IMPLEMENTATION AND CONTROL OF DISTRIBUTED MAXIMUM POWER POINT TRACKING IN SOLAR PHOTOVOLTAIC APPLICATIONS

BY

MARCEL SCHUCK

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Adviser:

Assistant Professor Robert C. N. Pilawa-Podgurski
ABSTRACT

Maximum power point tracking (MPPT) is an important feature in solar photovoltaic (PV) systems as it increases the energy yield from such systems by optimizing their operating point. Current state-of-the-art solutions perform MPPT for a whole system, a string of panels or an individual panel within a PV system. This results in a loss of energy when the optimal operating point varies among different units. In this thesis, the implementation of MPPT on the sub-panel level is presented. This enables increased energy output under mismatched conditions (e.g. due to partial shading) across the system. Three centrally controlled dc-dc buck converters with series connected outputs are implemented and tested.

As the maximum power point (MPP) of panel sections may vary, the proposed multiphase converters operate under asymmetric conditions. Therefore, a control technique to minimize the output current ripple under these conditions is mathematically derived and verified through simulations and experimental measurements.
ACKNOWLEDGMENTS

Thanks to Robert C.N. Pilawa-Podgurski, my thesis adviser, with whom I worked for the past one and a half years. He showed me the way to experimentally driven research in the field of power electronics. I am grateful for the given opportunities to learn from him as a great researcher and appreciate the openness he showed towards the research projects I pursued during my time in graduate school.

Moreover, I want to thank all my fellow researchers in the group. It has always been inspiring and eye-opening to discuss research related topics. Beyond the scope of our daily work, I am happy that I had the chance to get in contact with so many different people with great personalities.

I would like to thank all the people I met throughout my coursework in graduate school and that made a significant impact on my perception of the world of learning. Moreover they opened my eyes to the different techniques of problem solving among different nationalities. I especially want to thank Christopher Valicka and Sheikh Mashrafi for their help and the productive group work.

My closest friend throughout the past two years, Christoph Scheidiger, opened my eyes in many ways and significantly shaped my perception of the world. His presence made it easy to overcome hurdles no matter if in research or other areas of life. Our time in the United States is an unforgettable experience for me that will stick to my memory for the rest of my life.

I would like to thank my family for their support. Even though being far away apart has not always been easy for them and me, it was always good to know of their help and love.
# TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION ........................................... 1  
1.1 Introduction .................................................. 1  
1.2 Organization of this Thesis ................................. 2  

CHAPTER 2 DISTRIBUTED MAXIMUM POWER POINT TRACK-  
ING FOR SOLAR PHOTOVOLTAIC SYSTEMS .................. 4  
2.1 Motivation .................................................... 4  
2.2 Solar Photovoltaic Characteristics and Models ............ 4  
2.3 System Topologies ........................................... 10  
2.4 Sub-Module Distributed Maximum Power Point Tracking ... 13  

CHAPTER 3 CIRCUIT DESIGN FOR SUB-MODULE MPPT ...... 15  
3.1 Requirements ................................................ 15  
3.2 Converter Topology Selection ............................... 16  
3.3 System Level Considerations ............................... 17  
3.4 Application Specific Circuits ............................... 20  
3.5 Converter System Circuit ................................ 25  

CHAPTER 4 IMPLEMENTATION OF SUB-MODULE MPPT ...... 30  
4.1 Discrete Hardware Implementation .......................... 30  
4.2 MPPT Control ............................................... 44  
4.3 Automatic Bypass Mode Control ............................ 46  

CHAPTER 5 INTERLEAVED OPERATION OF MULTIPHASE  
DC-DC CONVERTERS ................................................. 49  
5.1 Concept Description ........................................ 49  
5.2 Shortcomings in the Given and Other Application ....... 52  
5.3 Motivation of a Frequency Domain Based Ripple Cancell-  
ation Technique ................................................. 56  

CHAPTER 6 IMPROVED CURRENT RIPPLE CANCELLATION  
FOR ASYMMETRIC INTERLEAVED CONVERTERS ............. 58  
6.1 Mathematical Derivations .................................. 58  
6.2 Buck Converter Application Example ....................... 71
CHAPTER 1
INTRODUCTION

1.1 Introduction

As the world’s demand for electrical energy is rapidly increasing, the prices for conventionally produced electrical energy are rising as well. This, along with environmental concerns, has caused a demand to shift away from producing energy from fossil fuels towards using renewable energies on a widespread basis. Among these renewable energy sources, solar photovoltaic (PV) has seen a fast development in recent years, which helped to increase its practical use for producing alternative energy.

This research is focused on a novel concept of using distributed power electronics together with the inverter, as used in state of the art solar systems, to increase the overall energy yield. Maximum power point tracking (MPPT) is performed on a sub-module level of a solar panel to increase energy yield in situations where there is a mismatch of the energy production between different solar cells. An interleaved multiphase dc-dc buck switching converter, which fits into the junction box of a conventional PV panel, is used to implement the MPPT. Theoretical considerations are verified by laboratory experiments that have been performed using an experimental prototype of the proposed system.

Moreover, a new control technique for the proposed distributed power electronic system is presented. Using interleaved multiphase converters at asymmetric operating points is a particularity of the considered application. The control technique takes this into account and helps to increase the system efficiency while reducing voltage- and current-ripples in the circuit. The control is universally applicable and its scope is not limited to the application presented here.
1.2 Organization of this Thesis

This thesis can be divided up into three major parts.

The first part, including Chapters 2 to 4, gives a detailed description of the proposed distributed power electronic system to improve solar photovoltaic system energy yield. The design process is outlined along with practical design considerations.

The second part, consisting of Chapters 6 and 7, describes the novel control technique that allows for improved ripple cancellation for the system under the given operating conditions. It also shows how the scope of the proposed technique can be generalized and extended to other applications, while verifying its performance through simulations as well as experimental results.

The third part, consisting of Chapter 8 outlines the practical characterization of the designed hardware through experimental results. The measurements are used to evaluate the performance of the system.

Chapter 2 presents the state of the art in solar photovoltaic systems. The basic characteristics of a photovoltaic cell are outlined and possible system topologies to interface solar cells and the power grid are presented. Consequently, the usage of distributed power electronics is motivated.

The circuit design requirements for implementing the proposed power electronic system are outlined in Chapter 3. Common converter topologies in power electronics are analyzed and rated based on their suitability for the given application. After selecting a suitable converter circuit, the different possibilities of interconnecting these converters, to achieve the desired functionality, are outlined. Two application specific circuits, namely a signal level shifting circuit and a bypass MOSFET gate driving circuit, are presented. These circuits generally do not see widespread use and had to be specifically designed and adjusted to meet the given requirements of the application at hand.

In Chapter 4 the discrete hardware implementation is shown. The process of loss and efficiency calculation is described and important considerations for component selection are given. Moreover, the fundamental requirements for efficient printed circuit board design of power electronic circuits are stated. Additionally, the chapter describes the control algorithms that are running on the microcontroller that outputs the control signals for the circuit.
Chapter 6 motivates the need for an optimized control technique to achieve improved ripple cancellation under the operating conditions dictated by the application described in the first part of the thesis. It shows the detailed steps of the mathematical derivation of the technique and evaluates its performance on a theoretical basis.

In addition, Chapter 7 shows the performance of the technique through simulations and experimental results. After a discussion of the achieved performance, areas for future work in the field are pointed out.

To evaluate the performance of the designed circuit and compare it to existing systems in solar photovoltaics, experimental results are shown in Chapter 8. After describing the measurement setups and presenting the results of different laboratory and field measurements, the performance of the proposed system is discussed. Finally, the chapter shows future research needs and opportunities.
2.1 Motivation

As solar PV sees increasingly widespread use as an alternative energy source, the price for the required equipment has significantly dropped. Nevertheless, it is competing against conventional forms of energy production and other renewable energies, causing economic considerations to be of high importance during the planning process of a PV system. The return on investment is usually quantified by using the energy yield over a certain period of time per installed kilo-watt-peak (kWp) [1], [2]. To maximize this energy yield of a PV system, it is necessary to operate all PV cells in the system at its MPP. This requires additional power electronics, performing MPPT, which will introduce losses into the system, due to the degree of efficiency of the employed circuitry. Furthermore, it increases overall system cost. For this reason, MPPT is usually not performed on the cell level but for units that comprise several cells. Such units can be sub-modules, panels, strings consisting of multiple series-connected panels or whole solar systems as illustrated in the subsequent sections [3].

2.2 Solar Photovoltaic Characteristics and Models

Generally, a photovoltaic cell consists of a p-n junction that is exposed to light, which results in the generation of charge carriers that can cause a current flow when the cell is connected to a load [4]. The available current flow is proportional to the incident light and depends on the semiconductor characteristics. In the scope of this thesis, it is sufficient to consider the resulting characteristics of a solar cell without the need to understand the underlying
physical principles in detail. The commonly used single diode equivalent circuit model for a PV cell is shown in Fig. 2.1 [5]. More complicated models using two diodes are described in [6] and [7]. For details about possible simplifications regarding the parasitic resistances, see [5]. Some models, such as the one shown in [8], also account for operating the cell in reverse bias, which is not relevant for the design of MPPT systems and will therefore not be considered here.

From the single diode model, the expression for the cell’s output current can easily be derived as

\[ I = I_{pv} - I_s \left[ e^{\frac{V + R_s I}{N k T}} - 1 \right] - \frac{V + R_s I}{R_{sh}}, \]  \hspace{1cm} (2.1)

where \( I_{pv} \) is the PV current, \( I_s \) is the saturation current and \( n \) is the diode ideality factor of the diode. The thermal voltage \( V_t \) is given by

\[ V_t = \frac{N k T}{q} \]  \hspace{1cm} (2.2)

with \( N \) representing the number of cells connected in series, \( k \) being the Boltzmann constant \( (1.3806503 \times 10^{23} \text{ J/K}) \), and \( q \) denoting the charge of an electron \( (1.60217646 \times 10^{19} \text{ C}) \). The temperature \( T \) needs to be inserted in Kelvin and can be calculated as follows:

\[ [K] = [C] + 273.15 \]
\[ [K] = ([F] + 459.67) \times 5/9. \]  \hspace{1cm} (2.3)

The resulting current over voltage curves of a PV cell [9] are displayed in Fig. 2.2 for different insolation levels and in Fig. 2.3 for different temperatures. Figures 2.4 and 2.5 show the corresponding power over voltage curves.
It can be seen that in each case there exists a unique MPP that is dependent on insolation and temperature. An exposition of the cell to full sunlight is usually considered to be equal to an insolation level of 1 kW/m$^2$.

A regular solar panel consists of 36 to 72 single PV cells connected in series (see Fig. 2.6). The series connection increases the voltage and makes it easier to interface the string of panels with the inverter in a grid tied PV system. In a series connection, each cell is able to produce its maximum power output under collective MPPT, as long as all cells are operating under perfectly matched conditions. However, this is usually not the case, as not only the temperature and irradiation across a panel may vary (e.g. due to partial shading), but manufacturing tolerances and aging may cause the cells to operate at different output currents. Because of the series connection, the overall output current of the string is limited by the weakest cell. This yields significant power losses under mismatched conditions. Partial shading causes the output current of the shaded cells to be significantly lower than the current produced by the surrounding cells. The cells with the lowered current get reverse biased by surrounding cells. This causes them to act as loads and to dissipate power, which leads to heat dissipation and eventual damage of
Figure 2.3: Output current characteristics of a PV cell at different temperatures and constant irradiation level.

Figure 2.4: Output power of a PV cell at different irradiation levels and constant temperature.

To prevent cell damage, bypass diodes are installed in solar panels for the cells.
strings of 12 - 24 cells. This divides the solar panel into 3 (e.g. Solarworld Sunmodule SW 235 poly [10]) or 6 (e.g. BP Solar BP7185N [11]) sub-modules. In the case of shading of one or more cells, the whole sub-module is bypassed by the conducting diode as shown in Fig. 2.7. Consequently, a single shaded cell can cause all cells that are connected to the same bypass diode to be taken out of the string. This results in significant reduction of power yield, as the whole sub-module does not contribute to the power generation any more.
Figure 2.6: Symbol and schematic drawing of the internal connection of a PV panel.

Figure 2.7: PV panel with one bypassed sub-module due to partial shading.
2.3 System Topologies

Theoretically, the maximum power output of a system consisting of many PV cells connected in series can be achieved when tracking the MPP of each cell separately [12], [13]. However, in practice the employed MPP trackers introduce losses into the system. Moreover, it is difficult to implement efficient MPP tracking circuits at a single cell level due to the low voltage. Therefore, MPP tracking is usually performed for several cells together.

A central inverter topology as shown in Fig. 2.8 is a common architecture for residential and commercial PV installations. In this architecture, a single MPP tracker that is built into the central inverter determines the MPP for all cells that are connected in series and parallel. Advantages of this topology are the comparably high degree of efficiency that can be achieved by using an inverter with high power rating and the low hardware cost. Nevertheless, as discussed above, operating all cells at the same MPP may result in significant loss in power yield.

The architecture shown in Fig. 2.9 (typically called string-level inverter) makes it possible to operate each string of panels in the installation at different MPPs. This allows for better adaptation to mismatch within the PV system. Nevertheless, the employed inverters usually have lower efficiency than a central inverter due to the lower power rating. Moreover, the system cost is increased because a higher number of inverters is required.

Taking the concept of splitting up MPP tracking to smaller units a step further, yields a topology such as the one shown in Fig 2.10 known as the micro-inverter architecture. Every solar panel is connected to its own inverter that also functions as a MPP tracker. These micro-inverters implement the interfacing of the PV panel with the grid for each panel individually and therefore have to operate at large voltage conversion ratios and comparably low power ratings. Moreover, the conversion from dc to ac has to be performed for every panel, reducing the conversion efficiency of the inverters. A high number of inverters is required, which increases system cost.

Even with the micro-inverter topology, the MPP is tracked for 36 to 72
cells (dependent on the panel size) together. In case of partial shading that yields a mismatch between cells within a panel, the bypass diodes of the corresponding sub-module will still conduct and thereby reduce the output power of the panel in intervals of 1/3 (3 sub-modules) or 1/6 (6 sub-modules).

To inhibit this behavior, a topology that uses MPP tracking on a sub-module level is presented in the next section. Through connecting a dc-dc converter to each sub-module, it is possible to further reduce the number of cells, for which MPP tracking is performed collectively. This allows operation of the panels without the bypass diodes and increases overall power yield.

Figure 2.8: Central inverter for the entire PV system.
Figure 2.9: Separate inverter for each string in the PV system.

Figure 2.10: One inverter per PV panel (micro-inverter).
2.4 Sub-Module Distributed Maximum Power Point Tracking

To overcome the disadvantages of the micro-inverter topology as described above, a solution using dc-dc converters is proposed here. The tree sub-modules of a panel are each connected to a separate converter that performs MPP tracking. In the case of partial shading, the sub-module that includes the shaded cell(s) is operated at its separate lower MPP without reducing the performance of the other sub-modules. The topology of the system is shown in Fig. 2.11. The converters are all connected in series at the output to achieve a high output voltage and consequently a lower voltage conversion ratio for the inverter of the PV system. Rather than just performing MPP tracking centrally through the inverter, distributed maximum power point tracking (DMPPT) is performed by dc-dc optimizers.
Figure 2.11: Distributed MPPT on the sub-module level.
CHAPTER 3

CIRCUIT DESIGN FOR SUB-MODULE MPPT

This chapter describes the circuit design process to implement the distributed MPPT system based on dc-dc converters as described in Chapter 2. At first, the circuit requirements, which underlie the subsequent converter topology selection, are outlined. Moreover, the possibilities of interconnection between converters is studied and application specific circuits are described.

3.1 Requirements

The central task of the dc-dc converter connected to the sub-module is to perform MPPT. The converter input current must therefore be adjustable over the whole range of possible MPP currents. Because a high output voltage of the PV system is desired, the outputs of the converters are preferably connected in series. The average output current of all converters is equal then, which requires the selected converter topology to be suitable for these load conditions. Other requirements concern the practical implementation of the converter. The converter needs to be sufficiently efficient to yield an increase in energy yield from the PV system. If the losses introduced by the additional power conversion at the sub-module level are higher than the amount of energy that can be captured in excess through using the converters, their usage is not justified. The given application is cost sensitive, requiring small size and low prize at a high efficiency. Regarding the implementation, this translates to low component count, low values for passive components as well as low voltage stresses for the employed switching devices.
3.2 Converter Topology Selection

As shown in [14], [15], the synchronous buck converter topology illustrated in Fig. 3.1 is suitable for the given application. It allows the use of only two switching devices that are rated for the maximum value of $V_{\text{in}}$, which is identical to the open circuit voltage of a sub-module. This voltage rating is lower than for boost-type converters, where $V_{\text{out}} > V_{\text{in}}$ and the devices have to be rated for $V_{\text{out}}$. Therefore, the buck topology offers high power density (small size) through operating at high switching frequency, and high efficiency. Its low component count allow for a discrete implementation at low cost.

The buck topology can only step down the input voltage, as its output voltage is given by

$$V_{\text{out}} = D \cdot V_{\text{in}}$$  \hspace{1cm} (3.1)

for an ideal converter, where $D$ denotes the duty ratio. At the same time, the input current is stepped up by

$$I_{\text{out}} = \frac{I_{\text{in}}}{D}.$$  \hspace{1cm} (3.2)

Therefore, MPPT is performed by stepping down the voltage of each sub-module. The output current is identical for all series connected converters and is determined by the global MPPT algorithm run by the central inverter (see control section in Chapter 4). The duty ratio of every converter is varied until $I_{\text{in}} = I_{\text{mpp}}$ at each sub-module. This yields a unique duty ratio for every converter that also determines its output voltage as shown in Eq (3.1). The output voltage will be lower for sub-modules with lower $I_{\text{mpp}}$ (e.g. due to partial shading), but each sub-module can contribute to the output power generation without being bypassed.

Even though it is not possible to achieve an increase of the string voltage,
this disadvantage is outweighed by the previously described benefits. Moreover, it is still possible to achieve a higher string voltage than without using dc-dc optimizers under most operating conditions. In the cases where one or more bypass diodes are conducting, the output voltage of the corresponding panel is significantly reduced. This is not the case when using the proposed MPPT topology. By employing a dc-dc buck converter as MPP tracker, each sub-modules can still contribute to the output voltage.

3.3 System Level Considerations

As most solar panels in the market are separated into three sub-modules, three converters are needed per panel. Building up these converters separately would cause redundancy in regard to the circuit control. Therefore, a system consisting of three dc-dc buck converters is presented here, where the control is performed by a single microcontroller. The basic circuit topology with the outputs connected in series is displayed in Fig. 3.2. The operation of the MOSFETS in each power stage is controlled centrally and the MPP tracking algorithm for the three sub-modules is run by the same microcontroller. The load is drawn across three converters here, but will be supplied by the whole PV system consisting of one or more strings of solar panels in a real application.

As mentioned before, it may be disadvantageous to use dc-dc converters for MPP tracking when there is no mismatch between sub-modules. To overcome this drawback, there are two possibilities when using a the buck topology. Firstly, the high-side MOSFET can be turned on permanently, while leaving off the low-side MOSFET. This way it is possible to bypass the circuit and avoid power loss due to modulation of the devices without the need for additional components. Providing a direct path between input and output through the inductor is another advantage of the buck topology. Nevertheless, the dc series resistance (DCR) of the inductor as well as the on-state resistance of the high-side switch will cause losses in the circuit even in bypass mode. To address this issue, it is possible to add a separate bypass MOSFET to each converter as shown in Fig. 3.3. While increasing component cost, this can improve power output under conditions where there is no mismatch. As the bypass MOSFET is not used for fast switching, a device
Figure 3.2: Three dc-dc buck converters used for sub-module level MPP tracking.

with significantly smaller on-state resistance than the high-side MOSEET can be chosen (see Chapter 4).
Figure 3.3: Three dc-dc buck converters with bypass MOSFETs.
3.4 Application Specific Circuits

3.4.1 Signal Voltage Level-Shifter

When using a circuit topology where the outputs of all converters are connected in series, it is necessary to shift the voltage level of the logic signals that are output by the microcontroller. Due to the series connection, the potentials of the common voltage levels of the middle and the upper converters have been shifted from ground to $V_{out1}$ and $V_{out1} + V_{out2}$ respectively. These are also the new logical ‘ground’ potentials for the gate drivers and other circuits that are controlled by the central microcontroller. Therefore, for a signal to be interpreted as a logical one, it is necessary to shift its potential from the 0 V to 5 V level to the levels of $V_{out1}$ to $V_{out1} + 5$ V (for the middle converter) or $V_{out1} + V_{out2}$ to $V_{out1} + V_{out2} + 5$ V (for the upper converter).

A circuit as shown in Fig. 3.4 can be used to perform this task. The signal is generated by the microcontroller at its standard logic signal voltage level with respect to ground. The node $V_{shift}$ is tied to the corresponding voltage level by which the signal should be shifted. When the logic signal generated by the microcontroller is zero, the capacitor charges to the voltage level of $V_{shift} - V_D$. $V_D$ denotes the forward voltage drop of the diode, which can be neglected in the following considerations. By now setting the logic signal level to 5 V, the lower node of the capacitor is tied to 5 V and the voltage across the capacitor remains constant. Consequently, the voltage at the upper node of the capacitor, which is also the node of the shifted signal, is on the level of $V_{shift} + 5$ V. By applying a logic signal to the lower node of the charged capacitor, the waveform of the signal is replicated with shifted voltage levels at the ‘shifted signal’ node.

The resistor in parallel to the diode has two functions and needs to be dimensioned accordingly. On the one hand, it acts as a fail-safe option when the logic signal is set to high for a prolonged period of time; on the other hand, it implements the function of a pull-down resistor.

