DC-DC CONVERTER WITH FUZZY CONTROLLER FOR SOLAR CELL APPLICATIONS ON MOBILE ROBOTS

J. CRUZ-LAMBERT, P. BENAVIDEZ, J. ORTIZ, N. GALLARDO, B. A. EROL, J. RICHEY, S. MORRIS AND M. JAMSHIDI

This paper is dedicated to Professor L. A. Zadeh on the occasion of his 95th birthday and the 50th year of the birth of fuzzy logic

Abstract. Emerging technologies increase the needs on self efficient mobile robotic applications that bring a new concern of sustainable and continuous power supply for the robotic platforms. This paper covers the various techniques and technologies used to design a solar powered robot, exploring the currently available products, software and limitations to this application. The main aim is to integrate a fuzzy logic based charging system which allows the batteries to be charged from solar panels, wall outlet, and a deploy-able solar charging station. The goal of this paper is to summarize the tested methods and results to expedite future researchers in the correct direction. This paper will cover only up to the design of the DC-DC converter and simulation, as further work is still pending implementation on actual hardware. Simulations results are provided to evaluate the feasibility of the paper for future implementations.

1. Introduction

Power management is a performance characteristic that is always taken into consideration when designing and building a mobile robot platform. When choosing a power source many factors are taken into consideration: weight of the battery, size, capacity, reliability, discharge/charge rate, etc. These different attributes come with varying price points which may limit the designers ability to obtain the perfect battery for the system. In this paper, power production and supply is looked at from a different approach; using renewable solar energy to power a mobile robot. By using a renewable approach, lesser number of batteries will be deployed, as a result of it, the weight and cost of the robot can be reduced. The viability of this approach depends on the continuing advances in solar cell technology.

Various cell configurations and solar panel placements were considered for efficiency and power output. Monocrystalline, polycrystalline and thin film solar panels were investigated. Thin film panels were ruled out on the basis of lower output power output ratings. Several approaches were found in the literature for the placement of solar panels on a mobile robot. Lever and Ray designed a long

Invited Paper: September 2016

Key words and phrases: Solar, Renewable, LiPo, Lithium Polymer, MPPT, Robotics, Fuzzy Controller, Energy.

duration, zero emissions robot for summertime research in Antarctica and Greenland that has a five panel box-like design [10]. In the most recent iteration of "Cool Robot," the combined solar panel output is 455W at peak power [13]. A hopping/jumping lunar robot named LunaRoo has a folding three thin-film solar panel design on a cube shaped platform that provides an estimated 2W of power [9]. Fei and Lv mounted a solar panel on a tilt and roll mount to a small mobile robot [6]. Kaplan and Uhing mounted an 18W solar panel to the top of a small Dagu 5 robot chassis with zero inclination angles [8]. From the literature it is obvious that the solar panels that are available on the market drive the overall shape and dimensions of a robot. This is likely due to the efficiency of the panels and relative difficulty in manufacturing a custom solar panel array.

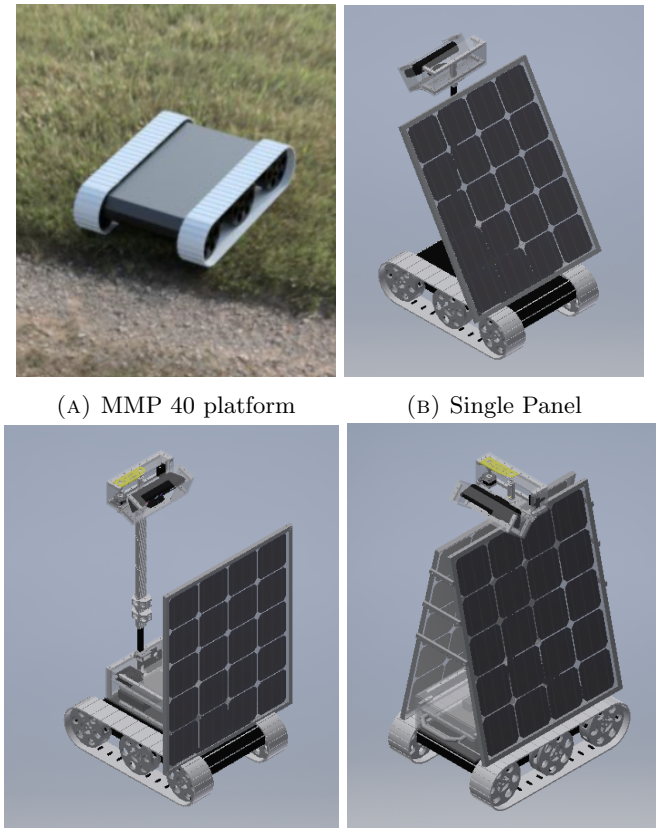
On the other hand, battery requirement varies for particular design and is determined based on the robot power load and payload. Lithium Polymer (LiPo) batteries were selected as the power source for the proposed robot since they have a competitive power density to weight ratio. Moreover, in order to extend the LiPo battery's life, it was decided to utilize a dual battery bank configuration. The idea is to charge one battery bank while the other is being utilized. This allows precise control over the battery charging profile without load variations interfering with any measurements. Therefore, it is undoubtedly crucial that the proposed system should have an adaptable power supply controller, which intelligently observe the system needs and decide the charging, utilizing and recharging schedules for each individual LiPo batteries, continuously. This scheduler should control the system in a way that the self efficiency will remain in entire task time, while observing the changes on the demands and the current power level of the in-use battery.

Therefore, this paper presents a hybrid power system implementing the use of solar panels and batteries on a robot. The main aim is to integrate a charging system which allows the batteries to be charged from solar panels, a wall outlet, or a deploy-able solar charging station. A fuzzy maximum power point tracking (MPPT) controller was used. The primary focus for the controller was to regulate the power input for a LiPo by means of Constant Current (CC), Constant Voltage (CV) control. Different structures of DC-DC converters were considered and based on our literature research, it was concluded that a Buck-Boost converter is the most appropriate option when working with solar cells at the power level of around $50W$. The proposed system is divided in three sections: design of solar panels, design of a DC-DC converter and design of a fuzzy controller to implement maximum power point tracking and battery management. Section 2 will explain the design constraints of the system. Section 3 covers the simulation design. Section 4 introduces the controller design. Section 5 provides the conclusions and future work.

2. **Design Constraints**

In this section, the design constraints for the system are addressed. Some of the high level requirements addressed in this section are the following: 1) the robot will operate outdoors for long durations, 2) batteries will be recharged only by solar energy and 3) the robot will be able to operate in both a low-power reconnaissance mode (static, no driving) and a high-power search mode (dynamic, driving).

