Simulation and Design of a Maximum Power Point Tracking System for Electrodialysis

Daniel Speckhard

Table of Contents

1. Motivation and System Specifications

a. Motivation

It's estimated 1.8 billion people use a source of water that is fecally contaminated.[1] A potential solution to this problem lies in distributed water filtration systems. One such system that offers the potential for continuous water filtration for a community is Electrodialysis.[2] Electrodialysis works by removing charged species in water through ion-selective membranes (see image below). Electrodialysis (ED) has been used to filter brackish water from the ground and sea. This process requires up to 250W to provide approximately four litres an hour of clean water continuously depending on contaminant and saline concentrations.[3]
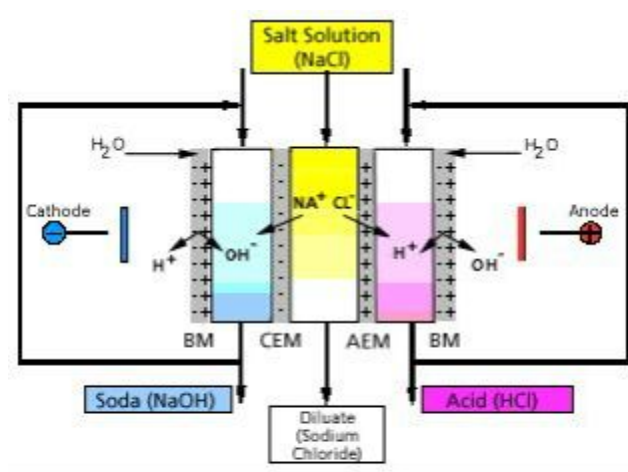

Figure 1: Electrodialysis mechanism courtesy of Fraunhofer IGB. [4]

A novel approach to this problem is to use solar power to provide sufficient energy for ED. A 250W solar panel is currently priced at a modest 350 U.S. Dollars which includes import

[1] "WHO | Water supply, sanitation and hygiene development." 2004. 2 Sep. 2015 <http://www.who.int/water_sanitation_health/hygiene/en/>

[2] Amor, Zakia et al. "Fluoride removal from brackish water by electrodialysis." *Desalination* 133.3 (2001): 215-223.

[3] "The Electrodialysis Cell PCCell ED 200 - PCA GmbH." 2005. 2 Sep. 2015 <http://www.pca-gmbh.com/pccell/ed200.htm>

[4] "Fraunhofer IGB." 2011. 2 Sep. 2015 <http://www.igb.fraunhofer.de/en.html>

and sales tax. [5] This research, performed at the Technical University of Vienna, aims to design and simulate a solar powered electrical system to power Electrodialysis cells and provide backup power in lead-acid batteries to continue to clean water when solar energy is not available.

b. Electrical Block Diagram

The proposed system contains a 250W solar panel which is doing two functions:

1. The ED cells are being powered directly from the solar panel.

2. Excess energy from the solar panels is to be stored in lead-acid batteries.

The proposed system block diagram is as seen below.
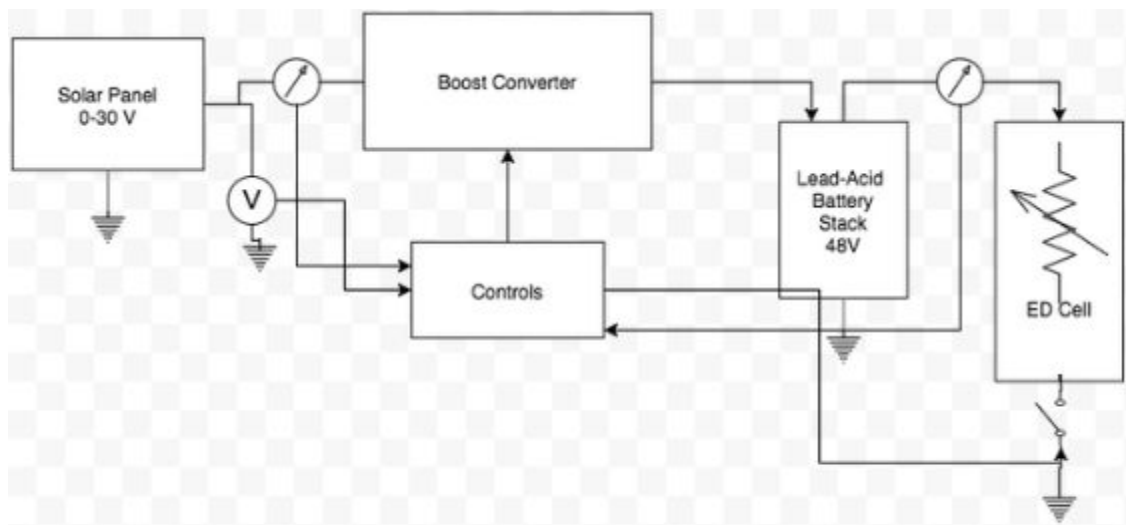

Figure 2: Proposed electrical system. Block diagram drawn with draw.io.

[5] "Renogy 250 Watts 24 Volts Monocrystalline Solar Panel." 2015. 2 Sep. 2015
<http://www.amazon.com/Renogy-Watts-Volts-Monocrystalline-Solar/dp/B00F9HUXWO>

The solar panel provides energy for a boost converter which charges the batteries. The control block determines what voltage to set the solar panels so as to achieve maximum power withdrawal from the panels. The ED cell current is controlled with a low-side switch.

