UPTEC F10 013
Examensarbete 30 hp
Februari 2010

Design and construction of a bidirectional DCDC converter for an EV application

Magnus Hedlund

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Design and construction of a bidirectional DCDC converter for an EV application

Magnus Hedlund

A Sliding Mode Control System for a Bidirectional DCDC Converter was designed and a low voltage prototype was constructed. The control system based its decisions solely on the latest available measurements, which improves performance when changing operative quadrant, since no memory needs reinitializing (such as for PI and state prediction methods). A boost control philosophy was presented, based on a current source approximation. The control was found to be stable without steady-state errors when the variance of the input/output dynamics was high.
The target application for the DCDC Converter is an EV (Electric Vehicle) with a flywheel driveline, which puts additional requirements of the converter. Among these are current and voltage control, bidirectionality, and a broad input voltage range.
Simulations were performed in Simulink prior to physical implementation, proving functionality of the proposed control system. The physical implementation of the control was done on a digital signal processor with code compiled from C. A median filter was designed to increase measurement efficiency for the current sensors which had shot-like noise distortions.

Handledare: Janaína Gonçalves de Oliveira
Ämnesgranskare: Hans Bernhoff
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F10 013

Contents

List of Figures

List of Tables

Nomenclature

- a_n - Fourier series constant
- C - Capacitor
- D - Diode
- d - Duty ratio [*dimensionless*]
- e - Error, defined as $e = y - r$
- E_{peak} - Peak energy consumption [J]
- f - Frequency [s^{-1}]
- k - Constant
- i - Momentaneous current through inductor element (in 4-quad circuit, from A to B) [A]
- i_A - Momentaneous current defined from plus to minus between A-side connectors [A]
- i_B - Momentaneous current defined from plus to minus between B-side connectors [A]
- i_C - Current through capacitor [A]
- I_i - Input current (time averaged) [A]
- i_i - Input current (momentaneous) [A]
- i_R - Current through resistor [A]
- I_o - Output current (time averaged) [A]
- i_o - Output current (momentaneous) [A]
- L - Inductance in inductor
- P_i - Input power [W]
- P_o - Output power [W]
- P_{peak} - Peak power consumption [W]
- U_i - Input voltage (time averaged) [V]
- u_i - Input voltage (momentaneous) [V]
- u_L - Voltage drop over inductor [V]
- U_o - Output voltage (time averaged) [V]
- u_o - Output voltage (momentaneous) [V]
- $\widehat{U}_o$ - Estimation of time averaged voltage [V]
- R - Resistance in resistor
- r - Control reference value
- S - Switch representation in equations, where $S \in \{0, 1\}$
- S_{down} - Lower switch in 4QC, located on output side [*dimensionless*]

- S_{off} - Switch with state off (not conducting), equal to 0 in equations [*dimensionless*]
- S_{on} - Switch with state on (conducting), equal to 1 in equations [*dimensionless*]
- S_{up} - Upper switch in 4QC, located on input side [*dimensionless*]
- T - Period time [s]
- t - Elapsed time [s]
- t_{on} - Time during which switch is in on state [s]
- t_{off} - Time during which switch is in off state [s]
- u - Control value
- $\mathbf{x}$ - State vector
- y - System output
- δ - Hysteresis band width
- η - Efficiency [*dimensionless*]
- σ - Decision law

Abbreviations

- 4QC - 4 Quadrant Converter
- DC - Direct Current
- ESR - Equivalent Series Resistance
- MISO - Multiple Input, Single Output
- PID - Proportional Integral Derivative
- PWM - Pulse Width Modulation
- SISO - Single Input, Single Output
- SMC - Sliding Mode Control
- SOC - State of Charge
- VSC - Variable Structure Control

Mother

Janaína Gonçalves de Oliveira

Nils Finnstedt

Johan Abrahamsson

Johan Lundin

Juan de Santiago

Venugopalan Kurupath

Remya Krishna

Thanks for everything!

1 Introduction

Development of an alternative vehicle technology The transport sector is heavily dependent on the internal combustion engine because of its good driving range and short refueling time. However, due to the growing environmental awareness, alternatives which are not dependent on fossil fuel are being investigated. The battery-powered electric vehicle (EV) has been proposed as an alternative since it can recharge its batteries from renewable energy sources (such as windpower or solar cells). The simple design of the electric engine is another favourable factor, since it involves fewer moving parts than combustion engines usually require. The main issues of the EV are the driving range and the time required to recharge, which are dependent on the energy storage element (the battery)[1].

Power Limits for Batteries Batteries consist of electrochemical cells, and provides electrical energy through chemical reactions. Recharging is done by reversing said reactions. The internal processes in batteries are complex, different charge/recharge-cycles affect the performance and efficiency, there are limits to how fast the recharging can be done, etc. Typically, an EV requires several hours of grid connection before being fully recharged.

Fast accelerations demands high power, which is demanding for batteries[2]. Batteries that can handle power peaks are expensive, in particular, lithium-ion batteries are less efficient when delivering high currents[3]. Similar to the acceleration problem is the deceleration situation, where attempts to retrieve energy by *regenerative braking* have been researched to improve driving range. The power peaks involved are (as when accelerating) high for short durations.

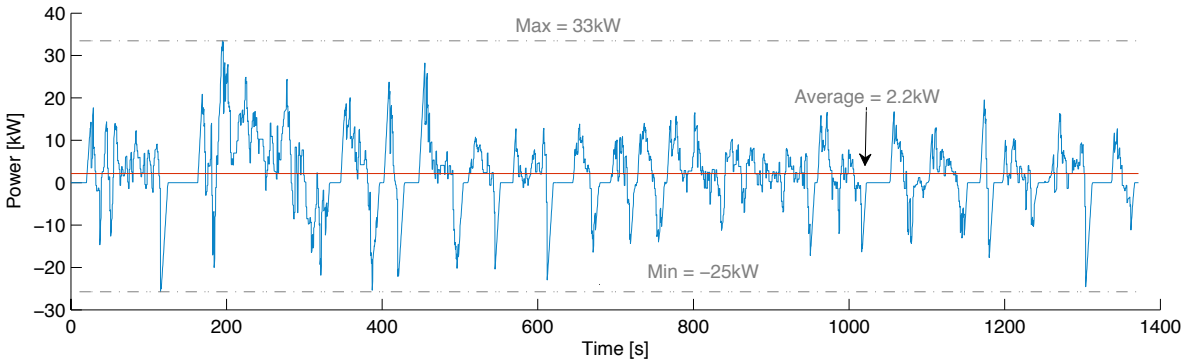

Figure 1: US Urban Drivecycle

Power Buffers Recent research show that the average power consumption is much lower than the power peaks that occur when accelerating or decelerating[4]. Figure 1 contains power consumption on an US Urban Drivecycle where max, minimum and average power consumptions are marked. The average power consumption is about 15 times smaller than the peak, as can be seen from the figure. A power buffer could improve performance by moving power transients away from the battery, letting it provide a small and constant power ($2.2kW$ if using data from Figure 1).

Discoveries in the field of supercapacitors have opened up new possibilities for high power energy storages, but the total energy stored is quite low in the context of an EV application[5]. Flywheels combine high power and energy density and can be designed to store the peak power energy[6]. An approximate comparsion between supercapacitors, flywheels and batteries can be seen in Figure 2. Batteries have comparatively low power ratings compared to supercapacitors and flywheels.

Simulations based on the US urban drivecycle have been done with a flywheel system acting as a power buffer[4], and the results can be seen in Figure 3. The battery discharge rate remains constant,

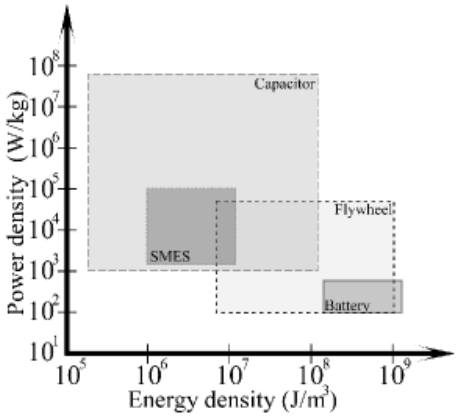

Figure 2: Approximate power and energy ratings of Energy Storages

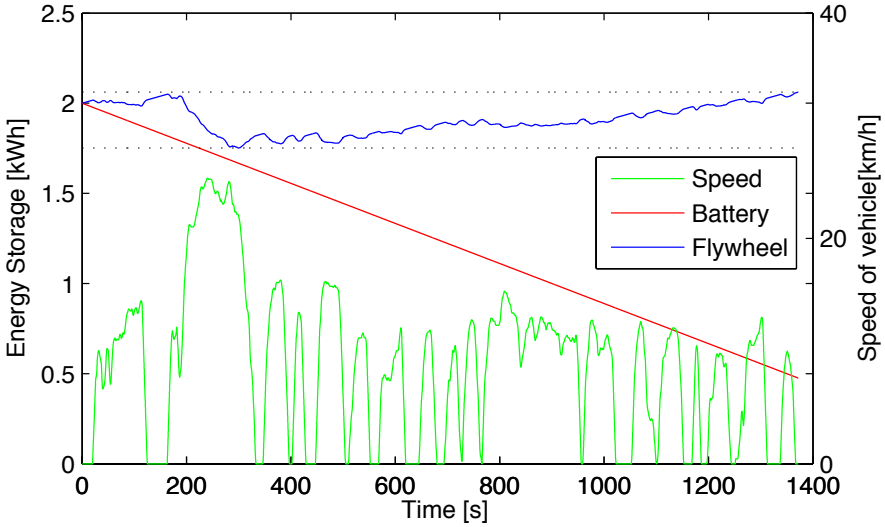

Figure 3: Simulation of the energy stored in the flywheel

while the flywheel delivers the instantaneous power required to accelerate (or stores the power generated when decelerating). In the simulation above, the maximum change of energy in the flywheel is $340Wh$.

A flywheel buffered electric driveline A driveline with a flywheel power buffer has been proposed[4]. The suggested setup consists of a high and a low voltage side, where the latter handles relatively low power levels. The high voltage side will then handle high powers with good efficiency, at much higher voltages than is feasible for batteries. Traditionally, flywheel systems are constructed on the same power levels as the battery[7, 8], which puts demands on battery and limits the maximum voltage level (thus decreasing efficiency). A model of the driveline can be seen in Figure 4. The flywheel is built on the same axis as a three-phase electric machine (which is double wound).

Transfer of energy between battery and flywheel machine The battery provides DC current, but the motor/generator is a three-phase machine. Hence, a bidirectional ACDC (inverter/rectifier) is used together with a DCDC Converter, which acts as a power valve. This report is focused on the latter, a 4-quadrant DCDC Converter, whose purpose is to control the amount of power and energy that is transferred between the flywheel and the battery.

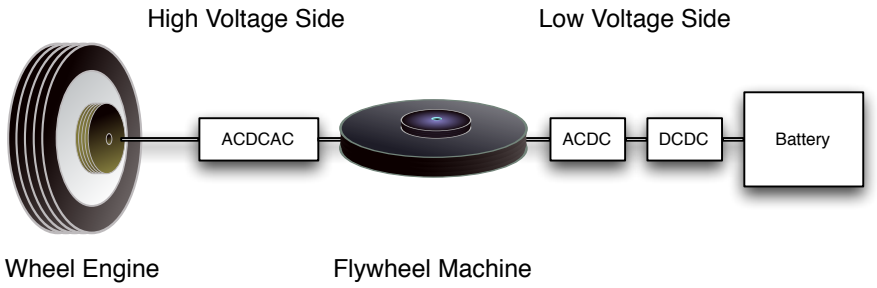

Figure 4: Flywheel driveline

Specification of the DCDC Converter The DCDC Converter specifications are as follows:

- Should control the power flow between the battery and the flywheel
- Must be bidirectional
- Should be able to provide both higher and lower output voltage (buck and boost mode)
- Should be able to control current, for battery charging and to stay inside motor winding limits
- The control must be parameter invariant, i.e. variations in the input/output dynamics must not affect performance
- The control must be able to operate in all four quadrants, and move between them seamlessly
- Testing voltages should span between 0 and 50 volts, but the design should be scaleable up to higher voltages and power levels
- The testing current should be between 0 and 10 amperes, but should also be scaleable

2 Theory

2.1 Switched Converters

2.1.1 Introduction to DCDC Converters

The purpose of a DCDC Converter is to shift the input DC voltage and current to a desired output DC voltage or current. As suggested in Figure 5, for a given input voltage and current the corresponding outputs can both be different.

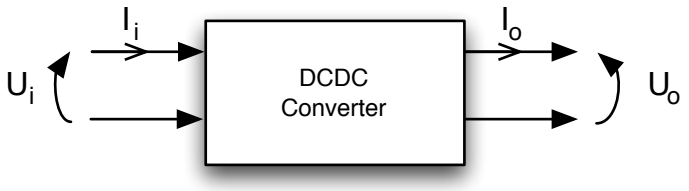

Figure 5: A DCDC Converter Model

In practice voltage shifts are done by turning discrete switches on and off very fast, choppering the input voltage. Specifically, if $U_\mathrm{i} > U_\mathrm{o}$ the device is working in BUCK MODE and is a STEP DOWN CONVERTER, and if $U_\mathrm{i} < U_\mathrm{o}$ it's in BOOST MODE and is a STEP UP CONVERTER.

If the relative efficiency of the device is assumed to be η, an approximation can be made from the power relation for voltage and current:

$$\begin{cases} P_i &= I_i U_i \\ P_o &= I_o U_o \\ \eta &= \frac{P_o}{P_i} \end{cases}$$

$$\Rightarrow I_i U_i \eta = I_o U_o \Leftrightarrow \frac{U_i \eta}{U_o} = \frac{I_o}{I_i} \tag{1}$$

If the inputs and system losses are kept constant, the output voltage will be inversely related to the output current.

$$\Rightarrow \frac{1}{U_o} \propto I_o \tag{2}$$

The result seen in equation 2 is a principle that holds for all converters, essentially it says that the output voltage can be higher if the current is smaller. Thus, if only the input voltage is kept constant, a higher output voltage demands a higher input current.

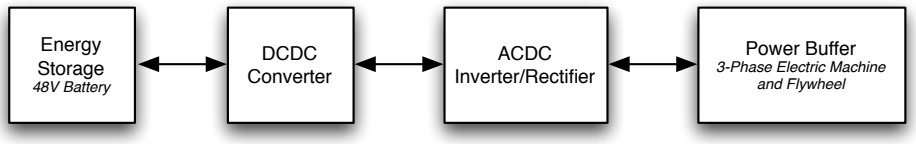

Figure 6: DCDC Converter in the low voltage side of a flywheel power system.

Starting from equation 1 and assuming constant losses and input voltage gives:

$$\Rightarrow \frac{U_i \eta}{U_o} = \frac{I_o}{I_i} \Rightarrow k I_i \propto P_o \tag{3}$$

Equation 3 shows us how a DCDC Converter can act as a valve for the power flow between two systems. An example power system with a DCDC Converter acting as a power valve is shown in Figure 6.

2.1.2 Buck and Boost Circuits

Two common DCDC Converters are the buck and boost choppers[9, 10].

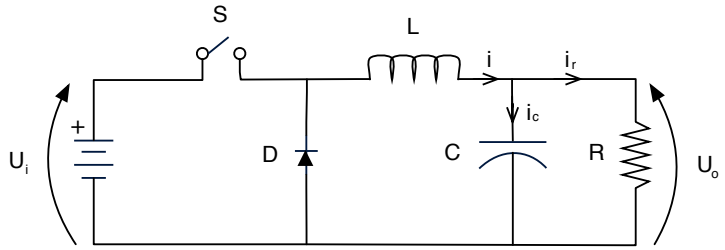

Figure 7: Buck Converter

They are both second-order dynamics devices, with two dynamic (capacitor & inductor) and two discrete components (switch & diode). Both are based on LCD-filters[1], and their respective setups can be seen in Figures 7 & 8.

The buck converter has a lower output than input voltage, $U_i > U_o$, which is done by chopping the input voltage as a PWM-signal[10]. The boost converter produces a higher output voltage ($U_o > U_i$) by draining more current from the input (as discussed above).