2.1. **Solar Panel Selection: Custom vs Commercial-off-the-Shelf.** When designing a solar robot, the tradeoff between the efficiency of the solar cells and the available space on the target robot must be considered. Most of the available solar panels on the market have an efficiency of 15%. This implies that the solar panel surface area will be large if the power demands are high. With a $50W$ power requirement from the system, the size of the solar panel is along the larger side of the spectrum. Limited by the physical space of the robot, a custom approach was taken to optimize the structure of the cells. Arrangements in pyramid, inverted pyramid, accordion, and flexible membrane structures were explored. Figure 1 shows some configurations of the solar panels on a "MMP-40" robot from "The Machine Lab".



(A) MMP 40 platform (B) Single Panel

(C) Alternate single panel (D) Two Panels

FIGURE 1. Options for Solar Panel Placement on MMP 40 Robot

Taking account of these concerns, an initial batch of 100 solar cells was used for constructing a custom panel. Solar cells were selected based on the highest W/m rating, normalized by the size of the solar cells. Tests were performed on the cells to

obtain actual performance characteristics. After testing the different arrangements, it was determined that a flat solar panel would be the easier to manufacture, and marginally more efficient than the other configurations. Once the solar cell structure was determined, construction of the solar panel began. Construction of a solar panel is a complicated process, involving laying the cells carefully in an array, soldering the traces together via small wires, and affixing the connected panels to the lexan polycarbonate plastic panel via epoxy and/or waterproof resin. One panel took approximately 28 hours to build. Building a solar cell on the sole merit of saving money is not a good reason to construct a custom panel after factoring in the time, cost, and materials. It is recommended that one should consider all factors that go in to constructing a solar panel and if at all possible, buy one that is off the shelf and readily available. Ultimately, due to the concerns about the durability of the resulting custom-built panel, an off the shelf panel was selected to be used on the robot. The single panel configuration in Figure 1 (a) was selected as the most logical for the "MMP-40" platform in this work.

2.2. **Battery Charging Characteristics Based on Robot Operation Modes.** Next the power demand of the proposed platform was examined. Having a fixed power generation, it was determined that the best approach was to investigate a scenario that illustrates the highest power demand and battery usage. To simplify the design of the electrical system, the solar panel is restricted to exclusively charge Lithium-Polymer batteries. This, in turn, reduced the complexity of the circuit to two components: the buck-boost converter and a single battery. To further reduce the load disturbance on the panel, an additional bank of batteries were added with the goal of always having one bank charging. Therefore only having the load of a buck-boost converter inefficiencies, as well as the battery charge cycle across the panels. The second bank is used to power the robot's electrical components such as CPU, sensors, and other miscellaneous equipment. The robots payload in terms of equipment consists of a standard payload used on an ACE Lab robot. Table 1 shows a breakdown on the components and the maximum expected load that is used in determining the power budget.

In Table 1, the motor and motor drivers demand the most power in the system. The batteries were then selected based on the power demand of the heaviest loads. First, the working voltage of the motors based on the manufacturer specifications were considered as shown in equation 1.

$$12.00V \pm 2.0 < V_{motor} < 24.00V \pm 2.0 \tag{1}$$

Using standard Lithium-Polymer batteries, each battery cell has a nominal operational voltage of $3.7V$. In this system, industry standards of $3.5V$ and $4.2V$ were used to determine when the battery was fully depleted or fully charged, respectively. This data was taken directly from the battery distributor "HobbyKing" [7]. With these parameters, an estimate can be made for the operating voltage range of the batteries as shown in equation 2.

$$21V < V_{batt} < 25.2V \tag{2}$$

The range for V_{batt} referenced in equation 2 was determined to be suitable for operation of the robot.

Item ID	Item Description	QTY	Expected Load Total
1	Motor (Left/Right)	2	$200.0W$
2	Xbox Kinect RGB-D Camera	1	$12.96W$
3	ODROID XU4 – ARM-based Octacore Microcomputer	1	$20.0W$
4	Head Rotation Stepper	1	$3.96W$
5	High Torque Servo	1	$6.6W$
6	USB Wi-Fi Adapter	1	$2.5W$
7	Microcontroller (ATMEL Mega class)	1	$1.25W$

TABLE 1. ACE Lab Standard Robot Payload

2.3. **Battery Selection Based on Operational Time.** Lastly, the watt/hours must be calculated to specify the operational time of the system. The power modes of the robot would operate in were narrowed down to two main modes. *Full Operational* (FO) mode where the previous table shows the worst case loading of the system. The second mode was called *Sensory Reconnaissance* (SR) power mode where only mission critical sensors are supplied with power. The mission critical sensors from Table 1 were determined to be: RGB-D vision (item 2), the main system controller (item 3), and wireless connectivity (item 6). These items are always operational during idle stage and during reconnaissance. Therefore, the solar panel needs to charge our batteries equivalent to the power consumption of items 2, 3, and 6 to break even ideally, plus losses. To appropriately plan the FO vs. SR run-time of the system, the condition in equation 3 must be maintained to keep the batteries energized. Additionally, care must be taken to remember the cases where sunlight is not available, i.e. nighttime, inclement weather, clouds, shade, etc.

$$E_{total} > P_{SR}t_{SR} + P_{FO}t_{FO} + P_{loss}t_{loss} \tag{3}$$

This power budget compared to the batteries power capacity was calculated to meet 1 hour of operation time goal. At worst case, the draw of both motors at full power would mean $200W/Hr$. Since the main constraint on the maximum power output of the solar panel was the size of the cells vs efficiency, it was calculated that a $50W$ solar panel could be installed on the robot. It was also found that $50W$ solar panels are readily available on the consumer market. This effectively gave a target load of $35.46W$ to $47.27W$, the max constant load derived from all but the motor drivers running. Therefore a battery of about $50WH$ would be needed. It was calculated that for the robot two $11.1V$ Lithium Polymer batteries rated at $30C$ with a capacity of $5000mAH$ would be sufficient to provide the required power. In order to drive the motors at their maximum speed, the system requires a pair of these batteries wired in series making a 5AH@22.2V system.

2.4. **Summary of System Constraints.** In summary, the aim of the paper is to optimize the amount of time a robot can spend outdoors. Having identified the largest and most constant load on our system, a solar panel was designed with $50W$ at $100W/m^2$ irradiance. Although this solar panel gave 29.08% down to 5.46% buffer from the reconnaissance load to almost full load requirements, it is restricted mostly by the state of the art technology, as well as the surface area the solar panel takes up. The main limitation are, listed in order of importance, size, cost, and efficiency. Once the solar panel was sized and limited, it was determined that the battery bank should be able to provided enough power for one hour of operation in reconnaissance mode. While in worse case scenario loading, the robot would only be operational for approximately 15 minutes. At this point, the robot would switch to the second battery bank and deplete that source, increasing our operation time to a little under 30 minutes in an ideal case.