c. System Realization

The boost converter in the system is PWM controlled. With a constant output voltage set by the batteries, the duty cycle of the PWM signal from the controls to the low-side boost FET will determine the input voltage seen across the solar panels. The PWM frequency will be 200 kHz, a frequency where many components are cheap, efficient and lightweight. The input voltage across the panels will determine how much power can be withdrawn from the panels. A sample 250W solar panel from Trinasolar shows how output power is affected by input voltage seen below.
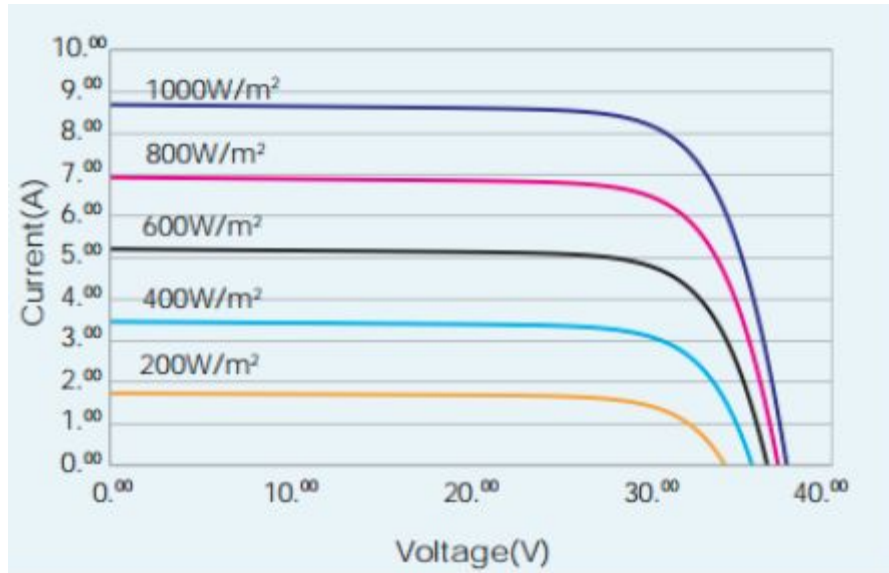
Figure 3: IV curves for PV Module TSM-245C courtesy of Trinasolar. [6]

The lead-acid battery stack is very easily realized by stacking 12V batteries in series. Lead-acid batteries regulate themselves to avoid overvoltage and undervoltage issues. This means that only the output voltage of the boost converter will be monitored for overvoltage and undervoltage issues. [7] Lead-acid batteries are robust and can handle temperature changes very well and for these reasons are chosen for energy storage.[8]

The electrodialysis cell is modeled as a variable resistor. A low-side switch implemented with a simple NMOSFET will suffice. A gate driver will be included. More often than not the ED cell will not draw 250W of power (this depends on pollutant concentrations). The low-side switch can be turned on and off at a frequency of around 200 Hz without significant noise/EMI issues at

[6] "TSM-PC05 TSM-PA05 - Trina Solar." 2013. 2 Sep. 2015
<http://www.trinasolar.com/HtmlData/downloads/us/products/multi/PA05_EN_datasheet_June_2012_US.pdf>
[7] "MDEQ - Lead Acid Batteries." 2013. 2 Sep. 2015
<http://www.deq.state.ms.us/MDEQ.nsf/page/Recycling_LeadAcidBatteries>
[8] "Technical Manual - Power-Sonic Corp." 2010. 2 Sep. 2015
<http://www.power-sonic.com/images/powersonic/technical/1277751263_20100627-TechManual-Lo.pdf>

such a low switching frequency. The low-side switch will ensure that excess energy from the boost converter will charge the battery.

2. Simulation

a. Solar Panel Simulation

The 250W solar panel can be modeled as a single solar cell. A subcircuit with a current source a diode drop and shunt and series resistances will model the panel accurately (diagram below).[9]
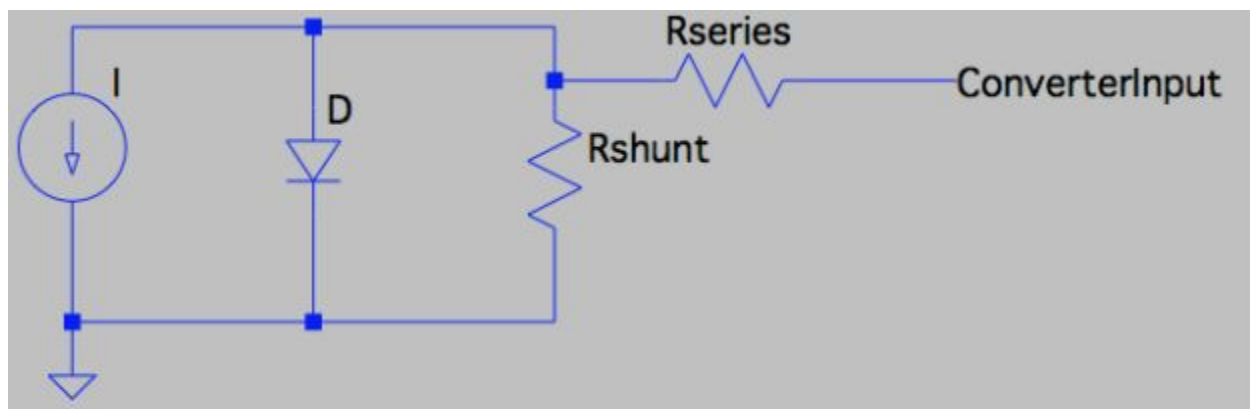
Figure 4: Solar panel model with shunt and series resistance. ConverterInput signifies the output terminal of the panel referenced to a common ground. Schematic drawn in LTSpice.

The open circuit voltage (OCV) is set by choosing the diode hard turn-on voltage (as long as the diode impedance is smaller than the series and output shunt impedance). The shunt resistance and series resistance as well as the current source will determine the output power characteristics. We can vary the output resistance to find a power vs. voltage curve which will tell us the voltage across the panel for optimal output power.

---

[9] Ramaprabha, R, and BL Mathur. "Modelling and simulation of solar PV array under partial shaded conditions." *Sustainable Energy Technologies, 2008. ICSET 2008. IEEE International Conference on* 24 Nov. 2008: 7-11.