When the signal at the lower node of the capacitor stays on its high level for a long time (low frequency of the logic input signal), the capacitor starts discharging through the resistor. The time it takes for the capacitor to discharge is determined by the time constant.
\[ \tau = RC \]  

(3.3)

and the voltage at the ‘shifted signal’ node shows an exponential decay

\[ v(t) = v_0 e^{-\frac{t}{\tau}}, \]  

(3.4)

where \( v_0 \) is the initial voltage \((V_{shift} + 5 \text{ V})\). This effectively resets the logic input signals for the gate drivers and other circuits and prevents malfunction due to controller failures. The time constant \( \tau \) needs to be chosen according to the frequency of the signal that needs to be shifted. Choosing the time constant as too long, results in a slow decay of the signal in the failure case. Conversely, a time constant that is too short will result in the appearance of discharge effects in the level shifted signal.

The losses in the circuit can be reduced by choosing a large value for \( R \). However, it needs to be considered that even when the signal is at the level representing a logical zero, the only direct resistive path between the logical circuit connected to the ‘shifted signal’ node and its logical ground potential \((V_{shift})\) is the resistor \( R \). Therefore, this resistor also needs to implement the functionality of a pull-down resistor that ties the logical circuit input to a defined potential at all times. Generally, this prohibits the usage of resistor values in the mega-ohm region, depending on the circuit. When choosing a resistor value in the kilo-ohm region, to implement the pull-down functionality, the capacitor value has to be increased accordingly to avoid a time constant that is too short. Figure 3.5 shows an oscilloscope screenshot of a gate driver signal with a frequency of \( f_{sw} = 100 \text{ kHz} \), shifted by \( V_{shift} = 12 \text{ V} \). The circuit shown in Fig. 3.4 has been implemented using a standard 1N4148 diode and values of \( R = 10 \text{ k\Omega} \) and \( C = 1 \mu\text{F} \). The voltage across the resistor equals the diode forward voltage drop when the diode is conducting (input signal low). For the case where the input signal is at its high state, the voltage drop across the resistor is equal to 5 V. The losses can be estimated by using

\[ P_{loss} = \frac{V^2}{R}. \]  

(3.5)

For the given component values, this results in a loss of 49 \( \mu\text{W} \) in the off-
state (assuming a diode forward voltage drop of 0.7 V) and 2.5 mW in the on-state. These values are very small in comparison to the overall processed power of the circuit and can therefore be neglected.

Figure 3.4: Circuit to perform the level shifting of logic signals.
3.4.2 Bypass MOSFET Gate Driver

To reduce losses when there is no mismatch the converters are bypassed in this case. When all three sub-modules of a panel have the same MPPT, it is not necessary to process the produced power through the distributed power electronic circuitry in order to achieve maximum energy yield. In fact, the processing would cause a lower output power than in the case where there is no intermediate circuitry, due to the degree of efficiency of the converters. Therefore, a MOSFET is used to bypass the whole circuit as shown in Fig. 3.3. Integrated solutions for this purpose exist [16], [17], [18], [19] but are relatively expensive in comparison to a standard MOSFET with similar on-state properties. Therefore, a MOSFET with discrete gate driving circuit is used here.

As the MOSFET is switched occasionally only (very low switching frequency), the gate driving losses can be neglected. Therefore, a device with very low on-state resistance can be chosen. To drive the MOSFET into its on-state, a positive voltage $V_{GS}$, which is higher than the threshold voltage, is required (N-channel FET). Similar to the level-shifting circuit described above, the first step is to generate a clock signal that oscillates between the voltage at the source node of the bypass MOSFET (low) and $V_S + 5$ V (high), when neglecting the diode forward voltage drop. This clock signal can be observed at point A of the circuit shown in Fig. 3.6. The considerations
regarding suitable component values are the same as for the level-shifting circuit. The signal gets rectified by the diode D\textsubscript{2} and charges the capacitor C\textsubscript{small} (the diode prevents charge from flowing back when the signal at point A is in its low state). The corresponding signal can be measured at point B of the circuit. C\textsubscript{small} is chosen to have a small capacity value, as the charge stored in it has to be removed quickly when the bypass MOSFET is switched off. Switching the device very slowly would yield increased losses during the transitioning process. R\textsubscript{big} has the same function as in the level-shifting circuit. In case the clock signal is removed, the capacitor C\textsubscript{small} discharges through the R\textsubscript{big} resistor.

The remaining elements as shown in Fig. 3.6 constitute the discharging circuit for the bypass MOSFET. To initiate the turn-off process, the clock signal is removed from the circuit. The MOSFET Q (general purpose device) is turned on for a short time by a discharge pulse. The pull-down resistor R ensures that V\textsubscript{GS} of the MOSFET Q goes back to zero after the pulse has been applied and stays at a defined potential at all times. The charge stored in C\textsubscript{small} and the gate capacitance of the bypass MOSFET are removed through the resistor R\textsubscript{small} to cause V\textsubscript{GS} of the bypass MOSFET to fall below the threshold voltage. Consequently, the MOSFET is turned off. To turn it back on, the clock signal needs to be applied to the circuit again.

The previously described signals are shown in Fig. 3.7 for the case where the bypass MOSFET is turned on. Figure 3.8 shows the turn-off process with the applied discharge pulse. The frequency of the clock signal was chosen to be 100 kHz and the input voltage of the circuit is 12 V.

Figure 3.6: Bypass MOSFET with gate driving circuit.
3.5 Converter System Circuit

Figures 3.9 to 3.12 show the complete circuit of the proposed system, including the bypass MOSFET drivers as well as level shifting and sensing components. The circuit has been divided into functional blocks that are displayed separately. Besides the individual converters, the central microcontroller with
the necessary input and output signals is shown. Level shifting is only necessary for the second and third converter, as the microcontroller is powered by the input voltage of the first converter through a linear voltage regulator. For each converter, the input voltage, as well as the voltages at both terminals of the inductor, are measured using the analog-digital converters (ADC) of the microcontroller. In a buck-converter one of the inductor terminals is at the output voltage level. Therefore, an output voltage measurement is available that is required for the MPPT algorithm described in Chapter 4. Sensing the voltage drop across the inductor allows for calculation of the average output current when the dc resistance (DCR) of the inductor is known. This makes it possible to perform current sensing without introducing additional losses (through a shunt resistor or additional hardware) into the circuit.
Figure 3.10: Schematic drawing of converter 1 including the bypass circuit and components for sensing.
Figure 3.11: Schematic drawing of converter 2 including the bypass circuit, level shifting, and components for sensing.
Figure 3.12: Schematic drawing of converter 3 including the bypass circuit, level shifting, and components for sensing.
CHAPTER 4

IMPLEMENTATION OF SUB-MODULE MPPT

This chapter describes the implementation of the proposed system. First, the process of component selection and placement on a printed circuit board (PCB) is outlined. Second, the necessary control algorithms are presented.

4.1 Discrete Hardware Implementation

The circuit shown in Fig. 3.1 was implemented using discrete components. This allows for high flexibility during the device selection and good matching with the determined requirements.

4.1.1 Loss Calculations

The loss calculations are outlined here for the sake of giving an orientation during the component selection process. They are not meant to be precise measures of the actual losses occurring in the circuit, but rather as estimates. Generally, it is hard to accurately determine the circuit losses beforehand, as gate-driving losses, control losses and losses in the PCB are not only dependent on the operating point of the system but also on the components available on the market. In practice, it is necessary to iterate and test different components. Alternatively, complex multiphysics simulations can be carried out. Nevertheless, the accuracy of these simulations is limited by the available models of the electronic components, the actual PCB layout and other factors [20], [21], [22]. These models are rarely fully provided by component manufacturers, such that it is difficult to achieve exact results.

For this project, simple loss calculations [23], that are considered a guidance during component selection, have been carried out. The target converter
efficiency was set to 99%. The allowed loss of 1% was split among the different loss types and components.

The inductor is usually the component of a switching converter that limits size and efficiency [24], [15]. Therefore, the inductor loss, mostly consisting of core losses and winding losses, was allowed to account for 0.5% of the overall losses. The remaining 0.5% were evenly split between losses in the power MOSFETs and other losses. The loss in the switching devices is mostly caused by their on-state and off-state (usually negligible) resistances and by the transient operation. The latter is due to the gate capacitance of each switch that needs to be charged and discharged periodically, which yields a loss in energy. The switching process itself causes energy dissipation, as voltage and current may have values of significant magnitude at the same time during the transition. The contribution of these loss-types to the overall loss is hard to estimate in practice, as they depend on the specific MOSFET and gate-driver combination and the PCB layout. The other losses, for which an allowance of 0.25% of the overall power was given, account for gate-driving losses, PCB losses, losses introduced through control and sensing, as well as capacitor losses.

For the calculation, a typical power rating of a solar panel of 250 W was considered. Consequently, each converter has to process a maximum power of 83.3 W corresponding to the output power of one sub-module. To account for shaded conditions, which will result in a decrease in output power of the sub-module, a power value of 75 W has been chosen as a basis for the subsequent calculations. This corresponds to about 90% of the nominal output power value. Generally, we expect relatively high duty ratios during the MPPT operation of the converters [15]. This is important to consider when balancing the on-state losses of the high-side and the low-side MOSFETs. In this thesis, a duty ratio of $D = 0.9$ has been used for the loss calculations.

For a solar panel with nominal output voltage of 36 V, the operating voltage at its MPP is usually around 30 V [10]. As illustrated in Chapter 2, this voltage does not experience major changes under moderate variation in irradiation, and is therefore assumed to be constant for the further course of the loss calculations. This yields a sub-module voltage of 10 V, which
corresponds to a converter input current of $I_{in} = 7.5$ A at 75 W.

![Figure 4.1: Triangular waveform corresponding to the inductor current in a buck converter.](image)

### 4.1.2 Inductor

The inductor is allowed to account for a power loss of $75 \text{ W} \times 0.5\% = 375 \text{ mW}$. Its losses can be estimated by adding up the different types of losses in the inductor as follows (see [25]):

$$P_{\text{loss, inductor}} = P_{\text{core}} + P_{\text{dcr}} + P_{\text{acr}}. \quad (4.1)$$

The core losses, $P_{\text{core}}$, are usually given by the inductor manufacturer as measured or calculated values. For ferrite cores, the loss can be calculated as

$$P_{\text{core}} = K \cdot f^x \cdot B^y \cdot V_e, \quad (4.2)$$

where $K$ is the material constant of the core (usually given by the manufacturer), $f$ is the frequency, $x$ is the frequency exponent, $B$ is the peak flux density, $y$ is the flux density exponent, and $V_e$ is the effective core volume. The loss is therefore influenced by the core material and geometry ($K, x, y, V_e$) as well as by the converter operating parameters ($f, B$).

The DCR of an inductor is caused by the resistance of the wire that it is wound from. The resulting ohmic losses can simply be calculated as

$$P_{\text{dcr}} = I_{\text{rms}}^2 \cdot DCR. \quad (4.3)$$
The rms value of a triangular waveform such as the one shown in Fig. 4.1 is given by

\[ I_{\text{rms}} = \sqrt{I_{\text{avg}}^2 + \frac{\Delta i^2}{12}}, \quad (4.4) \]

where \( I_{\text{avg}} \) denotes the average current value and \( \Delta i \) stands for the peak-to-peak current ripple. Assuming that the designed dc-dc buck converter operates in continuous conduction mode (CCM), its inductor current waveform will be a triangular waveform with \( I_{\text{avg}} = I_{\text{out}} \). The current ripple can be calculated by using

\[ \Delta i = \frac{D \cdot (1 - D) \cdot V_{\text{in}} \cdot f \cdot L}{I_{\text{avg}}}, \quad (4.5) \]

where \( L \) is the inductance value.

A further type of loss is caused by the ac resistance (ACR) of the winding. The loss is ohmic and can be expressed as

\[ P_{\text{acr}} = \bar{I}_{\text{rms}}^2 \cdot ACR. \quad (4.6) \]

\( \bar{I}_{\text{rms}} \) is the rms value of the ripple component of the current, given by

\[ \bar{I}_{\text{rms}} = \frac{\Delta i}{2\sqrt{3}}. \quad (4.7) \]

For the intended design of the converter in CCM, the ripple is relatively small compared to the output current. Therefore, the loss in the DCR accounts for the majority of the overall inductor loss. To obtain a quick estimate of the desired DCR value, the current ripple as well as \( P_{\text{core}} \) and \( P_{\text{acr}} \) are neglected here. Calculations using tools provided by the inductor manufacturer have shown that the introduced error is not significant. The simplifications immediately yield

\[ DCR = \frac{P_{\text{dcr}}}{\bar{I}_{\text{rms}}^2} = \frac{375 \text{ mW}}{(7.5 \text{ A})^2} = 6.67 \text{ m}\Omega. \quad (4.8) \]
Therefore, an inductor with a DCR lower than the calculated value must be chosen. Furthermore, its inductance value must be large enough to justify the assumptions of operation in CCM and small ripple.

### 4.1.3 Switching MOSFETs

Switching device losses can be calculated as the sum of losses caused by conduction (on-state), switching, and leakage (off-state).

\[
P_{\text{loss,switch}} = P_c + P_{\text{sw}} + P_l
\]

(4.9)

Due to their small contribution, leakage losses are usually neglected.

The conduction loss in both switching devices is allowed to account for 0.25% of the losses, corresponding to 75 W \times 0.25\% = 188 mW. This loss should be balanced evenly between the high-side and the low-side MOSFET to avoid excessive local power dissipation in one of the devices \(P_{\text{loss,high}} = P_{\text{loss,low}}\). Therefore, each MOSFET is allowed to dissipate 94 mW of power. The distribution of power loss between the devices is generally dependent on the operating conditions. Here, the previously defined point with \(D = 0.9\) is used for the calculations. Consequently, the high-side MOSFET needs to be chosen with a much lower on-state resistance \(R_{\text{DS,on}}\) than the low-side MOSFET, as it is on for 90\% of the time. Using the assumption of operation in CCM with small ripple, the loss of the device in the on-state can be calculated as

\[
P_{c,\text{high}} = I_{\text{out}}^2 \cdot R_{\text{DS,on,high}} \cdot D
\]

(4.10)

for the high-side MOSFET and as

\[
P_{c,\text{low}} = I_{\text{out}}^2 \cdot R_{\text{DS,on,low}} \cdot (1 - D)
\]

(4.11)

for the low-side MOSFET. Rearranging and inserting the derived values
yields

\[
R_{DS,\text{on, high}} = \frac{P_{c,\text{high}}}{I_{\text{out}}^2 \cdot D} = \frac{94 \text{ mW}}{(7.5 \text{ A})^2 \cdot 0.9} = 1.86 \text{ mΩ}
\]

\[
R_{DS,\text{on, low}} = \frac{P_{c,\text{low}}}{I_{\text{out}}^2 \cdot (1 - D)} = \frac{94 \text{ mW}}{(7.5 \text{ A})^2 \cdot (1 - 0.9)} = 16.71 \text{ mΩ}.
\]

The actually occurring switching losses are not only dependent on the used switching devices but also on the gate driver for those devices. They are influenced by the gate-source capacitance \(C_{GS}\) of the MOSFET as well as other parasitic capacitances as well as the source and sink-current of the gate driver. The driver itself needs to supply its internal logic, which will cause additional energy consumption. Because these losses are dependent on the actual device, they are accounted for in the next section here. General considerations and formulas to quantify switching losses are given in [26] and [27].

### 4.1.4 Other Losses

Losses imposed by switching, stray resistive and capacitive elements in the PCB as well as control and sensing are accounted for in this category. The latter are caused by the microcontroller and depend on the mode of operation. The control loss will be higher under high computational load and can be reduced by employing sleep modes of the controller. Generally, it is hard to quantify these losses exactly, such that the given margin of 188 mW has to be a sufficient estimate here.

### 4.1.5 Component Selection

In the following, the process of practical component selection is outlined. Requirements that go beyond the basic orientation given above are stated. The overall goal was to achieve a low cost design with small size such that the converters can fit into the junction box of a regular solar panel. This eliminates the need for a custom enclosure and thereby greatly helps to reduce overall cost and to increase flexibility. The bill of materials and details
about the implementation can be found in Appendix A.

4.1.6 Inductor

Along with a sufficiently low DCR, the inductor must have a saturation current that is higher than the maximum expected current to avoid additional losses and overheating due to saturation of the core material. Additionally, the inductor is the largest component of the circuit and thus has a major impact on overall converter size. Generally, choosing an inductor is about finding the right compromise between DCR, core losses and size that is best suitable for the application at hand.

4.1.7 Switching Devices

Choosing suitable switching devices is mainly about finding the right trade-off between low on-state resistance and low gate charge $Q_g$. The latter is directly related to the value of $C_{GS}$ and needs to be supplied and removed by the gate driver throughout each switching cycle. MOSFETs with low on-state resistance usually have a higher gate charge, as the gate area is larger than for MOSFETs with higher $R_{DS,on}$. A spreadsheet was used to compare available devices from different manufacturers on the market. Figures 4.2 and 4.3 show the gate charge for different on-state resistances in the desired regions for the high-side and the low-side MOSFET, respectively.

4.1.8 Gate Driver

For the high-side and the low-side, two gate drivers that can be integrated into the same package are required. A small package size was chosen to reduce the size of the circuit. Priority was given to the capability of driving a high sourcing and sinking current. This allows for fast switching operation with short transition times and reduces related losses caused by cross conduction.
Figure 4.2: Gate charge vs. on-state resistance for a selection of suitable high-side MOSFETs.

Figure 4.3: Gate charge vs. on-state resistance for a selection of suitable low-side MOSFETs.
The logic inputs of the integrated circuit (IC) have to be compatible to the microcontroller logic levels. For the purpose of high flexibility in regard to control (see Chapter 7), a gate driver with separate inputs for driving the high and low-side has been chosen.

Furthermore, the supply voltage range of the IC was an important selection criterion, as supplying the IC through an external voltage regulator would introduce additional losses and increase size and cost of the circuit.

### 4.1.9 Input and Output Capacitors

To reduce the equivalent series resistance (ESR) losses in the capacitors, the input and output capacitance, that is necessary for the converter, is divided up and implemented by multiple capacitors in parallel. The capacitors have to be rated for the highest occurring input voltage and should impose little losses in the dielectric. Multilayer ceramic capacitors with X5R dielectric have been chosen. For the given implementation, the capacitor losses are small in comparison to inductor and semiconductor losses.

### 4.1.10 Microcontroller

The microcontroller (MC) has to offer the functionality to control all three power stages centrally. By choosing a gate-driver with separate inputs for the high and the low-side switch, six outputs that are capable of producing a high frequency pulse width modulation (PWM) signal are required. Moreover, enough analog to digital converters (ADCs) have to be available to perform the required sensing of the input voltage (see Section 4.2). To leave enough computational resources to perform MPPT and other control tasks, the control of the power stages should preferably be done through hardware-based logic integrated into the MC. This enables operation at a comparably low overall system clock and reduces power consumption, which was another important selection criteria.
For the given implementation the AT90PWM316 [28] from Atmel has been chosen. It offers three hardware power stage controllers (PSCs) with two output pins each. This allows one to control the high and low-side switch of each converter separately and to adjust the dead-time with high flexibility. The MC is based on an 8-bit core operating at a maximum system clock of 8 MHz when driven by the internal oscillator. A high speed clock with up to 64 MHz allows for precise timing of the PSC outputs. The power consumption of the controller can be reduced by entering different sleep-modes dependent on the performed computational task.

### 4.1.11 Printed Circuit Board Layout

All three symmetric dc-dc buck converters, the central microcontroller and the bypass circuits are implemented on the same PCB. Additionally, connectors for power and signal measurement points are located around the circuit for easy interfacing. The converters have been optimized for size, such that they can easily fit into the junction box of a solar panel.

The PCB for the prototype is a four-layer board which was designed using EAGLE CAD. While the two outer layers are used to hold the components, the inner layers route signals from the microcontroller to the different converters and signals within the converter stages. The different layers are shown in detail in Appendix A. Figure 4.4 shows an annotated photograph of the PCB top-side with components. Figure 4.5 shows the bottom view of the PCB and Table 4.1 lists the converter specifications.
Figure 4.4: Annotated top view of the PCB with components.

Figure 4.5: Annotated bottom view of the PCB with components.
Table 4.1: Converter Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage Range</td>
<td>7-16 V</td>
</tr>
<tr>
<td>Output Voltage Range</td>
<td>0.8-14 V</td>
</tr>
<tr>
<td>Max. Output Power per Converter</td>
<td>100 W</td>
</tr>
<tr>
<td>Switching Frequency</td>
<td>100 kHz</td>
</tr>
<tr>
<td>Converter Efficiency</td>
<td>95%</td>
</tr>
<tr>
<td>Board Area per Converter</td>
<td>21 mm x 27 mm</td>
</tr>
<tr>
<td>Overall Board Area</td>
<td>63 mm x 27 mm</td>
</tr>
</tbody>
</table>

Particular attention during the layout process of power electronic circuits needs to be given to the power stage. Generally, all loops conducting high currents at high transient frequencies should be kept as tight as possible. This is to avoid stray inductances that may cause the induction of high voltage peaks and ringing [29], [30], [31], [32], [33]. These voltage peaks increase device stresses and may even lead to destruction of the power semiconductors. Moreover, they decrease the converter efficiency. While designing the PCB board for a buck converter, it is important to keep the input loop as tight as possible. The output loop is less critical, as the addition of stray inductance does not have severe effects due to the power inductor in this loop. Figure 4.6 shows the input loop as well as its implementation on the PCB board. The high and low-side MOSEFETs have been placed next to each other with the input capacitors in between. This assures that the loop marked as 1 is as tight as possible.

Another loop that conducts high transient currents is the loop between the gate-driver output and the gate of the MOSFET. To avoid voltage peaks and increase efficiency, the gate driver has been placed on the bottom side of the PCB right under the power stage MOSFETs. Multiple vias route the high gate sourcing and sinking currents (up to 4.5 A for the chosen gate driver) to the gates as shown in Fig. 4.7. This assures that the loops marked as 2 are as tight as possible during sourcing and sinking of the gate current.

