increased, but some higher-order harmonics were however also attenuated. Different from the simulations, the amplitude of the fundamental component slightly decreased in amplitude. The THD value was however still decreased, from 3.92 % to 1.9 %. This is a reduction of 51 % while the simulations gave a reduction of 58 %. A non-zero second harmonic can be seen that is equal in magnitude before and after the third harmonic attenuation. This is thought to originate from the gate driver elimination of small pulse gaps which occurs twice in each phase leg per period. This could possibly also add to the difficulties of attenuating the fifth and seventh harmonics as it is the only additional error, as compared to the simulations in Chapter 3.2.2. However, fixed resistances and voltage drops of the IGBTs and diodes were used in those simulations, whilst they vary with current level and temperature in the measurement.

5.3 Phase and amplitude jitter

Analyses of phase and amplitude jitter in the system were done with the DSWM which was the reference instrument used in the previous calibration system setup, but without the DUT. Due to unknown problems with the measurement equipment, only 50 V was used as the dc-link voltage. A switching frequency of 10 kHz and a modulation index of 0.995 was used for the PWM.

The measurement setup is seen in Figure 5.11. A resistive heater at 50 Ω was used

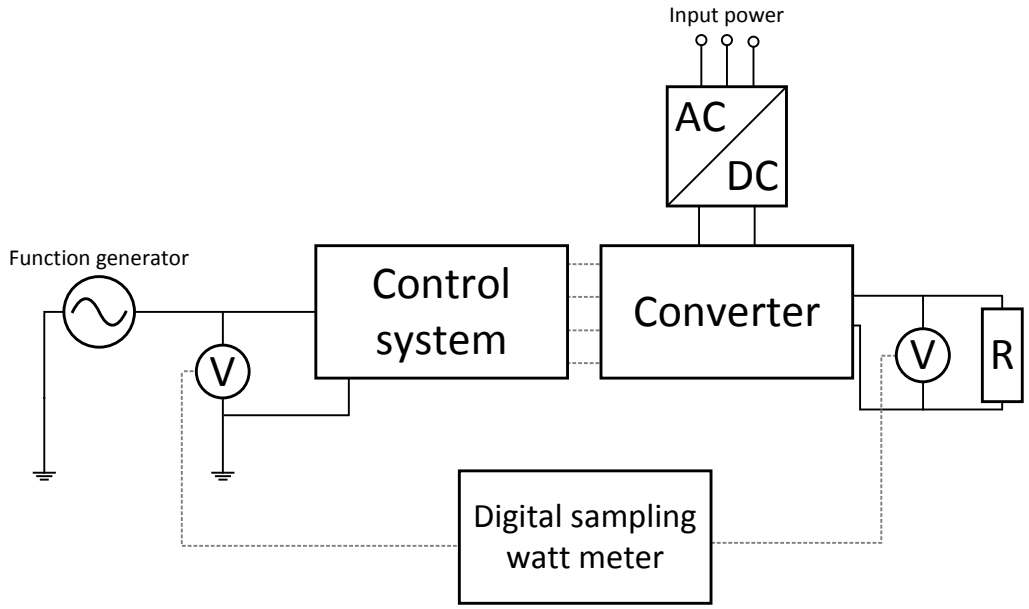


Figure 5.11 Measurement setup with the DSWM for measurements of phase and amplitude jitter in the complete system.

as load to the inverter which is seen as R in the figure. Measurements were done over one minute. About twice each second, the DSWM performed a fast fourier transform (FFT) over the period since the last computation, on the two input channels. The sampling frequency was about 10 kHz. The DSWM has a first-order low-pass filter with a cut-off frequency of 18 kHz. The switching harmonics at higher frequencies will therefore give aliasing effects.

The first measurements all gave the same kind of results, shown in Figure 5.12 with the error presented relative to the average phase shift between the reference voltage from the function generator and the fundamental voltage component in the inverter output voltage. The peak-to-peak value is 1.7665° which is equivalent to 30831 μrad. The reason for this proved to be an error in the structure of the FPGA code, common for all the algorithms. The estimation of the phase is written to a global variable in one while loop and read for generation of the PWM reference in another while loop as shown in Figure 5.13. The Loop Timer is based on the 40 MHz clock of the FPGA and sets the minimum time for execution of the loop. However, the PLL code cannot start before there is a new value generated by the ADC module. The sampling frequency of the ADC module is derived from its own clock of 12.8 MHz. Since the phase error drifts, the two loops must be running with different speeds. For a 50 Hz signal, an offset of one sample at 10 kHz corresponds to

$$\frac{50}{10^4} * 360 = 1.8^\circ \tag{5.4}$$

which is approximately the observed peak-to-peak jitter. The reference frequency is however not exactly 50 Hz and the true peaks might be missed due to the measurement windows. The period is about 16.8 s which gives a frequency of nearly 0.06 Hz and a drift of 5.95 μs/s.

By moving the reference calculation code to the same loop as the phase tracking for sequential execution, the linear phase error pattern was eliminated. Figure 5.14 shows

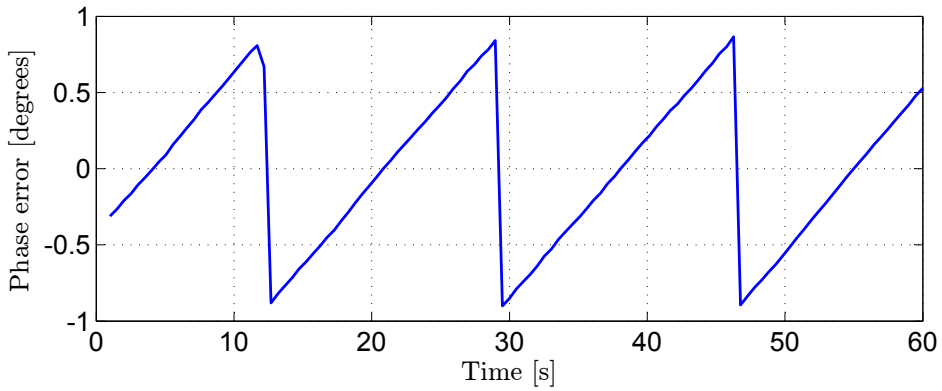


Figure 5.12 Results from the first tests of measuring the phase shift between reference and output from the inverter. The deviations from the average phase shift is shown.

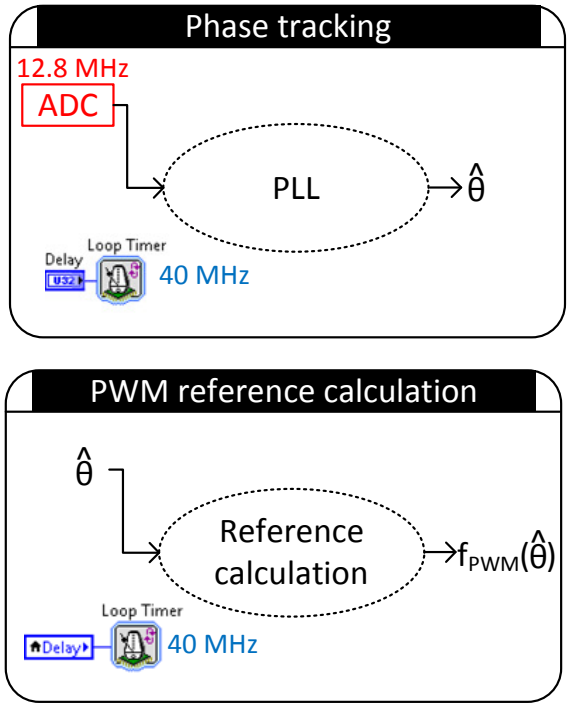


Figure 5.13 Structure of the original FPGA code for phase tracking and calculation of the PWM reference value in two separate while loops.

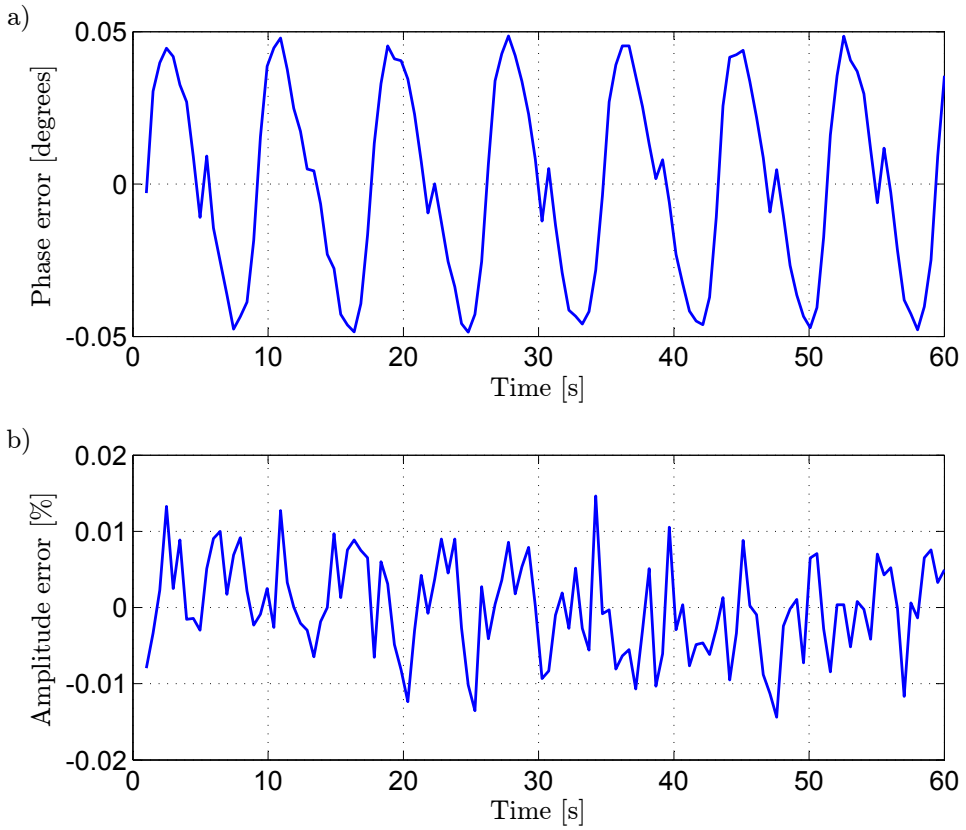


Figure 5.14 Typical results from the final tests of measuring jitter in the complete system. *a)* Jitter in phase error. *b)* Jitter in amplitude error.

the results of phase and amplitude jitter after this correction. As shown in *a)*, the phase error now has an oscillatory pattern with a frequency of approximately 0.12 Hz with a peak-to-peak value of around 0.1°. This frequency is about twice what is seen from Figure 5.12. The standard deviation is 0.032° or about 560 μrad.