SPICE simulation results in a simple output power versus voltage for a 280W panel with an OCV of 30V and MPPT around 24V at 280W. The output power curve has a clear apex which will be exploited by our MPPT controls. Appendix A contains all SPICE netlist files. This subcircuit will be used in boost converter simulations.
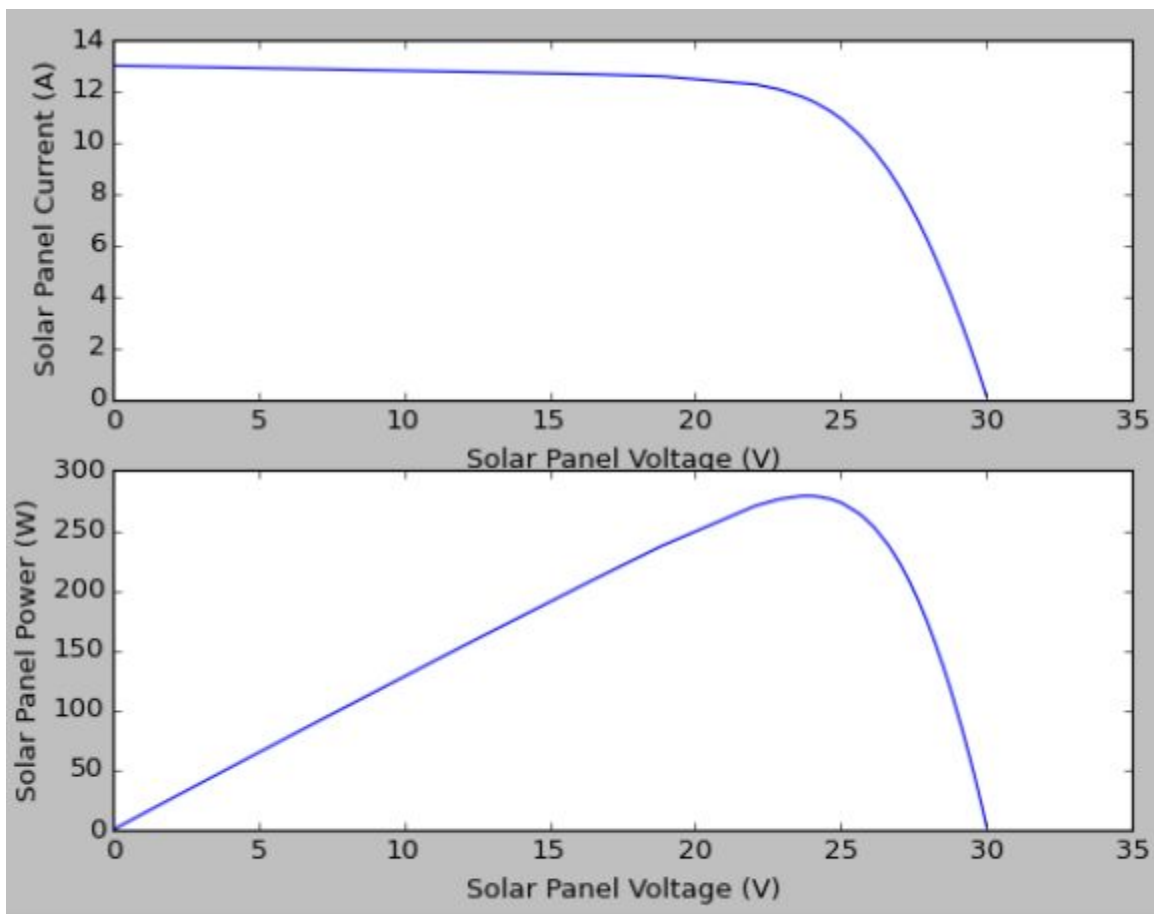

Figure 5: (Top) Solar panel current versus solar panel voltage. (Bottom) Solar panel power versus solar panel voltage. Simulated in LTSpice.

b. Lead-Acid Battery Simulation

The lead-acid battery string can be simulated as a huge capacitor with a small equivalent series resistance. The R-C network will determine charging/discharging time. For our MPPT simulations we will treat the lead-acid battery string as a constant voltage source at 48V. MPPT will be occurring in less than one second. In this time-scale the battery stack voltage will not be altered significantly and our constant voltage assumption is valid. The lead-acid battery stack can be balanced with large 1% resistors in parallel to the batteries. Any over-voltage in a single cell will cause energy to be dumped into the resistors to balance the cell stack. In future iterations of the lead-acid battery stack charge balancing can be implemented as well as individual battery cell monitoring.

c. Boost Converter Simulation

The boost converter is to be as efficient as possible. At maximum power intake the boost converter will output more than 5A. The diode in the boost converter will see all of the output current and thus its power loss is approximated as:

$$P_{diode-loss} = I_{rms}^2 * R_{on} + V_{threshold} * I_{avg}$$

To increase efficiency we would like to use the smallest resistance as possible. The voltage drop across a schottky diode is limiting in that respect. An alternative option is use synchronous rectification. [10] The R_{ds-on} value of a power MOSFET is smaller than the equivalent diode on-resistance and MOSFETs do not show any threshold voltage $V_{threshold}$ while on , which grants lower losses. Since the diode naturally commutates when the low-side switch turns off the high-side switch will be turned on with near zero-voltage across it. This means

---

[10] "Synchronous rectification boosts efficiency by reducing ..." 2013. 2 Sep. 2015
<http://www.ti.com/lit/slyt515>

hard-switching losses can be ignored. For our simulations, we will use a synchronous rectifying power MOSFET. The MOSFET model in this simulations will contain a ROSS/COSS. This COSS value and ROSS value will be approximate since FET datasheets do not specify ROSS values. The IRFB3207ZPBF 70V Mosfet is chosen for it's small $R_{ds\text{-}on}$. The $R_{ds\text{-}on}$ is approximately of 4 mOhms at 5A drain current and 10V gate to source.[11] The body diode will be lossy and modeled as an element with a threshold voltage and an on-resistance, with an estimated V_{on} of 0.6 V and a R_{on} of 0.5 Ohms. If the body diodes show to be a very lossy factor in our simulations they can be removed from our loss calculations by adding parallel schottky diodes.