The numbers give an orientation of the priority in which the loops should be designed. As the current through the power stage is generally higher than the gate-driving currents, this loop should be designed first. When the optimal layout is found, the gate-driver can be added, followed by the relatively
Figure 4.6: Input loop of the dc-dc buck converter and its implementation on the PCB.

uncritical loop consisting of the power inductor and the output capacitors.
Figure 4.7: Loop between the gate driver and the MOSFET gate and its implementation on the PCB.
4.2 MPPT Control

MPPT on the sub-module level is implemented using the common perturb and observe (P&O) algorithm as described in [15], [34]. Other MPPT algorithms exist [35] and may have advantages in regard to speed and tracking efficiency over a standard P&O algorithm, but are more complicated in terms of their implementation. On the system level, the inverter performs a global MPPT for all connected panels, which determines the common string current. Tracking is usually slow [15] (in the range of minutes) in comparison to the local MPPT on the sub-module level. Therefore, the output current of each dc-dc converter can be considered to be constant. This allows for an implementation without local current sensing, as the maximum power can be extracted from a sub-module by maximizing its output voltage. The MPPT algorithm is implemented through sweeping the duty ratio from low to high values, while continuously measuring the input voltage of the converter. It is easy to detect the maximum of the input voltage, as there exists a single global maximum (output behavior of the sub-module is not influenced by bypass diodes). Once this maximum is reached, the duty ratio is altered such that the operating point of the converter coincides with the MPP of the sub-module at all times. The flow diagram of the algorithm is shown in Fig. 4.8 and consists of two different modes. On the right, the initial coarse sweep, that allows for fast estimation of the MPP region, is shown. The steady-state tracking process is illustrated on the left.

To achieve operation of all sub-modules at their MPPs, the string current has to be high enough, such that the sub-module with the highest power output is not limited in its operation. If the string current is lower than $I_{mpp}$ of a sub-module, it cannot be operated at its MPP because the buck-converter can only step up current. This power loss should be detected by the inverter, which will raise the string current accordingly. Operating at a current that is much higher than the $I_{mpp}$ of the strongest sub-module will cause increased losses due to parasitic resistances in the converter, resulting in a decrease in output power. We rely on the central inverter to be able to find the optimal string current without the need for communication with the local MPP trackers as described in [15].
\[ D = D_{\text{min}} \]
\[ V_{\text{max}} = 0 \]
\[ D_{\text{peak}} = D \]

Sample \( V_{\text{out}} \)

\[ V_{\text{out}} > V_{\text{max}} \]

\[ V_{\text{max}} = V_{\text{out}} \]  
\[ D_{\text{peak}} = D \]

\[ D = D + \Delta D_{\text{coarse}} \]

\[ D > D_{\text{max}} \]

\[ D = D_{\text{peak}} \]

\[ D = D_{\text{min}} \]
\[ V_{\text{max}} = 0 \]
\[ D_{\text{peak}} = D \]

\[ V_{\text{out}} > V_{\text{max}} \]

\[ V_{\text{max}} = V_{\text{out}} \]  
\[ D_{\text{peak}} = D \]

\[ D = D + \Delta D \]

Observe \( V[n] = V_{\text{out}} \)

\[ V[n] > V[n-1] \]

\[ \Delta D = -\Delta D \]

\[ \text{Change Direction} \]

\[ \text{Perturb} \]

\[ D = D_{\text{peak}} \]

\[ \text{Steady State MPPT using Perturb & Observe} \]

---

Figure 4.8: Flowchart of the employed MPPT algorithm.
For some inverters (e.g. [36]) it is possible to set a fixed input current. This overcomes the issue of operating at a string current that is too low for the strongest sub-module. The current can be set high enough for all operating conditions, by choosing it according to a scenario with maximum irradiation and no shading. However, this may lead to increased losses at lower illumination as previously discussed. Nevertheless, these losses are usually much smaller than the power loss caused by operating away from the MPP.

4.3 Automatic Bypass Mode Control

In cases where there is no mismatch between sub-modules, the bypass MOSFETs should be activated automatically to avoid losses due to unnecessary converter operation. The control needs to detect these cases and is also required to determine when the local MPP tracking operation should be resumed after bypass-mode operation.

Within a solar panel, the central microcontroller senses the information for all three sub-modules. Therefore, it is easy to determine when all three converters are running at the same duty ratio. As $I_{\text{string}}$ is identical for all converters, the input current for all converters must be identical when operating at the same duty ratio. This input current coincides with $I_{\text{mpp}}$ of the sub-modules, which means that all sub-modules are operating at the same MPP and there is no mismatch.

In a per-panel inverter topology, this is a sufficient condition to determine when the bypass MOSFETs should be activated. In cases where a string of solar panels is connected to the same inverter, these conditions do not necessarily yield the desired results. There may be operating conditions, where no mismatch between the sub-modules of a panel but between the panels of a string are present. Consequently, all converters of one panel need to be operated at the same duty ratio without being bypassed to account for a different operating point of another panel. Under these conditions, a communication between the MPP trackers of different panels needs to be established to prevent operation in bypass mode.
Figure 4.9: Illustration of the change in sub-module output voltages at a constant string current under different irradiation levels (zoomed view around MPPs).

The condition for returning to the local MPP tracking can easily be detected by using the input voltage measurements of the central microcontroller within a panel. Figure 4.9 shows the voltages of three different sub-modules within a panel. In bypass mode, there was no mismatch between the sub-modules. As an example, a common irradiation level of 500 W/m² is assumed here. The string current has been adjusted accordingly to match the MPP (red line). As the irradiation conditions change across the panel, the microcontroller senses a difference in the input voltages, which triggers the local MPP operation. During the next global MPP sweep, the inverter is going to increase the string current, providing a sufficiently high current for optimal operation of the sub-module with increased radiation.

The flowchart of the automatic bypass mode control within a panel is shown in Fig. 4.10.
Figure 4.10: Flowchart of the employed per-panel bypass mode control algorithm.
5.1 Concept Description

Multiphase dc-dc converters are widely used in power electronics, as they enable the processing of high power by splitting the load-current into multiple phases. Conventional multiphase circuits are supplied by a common source, and the goal is to distribute the processed power evenly between the phases. An example circuit to step down the voltage and supply a high current load (e.g. a computer CPU) is shown in Fig. 5.1.

Current ripple cancellation is an important feature of multiphase switching converters as it enables each individual converter of the system to operate at a higher ripple than the overall current ripple through interleaving of the phases [37],[38],[39],[40]. This yields significantly lower values for the inductor and capacitors of each converter, and as shown in [41], can lead to significant reductions in converter size and cost while at the same time increasing the efficiency. Ripple cancellation in symmetric interleaved multiphase converters is easily achievable by phase shifting the targeted ripple waveform of each phase by an angle of \( \phi_n = (n - 1) \cdot \frac{360^\circ}{N} \), where \( N \) is the total number of phases and \( n \) is the considered phase [42],[43],[44]. This technique is well understood and described in the power electronics literature [45]. Figure 5.2 shows the three interleaved inductor current ripple waveforms for the buck converter topology displayed in Fig. 5.1 without dc part. All converters are operated at the same duty ratio of \( D = 0.7 \) and the phase currents are evenly shifted by \( 120^\circ \) to each other. The displayed waveform \( i_{\text{sum}}(t) \) is the sum of all three waveforms neglecting filter effects. It can be seen that its magnitude is significantly lower than the ripple magnitude in each phase. This yields a significant reduction of the overall output current ripple.
Figure 5.1: Multiphase dc-dc buck converter with three phases.

Figure 5.2: Ripple component of the current in each phase and summed ripple components for a three phase buck-converter operating at $D = 0.7$. 

50
If the duty ratio of all converters is set to $\frac{1}{N}$, it is possible to achieve a complete ripple cancellation as shown in Fig. 5.3. For converters that operate at a fixed duty ratio, the number of phases can be chosen such that optimal ripple cancellation is performed (e.g. choose 5 phases for a converter running at a $D = 1/5$).

The magnitude of the ripple in the time-domain is significantly reduced due to effective cancellation of certain ripple harmonics in the frequency-domain. The fundamental frequency of the ripple is determined by the switching frequency $f_{sw}$ of the converter. The number of phases $N$ corresponds to the cancellation of the harmonics; e.g. in a symmetrical two phase converter all odd harmonics ($k = 1, 3, 5, ...$) are canceled. Equivalently, for a symmetrical three phase converter the first and second harmonics and its multiples ($k = 1, 2, 4, 5, 7, 8, ...$) are canceled, leaving ripple frequency components at $k = 3, 6, 9, ...$ only. This is illustrated in Fig. 5.4 which displays an exemplary frequency content of the summed ripple components in a three phase interleaved converter at symmetrical operating conditions.

At the operating point where a complete ripple cancellation is possible
Figure 5.4: Harmonic content of the summed ripple components for a three phase buck-converter operating at $D = 0.7$.

$(D = 1/3)$, all harmonic components are canceled, only leaving a dc component.

5.2 Shortcomings in the Given and Other Application

In recent applications, such as MPPT for solar PV, multiple converters are supplied by sources that are often restricted to operate at different voltages and currents leading to asymmetric converter operation. To achieve improved ripple cancellation under these conditions, phase-shifting by uneven phase-angles is required. In Chapter 6 analytic formulas are derived to obtain suitable angles for phase-shifting.

Asymmetric phase-shifting has been used to account for imbalances in the converter phases due to component tolerances [46]. This was done for standard single-input single-output converter topologies such as shown in Fig. 5.1. However, phase-shifting for non-uniform duty ratios and different
input sources for each phase is not well understood. In the sub-module MPPT system presented in this thesis, all the dc-dc converters are connected in series at their outputs, supplying one common load. Therefore, in this application a multiple-input single-output topology is required. Figure 5.5 shows the familiar topology with individual sources and a series connection at the output, while Fig. 5.6 shows an example for multiple inputs with the same ground potential and all outputs connected in parallel.

In these cases, the goal is usually to minimize the output current ripple of the circuit. We have presented a method for output current ripple reduction in the distributed MPPT application in [47]. For the topology with the outputs connected in parallel identical results can be obtained.

Moreover, there exists a number of applications such as distributed power architectures (DPA), heterogeneous low voltage CMOS supplies and multilevel power supplies where converters, that are connected to a common input, are forced to operate under asymmetric conditions. In many implementations, including power conditioning systems (PCS) [48] and dc-bus power distribution [49], [50], a low input current ripple of the employed asymmet-
Figure 5.6: Three dc-dc buck converters in a multiple-input single-output (parallel) topology.

The schematic drawing of the corresponding multiphase buck converter is shown in Fig. 5.8.

The converters are connected in parallel at the input and are supplied by the same source (single-input). Yet, their outputs supply individual loads at different voltage levels (multiple-output), causing the duty cycles of the converters to be different. The efficiency of the system can be increased by using smaller input filters and less capacitance on the bus, which also improves its reliability. Therefore, it is desirable to minimize the ripple that the converters induce on the dc-bus. Significant efforts are made to maintain a constant bus voltage level (e.g. by dynamic control [49]). Chapter 6 introduces a mathematical framework and a derived control technique to minimize the input or output current ripple of asymmetric interleaved multiphase converters. The derivations are independent of the converter topology and can be universally applied.
Figure 5.7: Typical telecom systems DPA application with converters operating at non-uniform duty ratios.

Figure 5.8: Multiphase dc-dc buck converter with different output voltages generated from a single dc-bus.
5.3 Motivation of a Frequency Domain Based Ripple Cancellation Technique

To allow for effective ripple cancellation under asymmetric operating conditions, a frequency domain description of the current ripple waveform is used here. This allows one to actively influence the frequency content of the output ripple and is therefore preferable over a time-domain analysis. Even though the direct relation between the number of phases $N$ and the order of the canceled harmonics, as described above, does not hold for asymmetric converter operation, the ripple can still be significantly reduced by going beyond the conventional even phase-shifting technique. Being able to effectively move the frequency content of the ripple to higher harmonics relaxes the requirements for the filters used in switching converters.

Figure 5.9 shows the schematic drawing of an L-C low-pass filter as commonly used as a filter for switching converters. Its transfer function is given by:

$$\left| \frac{V_o}{V_i} \right| = \frac{1}{\omega LC + \omega L/R + 1},$$

(5.1)

where $R$ is an added damping resistor to prevent the resonant peak in the transfer function. In a dc-dc buck converter topology this resistor is not required, as the load resistor accounts for damping.

Figure 5.10 shows the Bode plot of the corresponding transfer function for component values of $R = 5 \, \Omega$, $L = 10 \, \mu H$ and $C = 1 \, \mu F$. It can be seen that the transfer function magnitude drops by -40 db/decade at high frequencies, yielding significant damping of higher harmonics. Therefore, the lowest harmonic component of the ripple dictates the filter requirements. A detailed outline considering filter design for power converters can be found in [51].

![Figure 5.9: Schematic drawing of a L-C low-pass filter.](image-url)
Deriving the technique in the frequency domain makes it possible to directly influence any frequency component of the ripple. This makes the proposed solution more universally applicable in comparison to a time domain analysis such as outlined in [52], [53].

Component values as well as size and cost of the output filter of a dc-dc switching converter are dictated by the fundamental harmonic ripple frequency, which is identical to the switching frequency for a single converter. Therefore, in practice, the goal is to minimize this fundamental harmonic component. Nevertheless, the proposed method can be applied to cancel any desired harmonic component of the ripple, as the concept is introduced for general values of $k$, where $k$ denotes the harmonic order.

Figure 5.10: Bode plot of the filter transfer function for $R = 5 \ \Omega$, $L = 10 \ \mu\text{H}$ and $C = 1 \ \mu\text{F}$.
CHAPTER 6

IMPROVED CURRENT RIPPLE CANCELLATION FOR ASYMMETRIC INTERLEAVED CONVERTERS

6.1 Mathematical Derivations

The mathematical derivations are carried out in a general format, such that the obtained results are independent of the converter topology and operating conditions.

6.1.1 Two Phase System \((N = 2)\)

For the case of \(N = 2\), a shift of the current waveforms between the phases of \(180^\circ\) yields the optimal cancellation effect. Nevertheless, a complete cancellation of the fundamental ripple component is not possible under asymmetric operating conditions.

6.1.2 Three-Phase System \((N = 3)\)

The required conditions for improved ripple cancellation in asymmetric multiphase circuits are derived for the case \(N = 3\) here. This yields sufficient conditions, as the cases with general \(N\) can easily be related back to this case as shown in [47], [46]. Generally, in multiphase dc-dc converters \(N\) current ripple waveforms can be observed whereas the shape varies dependent on the converter type and operation mode.

Figure 6.1 shows the input current for a buck converter in continuous conduction mode (CCM) and discontinuous conduction mode (DCM). Figure 6.2 shows the corresponding output currents. Figures 6.3 and 6.4 display the same waveforms for a boost-converter topology. These two converter types outline the most common waveforms that occur in dc-dc switching converters. Therefore, the proposed method can also be applied for other
converter types such as buck-boost, boost-buck (Čuk), flyback, and SEPIC. Depending on whether input or output ripple minimization is desired, the waveforms have to be selected according to the converter topology and its operating mode.

The asymmetric operation causes different magnitudes of the average current in each phase $I_{n}$, and different duty ratios $D_{n}$ can be observed. Summing up all $N$ ripple components yields the overall current ripple $\Delta i_{sum}$ whose magnitude can be minimized by adequately phase-shifting the ripple components of the individual phases. The goal is to minimize the lowest harmonic component of the ripple first, as it dictates the filter requirements. This is particularly important for input ripple cancellation, as the input filter may reduce converter performance and cause instability of the closed loop system [51].

To describe the waveforms shown in Figures 6.1-6.4 dependent on the phase-shift angles $\phi_{0n}$, they are represented by their Fourier series as

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cdot \cos(k(\omega t - \phi_0)) + b_k \cdot \sin(k(\omega t - \phi_0))$$

(6.1)
Figure 6.2: Output current waveforms for a buck converter in CCM (top) and DCM (bottom).

Figure 6.3: Input current waveforms for a boost converter in CCM (top) and DCM (bottom).
Figure 6.4: Output current waveforms for a boost converter in CCM (top) and DCM (bottom).

or in terms of the sum of phase-shifted cosine terms as

\[ f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} A_k \cdot \cos(k(\omega t - \phi_0) - \varphi_k), \]

(6.2)

where \( A_k \) is the magnitude and \( \varphi_k \) is the phase of the Fourier coefficient. The two forms are related by

\[ A_k = \sqrt{a_k^2 + b_k^2} \]

(6.3)

and

\[ \varphi_k = \text{atan2}(b_k, a_k), \]

(6.4)

where \( \text{atan2} \) is the inverse tangent function, that returns the corresponding angle in the interval of four quadrants \([-\pi, \pi]\). In the further course of the derivation, a notation with the pattern \( A_{nk} \) is used, where \( n \) is the phase index and \( k \) denotes the harmonic order.

The goal of the following derivations is to minimize the variation in current which corresponds to the expression inside the sum of the Fourier series given in Eq. (6.1) and Eq. (6.2).

The variable component of the current is denoted by \( i_n \) here and the overall
current ripple, when neglecting filter effects, can be written as

$$\tilde{i}_{\text{sum}}(t) = \sum_{n=1}^{N} \tilde{i}_n(t).$$

(6.5)

Using Eq. (6.2) yields

$$\tilde{i}_{\text{sum}}(t) = A_{1k} \cdot \cos(k(\omega t - \phi_{01}) - \varphi_{1k}) + \ldots$$

$$A_{2k} \cdot \cos(k(\omega t - \phi_{02}) - \varphi_{2k}) + \ldots$$

$$A_{3k} \cdot \cos(k(\omega t - \phi_{03}) - \varphi_{3k})$$

(6.6)

for $N = 3$, where $A_{nk}$ is determined based on the considered waveform.

This can equivalently be rewritten as a sum of phasors with magnitude $A_{nk}$ and phase

$$\theta_{nk} = k\phi_{0n} + \varphi_{nk}$$

(6.7)

as follows:

$$\tilde{I}_{\text{SUM}} = A_{1k} \cdot e^{-j\theta_{1k}} + A_{2k} \cdot e^{-j\theta_{2k}} + A_{3k} \cdot e^{-j\theta_{3k}},$$

(6.8)

where $\tilde{I}_{\text{SUM}}$ denotes the phasor of the summed current variation. Minimizing the overall current ripple or the variable component of the current respectively, corresponds to minimizing the magnitude of $\tilde{I}_{\text{SUM}}$. This can be achieved by minimizing the real and the imaginary part of the summed phasor. Using Euler’s identity, the above equation can be rewritten in its cartesian form

$$\tilde{I}_{\text{SUM}} = A_{1k} \cdot \cos(\theta_{1k}) - j \cdot A_{1k} \cdot \sin(\theta_{1k}) + \ldots$$

$$A_{2k} \cdot \cos(\theta_{2k}) - j \cdot A_{2k} \cdot \sin(\theta_{2k}) + \ldots$$

$$A_{3k} \cdot \cos(\theta_{3k}) - j \cdot A_{3k} \cdot \sin(\theta_{3k}).$$

(6.9)

Minimizing of this expression yields

$$\min_{\theta_{2k}, \theta_{3k}} |A_{1k} + A_{2k} \cdot \cos(\theta_{2k}) + A_{3k} \cdot \cos(\theta_{3k})| + |A_{2k} \cdot \sin(\theta_{2k}) + A_{3k} \cdot \sin(\theta_{3k})|,$$

(6.10)

where the first expression inside the absolute value corresponds to the real-
and the second expression corresponds to the imaginary part. $\theta_{1k} = 0$ has been chosen as a reference to simplify the expression.

To be able to obtain an analytic solution, we assume that the complete cancellation of a certain harmonic component of the current is possible by setting $\tilde{i}_{\text{sum}} = 0$ or $\tilde{I}_{\text{SUM}} = 0$, respectively.

$$
A_{1k} \cdot \cos(\theta_{1k}) - j \cdot A_{1k} \cdot \sin(\theta_{1k}) + \ldots
A_{2k} \cdot \cos(\theta_{2k}) - j \cdot A_{2k} \cdot \sin(\theta_{2k}) + \ldots
A_{3k} \cdot \cos(\theta_{3k}) - j \cdot A_{3k} \cdot \sin(\theta_{3k}) = 0,
$$

which immediately yields a set of equations that can be solved for the phase angles $\theta_{2k}$ and $\theta_{3k}$ ($\theta_{1k}$ can be chosen to be zero)

$$
A_{1k} + A_{2k} \cdot \cos(\theta_{2k}) + A_{3k} \cdot \cos(\theta_{3k}) = 0 \quad (6.12a)
A_{2k} \cdot \sin(\theta_{2k}) + A_{3k} \cdot \sin(\theta_{3k}) = 0. \quad (6.12b)
$$

An example of an asymmetric three-phase system, whose real and imaginary parts can be added up to zero by performing uneven phase-shifting, is illustrated in Fig. 6.5. The figure also contains an even three-phase reference system. Figure 6.6 shows the same phasors but with a conventional phase shift of $120^\circ$ between phases. It can be seen that the resulting summed phasor is non-zero.

To obtain an analytic solution, Eq. (6.12b) is rewritten with cosine terms only, which can be achieved by using the trigonometric identity

$$
\sin(x) = \begin{cases} 
\sqrt{1 - \cos^2(x)} & \text{for } x \in [0, \pi] \\
-\sqrt{1 - \cos^2(x)} & \text{for } x \in [\pi, 2\pi].
\end{cases} \quad (6.13)
$$

This yields four possible variations of Eq. (6.12b) dependent on the values
Figure 6.5: Asymmetric three-phase system with uneven phase-shift.