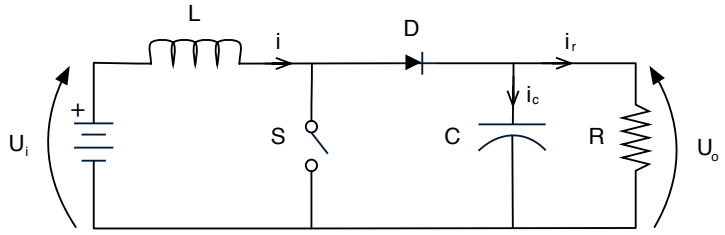

Figure 8: Boost Converter

2.1.3 Four Quadrant Chopper

The 4 Quadrant Chopper operates in all quadrants[11](operating modes) which means it is capable of transporting energy in both directions and of stepping voltage up and down (depicted in Figure 9). When operating the switches in a certain sense, the topology will be similar to the buck and the boost converters, and the same dynamics will therefore apply (with small modifications)[12]. Tables 1 & 2 shows which switches and diodes are active (denoted with A) during certain modes of operation.

[1] Inductor,Capacitor,Diode-filter

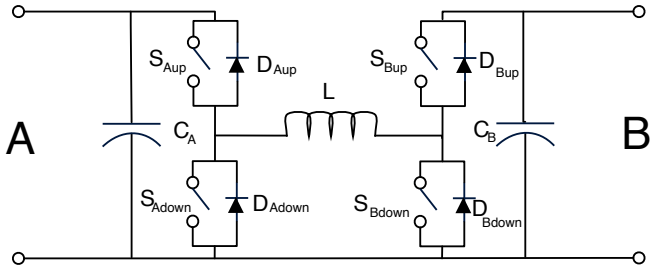

Figure 9: 4 Quadrant Converter

Energy direction	Mode	S_{Aup}	S_{Bdown}	S_{Bup}	S_{Adown}
A to B	Buck	A	0	0	0
A to B	Boost	1	A	0	0
B to A	Buck	0	0	A	0
B to A	Boost	0	0	1	A

Table 1: Modes of operation for switches

Energy direction	Mode	D_{Aup}	D_{Bdown}	D_{Bup}	D_{Adown}
A to B	Buck	0	0	1	A
A to B	Boost	0	0	A	0
B to A	Buck	1	A	0	0
B to A	Boost	A	0	0	0

Table 2: Modes of operation for diodes

2.2 Control Design

2.2.1 Introduction

The design specification (from section 1) states that the converter should control the energy flow between the battery and the flywheel/electric machine. Hence, the desired control variables are both output current and voltage. For safety reasons (and to protect components), both should always be monitored simoultaneously (see Table 3). The 4QC has four switches, and therefore four inputs.

Target Control Variable(s)	Usage in electric driveline application
Voltage Control	Useful in the end of many battery charging processes
Current Control	Battery Charging
Voltage Control and Current Limit	Power transfer to electric machine without burning windings
Current Control and Voltage Limit	Safe Battery Charging

Table 3: Control Situations

One control approach for switched converters is the use of a linearized system with techniques such as *Open Loop Control* or *PID*. A discussion about these two can be found in Appendix A.1, where it is stated that for non-linear systems (like switched converters) with large parameter variations (as in this application), linear control techniques have inherent drawbacks such as:

- Input and output parameter variations (discussed below) will affect the linearized models causing steady-state errors.
- Feedback controls with integral parts (to remove steady-state errors) are dependent on memory, but this memory becomes invalid when changing operating quadrant, since the topology of the 4QC changes with operation mode.

The arguments above leads to the development of a non-linear sliding mode control, found in Section 2.2.3.

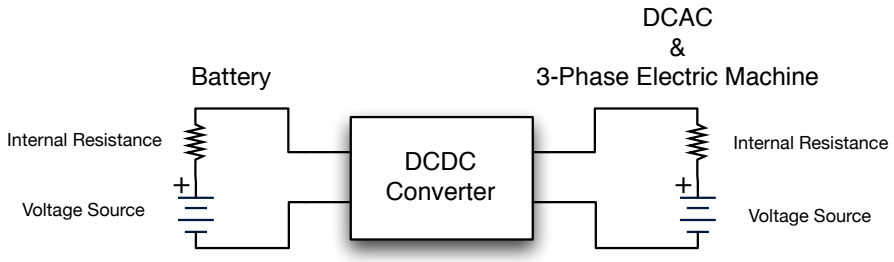

Figure 10: Input/Output Dynamics

2.2.2 System input/output dynamics

The variations of the input and output of the specific application is an important detail when designing a control system. Table 4 contains comments on the model presented in Figure 10. The model is a simplification of both the battery and the flywheel system. On the battery side, the voltage level will vary with the SOC. Also, there is internal resistance which will limit the maximum current. The electric machine and the ACDC-stage has its own dynamics. When the machine is at rest, the back

emf will be zero, and the resistance in the windings will be the only resistance in the loop. When the back emf is very high, the equivalent resistance[2] in the machine will be high. Models which are based on constant resistance in the output are therefore not reliable.

	Battery Side	DCAC & Electric Machine Side
Internal Resistance	Limits the maximum current that can be drawn. Is dependent on external physical factors, such as temperature and charge/discharge conditions.	Assumed constant, but low.
Voltage Source	Varies with SOC.	Equals zero when machine is at rest, since no back-emf is produced. When the machine rotates, this voltage will be proportional to the speed.

Table 4: Input/Output dynamics overview

2.2.3 Sliding Mode Control

The switched converters are non-linear because of the discrete switches that changes the dynamics (the state equations) of their systems when changing state. Switched mode converters therefore belong to the class of Variable Structure Systems[3]. A non-linear control technique specialised on VSS is Sliding Mode Control[13, 14, 15]. The general idea is to create artificial stable points by studying the system dynamics and then create conditions for when to turn switches on or off. A sliding mode control system is based on the principal behaviour of the circuit which stays the same even though parameters may vary. If sliding mode can be implemented in the 4QC then the problems with parameter variations will be effectively resolved.

Design procedure Firstly all substructures of the system are identified and their respective state equations are found. For two-dimensional systems (such as the buck and the boost topologies), the system presentation is simple (see Section 3.1.2). For higher-order systems, the mathematical analysis becomes harder. Generally, consider the state vector system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t, S_1, S_2, .., S_n) \tag{4}$$

where $\mathbf{x}$ is the state vector (with N elements), t is time and S represent discrete switch states. The states are usually defined as the physical state minus the reference value, i.e. the error.

The goal is then to make all state variables reach origo. Mathematical analysis is performed to define a switching plane or path through the N-dimensional state space, which defines which mode (topology) to work in. Defining this plane is an optimality problem, no matter which point in vector space the system starts from, the mode that makes the states reach origo in the smallest amount of time is always chosen. A general decision law for a plane only depending on the current state can be written on this form:

[2] If the voltage source and internal resistance in Figure 10 are assumed to be just one resistive load.

[3] For example, a buck circuit has two states (either the switch is on or it is off) and each state comes with its own set of state equations. Different state equations for circuits with switches are deduced in Section 3.1.1.

$$\sigma_m = \mathbf{g}(\mathbf{x}) = \sum_{i=1}^{N} c_i x_i = 0 \tag{5}$$

Suitable control laws can then be constructed for each switch:

$$S_k = \begin{cases} 1 & , \quad \sigma_k > 0 \\ 0 & , \quad \sigma_k < 0 \end{cases} \tag{6}$$

Also, a hysteresis band is added to the control law, such that:

$$S_k = \begin{cases} 1 & , \quad \sigma_k > \delta \\ 0 & , \quad \sigma_k < -\delta \end{cases} \tag{7}$$

For a one-dimensional system this work is trivial and for two-dimensional systems studies of the phase portrait is a feasible method for defining mode boundaries. For higher-order systems, there is no simple way of visualizing the state space, but with careful analysis of the equations, boundaries can be established. These boundaries are commonly reffered to as the SLIDING SURFACE, hence the name of the control system Sliding Mode Control.

Convergence There are three conditions which must hold in order for the system to converge[14, 15].

- Existence Condition. The trajectories of the subsystems must be directed towards the sliding surface when they are close to it.
- Hitting Condition. All initial conditions must lead to the system to reach the sliding surface.
- Stability Condition. The movement along the sliding surface should be directed to a stable point (the reference point).

Advantages & Drawbacks

\+ Control does not require linearization.

\+ Is based on principal behaviour, which makes it insensitive to parameter variations.

\+ Decicions are based on current system state, no memory required (mode transistions are possible).

\- Mathematical analysis can be difficult.

\- Sensor speed sets limits for minimum hysteresis band width.

Conclusion Sliding Mode Control (abbreviated SMC) can be an optimal non-linear controller. It can be configured for n input dimensions, and m output dimensions. It can be viewed as a multidimensional hysteresis controller. Even though analysis is quite tedious, once the optimal boundaries have been defined, many control targets such as stabilization, tracking and insensitivity to load and input conditions are very good. Better then any PID controller based on a linearized model made for specific working conditions. This is the control that is implemented in the 4QC. The next chapter discusses the design of the controller.

3 Modelling

In section 3 a sliding mode control system is designed for the 4QC.

3.1 Dynamics of Buck Converter

3.1.1 State equations

The voltage over the capacitor and the current through the inductor are chosen as state variables. The derivations for the following state equations can be found in Appendix A.6:

$$\frac{di}{dt} = \frac{1}{L}(Su_i - u_C) \tag{8}$$

$$\frac{du_o}{dt} = \frac{i}{C} - \frac{u_o}{RC} \tag{9}$$

The equations above fully describe the ideal circuit[4]. There are two dynamic components (C,L), hence the system is considered second-order. This system can be represented by using two linear models, to handle the built-in non-linearities (the switch & diode). The linear system when the switch is on ($S = 1$) is:

$$\begin{bmatrix} \dot{i} \\ \dot{u_o} \end{bmatrix} = \begin{pmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{pmatrix} \begin{bmatrix} i \\ u_o \end{bmatrix} + \begin{bmatrix} \frac{u_i}{L} \\ 0 \end{bmatrix} \tag{10}$$

The corresponding system for when the switch is off ($S = 0$) is:

$$\begin{bmatrix} \dot{i} \\ \dot{u_o} \end{bmatrix} = \begin{pmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{pmatrix} \begin{bmatrix} i \\ u_o \end{bmatrix} \tag{11}$$

To construct a more realistic model equivalent series resistances and inductances can be added to the circuit above. If Equation 10 and 11 is combined the final model is:

$$\begin{bmatrix} \dot{i} \\ \dot{u_o} \end{bmatrix} = \begin{pmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{pmatrix} \begin{bmatrix} i \\ u_o \end{bmatrix} + \begin{bmatrix} \frac{u_i}{L} \\ 0 \end{bmatrix} S \tag{12}$$

3.1.2 Phase portraits of Buck Converter

A phase portrait for the two state variables derived for the buck converter above can be seen in Figures 11 and 12. Each portrait is corresponding to a certain switch state. Both systems consist of a single stable point and no matter what state the systems are in, they will converge to these points. The system equations have all been simplified with unity parameter values. It is the principal behaviour which is of interest here[5].

[4] Note that the ideal voltage drop over the diode is assumed to be zero.

[5] Note that this model does not account for the inductor diode, in reality, the inductor current should not be able to become negative. This simplification is disregarded since the region of interest is in non-negative state vector values.

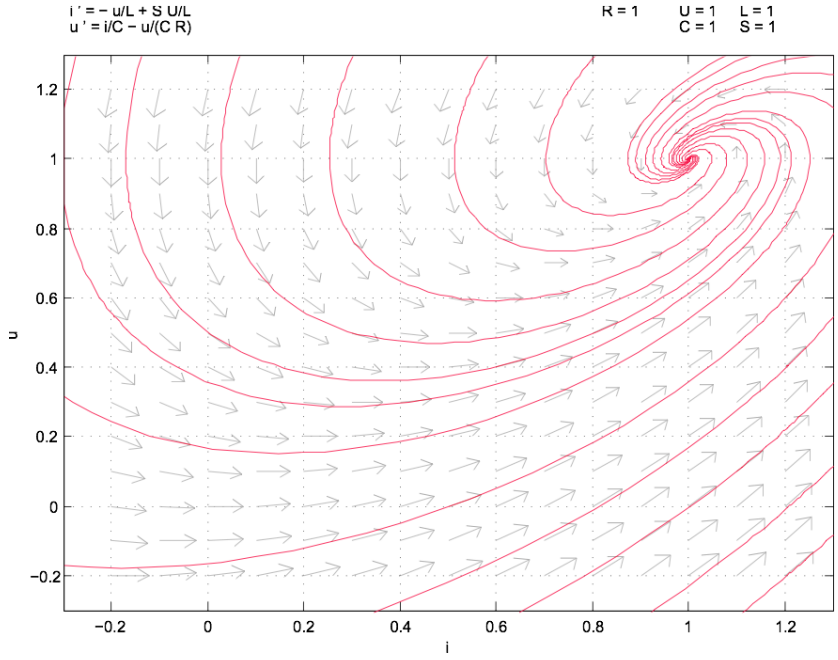

Figure 11: Buck Phase Portrait, with the switch in the "on" state

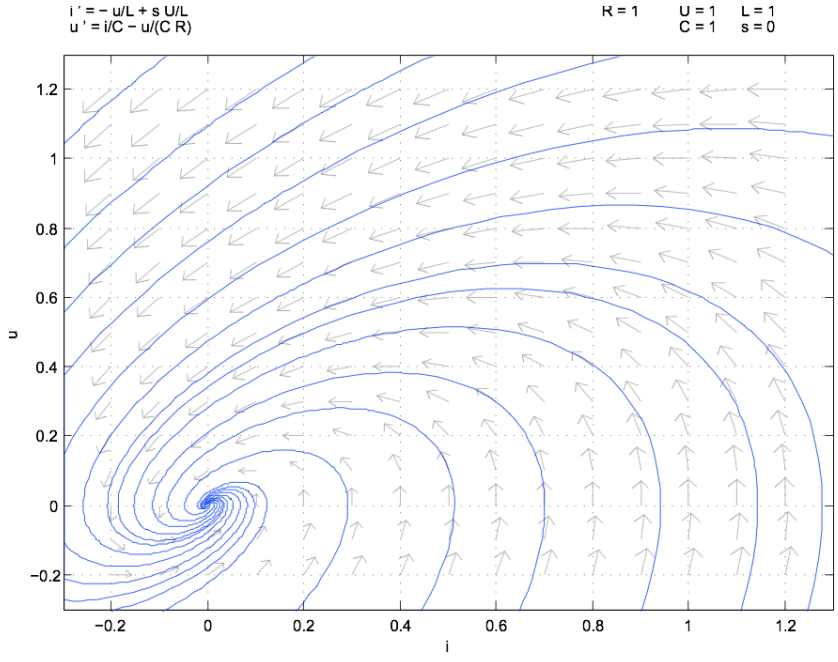

Figure 12: Buck Phase Portrait, with the switch in the "off" state

3.1.3 Control Idea

The output voltage is chosen as the only decision variable, and a hysteresis band (marked in green) is placed between the two stable points. The off-mode is used above the hysteresis band, and the

on-mode in the opposite and thus a new stable point has now been artificially created. This has been illustrated in Figure 13.

All three convergence criteras (from Section 2.2.3) can be graphically verified by studying the figure. In the phase portrait below the state variable was not adjusted so the reference was located in origo.