d. MPPT Control Strategy

The boost converter will take in a PWM signal into its low side FET where the duty cycle will determine the input voltage across the panel. The lossless boost converter transfer function is the reciprocal of the lossless buck converter transfer function.

$$\frac{V_{out}}{V_{in}} = \frac{1}{1-D}$$

With V_{out} a constant voltage source (this is how we modeled the battery stack), we have the below equation for the approximate panel voltage.

$$V_{panel} = \frac{V_{battery-stack}}{1-D}$$

[11] "IRFB_S_SL3207ZPbF Product Datasheet - International ..." 2006. 4 Sep. 2015
<http://www.irf.com/product-info/datasheets/data/irfs3207zpbf.pdf>

The voltage across the panel and the current from the panel will be sensed continuously. A LPF is used to smoothen the voltage and current waveforms over ten switching cycles to avoid 200 kHz ripples. The duty cycle of the converter will be changed at a frequency of 20 kHz.

The control block representing an 8-bit AVR microcontroller will compare new input powers to old output powers and follow the hill-climbing method represented in the below chart. [12]

Duty Cycle Condition	Input Power Condition	Duty Cycle Change
$D_{new} > D_{old}$	$P_{new} > P_{old}$	Increase D
$D_{new} < D_{old}$	$P_{old} < P_{new}$	Decrease D
$D_{new} > D_{old}$	$P_{new} < P_{old}$	Decrease D
$D_{new} > D_{nold}$	$P_{old} < P_{new}$	Increase D

If the duty cycle condition and the input power condition are met we change the duty cycle as seen above. There are only four possible states. Therefore, the micro-controller only has to evaluate two statements to determine if it should increase D. If those two statements are false, the microcontroller will decrease D. In SPICE this control is implemented with controllable voltage sources and simple sample and hold circuits sampling at 20 kHz. The amount D is increased each cycle is a fixed amount in this control scheme. More complicated control schemes will change D by an amount each cycle determined the relative change in power withdrawn from the solar panels. Our simulations will optimize this change in duty cycle through simulation. If the newly sampled power and the recently stored power are equal to each other, we will increase duty cycle. This is done to avoid the control from keeping the converter in an open circuit voltage state.

---

[12] Xiao, Weidong, and William G Dunford. "A modified adaptive hill climbing MPPT method for photovoltaic power systems." *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual* 20 Jun. 2004: 1957-1963.

3. MPPT Simulation Results

        The components of the boost converter can be sized quite easily. The inductor is chosen to ensure CCM operation. An input capacitor is placed across the solar panel to ensure the panel voltage doesn't change abruptly and current is drawn in spurts from the panel. The size of the input capacitor will affect our control loop. We can change the frequency of duty cycle change and the solar panel input capacitor value. The MPPT frequency is kept constant. The MPPT frequency is held at 20 kHz for these simulations. The amount the duty cycle is changed each cycle begins at a modest 0.6%. This duty cycle incremental change will affect the ripple voltage seen on the boost converter input. The input capacitor to the boost is chosen for startup time, stability and input voltage ripple. We sweep through several different values of input capacitance to see this effect.
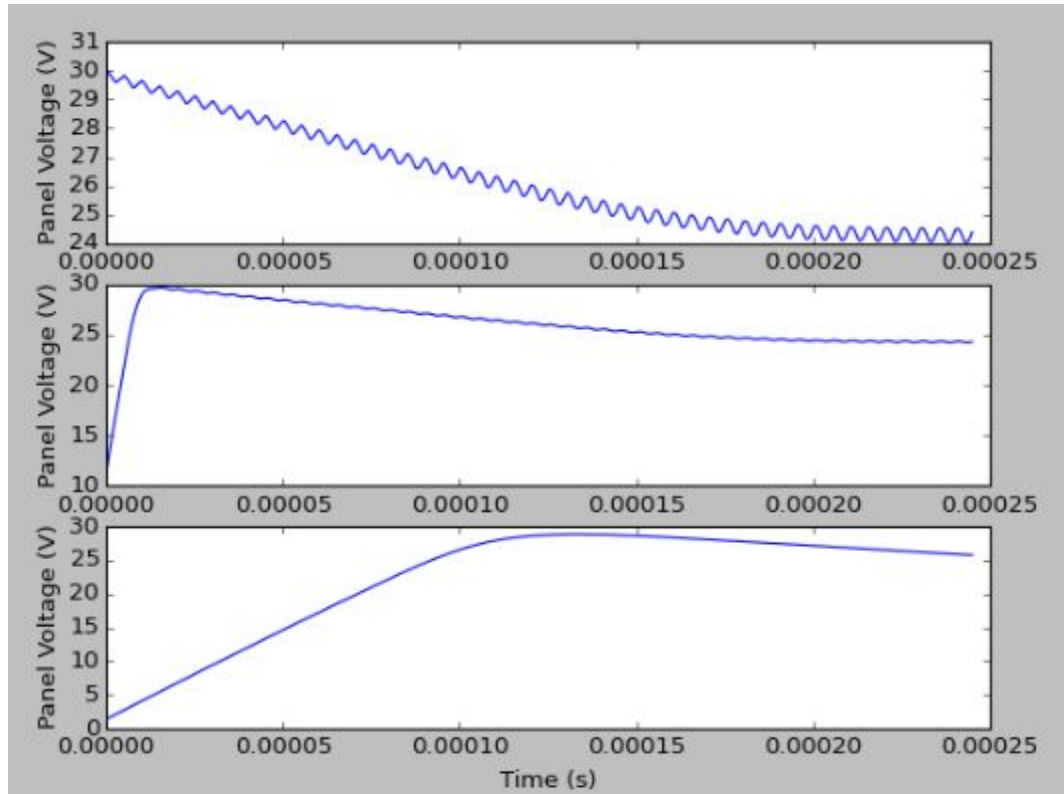
Figure 6: Panel voltage versus time for first 50 boost switching cycles. (Top) 0.47uF input capacitance. (Middle) 4.7uF input capacitance. (Bottom) 47uF input capacitance. Simulated in LTSpice.