Figure 6.6: Asymmetric three-phase system with even phase-shift, resulting in a non-zero sum of the phasors.
of the angles $\theta_{2k}$ and $\theta_{3k}$

\[ A_{2k} \cdot \sqrt{1 - \cos^2(\theta_{2k})} + A_{3k} \cdot \sqrt{1 - \cos^2(\theta_{3k})} = 0 \]
\[ \text{for } \mod (\theta_{2k}, 2\pi) \in [0, \pi] \quad (6.14a) \]
\[ \text{and } \mod (\theta_{3k}, 2\pi) \in [0, \pi] \]

\[ A_{2k} \cdot \sqrt{1 - \cos^2(\theta_{2k})} - A_{3k} \cdot \sqrt{1 - \cos^2(\theta_{3k})} = 0 \]
\[ \text{for } \mod (\theta_{2k}, 2\pi) \in [0, \pi] \quad (6.14b) \]
\[ \text{and } \mod (\theta_{3k}, 2\pi) \in [\pi, 2\pi] \]

\[ -A_{2k} \cdot \sqrt{1 - \cos^2(\theta_{2k})} + A_{3k} \cdot \sqrt{1 - \cos^2(\theta_{3k})} = 0 \]
\[ \text{for } \mod (\theta_{2k}, 2\pi) \in [\pi, 2\pi] \quad (6.14c) \]
\[ \text{and } \mod (\theta_{3k}, 2\pi) \in [0, \pi] \]

\[ -A_{2k} \cdot \sqrt{1 - \cos^2(\theta_{2k})} - A_{3k} \cdot \sqrt{1 - \cos^2(\theta_{3k})} = 0 \]
\[ \text{for } \mod (\theta_{2k}, 2\pi) \in [\pi, 2\pi] \quad (6.14d) \]
\[ \text{and } \mod (\theta_{3k}, 2\pi) \in [\pi, 2\pi]. \]

Due to the periodicity of the cosine function, Eq. (6.14a) and Eq. (6.14d) as well as Eq. (6.14b) and Eq. (6.14c) are linearly dependent. To achieve an imaginary part equal to zero, the first and the second term of the equation have to be of opposite sign, as the magnitudes $A_{nk}$ are always positive. This condition leaves us with Eq. (6.14b) and Eq. (6.14c) as possible solutions, which are linearly dependent. Eq. (6.14b) has been chosen to obtain the solution outlined below. The system consisting of Eq. (6.12a) and Eq. (6.12b) can now be rewritten as

\[ A_{1k} + A_{2k} \cdot \cos(\theta_{2k}) + A_{3k} \cdot \cos(\theta_{3k}) = 0 \]
\[ (6.15a) \]
\[ A_{2k} \cdot \sqrt{1 - \cos^2(\theta_{2k})} - A_{3k} \cdot \sqrt{1 - \cos^2(\theta_{3k})} = 0. \]
\[ (6.15b) \]

Resubstituting as shown in Eq. (6.7) and rearranging yields the solutions
for the desired phase-shift angles

\[ \phi_{01} = -\frac{\phi_{1k}}{k} \]  

\[ \phi_{02} = \frac{1}{k} \left[ \cos^{-1}\left(\frac{1}{2} \cdot \frac{A_{2k}^2 - A_{2k}^2 - A_{1k}^2}{A_{1k} \cdot A_{2k}} \right) - \phi_{2k} \right] \]  

\[ \phi_{03} = \frac{1}{k} \left[ 2\pi - \cos^{-1}\left(\frac{1}{2} \cdot \frac{A_{2k}^2 - A_{1k}^2 - A_{3k}^2}{A_{1k} \cdot A_{3k}} \right) - \phi_{3k} \right] , \]

where Eq. (6.16b) was obtained by periodicity of the \( \cos^{-1} \) function.

For this solution to exist, the absolute values of the arguments of the \( \cos^{-1} \) terms have to be less than or equal to one, which places non-unique requirements on the combinations of current ripple magnitude and duty cycle of the phases. For the sake of easier implementation, \( \phi_{01} \) can be set to zero by adding its value to all three phase angles. This shifts the whole system, but preserves the relative position of the phasors to each other.

If a complete ripple component cancellation is not possible (conditions described above do not hold), the angles have to be calculated by using Eq. (6.10).

To visualize this function, \( k = 1 \) (fundamental ripple frequency) has been chosen. Figure 6.7 shows a surface plot of the function for fixed \( A_{nk} \) values, \( \theta_{11} = 0 \) and \( \theta_{21}, \theta_{31} \) variable. In this case a unique minimum exists that coincides with the values obtained by the analytic formulas derived above.

Figure 6.9 shows an example for a different set of \( A_{nk} \) values, where it is not possible to perform a complete ripple component cancellation. The solution of this problem shows that it is optimal to either operate phases without or with a 180° phase-shift. The two phases with the smaller magnitude of the first harmonic ripple component are operated in phase, while the phase with the highest magnitude is operated with a 180° phase shift to them. This means that the ripple is increased first by adding up two phases, to then cancel the ripple caused by the third phase. It can be shown that this yields the minimum absolute value of \( \bar{I}_{SUM} \) in all cases where a complete cancellation is not possible. The corresponding surface plot is shown in Fig. 6.8.

Consequently, the derived ripple minimization method can be implemented as a control technique that operates in two different modes as shown in
Figure 6.7: $|A_{\text{sum}}|$ depending on the values of $\theta_{21}$ and $\theta_{31}$. A unique minimum exists and is highlighted by the red marker.
Figure 6.8: $|A_{\text{sum}}|$ depending on the values of $\theta_{21}$ and $\theta_{31}$. The highlighted minimum is obtained for $\theta_{21} = \theta_{31} = 180^\circ$. 
\[ \Delta i_1 = 6.26 \text{ A}, \quad D_1 = 0.7, \quad \theta_{11} = 0^\circ \]
\[ \Delta i_2 = 3.26 \text{ A}, \quad D_2 = 0.15, \quad \theta_{21} = 180^\circ \]
\[ \Delta i_3 = 1.91 \text{ A}, \quad D_3 = 0.1, \quad \theta_{31} = 180^\circ \]

**Figure 6.9:** Asymmetric three-phase system with phase-shifts of 0° and 180° only.

---

**Figure 6.10:** Flowchart of the current ripple cancellation algorithm.

Fig. 6.10. Based on the values of \( A_{nk} \) the control decides if a complete cancellation is possible or not. If the arguments of the \( \cos^{-1} \) terms are less than or equal to one, a \( \tilde{I}_{\text{SUM}} \) phasor with an absolute value of zero can be achieved. The corresponding angles are calculated by Eq. (6.16a) - Eq. (6.16c). Otherwise the \( A_{nk} \) values are sorted by magnitude and \( \phi_{nk} = 0^\circ \) for \( \max |A_{nk}| \), and \( \phi_{nk} = \pi \) for the remaining \( A_{nk} \).
6.1.3 Higher Number of Phases ($N > 3$)

For the general case of many phases ($N > 3$), a phasor sorting technique similar to the one described in [46] is used together with the introduced framework. The case for general $N$ can best be analyzed by considering a phasor diagram (e.g. for $N = 6$) as outlined in Fig. 6.11 (left).

It can be seen that due to the asymmetric phase operation, the summed phasors are non-zero when simply obtaining the phase-shift angles by $\phi_{nk} = (n - 1) \cdot \frac{360^\circ}{N}$. A decrease of the summed ripple component can be achieved by sorting the phases. Placing phasors with similar magnitudes such that they point in opposite directions (shifted by $180^\circ$), but still keeping an even phase-shift of $\frac{360^\circ}{N}$ between phases, yields a lower magnitude of the summed ripple phasor as shown in Fig. 6.11 (right).

To achieve optimal ripple cancellation for the general case of $N > 3$, this sorting algorithm is applied for the $N-2$ phasors with the largest magnitudes. Subsequently, the summed ripple component for these phases is calculated and used together with the two remaining ripple components to form a new three-phase system for which the phase-shift angles are found as outlined in the case for $N = 3$.

![Figure 6.11: A lower magnitude of the summed ripple phasor can be achieved by sorting the phases for $N > 3$.](image)

Left - Unorganized phasors
Right - Sorted phasors (Phase 5 and Phase 6 swapped)
6.2 Buck Converter Application Example

The magnitudes of the phasors $A_{nk}$ have to be calculated based on the Fourier series representation of the desired current waveform. As this waveform is dependent on the converter topology and operating point (see Chapter 6.1), the example of a dc-dc buck converter has been chosen here to illustrate output ripple cancellation in CCM. The process of obtaining the values for $A_{nk}$ is the same for input ripple cancellation and other converter types in different operating modes.

6.2.1 Output Ripple Cancellation

The corresponding output current ripple waveform for a buck converter in CCM has been shown in Fig. 6.2. For a general triangular waveform, the coefficients $a_{nk}$ and $b_{nk}$ can be found as follows:

$$a_{nk} = \frac{\Delta i_n}{(\pi \cdot k)^2} \cdot \frac{1}{D_n \cdot (1 - D_n)} \cdot \left(\cos(2\pi k D_n) - 1\right)$$  \hspace{1cm} (6.17a)

$$b_{nk} = \frac{\Delta i_n}{(\pi \cdot k)^2} \cdot \frac{1}{D_n \cdot (1 - D_n)} \cdot \sin(2\pi k D_n),$$  \hspace{1cm} (6.17b)

or equivalently as \[54\]

$$A_{nk} = \frac{2\Delta i_n}{(\pi \cdot k)^2} \cdot \frac{1}{D_n \cdot (1 - D_n)} \cdot \sin(\pi k D_n)$$  \hspace{1cm} (6.18a)

$$\varphi_{nk} = \text{atan2}(b_{nk}, a_{nk}) = \begin{cases} 
-\arctan\left(\cot(\pi k D_n)\right) & a_{nk} > 0 \\
-\arctan\left(\cot(\pi k D_n)\right) + \pi & b_{nk} \geq 0, a_{nk} < 0 \\
-\arctan\left(\cot(\pi k D_n)\right) - \pi & b_{nk} < 0, a_{nk} < 0 \\
+\frac{\pi}{2} & b_{nk} > 0, a_{nk} = 0 \\
-\frac{\pi}{2} & b_{nk} < 0, a_{nk} = 0 \\
\text{not defined} & b_{nk} = 0, a_{nk} = 0 
\end{cases}$$  \hspace{1cm} (6.18b)

Note that the angles $\varphi_{nk}$ are independent of the current ripple magnitude $\Delta i_n$, but only dependent on the harmonic $k$ and the duty cycle $D_n$. Substituting the magnitudes and angles of the Fourier coefficients based on
Eqs. (6.18a) and (6.18b) yields the solution which is dependent on the current ripple magnitudes \( \Delta_i^m \) and duty cycles \( D_n \) of each phase as shown in Eq. (6.19a) - Eq. (6.19c).

\[
\phi_{01} = -\frac{\varphi_{1k}}{k} \quad (6.19a)
\]

\[
\phi_{02} = \frac{1}{k} \left\{ \cos^{-1} \left( \frac{1}{2} \cdot \frac{\Delta i_1^m}{D_1(1-D_1)} \cdot \frac{\Delta i_2^m}{D_2(1-D_2)} \cdot \sin(\pi kD_1) \cdot \sin(\pi kD_2) \right) - \ldots \right. \\
\left. \frac{\Delta i_3^2}{D_3^2(1-D_3)^2} \cdot \sin^2(\pi kD_3) - \ldots \right) \Bigg) - \varphi_{2k} \right\} \quad (6.19b)
\]

\[
\phi_{03} = \frac{1}{k} \left\{ 2\pi - \cos^{-1} \left( \frac{1}{2} \cdot \frac{\Delta i_1^m}{D_1(1-D_1)} \cdot \frac{\Delta i_2^m}{D_2(1-D_2)} \cdot \sin(\pi kD_1) \cdot \sin(\pi kD_3) \right) - \ldots \right. \\
\left. \frac{\Delta i_2^2}{D_2^2(1-D_2)^2} \cdot \sin^2(\pi kD_2) - \ldots \right) \Bigg) \Bigg) - \varphi_{3k} \right\} \quad (6.19c)
\]

Generally, an expression for the current ripple magnitude \( \Delta i_n \) can easily be obtained from the underlying differential equations of the converter. It can then be inserted into Eqs. (6.19b) and (6.19c). This results in a closed form expression to obtain the phase-shift angles that yield optimal current ripple cancellation for the \( k \)th harmonic component.

Because the current ripple magnitude \( \Delta i_n \) is dependent on the inductor value \( L_n \) of the converter, the proposed technique can also be applied to compensate for component value tolerances among different phases.

For a buck-converter, the magnitude of the current ripple can be calculated
from the input voltage and duty cycle as

\[ \Delta i_n = \frac{V_{\text{in},n}}{L_n} D_n (1 - D_n) T = \frac{V_{\text{out},n}}{L_n} (1 - D_n) T. \quad (6.20) \]

Using this in Eqs. (6.19b) and (6.19c) yields two expressions that are dependent on the input voltages \( V_{\text{in},n} \), duty cycles \( D_n \) and inductor values \( L_n \). If the inductor values are equal for all three phases, they cancel out and the expression can be simplified further. This has been used to perform output current ripple cancellation for a series topology of buck converters as shown in Fig. 3.2. The results are outlined in Chapter 7.

Similar relations hold true for a parallel topology. Here, the output voltage \( V_{\text{out}} \) is identical for all three converters. The output current is given by

\[ I_{\text{out}} = \frac{I_{\text{in},1}}{D_1} + \frac{I_{\text{in},2}}{D_2} + \frac{I_{\text{in},3}}{D_3} \quad (6.21) \]

for ideal converters. The inductor current ripple magnitude is conveniently expressed as

\[ \Delta i_n = \frac{V_{\text{out}}}{L_n} (1 - D_n) T \quad (6.22) \]

in this case. Inserting this into the general result yields a solution that only depends on the duty cycle in each phase and is independent of the different input currents.

For easier understanding, the cancellation of a harmonic component of the summed ripple current is illustrated in the time domain and in the frequency domain here. As discussed in Section 5.3, it is important to cancel the fundamental ripple component to release the filter requirements or achieve significantly smaller ripple. The harmonic index \( k \) is set to one, which yields a matching of the fundamental components of the Fourier series.

Figure 6.12 shows the ripple component of the inductor current in each of the phases for a dc-dc buck converter. It is operating at a switching frequency of 100 kHz and duty ratios of \( D_1 = 0.6 \), \( D_2 = 0.7 \) and \( D_3 = 0.8 \) as an example.

Figure 6.13 shows the frequency content of each inductor current ripple
Figure 6.12: Ripple component of the inductor currents in the different phases ($f_{sw} = 100$ kHz).
Figure 6.13: Fourier coefficients of the current ripple components.

component. The values are displayed at multiples of the fundamental frequency $k \cdot f_{sw}$ for $k = 0$ to $k = 10$.

$\theta_{11}$, $\theta_{21}$ and $\theta_{31}$ have been obtained by performing the outlined calculation steps. The parameters and results are displayed in Fig. 6.5. The angles $\phi_{01}$, $\phi_{02}$ and $\phi_{03}$, by which the fundamental harmonic components need to be shifted, can now easily be obtained. The results are as displayed in Fig. 6.12. Shown in Fig. 6.14 (top) are the fundamental harmonic components ($f = f_{sw} = 100$ kHz) of the triangular ripple waveforms, while Fig. 6.14 (bottom) displays the sum of all three fundamentals ($i_{\text{sum1}}(t)$).

It can be seen that the sum of all three fundamental components is zero. This is also shown in Fig. 6.15, where the fundamental harmonic component in the frequency domain (at $k = 1$) is zero. Therefore, the filter requirements
Figure 6.14: Fundamental component of the ripple currents displayed in Fig. 6.12, and sum of the fundamental components.

at the output are relaxed and the filter cutoff frequency can be increased. As illustrated in Fig. 6.15 (top), the cancellation of the fundamental component also yields a significant reduction of the ripple magnitude. To show the performance of the proposed technique, the sum of the phase currents is also displayed for an even phase shift ($\phi_{01} = 0^\circ$, $\phi_{02} = 120^\circ$, $\phi_{03} = 240^\circ$) (blue waveforms) under the same operating conditions.
Figure 6.15: Top - Added ripple components in the time domain.
Bottom - Added ripple components in the frequency domain.
CHAPTER 7

VERIFICATION OF THE PROPOSED
CONTROL TECHNIQUE

To verify the performance of the proposed ripple cancellation method, simulations are carried out. Furthermore, the hardware implementation of a multiphase dc-dc buck converter as presented in Chapter 4 is used to obtain experimental results.

7.1 Simulation Results

To evaluate the performance of the proposed ripple cancellation method, simulations in MATLAB and LTspice were carried out. The buck converter topology shown in Fig. 3.2 has been used as a sample implementation. By calculating the current ripple magnitudes for this topology and inserting into Eqs. (6.19b) and (6.19c), the desired phase angles can be obtained, which are nonlinearly dependent on \( V_{in,n} \) and \( D_n \). For direct comparison, the converter has also been simulated using identical parameters but with a conventional 120° phase-shift. The phase angles are dependent on six variables (for \( N = 3 \)) that may be completely independent or dependent on each other based on the specific application. In the given application of distributed MPPT for example, \( V_{in,n} \) and \( D_n \) are related by the I-V curves of the sub-modules as discussed in Chapter 2. However, the voltages at the MPPs, which are equal to the converter input voltages, can be considered to be approximately constant over a certain range of operating conditions [45] (see Chapter 2). Therefore, for the simulation with results displayed in Fig. 7.1, the input voltages have been chosen equally as \( V_{in,1} = V_{in,2} = V_{in,3} = 12 \) V.

The effective improvement depends on the specific point of operation for each converter, and the more the calculated phase angles deviate from the conventional case (\( \phi_{01} = 0^\circ, \phi_{02} = 120^\circ, \phi_{03} = 240^\circ \)), the higher the improve-
Figure 7.1: Improvement in ripple cancellation for $V_{IN1} = V_{IN2} = V_{IN3} = 12$ V and $D_1 = 0.9$, $D_2$ and $D_3$ variable.

The relative decrease in current ripple achieved with the proposed technique has been calculated in percent as

$$\left| \frac{\Delta i_{sum} - \Delta i_{sum,sym}}{\Delta i_{sum,sym}} \right| \cdot 100\%$$  \hspace{1cm} (7.1)

where $\Delta i_{sum,sym}$ is the overall current ripple magnitude for the case with conventional (symmetric) phase-shift. Figure 7.1 shows the improvement as a surface plot, where $D_1$ has been fixed at 0.9 (running one converter at a high duty cycle is typical for MPPT applications) while $D_2$ and $D_3$ are varied.

For this specific case, on average, a 25% lower output current ripple can be achieved compared to a conventional phase-shift implementation. Clearly, the relative improvement is zero at symmetric operating points ($D_1 = D_2 = D_3$).

Moreover, the topology shown in Fig. 3.2 has been implemented in LTspice to verify the proposed technique in a circuit simulator while taking output capacitor effects into account. The simulation used the circuit parameters
Figure 7.2: Simulation of the converter output current with conventional and proposed phase-shifting technique.

$C_{\text{IN},n} = 40 \ \mu F$, $C_{\text{OUT},n} = 30 \ \mu F$, $L_n = 4.7 \ \mu H$ and a switching frequency of 100 kHz. The simulation has been carried out using asymmetric operating conditions with input voltages $V_1 = 14 \ \text{V}$, $V_2 = 12 \ \text{V}$, and $V_3 = 10 \ \text{V}$ and duty cycles $D_1 = 0.6$, $D_2 = 0.7$, and $D_3 = 0.8$, which corresponds to the phasor diagram shown in Fig. 6.5. The results illustrate the performance of the proposed technique for a case with different input voltages. The output current waveforms are displayed in Fig. 7.2. The conventional phase-shifting technique yields a load-current ripple of 35 mA, whereas the proposed technique reduced the ripple to 20 mA. Therefore, the relative improvement is 43% (calculated by Eq. (7.1)) for this operating point.

7.2 Experimental Results

A hardware implementation presented in Chapter 4 has been used for the experimental verification. The converters were connected as shown in Fig. 3.2. To implement the phase-shift, the three PSCs were started at different times using the internal timer of the MC. The desired uneven phase angles can either be calculated online or taken from a precalculated lookup table.

Figure 7.3 shows the phase currents as well as the load-current, measured with the converters of the experimental prototype at the operating conditions...
\[ V_1 = 14 \text{ V}, \quad V_2 = 12 \text{ V}, \quad \text{and} \quad V_3 = 10 \text{ V} \] and duty cycles \( D_1 = 0.6, \quad D_2 = 0.7, \) and \( D_3 = 0.8 \) (see phasor diagram shown in Fig. 6.5). The common output current has been chosen to be 5 A. An even phase-shift with \( \phi_{01} = 0^\circ, \quad \phi_{02} = 120^\circ, \) and \( \phi_{03} = 240^\circ \) has been applied, which yields an output current ripple of 220 mA.

Shown in Fig. 7.4 is the measurement with the same topology and same operating conditions, but with the proposed uneven phase-shifting technique applied. The phase angles are now \( \phi_{01} = 0^\circ, \quad \phi_{02} = 138.4^\circ, \) and \( \phi_{03} = 185.3^\circ, \) yielding an output current ripple of 60 mA. This corresponds to a relative improvement of 73% (calculated by Eq. (7.1)).

To analyze the frequency content of the output ripple waveforms, a fast Fourier transform (FFT) was performed using the mathematical tool set of the oscilloscope. The FFT is carried out using a Hanning window, as it effectively reduces spectral leakage and provides good frequency resolution and amplitude accuracy. The results are shown in Fig. 7.5 for an even phase-shift and in Fig. 7.6 for the proposed phase-shifting technique. It can be seen that the fundamental ripple component is attenuated by 14.8 dB when using the proposed method.
Figure 7.3: Measured phase currents and load current of the experimental prototype in series connection with even phase-shift.

Figure 7.4: Measured phase currents and load current of the experimental prototype in series connection with proposed phase-shift and reduced output current ripple.
Figure 7.5: Spectrum of the output current with even phase-shift.

First harmonic ripple component (f = 100 kHz)
Amplitude: -23.6 dB

Figure 7.6: Spectrum of the output current with proposed phase-shift.

First harmonic ripple component (f = 100 kHz)
Amplitude: -38.4 dB
7.3 Conclusions

An analytical derivation of a control strategy that enables interleaved operation and improved ripple cancellation for multiphase converters with non-uniform duty cycles has been presented. Through simulations and experimental results, we have illustrated the improved performance of the proposed method compared to conventional ripple cancellation methods. The calculation of uneven phase angles based on the Fourier space representation of the current ripple goes beyond the previous works in the field [55], [56], [52], [53]. The application of the proposed technique to multiphase converters with different input voltages, running at uneven duty cycles, extends the scope of [46] to the general case.