3. Simulation Design

Having defined and constrained this robotics platform, simulations were carried out including the electronics payload and sources with the aid of Matlab. At first, the entire system was simulated in one model. Simulation of the entire circuit in Matlab/Simulink was not providing clear indications as to which components were under-performing against our performance criteria. As a result, problematic points of the circuit had a cascading effect on the simulation. Therefore the simulation was split into two parts in order to simplify the simulations performed and gain insight into subsystem performance. Solar panel modeling is covered in Section 3.1. The DC/DC converter modeling is performed in Section 3.2.

3.1. **Solar Panel Modeling.** The system was decomposed into different functional pieces in order to better identify where the error lies, first starting with the solar panel. The PowerSim library for Matlab was used in order to have all the advance PSPICE simulation components already tested by a trusted source. To create the solar panel model, the performance of the panel was measured on a sunny day, and the resulting power characteristics were entered into the PowerSim solar cell model. The parameters in Table 2 were used for our simulation, based on the measured values of our cells. Equation 4 was used to solve for the output current of the solar cell.

$$I = I_{ph} - I_S \exp(\frac{V + IR_S}{NV_t}) - (V + IR_S)R_p \tag{4}$$

where $I_{ph} = I_r(\frac{I_{ph0}}{I_{r0}})$, I_{ph0} is the solar generated current, I_{r0} is measured irradiance for I_{ph0}, I_S is the diode saturation current, V_t is the thermal voltage, N is the quality factor. In order to verify the model matched our measured values, the following test circuit was designed in Matlab/Simulink as shown in Figure **??**. After running the simulation for $10s$, the following results were obtained during simulation in Figure 4. These results showed the panel should be able to output a peak of $37.75W$. Just enough power to fulfill the requirements stated in Section 1, but on the low end of the expected $50W$. Upon reviewing the solar cell manufacturer's

data sheet, the expected short circuit output current was $6A$. Re-running the simulation with the short circuit current set to $6.0A$ resulted in a $55.5W$ panel. After the panel was constructed, the experimental short circuit current was measured at $5.8A$. This resulted in an estimated specification of a $53.5W$ solar panel.

After reviewing the simulation results, functional representation of the physical panel had been achieved in the PowerSim simulation software. Although the simulation accurately represents the solar panel constructed, the model parameters were fine tuned based on new measurement data.

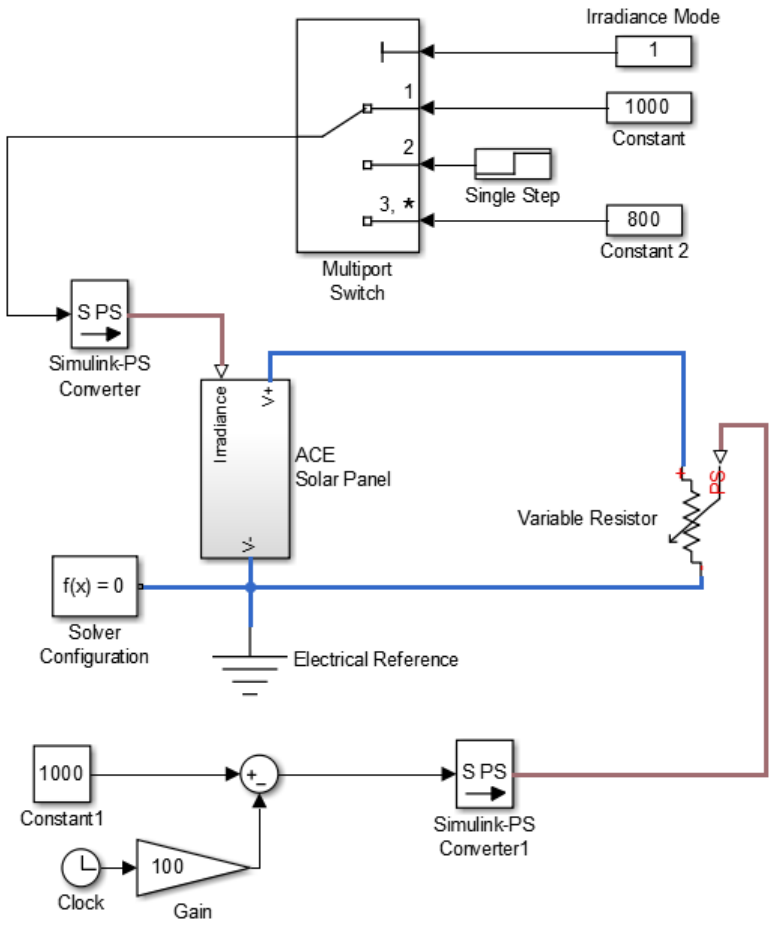


FIGURE 2. Matlab/Simulink model: Modeling Linear Resistance to Observe Solar Panel Response