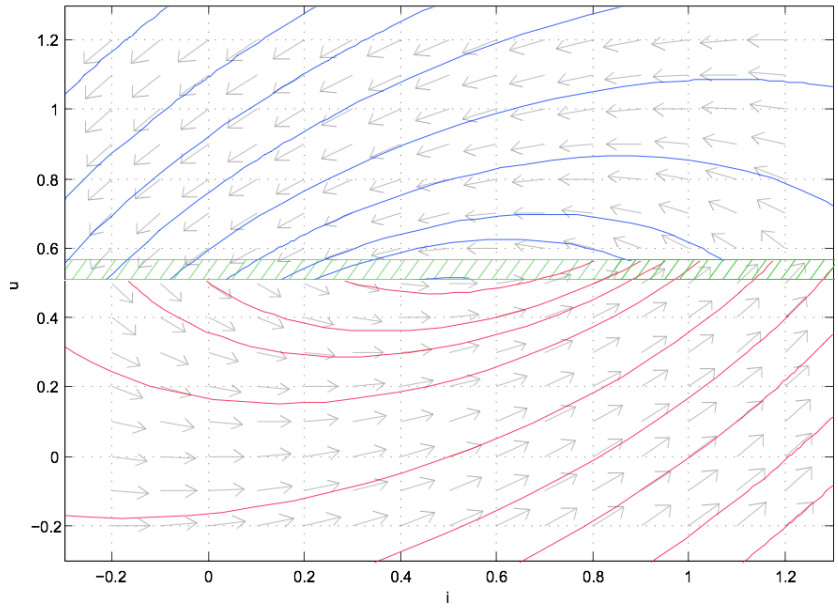

Figure 13: Buck Dynamics with Hysteresis Band

3.2 Dynamics of Boost Converter

3.2.1 State equations

The voltage over the output and the current through the inductor are chosen as state variables (same as for the buck). The derivations are similiar to the buck above, and can be found in Appendix A.7:

$$\begin{bmatrix} \dot{i} \\ \dot{u_o} \end{bmatrix} = \begin{pmatrix} 0 & -\frac{1}{L}(1-S) \\ \frac{1}{C}(1-S) & -\frac{1}{RC} \end{pmatrix} \begin{bmatrix} i \\ u_o \end{bmatrix} + \begin{bmatrix} \frac{u_i}{L} \\ 0 \end{bmatrix} \tag{13}$$

3.2.2 Phase portraits of Boost Converter

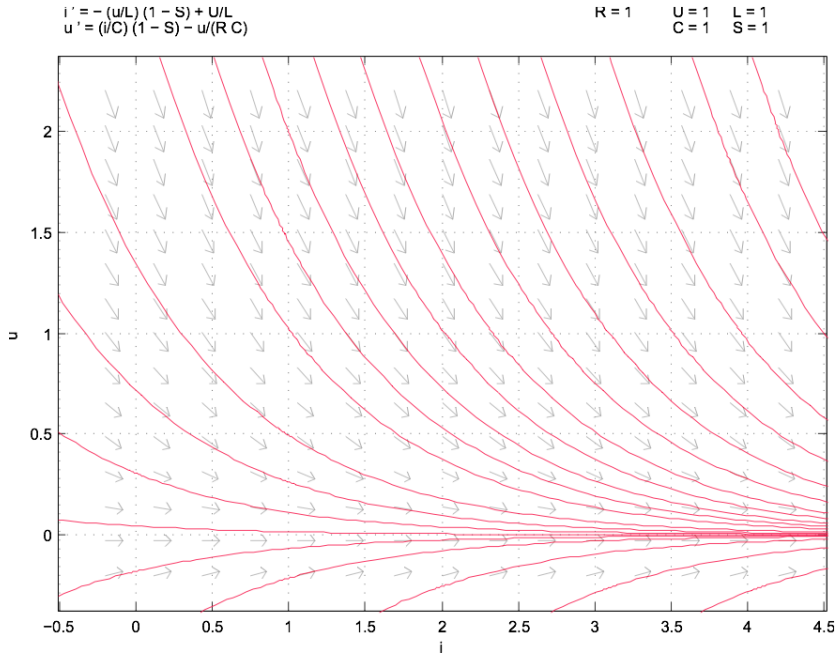

Figure 14: Boost Phase Portrait, with the switch in the "on" state

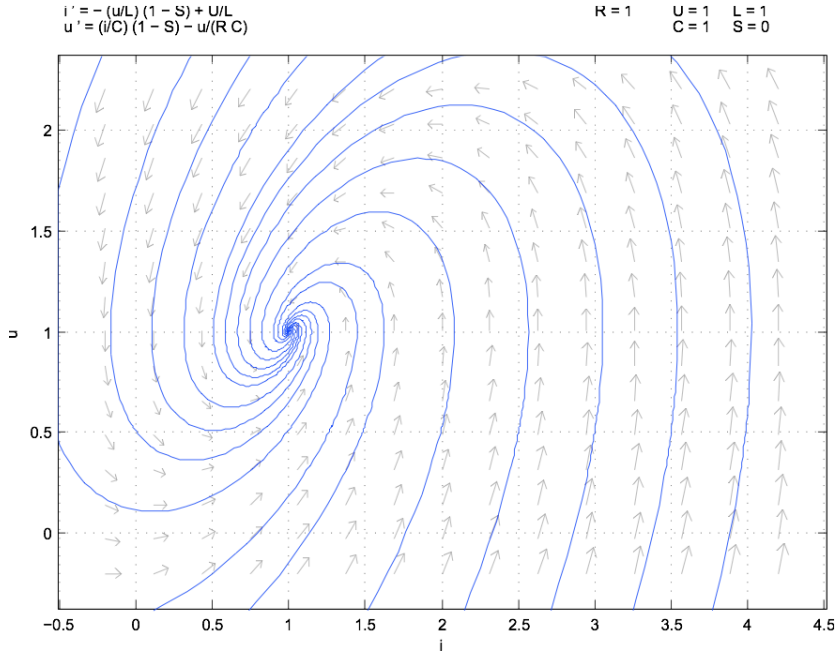

Figure 15: Boost Phase Portrait, with the switch in the "off" state

3.2.3 Problems with control idea

An attempt similar to the buck control idea (3.1.3) is not possible here, which can be understood as follows. Assume only voltage u is used as the control decision (a 1D decision law). Then there is no possible (1D) sliding surface which ever takes the system above the input voltage (can be verified by looking at Figure 16). Also, a sliding surface below the input voltage is non-feasible, since the inductor current will never reach a stable point (and grow to infinity).

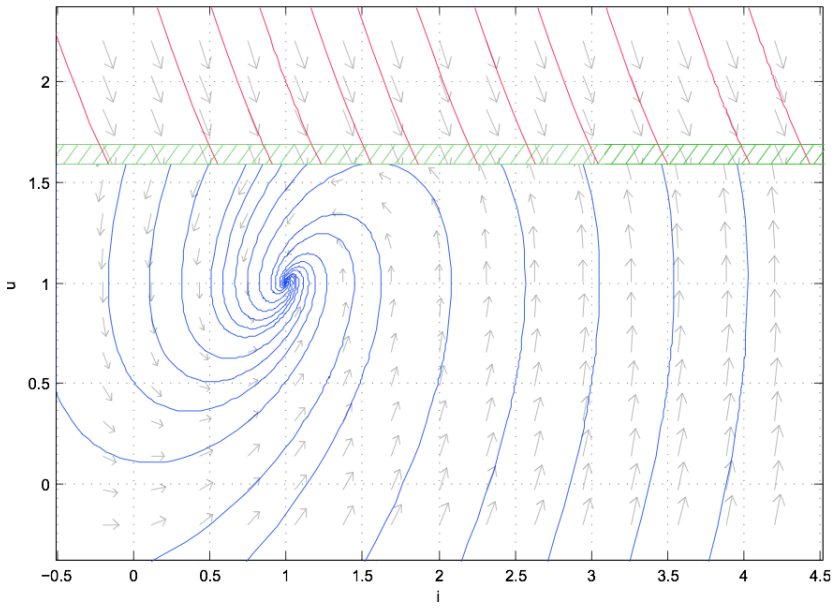

Figure 16: Instable boost control with a 1D sliding line

Another approach is to implement a two-dimensional decision law[16]. A suggestion is shown in Figure 17 with an artificial stable point. Hence, a 2D sliding surface can be a stable control for a boost converter, but will require two reference values (both for inductor current and for output voltage). The inductor current reference is different for different input/output dynamics, solutions exist to approximate the reference with a PI regulator[16]. However, memory dependent controllers are (as stated before) unsuitable for switching between different quadrants.

3.3 Dynamics of 4 Quadrant Converter

Direct analogies can be made with buck and boost circuits in order to determine operation for the 4QC. Table 1 contains a list of operating modes.

Assuming A as input side and B as output, let all switches be turned off (not conducting) except S_{Aup} which is switching. The 4QC now has similar dynamics as the buck (a comparsion can be seen in Appendix A.3). The only difference is that the 4QC have additional diodes (which do not interfere with standard current directions) and also a parallell capacitor at the input.

Input Capacitor The input capacitor is not relevant to the system dynamics and can be neglected when considering control system behaviour (as can be seen in Simulations Section 4.3). The component can actually be seen as a low-pass filter on the input signal (Figure 18), which strives to keep the voltage into the converter constant. No additional dynamics that affect the control system is thus added by having an input capacitor, hence the buck and boost models are considered accurate.

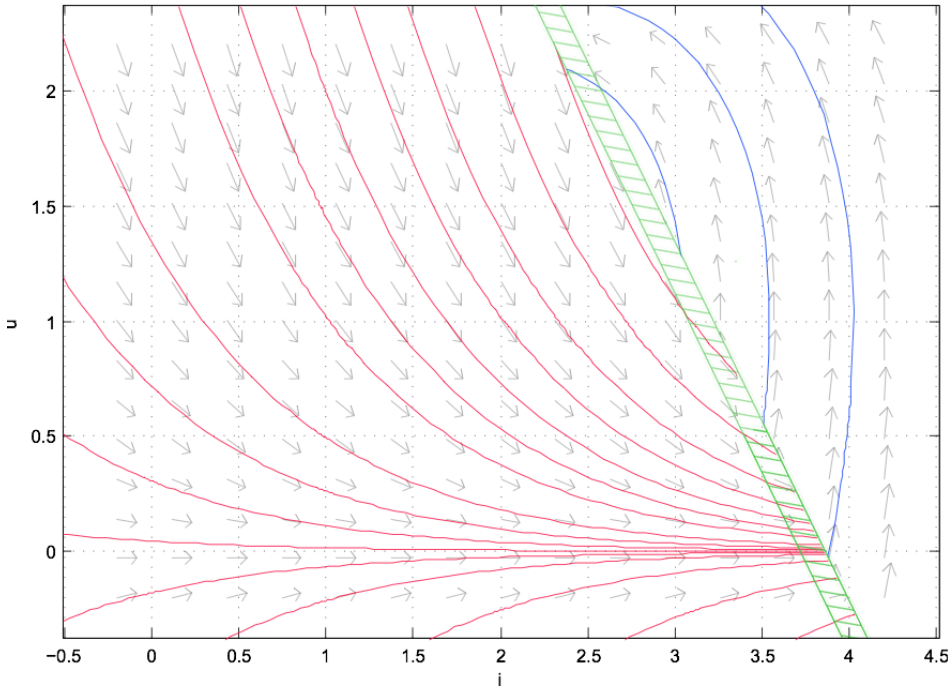

Figure 17: Boost control with 2D sliding surface

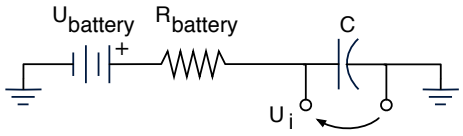

Figure 18: Input Low Pass Filter Capacitor

3.3.1 Buck Mode

The control for Buck mode is implemented exactly as in Section 3.1.3. Using the notations from Figure 9, the A_{up} or B_{up} is switched, depending on direction.

3.3.2 Boost Mode

The boost mode sliding mode control was not easily implemented from the boost circuit, since the control had to be regulated along (at least) two dimensions (as stated in Section 3.2.3). To configure the 4QC for boost operation, the lower side switch of the output side is controlled. The only reference available is the output voltage[6]. A decision law for the lower switch will be a function of it solely:

$$\sigma_1 = \mathbf{g}(u_o) = u_o = r_u \Leftrightarrow \sigma_1 = u_o - r_u = 0 \tag{14}$$

with the corresponding control law for switch S_{down}:

$$S_{down} = \begin{cases} 0 & , \quad \sigma_1 > \delta \\ 1 & , \quad \sigma_1 < -\delta \end{cases} \tag{15}$$

[6]Or current, the same argumentation is applicable since $i \propto u$ for a (on a small enough timescale) resistive load.

However, the current in the inductor must be higher than a certain value for this control to apply, which can be understood from the phase portraits in Section 3.2.2. If not, the system will never have a higher output voltage than is available at the input terminals.

To ensure this condition, assume the 4QC could be realised with the inductor current as provided from a constant current source (see Figure 19).

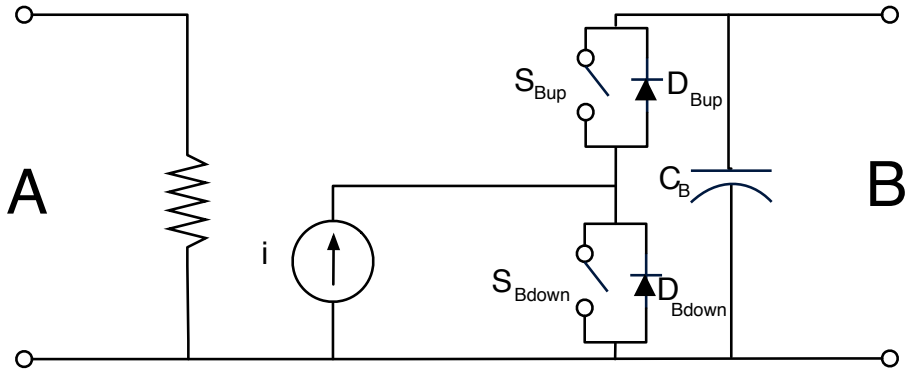

Figure 19: 4QC modeled with Current Source

The new state equations (for a system with resistive output R at side B) become simple.

$$\begin{cases} \frac{di}{dt} &= 0 \\ \frac{du_o}{dt} &= \frac{i}{C}(1 - S_{down}) - \frac{u_o}{RC} \end{cases} \tag{16}$$

Also assume the current source is powerful enough to provide the output power. Assume: $S = 0$, then

$$\frac{du_o}{dt} > 0 \tag{17}$$

must be true in order for the current source to be able to raise the output voltage. The second equation in Equation 16 is used together with Equation 17 to create the following condition:

$$i > \frac{u_o}{R} = i_o \tag{18}$$

The flow from the current source must be larger than the extracted output current, however how much larger depends on the operating conditions (since the equivalent R varies with input/output dynamics). The current reference is considered a tuning criterion.

Boost Control Law The boost control problem is now reduced to the problem of realising a current source, with help from the other switches in the 4QC-circuit. Let the boost switch (S_{down} on the ouput side) be conducting. The current in the inductor will now increase indefinitely. By switching S_{up} on the input side, the inductor current can be kept inside a hysteresis band. If the current source criterium above (Equation 18) is fulfilled, the boost switch can now start to operate and let the ouput draw current. As long as the current source is powerful enough, the boost mode will operate in a stable manner. A refence value for the current in the inductor must be chosen, in order for the system to be stable. Usually, if this value is chosen high enough, there will be no problem for the system to deliver power[7].

A decision law for input S_{up} is defined as:

[7]In order to reduce resistive losses in the inductor, the inductor current reference value should not be set unnecessarily high. It is possible to construct some PID-based method for choosing this value. Since only the efficiency (and not the output power) is affected by the suggested memory-based controller that would have to be introduced, no problems with a change of quadrant would arise. However, optimizing efficiency is not in the scope of this report.

$$\sigma_2 = \mathbf{g}(i_L) = i_L = i_{Lref} \Leftrightarrow \sigma_2 = i_L - i_{Lref} = 0 \tag{19}$$

and a new control law for both input S_{up} and output S_{down} is constructed:

$$\begin{bmatrix} S_{down} & S_{up} \end{bmatrix} = \begin{cases} \begin{bmatrix} 1 & 0 \end{bmatrix} & , \quad \sigma_1 > \delta_1 \bigwedge \sigma_2 > \delta_2 \\ \begin{bmatrix} 1 & 1 \end{bmatrix} & , \quad \sigma_1 > \delta_1 \bigwedge \sigma_2 < \delta_2 \\ \begin{bmatrix} 0 & 1 \end{bmatrix} & , \quad \sigma_1 < \delta_1 \bigwedge \sigma_2 > \delta_2 \\ \begin{bmatrix} 1 & 1 \end{bmatrix} & , \quad \sigma_1 < \delta_1 \bigwedge \sigma_2 < \delta_3 \end{cases} \tag{20}$$

Hysteresis Band Three different hysteresis band widths are introduced in the control law (δ_1, δ_2 and δ_3) and are implementation-specific tuning variables. Generally, the following condition should hold:

$$\delta_3 > \delta_2 \tag{21}$$

The condition can be justified as follows: The fourth control law mode represent a lack of energy in the system, where no power is supplied to the ouput and the inductor is being charged. The output switch is actually overridden to charge the inductor (even though the output reference is below its hysteresis band). Leaving the output unattended like this to power the inductor may cause unwanted disturbances in the ouput. Therefore, the fourth mode should not be active other than in a startup situation. By having δ_2 smaller than δ_3, the inductor will be charged before the output is deprived of energy. If the system finds itself in the fourth region repeatedly, it is likely that the condition presented in Equation 18 is not fulfilled.