7.4 Future Work

7.4.1 Phase Shifting for $N > 3$

For the case of more than three phases, a sorting algorithm has been presented that relates this case back to the case where $N = 3$. This is based on the intention of minimizing a single harmonic component of the ripple (usually $k = 1$). For $N = 3$ it is only possible to cancel a single harmonic component under asymmetric operating conditions. Nevertheless, it should be investigated, whether it is possible to cancel two different harmonics with a phase count of $N = 6$, three different harmonics with a phase count of $N = 9$, and so on. In the outlined application of sub-module MPPT, the implementation of such a control would require communication between converters of different solar panels. For other applications with a central control for all converter-phases, it can easily be implemented.

7.4.2 Analysis of the Computational Efforts for Different Methods of Implementation

Calculating the required uneven phase-shift angles during MC operation requires the highest computational power or time and little memory. Contrarily, an implementation using a lookup-table with values that have been
calculated beforehand requires low computational effort but uses more memory. A analysis should be carried out to determine the actual requirements in terms of computational power and memory. This will make it easier to select the most suitable solution based on the available hardware and application.
CHAPTER 8

EXPERIMENTAL CHARACTERIZATION

8.1 Experimental Setup

The efficiency of the proposed symmetrical dc-dc buck converter topology is measured using the setup shown in Fig. 8.1. The dc power supply and the electronic load are controlled by a computer via GPIB bus. To manipulate the duty ratio, a RS-232 communication between the microcontroller of the converter and the computer is established. Input and output power is measured by using two digital multimeters, that measure voltage and current separately at each terminal. The measurements are triggered simultaneously and the obtained values are sent to the computer via GPIB. Measurements and communication are controlled by a Python script (see Appendix C).
8.2 Experimental Results

All efficiency measurements include control and gate-driving losses in addition to the power stage losses.

Figure 8.2 shows the efficiency plotted over the output voltage at different output currents. The input voltage has been fixed to 10 V. The achieved peak efficiency for these operating conditions is 96.5%.

Figure 8.3 shows the same data for an input voltage of 12 V. The achieved peak efficiency for these operating conditions is 96.0%.

To obtain the results displayed in Fig. 8.4, an input voltage of 15 V has been chosen. The achieved peak efficiency is 95.0%.

The input voltages have been selected to match the typical range that is observed for the output voltage of a sub-module. Lower output voltages occur when a sub-module is shaded. The string current is affected by the insulation of the PV system.

To compare the actual losses in the circuit with the calculations shown in Chapter 4, the loss distribution at the operating point $V_{in} = 10$ V, $D=0.9$, $I_{out} = 5$ A is analyzed further. The control losses are constant over a wide range of operating points. The typical microcontroller power consumption has been determined to be 0.11 W. The gate-driver supply losses were found to increase as a higher input voltage is applied. Table 8.1 shows the different loss types and their numerical values. Figure 8.5 outlines the contribution of
Figure 8.2: Converter efficiency over output voltage at different output currents. The input voltage has been fixed to 10 V.

Figure 8.3: Converter efficiency over output voltage at different output currents. The input voltage has been fixed to 12 V.
the individual loss types to the total loss in a pie chart. It can be seen that the majority of the overall loss is caused by switching and package losses. Possible improvements are discussed in the next section.

To evaluate the MPPT performance of the circuit, laboratory experiments using a solar emulator as described in [57] have been carried out. The output characteristics of a solar panel are realistically emulated by replacing the photo-current source (irradiation) with a laboratory power supply that is operated in constant current mode. The MPP algorithm was implemented with a duty cycle step-size of 1%, which allows for fast tracking and good steady-state accuracy under practical conditions. For the initial sweep the step-size is increased to 5% to allow for fast estimation of the MPP region. The microcontroller used in this project offers a 12 bit PWM resolution which allows to significantly reduce the step-size if desired. To increase the precision of the output voltage measurements, 64 ADC samples were averaged per measurement (oversampling [58]). Table 8.2 shows the MPPT parameters.

A Solar Power SPI-020M-9.5 panel with a nominal output power of 20 W was used for the experiment. The external current source of the solar emulator was set to 2.5 A, which corresponds to the short circuit current of the
Table 8.1: Loss distribution

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input voltage</td>
<td>10 V</td>
</tr>
<tr>
<td>Duty ratio</td>
<td>0.9</td>
</tr>
<tr>
<td>Output current</td>
<td>5 A</td>
</tr>
<tr>
<td>Efficiency</td>
<td>95.81%</td>
</tr>
<tr>
<td>Input power</td>
<td>44.73 W</td>
</tr>
<tr>
<td>Output power</td>
<td>42.86 W</td>
</tr>
<tr>
<td>Total losses</td>
<td>1.87 W</td>
</tr>
<tr>
<td>Control loss</td>
<td>113 mW</td>
</tr>
<tr>
<td>Gate-driver supply loss</td>
<td>77 mW</td>
</tr>
<tr>
<td>Power stage conduction losses</td>
<td>91 mW (calculated)</td>
</tr>
<tr>
<td>Inductor DCR loss</td>
<td>46 mW (calculated)</td>
</tr>
<tr>
<td>Inductor core loss</td>
<td>2 mW (calculated)</td>
</tr>
<tr>
<td>Switching losses</td>
<td>1.541 W (calculated)</td>
</tr>
</tbody>
</table>

Figure 8.5: Contribution of the individual loss types to the overall loss.

Table 8.2: MPPT parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duty cycle step-size</td>
<td>1%</td>
</tr>
<tr>
<td>ADC resolution</td>
<td>10 bit</td>
</tr>
<tr>
<td>Samples per measurement</td>
<td>64</td>
</tr>
<tr>
<td>Initial sweep step-size</td>
<td>5%</td>
</tr>
<tr>
<td>Minimum duty cycle</td>
<td>5%</td>
</tr>
<tr>
<td>Maximum duty cycle</td>
<td>99%</td>
</tr>
</tbody>
</table>
panel. The measured I-V and P-V curves of the panel at these operating conditions are shown in Fig. 8.6.

The MPP tracker was connected to the panel, while input voltage and input current were measured separately. Figure 8.7 shows the output power of the panel (input power of the converter) over time. The MPPT operation was intentionally slowed down by introducing delays into the control code to show the different steps of operation. The initial sweep to find the approximate location of the MPP is clearly visible in the beginning. After arriving at the MPP (steady state), the power varies slightly around its maximum value, which is due to the inherent oscillation of the employed perturb and observe algorithm.

The tracking efficiency, which is given by

$$\eta_{\text{MPPT}} = \frac{P_{\text{MPP}}}{P_{\text{MPP,tracked}}}$$

(8.1)

has been calculated. Here, $P_{\text{MPP}}$ denotes the actual MPP power of the panel (as obtained from the P-V curve), while $P_{\text{MPP,tracked}}$ is the power extracted by the MPP tracker. For the given measurement, a tracking efficiency of 98.6% was calculated. The efficiency can be increased by increasing the
Figure 8.7: Panel output power over time with connected MPP tracker.

PWM resolution as well as the precision of the output voltage measurements (averaging over more samples).

8.3 Conclusions

In the previous chapters, the implementation and control of a sub-module level MPPT circuit, using three dc-dc buck converters, has been presented. The usage of a single microcontroller for all power stages reduces redundancy and thereby increases power efficiency. Moreover, it facilitates automatic bypass mode detection and size reduction, such that the circuit can be placed in the junction box of a regular solar panel without the need for a separate enclosure that would increase system cost.

8.4 Future Work

The central control of three power-stages in per sub-module MPPT goes beyond the scope of most previous works in the field. Nevertheless, there are
areas with potential for optimization.

8.4.1 Power Stage Optimization

Despite the optimization of the power stages for the typically high duty ratios during MPPT operation, the converter efficiency is low in comparison to other works [14], [15]. This is likely due to the discrete implementation of the power stage, which results in increased PCB and switching losses in comparison to an integrated solution. Therefore, a redesign of the circuit using an integrated power stage, such as a DrMOS device, is recommended. This would help to increase efficiency, simplify the board layout, and further reduce cost and size.

8.4.2 Field Measurements

To verify the performance of the proposed distributed sub-module MPPT circuit under real-life conditions, a comparative performance analysis should be carried out in the field. A solar panel, connected to a microinverter which feeds the produced energy into the grid, would provide an easy to implement test setup. The connection of the solar panel to the inverter can be done directly in the conventional way as shown in Fig. 8.8, and with the proposed MPP trackers in between as shown in Fig. 8.9. The MPPT circuits are connected in series at their outputs and the bypass diodes, as shown in the conventional case, can be removed. The measurement will evaluate the interaction between the local MPPT algorithm running for each dc-dc converter and the global MPPT algorithm, that is run by the microinverter.

The power delivered through the dc-bus can be monitored by sampling the voltage and current as shown in Fig. 8.8 and Fig. 8.9. Integration over time yields the corresponding energy generation. To compare the conventional topology with bypass diodes to the topology with the sub-module MPPT, periodic switching between the setups should be implemented. This allows the use of the same solar panel and microinverter in both cases, and thereby effectively eliminates errors that may be caused by differences in the measurement setup. Such differences could result from manufacturing variations.
between two solar panels or microinverters, as well as slight deviations in irradiation between panels in the measurement setup.

8.4.3 Bypass Mode Detection for Multiple Panels

When the sub-modules of a panel operate without mismatch but conditions at different panels are unequal, it is necessary to run the three converters within a panel at the same duty ratios without bypassing. It needs to be evaluated how the automatic bypass mode detection can be implemented under these conditions.

A straightforward option would be to establish communication between all converters in the string, to transmit information about the MPPs of all sub-modules. Only if the entire string is operating at the same MPP can all converters be bypassed without lowering the output power. Otherwise, bypass mode should be entered for all sub-modules operating at the equal
Figure 8.9: Measurement setup to determine the energy yield of a solar panel connected to a microinverter with intermediate sub-module MPPT.

MPP with the highest $I_{mpp}$. The inverter will then adjust the string current to this value of $I_{mpp}$, allowing sub-modules at lower $I_{mpp}$ to operate at their individual MPPs though the dc-dc converters. A drawback of this approach is that using conventional communication methods will increase system cost and decrease efficiency.

Therefore, it should be evaluated if an algorithm without communication can be implemented. A first approach can be to monitor the change in string current, after the bypass mode for a whole panel has been activated. To detect bypass conditions for all converters within the same panel, the control described in Section 4.3 can be applied. Shortly after bypassing the converters, it can generally not be assumed that $I_{string} = I_{mpp}$ of the corresponding panel. It is necessary to wait for the global MPPT algorithm to adjust $I_{string}$ at its next operating cycle. If $I_{string} = I_{mpp}$ after the adjustment, the converters of this panel can remain bypassed. If not, there exists a panel in the string that operates at a different MPP and the converters need to resume normal MPPT operation. The decision whether $I_{string} = I_{mpp}$ or not cannot be made directly, as there is no local current sensing. Nevertheless,
it can easily be checked if $V_{\text{mpp}}$ before bypassing and after the readjustment of the string current remains the same.

8.4.4 Communication

For control, diagnosis and to obtain field measurement data, it is desirable to communicate between converters. The communication should be implemented without reducing the system efficiency or increasing system cost. Wireless communication methods have comparably high energy demand and require additional hardware components, which will increase system cost. Conventional wired communication (e.g. I$^2$C) requires additional signal lines that may be difficult to implement and impose additional cost. Consequently, wired communication using the exiting power-lines is preferred. Research has been done for using the common input voltage bus of multiple converters to transmit information between them [59], [60], [61], [62]. However, this is not applicable to the system presented in this work, as converters are supplied by individual sources rather than a common bus. Instead, the outputs of all converters are connected in series. A communication signal can be modulated onto the output bus by using amplitude, phase or frequency modulation. Amplitude modulation requires increasing the ripple magnitude, which introduces additional losses. The phase of the output current of each converter is usually fixed by interleaved operation (see Chapter 6). Modulating the switching frequency can easily be performed by controlling the PSCs accordingly. For reliable communication via the output bus, its transmission characteristics should be investigated further. A neighbor-to-neighbor communication can be sufficient for signal transmission at low communication speeds.
APPENDIX A

SCHEMATIC DRAWING, PCB LAYOUT, AND COMPONENT LISTINGS FOR THE SUB-MODULE MPPT

A.1 Schematic Drawing

To improve visibility, the schematic drawing of the converter is split into functional units that are shown separately.

Figure A.1: Schematic drawing of the microcontroller with peripheral circuits.
Figure A.2: Schematic drawing of the first converter.
Figure A.3: Schematic drawing of the second converter including level shifting circuits.
Figure A.4: Schematic drawing of the third converter including level shifting circuits.
A.2 PCB Layout

Figure A.5: PCB layout top view including connectors.
Figure A.6: PCB layout bottom view including connectors.
Figure A.7: Top silkscreen for all three converters.
Figure A.8: Bottom silkscreen for all three converters.
Figure A.9: Top-layer copper for all three converters.
Figure A.10: PCB middle layer 1.
Figure A.11: PCB middle layer 2.
Figure A.12: Bottom-layer copper for all three converters.
A.3 Component Listings

The following pages contain the component listings for each converter within the three converter system separately. Furthermore, the listings are divided up into components that are placed on the top-side of the PCB and components that are placed on the bottom-side.
Table A.1: Component listing for converter 1 top side

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Type</th>
<th>Value</th>
<th>Package</th>
<th>Manufacturer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT90PWM316</td>
<td>AVR AT90PWM316</td>
<td>MEGA</td>
<td>QFN32,7x7</td>
<td>Atmel</td>
<td>Microcontroller</td>
</tr>
<tr>
<td>Q_HL1</td>
<td>FDMS8023S</td>
<td>Power 56</td>
<td>Power 56</td>
<td>Fairchild Semiconductor</td>
<td>High-side MOSFET</td>
</tr>
<tr>
<td>Q_LOW1</td>
<td>FDS8882</td>
<td>8-SSOC</td>
<td>ST</td>
<td>Fairchild Semiconductor</td>
<td>Low-side MOSFET</td>
</tr>
<tr>
<td>Q_BYP1</td>
<td>STL150N3LLH5</td>
<td>Power 56</td>
<td>Power 56</td>
<td>ST</td>
<td>Bypass MOSFET</td>
</tr>
<tr>
<td>C20</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>Input voltage divider filter</td>
</tr>
<tr>
<td>R16</td>
<td>Resistor, 1%</td>
<td>1k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Bypass MOSFET gate discharge resistor</td>
</tr>
<tr>
<td>Q1</td>
<td>NDS355AN</td>
<td>NDS355AN</td>
<td>SSOT3</td>
<td>Fairchild</td>
<td>Discharge MOSFET</td>
</tr>
<tr>
<td>R17</td>
<td>Resistor, 1%</td>
<td>10M</td>
<td>R0603</td>
<td>Yageo</td>
<td>Discharge MOSFET gate discharge resistor</td>
</tr>
<tr>
<td>R9</td>
<td>Resistor, 1%</td>
<td>100k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Input voltage divider upper resistor</td>
</tr>
<tr>
<td>C8</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Input capacitor</td>
</tr>
<tr>
<td>C10</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Input capacitor</td>
</tr>
<tr>
<td>C18</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Input capacitor</td>
</tr>
<tr>
<td>R31</td>
<td>Resistor, 1%</td>
<td>0</td>
<td>R0603</td>
<td>Yageo</td>
<td>Q_LOW1 gate drive resistor</td>
</tr>
<tr>
<td>R32</td>
<td>Resistor, 1%</td>
<td>0</td>
<td>R0603</td>
<td>Q_HL1 gate discharge resistor</td>
<td></td>
</tr>
<tr>
<td>C52</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>filter capacitor between AREF and AGND</td>
</tr>
<tr>
<td>C54</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>low pass filter capacitor for AVCC</td>
</tr>
<tr>
<td>R55</td>
<td>Resistor, 1%</td>
<td>100k</td>
<td>R0603</td>
<td>Yageo</td>
<td>low pass filter capacitor for AVCC</td>
</tr>
<tr>
<td>C23</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Output capacitor</td>
</tr>
<tr>
<td>C24</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Output capacitor</td>
</tr>
<tr>
<td>C25</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Output capacitor</td>
</tr>
<tr>
<td>C51</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>voltage regulator output capacitor</td>
</tr>
<tr>
<td>VDO</td>
<td>TLV70450</td>
<td>TLV70450</td>
<td>SOT23-5L</td>
<td>Texas Instruments</td>
<td>Voltage regulator</td>
</tr>
<tr>
<td>R32</td>
<td>Resistor, 1%</td>
<td>10M</td>
<td>R0603</td>
<td>Yageo</td>
<td>Q_LOW1 gate discharge resistor</td>
</tr>
<tr>
<td>C27</td>
<td>Ceramic 50V C0G, NP0</td>
<td>10pF</td>
<td>C0603</td>
<td>Murata</td>
<td>Bypass circuit boot capacitor</td>
</tr>
<tr>
<td>R10</td>
<td>Resistor, 1%</td>
<td>30k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Input voltage divider lower resistor</td>
</tr>
<tr>
<td>R14</td>
<td>Resistor, 1%</td>
<td>30k</td>
<td>R0603</td>
<td>Yageo</td>
<td>VL1 voltage divider lower resistor</td>
</tr>
</tbody>
</table>
Table A.2: Component listing for converter 1 bottom side

<table>
<thead>
<tr>
<th>Bottom Side Ref.</th>
<th>Type</th>
<th>Value</th>
<th>Package</th>
<th>Manufacturer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>SER2000 Series</td>
<td>SER2000 Series</td>
<td>SER2300</td>
<td>Coilcraft</td>
<td>Power Inductor</td>
</tr>
<tr>
<td>D10</td>
<td>1A Diode</td>
<td>?</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Boot diode for gate driver</td>
</tr>
<tr>
<td>R27</td>
<td>Resistor, 1%</td>
<td>100k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Boot resistor for gate driver</td>
</tr>
<tr>
<td>C31</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>Boot capacitor for gate driver</td>
</tr>
<tr>
<td>R12</td>
<td>Resistor, 1%</td>
<td>30k</td>
<td>R0603</td>
<td>Yageo</td>
<td>_VL_1 voltage divider upper resistor</td>
</tr>
<tr>
<td>C21</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>_VL_1 voltage divider filter</td>
</tr>
<tr>
<td>R13</td>
<td>Resistor, 1%</td>
<td>?</td>
<td>R0603</td>
<td>Yageo</td>
<td>_VL_1 voltage divider lower resistor</td>
</tr>
<tr>
<td>C36</td>
<td>Ceramic 16V X5R</td>
<td>30k</td>
<td>C0603</td>
<td>Yageo</td>
<td>Gate driver Vdd capacitor</td>
</tr>
<tr>
<td>IC1</td>
<td>FAN7390</td>
<td>FAN7391</td>
<td>SOP08</td>
<td>Fairchild Semiconductor</td>
<td>Gate driver</td>
</tr>
<tr>
<td>D1</td>
<td>1N4148 Diode</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Bypass circuit boot diode</td>
</tr>
<tr>
<td>C26</td>
<td>Ceramic 50V X7R</td>
<td>1nF / 1000pF</td>
<td>C0603</td>
<td>Murata</td>
<td>Bypass circuit shift capacitor</td>
</tr>
<tr>
<td>D2</td>
<td>1N4148 Diode</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Bypass circuit capacitor charging diode</td>
</tr>
</tbody>
</table>
Table A.3: Component listing for converter 2 top side