The amplitude jitter shown in *b)* is more random than the phase jitter and does not show the same oscillatory pattern. This is therefore thought to mainly originate from the dc source. The standard deviation of the amplitude error is about 63 ppm. The stability of the voltage of the dc source is specified to be 50 ppm [59].

When comparing the input reference voltage to the fundamental voltage component of the PWM signals, directly from the CompactRIO™ system, the phase shift pattern still contained the same oscillatory pattern as was seen when comparing with the inverter output. No significant impact on the phase error could therefore be found to originate from the inverter system. Another function generator was also tested, 50.05, 50.1 and 70 Hz for the reference voltage and a different power supply to the CompactRIO™ chassis was used but the same result was obtained in all cases. The sampling frequency of the DSWM is dependent on the reference frequency and set to capture a certain amount of samples per period. Therefore, different sampling frequencies were tested which should have changed the aliasing effect. No difference was however seen in the phase error. To rule out any other problem with the DSWM, a PC-connected oscilloscope was used with a sampling frequency of 1 MHz. Data was saved over 30 seconds and different window sizes was used for performing the FFT for phase comparison. The same accuracy was not obtained as

with the DSWM but the shape of the phase error however remained the same throughout all different tests.

The digital I/O module is synchronized with the FPGA clock. But the ADC module, as described above, has its own clock from which the sampling frequency is derived and it is asynchronous with the FPGA clock. This can cause a drift between the sampling and the output and is thus a possible source of the phase error [60]. This is also indicated by the results that was shown in Figure 5.12 which originated from the phase tracking and the PWM reference calculation loops executing at different speeds. These loops both used the Loop Timer block with the same delay which aimed to synchronize them. However, Loop Timer is not recommended to use in the same loop that acquires sampled data from the NI 9239 module, as samples may be missed [61]. With Loop Timer removed the execution is instead synchronous with the sampling from the ADC module. The removal of Loop Timer however had no effect on the phase error.

When analyzing the phase shift between the third harmonic and the fundamental component of the output voltage, the same oscillatory pattern as for the phase shift of the fundamental component relative to the reference voltage was seen. This is therefore thought to be the reason to the poor results for the harmonics attenuation test.

The clock of the ADC module cannot be changed. It can however be imported and used in the FPGA code. Attempts were done to use the ADC clock for updates of the PWM output. These were however not successful, but this should be further investigated.

Chapter 6

Conclusions and future work

6.1 Conclusions

A total of four different phase tracking algorithms were investigated. These were the TD-PLL and the IP-PLL which are based on the synchronous reference frame, the E-PLL based on the gradient descent method and the KF-PLL based on the Kalman filter. Simulations with MATLAB® showed that only the IP-PLL, E-PLL and the KF-PLL could be suitable for the calibration system. The TD-PLL gave large errors at small deviations from 50 Hz in the input. This was due to the number of delayed samples not being an integer for some frequencies.

The algorithms were tuned to reach steady state right within 10 seconds at a 1 Hz input frequency step. The IP-PLL had the best performance overall, including inputs with dc offset and low-order harmonic. This is mainly due to its low-pass filters. For a dc offset, the IP-PLL had a standard deviation in the phase error of 50 μrad while the E-PLL had 350 μrad and the KF-PLL had 570 μrad. For a third harmonic at 10 % of the fundamental amplitude, the IP-PLL had a standard deviation in the phase error of 50.3 μrad while the E-PLL had 776 μrad and the KF-PLL had 889 μrad. By adding estimations of dc offset and harmonics to the E-PLL and the KF-PLL, these errors could be completely eliminated and improved the phase tracking of the fundamental component. For the E-PLL however, it was found that the nominal amplitude must be set closely to the actual to avoid convergence problems. The KF-PLL has no such problems but is on the hand harder to tune. It was however shown that the settings of the magnitudes of the process and measurement covariance matrices could be simplified to only setting the ratio between them. With frequency variations at 50 mHz with an amplitude of 4 mHz in the input signals, the KF-PLL was superior with a standard deviation of 48.5 μrad compared to 1150 μrad for the IP-PLL and 811 μrad for the E-PLL. The KF-PLL is therefore the best performing algorithm overall, with improvements included.

The IP-PLL, the E-PLL and the KF-PLL were all implemented with LabVIEW™ code on the FPGA chip of a cRIO-9076 chassis CompactRIO™ system with a NI 9239 ADC module and a NI 9402 digital I/O module. A graphical user interface was designed in which the results of the phase tracking can be viewed. Additional to the original KF-PLL, a simplified KF-PLL was implemented by elimination of the state covariance matrix update. Instead, only steady state values were used that were obtained at 50 Hz. It was shown that this gave a worse transient error and the time to reach steady state was slightly delayed, but equal steady state performance as the original KF-PLL. With this simplification, estimation

of dc offset and three harmonics additional to the fundamental component was fitted on the FPGA chip. Only estimation of dc offset was successfully added to the E-PLL. Tests were done with a function generator as input. The response of the IP-PLL indicated that a dc offset was present in the sampled signal. Different from the simulations, the KF-PLL gave best results with dc offset in the input. The original KF-PLL gave a standard deviation in the phase error of 131 μrad and the IP-PLL gave 1120 μrad, for a dc offset of 0.41 %. The phase error was not calculated for the E-PLL. However, the standard deviation of the frequency was 8.81 mHz for the E-PLL but 549 μHz for the IP-PLL and 63.5 μHz for the original KF-PLL. With the simplified KF-PLL, a standard deviation of 5.1 μrad in the phase error and 4.66 μHz in the estimated frequency was obtained. The IP-PLL with a minimized cut-off frequency of the low-pass filters gave a standard deviation of 49.2 μrad in the phase error and 24.5 μHz in the estimated frequency. For the simplified KF-PLL, also the grid voltage was sampled as an input. The estimated error peaked at 1000 μrad with about 700 μrad in standard deviation. This is close to a fourth of what was obtained in the previous calibration system.

A single-phase full-bridge converter was designed and built to be controlled by the CompactRIO™ system for inverter operation. The phase estimation of the input signal was used to generate the PWM reference signal. The PWM pulses were outputted through the I/O module. In the graphical user interface, possibility to change the phase offset to the input reference signal was added. Additionally, the third, fifth and seventh harmonics can be added to the reference for attenuation of the same harmonics in the inverter output. Simulations showed how all three harmonics, but mainly the third could be attenuated. THD for the non-compensated case was 1.86 %. This decreased to 0.78 %, 0.59 % and finally to 0.52 % when all three harmonics had been compensated. When tested on the inverter during measurements, only the third harmonic was successfully attenuated. THD decreased from 3.92 % to 1.9 %. This is a reduction of 51 % while the simulations gave a reduction of 58 %. The limited attenuation of the low-order harmonics is thought to be partially due to the distortion from the gate driver board which eliminates small gate pulses and gaps. Additionally, in the simulations, the forward voltage drops and resistances of the IGBTs and diodes where static whilst they actually vary with temperature and current level.

The inverter was designed for an output of 91.6 A peak output current or 64.6 A RMS with a dc-link voltage of 354 V at a power factor of 0.9. The output power during the measurements was limited by the amount of ripple in the dc-link voltage that the dc-source could handle. An extra capacitance of 6.6 mF was added to the already installed 13.4 mF for the measurements to smooth out the dc-link voltage further. The highest losses in the measurements were obtained for the highest RMS value of the load current which was 79.03 A at a dc-link voltage of 115 V. This was measured at a power factor of 0.097 with an input power of 862.5 W. The measured losses were 259.9 W while the estimated losses were 279 W. The estimated losses were however calculated to obtain maximum losses which was the method used when choosing the IGBT module. By using three 250/1.8 V transformers in parallel, the output current could be boosted to 2 kA RMS in the large cable loop that is supposed to be the load.