For this duty cycle increment of 0.6% and initial duty cycle of 0.5, we see the input capacitance greatly changes the MPPT tracking ripple on the input of the boost. The MPPT is 24V in this simulation. The ripple on the 0.47uF capacitor is 0.5V peak to peak while the 4.7uF ripple is 0.13V peak to peak. The 47uF input capacitance greatly slows down the startup time and in 50 switching cycles still approaches the MPPT. A slow response time is larger issue when the input voltage is over the open circuit voltage (OCV) of the solar panel and the MPPT control must bring the panel voltage down. The 4.7uF capacitor is kept for future simulations.

Let us now vary the duty cycle increment so as to optimize our current setup. The converter waveforms help one understand how the duty cycle reference voltage changes when

the converter starts in an OCV state, a duty cycle of 0.35. Below, in Figure 7, the duty cycle

reference voltage very slowly rises 2% over a tenth of a second as it tries to reach a MPPT

reference duty cycle of 2.5V. As soon as the panel escapes OCV the duty cycle reference

voltage climbs at a much faster rate than seen in Figure 7.
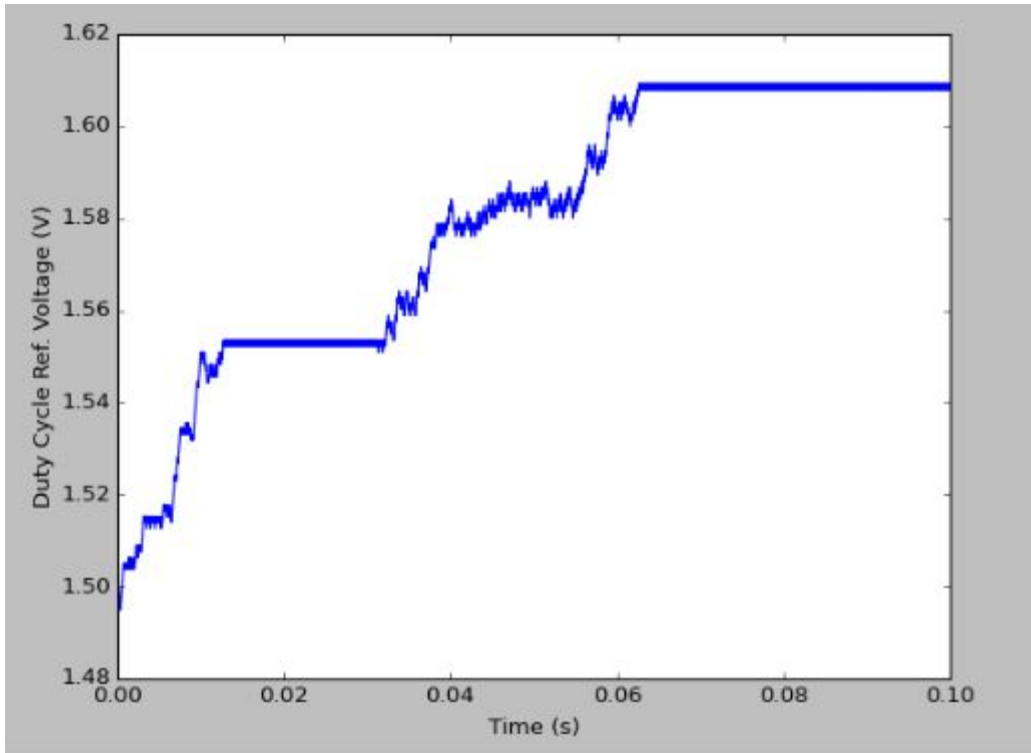
Figure 7: Duty cycle reference voltage versus time for a duty cycle
increment of 0.2%. The duty cycle is equal to the reference voltage
divided by 5V. Simulated in LTSpice.

Duty Cycle Percent Increment (%)	Time to Exit OCV and FInd MPPT (ms)	Average Ripple Voltage at MPPT (V)	Average MPPT Offset (V)
0.2	176	0.13	0.2
0.6	16.6	0.13	0.35
1.4	9.7	0.13	0.55

The above table shows that the duty cycle percent increment choice is a balance between MPPT offset voltage (the difference between the MPPT and where the controller believes the MPPT to lie) and the time to exit an OCV state from a duty cycle of 0.3 and reach the MPPT. We will continue to use a 0.6% duty cycle percent increment to not trade off one category for the other. The average ripple voltage at MPPT is determined by the input capacitor. which is constant through the three tests. The average ripple voltage will determine the size of the input filter (i.e. inductor) required to avoid large current pulses coming out of the solar panel. The offset from the MPPT is also determined from our panel I-V curve's steepness near the MPPT.