3.3.3 Quadrant Transitions

The control laws for single quadrants above must handle quadrant transitions well. The following paragraphs discuss these.

From buck to boost When changing from buck to boost, the inductor must first be powered up to its reference value. While this is happening, the system output will drop.

From boost to buck The excess energy stored in the inductor will be spilled out in the load, but this should not be very significant.

Current direction When a change of power direction is required, the voltage drop over the inductor is reversed and the current changes accordingly.

3.3.4 Current and Voltage Control

Current and voltage control are equivalent if the output are assumed to be resistive on a short timescale, since the voltage drop is then proportional to the current. This means that the same dynamics (as derived above) apply to both current and voltage control. The desired control variable is then measured, and fed into the control algorithm, regardless if it's voltage or current.

4 Simulations

Simulations were performed in Matlab Simulink for boost, buck and 4QC circuits.

4.1 Buck Converter

The simulation for the buck converter was done with a constant duty ratio PWM generator as control for the switch (with duty ratio at 70%). The simulation parameters are noted in Appendix A.2.1 and results are presented in Figure 20. A simulation with a sliding mode controller[8] was also done (seen in Figure 21), and a comparsion of the rise time is seen in Figure 22.

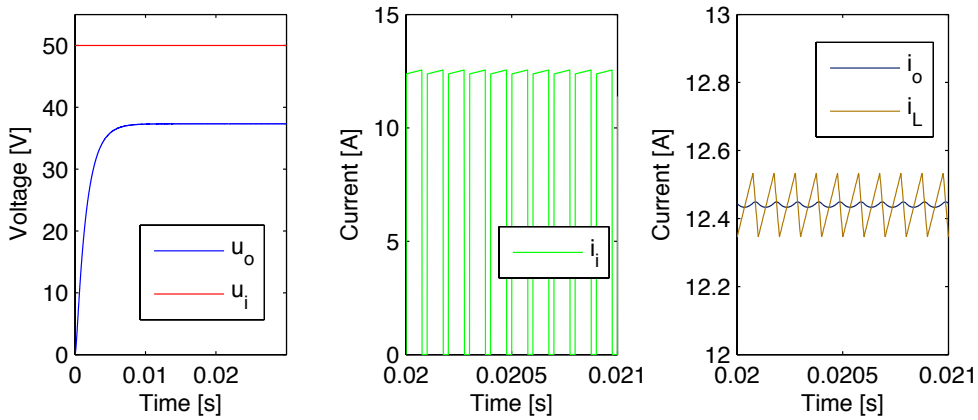

Figure 20: Buck Simulation, constant duty ratio

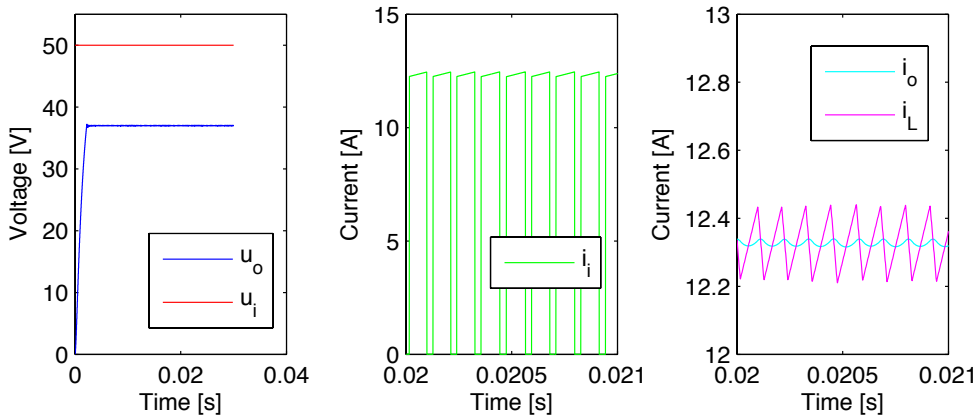

Figure 21: Buck with Sliding Mode Control

[8]The Simulink Block Diagram for the Sliding Mode Controlled Buck Converter can be found in Appendix A.2.2.

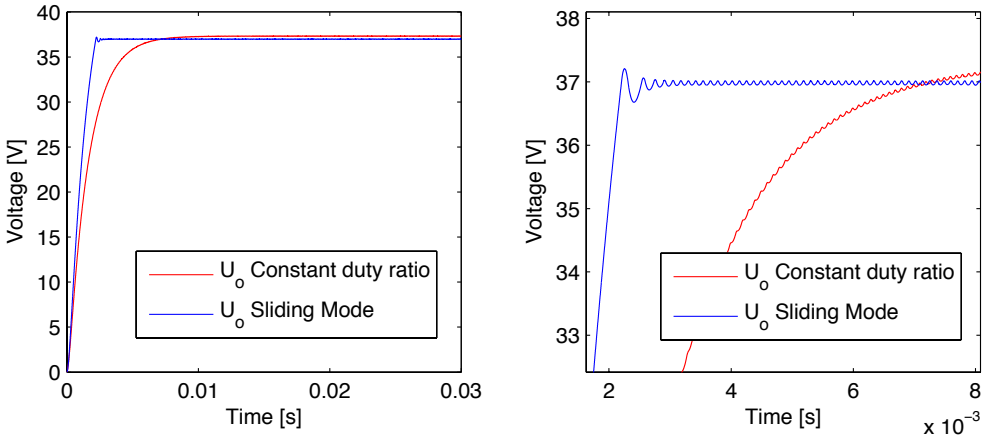

Figure 22: Comparsion between constant duty ratio and SMC for buck converters

Duty ratio comparsion The comparsion in Figure 22 displays that the optimal nonlinear regulator is fast. Even though the correct duty ratio is chosen in the left graph, the rise time is slower. A duty ratio comparsion can be made between the system rise time for constant duty ratio, PID and Sliding Mode (see Figure 23). Sliding mode does not incorporate duty ratio as such but, when close to the decision surface, the period time T and duty ratio can be estimated[9]. As seen from the figure, the equivalent duty ratio for sliding mode control is 100%, which means more energy is transported into the system. Therefore, the rise time must be faster than for constant duty ratio. On the other hand, a PID-controller can increase its duty ratio over time to decrease rise time.

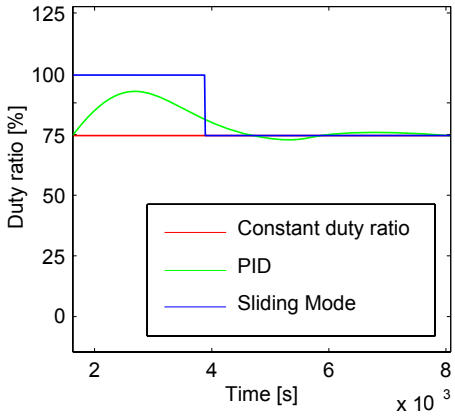

Figure 23: Duty Ratio Comparsion

4.2 Boost Converter

The simulation for the boost converter was done with the parameters marked in Appendix A.2.3, and the control signal was a PWM wave with a constant duty ratio. The results from the simulation can be seen in Figure 24 and as expected the output voltage is higher at the output than the input.

[9]Comparsions between the middle graph in Figures 20 & 21 shows similarity between PWM and Sliding Mode when close to sliding line.

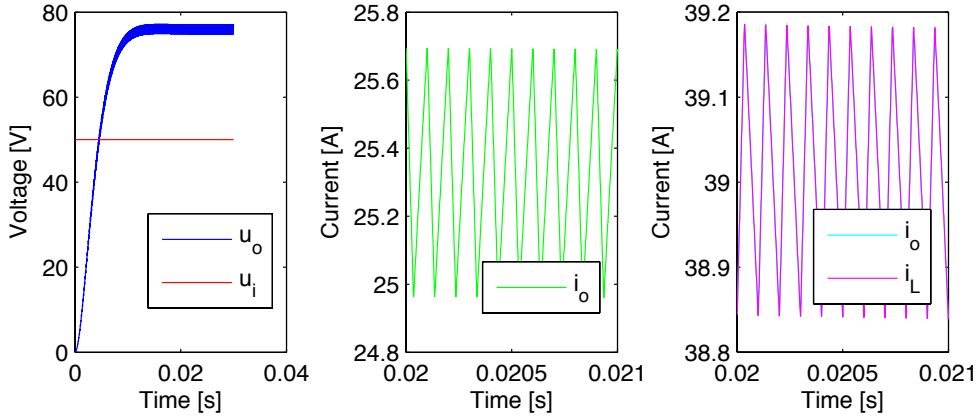

Figure 24: Boost simulation, constant duty ratio

4.3 4 Quadrant Converter

Buck The simulation model used for the 4QC with parameters can be seen in Appendix A.2.4, with control system set for buck mode. A model of the 1D control system (presented in Section 3.1.3) is shown in Appendix A.2.5.

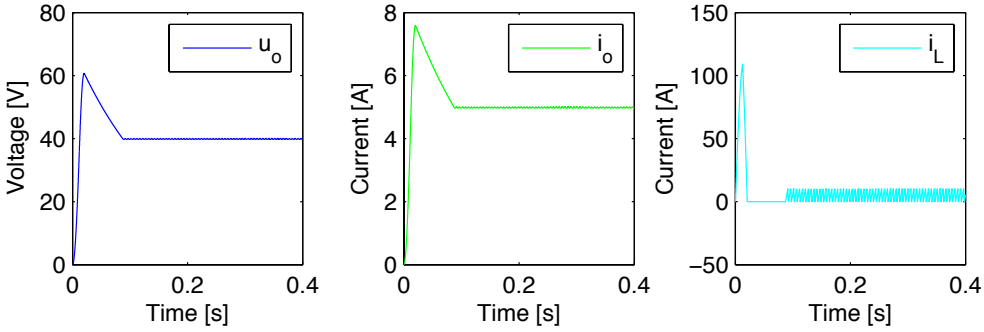

Figure 25: Simulation Results for Buck Control (1D)

Simulations showed the results presented in Figure 25. The overshoot effect is due to the large size of the capacitor, since it requires a lot of charge to reach desired voltage, the inductor will be overcharged. To avoid this, a maximum inductor current can be defined, which means that whilst the capacitor is being charged fast, the inductor will never store so much energy that there will be a large overshoot. Defining a maximum current also provides safety for the inductor component. The block diagram for the modified control system is presented in Appendix A.2.6. The rise time results have no overshoot with the second control suggestion, as is shown in Figure 26.

Boost The boost system is implemented as in Section 3.3.2, and the control block diagram is presented in Appendix A.2.7.

The corresponding control results are as presented in Figure 27.

The results imply that the boost control is working properly. Note that the inductor current goes low when the output voltage becomes higher than the input voltage. However, when the system has reached steady state, the inductor current is limited.

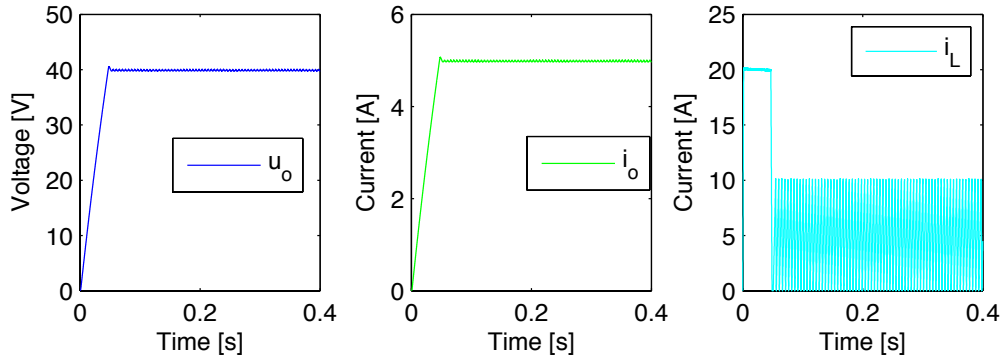

Figure 26: Simulation Results for Buck Control (2D)

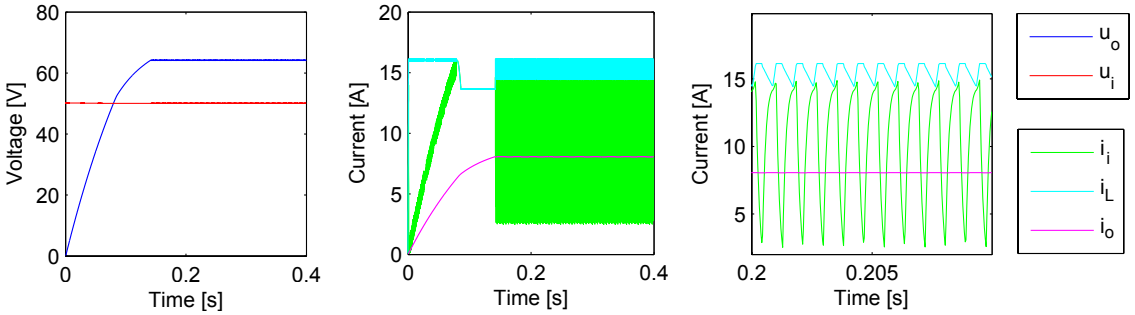

Figure 27: Boost Simulation Results

5 Implementation

A prototype 4QC was built, with the control system proposed in the modelling and simulations sections above. The control strategies are not voltage or current level specific, but components were chosen so that voltage levels could be raised above the maximum test voltage of 50 volts. For example, the IGBT switches can handle several hundred volts. The final circuit scheme can be seen in Appendix A.5 and a list of major parts can be seen in Appendix A.4.