The phase and amplitude jitter was analyzed with the DSWM, which was also used in the previous calibration system as the reference instrument. The aim was to reach a

standard deviation of 20-50 ppm accuracy in the amplitude at an output current of 2 kA. The phase error should have a standard deviation of 20-50 μrad. The amplitude jitter was found to be about 63 ppm. However, only a 1 A output was used. This is equivalent to about 139 A at the secondary side of the boosting transformer. The amplitude jitter is thought to originate mainly from the dc-source used, which had a stability of 50 ppm. The phase jitter was around 560 μrad. This was measured both at the output of the inverter and directly at the output of the CompactRIO™ system. As shown in Figure 5.14, the pattern of the phase error was oscillatory with a frequency of about 0.12 Hz and an amplitude of 0.1°. The ADC module has a clock that is asynchronous with the FPGA clock. It is thought that a drift between these causes the behavior of the phase error.

6.2 Future work

Further research is required to correct the oscillating phase shift between the input and the output from the CompactRIO™ system. This is thought to be related to a drift between the clocks of the ADC module and the FPGA chip. The clock of the ADC module cannot be changed. However, it can be exported and used on the FPGA chip. The FPGA code should be updated to synchronize with the ADC module to see if this changes the behavior of the phase shift. It is however possible that the error is in the implementation of the PWM generation as this could not be ruled out. The behavior was the same for all the different algorithm, these alone are therefore not likely the source of the problem.

A suitable dc source must be obtained. It must specifically be investigated the amount of dc-ripple it can handle. Possibly, additional electrolytic capacitors must be used as was needed when testing the inverter. The voltage stability is directly related to the jitter in the amplitude of the output from the inverter. This must therefore also be carefully considered.

The implementation of harmonics attenuation must be improved as the current system only worked for the third harmonic. It also caused the amplitude of the fundamental component to decrease. The implementation was based on simulations in which the forward voltage drops and resistances of the IGBTs and diodes were fixed. Additionally, the gate driver board introduced second order harmonics that were not considered in the simulations. It is possible that another method must be used to reduce the low-order harmonics. However, the phase shift between the third harmonic and the fundamental component in the output voltage varied with the same shape as the phase shift between the fundamental component and the reference voltage. This is therefore also though to be related to the asynchronous ADC and FPGA clock.

The Kalman filter was implemented with overall good performance. It is however hard to set the values of the process and measurement covariance matrices. There are also other recursive filter algorithms that are based on a discrete time state space model of the system that could be used. One is the recursive least squares (RLS) filter. Instead of the values for the process and measurement covariance, a parameter called the forgetting factor must be set. This can be related to the bandwidth of the filter which makes it easier to tune [62]. However, it is possible that the RLS algorithm, which also consist of many matrix operations, is too complex and cannot be simplified to fit withdc offset and harmonics estimations on the FPGA chip.

References

[1] Swedish Energy Agency, "Långsiktsprognos 2012." https://energimyndigheten.a-w2m.se/Home.mvc?ResourceId=2698, 2013. Accessed 2014-02-17.

[2] Swedish Energy Agency, "Energiläget 2013." https://energimyndigheten.a-w2m.se/Home.mvc?ResourceId=2785, 2013. Accessed 2014-02-17.

[3] A. Lindskog, A. Manikoff, and S. Svensson, "A 600 MVA reference measuring and generation system for high voltage loss measurement under low power factor conditions," in *Metrology 2011*, 2011.

[4] R. E. Best, *Phase-Locked Loops: Design, Simulation, and Applications, Sixth Edition*. New York, USA: McGraw-Hill, 2007.

[5] S. Golestan, M. Monfared, and F. D. Freijedo, "Design-Oriented Study of Advanced Synchronous Reference Frame Phase-Locked Loops," *IEEE Transactions on Power Electronics*, vol. 28, pp. 765 – 778, February 2013.

[6] J. Encinas, *Simplified operation of PLL circuits*. New York, USA: Springer US, 1993.

[7] M. Karimi-Ghartemani, "A Unifying Approach to Single-Phase Synchronous Reference Frame PLLs," *IEEE Transactions on Power Electronics*, vol. 28, pp. 4550 – 4556, October 2013.

[8] S. Golestan, M. Monfared, F. D. Freijedo, and J. M. Guerrero, "Dynamics Assessment of Advanced Single-Phase PLL Structures," *IEEE Transactions on Power Delivery*, vol. 60, pp. 2167 – 2177, June 2013.

[9] B. Thomas, *Modern Reglerteknik*. Stockholm, Sweden: Liber, 2008.

[10] L. G. B. Rolim, J. Diogo Rodrigues da Costa, and M. Aredes, "Analysis and Software Implementation of a Robust Synchronizing PLL Circuit Based on the pq Theory," *IEEE Transactions on Industrial Electronics*, vol. 53, pp. 1919 – 1926, December 2006.

[11] R. Choudhury, *Modern Control Engineering*. New Delhi, India: Prentice Hall, 2005.

[12] S. M. Silva, B. M. Lopes, B. J. C. Filho, R. P. Campana, and W. C. Boaventura, "Performance Evaluation of PLL Algorithms for Single-phase Grid-connected Systems," *Conference Record of the 2004 IEEE Industry Applications Conference, 2004. 39th IAS Annual Meeting.*, vol. 4, pp. 2259 – 2263, October 2004.

[13] M. Karimi-Ghartemani, B.-T. Ooi, and A. Bakhshai, "Application of Enhanced Phase-Locked Loop System to the Computation of Synchrophasors," *IEEE Transactions on Power Delivery*, vol. 26, pp. 22 – 32, January 2011.

[14] M. Karimi-Ghartemani, "A distortion-free phase-locked loop system for FACTS and power electronic controllers," *Electric Power Systems Research*, vol. 77, pp. 1095 – 1100, June 2007.

[15] T. McKelvey, "A Brief Introduction to Kalman Filtering." Course SSY130 - Applied signal processing, November 2011. Dept. of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden.

[16] B. D. Anderson and J. B. Moore, *Optimal Filtering*. New York, USA: Dover Publications, INC, 1979.

[17] U. Dahlbom, *Matematisk statistik*. Gothenburg, Sweden: HB Matematiklitteratur, 2003.

[18] Y. Tong, *The Multivariate Normal Distribution*. New York, USA: Springer-Verlag, 1990.

[19] R. Cardoso, R. F. de Camargo, H. Pinheiro, and H. Gründling, "Kalman filter based synchronisation methods," *Generation, Transmission & Distribution, IET*, vol. 2, pp. 542 – 555, July 2008.

[20] S. Liu, "An adaptive Kalman filter for dynamic estimation of harmonic signals," *8th International Conference On Harmonics and Quality of Power Proceedings, 1998. Proceedings*, vol. 2, pp. 636 – 640, October 1998.

[21] K. D. Brabandere, T. Loix, K. Engelen, B. Bolsens, J. V. den Keybus, J. Driesen, and R. Belmans, "Design and Operation of a Phase-Locked Loop with Kalman Estimator-Based Filter for Single-Phase Applications," *32nd Annual Conference on IEEE Industrial Electronics, IECON 2006*, pp. 525 – 530, November 2006.

[22] A. Routray, A. K. Pradhan, and K. P. Rao, "A Novel Kalman Filter for Frequency Estimation of Distorted Signals in Power Systems," *IEEE Transactions on Instrumentation and Measurement*, pp. 469 – 479, June 2002.

[23] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics - Converters, Applications, and Design, Third Edition*. New Jersey, USA: John Wiley & Sons, 2003.

[24] M. H. Rashid, *Power Electronics Handbook - Devices, Circuits, and Applications (3rd Edition)*. Oxford, UK: Elsevier, 2011.

[25] Arendt Wintrich and Ulrich Nicolai and Werner Tursky and Tobias Reimann, "Application Manual Power Semiconductors," November 2011.

[26] Mitsubishi Electric, "General Considerations for IGBT and Intelligent Power Modules," September 1998. Application note.

[27] Semikron, "IGBT Driver Calculation," October 2007. Application note.

[28] National Instruments, "What Is NI CompactRIO?." http://www.ni.com/compactrio/whatis/, 2014. Accessed 2014-03-31.

[29] National Instruments, "NI LabVIEW for CompactRIO Developer's Guide." http://www.ni.com/pdf/products/us/fullcriodevguide.pdf, 2014. Accessed 2014-05-24.

[30] C. Maxfield, *FPGAs: Instant Access*. Oxford, UK: Newnes, 2008.

[31] National Instruments, "IP Corner: The LabVIEW Fixed-Point Data Type Part 1 – Fixed-Point 101." http://www.ni.com/newsletter/50303/en/, December 2011. Accessed 2014-03-21.

[32] National Instruments, "What Is LabVIEW?." http://www.ni.com/newsletter/51141/en/, August 2013. Accessed 2014-05-29.

[33] Daniel Reinholz, "Euler's Method." http://www.ocf.berkeley.edu/~reinholz/ed/08fa_m160/lectures/eulers_method.pdf, February 2010. Accessed 2014-06-17.

[34] Mathworks, "Continuous-Discrete Conversion Methods." http://www.mathworks.se/help/control/ug/continuous-discrete-conversion-methods.html, 2014. Accessed 2014-06-13.

[35] C. Nordling and J. Östermalm, *Physics Handbook for Science and Engineering*. Lund, Sweden: Studentlitteratur, 2006.

[36] D. K. Cheng, *Field and Wave Electromagnetics, second edition*. Reading, USA: Addison-Wesley Publishing Company, 1989.

[37] R. Lundin, "A Handbook Formula for the Inductance of a Single-Layer Circular Coil," *Proceedings of the IEEE*, vol. 73, pp. 1428 – 1429, September 1985.