We can look more closely at the input voltage and input current in the below plot where we begin with a duty cycle of 0.5 and examine 20 boost switching cycles in Figure 8 (see below). We note the boost converter current is clearly in CCM from its inductor current. The inductor chosen for simulation has a DCR of 15mOhms which is aggressive for a 47.5uH inductor. The losses in the loss table below show the importance of this DCR. The boost converter appears 97% efficient. This number will drop with a microcontroller an LDO to power the microcontroller and gate drivers (high-side driver especially) to switch the MOSFETs.
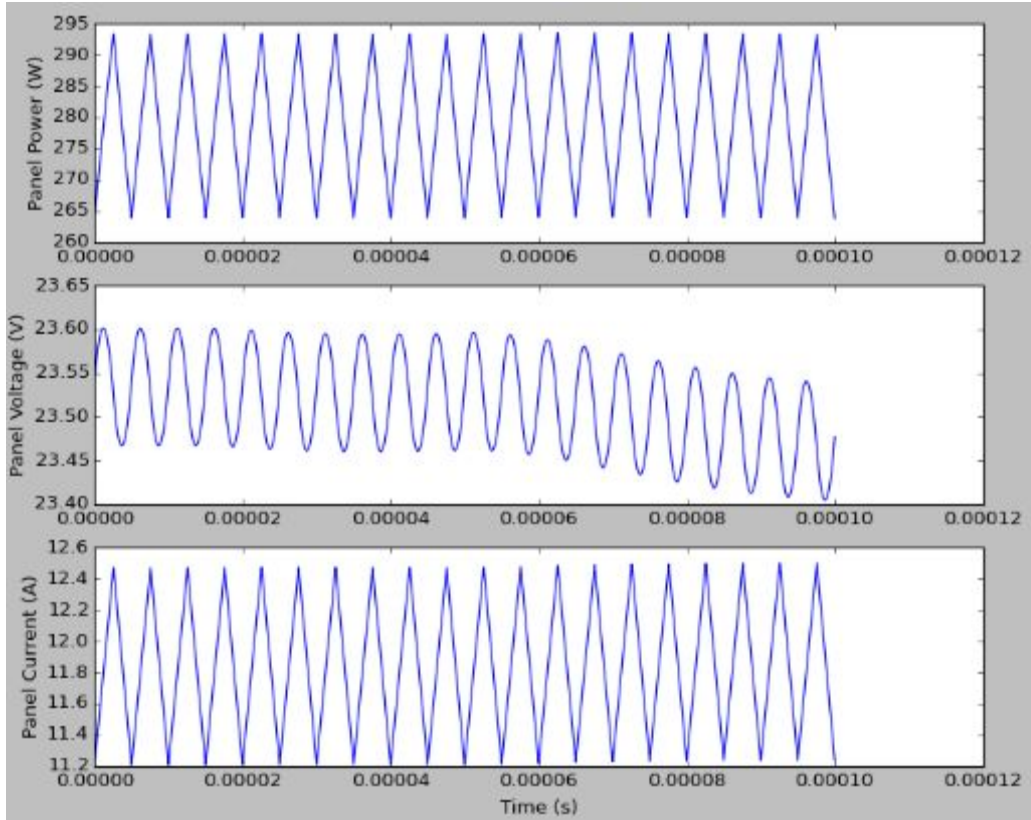
Figure 8: MPPT boost converter waveforms for 20 switching cycles. D percent increment is 0.6%. Input capacitance set at 4.7uF.

Low-Side FET Power (W)	1.5
High-Side FET Power (W)	0.05
Inductor Power (W)	1.7
Power Loss due to MPPT Offset	1.3

4. System Prototype

A simple prototype on a universal PCB was attempted in the final weeks of research at TU-Wien. An Arduino Micro microcontroller was used for the control block and PWM output. The Arduino has in built analog to digital converters which simplify prototyping. It also has PWM output which with a bit of hacking in the Arduino IDE can be run up to 200 kHz. The SI8234BB will work as a synchronous rectification IC gate driver with the dead time set by a simple resistor. A TPS7A107A LDO is used to power the microcontroller off of the batteries. This scheme has yet to be optimized to reduce losses. Perhaps the LDO can be run off the low-side 12V lead-acid battery in the battery stack to reduce losses. The universal PCB prototype was used to gain insight into the gate driver IC and Arduino Micro microcontroller. The gate driver IC appears fully suitable for the next iteration PCB. The Arduino Microcontroller appears to have some timing issues that have not yet been fully resolved. Likely, a 8-bit Atmel Tiny 26L will work better than the Arduino Microcontroller.

A 250W solar panel is very difficult to operate in a laboratory setting requiring large UV light installations to power the panel. A simple solution used in the power electronics laboratory at TU-Wien involves an NPN transistor and a several diodes (see figure 10 below). The diode string emulates the V-I characteristics of the solar panel, whereas the transistor acts as a current multiplier. LTSpice simulations confirm the output power versus voltage plot can be made to very closely model a solar panel.
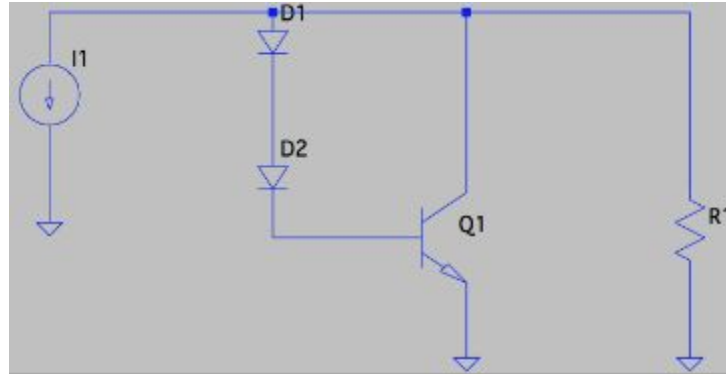
Figure 10: Schematic to emulate a solar panel. Resistor R1 models the MPPT converter load. Drawn in LTSpice.

5. Conclusion and Next Steps

The hill climbing method and boost converter to battery scheme is validated through simulation. A 250W PCB will now be laid out with insight from the universal PCB. The PCB will be laid out in late September 2015 and a results should be present in November 2015. A new microcontroller and a new LDO scheme will be implemented in the next generations of this PCB.

6. Appendix

a. Solar Module Subcircuit SPICE Netlist

```
**Solar Module Test
* TU WIEN - MAY 2015
* Prof. Ertl's Lab


.PARAM:
+FS = 200K
+T = 1/{FS}
+D = 0.64375
+T2 = 50*T
;+ISUN={ISUN}
+IS = 1E-11
+RSER = 0.5
+RSHUNT = 1MEG
+VMAX = 21
+IsMAX = 0.33
+N = {VMAX*38.6/LOG10({ISMAX}/{IS})}
+XTI ={IS}
+EG = {1.11*N}
+R1={R1}



X1 R00 0 PV
R1 R00 0 {R1}



.STEP PARAM R1 LIST   0.001 0.01 0.05 0.075 0.1 0.3 0.5 0.8 1 1.3 1.5 1.8 1.9 2 2.02 2.04
2.06 2.08 2.1 2.2 2.3 2.5 2.7 3 3.3 3.6 4 4.5 5 6 7 10 15 20 25 35 50 300
* PV.cir: simple yet effective PV Array model
.subckt PV A K params: Rp=1 Rs=3m Isc=13 n=50 m=1

* Rp sets voltage slope of curve
* Rs sets current slope of curve
* Isc= short circuit current
* n= This gives the number of series cells
```