<table>
<thead>
<tr>
<th>Top side Ref.</th>
<th>Type</th>
<th>Value</th>
<th>Package</th>
<th>Manufacturer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q_HI.2</td>
<td>FDMS8023S</td>
<td>FDMS8023S</td>
<td>Power 56</td>
<td>Fairchild Semiconductor</td>
<td>High side MOSFET</td>
</tr>
<tr>
<td>Q_LOW.2</td>
<td>FDSS8882</td>
<td>FDSS8883</td>
<td>S-80OF</td>
<td>Fairchild Semiconductor</td>
<td>Low side MOSFET</td>
</tr>
<tr>
<td>Q_BYP.2</td>
<td>STL150N3LLH5</td>
<td>STL150N3LLH6</td>
<td>PowerFFlat? (6x5) ST</td>
<td>ST</td>
<td>Bypass MOSFET</td>
</tr>
<tr>
<td>C6</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>Input voltage divider filter</td>
</tr>
<tr>
<td>C9</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>VL_2 voltage divider filter</td>
</tr>
<tr>
<td>R8</td>
<td>Resistor, 1%</td>
<td>1k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Bypass MOSFET gate discharge resistor</td>
</tr>
<tr>
<td>Q2</td>
<td>NDS355AN</td>
<td>NDS355AN</td>
<td>SSOT3</td>
<td>Fairchild</td>
<td>Discharge MOSFET</td>
</tr>
<tr>
<td>R19</td>
<td>Resistor, 1%</td>
<td>10M</td>
<td>R0603</td>
<td>Yageo</td>
<td>Input voltage divider upper resistor</td>
</tr>
<tr>
<td>C2</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Input capacitor</td>
</tr>
<tr>
<td>C3</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Input capacitor</td>
</tr>
<tr>
<td>C4</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Input capacitor</td>
</tr>
<tr>
<td>C5</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Input capacitor</td>
</tr>
<tr>
<td>R45</td>
<td>Resistor, 1%</td>
<td>0</td>
<td>R0603</td>
<td>Q_LOW.2 gate drive resistor</td>
<td></td>
</tr>
<tr>
<td>R25</td>
<td>Resistor, 1%</td>
<td>0</td>
<td>R0603</td>
<td>Gate driver supply voltage bridge</td>
<td></td>
</tr>
<tr>
<td>R47</td>
<td>Resistor, 1%</td>
<td>10M</td>
<td>R0603</td>
<td>Q_HI.2 gate discharge resistor</td>
<td></td>
</tr>
<tr>
<td>R26</td>
<td>Resistor, 1%</td>
<td>0</td>
<td>R0603</td>
<td>Yageo</td>
<td>Input capacitor</td>
</tr>
<tr>
<td>R7</td>
<td>Resistor, 1%</td>
<td>10M</td>
<td>R0603</td>
<td>Q_HI.2 gate drive resistor</td>
<td></td>
</tr>
<tr>
<td>C11</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Output capacitor</td>
</tr>
<tr>
<td>C12</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Output capacitor</td>
</tr>
<tr>
<td>C13</td>
<td>Ceramic 16V X5R</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Output capacitor</td>
</tr>
<tr>
<td>R46</td>
<td>Resistor, 1%</td>
<td>10M</td>
<td>R0603</td>
<td>Q_LOW.2 gate discharge resistor</td>
<td></td>
</tr>
<tr>
<td>C15</td>
<td>Ceramic 50V C0G, NP0</td>
<td>10pF</td>
<td>C0603</td>
<td>Murata</td>
<td>Bypass circuit boot capacitor</td>
</tr>
<tr>
<td>R23</td>
<td>Resistor, 1%</td>
<td>15k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Input voltage divider lower resistor</td>
</tr>
<tr>
<td>R6</td>
<td>Resistor, 1%</td>
<td>15k</td>
<td>R0603</td>
<td>V_L.2 voltage divider lower resistor</td>
<td></td>
</tr>
<tr>
<td>STAT_LED</td>
<td>LED red 10mA 10mcd</td>
<td>LED</td>
<td>CHIP-LED0805</td>
<td>Lite-On Inc</td>
<td>Status LED, on board of converter 2</td>
</tr>
<tr>
<td>R54</td>
<td>Resistor, 1%</td>
<td>330</td>
<td>R0603</td>
<td>Stackpole Electronics Inc</td>
<td>Status LED series resistor</td>
</tr>
<tr>
<td>D12</td>
<td>1N4148 Diode</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Level shifting diode discharge 2 signal</td>
</tr>
<tr>
<td>R22</td>
<td>Resistor, 1%</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Level shifting resistor discharge 2 signal</td>
</tr>
<tr>
<td>C34</td>
<td>Ceramic 50V X7R</td>
<td>1uF</td>
<td>C0603</td>
<td>Murata</td>
<td>Level shifting capacitor discharge 2 signal</td>
</tr>
<tr>
<td>D7</td>
<td>1N4148 Diode</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Level shifting diode PWM_HI.2 signal</td>
</tr>
<tr>
<td>R20</td>
<td>Resistor, 1%</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Level shifting resistor PWM_HI.2 signal</td>
</tr>
<tr>
<td>C16</td>
<td>Ceramic 50V X7R</td>
<td>1uF</td>
<td>C0603</td>
<td>Murata</td>
<td>Level shifting capacitor PWM_HI.2 signal</td>
</tr>
<tr>
<td>D8</td>
<td>1N4148 Diode</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Level shifting diode PWM_HI.2 signal</td>
</tr>
<tr>
<td>R21</td>
<td>Resistor, 1%</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Level shifting resistor PWM_HI.2 signal</td>
</tr>
<tr>
<td>C32</td>
<td>Ceramic 50V X7R</td>
<td>1uF</td>
<td>C0603</td>
<td>Murata</td>
<td>Level shifting capacitor PWM_HI.2 signal</td>
</tr>
<tr>
<td>D28</td>
<td>1N4148 Diode</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Level shifting diode CLK_2 signal</td>
</tr>
<tr>
<td>R38</td>
<td>Resistor, 1%</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Level shifting resistor CLK_2 signal</td>
</tr>
<tr>
<td>C33</td>
<td>Ceramic 50V X7R</td>
<td>1uF</td>
<td>C0603</td>
<td>Murata</td>
<td>Level shifting capacitor CLK_2 signal</td>
</tr>
</tbody>
</table>
**Table A.4: Component listing for converter 2 bottom side**

<table>
<thead>
<tr>
<th>Bottom Side Ref.</th>
<th>Type</th>
<th>Value</th>
<th>Package</th>
<th>Manufacturer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2</td>
<td>SER2000 Series</td>
<td>SER2000</td>
<td>SER2300</td>
<td>Coilcraft</td>
<td>Power Inductor</td>
</tr>
<tr>
<td>D27</td>
<td>1A Diode</td>
<td></td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Boot diode for gate driver</td>
</tr>
<tr>
<td>R37</td>
<td>Resistor, 1%</td>
<td>100k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Boot resistor for gate driver</td>
</tr>
<tr>
<td>C1</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>Boot capacitor for gate driver</td>
</tr>
<tr>
<td>R4</td>
<td>Resistor, 1%</td>
<td>100k</td>
<td>R0603</td>
<td>Yageo</td>
<td>VL,2 voltage divider upper resistor</td>
</tr>
<tr>
<td>R37</td>
<td>Resistor, 1%</td>
<td>100k</td>
<td>R0603</td>
<td>Yageo</td>
<td>VL,2 voltage divider upper resistor</td>
</tr>
<tr>
<td>C7</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>VL,2 voltage divider filter</td>
</tr>
<tr>
<td>R5</td>
<td>Resistor, 1%</td>
<td>15k</td>
<td>R0603</td>
<td>Yageo</td>
<td>VL,2 voltage divider lower resistor</td>
</tr>
<tr>
<td>C17</td>
<td>Ceramic 16V X5R</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>Gate driver Vdd capacitor</td>
</tr>
<tr>
<td>IC2</td>
<td>FAN7390</td>
<td>FAN7391</td>
<td>SOP08</td>
<td>Fairchild Semiconductor</td>
<td>Gate driver</td>
</tr>
<tr>
<td>D4</td>
<td>1N4148 Diode</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Bypass circuit boot diode</td>
</tr>
<tr>
<td>C14</td>
<td>Ceramic 50V X7R</td>
<td>1nF / 1000pF</td>
<td>C0603</td>
<td>Murata</td>
<td>Bypass circuit shift capacitor</td>
</tr>
<tr>
<td>D5</td>
<td>1N4148 Diode</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Bypass circuit capacitor charging diode</td>
</tr>
<tr>
<td>Ref.</td>
<td>Value</td>
<td>Package</td>
<td>Manufacturer</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>---------</td>
<td>--------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>QHI3</td>
<td>1uF</td>
<td>C0603</td>
<td>Yageo</td>
<td>Input voltage divider filter</td>
<td></td>
</tr>
<tr>
<td>R39</td>
<td>1k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Gate driver supply voltage bridge</td>
<td></td>
</tr>
<tr>
<td>C28</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Output capacitor</td>
<td></td>
</tr>
<tr>
<td>R36</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Level shifting capacitor discharge resistor</td>
<td></td>
</tr>
<tr>
<td>C41</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Output capacitor</td>
<td></td>
</tr>
<tr>
<td>C42</td>
<td>10uF</td>
<td>C0805</td>
<td>Taiyo Yuden</td>
<td>Output capacitor</td>
<td></td>
</tr>
<tr>
<td>C44</td>
<td>10uF</td>
<td>C0603</td>
<td>Murata</td>
<td>Bypass circuit boot capacitor</td>
<td></td>
</tr>
<tr>
<td>R23</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Input voltage divider lower resistor</td>
<td></td>
</tr>
<tr>
<td>R35</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Level shifting voltage divider lower resistor</td>
<td></td>
</tr>
<tr>
<td>D13</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Level shifting diode discharge 3 signal</td>
<td></td>
</tr>
<tr>
<td>R43</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Level shifting resistor discharge 3 signal</td>
<td></td>
</tr>
<tr>
<td>C47</td>
<td>1uF</td>
<td>C0603</td>
<td>Murata</td>
<td>Level shifting capacitor discharge 3 signal</td>
<td></td>
</tr>
<tr>
<td>D9</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Level shifting diode PWM,LO,3 signal</td>
<td></td>
</tr>
<tr>
<td>R41</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Level shifting resistor PWM,LO,3 signal</td>
<td></td>
</tr>
<tr>
<td>C45</td>
<td>1uF</td>
<td>C0603</td>
<td>Murata</td>
<td>Level shifting capacitor PWM,LO,3 signal</td>
<td></td>
</tr>
<tr>
<td>D11</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Level shifting diode PWM,HL,3 signal</td>
<td></td>
</tr>
<tr>
<td>R42</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Level shifting resistor PWM,HL,3 signal</td>
<td></td>
</tr>
<tr>
<td>C46</td>
<td>1uF</td>
<td>C0603</td>
<td>Murata</td>
<td>Level shifting capacitor PWM,HL,3 signal</td>
<td></td>
</tr>
<tr>
<td>D15</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Level shifting diode CLK,3 signal</td>
<td></td>
</tr>
<tr>
<td>R48</td>
<td>10k</td>
<td>R0603</td>
<td>Yageo</td>
<td>Level shifting resistor CLK,3 signal</td>
<td></td>
</tr>
<tr>
<td>C49</td>
<td>1uF</td>
<td>C0603</td>
<td>Murata</td>
<td>Level shifting capacitor CLK,3 signal</td>
<td></td>
</tr>
<tr>
<td>D25</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Vh,3 voltage divider diode</td>
<td></td>
</tr>
<tr>
<td>D26</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>VL,3 voltage divider diode</td>
<td></td>
</tr>
</tbody>
</table>
### Table A.6: Component listing for converter 3 bottom side

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Type</th>
<th>Value</th>
<th>Package</th>
<th>Manufacturer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3</td>
<td>SER2000 Series</td>
<td>SER2000</td>
<td>SER2300</td>
<td>Coilcraft</td>
<td>Power Inductor</td>
</tr>
<tr>
<td>D14</td>
<td>1A Diode</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td></td>
<td>Boot diode for gate driver</td>
</tr>
<tr>
<td>R44</td>
<td>Resistor, 1%</td>
<td>R0603</td>
<td></td>
<td></td>
<td>Boot resistor for gate driver</td>
</tr>
<tr>
<td>C48</td>
<td>Ceramic 16V X5R</td>
<td>C0603</td>
<td>Yageo</td>
<td></td>
<td>Boot capacitor for gate driver</td>
</tr>
<tr>
<td>R28</td>
<td>Resistor, 1%</td>
<td>R0603</td>
<td>Yageo</td>
<td></td>
<td>VL3 voltage divider upper resistor</td>
</tr>
<tr>
<td>R44</td>
<td>Resistor, 1%</td>
<td>R0603</td>
<td>Yageo</td>
<td></td>
<td>VL3 voltage divider upper resistor</td>
</tr>
<tr>
<td>C38</td>
<td>Ceramic 16V X5R</td>
<td>C0603</td>
<td>Yageo</td>
<td></td>
<td>VL3 voltage divider filter</td>
</tr>
<tr>
<td>R24</td>
<td>Resistor, 1%</td>
<td>R0603</td>
<td>Yageo</td>
<td></td>
<td>VL3 voltage divider lower resistor</td>
</tr>
<tr>
<td>C50</td>
<td>Ceramic 16V X5R</td>
<td>C0603</td>
<td>Yageo</td>
<td>Gate driver Vdd capacitor</td>
<td></td>
</tr>
<tr>
<td>IC3</td>
<td>FAN7390</td>
<td>SOP08</td>
<td>Fairchild Semiconductor</td>
<td>Gate driver</td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>1N4148 Diode</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Bypass circuit boot diode</td>
</tr>
<tr>
<td>C43</td>
<td>Ceramic 50V X7R</td>
<td>C0603</td>
<td>Murata</td>
<td></td>
<td>Bypass circuit shift capacitor</td>
</tr>
<tr>
<td>D6</td>
<td>1N4148 Diode</td>
<td>1N4148</td>
<td>SOD323-W</td>
<td>Diodes Inc</td>
<td>Bypass circuit capacitor charging diode</td>
</tr>
</tbody>
</table>
APPENDIX B

MICROCONTROLLER C CODE FOR THE
SUB-MODULE MPPT

/*****************************************
// MPPT microcontroller code
// Marcel Schuck 2013
/*****************************************

#define F_CPU 8000000UL // 8MHz
#define UART_BAUD_RATE 38400 // Set baud rate
#define F_PLL 32000000 // PLL frequency 32MHz

/*****************************************
// Setup switching frequency
/*****************************************
#define F_SW 100000 // Switching frequency 100kHz

#define PERIOD_TICKS (int)(F_CPU/F_SW)

#define ZERO_PSC1 19974 // Timer offset to start PSC1
- value has to be adjusted experimentally
#define ZERO_PSC2 39956 // Timer offset to start PSC2
- value has to be adjusted experimentally

/* _____ INCLUDES _________________ */
#include <stdlib.h>
#include <stdbool.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <avr/pgmspace.h>
#include "compiler.h"
#include "mcu.h"
#include "pll_drv.h"
#include "uart.h"
#include "adc.h"
# define RESETB (int)(F_PLL/F_SW) // typecast!

// PSC minimal dead-time
#define SETA 0

// PSC 0
int reset0a = 160;
int set0b = 161;

// PSC 1
int reset1a = 160;
int set1b = 161;
int shift2;

// PSC 2
int reset2a = 160;
int set2b = 161;
int shift3;

bool clk_out; // activate/deactivate output of clk signal
bool bypassed = 0; // bypass mode status

// Variables for serial communication
unsigned int length = 100;
char Input[length];
unsigned int count;
unsigned int c;

/*F***************************************************************************/
* NAME: Set Duty cycle 1
******************************************************************************/
void Set_D1 (int D1)
{
    reset0a = (int)((float)D1/100*RESETB);
    set0b = reset0a + 1;

    // Write to power stage
    OCR0RAH = HIGH(reset0a);
    OCR0RAL = LOW(reset0a);
    OCR0SBH = HIGH(set0b);
    OCR0SBL = LOW(set0b);
    OCR0RBH = HIGH(RESETB);
}

/*F***************************************************************************/
* NAME: PSC0_Init
*********************************************************************/

void PSC0_Init (void)
{
    PSOC0 = (1<<POEN0A)|(1<<POEN0B);
    OCR0SAH = HIGH(SETA);
    OCR0SAL = LOW(SETA);
    OCR0RAH = HIGH(reset0a);
    OCR0RAL = LOW(reset0a);
    OCR0SBH = HIGH(set0b);
    OCR0SBL = LOW(set0b);
    OCR0RBH = HIGH(RESETB);
    OCR0RBL = LOW(RESETB);
    PCNF0 = (0<<PMODE01)|(1<<PCLKSELO)|(1<<POPO);
}

void PSC0_Run (void)
{
    PCTL0 = (1<<PRUN0); /* RUN !! */
}

void PSC0_Stop (void)
{
    PCTL0 = (0<<PRUN0); /* Stop !! */
}

/*F*********************************************************************/

* NAME: Set Duty cycle 2
*********************************************************************/

void Set_D2 (int D2)
{
    reset1a = (int)((float)D2/100*RESETB);
    set1b = reset1a + 1;
    // Write to power stage
    OCR1RAH = HIGH(reset1a);
    OCR1RAL = LOW(reset1a);
    OCR1SBH = HIGH(set1b);
    OCR1SBL = LOW(set1b);
    OCR1RBH = HIGH(RESETB);

    /*F*********************************************************************/

* NAME: Phase shift 2
*********************************************************************/

void Phaseshift_2 (int Angle2)
{
shift2 = (int)((float)Angle2/360*PERIOD_TICKS);

// Write to registers
OCR1AH = HIGH(ZERO_PSC1+shift2);
OCR1AL = LOW(ZERO_PSC1+shift2);
}

/*F*************************************************************************/
* NAME: PSC1 Init
***************************************************************************/
void PSC1_Init (void)
{
    PSOC1 = (1<POEN1A)|(1<POEN1B);
    OCR1SAH = HIGH(SETA);
    OCR1SAL = LOW(SETA);
    OCR1RAH = HIGH(reset1a);
    OCR1RAL = LOW(reset1a);
    OCR1SBH = HIGH(set1b);
    OCR1SBL = LOW(set1b);
    OCR1RBH = HIGH(RESETB);
    OCR1RBL = LOW(RESETB);
    PCNF1 = (0<<PMODE11)|(1<<PCLKSEL0)|(1<<POP1); */
    on PLL */
    PFRC1A = 0;
    PFRC1B = 0;
}

void PSC1_Run (void)
{
    PCTL1 = (1<<PRUN1); /* RUN !! */
}

void PSC1_Stop (void)
{
    PCTL1 = (0<<PRUN1); /* Stop !! */
}

/*F*************************************************************************/
* NAME: Set Duty cycle 3
***************************************************************************/
void Set_D3 (int D3)
{
    reset2a = (int)((float)D3/100*RESETB);
    set2b = reset2a + 1;
// Write to power stage
OCR2RAH = HIGH(reset2a);
OCR2RAL = LOW(reset2a);
OCR2SBH = HIGH(set2b);
OCR2SBL = LOW(set2b);
OCR2RBH = HIGH(RESETB);
}

/*F******************************************
* NAME: Phase shift 3
*************************************/
void Phaseshift_3 ( int Angle3 )
{
    shift3 = (int)((float)Angle3/360*PERIOD_TICKS);
    // Write to registers
    OCR1BH = HIGH(ZERO_PSC2+shift3);
    OCR1BL = LOW(ZERO_PSC2+shift3);
}

/*F******************************************
* NAME: PSC2 Init
*************************************/
void PSC2_Init ( void )
{
    PSOC2 = (1<<POEN2A)|(1<<POEN2B);
    OCR2SAH = HIGH(SETA);
    OCR2SAL = LOW(SETA);
    OCR2RAH = HIGH(reset2a);
    OCR2RAL = LOW(reset2a);
    OCR2SBH = HIGH(set2b);
    OCR2SBL = LOW(set2b);
    OCR2RBH = HIGH(RESETB);
    OCR2RBL = LOW(RESETB);
    PCNF2 = (0<<PMODE21)|(1<<PCLKSEL0)|(1<<PO2);
    PFRGC2A = 0;
    PFRGC2B = 0;
}

void PSC2_Run ( void )
{
    PCTL2 = (1<<PRUN2);
}

void PSC2_Stop ( void )
PCTL2 = (0<<PRUN2);

/*F******************************************************
* NAME: Functions to control bypass mode
*******************************************************/

void Bypass_On (void) {
    PSC0_Stop();
    PSC1_Stop();
    PSC2_Stop();
    _delay_us(5);
    clk_out = 1;
    bypassed = 1;
}

void Bypass_Off (void) {
    clk_out = 0;
    PORTD &= ~(1<<PD2);
    PORTC &= ~(1<<PC2);
    PORTB &= ~(1<<PB3);
    // Output discharge pulse
    PINC = (1<<PC1);
    PINC = (1<<PC3);
    PINB = (1<<PB4);
    _delay_us(2);
    PINC = (1<<PC1);
    PINC = (1<<PC3);
    PINB = (1<<PB4);
    _delay_us(5);
    // Start normal power stage operation again
    PSC0_Run();
    PSC1_Run();
    PSC2_Run();
    bypassed = 0;
}
unsigned int d1 = 1;
unsigned int dpeak1 = d1;
unsigned int vout_old1 = 0;
unsigned int vout1 = 0;

void Sweep1 (void) {
    while (d1 < 100) {
        Set_D1(d1);
        vout1 = adcRead(ADC_CH_ADC2);
        while (adcIsBusy());
        if (vout1 >= vout_old1) {
            vout_old1 = vout1;
            dpeak1 = d1;
        }
        d1 = d1 + 5;
    }
    Set_D1(dpeak1);
}

unsigned int d2 = 1;
unsigned int dpeak2 = d2;
unsigned int vout_old2 = 0;
unsigned int vout2 = 0;

void Sweep2 (void) {
    while (d2 < 100) {
        Set_D2(d2);
        vout2 = adcRead(ADC_CH_ADC8);
        while (adcIsBusy());
        if (vout2 >= vout_old2) {
            vout_old2 = vout2;
            dpeak2 = d2;
        }
    }
}
unsigned int d2 = 1;
unsigned int dpeak2 = d2;
unsigned int vout_old2 = 0;
unsigned int vout2 = 0;
void Sweep2 (void)
{
    while (d2 < 100)
    {
        Set_D2(dpeak2);
        d2 = d2 + 5;
    }
}

/*F***********************************************/
/* NAME: MPPT functions*****************************/
/*F***********************************************/
signed char dir1 = 1;

void MPP_Tracking1 (void)
{
    vout1 = adcRead(ADC_CH_ADC2);
    while (adcIsBusy());
    if (vout1 <= vout_old1)
    {
        dir1 = -dir1; // Perturb in other direction
    }
    vout_old1 = vout1;
}
d1 = d1 + dir1;
Set_D1(d1);
}

signed char dir2 = 1;

void MPP_Tracking2 (void)
{
    vout1 = adcRead(ADC_CH_ADC8);
    while (adcIsBusy());
    if (vout2 <= vout_old2)
    {
        dir2 = -dir2; // Perturb in other direction
    }
    vout_old2 = vout2;
    d2 = d2 + dir2;
    Set_D2(d2);
}

signed char dir3 = 1;

void MPP_Tracking3 (void)
{
    vout1 = adcRead(ADC_CH_ADC6);
    while (adcIsBusy());
    if (vout3 <= vout_old3)
    {
        dir3 = -dir3; // Perturb in other direction
    }
    vout_old3 = vout3;
    d3 = d3 + dir3;
    Set_D3(d3);
}

ISR(TIMER0_COMP_A_vect)
{
    if (clk_out == 1)
    {
        // ISR code
    }
}
PIND = (1<<PD2); // Toggle pin D2
PINC = (1<<PC2); // Toggle pin C2
PINB = (1<<PB3); // Toggle pin B3

ISR(TIMER1_COMPA_vect, ISR_NAKED)
{
    PSC1_Run(); // Start PSC1
    reti();
}

ISR(TIMER1_COMPB_vect, ISR_NAKED)
{
    PSC2_Run(); // Start PSC2
    reti();
}

ISR(TIMER1_OVF_vect, ISR_NAKED)
{
    TCCR1B &= ~(1<<CS10); // Remove clock signal - STOP Timer0
    reti();
}

/*F***************************************************/
* NAME : main
***************************************************/
int main (void)
{
    // Configure comparator
    AC1CON |= ((1<<AC1EN) | (1<<AC1IE) | (1<<AC1IS1) | (1<<AC1ISO) | (1<<AC1M2) | (0<<AC1M1) | (1<<AC1M0));

    // Configure DAC
    DACON |= ((0<<DAATE) | (1<<DALA) | (1<<DAOE) | (1<<DAEN));

    // Define outputs for bypass mode operation
    DDRC |= (1<<PORTC1); // Discharge signal output 1
    DDRC |= (1<<PORTC3); // Discharge signal output 2
DDRB |= (1<<PORTC4); // Discharge signal output 3

DDRD |= (1<<PORTD2); // Set PD2 as output for clk
DDRC |= (1<<PORTC2); // Set PC2 as output for clk
DDRB |= (1<<PORTB3); // Set PB3 as output for clk

// Initialize UART
uart_init(UART_BAUD_SELECT(UART_BAUD_RATE,F_CPU));

// Setup ADC
adcInit(00, 10);
adcStart(ADC_CH_ADC2);
adcStart(ADC_CH_ADC6);
adcStart(ADC_CH_ADC8);

// For accuracy of timing in starting the other power stages, D1, D2, and D3 must be defined in advance to NOT interfere with interrupt action.