[38] M. Salcone and J. Bond, "Selecting Film Bus Link Capacitors For High Performance Inverter Applications," *IEMDC '09. IEEE International Electric Machines and Drives Conference, 2009*, pp. 1692 – 1699, May 2009.

[39] M. N. Anwar and M. Teimor, "An Analytical Method for Selecting Dc-Link-Capacitor of a Voltage Stiff Inverter," *Conference Record of theIndustry Applications Conference, 2002. 37th IAS Annual Meeting*, vol. 2, pp. 803 – 810, October 2002.

[40] F. Renken, "The DC-Link Capacitor Current in Pulsed Single-Phase H-Bridge Inverters," *2005 European Conference on Power Electronics and Applications*, September 2005.

[41] S. P. Valsan, T. Vaidya, and M. Chaudhary, "Modified Reference PWM for Harmonic Reduction in Inverters," *2010 India International Conference on Power Electronics (IICPE)*, pp. 1 – 4, January 2011.

[42] National Instruments, "CompactRIO Integrated Systems with Real-Time Controller and Reconfigurable Chassis NI cRIO-907x," February 2014. Accessed 2014-03-28.

[43] National Instruments, “NI 9229/9239 - Operating instructions and specifications,” November 2012. Manual.

[44] National Instruments, “NI 9402 - Operating instructions and specifications,” February 2009. Manual.

[45] National Instruments, “Using the Single-Precision Floating-Point Data Type (FPGA Module).” http://zone.ni.com/reference/en-XX/help/371599H-01/lvfpgaconcepts/fpgasingleprecisfloat/, June 2012. Accessed 2014-05-17.

[46] EPCOS, *B43456A9688M*, November 2012. Datasheet.

[47] Vishay, *MKP1848622454P4*, November 2012. Datasheet.

[48] Semikron, “About semikron.” http://www.semikron.com/skcompub/en/about_semikron-71.htm, 2014. Accessed 2014-03-19.

[49] Semikron, *SK100GB066T*, October 2009. Datasheet.

[50] Semikron, *SK80GB063*, November 2006. Datasheet.

[51] Semikron, *SKM100GB063D*, September 2006. Datasheet.

[52] Semikron, *SK150GB066T*, June 2009. Datasheet.

[53] Semikron, *SKM145GB066D*, October 2009. Datasheet.

[54] Semikron, *SKHI 23/17R*, June 2006. Datasheet.

[55] Semikron, *KL 285 P3 300 mm*, March 2005. Datasheet.

[56] Texas Instruments, *UCC27524*, June 2012. Datasheet.

[57] Semikron, *SKF 3-230-01*, May 2003. Datasheet.

[58] Wavetek Ltd, *Model 29 10 MHz DDS Function Generator*, 1.1 ed., 1997. Maintenance manual.

[59] Delta Elektronika, *SM 300-20*, September 2012. Datasheet.

[60] National Instruments, “Reference Applications for cRIO Order Analysis Using a High Speed Tach.” http://www.ni.com/example/31301/en/, April 2010. Accessed 2014-06-08.

[61] National Instruments, “Understanding Loop Timing for the NI 9225/9227/9229/923x (FPGA Interface).” http://zone.ni.com/reference/en-XX/help/370984R-01/criodevicehelp/923x_understanding_loop_timing/, October 2009. Accessed 2014-06-09.

[62] M. Beza and M. Bongiorno, “Application of Recursive Least Square (RLS) Algorithm with Variable Forgetting Factor for Frequency Components Estimation in a Generic Input Signal,” *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 2164 – 2171, September 2012.

[63] National Instruments, "Re: Problems about producing SPWM." `https://decibel.ni.com/content/message/66456#66456`, February 2014. Accessed 2014-05-17.

[64] National Instruments, "Understanding Timing Considerations for FPGA VIs." `http://zone.ni.com/reference/en-XX/help/371599G-01/lvfpgaconcepts/registers/`, June 2011. Accessed 2014-06-02.

[65] National Instruments, "High Throughput Math Functions." `http://zone.ni.com/reference/en-XX/help/371599H-01/lvfpga/ht_math_pal/`, June 2012. Accessed 2014-05-28.

Appendix A

MATLAB® code

A.1 The Kalman filter

The KF-PLL as implemented with MATLAB®. For simulations presented in Chapter 3.1.3.

```
function [X_k, est_phase_PLL, est_frequency_PLL, phase_error_PLL] = ...
    kalmanfilter_postPLL(Y_k,A,C,Q,R,x0,P0,Ts,w0,Xi,wn)
% INPUTS
%     Y: Measurement data
%     A: Transition matrix
%     C: Measurement matrix
%     Q: Covariance matrix of the process noise
%     R: Covariance matrix of the measurement noise
%     x0: State initiates
%     P0: State variance initiate
%     Ts: Sampling time, used for determination of PLL controller values
%     w0: Nominal frequency, used in PLL
%     Xi: Damping constant, used for determination of PLL controller values
%     wn: Bandwidth, used for determination of PLL controller values
% OUTPUTS
%     X_k: Kalman state estimates
%     est_phase_PLL: PLL phase estimates [rad]
%     est_frequency_PLL: PLL frequency estimates [Hz]
%     phase_error_PLL: PLL estimation phase error [rad]

[~,N] = size(Y_k); % number of samples
n = length(A); % number of states
X_khat = zeros(n,N+1); % memory allocation for Kalman predicted states
X_k = zeros(n,N+1); % memory allocation for Kalman filtered states

% KF initialization:
X_k(:,1) = x0;      % first estimation
P = P0;             % initial covariance matrix
X_khat(:,1) = A*x0; % first prediction
P = A*P*A' + Q;     % update of covariance matrix

% PLL controll parameters:
Ki = wn^2; % integral gain for frequency updating PLL
Kp = 2*Xi*wn; % proportional gain for frequency updating PLL
est_frequency_PLL = zeros(1,N); % estimated frequency vector allocation
phase_error_PLL = zeros(1,N); % phase error vector allocation
est_phase_PLL = zeros(1,N); % estimated phase vector allocation
est_phase_PLL(1) = 0; % initial phase
I_e = 0; % integrated error initiation
I_w = 0; % integrated phase initiation

% KF-PLL iterations:
for k=2:N+1
```

```
    % KF:
    X_k(:,k) = X_khat(:,k-1) + P*C'*(C*P*C' + R)^-1*(Y_k(:,k-1) - C*X_khat(:,k-1));
        P = P - P*C'*(C*P*C' + R)^-1*C*P; % state variance update
    X_khat(:,k) = A*X_k(:,k); % predicted state
        P = A*P*A' + Q; % predicted variance

    % SRF-PLL:
    alpha = X_k(2,k); % cosine
    beta = X_k(1,k); % sine
    u_dq = (alpha + 1i*beta)*(cos(est_phase_PLL(k-1)) - 1i*sin(est_phase_PLL(k-1)));
    phase_error_PLL(k-1) = angle(u_dq); % estimated phase error
    e_phase = phase_error_PLL(k-1); % PLL input error
    I_e = I_e + Ki*Ts*e_phase; % integral controller
    I_p = Kp*e_phase; % proportional controller
    est_frequency_PLL(k-1) = w0 + I_p + I_e; % estimated frequency
    I_w = I_w + Ts*est_frequency_PLL(k-1); % phase estimation integration
    if k <= N % don't save data last iteration
        est_phase_PLL(k) = I_w; % phase estimate
    end

    % Update of transition matrix:
    wk = est_frequency_PLL(k-1);
    A = [cos(wk*Ts) sin(wk*Ts) ;...
        -sin(wk*Ts) cos(wk*Ts) ];

    est_frequency_PLL(k-1) = est_frequency_PLL(k-1)/(2*pi); % rad/s to Hz

end
```

Appendix B

Kalman filter gain factor coefficients

Steady-state coefficients for the gain factor $\boldsymbol{G}_k$ of the KF-PLL. Used in the simplified KF-PLL as presented in Chapter 4.1.2. The coefficients were obtained by running the KF-PLL algorithm for 20 seconds with according number of states included for a fundamental frequency at 50 Hz. Only odd harmonics were tested. Since the elements of the transition matrix $\boldsymbol{A}$ are dependent on the sampling time, simulations were done for both 10 and 25 kHz sampling frequency. Table B.1 shows the values at 10 kHz sample rate and Table B.2 at 25 kHz sample rate. The first index of the coefficient is the harmonic and the second index refers to the in-phase (1) or orthogonal (2) state. It is seen how the in-phase component variances are similar overall but the variances of the orthogonal component varies significantly when additional states are included.

Table B.1 Steady-state coefficients for $\boldsymbol{G}_k$ at 10 kHz.

	Included harmonics			
	3rd	*3rd & 5th*	*3rd, 5th & 7th*	*3rd, 5th, 7th & 9th*
G_{dc}	0.000998088455	0.000997383103	0.000996678242	0.000995973878
$G_{1,1}$	0.001410523695	0.001409623485	0.001408673786	0.001407705610
$G_{1,2}$	0.000052763934	0.000500773370	0.000048715523	0.000047883678
$G_{3,1}$	0.001410823643	0.001410147968	0.001409251869	0.001408304854
$G_{3,2}$	0.000044020182	0.000045098181	0.000027280142	0.000024604280
$G_{5,1}$		0.001409791567	0.001409156457	0.001408277516
$G_{5,2}$		0.000045098181	0.000031829528	0.000026122228
$G_{7,1}$			0.001408720737	0.001408123529
$G_{7,2}$			0.000047338404	0.000033407502
$G_{9,1}$				0.001407648724
$G_{9,2}$				0.000049527800

Table B.2 Steady-state coefficients for $\boldsymbol{G}_k$ at 25 kHz.