* m= This gives the number of parallel strings
I1 A K {-Isc*m}
D1 A K cell n={n} m={m}
Rp A K {Rp*n/m}
.model cell d Rs={Rs}
+ Is=315n n=1.262 Cjo=50u
+ Tnom=25 Trs1=10m Trs2=0m1
.ends PV

;.STEP PARAM ISUN LIST 4 10 20 50
.model DIDEAL D(IS={IS} RS={RS} N={N} XTI={XTI} EG={EG})

.MEASURE TRAN POWEROUT AVG {V(R00)*I(R1)}
.MEASURE TRAN VOLTAGEIN AVG {V(R00)}
.TRAN {1/(200*FS)} {2000/FS} {1000/FS} {1/(200*FS)} UIC

b. Boost MPPT Circuit SPICE Netlist

**BOOST MPPT
*JUNE 2015
*TU-WIEN

.PARAM:

+ COSS = 1600P; COSS OF N-MOSFET
+ ROSS = 200m
+ COSSD1=500P
+ ROSSD1=200M
+ L = 47.5u;47.5u;47.5u;47.5u;47.5u;105U
+ ESRL = 15M;
+ VIN = 17.1;
+ VOUT = 48;
+ ROUT = 5.2;38.4; OHMIC RESISTANCE OF DESAL CELL
+ COUT = 47.7U;
+ RDSON = 0.1m;9M; MOSFET RDSON
+ RON = 0.0001;0.35; DIODE ON RESISTANCE
+ VT = 7; MOSFET TURN ON VOLTAGE

```
+ FS = 200K; SWITCHING FREQUENCY
+ T = 1/{FS}; SWITCHING PERIOD
+ D = 0.5;{D};0.8;0.64375; MPPT UNREGULATED DUTY CYCLE
;+R1={R1}
+CIN = 4.7u
+T2=10*T
+TDELAY1 = 100U
+TDELAY2= 100.001U


;.step param D list 0.1 0.3 0.7 0.9
;.step param T2 list 1m 0.01m 0.05m

*** Main Boost ****
X1 Rb0 0 PV
CB1 Rb0 0 {CIN};Previously 150u 19.6 voltage at d=0.64, MPPT at ~24.87V 280w
L1 Rb0 Rb1 {L}
RESRL RB1 RB2 {ESRL}

*** MOSFET ***
SB1 Rb2 0 RbG 0 OPTIMOS

*** COSS ****
;COSS RB2 RBCOSS {COSS}

;ROSS RBCOSS 0 {ROSS} ; ESTIMATED VALUE
** GATE VOLTAGE ***
VG RbG1 0 PULSE(7.0 0 {{T}*{D}} 80P 80P {{T}*(1-{D}-160P)} {T})

SB2 RBG1 RBG RBTIMER1 0 SWIDEAL

VBTIMER1 RBTIMER1 0 PWL(0 7 {TDELAY1} 7 {TDELAY2} 0)

SB3 RX7 RBG RBTIMER2 0 SWIDEAL

VBTIMER2 RBTIMER2 0 PWL(0 0 {TDELAY1} 0 {TDELAY2} 7)



*********************

*** SYNC REC MOSFET ***
;SBSYNC Rb2 RB3 Rb2 RB3 OPTIMOSYNCREC
```

```
*** COSS ****
;COSSSYNC RB2 RBCOSSYNC {COSS}

;ROSSSYNC RBCOSSYNC RB3 {ROSS} ; ESTIMATED VALUE

D1 Rb2 Rb3 DIDEAL; IDEAL DIODE
;COSSD1 RB2 RB2DOSS {COSSD1}
;ROSSD1 RBSDOSS RB3 {ROSSD1}


COUT Rb3 0 {COUT};

;ROUT Rb3 0 {ROUT};

vout rb3 0 {vout};


**** Current sensing and LPF *****

HCS RCS1 0 L1 1;
;LC1 RCS1 RCS2 100n;
CC1 RCS2 0 800n;10.2u;
rc1 rcs2 rcs1 10

******** CONTROLS *********

BVPOWERIN R00 0 V=v(rcs2)*V(rb0);V=V(RCS2)*V(RB0)


S1 R00 R01 RTIMER1 0 SWIDEAL

*****RTIMER1 SIGNAL *****
; THIS SIGNAL TURNS ON S1 FOR 0.05*T2 TO CHARGE UP C1
; THIS SIGNAL IS 7V FROM 0 - 0.05T2 THEN GOES TO 0V
; This is the Pnew

VTIMER1 RTIMER1 0 PULSE(0 7 {0.45*T2} 0 0 {0.05*T2} {T2})

*************************


;S3 R01 0 RTIMER3 0 SWIDEAL
```

*****RTIMER3 SIGNAL *****
; THIS SIGNAL TURNS ON S3 FOR 0.05*T2 TO DISCHARGEC C1
; THIS SIGNAL IS 7V FROM 0.35T2 - 0.4T2 THEN GOES TO 0V

;VTIMER3 RTIMER3 0 PULSE(0 7 {0.95*T2} 0 0 {0.05*T2} {T2})

****

C1 R01 0 100n ; PNEW VOLTAGE

BV1 R02 0 V = V(R01)

S2 R02 R03 RTIMER2 0 SWIDEAL

*****RTIMER2 SIGNAL *****
; THIS SIGNAL TURNS ON S2 FOR 0.05*T2 TO CHARGE C2 (SWAP POLD FOR PNEW)
; THIS SIGNAL IS 7V FROM 0.85 - 0.9T2 THEN GOES TO 0V

VTIMER2 RTIMER2 0 PULSE(0 7 {0.85*T2} 0 0 {0.05*T2} {T2})