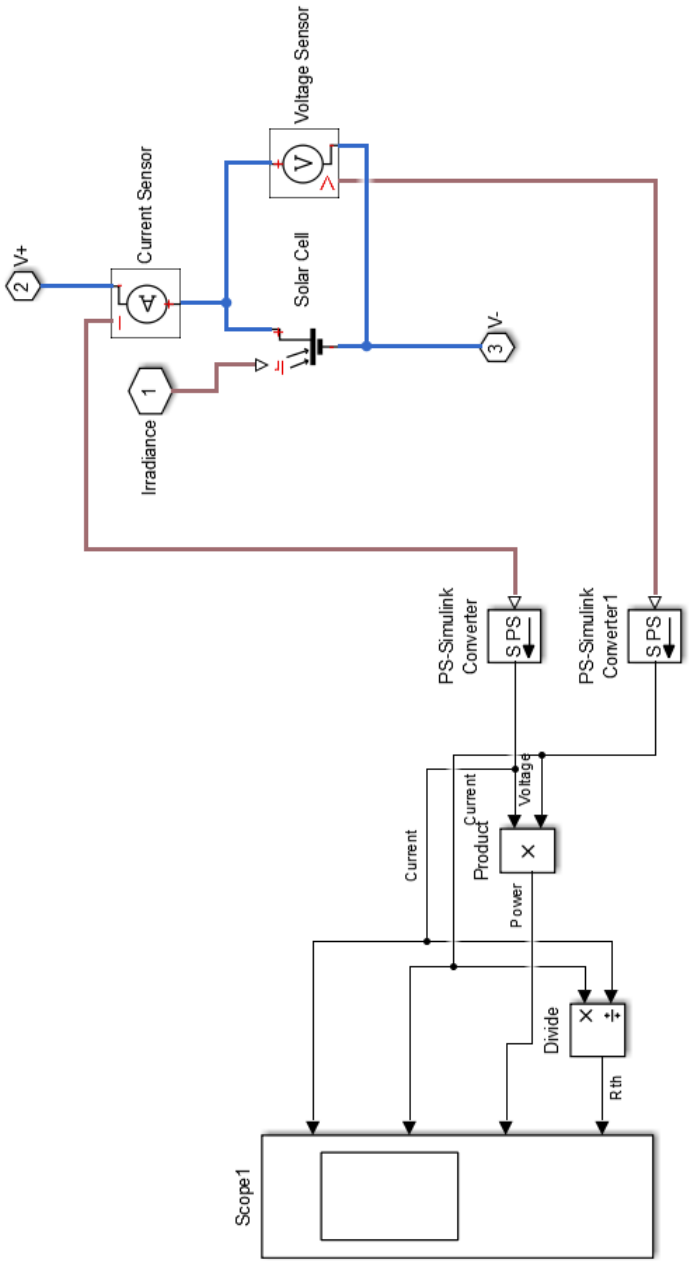


FIGURE 3. Matlab/Simulink Model: Solar Panel Sub-system

Item ID	Parameter	Notation	Value
1	Short Circuit Current	I_{SC}	$4.08A$
2	Open circuit voltage	V_{oc}	$0.6V$
3	Irradiance used for measurements	I_{r0}	$1000.0W/m^2$
4	Quality Factor	N	1.5
5	Series Resistance	R_s	0Ω

TABLE 2. Solar-Cell Characteristics

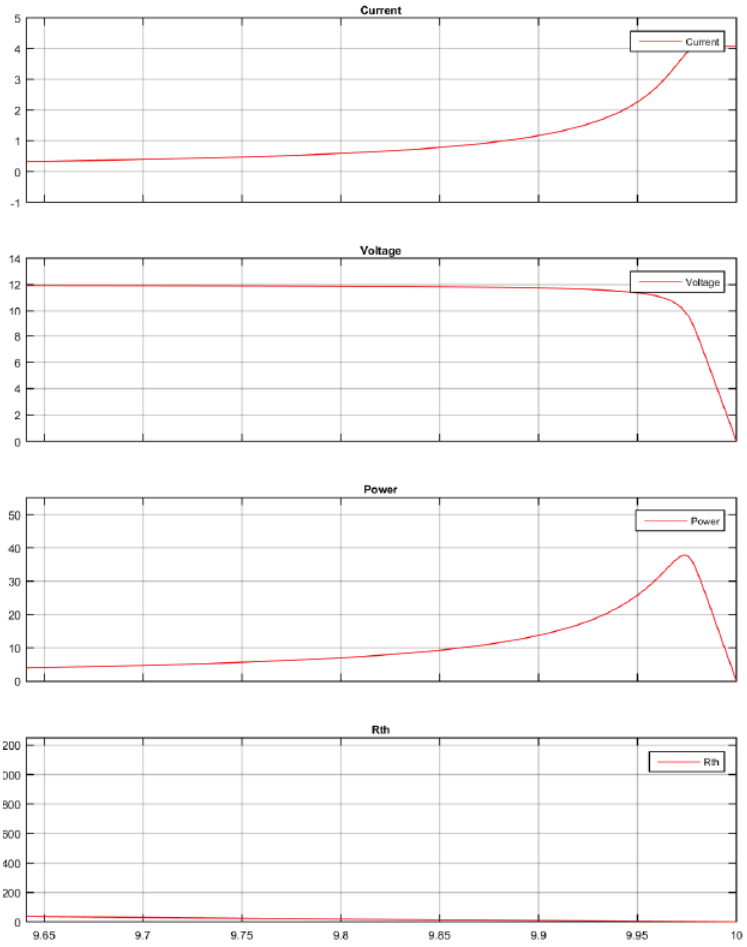


FIGURE 4. Simulation Results

3.2. **DC Converter Modeling.** Initially, a buck converter was selected for the DC converter of the system. Once the buck converter was modeled, the results showed that the buck converter worked, but lacked a principal component. The solar panel is expected to output $10V$ during peak power demand [4]. A single battery under peak demand would need to supply $10.5V$ at $5A$ [5]. Since it wouldnt be possible to meet the battery demand and solar cell characteristics, the converter would need to step-up the voltage, at reduced current output to make charging possible. Thus it was necessary to use a boost converter, but with a dropout voltage of at least 1 volt, it wouldnt be feasible either. Finally, it was determined that a buck-boost converter would be the best choice for this application. Since the boost converter uses an inductor independently to power the circuit, it is possible the voltage needed for the batteries. Since the input voltage is within 2 volts from the output voltage, most of the losses would be due to non-idealities in the inductor, but could still reach high efficiencies.

3.2.1. *Simulation in Matlab/Simulink.* Figure 5 shows the simulation model in Matlab/Simulink to create a buck-boost converter.

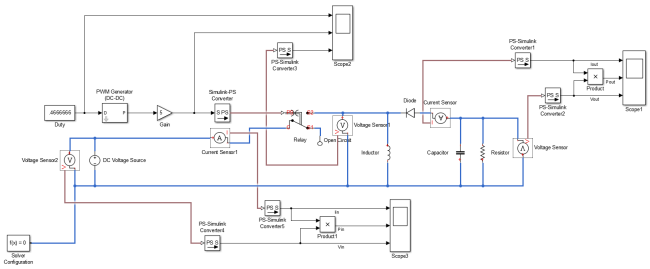


FIGURE 5. Matlab/Simulink Buck-Boost Circuit [11]

Although Matlab comes with built-in DC-DC converters, it is lacking in controlability of external systems. Most off-the-shelf converters and the Matlab models have built in controllers that regulate the voltage and current internally via proprietary hardware. For this paper, one major goal is to build a fuzzy controller that directly drives the PWM duty cycle of the buck-boost converter. Using the Matlab/Simulink circuit shown in Figure 5, The buck-boost converter in [11] was implemented in Matlab.

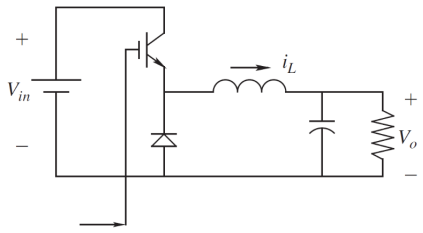


FIGURE 6. Buck Boost Circuit

The values in Table 3 were used for the Buck-Boost circuit parameters.