	Included harmonics			
	3rd	*3rd & 5th*	*3rd, 5th & 7th*	*3rd, 5th, 7th & 9th*
G_{dc}	0.000998092785	0.000997388230	0.000996684131	0.000995980517
$G_{1,1}$	0.001405449615	0.001405055059	0.001404350260	0.001403527352
$G_{1,2}$	0.000130727944	0.000124043717	0.000120660107	0.000118597436
$G_{3,1}$	0.001407249718	0.001408253447	0.001407882824	0.001407191986
$G_{3,2}$	0.000109666070	0.000079929916	0.000068003456	0.000061360363
$G_{5,1}$		0.001406021507	0.001407277483	0.001407011271
$G_{5,2}$		0.000112561478	0.000079552503	0.000065373264
$G_{7,1}$			0.001404547833	0.001406039567
$G_{7,2}$			0.000118338074	0.000083708516
$G_{9,1}$				0.001403059281
$G_{9,2}$				0.000124012239

Appendix C

LabVIEW™ code

C.1 RT controller

Figure C.1 shows the initialization code for the IP-PLL. The selector labeled *Choice of PLL* is connected to a case structure that shows only the chosen code. Code for initialization of the other phase tracking algorithms are similar. Only the IP-PLL has the math script module for calculation of low low pass filter coefficients. The $c2d$ function is used to discretize the continuous time filter function to acquire the parameters a and b. Three wires continue to the main code. The green wire is the reference to the FPGA and the thick yellow wire contains error signals.

After the initialization block are the First In First Out (FIFO) memory buffers configured and started before the three main loops. Two loops are for network communication and one is for reading data from the FPGA FIFOs. The loops have relative priority to each other where only the loop with the highest priority has guaranteed timing [29]. This is seen in Figure C.2.

The top priority loop of the RT controller is the middle loop for acquisition of data from the FIFOs that the FPGA writes to. It has a relative priority of 255 as seen in its top left corner. If the RT controller does not read data fast enough, the buffer will be filled and data will be lost.

The two remaining loops are for network communication. The top loop is updated each 100 ms with the lowest priority of 90. This loop buffers data to a network stream in a cluster that contains four one dimensional arrays of data from the FPGA. The bottom loop has a higher priority (100) and updates with a frequency to match the sampling rate. Eight network variables are used to read single values that are set at the host computer. These are settings for the PWM references and are written to the FPGA.

C.2 Host computer

The code for the main user interface on the host computer is seen in Figure C.3. It is placed in a sequence structure of three frames. One initialization frame, one main frame and one closing frame. An endpoint of the network stream setup on the RT controller is created on the host computer. This reads all the buffered data originally from the FPGA. Data is presented in charts and also saved to text files if *Write to file?* is True. The lower loop updates the PWM reference settings each 100 ms. The phase error and the frequency are

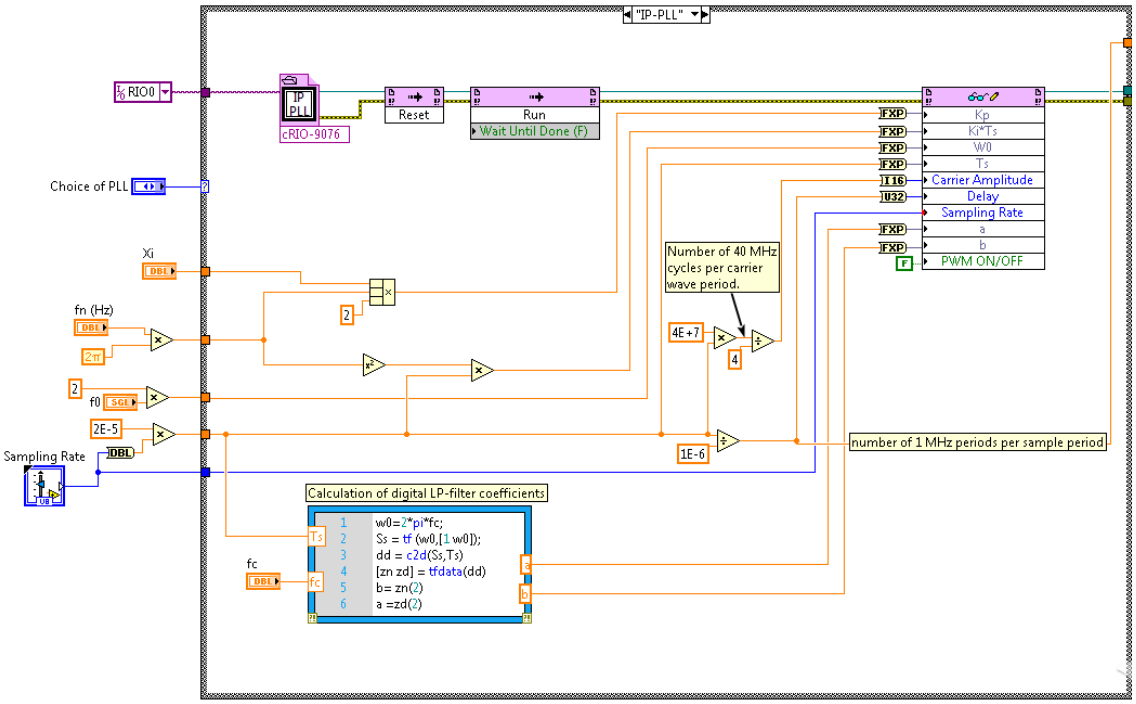


Figure C.1 LabVIEW™ code for initialization and starting of the IP-PLL algorithm on the FPGA from the RT controller.

transformed to correct values at the host computer to relieve the CompactRIO™ unit from unnecessary operations. For trigonometric operations, the FPGA works with π radians rather than radians to save resources. This is equal to dividing any angle with π. Therefore the phase error must be multiplied with π and the angular frequency is changed to the unit of Hz by division with two.

C.3 FPGA

The FPGA code consist of four loops. Three loops for PWM that are common for all the phase tracking algorithms and one loop for phase tracking. The phase tracking loop outputs the estimated phase to the PWM controller by writing to the global variable *phase*. In addition, it writes data of the input signal, the estimated signal, the estimated frequency and the estimation error to FIFOs that are read by the RT controller.

All subVIs and all arithmetic operators in the FPGA code are non-reentrant. This means that there is only one instance of the code on the FPGA and only one process can access it at a time. If the subVI is reentrant, it exists in multiple setups on the FPGA. This gives the fastest execution but also takes up more physical resources of the FPGA [29] [63]. Without this solution, all the algorithms would not fit on the FPGA chip. Initial resource estimations were several 100 % of the available space. Even though the execution speed of the algorithms are slowed down, all fit in to a 50 kHz period which is the maximum sampling rate of the ADC module. Moreover, the gate driver board can only support the IGBTs up to 45 kHz as was mentioned in Chapter 4.2.3.

Additional to lack of space, timing violation was a common issue when compiling the code for the FPGA chip. When functions are executed on the FPGA chip, registers are