// Set duty cycle D1 here
Set_D1(50);

// Set duty cycle D2 here
Set_D2(50);

// Set duty cycle D3 here
Set_D3(50);

// Set phase-shift of converter 2 in degree here
Phaseshift_2(120);

// Set phase-shift of converter 3 in degree here
Phaseshift_3(240);

Start_pll_32_mega();
Wait_pll_ready();
PSC0_Init();
PSC1_Init();
PSC2_Init();

TIMSK1 |= ((1<<OCIE1A) | (1<<OCIE1B) | (1<<TOIE1));  // Enable Interrupts

// Setup Timer0 for generation of clk signal for bypass circuits
TCCR0A |= (1<<WGM01);  // Set mode of operation to Clear Timer on Compare Match (CTC)
OCR0A = 39;  // Output compare match, equals a 100 kHz output at f_CPU = 8Mhz
TIMSK0 |= (1<<OCIE0A);  // Enable compare match A interrupt

// Enable Interrupts globally
sei();

TCCR1B |= (1<<CS10);  // Clock IO without prescaler
   - START Timer1

PSC0_Run();  // Start PSC0
// Other PSCs are started in interrupt routines

_delay_ms(20);  // wait until power stages are running

// Perform initial coarse sweep to find the region of the
  // MPPs for each sub-module.
Sweep1();
Sweep2();
Sweep3();

for (;;)
{
  // Receive data from Python script
  // Receiving is based on interrupts and a ringbuffer.
  // Here, the data is read out of the ringbuffer.
  c = uart_getc();

  // Skip if no data has been received
if ( !(c & UART_NO_DATA) ) // there is new data
{
count = 0;

while ( (char) c != '\r' ) // until return
{
    if (count < length && !(c & UART_NO_DATA))
    {
        Input[count++] = (char) c;
    }
    c = uart_getc();
}
Input[count] = '\0';
// Set same duty ratios to power stages
Set_D1(atoi(Input));
PSC0_Init();
PSC0_Run();

Set_D2(atoi(Input));
PSC1_Init();
PSC1_Run();

Set_D3(atoi(Input));
PSC2_Init();
PSC2_Run();

uart_puts( "Received: \"" ); // Return input
uart_puts( Input );
uart_puts( "\\" );
}

// Perform MPP tracking
MPP_Tracking1();
MPP_Tracking2();
MPP_Tracking3();

// Check for bypass conditions
// If all three duty ratios are within a certain range to each other...
if (d2 <= d1+2 && d2 >= d1-2 && d3 <= d1+2 && d3 >= d1-2)
{
    // ...activate bypass mode
Bypass_On();

while (bypassed == 1)
{
    // Sample output voltages continuously
    vout1 = adcRead(ADC_CH_ADC6);
    while (adcIsBusy());
    vout2 = adcRead(ADC_CH_ADC8);
    while (adcIsBusy());
    vout3 = adcRead(ADC_CH_ADC6);
    while (adcIsBusy());

    // If all three output voltages are not within a certain range to each other...
    if (! (vout2 <= vout1+10 && vout2 >= vout1-10 && vout3 <= vout1+10 && vout3 >= vout1-10))
    {
        // ...deactivate bypass mode
        Bypass_Off();
    }
}
APPENDIX C

PYTHON CONTROL CODE FOR THE EFFICIENCY MEASUREMENT SETUP

```python
#!/bin/env python

# ==============================================================
# Python script for efficiency measurements with fixed
# input voltage, variable current,
# and variable duty ratio.
# Communication with converter via serial port.
#
# Robert Pilawa-Podgurski, Marcel Schuck
# ==============================================================

# Set input voltage for given measurement
input_voltage = 10.0

# List of all output current measurement points
output_currents = np.arange(0, 6, 1)

# List of all duty ratios
duty_ratios = np.arange(0.1, 1, 0.1)

# ==============================================================
# IMPORTS
# ==============================================================
import sys
import time
import datetime
from array import array
from PILawa_instruments import *
import os
import glob
import numpy as np
import serial

# ==============================================================
# MAIN PROGRAM
# ==============================================================
```
delimiter = ', '

t = time.strftime('%Y%m%d')
current_directory = os.getcwd()
print "Current directory: %s" % current_directory
foldername = current_directory + "/data/" + t
print foldername
if not os.path.exists(foldername):
    os.makedirs(foldername)

# Initialize serial communication to the converter
ser = serial.Serial(port = '/dev/ttyS0', baudrate = 38400)
# Initialize GPIB adapter, Prologix GPIB-USB controller
gpib = prologix_serial(port = '/dev/ttyUSB0', baudrate = 115200, debug = False, timeout = 5)
# Initialize electronic load, HP6060B
eload = prologix_6060b(prologix = gpib, addr = 11, mode = "VOLT", rang = "20", debug = False)
# Initialize power supply, HP 6632A
power_supply_1 = prologix_6632a(prologix = gpib, addr = 5, debug = False)

# Meter for input voltage, Agilent 34410A
meter1 = prologix_34401a(prologix = gpib, addr = 12, mode = "VOLT", maxrange = "25", nplc = "2", debug = False)
# Meter for input current, Agilent 34410A using Agilent 34330A shunt resistor with 1mV/A
meter2 = prologix_34401a(prologix = gpib, addr = 13, mode = "VOLT", maxrange = "1", nplc = "2", debug = False)
# Meter for output voltage, Agilent 34410A
meter3 = prologix_34401a(prologix = gpib, addr = 14, mode = "VOLT", maxrange = "25", nplc = "2", debug = False)
# Meter for output current, Agilent 34410A using Agilent 34330A shunt resistor with 1mV/A
meter4 = prologix_34401a(prologix = gpib, addr = 15, mode = "VOLT", maxrange = "1", nplc = "2", debug = False)
meterList = [meter1, meter2, meter3, meter4]
meter_addrList = [12, 13, 14, 15]

# Scale voltage readings through shunt resistors
scaleList = [1, 1000, 1, 1000]

# Setup power supply
power_supply_1.setVoltage(input_voltage)
```python
time.sleep(0.2)
power_supply_1.setCurrent(5)
time.sleep(0.2)
power_supply_1.activate()
time.sleep(0.2)

# Setup electronic load
eload.setMode("CURR")
eload.setSlew(2000000)
eload.setVoltage(0.0)
time.sleep(0.2)

header_string="("Iref","D","Vin","Iin","Vout","Iout"
,"Pin","Pout","Efficiency")"

t2=time.strftime('%H%M%S')
filename = "efficiency_input_voltage_%.0fV_" % (input_voltage) + t2
f=open(foldername + "/" + filename,"w")
timestamp = time.strftime('%Y%m%d%H%M%S')
f.write(delimiter.join(header_string))
f.write('
')

# Take measurements for all values in the predefined output current list
for current in output_currents:
eload.setValue(current)
time.sleep(1)

# Take measurements for all values in the predefined duty ratio list
for ratio in duty_ratios:
    # Set converter duty ratio
    ser.write(str(ratio*100)+'')
time.sleep(2)
    for x in meterList:
        x.waitForTrigger()
time.sleep(0.2)

    print ("current: %s" % current)
    print ("duty ratio: %s" % ratio)

    # Trigger all meters at the same time
gpib.trigger_devices(meter_addrList)

meterdata = []
for meter,scale in zip(meterList, scaleList):
data = meter.readData()
    try:
```

132
scaled_value=float(data)*scale

except:
    scaled_value=0

meterdata.append(scaled_value)

# Calculate input power
pin = meterdata[0]*meterdata[1]

# Calculate output power
pout = meterdata[2]*meterdata[3]

print("%s %s %s %s"%(meterdata[0],
                   meterdata[1],meterdata[2],meterdata[3]))

try:
    # Calculate efficiency
    eff = pout/pin
except:
    eff = 0

if eff>1: eff=1
if eff<0: eff=0

print "Output current: %s, Duty Ratio: %s, Pin: %.5f, Pout: %.5f, Eff: %.5f" % (current,
                                                                                            ratio, pin, pout, eff)
f.write("%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s
"
                          % (str(current), delimiter, str(ratio),
                             delimiter, str(meterdata[0]), delimiter,
                             str(meterdata[1]), delimiter, str(meterdata
                             [2]), delimiter, str(meterdata[3]),
                             delimiter, str(pin),delimiter, str(pout),
                             delimiter, str(eff)))

f.close()

# Deactivate power supply and electronic load
eload.setMode("CURR")
eload.setValue(0.00)
power_supply_1.deactivate()
time.sleep(1)
%%% Calculation of phase angles and detailed visualization of the
%%% proposed ripple cancellation technique using uneven phase-angles
%%% Marcel Schuck 2013

close all;
clear;
clc;

grey = [0.6,0.6,0.6];

%%% Inputs

%%% Define input voltages
V_IN1=14;
V_IN2=12;
V_IN3=10;

%%% Common average output current [A]
I_out_avg = 0;

%%% Inductor value [H]
L = 4.7e-6;

%%% Define duty cycles
D_1=0.6;
D_2=0.7;
D_3=0.8;

f_sw = 100000;
omega = 2*pi*f_sw;
T_sw = 1/f_sw;
t = [0:T_sw/1000:5*T_sw];
\[ \Delta_i_1 = \frac{V_{IN1}}{L} \left( D_1 - (D_1)^2 \right) T_{sw}; \]
\[ \Delta_i_2 = \frac{V_{IN2}}{L} \left( D_2 - (D_2)^2 \right) T_{sw}; \]
\[ \Delta_i_3 = \frac{V_{IN3}}{L} \left( D_3 - (D_3)^2 \right) T_{sw}; \]
\[ D_{ref} = \frac{(D_1 + D_2 + D_3)}{3}; \]
% Average current ripple for reference system
\[ \Delta_i_{ref} = \frac{\Delta_i_1 + \Delta_i_2 + \Delta_i_3}{3}; \]
\[ a_{10_{ref}} = I_{out\_avg}; \]
\[ a_{20} = I_{out\_avg}; \]
\[ a_{30} = I_{out\_avg}; \]
\[ f_1 = a_{10}; \% Fourier series cartesian form \]
\[ g_1 = a_{10}; \]
\[ f_2 = a_{20}; \% Fourier series cartesian form \]
\[ g_2 = a_{20}; \]
\[ f_3 = a_{30}; \% Fourier series cartesian form \]
\[ g_3 = a_{30}; \]
% Coefficient number
\[ k = 1; \]
% Reference system
\[ a_{k_{ref}} = 0.5 \frac{\Delta_i_{ref}}{((\pi k)^2)} \left( \frac{1}{D_{ref}} + \frac{1}{1 - D_{ref}} \right) \left( \cos(2 \pi k D_{ref}) - 1 \right); \]
\[ b_{k_{ref}} = 0.5 \frac{\Delta_i_{ref}}{((\pi k)^2)} \left( \frac{1}{D_{ref}} + \frac{1}{1 - D_{ref}} \right) \sin(2 \pi k D_{ref}); \]
\[ A_{k_{ref}} = \sqrt{a_{k_{ref}}^2 + b_{k_{ref}}^2}; \]
\[ \phi_{k_{ref}} = \arctan(b_{k_{ref}}/a_{k_{ref}}); \]
\[ if(\phi_{k_{ref}} < 0) \]
\[ \phi_{k_{ref}} = \phi_{k_{ref}} + 2 \pi; \]
% Phase 1
\[ a_{1k} = 0.5 \frac{\Delta_i_1}{((\pi k)^2)} \left( \frac{1}{D_1} + \frac{1}{1/(1-D_1)} \right) \left( \cos(2 \pi k D_1) - 1 \right); \]
\[ b_{1k} = 0.5 \frac{\Delta_i_1}{((\pi k)^2)} \left( \frac{1}{D_1} + \frac{1}{1/(1-D_1)} \right) \sin(2 \pi k D_1); \]
\[ A_{1k} = \sqrt{a_{1k}^2 + b_{1k}^2}; \]
\[ \phi_{1k} = \arctan(b_{1k}/a_{1k}); \]
% phi_1k = 2*atan(b_1k/(A_1k+a_1k))
if (phi_1k < 0)
    phi_1k = phi_1k + 2*pi;
end

%%% Phase 2
a_2k = 0.5*Delta_i2/((pi*k)^2) * ( (1/D_2 + 1/(1-D_2)) * (cos(2*pi*k*D_2) - 1 ) )
b_2k = 0.5*Delta_i2/((pi*k)^2) * (1/D_2 + 1/(1-D_2)) * sin(2*pi*k*D_2);
A_2k = sqrt(a_2k^2 + b_2k^2);
phi_2k = atan2(b_2k, a_2k);
if(phi_2k < 0)
    phi_2k = phi_2k + 2*pi;
end

%%% Phase 3
a_3k = 0.5*Delta_i3/((pi*k)^2) * ( (1/D_3 + 1/(1-D_3)) * (cos(2*pi*k*D_3) - 1 ) )
b_3k = 0.5*Delta_i3/((pi*k)^2) * (1/D_3 + 1/(1-D_3)) * sin(2*pi*k*D_3);
A_3k = sqrt(a_3k^2 + b_3k^2);
phi_3k = 2*atan(b_3k/(A_3k+a_3k));
if(phi_3k < 0)
    phi_3k = phi_3k + 2*pi;
end

%%% Calculate angles
Phi_01 = -phi_1k/k;
Phi_02 = 1/k*( acos(0.5*(A_3k^2-A_2k^2-A_1k^2)/(A_1k*A_2k)) - phi_2k );
Phi_03 = 1/k*( 2*pi - acos(0.5*(A_2k^2-A_1k^2-A_3k^2)/(A_1k*A_3k)) - phi_3k );

 Phi_01 = -phi_1k/k;
 Phi_02 = 1/k*( acos( 1/2 * ((Delta_i3^2/(D_3^2*(1-D_3)^2)))*sin(pi*k*D_3)^2 - (Delta_i2^2/(D_2^2*(1-D_2)^2)))*sin(pi*k*D_2)^2 - (Delta_i1^2/(D_1^2*(1-D_1)^2)))*sin(pi*k*D_1)^2 ) / (((Delta_i1/(D_1*(1-D_1)))*sin(pi*k*D_1))*(Delta_i2/(D_2*(1-D_2)))*sin(pi*k*D_2)) - phi_2k );
 Phi_03 = 1/k*( 2*pi - acos( 1/2 * ((Delta_i2^2/(D_2^2*(1-D_2)^2)))*sin(pi*k*D_2)^2 - (Delta_i1^2/(D_1^2*(1-D_1)^2)))*sin(pi*k*D_1)^2 - (Delta_i3^2/(D_3^2*(1-D_3)^2)))*sin(pi*k*D_3)^2 - (Delta_i1^2/(D_1*(1-D_1)))*sin(pi*k*D_1)) / (((Delta_i1/(D_1*(1-D_1)))*sin(pi*k*D_1))*(Delta_i3/(D_3*(1-D_3)))^2) - phi_3k );
\[ \binom{2*(1-D_3)^2)*sin(pi*k*D_3)^2) / ((Delta_i1/(D_1 * (1-D_1)))*sin(pi*k*D_1))*(Delta_i3/(D_3*(1-D_3))) *sin(pi*k*D_3)) } - phi_3k) \]

%%% Rotate entire system if desired
Phi_01_old = 0;
Phi_01 = Phi_01 - Phi_01_old;
Phi_02 = Phi_02 - Phi_01_old;
Phi_03 = Phi_03 - Phi_01_old;

Theta_1 = phi_1k + k*Phi_01;
Theta_2 = phi_2k + k*Phi_02;
Theta_3 = phi_3k + k*Phi_03;

%%% Even phase angles for reference
% Theta_1 = 0;
% Theta_2 = 2/3*pi;
% Theta_3 = 4/3*pi;

Theta_1_deg = radtodeg (Theta_1)
Theta_2_deg = radtodeg (Theta_2)
Theta_3_deg = radtodeg (Theta_3)

%%% Define complex phasors
Phase1k = A_1k * exp(i*Theta_1);
Phase2k = A_2k * exp(i*Theta_2);
Phase3k = A_3k * exp(i*Theta_3);

Phi_01_deg = radtodeg (Phi_01)
Phi_02_deg = radtodeg (Phi_02)
Phi_03_deg = radtodeg (Phi_03)

%%% Define complex numbers for reference system
Phase1k_ref = A_k_ref * exp(i*(0 - Phi_01_old));
Phase2k_ref = A_k_ref * exp(i*((2/3)*pi - Phi_01_old));
Phase3k_ref = A_k_ref * exp(i*((4/3)*pi - Phi_01_old));

%%% Plot corresponding arrows in complex plane
figure ('Color', 'w')
set (gcf, 'PaperUnits', 'centimeters');
set (gcf, 'PaperSize', [15 15]);
axis([-6 10 -6 6], 'equal')
set(gca, 'FontSize', 14, 'FontWeight', 'Bold', 'LineWidth' ,2, 'xlim', [-2,3], 'ylim', [-2,2], 'position', [0.12 0.13 0.85 0.85]);
\begin{verbatim}
155 [hx,hy] = format_ticks(gca,{'$-2$','$-1$','$0$','$1$','$2$','$3$'},{'$-2$','$-1$','$0$','$1$','$2$'},[-2, -1, 0, 1, 2, 3],[-2, -1, 0, 1, 2],0,0,[],[],
   'FontSize',14,'FontWeight','Bold');
156 h = xlabel('Real Part','Interpreter','latex');
157 pos = get(h,'pos'); % Read position [x y z]
158 set(h,'pos',[0 -2.4])
159 h = ylabel('Imaginary Part','Interpreter','latex');
160 pos = get(h,'pos'); % Read position [x y z]
161 set(h,'pos',[-2.5 0])
162 grid on;
163 hold on;

164 \textbf{%%% Plot actual system}
165 handlevector(2) = arrow([0 0],[real(Phase1k) imag(Phase1k)],'Width',2.5,'EdgeColor','b','FaceColor','b');
166 handlevector(3) = arrow([0 0],[real(Phase2k) imag(Phase2k)],'Width',2.5,'EdgeColor','g','FaceColor','g');
167 handlevector(4) = arrow([0 0],[real(Phase3k) imag(Phase3k)],'Width',2.5,'EdgeColor','r','FaceColor','r');
168 handlevector(5) = arrow([0 0],[real(Phase1k+Phase2k+Phase3k) imag(Phase1k+Phase2k+Phase3k)],'Width',2.5,'EdgeColor','k','FaceColor','k');

169 \textbf{%%% Plot symmetric reference system}
170 plot([0 real(Phase1k_ref)], [0 imag(Phase1k_ref)],'LineWidth',2,'Color',grey);
171 plot([0 real(Phase2k_ref)], [0 imag(Phase2k_ref)],'LineWidth',2,'Color',grey);
172 plot([0 real(Phase3k_ref)], [0 imag(Phase3k_ref)],'LineWidth',2,'Color',grey);

173 real_sum = real(Phase1k) + real(Phase2k) + real(Phase3k)
174 imaginary_sum = imag(Phase1k) + imag(Phase2k) + imag(Phase3k)

175 \textbf{%%% Plot inductor ripple component without average value}
176 \textbf{%%% Currents with asymmetric phase shift}
177 I_1 = (Delta_i1/2)*sawtooth(2*pi*f_sw*t-Phi_01,D_1);
\end{verbatim}
\[ I_2 = \left( \frac{\Delta_i}{2} \right) \text{sawtooth}(2\pi f_{sw} t - \Phi_{02}, D_2); \]
\[ I_3 = \left( \frac{\Delta_i}{2} \right) \text{sawtooth}(2\pi f_{sw} t - \Phi_{03}, D_3); \]
\[ I_{\text{sum}} = I_1 + I_2 + I_3; \]
\[ \Delta I_{\text{sum}} = \max(I_{\text{sum}}) - \min(I_{\text{sum}}) \]