Item ID	Description	Value	Units
L1	Fixed Inductor	1	mH
C1	Fixed Capacitor	1000	uF
u1	PWM Generator	65.2	kHz
D	Duty Cycle	0.9	N/A

TABLE 3. Buck-Boost Circuit Parameters

The inductor value L1 was determined by using equation 5 from [11]:

$$L = \frac{Vin}{\Delta_{i_L}}(DTs) \tag{5}$$

where $Vin = 12V$, $\Delta_{i_L} = 25mA$ to $5mA$, D was selected arbitrarily for the simulation(later determined by the fuzzy controller) and $Ts = 1/f_s = 1/65.2k$.

In the literature, the inductor value was specifically chosen for the task. Practical considerations or guidance for the capacitance value is omitted from the Ned Mohan text [11]. To determine a practical value, a commercial off-the-shelf boost converter was examined that had similar performance characteristics. Using this as a starting point, a $2000\mu F$ is used. The frequency value was selected upon inspection of examples from [11].

Once the simulation was configured, after running the simulation it was found that the simulation clock would stop after $2.5\mu s$. Although the simulation was repeated using various duty cycles along with other stepping methods, every simulation took more than 5 hours to run, and stopped after a few micro-seconds. This was unacceptable for controller design several iterations of the simulation needed to be executed. Matlab/Simulink was found to be unable to deliver the performance needed. After attempting to optimize the circuit in Matlab, it was decided to use a different simulation platform.

3.2.2. *Simulation in Proteus 8.* Proteus 8 professional was used as alternative to simulate the discussed buck-boost converter. Figure 7 shows the circuit built to simulate the buck-boost converter. The simulation for the circuit in Figure 7 ran significantly faster compared to the one in Matlab/Simulink. However, a quick simulation was only one of the issues, as the output from the simulation did not show desirable results, shown in the Figure 8.

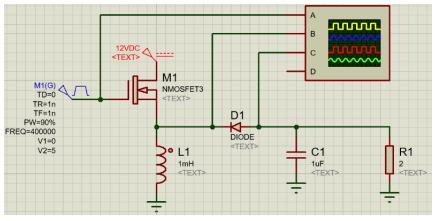


FIGURE 7. Proteus Model of Buck-boost Converter

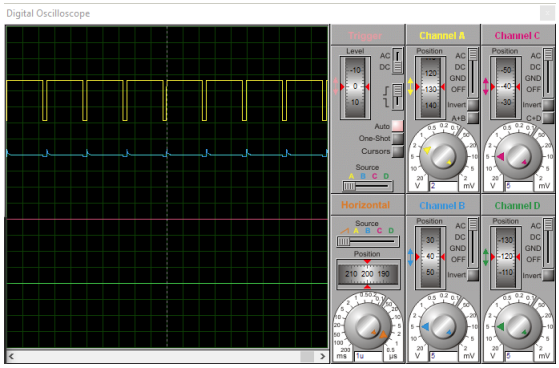


FIGURE 8. Proteus Model Output – TOP/YEL: PWM Input, BELOW/BLU: Voltage Output of Inductor. Middle/RED: Output Voltage

From the oscilloscope output, a near zero voltage is observed in blue (second line from the top) and with significant voltage spikes. Since ripple voltage is undesirable on the batteries, the problem was mitigated by altering the output capacitance value using $1\mu F$, $10\mu F$ and $100\mu F$ capacitor. For each simulation, the results did not change significantly. From the equation earlier, it is expected a $1.08mH$ inductor to give a ripple current of about $25mA$. Having a voltage of nearly zero became the biggest hurdle. Further literature review is necessary in order to properly size the capacitance.

3.2.3. *Simulation using Android App EveryCircuit.* In a last attempt to get a better understanding of the system, an android application named EveryCircuit was used to test the system response. With the aid of EveryCircuit, knowlege was gained in understanding that the capacitor was being charged when the inductor was discharging, the capacitance was not large enough to remove the ripple on the output. Every simulation attempt made to change the capacitance value resulted in either the programs crashing, or undesirable results. Therefore, a more extensive literature review was performed to understand the requirement for matching capacitance to the inductance.

3.2.4. *Summary of Simulation Results.* After reviewing multiple circuits online, it seems the capacitor value was determined based on the inductor value. A consumer grade lithium polymer charger was found in the lab and it was dismantled. It was found to have a $2200\mu F$ capacitor. After adjusting this parameter in the Matlab simulation, the ripple at the output was under $0.02mV$. However, comparing this to results to EveryCircuit application, it shows most of the power is dissipated to the capacitor instead of the load. To mitigate the issue, the capacitor value was lowered to $1000\mu F$. This yielded a ripple voltage of $0.2mV$.

4. **Controller Design**

The design of the controller began first with determining what the inputs and output variables are. From observation, the only control input is the duty cycle of

the buck-boost converter. From the load the voltage and current are obtained and used to derive power. Therefore, those are the inputs. For a discharged battery, a constant current must be applied at 1C. When the battery reaches a voltage of 4.2V per cell, the system must switch to constant voltage charging. After the current has reached 0.1(1C) the battery is considered fully charged. Knowing this information, the fuzzy membership functions and rule set were defined. The Fuzzy system is divided into two controllers. One determines which battery needs to be charged, based on the charge state of each battery, and usage. The other fuzzy controller, directly controls the PWM signal of the buck converter. Each controller is named accordingly to their function, thus the following are used: battery_selector.fis and battery_charger.fis. Figure 9 shows the circuit layout for the system being controlled via the two controllers.

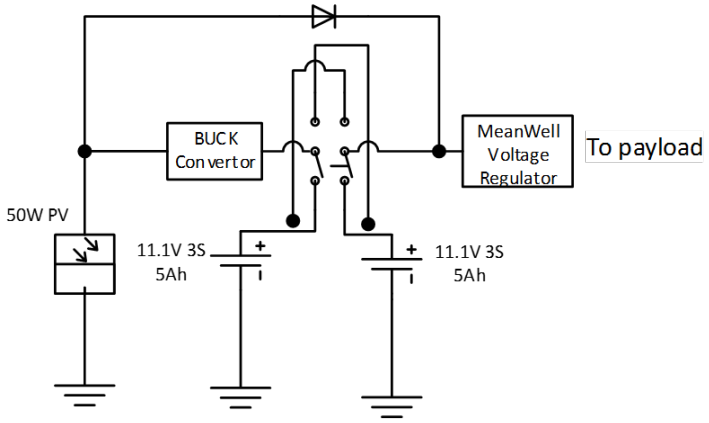


FIGURE 9. Physical Circuit Layout to be Controlled by Fuzzy Controllers

4.1. **Fuzzy Battery Selector Controller.** The battery selector fuzzy controller consists of taking the following state during a battery charge cycle:

(1) Empty The battery voltage is at or below 3.5 and considered empty.
(2) Maximum Current The battery may accept up to 1C to recharge its initial 80-90%.
(3) Fixed Voltage The battery voltage is now at 4.15 per cell, and requires a constant voltage to reach 100% capacity.
(4) Full Battery the battery is considered full when the voltage has reach 4.20V per cell.