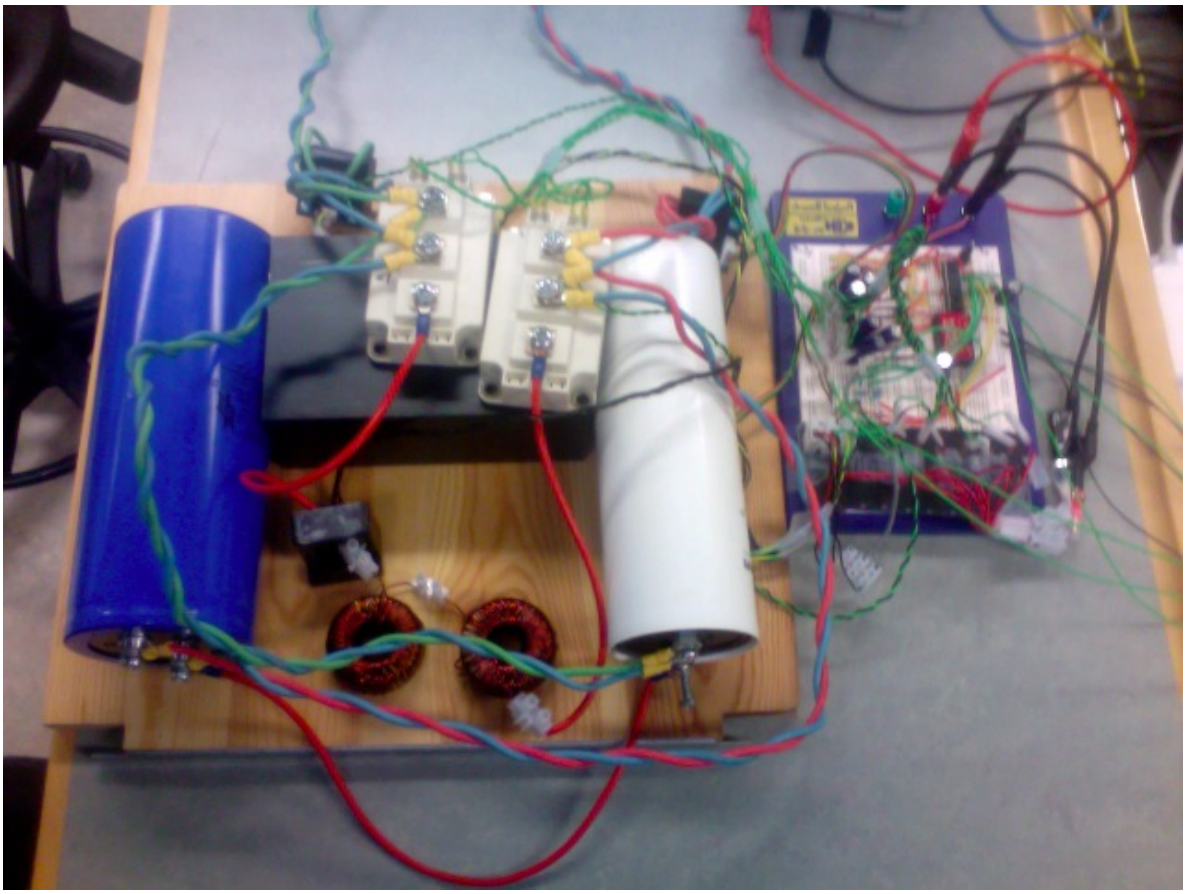

Figure 28: Photograph of converter prototype

5.1 Microprocessor Code

The processor was programmed with *MpLab 8.13*, and the interface between computer and chip was an *ICD2* unit. The code was written in C, and compiled in the chip manufacturer's compiler *C30*. The final code can be found in Appendix B. The code is divided into seven sections, namely:

1. Preprocessor Instructions
2. Variables & Definitions
3. Prototypes
4. Init Functions
5. Interrupts
6. Functions (setters)
7. Main Loop

Configuration The first four sections are configuration details, clock frequency, allocation of memory and so on. The *Init Functions* section deal with peripherals (such as the ADC) and their configuration. The last three sections contain the operative code. Two empty timer interrupts can be seen in section 5, and they provide functionality for circuit testing.

Runtime Description

ADC Every iteration in the microchip starts with the ADC finishing a measurement of its five inputs, AN1 to AN5. The ADC-interrupt is then called and its only function is to set a "Measurement Done"-bit which will make the Main Loop perform certain things the next time it is starting over.

The Main Loop When discussing the interior of the Main function, everything inside the *while-statement* is relevant at runtime and the rest are startup calls. The main loop polls the "Measurement Done"-bit until a new measurement has been done and then starts the main computing stage.

- Firstly, a median filter is applied (discussed in a paragraph below), which means the new measurements are considered, and then an estimation of the real value is produced. This is the first subsection of the main loop.
- The second subsection deals with how to decide whether to use buck or boost mode. The Boost mode is used when uncertain, which means the current in the inductor is kept high. A transition to buck mode is done when voltage measurements note that the output voltage drop is lower than in the input. Then buck mode is employed, and the excess inductor energy is dumped into the output. If the current reference is not reached, the circuit will automatically go to the boost mode.
- The third subsection applies the buck or boost control to the switch control ports. The implementations are the ones used in the Modelling and Simulations sections above, except for a small modification. When running the circuit with the simulated control systems, small glitches in the output were detected, together with audible "clicks". These came from the capacitors (which seemed to be shorted for very short durations). Measurements showed that the "clicks" occured when both switches were operated simoultaneously. Therefore, very small delays were introduced to the control system so that two switches would never change state in the same instant, and this protection layer removed the "click" symptoms.
- The fourth subsection stores the decisions, and makes sure the output pins RE0 to RE3 contains the correct states.

5.2 Current Sensor Filtering

The current sensor signal was measured at a high frequency, and as Figure 29 suggests, the current sensor suffers from shot noise. A median filter was employed to increase precision of the current measurements.

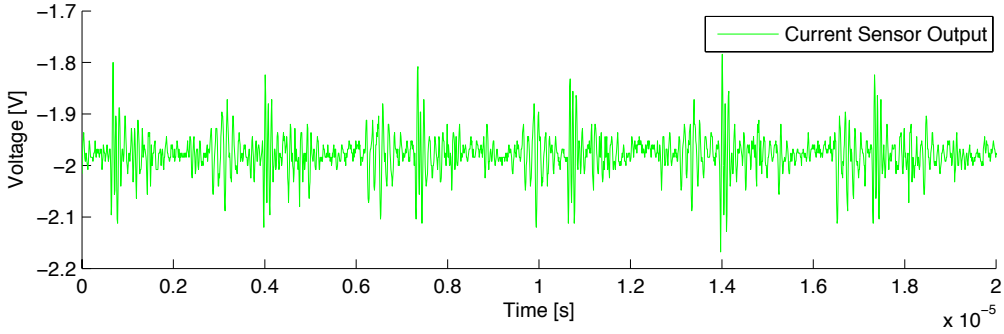

Figure 29: Current-Sensor-Signal

Median Filter The median filter contained 8 datapoints of the most recent measurements. The median of the dataset was extracted and returned as output. Theoretically, a small time-delay is introduced to the measurement system due to the data buffer, but this can be circumvented with high sampling frequencies. The calculation time required by the control system set the limit for the measurement systems bandwidth to 80kHz, where the buffer is completely refreshed at a 10kHz rate.

The implementation in code of the filter was done in two separate ways. For small filter sizes (such as 8 data points), the fastest one was implemented by resorting the entire buffer at every time step. The number of instructions were then kept constant and also all relevant code and data could be kept inside the processor (and thus, no data read delay). The other implementation was created for larger filter sizes, with variable computing length and partly pre-sorted lists. The philosophy of the second algoritm was: Write over the oldest data (in the set with pre-sorted values) with the latest measurement, and then move it up or down until the list is sorted again. When considering larger datasets, the average time for this method is smaller than for the constant time sorting algoritm.

6 Experimental Results

6.1 Transition from buck to boost

In a situation where the flywheel is sending all of its energy to the battery, the converter would see a constantly decreasing input voltage since the back-emf decreases with rotational speed. A measurement was made with the flywheel placed at the input of the system, and a constant output current was fed through a resistive load.

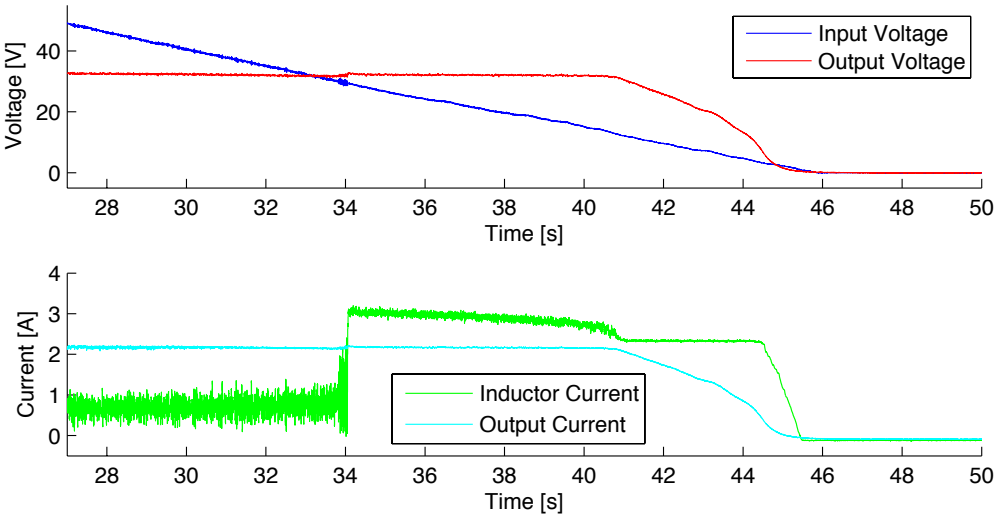

Figure 30: Transition from buck to boost

The transistion from buck to boost can be seen between 33 to 35 seconds (in Figure 30). When the system enters boost mode the inductor current is lifted to a higher level (seen in the second graph). At 45 seconds the output voltage is decreasing, because the inductor is not recharged enough. To keep the output voltage stable for a longer while, a higher inductor current reference must be chosen. The power relations (discussed in 2.1.1) explain why the input current must go up when the input voltage decreases.

6.2 Transition from boost to buck

A measurement was done with an increasing input voltage source and the output current fed through a resisitive load. The transition from boost to buck can be seen between 14 to 16 seconds in the figure. The inductor current drops when the system enters buck mode. The same phenomena as in Figure 30 can be seen when the output voltage is lower than the reference.

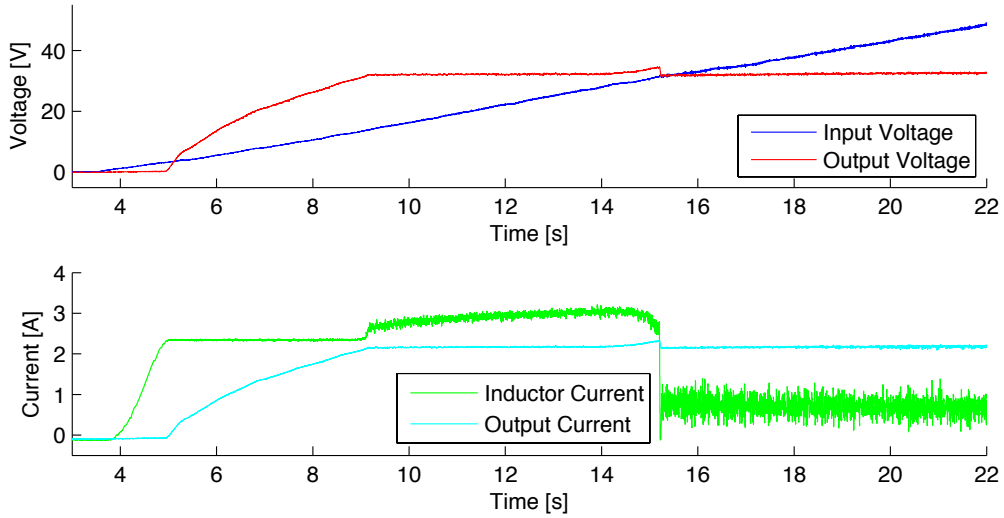

Figure 31: Boost to buck

6.3 Change of Current Direction

In this setup, 12V batteries were placed at each input. The current limit was 5 ampere, and the energy direction was changed on a regular interval (seen in Figure 32). The inductor current show the different directions of the power transfer, note that the transition from positive to negative is done without disturbances.

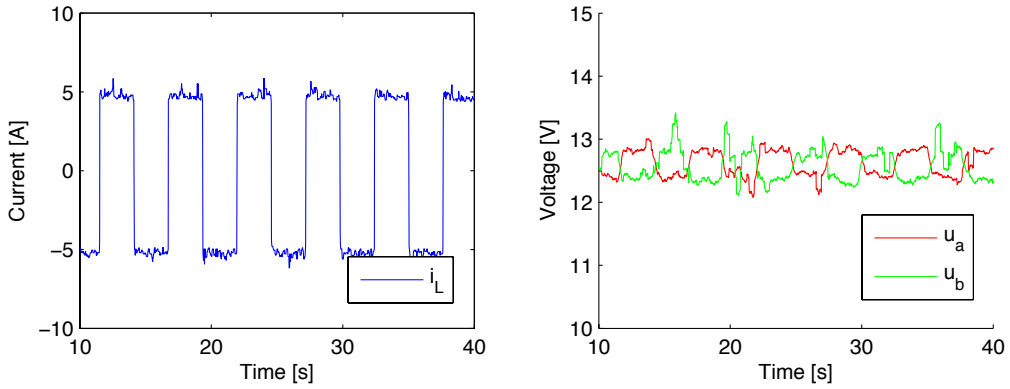

Figure 32: Current direction transitions in inductor (left graph), and voltages on A- and B-sides of the converter (right graph)

6.4 Fast changes of operating quadrants

The power throughput in this measurement was changed on regular intervals by the microprocessor, and the input source of power was the flywheel. The results can be seen in Figure 33. Notice the quick changes between boost and buck mode. Fast changes of operating quadrant can be seen, and the transitions are without significant disturbances.

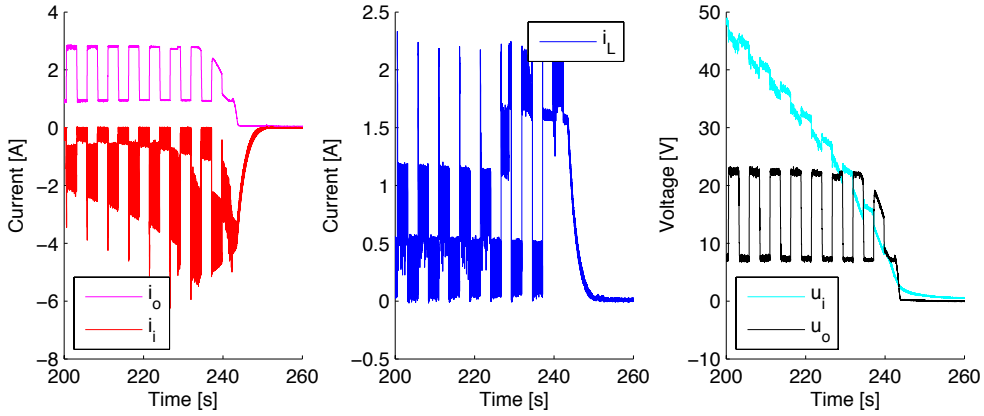

Figure 33: Fast changes of operating quadrants

6.5 Median Filter Accuracy

The efficiency of the current filter was established by letting the dspic30F2010 determine if the measured current was above a certain treshold value. The non-filtered and median filtered treshold output can be seen as a function of the current in Figure 34. Note that the median filtered signal has been biased (for clarity). A smaller region of uncertainty in the figure corresponds to a higher degree of precision. Note that the accuracy remains constant, even though the precision is increased.

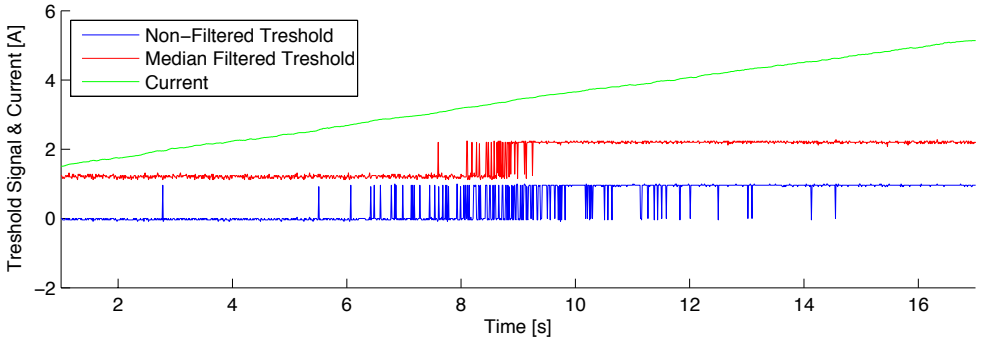

Figure 34: Treshold of current value

7 Discussion and Conclusion

The 4-quadrant DCDC Converter can be controlled in such a fashion that the transition between operating quadrants is nearly seamless, with the proposed non-linear regulator. The control was based on Sliding Mode and used a current-source simile to design a control for the boost mode. Switching between quadrants (operating modes) was facilitated by removing memory dependence throughout the control (made possible by the current-source). Parameter invariant control (i.e. no steady-state errors) was obtained by basing the control on the principal behaviour and thus ensuring convergence.

The proposed control system handles quadrant transitions well, as seen in the results section. The provided microprocessor code can handle current control and voltage control. The device topology is fully bidirectional, and there were no problems changing direction of the power.

A prototype was constructed, which required a lot of trial and error. Much of the work was focused on an efficient control system. However, the implementation itself has room for improvements. Especially, making the drivers run with four switches connected was difficult, and more effort could be made in optimizing switching times.

Experiments were performed connected to the flywheel, to test performance under conditions relevant to the EV system presented in the Introduction. The converter produced a stable current to the output even though the emf varied from 50 to 0 volts in the flywheel, which is useful for battery charging.