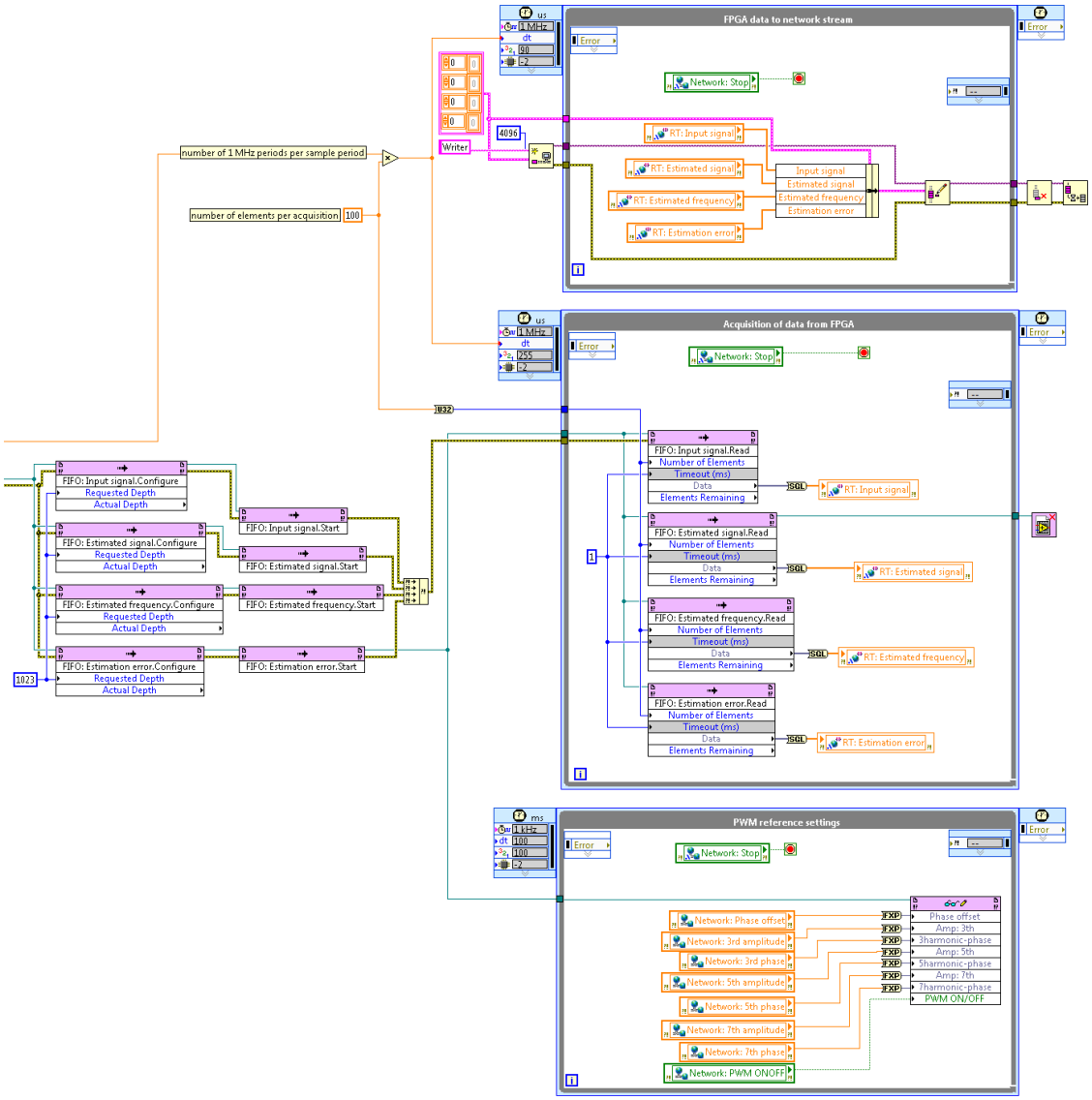


Figure C.2 LabVIEW™ code for the main loops of the RT controller VI.

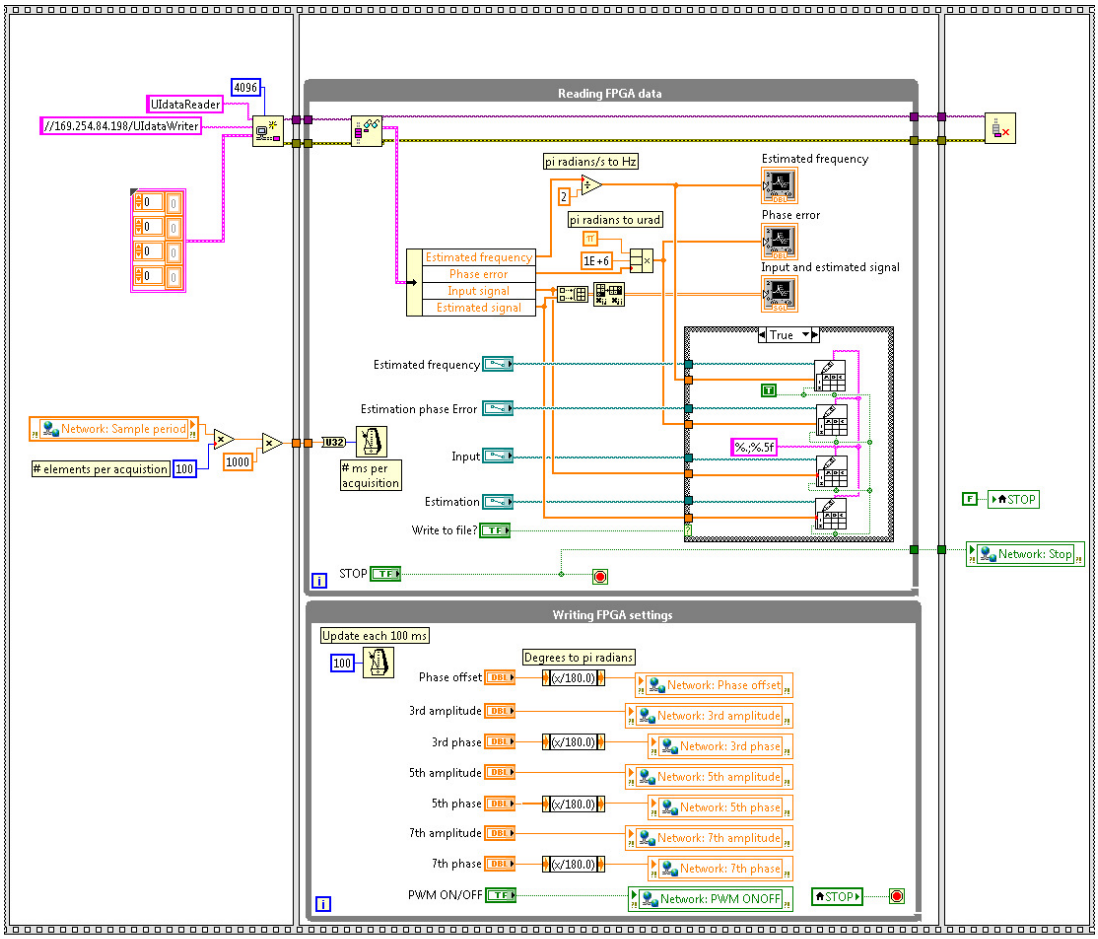


Figure C.3 LabVIEW™ code for the main user interface.

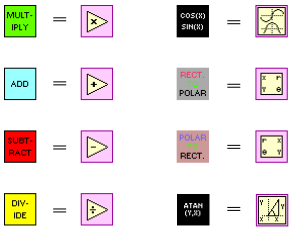


Figure C.4 LabVIEW™ code blocks for math operations used as non-reentrant subVIs in the FPGA code.

used between the different logic blocks. These store the data between clock cycles. The data must manage to travel between two registers during one clock cycle. The maximum clock frequency possible is limited by the longest travel time between two consecutive registers on the same data path. This becomes a problem when the space utilization is large and the number of alternative routes are small. This is solved by changing the code such as adding extra registers or other functional blocks to store data between iterations [64].

The original blocks and corresponding subVIs for math operations are seen in Figure C.4. These are all high-throughput math blocks for operations with the fixed-point data type on the FPGA. They do not work on the RT controller [65]. The configuration of the data type of the inputs and outputs can be changed. The four blocks to the left are for simple arithmetic operations. These all take one clock cycle to execute. The four blocks to the right are for trigonometric operations. Their execution time is dependent on the size of the data type. The polar-to-rectangular and the rectangular-to-polar converter blocks take 68 cycles to execute with 64 bit input and outputs. The sine and cosine and the two-argument arctangent blocks take 65 cycles to execute as most.

C.3.1 PWM controller

The three PWM loops are seen in Figure C.5 together with initiation of the NI 9239 ADC module outside the loop. The top loops are single-cycled timed loops and they execute at 40 MHz which is the base FPGA clock speed. The bottom loop is a regular while loop that executes at the sampling rate. The error signal, from the ADC starting block, is continued to the phase tracking loop for synchronized start of execution.

In the PWM reference calculations loop, the phase is read from a global variable that is written to in the phase tracking loop. Phase offset is added and harmonics references are generated and subtracted from the fundamental reference. The result is multiplied with the carrier wave amplitude for unity modulation index. This is written to the global variable *Reference_A* and its inverse to *Reference_B*. In the PWM signals generation loop, the references are read and compared to the carrier wave running at a fixed period depending on the sampling rate. The boolean results are written to the global variables *PWM_A* and *PWM_B* respectively. In the PWM signals output loop the respective PWM signals and inversions are written to the NI 9402 module outputs.

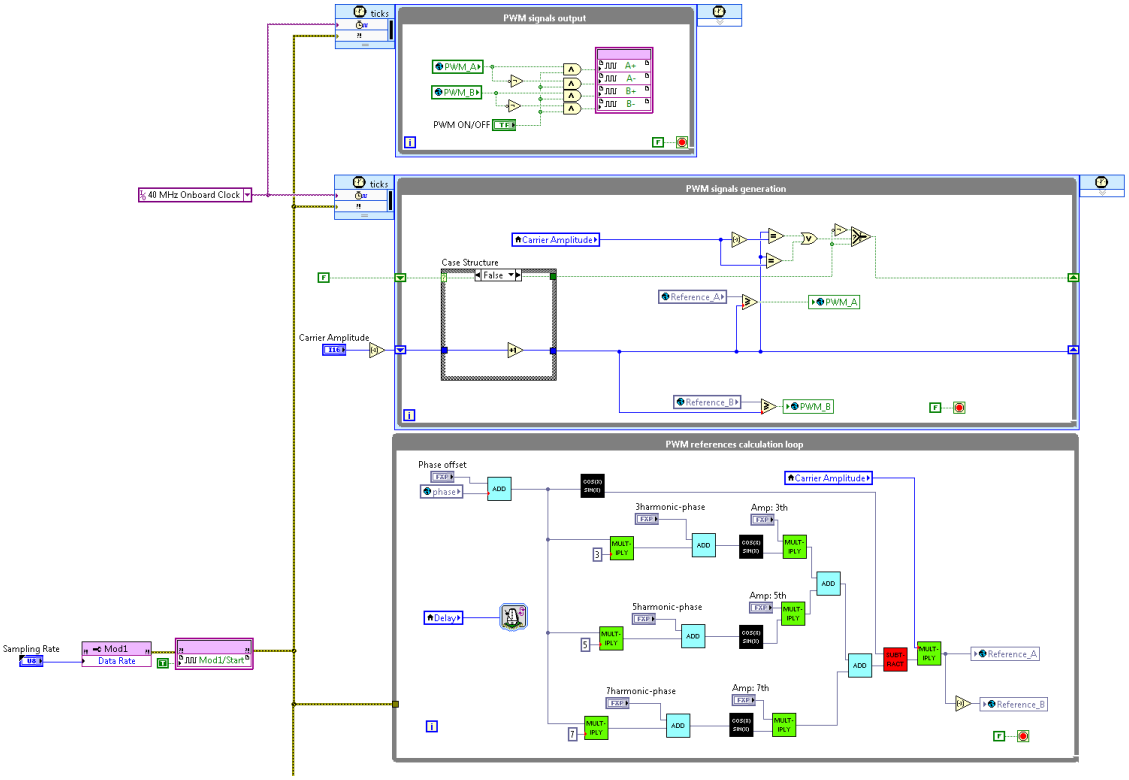


Figure C.5 LabVIEW™ code for generation and output of PWM signals on the FPGA.