Using these four states as an input, the battery selector fuzzy controller determines which batteries will benefit the most from charging, assuming a maximum irradiance is available. The output is a logic true/false from toggling a DPDT relay. Figures 10 provides a screenshot of the fuzzy inference system in Matlab. Figures 11 and 12 show the input and output membership functions of the FIS. Figure 13 shows the rules for the FIS.

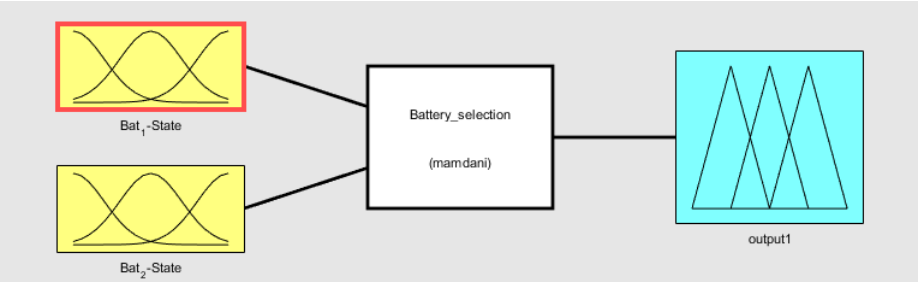


FIGURE 10. Fuzzy Inference System – SystemBatterySelector.fis

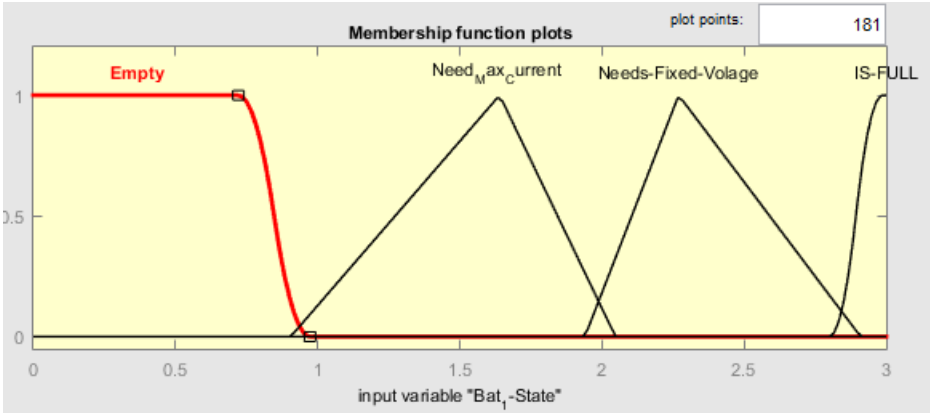


FIGURE 11. Input States for the Fuzzy Inference System

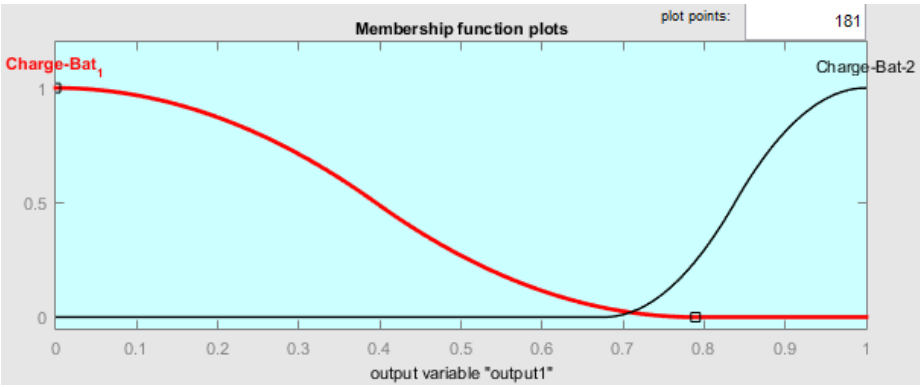


FIGURE 12. Output States for the Fuzzy Inference System

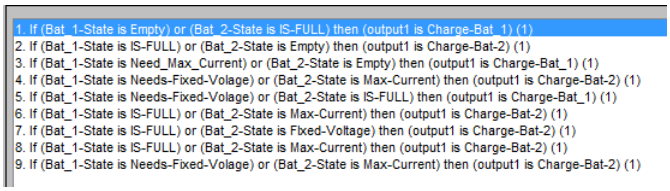


FIGURE 13. Rules for the Fuzzy Inference System

4.2. **Fuzzy Battery Charging Controller.** Next, the battery charging controller was designed based on the following constraints:

- Power output of the solar panel measured at the point in the circuit after the buck converter
- Battery charging current
- Battery charging voltage

Since only one variable of the buck-boost converter can be controlled at a time using conventional techniques, it became apparent that this would be the most complicated portion of the controller. Therefore, battery charging was separated into two controllers, the Fuzzy CC, and Fuzzy CV. Figure 14 shows the block in Matlab/Simulink that represents the fuzzy controller.



FIGURE 14. System: Fuzzy Battery Charging

Figures 15 and 16 show the full detail of the fuzzy controller inside the block in Figure 14.