C2 R03 0 100N ; POLD VOLTAGE

***************** Make V(rx1) = 30mV or -30mV depending on higher/lower Power

BVX RX0 0 V=if((v(r01)>=v(r03)&v(rx3)>=V(rx5)) | (v(r01)<=v(r03)&v(rx3)<v(rx5)) , 0.07, -0.07);
if pnew >pold, bvx = 0.03, else -0.03

S4 RX0 RX1 RTIMER4 0 SWIDEAL

*****RTIMER4 SIGNAL *****
; THIS SIGNAL TURNS ON S4 FOR 0.05*T2 TO CHARGE C2 (SWAP POLD FOR PNEW)
; THIS SIGNAL IS 7V FROM 0.6 - 0.65T2 THEN GOES TO 0V

VTIMER4 RTIMER4 0 PULSE(0 7 {0.6*T2} 0 0 {0.05*T2} {T2})

C3 RX1 0 100N; this is Vdg

```
*********** VB ***********

BVB RX2 0 V={V(rx1)+V(rx5)}

S5 RX2 RXX2 RTIMER5 0 SWIDEAL

VTIMER5 RTIMER5 0 PULSE(0 7 {0.99995*T2} 0 0 {0.00005*T2} {T2})

*** Extra switch to prevent Vref new updating before mppt is tracking
S7 RXX2 RX3 RTIMER7 0 SWIDEAL

VTIMER7 RTIMER7 0 PWL(0 0 {TDELAY1} 0 {TDELAY2} 7)


C4 RX3 0 100N IC={D*5}; THIS IS VB = V(RX3) ; this is vref new
**********

*********** VA ***********
;THIS IS THE SIGNAL WE USE FOR PWM AGAINST SAWTOOTH

BVA RX4 0 V=V(rx3) IC={D*5}

S6 RX4 RX5 RTIMER6 0 SWIDEAL

;VTIMER6 RTIMER6 0 PULSE(0 7 {0.99975*T2} 0 0 {0.05/200*T2} {T2})
VTIMER6 RTIMER6 0 PULSE(0 7 {0.7*T2} 0 0 {0.05*T2} {T2})

C5 RX5 0 10N IC={D*5}; THIS IS VB = V(RX3) ; this is vref old with new algorithm

**********************************

***** We need to store



***** Sawtooth function for PWM *****

VSAWTOOTH RX6 0 PULSE({0.5*T} 5 0 {T} 0 0 {T})
```

************* PWM SIGNAL *******

BVG RX7 0 V=IF(V(RX3)>V(RX6), 7.00,0)

****** SOLAR PANEL ******

* PV.cir: simple yet effective PV Array model
.subckt PV A K params: Rp=1 Rs=3m Isc=13 n=50 m=1

* Rp sets voltage slope of curve
* Rs sets current slope of curve
* Isc= short circuit current
* n= number of series cells
* m= number of parallel strings
I1 A K {-Isc*m}
D1 A K cell n={n} m={m}
Rp A K {Rp*n/m}
.model cell d Rs={Rs}
+ Is=315n n=1.262 Cjo=50u
+ Tnom=25 Trs1=10m Trs2=1m
.ends PV

;.STEP PARAM ISUN LIST 4 10 20 50
.MODEL DIDEAL D(RON=.25 ROFF=10MEG VT=0.55); ron-0.25 von-550mV coss - 500pF ross
- 4mOhm (estimated)
.MODEL SWIDEAL SW(RON=.2 ROFF=10MEG VT=4.4 VH=0.01)
.MODEL OPTIMOS SW(RON=.0015 ROFF=10MEG VT=6.9 VH=0.0)
.MODEL OPTIMOSYNCREC SW(RON=.0015 ROFF=10MEG VT=0.00 VH=0.0)
; NOTE I NEED A REAL DIODE IT APPEARS WITH SATURATION CURRENT AND ALL THAT
JAZZ

.TRAN {1/(200*FS)} {40000/FS} {1/FS} {1/(200*FS)} UIC

c. Arduino Microcontroller Script

```
// MPPT control in Arduino Micro:
// TU-Wien - June 2015 - Ertl Lab - Speckhard

/* The code will anlog input sample voltage across sense resistor at 2kHz. (Maybe I will
implement higher sampling rate and
digitally LPF the samples). Store this value as new current. Sample voltage across cap. New
voltage. Multiply two values for new power.
Compare to old Power.*/

char  CurrentPin = A0; // Analog input pin to read current as a voltage
char VoltagePin = A4; // analog input pin to read voltage, spaced out a bit
char PWMpin = A3; // PWM output PIN
int Duty = 250; // initial duty cycle value is 0.5
int OldPower = 0;// GetPower(); // This is the first old power
int OldVoltage = 0;
int DutyChange = 0;

void setup() {
Serial.begin(9600);
pinMode(CurrentPin, INPUT);
pinMode(VoltagePin, INPUT);
pinMode(PWMpin, OUTPUT);
}

int GetPower(int Voltage) {
  // function gets power and returns int with units V^2
  //int Voltage = analogRead(VoltagePin);
  delay(100); // This is introduced to avoid jitter being picked up on A0, need to optimize
  int Current = analogRead(CurrentPin);
  int Power = Voltage*Current;
  //int OutputArray[] = {Power, Voltage};
  return Power;
}


void loop() {
  // put your main code here, to run repeatedly:
int NewVoltage = analogRead(VoltagePin);
int NewPower = GetPower(NewVoltage);
```

```
// delay(100);

if (((NewPower > OldPower) & (NewVoltage<OldVoltage)) | ((NewPower < OldPower) &
(NewVoltage>OldVoltage)))
{
  int DutyChange = 7;
}
  else {
    int DutyChange = -7;
  }

Duty = Duty + DutyChange;

float PWMscale = 255.0/500.0 ;
float DutyScaled = Duty * PWMscale;

analogWrite(PWMpin, DutyScaled); // PWM output on PWMpin of duty cycle PWMvalue/255

OldPower = NewPower;

}
```