C.3.2 The inverse Park PLL

The IP-PLL has one subVI for the low-pass filter. This is blue and marked LPF. The main code is seen in Figure C.6 and the low-pass filter subVI in Figure C.7. The blocks with arrows on them are feedback node that stores data for one iteration and are therefore equal to unit delays. The phase estimation is wrapped to [-2,2] to fit to the output of the rectangular-to-polar block which cannot be changed. It also keeps control of the phase value to overflow properly.

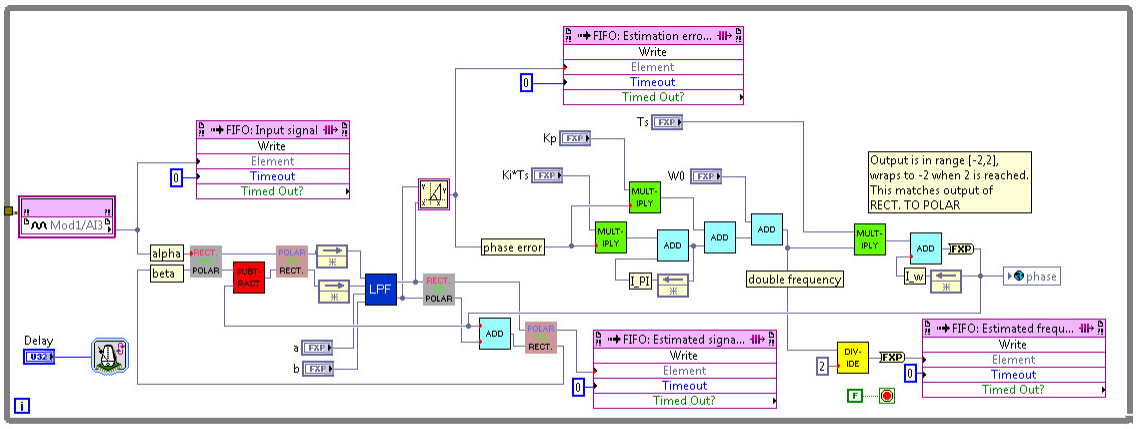


Figure C.6 LabVIEW™ code for The IP-PLL FPGA implementation.

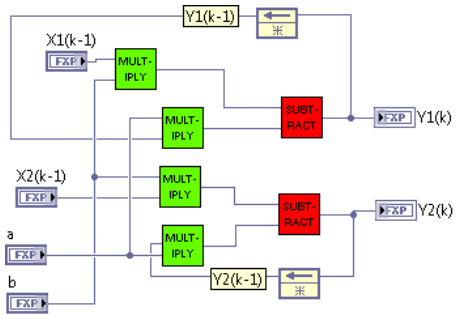


Figure C.7 LabVIEW™ code for the digital low-pass filter FPGA implementation used as a subVI in the IP-PLL.

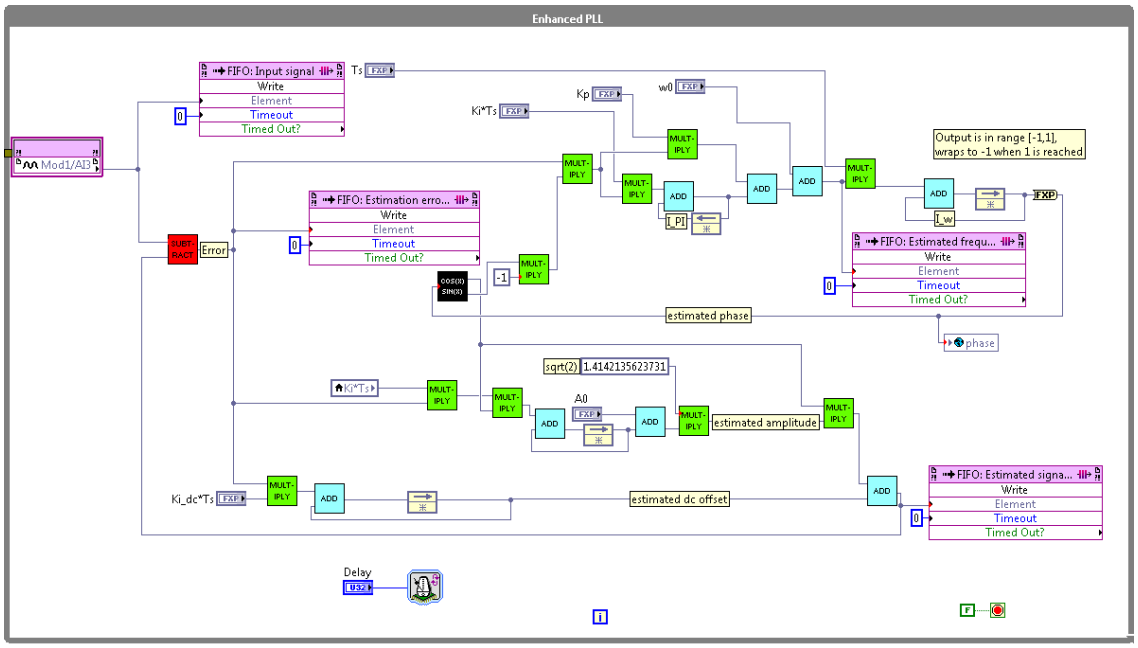


Figure C.8 LabVIEW™ code for the E-PLL FPGA implementation.

C.3.3 The enhanced PLL

The E-PLL has no subVIs and is seen in Figure C.8. The upper part is the phase estimator, the middle part is the amplitude estimator and the bottom part is the dc-offset estimator.

C.3.4 The Kalman filter PLL

The original KF-PLL with only estimation of the fundamental component is seen in Figure C.9. Instead of feedback nodes, shift registers are used to store data between iterations. These are the arrow nodes at the far right of the loop which stores value to be read at the far left of the loop in the next iteration. The left part of the code is update of the estimation. In the middle part there is state prediction and update of the state covariance matrix $\boldsymbol{P}$. This is followed by the subVIs for the SRF-PLL at the top, seen in Figure C.10 and prediction of $\boldsymbol{P}$, seen in Figure C.11, in the bottom.

The majority of the code is for the matrix multiplications related to $\boldsymbol{P}$. One simplification has been made which is that the matrix is symmetric and thus is the top right ($P12$) and bottom left ($P12$) variables equal. Since the simplified KF-PLL gave good results, the

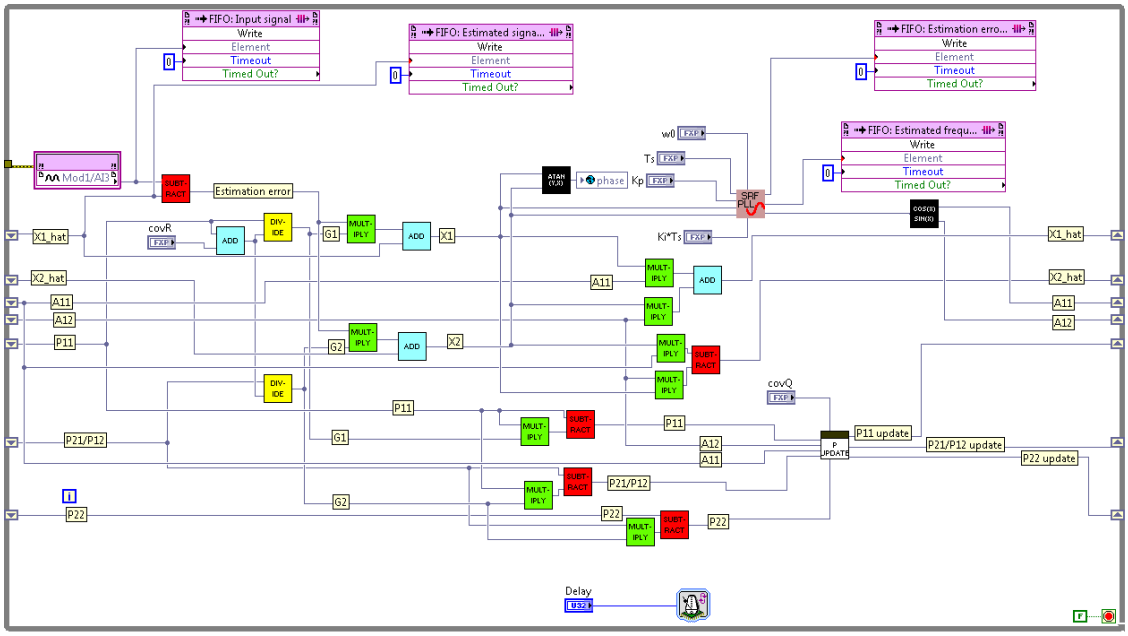


Figure C.9 LabVIEW™ code for the KF-PLL FPGA implementation.