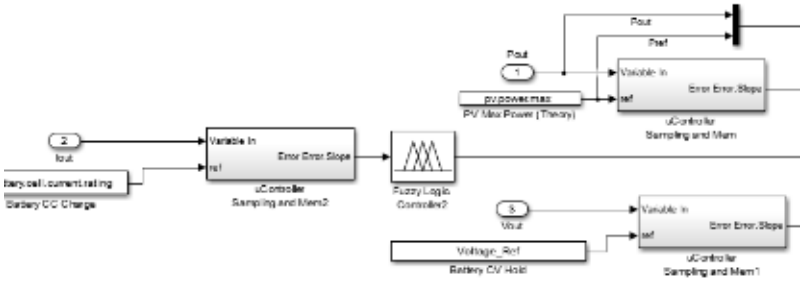


FIGURE 15. System: Fuzzy Battery Charging

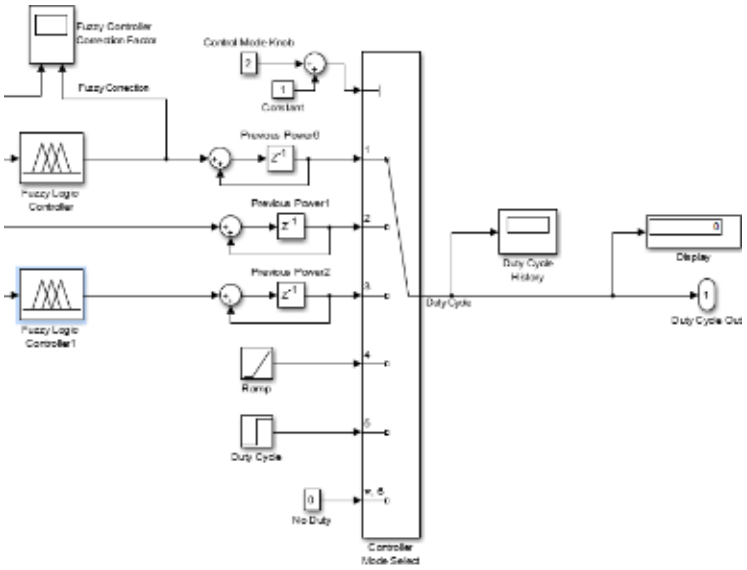


FIGURE 16. System: Fuzzy Battery Charging

Figure 17 shows the switch indicator used to show which controller is active based on the battery charge state. When the battery is empty, $Vbatt = 3.5V * 3 = 10.5V$ (for a 3 cell battery) then the mode is in "Maximum Current" mode. If $Vbatt > 4.1V * 3 = 12.3V$ (for a 3 cell battery) (12.3) then the mode is "Constant Voltage". The "Maximum Power" mode is a simulation parameter designed for future use to test whether or not maximum power could be extracted from the circuit.

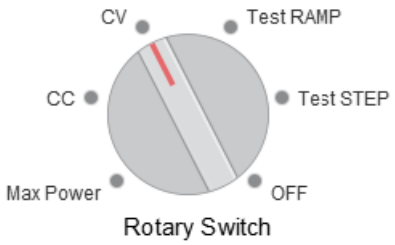


FIGURE 17. System: Fuzzy Battery Charging – Mode Switching

Figures 18-20 show some of the initial controller designs memberships functions, rules and outputs. For testing purposes, a power tracking controller was designed in case maximum power point was of interest in the future applications.

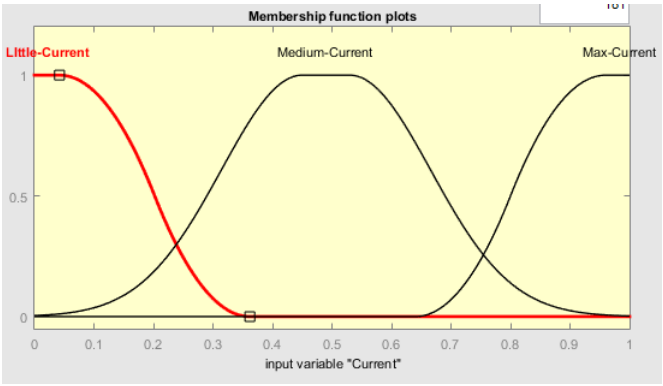


FIGURE 18. System: Fuzzy Battery Charging – Current input

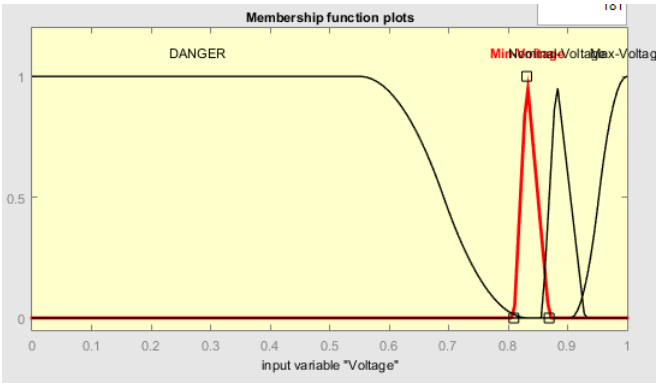


FIGURE 19. System: Fuzzy Battery Charging – Voltage States input

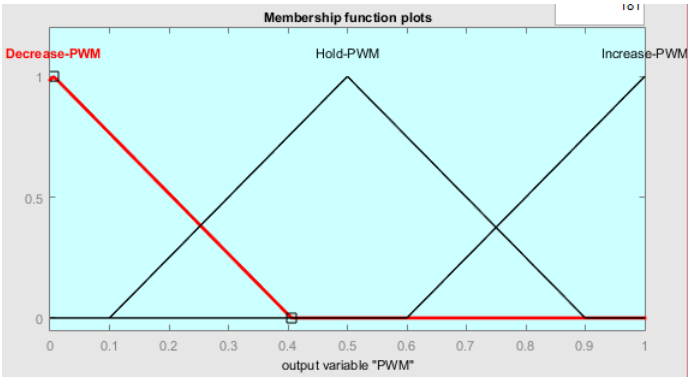


FIGURE 20. System: Fuzzy Battery Charging – PWM Output to Buck Converter Input

4.3. **Hardware Review.** One consideration was to build the circuit and test the controller on an real circuit in tandem with working on the simulations. Although this case was not ideal, the documentation and papers regarding buck-boost controllers was readily available and abundant which was encouraging in taking this approach. At first, the AVR451: BC100 was used. This is a prototype circuit that has three buck converters, each built to specific power levels. The prototype circuit worked but came with a few limitations. Most importantly, technical support for the circuit is practically non-existent. Outside the included technical library, limited external/update documentation was available [1, 2].

4.3.1. *AVR451 Programming using ATMEL-based IDE.* Unfortunately, the code for the AVR451 was written for an ATMEL AVR Studio software version from 2007. It was a challenge to import the code into any newer version of the IDE from ATMEL. However, thanks to the work of Rosenberg [12], most of the code had been ported to somewhat compatible version of AVR Studio. Other users of the AVR451 chip have noted problems with the age of the reference design and lack of support [3]. As updating the code progressed to accept new functions, it became more apparent that the original author, AVR, spent considerable amount of time optimizing the code. Using modern libraries and code conflicted with the existing, out-dated highly optimizer-specific code. The decision was made to abandon this platform and create a custom platform using the more widely supported, Arduino platform.