Sliding mode is dependent on high current measurement bandwidths, which required some extra work on a fast filter. More sophisticated methods could be employed, such as Kalman state estimators (which could incorporate input/output dynamics).

Future work The prototype should be built for higher powers, so that efficiency can be measured and optimized. The device should have a defined rating, so that specific components could be bought/-constructed. Specifically, the inductors needs to be able to handle higher current. Better drivers should be built to improve switching frequency.

References

[1] D. Naunin. Electric vehicles. *IEEE International Symposium on Industrial Electronics, pp.11-24*, 1996.

[2] Nihal Kularatna. *Modern Component Families and Circuit Block Design, chapter 9.* Newnes, 2000.

[3] Nihal Kularatna. *Modern Component Families and Circuit Block Design, chapter 9.7 figure 9-11.* Newnes, 2000.

[4] J. Santiago, J. G. Oliveira, J. Lundin, J. Abrahamsson, A. Larsson, and H. Bernhoff. Design parameters calculation of a novel driveline for electric vehicles. *World Electric Vehicle Journal Vol. 3*, 2009.

[5] Richard M. Stephan, Rubens de Andrade Jr., and Guilherme G. Sotelo. Third generation of flywheels: A promising substitute to batteries. *Eletronica de Potencia, vol. 13, no. 3*, August 2008.

[6] P.H. Mellor, N. Schofield, and D. Howe. Flywheel and supercapacitor peak power buffer technologies. *IEE Colloquium (Digest), n50, pp.47-52*, 2000.

[7] B. Szabados and U. Schaible. Peak power bi-directional transfer from high speed flywheel to electrical regulated bus voltage system: a practical proposal for vehicular technology. *IEEE Transactions on Energy Conversion. V. 13, Issue: 1, pp. 34- 41.*, March 1998.

[8] Mecrow, J.S.Burdess, J. N. Fawcett, J.G. Kelly, and P. G. Dickinson. Design principles for a flywheel energy store for road vehicles. *IEEE Transactions on Industry Applications. V. 32, No. 6. pp. 1402-1407*, November/December 1996.

[9] J.P Agrawal. *Power Electronics Systems: Theory and Design, chap. 6.* Prentice-Hall, 2001.

[10] N. Mohan, T. M. Undeland, and W. P. Robbins. *Power Electronics: Converters, Applications, and Design, 2nd ed.* John Wiley & Sons, New York, 1995.

[11] Timothy L. Skvarenina. *The Power Electronics Handbook, chapter 2.2.* CRC Press, 2001.

[12] F. Caricchi, F. Crescimbirii, and A. Di Napoli. A 20 kw water-cooled prototype of a buck-boost bidirectional dc-dc converter topology for electrical vehicle motor drives. *Applied Power Electronics Conference and Exposition, 1995. APEC '95. Conference Proceedings 1995., Tenth Annual*, 1995.

[13] Timothy L. Skvarenina. *The Power Electronics Handbook, chapter 8.* CRC Press, 2001.

[14] V. I. Utkin. *Sliding Modes and Their Application in Variable Structure Systems.* MIR Publishers, Moscow, 1978.

[15] U. Itkis. *Control Systems of Variable Structure.* John Wiley & Sons, New York, 1976.

[16] Timothy L. Skvarenina. *The Power Electronics Handbook, chapter 8.6.* CRC Press, 2001.

A Appendix

A.1 Linear Control - Open Control & PID

The Linear Approximation In order to control systems by classic linear control theory, linearisation can be done by *state space averaging* where the duty ratio is treated as an input signal. A linearized system can be used to create different control systems, among them Open Control and PID. The following paragraphs discuss why these solutions aren't feasible for the 4QC.

Open Control The simplest consideration for a control system is open control, where the reference r for the target output is used to calculate an appropriate duty ratio. The control model is presented in Figure 35.

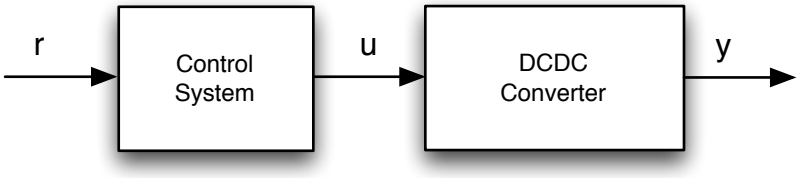

Figure 35: Open Control Model

For the buck converter the implementation could be based on its transfer function $U_o = dU_i$. Just let the control signal u be:

$$u(r) = d = \frac{U_o}{U_i} = \begin{cases} \frac{r}{U_i} & , \quad r = U_o \\ R\frac{r}{U_i} & , \quad r = I_o \end{cases} \tag{22}$$

If the input voltage U_i is measured, the appropriate duty ratio can be calculated for a voltage control mode by inserting the reference value as seen in equation 22. If operating in current control mode, Ohms Law will allow the equivalent current to be predicted (seen in the same equation). Losses can be treated as a constant voltage drop in the input voltage.

Drawbacks

- The Converter losses are assumed to be constant, and if they vary, the output will be biased.
- The output is assumed to be constant resistive, but the electric machine will change its emf with its rotational speed.

Open control assumes much about the system that cannot be taken for granted. The switching and component losses are assumed to be zero and (more importantly) constant. Also, the output is assumed to be purely resistive and constant. The input is assumed to be an ideal voltage source without a limited charge and no internal resistance.

When sending energy to the flywheel, one has to know that an electric machine will have a very low equivalent resistance when at rest, and a very high one when it is rotating (when the back-emf has gone up). On the other side of the system, when sending energy to the battery, it cannot be treated as a constant load, since its own emf will differ depending on many physical factors (state of charge, temperature, previous load cycle conditions).

Of course, actions could be taken to account for output biases in the control function $u(r)$. The deviations are, however, depending on very many variables and can even be a potential danger if the system meets operating conditions not fully accounted for. If there is even the slightest modeling error, the system will never reach the reference value.

Conclusion Simple to deviate and implement, but for a vehicle driveline application, the input and the output of the devices will have so large dynamics that some kind of feedback control has to be implemented.

PID A much more robust way of controlling the system is to look at the output error, defined as $e = y - r$. For linear systems, there is a lot of theory done in this field (called LINEAR CONTROL, PID CONTROL etc). Even though the system is comprised of discrete switching states, the input control variable can still be treated as continious, if the input to a PWM-WAVE is used.

PID control is basically an error based control method, where information from the output of the system is used to determine the error of the desired output control variable (see Figure 36). This concept is also called FEEDBACK CONTROL, or CLOSED-LOOP CONTROL.

PID is equipped with three different components all influencing the control decision. These are called the PROPORTIONAL, the INTEGRAL and the DERIVATIVE part of the control. The control law looks like this:

$$u(t, e(t)) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(\tau)}{d\tau} \tag{23}$$

The proportional term looks at the momentaneous error, while the integral term monitors the error over time. The derivative term looks at the change of the error, and can slow down the control to account for momentum problems (overshoot).

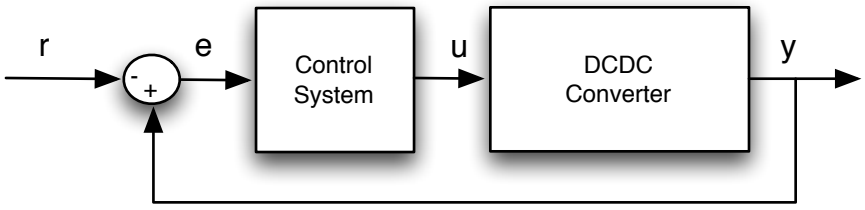

Figure 36: Feedback Control Model

Optimal - Linear - Regulator The theory of the optimal PID-regulator is well known. The system dynamics are modelled, and then the three values can be computed following standard procedure, to produce the optimal response time and system accuracy.

Drawbacks In this application, PID suffers from a few weaknesses.

- It's optimality is based on a linear model. Switching converters can be modeled as linear, but these models are only approximations of their behaviour. Even for ideal components, discrete switches in the system will make it non-linear! Even if PID can be the optimal linear regulator, there might exist an even better optimal regulator (in the sense that it can be non-linear).
- The optimal (linear) configuration may become inoptimal when the input and output dynamics vary much, since optimality is based on constant input/output dynamics (discussed in 2.2.2).
- Also, when using a four-quadrant topology such as the one in section 2.1.3, the memory of the controller (the integral part) will become nonrelevant when moving between quadrants. Therefore, even though the operation in buck or boost mode is very good, the transition between them might be problematic. When the dynamics of the input and output are different to the linearized model, the proportional term will be far from optimal. The integral term is then the only thing that removes the system errors, and a transition requires a complete reset of its memory, which is seen as large output disturbances.

Conclusion The PID have trouble handling parameter variations, and changing of operating quadrants. The next step is to account for non-linearity in the feedback control.

A.2 Simulink Block Diagrams

In all the following diagrams, the *Relay* component implements a hysteresis band.