%%% Currents with even phase shift
\[ I_{1\text{even}} = \left( \frac{\Delta_i}{2} \right) \text{sawtooth}(2\pi f_{sw} t - 0, D_1); \]
\[ I_{2\text{even}} = \left( \frac{\Delta_i}{2} \right) \text{sawtooth}(2\pi f_{sw} t - (2/3)\pi, D_2); \]
\[ I_{3\text{even}} = \left( \frac{\Delta_i}{2} \right) \text{sawtooth}(2\pi f_{sw} t - (4/3)\pi, D_3); \]
\[ I_{\text{sum\_even}} = I_{1\text{even}} + I_{2\text{even}} + I_{3\text{even}}; \]
\[ \Delta I_{\text{sum\_even}} = \max(I_{\text{sum\_even}}) - \min(I_{\text{sum\_even}}) \]

figure('Color', 'w','units','centimeter','position',[0 17 15])

p1 = subplot(3,1,1)
set(gca,'FontSize',10,'LineWidth',2,'xlim',[0,4*T_sw], 'ylim', [-4,4]);
grid on;
hold on;
handlevector(1) = plot(t,I_1,'b','LineWidth',2);
[hx,hy] = format_ticks(gca,{'$0$','$1 \times 10^{-5}$','$2 \times 10^{-5}$','$3 \times 10^{-5}$','$4 \times 10^{-5}$'},
{'-4$','-2$','0$','2$','4$'},[0,1e-5,2e-5,3e-5,4e-5],[-4,-2,0,2,4],0,0,[],[],'Fontsize',14,'FontWeight','Bold');
h = ylabel('$i_1(t)$ / A', 'Interpreter','latex','FontSize',14,'FontWeight','Bold');
pos = get(h,'pos'); % Read position [x y z]
set(h,'pos',[-0.3e-5 0])

h_legend=legend(handlevector(1),'{$\Delta i_{\{1\}}=3.58$ A, $D_{\{1\}}=0.6$,\phi_{\{01\}}=0^\circ}', 'Interpreter','latex');
set(h_legend,'FontSize',14,'FontWeight','Bold');

p2 = subplot(3,1,2)
set(gca,'FontSize',10,'LineWidth',2,'xlim',[0,4*T_sw], 'ylim', [-4,4]);
grid on;
hold on;
handlevector(2) = plot(t,I_2,'g','LineWidth',2);
\[
\begin{align*}
220 & \text{[hx,hy] = format_ticks(gca,\{'$0$','$1 \times 10^{-5}$', '}$2 \times 10^{-5}$','$3 \times 10^{-5}$','$4 \times 10^{-5}$'}\},\{'$-4$','$-2$','$0$','$2$','$4$'\},[0, 1e-5, 2e-5, 3e-5, 4e-5],[-4, -2, 0, 2, 4]),0,0,[],[],'FontSize',14,'FontWeight','Bold');
221 & \text{h = ylabel('$i_2(t)$ / A',' Interpreter','latex',}
222 & \text{'FontSize',14,'FontWeight','Bold');}
223 & \text{pos = get(h,'pos'); \% Read position [x y z]}
224 & \text{set(h,'pos',[-0.3e-5 0])}
225 & \text{h_legend=legend(handlevector(2),\{'$\Delta i_2=2.68$ A, $D_2=0.7, \phi_02=138.4^\circ$\}, '}
226 & \text{ Interpreter','latex');}
227 & \text{set(h_legend,'FontSize',12,'FontWeight','Bold');}
228 & \text{p3 = subplot(3,1,3)
229 & \text{set(gca,'FontSize',10,'LineWidth',2,'xlim', [0,4*T_sw ], 'ylim', [-4,4]);
230 & \text{grid on; hold on;}
231 & \text{handlevector(3) = plot(t,I_3,'r','LineWidth',2);
232 & \text{[hx,hy] = format_ticks(gca,\{'$0$','$1 \times 10^{-5}$', '}$2 \times 10^{-5}$','$3 \times 10^{-5}$','$4 \times 10^{-5}$'}\},\{'$-4$','$-2$','$0$','$2$','$4$'\},[0, 1e-5, 2e-5, 3e-5, 4e-5],[-4, -2, 0, 2, 4]),0,0,[],[],'FontSize',14,'FontWeight','Bold');
233 & \text{h = xlabel('Time / s',' Interpreter','latex','FontSize'
234 & \text{,14,'FontWeight','Bold');}
235 & \text{pos = get(h,'pos'); \% Read position [x y z]
236 & \text{set(h,'pos',[2e-5 -5.5])}
237 & \text{h = ylabel('$i_3(t)$ / A',' Interpreter','latex','}
238 & \text{FontSize',14,'FontWeight','Bold');}
239 & \text{pos = get(h,'pos'); \% Read position [x y z]
240 & \text{set(h,'pos',[-0.3e-5 0])}
241 & \text{h_legend=legend(handlevector(3),\{'$\Delta i_3=1.70$ A, $D_3=0.8, \phi_03=185.3^\circ$\}, '}
242 & \text{ Interpreter','latex');}
243 & \text{set(h_legend,'FontSize',12,'FontWeight','Bold');}
244
\%
245 & \text{%%% Plot fundamental component of ripple waveform}
246 & \text{f_1 = f_1 + a_1k*cos(k*omega*t-k*Phi_01) + b_1k*sin(k*omega*t-k*Phi_01);}
247 & \text{g_1 = g_1 + A_1k*cos(k*omega*t-k*Phi_01-phi_1k);
248 & \text{f_2 = f_2 + a_2k*cos(k*omega*t-k*Phi_02) + b_2k*sin(k*omega*t-k*Phi_02);}
\end{align*}
\]
\[
g_2 = g_2 + A_2 k \cos (k \omega t - k \Phi_02 - \phi_2 k);
\]

\[
f_3 = f_3 + a_3 k \cos (k \omega t - k \Phi_03) + b_3 k \sin (k \omega t - k \Phi_03);
\]

\[
g_3 = g_3 + A_3 k \cos (k \omega t - k \Phi_03 - \phi_3 k);
\]

```python
figure('Color', 'w', 'units', 'centimeter', 'position', [0 0 17 20])
p1 = subplot(4,1,1)
set(gca, 'FontSize', 10, 'LineWidth', 2, 'xlim', [0, 4*T_sw ], 'ylim', [-4, 4]);
grid on;
hold on;
handlevector(1) = plot(t, g_1, 'b', 'LineWidth', 2);
[hx, hy] = format_ticks(gca, {='$0$', '$1 \times 10^{-5}$', '$2 \times 10^{-5}$', '$3 \times 10^{-5}$', '$4 \times 10^{-5}$'}, {='$-4$','$-2$','$0$','$2$','$4$'}, [0, 1e-5, 2e-5, 3e-5, 4e-5, [-4, -2, 0, 2, 4], 0, 0, [], [], 'FontSize', 14, 'FontWeight', 'Bold');
h = ylabel('$i_{11}(t)$ / A', 'Interpreter', 'latex', 'FontSize', 14, 'FontWeight', 'Bold');
pos = get(h, 'pos'); % Read position [x y z]
set(h, 'pos', [-0.4e-5 0])
% h_legend = legend(handlevector(1), {'$\Delta i_{11}=3.58$ A, $D_{1}=0.6, \phi_{01}=0^\circ$'}, 'Interpreter', 'latex');
set(h_legend, 'FontSize', 12, 'FontWeight', 'Bold');
p2 = subplot(4,1,2)
set(gca, 'FontSize', 10, 'LineWidth', 2, 'xlim', [0, 4*T_sw ], 'ylim', [-4, 4]);
grid on;
hold on;
handlevector(2) = plot(t, g_2, 'g', 'LineWidth', 2);
[hx, hy] = format_ticks(gca, {='$0$', '$1 \times 10^{-5}$', '$2 \times 10^{-5}$', '$3 \times 10^{-5}$', '$4 \times 10^{-5}$'}, {='$-4$','$-2$','$0$','$2$','$4$'}, [0, 1e-5, 2e-5, 3e-5, 4e-5, [-4, -2, 0, 2, 4], 0, 0, [], [], 'FontSize', 14, 'FontWeight', 'Bold');
h = ylabel('$i_{21}(t)$ / A', 'Interpreter', 'latex', 'FontSize', 14, 'FontWeight', 'Bold');
pos = get(h, 'pos'); % Read position [x y z]
set(h, 'pos', [-0.4e-5 0])
% h_legend = legend(handlevector(2), {'$\Delta i_{2}=2.68$ A, $D_{2}=0.7, \phi_{02}=156.4^\circ$'}, 'Interpreter', 'latex');
set(h_legend, 'FontSize', 12, 'FontWeight', 'Bold');
```

Interpreter', 'latex');
set(h_legend,'FontSize',12,'FontWeight','Bold');
p3 = subplot(4,1,3)
set(gca,'FontSize',10,'LineWidth',2,'xlim',[0,4*T_sw],
'ylim',[-4,4]);
grid on;
hold on;
handlevector(3) = plot(t,g_3,'r','LineWidth',2);
[hx,hy] = format_ticks(gca,{'$0$','$1 \times 10^{-5}$','$2 \times 10^{-5}$','$3 \times 10^{-5}$','$4 \times 10^{-5}$'},
{'$-4$','$-2$','$0$','$2$','$4$'},[0, 1e-5, 2e-5, 3e-5, 4e-5],
[-4, -2, 0, 2, 4],0,0,[],[],'FontSize',14,'FontWeight','Bold');
h = ylabel('$i_{31}(t)$ / A','Interpreter','latex','FontSize',14,'FontWeight','Bold');
pos = get(h,'pos '); % Read position [x y z]
set(h,'pos',[-0.4e-5 0])
% h_legend=legend(handlevector(3),{'$\Delta i_{3}=1.70$ A, $D_{3}=0.8, \phi_{03}=221.3^\circ$'},
',Interpreter','latex');
set(h_legend,'FontSize',12,'FontWeight','Bold');
p4 = subplot(4,1,4)
set(gca,'FontSize',10,'LineWidth',2,'xlim',[0,4*T_sw],
'ylim',[-0.5,0.5]);
grid on;
hold on;
handlevector(3) = plot(t,(g_1+g_2+g_3),'k','LineWidth',2);
[hx,hy] = format_ticks(gca,{'$0$','$1 \times 10^{-5}$','$2 \times 10^{-5}$','$3 \times 10^{-5}$','$4 \times 10^{-5}$'},
{'-0.5$','0$','0.5$'},[0, 1e-5, 2e-5, 3e-5, 4e-5],[-0.5, 0, 0.5],0,0,[],[],'FontSize',14,'FontWeight','Bold');
h = xlabel('Time / s','Interpreter','latex','FontSize',14,'FontWeight','Bold');
pos = get(h,'pos '); % Read position [x y z]
set(h,'pos',[2e-5 -0.75])
h = ylabel('$i_{\text{sum}1}(t)$ / A','Interpreter','latex','FontSize',14,'FontWeight','Bold');
pos = get(h,'pos '); % Read position [x y z]
set(h,'pos',[-0.4e-5 0])
% h_legend=legend(handlevector(3),{'$\Delta i_{3}=1.70$ A, $D_{3}=0.8, \phi_{03}=221.3^\circ$'},
',Interpreter','latex');
set(h_legend,'FontSize',12,'FontWeight','Bold');
set (p1 ,' Position ' ,[.12 .82 .82 .17])
set (p2 ,' Position ' ,[.12 .57 .82 .17])
set (p3 ,' Position ' ,[.12 .32 .82 .17])
set (p4 ,' Position ' ,[.12 .08 .82 .17])
hold off

%%% Reinitialize
a_10 = I_out_avg;
a_20 = I_out_avg;
a_30 = I_out_avg;
f_1 = a_10 ; % Fourier series cartesian form
g_1 = a_10 ;
f_2 = a_20 ; % Fourier series cartesian form
g_2 = a_20 ;
f_3 = a_30 ; % Fourier series cartesian form
g_3 = a_30 ;
sum = a_10 + a_20 + a_30;
sum_sym = a_10 + a_20 + a_30;

%%% Plot spectrum of each ripple waveform
figure (10);
set (10 , 'Color ' , 'w','units ' , 'centimeter ',' position ' ,[0
0 17 15])
p1 = subplot (3 ,1 ,1)
set (gca ,' FontSize ' ,10, ' LineWidth ' ,2, 'xlim ', [0 , 10] ,
'ylim ', [0 , 3]) ;
grid on;
hold on;
plot (0 ,0 , 'o',' MarkerEdgeColor ','b',' MarkerFaceColor ','
'b','MarkerSize',5)
[hx ,hy] = format_ticks (gca ,{ '$0$ ','$1$ ','$2$ ','$3$ ',
' $4$ ','$5$ ','$6$ ','$7$ ','$8$ ','$9$ ','$10$ '},{ '$0$ ',
' $1$ ','$2$ ','$3$ '},[0 , 1, 2, 3, 4, 5, 6, 7, 8, 9, 10], [0, 1, 2, 3],0,0,[0.1],[], 'FontSize' ,14,' FontWeight' , 'Bold');
h = ylabel ('$A _ {1k}$',' Interpreter ','latex ',' FontSize '
hpos = get (h,'pos ');

% Read position [x y z]
```matlab
set(h,'pos',[ -0.75 1.5])
p2 = subplot(3,1,2)
set(gca,'FontSize',10,'LineWidth',2, 'xlim', [0, 20],
'ylim', [0, 3]);
grid on;
hold on;
plot(0,0,'o','MarkerEdgeColor','g','MarkerFaceColor','g','MarkerSize',5)
[hx,hy] = format_ticks(gca,{$0$,$1$,$2$,$3$,$4$,$5$,$6$,$7$,$8$,$9$,$10$,$11$,$12$,$13$,$14$,$15$,$16$,$17$,$18$,$19$,$20$},{$0$,$1$,$2$,$3$},[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20],[0, 1, 2, 3],0,0,[0,1],[],'FontSize',14,'FontWeight','Bold');
h = ylabel('A_{2k}','Interpreter','latex','FontSize',14,'FontWeight','Bold');
pos = get(h,'pos'); % Read position [x y z]
set(h,'pos',[-0.75 1.5])

p3 = subplot(3,1,3)
set(gca,'FontSize',10,'LineWidth',2, 'xlim', [0, 20],
'ylim', [0, 3]);
grid on;
hold on;
plot(0,0,'o','MarkerEdgeColor','r','MarkerFaceColor','r','MarkerSize',5)
[hx,hy] = format_ticks(gca,{$0$,$1$,$2$,$3$,$4$,$5$,$6$,$7$,$8$,$9$,$10$,$11$,$12$,$13$,$14$,$15$,$16$,$17$,$18$,$19$,$20$},{$0$,$1$,$2$,$3$},[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20],[0, 1, 2, 3],0,0,[0,1],[],'FontSize',14,'FontWeight','Bold');
h = xlabel('$k$','Interpreter','latex','FontSize',14,'FontWeight','Bold');
pos = get(h,'pos'); % Read position [x y z]
set(h,'pos',[5 -0.85])
h = ylabel('A_{3k}','Interpreter','latex','FontSize',14,'FontWeight','Bold');
pos = get(h,'pos'); % Read position [x y z]
set(h,'pos',[-0.75 1.5])
```

144
%% Plot summed ripple waveform and spectrum of summed ripple waveform

figure(11);
set(11, 'Color', 'w', 'units', 'centimeter', 'position', [0 0 17 15])
p4 = subplot(2, 1, 1)
set(gca, 'FontSize', 10, 'LineWidth', 2, 'xlim', [0, 4*T_sw], 'ylim', [-2, 3]);
grid on;
hold on;
handlevector(1) = plot(t, I_sum_even, 'r', 'LineWidth', 2, 'LineStyle', '--');
handlevector(2) = plot(t, I_sum, 'k', 'LineWidth', 2);

[hx, hy] = format_ticks(gca, {'$0$ ', '$1 \times 10^{-5}$ ', '$2 \times 10^{-5}$ ', '$3 \times 10^{-5}$ ', '$4 \times 10^{-5}$ ', '$-2$ ', '$-1$ ', '$0$ ', '$1$ ', '$2$ ', '$3$ '}, [0, 1e-5, 2e-5, 3e-5, 4e-5], [-2, -1, 0, 1, 2, 3], 0, 0, [0.05, 0, 0, 0], 'FontSize', 14, 'FontSize', 'Bold');

h = xlabel('Time / s', 'Interpreter', 'latex', 'FontSize', 14, 'FontSize', 'Bold');

pos = get(h, 'pos');

h = ylabel('$i_\text{sum}(t)$ / A', 'Interpreter', 'latex', 'FontSize', 14, 'FontSize', 'Bold');

pos = get(h, 'pos');

h_legend = legend(handlevector([1 2]), {'Sum of ripple components with even phase-shift $\Delta i_{\text{sum}} = 3.80$ A', 'Sum of ripple components with proposed phase-shift $\Delta i_{\text{sum}} = 2.38$ A'}, 'Interpreter', 'latex');
set(h_legend, 'FontSize', 12, 'FontSize', 'Bold');

p5 = subplot(2, 1, 2)
set(gca, 'FontSize', 10, 'LineWidth', 2, 'xlim', [0, 20], 'ylim', [0, 1.5]);
grid on;
hold on;
handlevector(1) = plot(0, 0, 's', 'MarkerEdgeColor', 'r', 'MarkerFaceColor', 'r', 'MarkerSize', 6);
handlevector(2) = plot(0, 0, 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'k', 'MarkerSize', 6);

[hx, hy] = format_ticks(gca, {'$0$ ', '$1$ ', '$2$ ', '$3$ ', '$4$ ', '$5$ ', '$6$ ', '$7$ ', '$8$ ', '$9$ ', '$10$ ', '$11$ '})
for k = 1:20
  
  \textit{%% Phase 1}

  \texttt{a_{1k} = 0.5*Delta_i1/((pi*k)^2) * ( (1/D_1 + 1/(1-D_1)) * (cos(2*pi*k*D_1) - 1) )};
  \texttt{b_{1k} = 0.5*Delta_i1/((pi*k)^2) * (1/D_1 + 1/(1-D_1)) * sin(2*pi*k*D_1)};
  \texttt{A_{1k} = sqrt(a_{1k}^2 + b_{1k}^2) ;}
  \texttt{phi_{1k} = atan2(b_{1k}, a_{1k}) ;}
  \texttt{\% phi_{2k} = 2*atan(b_{2k}/(A_{2k}+a_{2k}))}
  \texttt{if(phi_{1k} < 0)}
    \texttt{phi_{1k} = phi_{1k} + 2*pi ;}
  \texttt{end}

  \textit{%% Phase 2}

  \texttt{a_{2k} = 0.5*Delta_i2/((pi*k)^2) * ( (1/D_2 + 1/(1-D_2)) * (cos(2*pi*k*D_2) - 1) )};
  \texttt{b_{2k} = 0.5*Delta_i2/((pi*k)^2) * (1/D_2 + 1/(1-D_2)) * sin(2*pi*k*D_2) ;}
  \texttt{A_{2k} = sqrt(a_{2k}^2 + b_{2k}^2) ;}
  \texttt{phi_{2k} = atan2(b_{2k}, a_{2k}) ;}
  \texttt{\% phi_{2k} = 2*atan(b_{2k}/(A_{2k}+a_{2k}))}
  \texttt{if(phi_{2k} < 0)}
    \texttt{phi_{2k} = phi_{2k} + 2*pi ;}
  \texttt{end}
%%% Phase 3
a_3k = 0.5*Delta_i3/((pi*k)^2)*((1/D_3 + 1/(1-D_3))*cos(2*pi*k*D_3) - 1);  
b_3k = 0.5*Delta_i3/((pi*k)^2)*((1/D_3 + 1/(1-D_3))*sin(2*pi*k*D_3));
A_3k = sqrt(a_3k^2 + b_3k^2);  
phi_3k = atan2(b_3k,a_3k);  
% phi_3k = 2*atan(b_3k/(A_3k+a_3k))
if(phi_3k < 0)
  phi_3k = phi_3k + 2*pi;
end

Theta_1k = phi_1k + k*Phi_01;  
Theta_2k = phi_2k + k*Phi_02;  
Theta_3k = phi_3k + k*Phi_03;

%%% Define complex phasors
Phase1k = A_1k*exp(i*Theta_1k);  
Phase2k = A_2k*exp(i*Theta_2k);  
Phase3k = A_3k*exp(i*Theta_3k);  
PhaseSumk = Phase1k + Phase2k + Phase3k;

phi_Sumk = atan2(imag(PhaseSumk),real(PhaseSumk));  
if(phi_Sumk < 0)
  phi_Sumk = phi_Sumk + 2*pi;
end

A_Sumk = abs(PhaseSumk);

Theta_1k_sym = phi_1k + k*0;  
Theta_2k_sym = phi_2k + k*(2/3)*pi;  
Theta_3k_sym = phi_3k + k*(4/3)*pi;

%%% Define complex phasors
Phase1k_sym = A_1k*exp(i*Theta_1k_sym);  
Phase2k_sym = A_2k*exp(i*Theta_2k_sym);  
Phase3k_sym = A_3k*exp(i*Theta_3k_sym);
PhaseSumk_sym = Phase1k_sym + Phase2k_sym +  
   Phase3k_sym;
phi_Sumk_sym = atan2(imag(PhaseSumk_sym),real(PhaseSumk_sym));
if(phi_Sumk_sym < 0)
    phi_Sumk_sym = phi_Sumk_sym + 2*pi;
end

A_Sumk_sym = abs(PhaseSumk_sym);

% Plot magnitude
figure(10);
subplot(p1);
plot(k,A_1k , 'o',' MarkerEdgeColor ','b', 'MarkerFaceColor ', 'b', 'MarkerSize',5);
plot([k k],[0 A_1k ],'b',' LineWidth',3);

subplot(p2);
plot(k,A_2k , 'o',' MarkerEdgeColor ','g', 'MarkerFaceColor ', 'g', 'MarkerSize',5);
plot([k k],[0 A_2k ],'g',' LineWidth',3);

subplot(p3);
plot(k,A_3k , 'o',' MarkerEdgeColor ','r', 'MarkerFaceColor ', 'r', 'MarkerSize',5);
plot([k k],[0 A_3k ],'r',' LineWidth',3);

figure(11);
subplot(p5);
plot(k,A_Sumk_sym, 's',' MarkerEdgeColor ', 'r', 'MarkerFaceColor ', 'r', 'MarkerSize',6);
plot([k k],[0 A_Sumk_sym ],'r',' LineWidth',2);

plot(k,A_Sumk , 'o',' MarkerEdgeColor ','k', 'MarkerFaceColor ', 'k', 'MarkerSize',6);
plot([k k],[0 A_Sumk ],'k',' LineWidth',2);

g_1 = g_1 + A_1k*cos(k*omega*t-k*Phi_01-phi_1k);
g_2 = g_2 + A_2k*cos(k*omega*t-k*Phi_02-phi_2k);
g_3 = g_3 + A_3k*cos(k*omega*t-k*Phi_03-phi_3k);
sum = sum + A_Sumk*cos(k*omega*t-phi_Sumk);
end

%%% Control plot
figure(15);
set(gca,'FontSize',12,'LineWidth',2,'xlim', [0,5*T_sw]);
title('Truncated Fourier Series, polar result','
    FontWeight','bold');
grid on;
hold on;
plot(t,g_1,'b','LineWidth',2);
plot(t,g_2,'g','LineWidth',2);
plot(t,g_3,'r','LineWidth',2);
plot(t,g_1+g_2+g_3,'k','LineWidth',3);
plot(t,sum,'c','LineWidth',2);
REFERENCES