4.3.2. *AVR451 Programming using Arduino-based IDE.* This approach was much more fruitful than the platform mentioned above; Within a matter of days, the required output PWM signal was determined to be $1.38KHz$ with an adjustable duty cycle via serial communications. While this is not a fuzzy controller, the core foundation for the micro-controller to accept higher level control signals was created. With this, the design of the fuzzy controller began, even though the tuning would have to wait until a full simulation had been realized.

5. Conclusions and Future Work

In this paper, practical exploratory research into development of a solar powered robot has been accomplished. The solar cell and panels were simulated according to the actual measured values. Although the panel still needs to be fully characterized at various irradiance levels, initial results are very promising. Simulation of a buck-boost controller has proven to be more challenging than anything else. The decision to implement the hardware before the simulation is not recommended, and therefore the hardware testing has been suspended until more literature review is evaluated. Since the system isn't fully developed, the fuzzy controller has been created with an understanding of the system goals. Actual implementation of the controller will be done via a linux computer, and micro-controller for digital signal processing. As the individual sub-systems continued to be defined through simulations, it became increasingly difficult to add more complexity to the system without having a direct impact on the simulation performance. Therefore, each sub-system will be tested independently of each other. For initial simulation, all systems were treated as

linear time invariant, each system was configured in a controlled environment, test and evaluated, one variable at a time. Once more reasonable results are acquired, an actual circuit will be implemented for the final, all inclusive system test.

Overall, the work in this paper has demonstrated that it is feasible to use a fuzzy controller, buck-boost converter and solar panels in for charging batteries. Attempts to implement this controller with existing hardware proved challenging due to support/dated equipment. While various software and hardware were tested, it was determined that the best approach to designing a custom power solar robot is to design all the components from the ground up. With the exception of solar panels, the electronics should be sized and built according to the demand of the robot. In the systems current state, further simulation is needed before the design can proceed with a high level of confidence to the embedded hardware. However, as shown here, it is possible to start initial hardware testing and software coding. Another underestimated challenge that was encountered while trying to get the robot ready for implementation was that most of the hardware was sourced and acquired as commercial-off-the-shelf products. It was then necessary to also focus on manufacturing the robot, and not just electronics programming. Many hours were spent designing CAD files for 3D printing. Since manufacturing has become an iterative process, many parts were printed a multiple times to get the fit/functionality right. Although the buck-boost converter was built, applying a PWM duty cycle step to the circuit results in either low power output, or overheated components. Further research is needed on the MOSFET technology and the exploration of high power technology such as insulated gate bipolar transistors (IGBTs) as a substitute for this application is necessary. Once the hardware buck-boost converter is designed, then the fuzzy controller can directly drive the hardware, and final tuning can start. While it is possible to tune the fuzzy controller on the simulation, each 1 second run takes about 15 minutes to run, and it does not include all the unknown variables that the real circuit will have.

References

[1] AVR451: BC100 Hardware User's Guide-doc8088.pdf., Available: http://www.atmel.com/images/doc8088.pdf, 2016.

[2] AVR458: Charging Lithium-Ion Batteries with ATAVRBC100 - doc8080.pdf., Available: http://www.atmel.com/images/doc8080.pdf, 2016.

[3] Avr microcontrollers forums topic-megaavr and tinyavr bc100., Available: http://www.avrfreaks.net/forum/bc100, 2016.

[4] S. Bidyadhar and P. Raseswari, *A comparative study on maximum power point tracking techniques for photovoltaic power systems*, Sustainable Energy, IEEE Transactions on, **4(1)** (2013), 89–98.

[5] J. Fattal and P. B. D. N. Karami, *Review on different charging techniques of a lithium polymer battery*, In Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), 2015 Third International Conference on, IEEE, (2015), 33–38.

[6] Y. Fei and H. Lv, *Design of the solar-driven module on modular mobile robot*, Mechatronics and Machine Vision in Practice (M2VIP), 2012 19th International Conference, (2012), 470–473.

[7] HobbyKing ZIPPY Flightmax 5000mah 3s1p 30c, Available: http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct=8587, 2016.

[8] A. Kaplan and P. Uhing and N. Kingry and R. D. Adam, *Integrated path planning and power management for solar-powered unmanned ground vehicles*, 2015 IEEE International Conference on Robotics and Automation (ICRA), (2015), 982-987.
[9] J. Leitner and W. Chamberlain and D. G. Dansereau and M. Dunbabin and M. Eich and T. Peynot and J. Roberts and R. Russell and N. Snderhauf, *LunaRoo: Designing a hopping lunar science payload*, 2016 IEEE Aerospace Conference, (2016), 1-12.
[10] J. H. Lever, A. Streeter and LR. Ray, *Performance of a solar-powered robot for polar instrument networks*, Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006, ICRA 2006, (2006), 4252–4257.
[11] Ned Mohan, Power Electronics: A First Course, Wiley, 2012.
[12] Projects/avr_bc100.git., Available: http://git.kpe.io/?p=avr_bc100.git, 2016.
[13] L. Ray, A. Adolph, A. Morlock and B. Walker and M. Albert and J. H. Lever and J. Dibb, *Autonomous rover for polar science support and remote sensing*, 2014 IEEE Geoscience and Remote Sensing Symposium, (2014), 4101–4104.

J. Cruz-Lambert, Electrical and Computer Engineering Department, The University of Texas at San Antonio, San Antonio, Texas, USA
E-mail address: josueilambert@yahoo.com

P. Benavidez*, Electrical and Computer Engineering Department, The University of Texas at San Antonio, San Antonio, Texas, USA
E-mail address: patrick.benavidez@gmail.com

J. Ortiz, Electrical and Computer Engineering Department, The University of Texas at San Antonio, San Antonio, Texas, USA
E-mail address: jaqdandori@gmail.com

N. Gallardo, Electrical and Computer Engineering Department, The University of Texas at San Antonio, San Antonio, Texas, USA
E-mail address: hbq744@my.utsa.edu

B. A. Erol, Electrical and Computer Engineering Department, The University of Texas at San Antonio, San Antonio, Texas, USA
E-mail address: berat.erol@utsa.edu

J. Richey, Electrical and Computer Engineering Department, The University of Texas at San Antonio, San Antonio, Texas, USA
E-mail address: xje791@my.utsa.edu

S. Morris, Electrical and Computer Engineering Department, The University of Texas at San Antonio, San Antonio, Texas, USA
E-mail address: shane.morris33@yahoo.com

M. Jamshidi, Electrical and Computer Engineering Department, The University of Texas at San Antonio, San Antonio, Texas, USA
E-mail address: moj@wacong.org

∗Corresponding author