A.2.1 Buck Converter Model with constant duty ratio

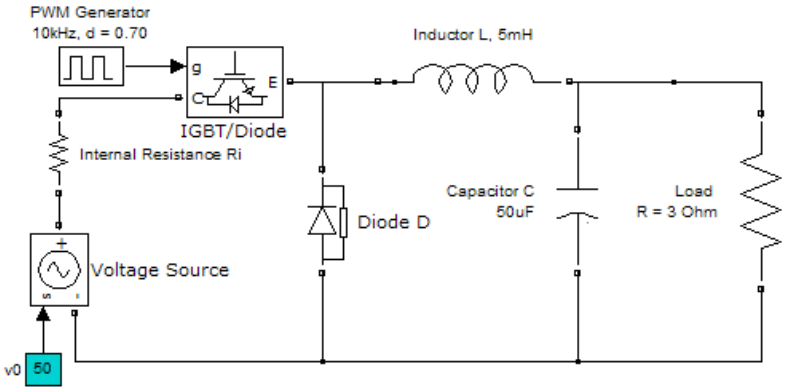

Figure 37: Buck Simulink Model

A.2.2 Buck Converter Model with Sliding Mode Control

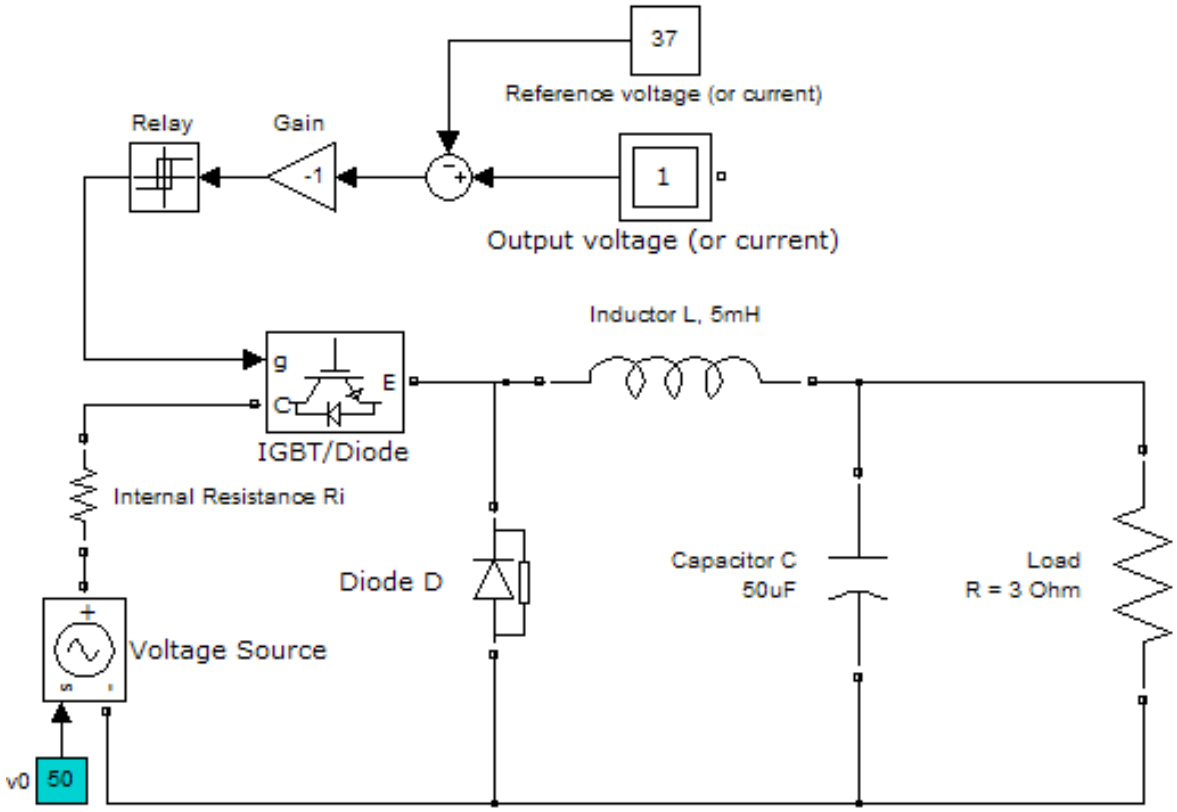

Figure 38: Simulation Block Diagram of Buck Converter with Sliding Mode Control

A.2.3 Boost Model with constant duty ratio

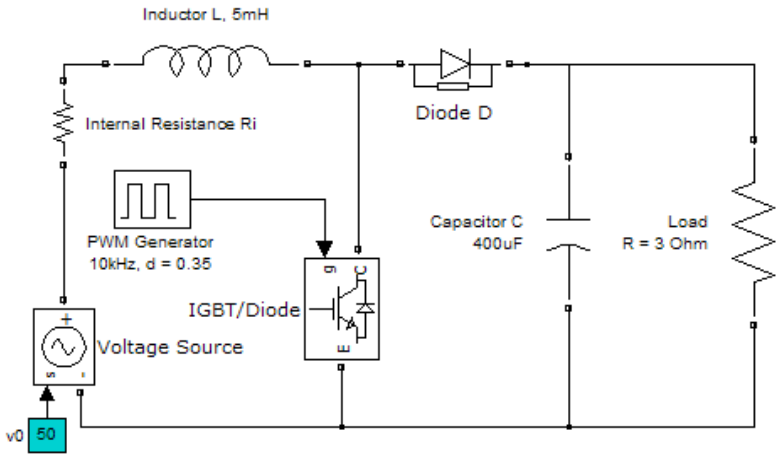

Figure 39: Boost Simulink Model

A.2.4 4QC Model with Buck Control

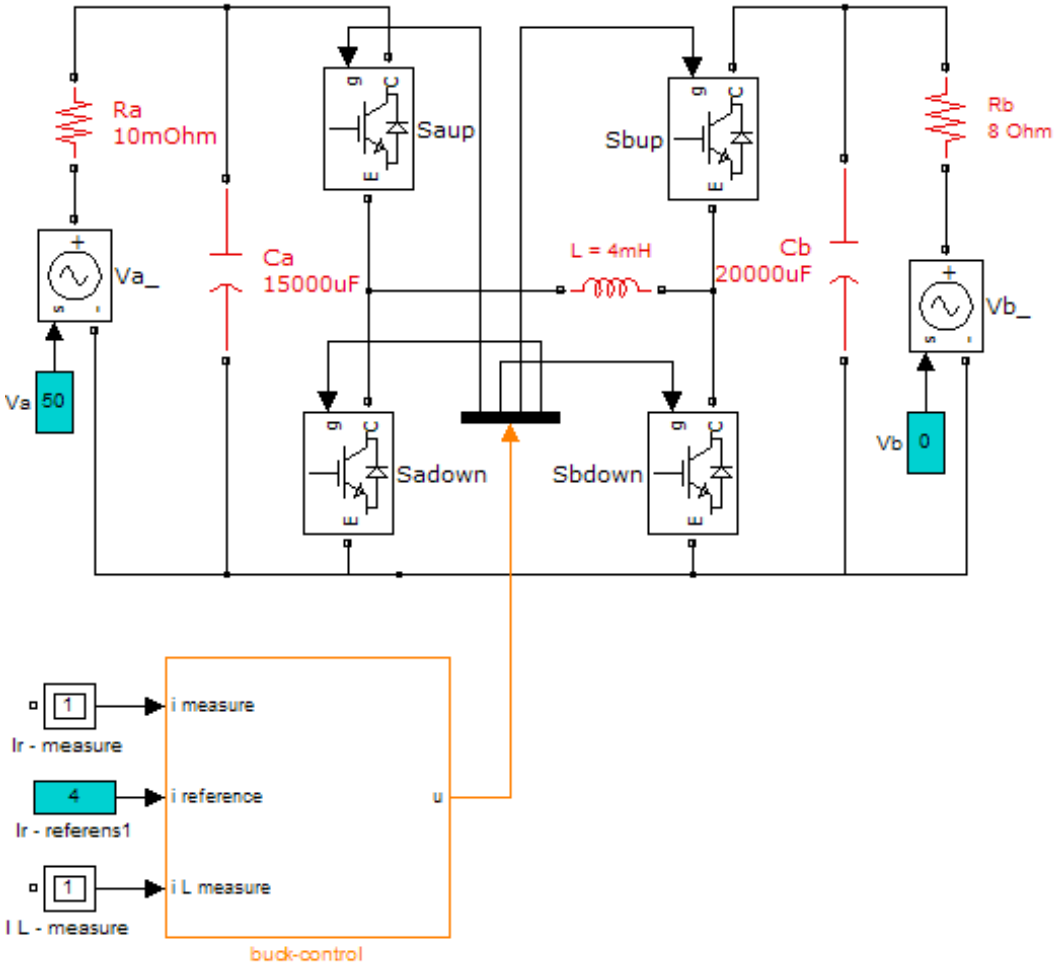

Figure 40: 4QC Simulation Model

A.2.5 4QC Buck Control 1D Model

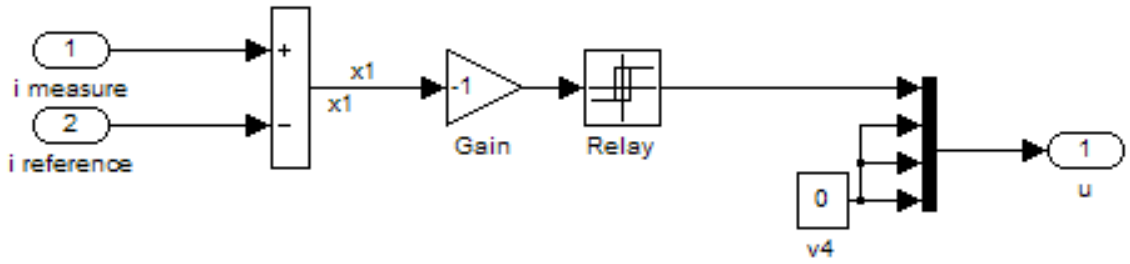

Figure 41: 4QC Buck Control System (1D)

A.2.6 4QC Buck Control 2D Model

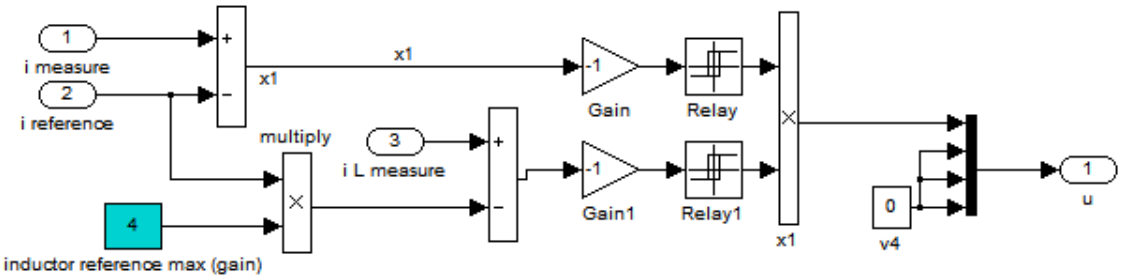

Figure 42: 4QC Buck Control System (2D)

A.2.7 4QC Boost Control Model

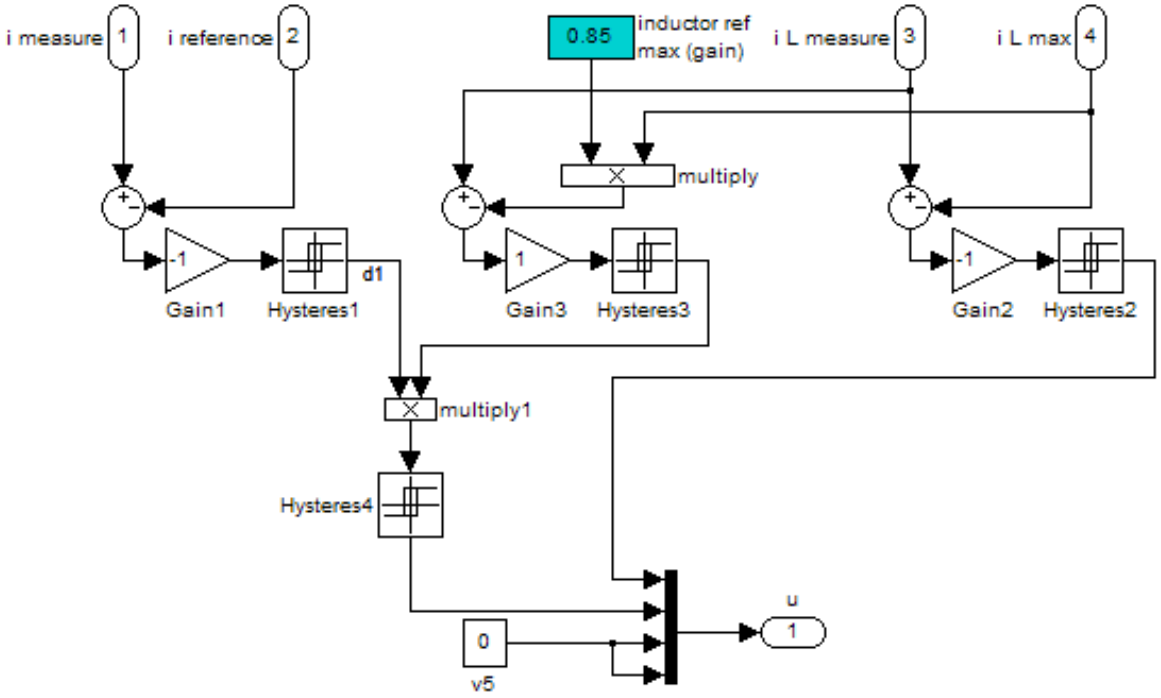

Figure 43: 4QC Boost Block Diagram

A.3 4QC in buck mode compared with Buck Circuit

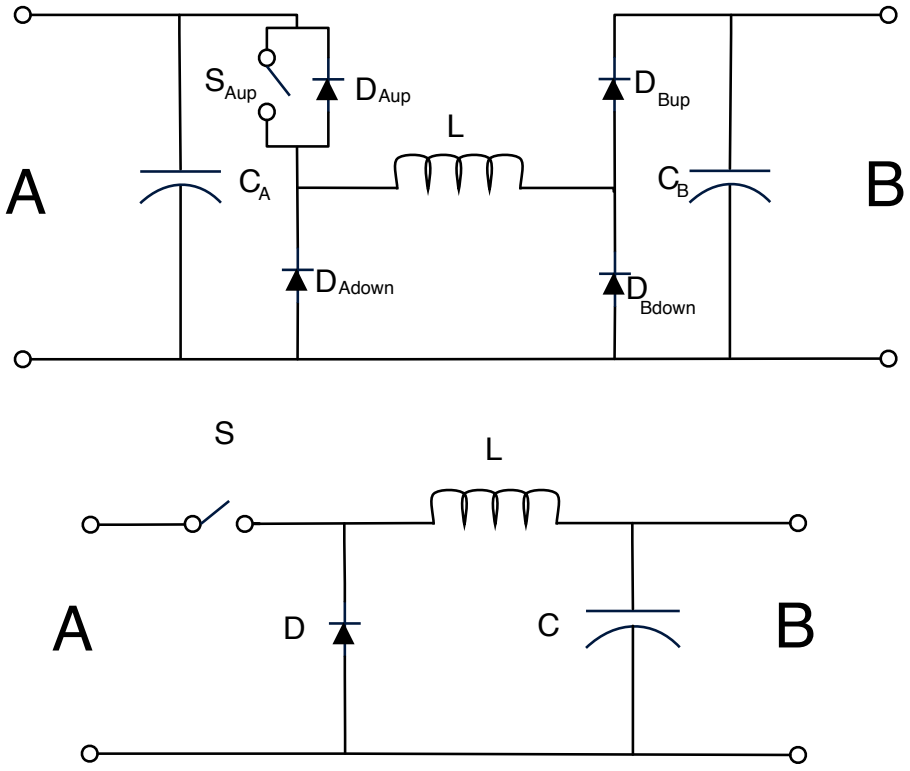

Figure 44: 4QC in buck mode compared with Buck Circuit

A.4 Parts List

Part	Description
Semikron SemiSel dual IGBT SKM 600GB066D	To make sure the converter can be scaled up for higher power ratings, high capacity switches were used. Each case contains two switches, and each one is fitted with antiparallell power diodes. [ref datasheet]
IR2110 Gate Driver	The gate driver used was a Hi & Lo-side driver, and since the 4QC requires four drivers, two casings are required. The driver power interfaces were supplied by floating power supplies (Traco Power, see below). [ref datasheet]
Traco Power	A floating power supply, provides a floating ground and 15 volts dc.
dspic30F2010 Digital Signal Processor	The microprocessor from Microhip Corp was used. Specifically the ADC and processor for filtering and control was used. [ref datasheet]
Voltage Measurement	Done with the inbuilt ADC of the dspic30F2010, but scaled down with voltage division over resistors.
HAL-50S Current Transducer	Current was transformed into voltage (which could be measured through the ADC). The current transducer was very linear, but had problems with shot noise distortion (see Figure ?).
Capacitors	Two 200Volts+ rated capacitors were used in the power circuit, with capacitance 15000uF and 20000uF. In the control circuit, stabilising capacitors of 100uF were employed.
Inductors	Two inductor elements in series produced a total inductance of 4mH (measured at 5kHz).

Table 5: Parts List

A.5 Hardware drawing

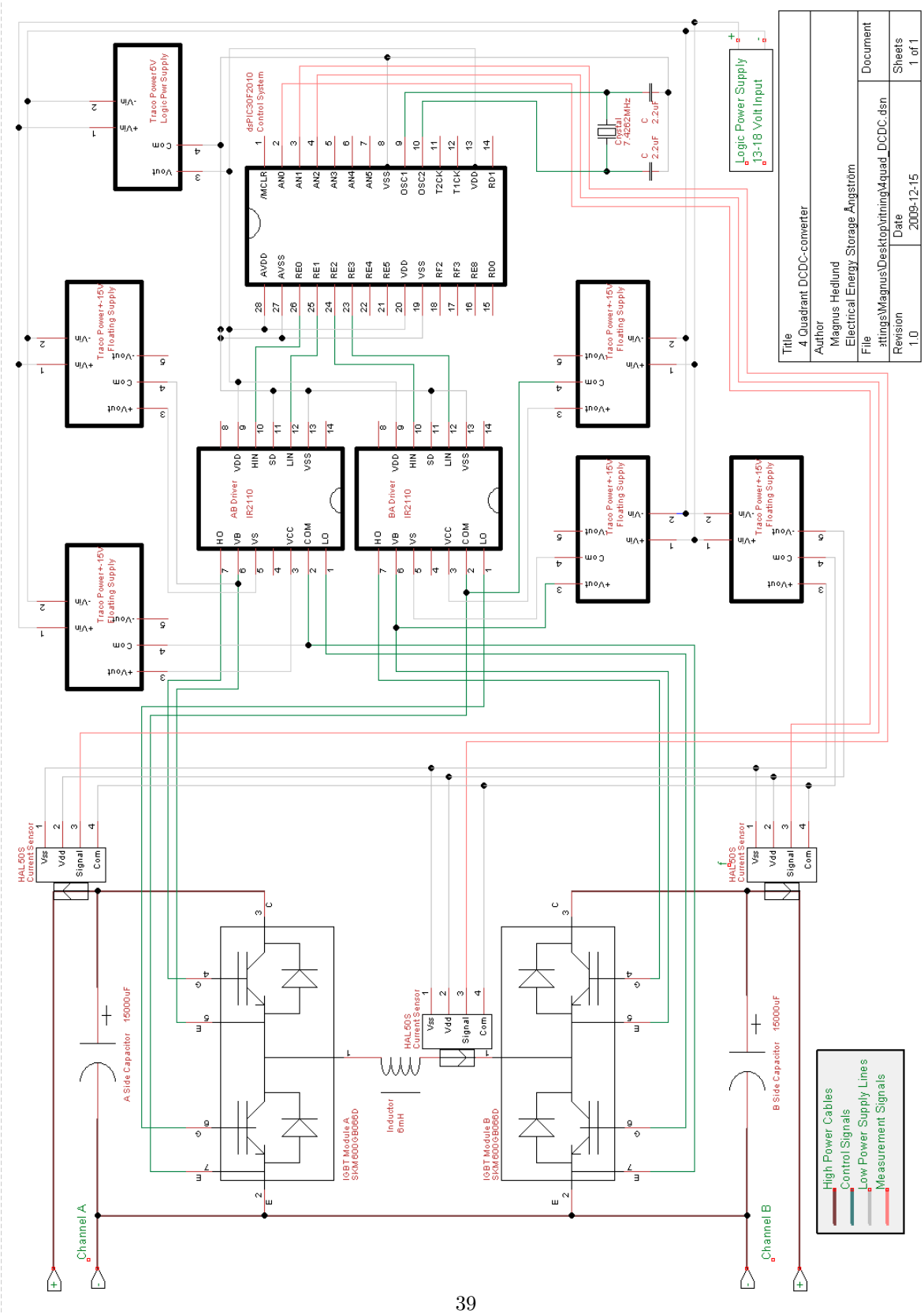

Figure 45: Hardware Drawing

A.6 Buck State Equations

The circuit with necessary notations is shown in Figure 7. The switch is turned on ($S = 1$) and Kirchoff's law around a circuit is applied:

$\sum u = 0 \Rightarrow u_i - u_C - u_L = 0 \Leftrightarrow$

$u_i - u_C - L\frac{di}{dt} = 0 \Leftrightarrow$

$\frac{di}{dt} = \frac{1}{L}(u_i - u_C)$

The same equation for the off-state:

$\sum u = 0 \Rightarrow u_C - u_L = 0 \Leftrightarrow$

$-u_C - L\frac{di}{dt} = 0 \Leftrightarrow$

$\frac{di}{dt} = \frac{-u_C}{L}$

Now apply Kirchoff's law to a node in the system.

$\sum i = 0 \Rightarrow i = i_C + i_R \Leftrightarrow$

$\frac{i}{C} = \frac{i_C}{C} + \frac{i_R}{C} \Leftrightarrow \frac{du_o}{dt} = \frac{i}{C} - \frac{i_R}{C} = \left[i_R = \frac{u_o}{R}\right] = \frac{i}{C} - \frac{u_o}{RC} \Leftrightarrow$

$\frac{du_o}{dt} = \frac{i}{C} - \frac{u_o}{RC}$

A.7 Boost State Equations

The circuit with necessary notations is shown in Figure 8. The switch is turned on ($S = 1$) and Kirchoff's law around a circuit is applied:

$\sum u = 0 \Rightarrow u_i - u_L = 0 \Leftrightarrow$

$u_i - L\frac{di}{dt} = 0 \Leftrightarrow$

$\frac{di}{dt} = \frac{u_i}{L}$

The same equation for the off-state:

$\sum u = 0 \Rightarrow u_i = u_o + u_L \Leftrightarrow$

$\frac{di}{dt} = \frac{u_i - u_o}{L}$

Now apply Kirchoff's law to a node in the system. $S = 0$ is assumed first.

$\sum i = 0 \Rightarrow i = i_o + i_R \Leftrightarrow$

$\frac{i}{C} = \frac{i_C}{C} + \frac{i_R}{C} \Leftrightarrow \frac{du_o}{dt} = \frac{i}{C} - \frac{i_R}{C} = \left[i_R = \frac{u_o}{R}\right] = \frac{i}{C} - \frac{u_o}{RC} \Leftrightarrow$

$\frac{du_o}{dt} = \frac{i}{C} - \frac{u_o}{RC}$

With the switch is $S = 1$.