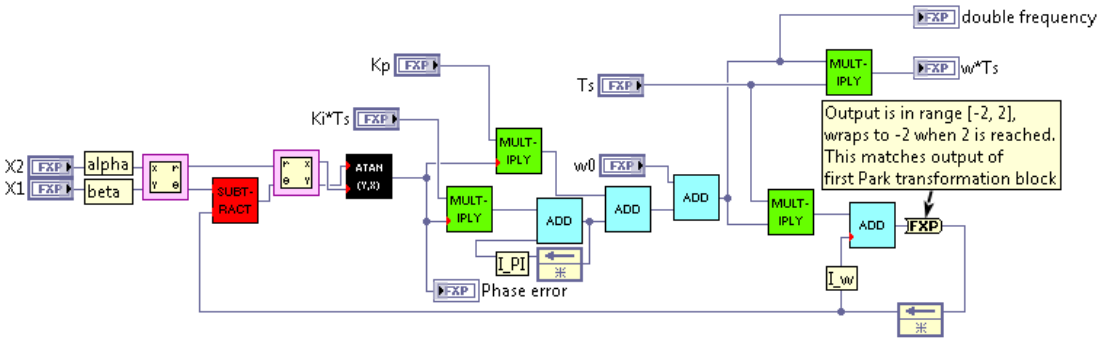


Figure C.10 LabVIEW™ code for the SRF-PLL FPGA implementation used as a subVI in the KF-PLL and the simplified KF-PLL.

code for the original algorithm was not improved.

The simplified KF-PLL with estimation of dc-offset and three harmonics is seen in Figure C.12. Additional subVIs to the original algorithm is state update, state prediction and harmonics estimation. These are seen in Figure C.13, Figure C.14 and Figure C.15 respectively. All feedback nodes are put outside the subVIs to store data that is not common for the different states.

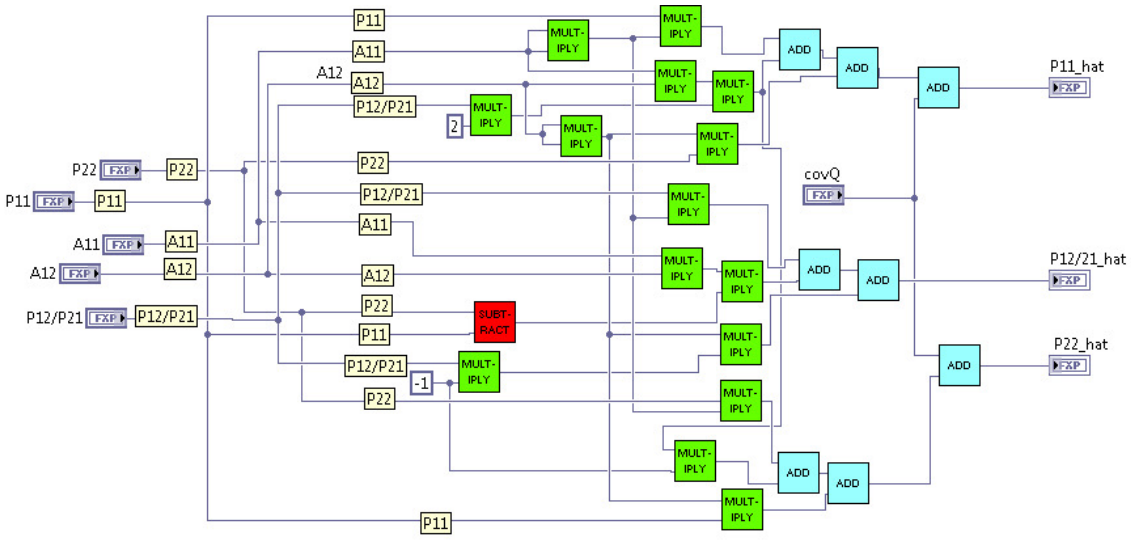


Figure C.11 LabVIEW™ code for the $\boldsymbol{P}$ prediction implementation used as a subVI in the KF-PLL.

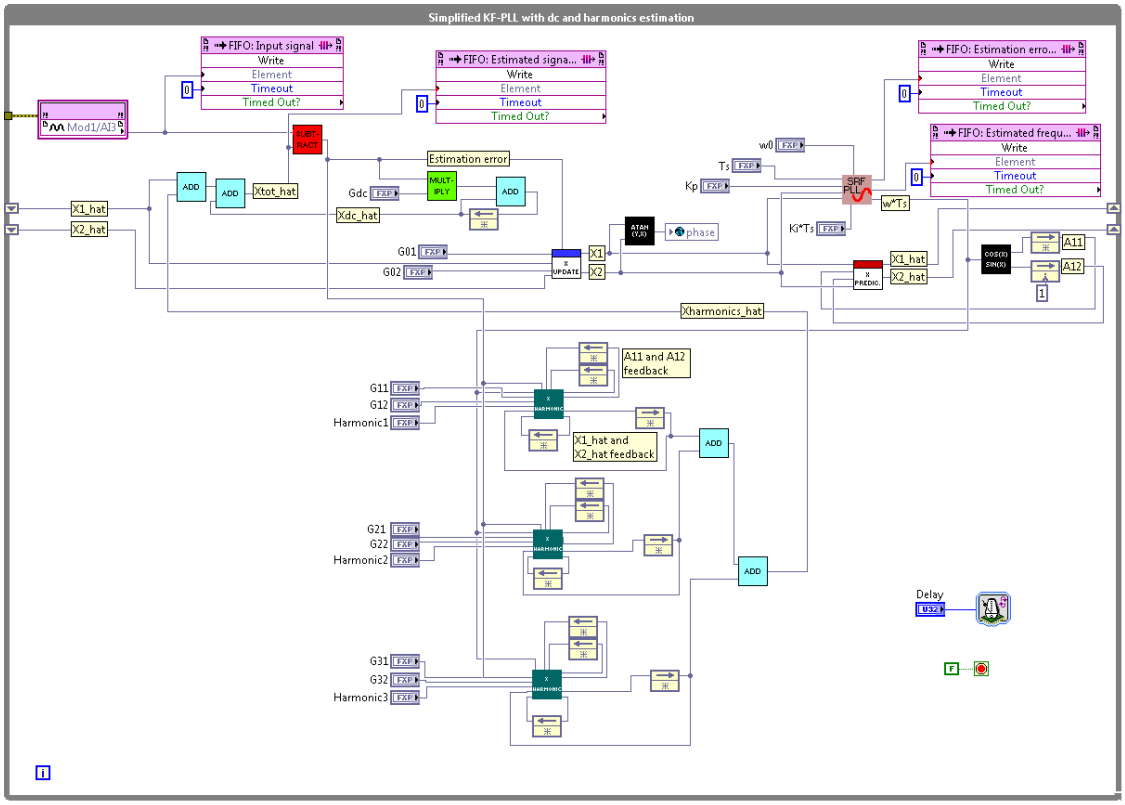


Figure C.12 LabVIEW™ code for the simplified KF-PLL FPGA implementation.

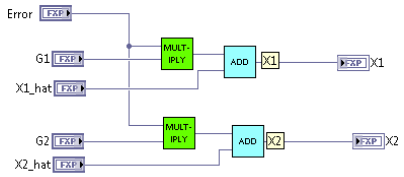


Figure C.13 LabVIEW™ code for the state update FPGA implementation used as a subVI in the simplified KF-PLL.

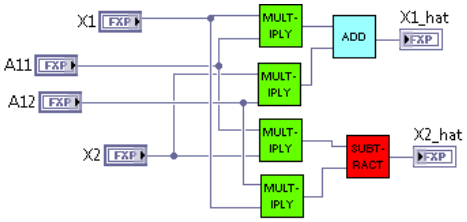


Figure C.14 LabVIEW™ code for the state prediction FPGA implementation used as a subVI in the simplified KF-PLL.

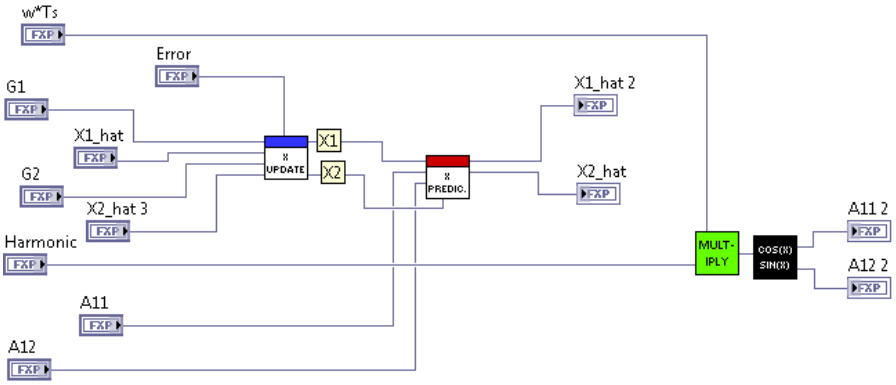


Figure C.15 LabVIEW™ code for the harmonics estimation FPGA implementation used as a subVI in the simplified KF-PLL.