$\sum i = 0 \Rightarrow 0 = i_o + i_R \Leftrightarrow$

$0 = \frac{i_C}{C} + \frac{i_R}{C} \Leftrightarrow \frac{du_o}{dt} = -\frac{i_R}{C} = \left[i_R = \frac{u_o}{R}\right] = -\frac{u_o}{RC} \Leftrightarrow$

$\frac{du_o}{dt} = -\frac{u_o}{RC}$

The three equations above can be summarized into a state equation, where u_o and i are state variables.

B Processor Code

```
/*
Copyright Magnus Hedlund <vindarmagnus@gmail.com>
This Document is divided in to six sections
        - Preprocessor Instructions
        - Variables & Definitions
        - Prototypes
        - Init Functions
        - Interrupt Functions
        - Main Loop


        About ADC-configuration
        Measurements are being done on AN1 to AN5.
    The results are found in ADCBUF0 to ADCBUF4.
        ADCBUF0 - Current out of B side.
        ADCBUF1 - Current through inductor (defined positive from A to B)
        ADCBUF2 - Current out of A side.
        ADCBUF3 - Voltage on A side.
        ADCBUF4 - Voltage on B side.
*/


// ------------------------------------------------------------------
// Preprocessor Instructions
// ------------------------------------------------------------------
#include "p30f2010.h"

_FOSC(CSW_FSCM_OFF & XT_PLL16);
// Oscillator source and PLLx16
_FWDT(WDT_OFF);
//Turn off the Watch-Dog Timer.
_FBORPOR(MCLR_EN & PWRT_OFF);
//Enable MCLR reset pin and turn off the power-up timers.
_FGS(CODE_PROT_OFF);
//Disable Code Protection



// ------------------------------------------------------------------
// Variables & Definitions Section
// ------------------------------------------------------------------
unsigned int measure_isNew;
// Value from ADC interrupt, answers: "do we have a new measurement?"

#define FILTER_SIZE 7
// Size of Median Filter Memory
#define PIX_SORT(a,b) { if ((a)>(b)) PIX_SWAP((a),(b)); }
// Function used in median filter
#define PIX_SWAP(a,b) { unsigned int temp=(a);(a)=(b);(b)=temp;}
// Function used in median filter
```

```
unsigned int set_output[FILTER_SIZE];
// Median Filter Buffer (stores old output measurements)
unsigned int idc_output = FILTER_SIZE;
// Points at the oldest sample of set_output
//(counter in filter data set)
unsigned int median_output = 0;
// The median from the set_output-array (filter output)
unsigned int set_inductor[FILTER_SIZE];
// Median Filter Buffer (stores old inductor measurements)
unsigned int idc_inductor = FILTER_SIZE;
// Points at the oldest sample of set_inductor
unsigned int median_inductor = 0;
// The median from the set_inductor-array

unsigned int switch_Aup = 0;
// Switch A, upper. Used in A to B mode.
unsigned int switch_Bdown = 0;
// Switch B, lower. Used in A to B mode.
unsigned int switch_Bup = 0;
// Switch B, upper. Used in B to A mode.
unsigned int switch_Adown = 0;
// Switch A, lower. Used in B to A mode.
unsigned int *upperSwitch;
// Pointer for used switch.
unsigned int *lowerSwitch;
// Pointer for used switch.

unsigned int i_ref = 160*0;
// 52 är 1 amp. 110 2amp. 220 4 amp. 340 6amp. 440 är 8 amp.
unsigned int iL_max = 100;
unsigned int bias = 512;
unsigned int outbias = 0;
unsigned int indbias = 0;
unsigned int boost = 1;
unsigned int direction = 1;
//A to B is 1, B to A is 0. Do not change this value from here.
unsigned int upperSwitch_old, lowerSwitch_old, delay1, delay2;
unsigned int delay1_active = 0;
unsigned int delay2_active = 0;
unsigned int timerdelay = 0;
#define DELAY 3


// Timers. 1500 responds to 5kHz
#define TIMER_PERIOD 1500*200
#define CARRIER_ON 5*150
#define CARRIER_OFF 5*150

// Prototype Section
int main (void);
void init_Ports (void);
```

```
void init_ADC (void);
void init_Filter (void);
void init_Timer1 (void);
void init_Timer2 (void);
void setDirection (unsigned int direction);


// ------------------------------------------------------------------
// Init Functions Section
// ------------------------------------------------------------------

void init_Ports() {
        // Set all E-ports to digital outputs
        TRISEbits.TRISE0 = 0;
        TRISEbits.TRISE1 = 0;
        TRISEbits.TRISE2 = 0;
        TRISEbits.TRISE3 = 0;
        TRISEbits.TRISE4 = 0;
        TRISEbits.TRISE5 = 0;
}

void init_ADC() {
        ADCON1bits.FORM = 0b00;             // Output is given as integers
        ADCON1bits.SSRC = 7;                // Internal counter starts sampling.
        ADCON1bits.ASAM = 1;                // Automatic Sampling ON
        ADCON1bits.SIMSAM = 0;              // Simoultaneous Sampling OFF
        ADCON2bits.VCFG = 0b000;            // Voltage Reference (internal: 0b000)
        ADCON2bits.CSCNA = 1;               // ..
        ADCON2bits.CHPS = 0b00;             // ..
        ADCON2bits.SMPI = 4;                // ..
        ADCON3bits.SAMC = 1;                // Defines Sampling Speed
        ADCON3bits.ADCS = 9;                // Defines Sampling Speed
        ADCHS = 0x0000000000000000;         // ..
        ADPCFG = 0;                         // ..
        ADCSSL = 0b0000000000111110;        // Defines ports to sample
        IFS0bits.ADIF = 0;                  // Pre-Clear Interrupt Flag
        IEC0bits.ADIE = 1;                  // Enable ADC-Interrupt
        ADCON1bits.ADON = 1;                // Enable ADC
}

void init_Filter() {
        // Empty the Filter Array upon reset.
        unsigned int i;
        for (i=0; i<FILTER_SIZE; i++) {
                set_output[i] = 0;
                set_inductor[i] = 0;
        }
}

void init_Timer1() {
        _T1IP = 1;                          // Priority Set to Default (4)
        TMR1 = 0;                           // Timer Counter (set to zero)
```

```
        PR1 = 64000;                    // Timer Period
        T1CON = 0x8002;                 // Configuration Bits
        //(the first bit is the On Bit, on: 0x8002 off:0x0002)
        _T1IF = 0;                      // Pre-Clear Interrupt Flag
        _T1IE = 1;                      // Enable Timer-Interrupt
}

void init_Timer2() {
        _T2IP = 1;                      // Priority Set to Default (4)
        TMR2 = 0;                       // Timer Counter (set to zero)
        PR2 = CARRIER_ON;               // Timer Period
        T2CON = 0x8002;                 // Configuration Bits
        //(the first bit is the On Bit, on: 0x8002 off:0x0002)
        _T2IF = 0;                      // Pre-Clear Interrupt Flag
        _T2IE = 1;                      // Enable Timer-Interrupt
}


// ------------------------------------------------------------------
// Interrupt Function Section
// ------------------------------------------------------------------

void __attribute__((interrupt, no_auto_psv)) _ADCInterrupt(void) {
        measure_isNew = 1;
        IFS0bits.ADIF = 0; // Clear Interrupt Flag
}

void __attribute__((interrupt, no_auto_psv)) _T1Interrupt(void) {
        //setDirection(!direction);
        if(!timerdelay--) {
                timerdelay = 300;
                setDirection(!direction);
                /*if (i_ref == 160) {
                        i_ref = 50;
                } else {
                        i_ref = 160;
                }*/
        }
        _T1IF = 0; // Clear Interrupt Flag
}

void __attribute__((interrupt, no_auto_psv)) _T2Interrupt(void) {
        _T2IF = 0; // Clear Interrupt Flag
}


// ------------------------------------------------------------------
// Functions
// ------------------------------------------------------------------

void setDirection (unsigned int directionN) {
        // Set function for the current direction.
```

```
        if (directionN == 1) {
                // Send current from A to B.
                upperSwitch             = &switch_Aup;
                lowerSwitch             = &switch_Bdown;
                switch_Bup              = 0;
                switch_Adown            = 0;
                direction               = 1;
                outbias                 = 0;
                indbias                 = 4;

        } else {
                // Send current form B to A.
                upperSwitch             = &switch_Bup;
                lowerSwitch             = &switch_Adown;
                switch_Aup              = 0;
                switch_Bdown            = 0;
                direction               = 0;
                outbias                 = 0;
                indbias                 = 4;
        }
}


// ------------------------------------------------------------------------
// Main Loop Section
// ------------------------------------------------------------------------

int main (void) {
 init_Ports();
 init_ADC();
 init_Filter();
 //init_Timer1(); // Should be commented out. :)
 //init_Timer2();
 setDirection(1); // Send current from A to B
 setDirection(0); //  No, the other way, please!
 while(1) {
  if(measure_isNew) {
   measure_isNew = 0;
   /*
   This main loop is executed whenever a new measurement
   has been made, and a new decision about the continued
   operation can be taken. It is divided into 4 subsections.
   */

   // 1. Filtering of current measurements
   // ---------------------------------------
   if (direction) {set_output[--idc_output] = ADCBUF0;}
   else {set_output[--idc_output] = ADCBUF2;}
   if (!idc_output) idc_output = FILTER_SIZE;
   PIX_SORT(set_output[0], set_output[5]) ;
   PIX_SORT(set_output[0], set_output[3]) ;
   PIX_SORT(set_output[1], set_output[6]) ;
```

```
PIX_SORT(set_output[2], set_output[4]) ;
PIX_SORT(set_output[0], set_output[1]) ;
PIX_SORT(set_output[3], set_output[5]) ;
PIX_SORT(set_output[2], set_output[6]) ;
PIX_SORT(set_output[2], set_output[3]) ;
PIX_SORT(set_output[3], set_output[6]) ;
PIX_SORT(set_output[4], set_output[5]) ;
PIX_SORT(set_output[1], set_output[4]) ;
PIX_SORT(set_output[1], set_output[3]) ;
PIX_SORT(set_output[3], set_output[4]) ;
median_output = set_output[3];

set_inductor[--idc_inductor] = ADCBUF1;
if (!idc_inductor) idc_inductor = FILTER_SIZE;
PIX_SORT(set_inductor[0], set_inductor[5]) ;
PIX_SORT(set_inductor[0], set_inductor[3]) ;
PIX_SORT(set_inductor[1], set_inductor[6]) ;
PIX_SORT(set_inductor[2], set_inductor[4]) ;
PIX_SORT(set_inductor[0], set_inductor[1]) ;
PIX_SORT(set_inductor[3], set_inductor[5]) ;
PIX_SORT(set_inductor[2], set_inductor[6]) ;
PIX_SORT(set_inductor[2], set_inductor[3]) ;
PIX_SORT(set_inductor[3], set_inductor[6]) ;
PIX_SORT(set_inductor[4], set_inductor[5]) ;
PIX_SORT(set_inductor[1], set_inductor[4]) ;
PIX_SORT(set_inductor[1], set_inductor[3]) ;
PIX_SORT(set_inductor[3], set_inductor[4]) ;
median_inductor = set_inductor[3];

// 2. Decide wether to Buck or to Boost.
// ------------------------------------
if (direction) {
 if (median_output < bias-outbias + i_ref - 5) {
  // If we are here, we haven't reached our goal yet.
  // Go to boost.
  boost = 1;
 } else {
  // If we are here, we have reached our goal.
  // Should we go to buck instead? We should look at the voltages
      // at this steady state.
  if (ADCBUF4 + 5 < ADCBUF3) {boost = 0;} else {boost = 1;}
      // Condition: output + eps < input. The small bias eps is
      // because of voltage losses in system.
 }
} else {
 if (median_output > bias - i_ref + 4) {
  boost = 1;
 } else {
  if (ADCBUF3 + 5 < ADCBUF4) {boost = 0;} else {boost = 1;}
 }
}
//boost = 1; // For debugging only. Comment away.
```

```
// 3. Operate in Boost or Buck.
// ----------------------------
if (boost) { // Boost
 upperSwitch_old = *upperSwitch;
     // Preserve last state of the upper switch
 lowerSwitch_old = *lowerSwitch;
     // Preserve last state of the lower switch
 if (!delay2_active) {
      // No delay function active. Verify that inductor current is
      // going in the right direction.
  if (direction) {
       // The inductor current will be reversed if
       //we change direction, so we need appropriate conditions.
   if (median_inductor > bias + iL_max + 30) {*upperSwitch = 0;}
   if (median_inductor < bias + iL_max) {*upperSwitch = 1;}
  } else {
   if (median_inductor < bias - iL_max - 30) {*upperSwitch = 0;}
   if (median_inductor > bias - iL_max) {*upperSwitch = 1;}
  }
  if (upperSwitch_old != *upperSwitch) delay1_active = 1;
      // Turn on delay, so that lowerSwitch doesn't change state
      // until some time has passed.
 } else {
      // Delay function active, we can't change state of upperswitch
      // until we know that lowerswitch is at a steady state.
 delay2++; // Increment the delay timer.
 if (delay2 > DELAY) {
      // Check if delay timer has reached it's limit.
  delay2 = 0; // Reset delay timer.
  delay2_active = 0;
      // Turn-off delay timer 2, thus allowing upperswitch
      // to change state.
 }
}

if (!delay1_active) {
     // No delay function active. Verify that output current
     // is going in the right direction.
 if (median_output > bias-outbias + i_ref + 6) {*lowerSwitch = 1;}
 if (median_output < bias-outbias + i_ref - 4) {*lowerSwitch = 0;} // SHOULD BE ZERO!
  if (direction) {
   if (median_inductor < bias + iL_max - 10) {*lowerSwitch = 1;}
  } else {
   if (median_inductor > bias - iL_max + 10) {*lowerSwitch = 1;}
  }
  if (lowerSwitch_old != *lowerSwitch) delay2_active = 1;
      // Turn on delay, so that upperSwitch doesn't change state
      // until some time has passed.
 } else {
      // Delay function active, we can't change state of
      // lowerswitch until we know that upperswitch is at a steady state.
```

```
      delay1++; // Increment the delay timer.
      if (delay1 > DELAY) {
           // Check if delay timer has reached it's limit.
       delay1 = 0;
           // Reset delay timer.
       delay1_active = 0;
           // Turn-off delay timer 1, thus allowing lowerswitch to
          // change state.
      }
         }
    }

    if (!boost) {// Buck
     delay1 = 0;
         delay2 = 0;
     delay1_active = 0;
     delay2_active = 0;
     *lowerSwitch = 0;
     //if (median_inductor < bias + i_ref + 5 -80)        {*upperSwitch = 1;}
     //if (median_inductor > bias + i_ref - 5 -80)        {*upperSwitch = 0;}
     if (median_output < bias-outbias + i_ref + 10)       {*upperSwitch = 1;}
     if (median_output > bias-outbias + i_ref)            {*upperSwitch = 0;}
          if (direction) {
          if (median_inductor > bias + i_ref/3 + 10) {*upperSwitch = 0;}
          } else {
           if (median_inductor < bias - i_ref/3 - 5) {*upperSwitch = 0;}
          }
    }

    // 4. Store decisions in output latches.
    // ---------------------------------------
    LATEbits.LATE0 = switch_Aup;
    LATEbits.LATE1 = switch_Bdown;
    LATEbits.LATE2 = switch_Bup;
    LATEbits.LATE3 = switch_Adown;
    if (ADCBUF0 > 512+60) {LATEbits.LATE4 = 1;}
    else {LATEbits.LATE4 = 0;}
    if (median_output > 512+60) {LATEbits.LATE5 = 1;}
    else {LATEbits.LATE5 = 0;}

    //if (ADCBUF2 < 512 - 125 - 4)                       {LATEbits.LATE4 = 0;}
    //if (median_output > bias) {LATEbits.LATE4 = 1;}
    // else {LATEbits.LATE4 = 0;}
   }
  }
  return 0;
}
```