C.O.R.E
(Controller for Organic Range of Exoskeleton)

Daniel Reveron, Brandon Johnson, Kelvin Feliciano, Gavin Bell

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* --**

C.O.R.E. consists of a control system for an exoskeleton arm that will be attached to the upper limb of a wheelchair-bound patient with low-level limb functionality. The system will work electrically with 2 degrees of freedom for the hand and the elbow. When the patient applies any minimal force in a target muscle, C.O.R.E. will use electromyography sensors to read the bio-electrical signals originated by muscle contraction and produce movement. Electrical readings from the muscle will be processed by a control unit that will determine where to redirect the power. For the elbow, the EMG will also provide a stepper motor with information required to move in either direction for flexion and extension. The hand will only operate on a digital event, as the hand opens and closes. A feedback-controlled system will be set in place in the pneumatic system that controls the hand to better guarantee the safety of the user. What makes this exoskeleton arm special in comparison to its market counterparts is that it will be produced at a much cheaper rate than that of its average production cost by as low as 10% of the market value.

***Index Terms* -- Actuators, EMGs, pneumatic systems, microcontrollers, DC motors, control design, biomedical engineering**

I. INTRODUCTION

Limbitless Solutions is a non-profit organization that produces biomedical applications at low cost for patients with physical disabilities throughout the country. They are established on the University of Central Florida's main campus, where people with the same state of mind get together and find suitable, affordable solutions for a great variety of issues. Patients come from many different states and request diverse prosthetic parts and other devices that they can use in order to improve their quality of life. Under the same token, students and companies alike help the organization further advance for society and these patients by providing bright design ideas and funding respectively.

During the spring semester of 2016, several groups of mechanical engineering students from UCF came together, with the help from L.S., and thought up an idea that would not only improve the capabilities of a patient with limited limb mobility, but would also allow the patient to be fluidly integrated into society by feeling less hindered by robotic movement and the bulkiness that is prone to occur as a result of designing a physical device. The idea consisted of an exoskeleton arm called "Carapace" that would be fully made of soft materials and would be driven by pneumatic actuators. The exoskeleton would essentially read inputs from the patient's muscle signals and transform them into actuation outputs. Due to the complexity of the electrical framework that would be needed to control the actuators, the electrical design would require to be in its own project in order to develop; thus, the mechanical teams required a team that is specialized in the electrical components that would control the actuators and the software that would run the components. Since our team had planned to present a biomedical senior design at the same time as the mechanical team was to present theirs, Limbitless Solutions decided to join efforts and group both teams together in order to make this system a reality, while still preserving enough independence for both teams to complete their design without risking each other's success in case either team would not be able to deliver their design on time. Although the fully-pneumatic design showed great promise in the making of a fluid exoskeleton, concerns arose from both teams regarding completion of the project by the December deadline. Additionally, some changes to current design ensued due to the nature of the patient's disability, who suffers from Arthrogryposis Multiple Congenita. This condition is typically caused at birth and affects his joints by having them fused together. His muscles are also deteriorated due to lack of use, making any signals harder to obtain from the muscle. For this reason, the electrical team, also named group C.O.R.E. (Controller for Organic Range of Exoskeleton), proposed a solution that would guarantee a better adjustment to the patient's measures as well as the completion of the project by the given deadline. The solution consists of a hybridization of both electrical motors for the elbow and pneumatic actuators for the hand. Furthermore, the number of degrees of freedom to be regarded for input and actuation was reduced from eight to three. Two of these degrees of freedom would be operated electro-mechanically while the third would be of pure mechanical nature for the wrist. This design was approved by all the stakeholders involved in the system.

II. INPUT DESIGN

EMG stands out as a simple and cost-efficient approach to non-invasive signal acquisition. In a high level explanation, surface electrodes are applied to the muscle of interest and record any electrical activity that occurs. Although electrodes are typically implanted with needles through the skin, surface electrodes could be placed on the arm instead for a non-invasive procedure. The EMG approach would require less sensors than EEG to be placed on the patient, as most of the electrical activity in the muscle pertains to motor stimulation. Furthermore, the sensors could be easily concealed in the exoskeleton arm's framework, providing a more aesthetic design for the patient to use. Lastly, this method of reading muscle inputs would allow a more direct mapping between the inputs and the microcontroller, which will result in a simpler system. EMGs are typically less accurate, however, as the muscle cells are shielded by more layers of fat and skin, and the signal is usually weaker by the time it reaches the muscle fibers.

Overall, the use of EMG sensors shows a better promise than EEG for the scope of this project.

Due to the complexity of the human body in terms of expectations and behaviors, EMG calibration cannot be adjusted to fit any user's needs toward the exoskeleton arm; assessment must be made to fit each patient individually. Many factors must be taken into consideration when selecting the best products for this particular design. Moreover, the analogical nature of the project makes assessing the patient's condition in high detail a necessity in order to create a functional system. If the patient were to use an uncalibrated design that did not require previous evaluation of the user's capabilities, safety hazards would most likely be produced. For this reason, sound testing procedures must be taken. The most important benefit from measure testing is that it will allow the team to properly calibrate design to fit the patient's needs and capabilities. Another key advantage of assessment tests is that they will help determine what needs to be measured in order to properly constrain the system.

Several choices were defined for EMG sensor selections. Originally, the MyoWare Sensor was to be used for signal retrieval of the exoskeleton's control system. It consists of a small PCB sensor that contains two types of outputs. One output is the raw signal, which contains both positive and negative magnitudes from the arm, while the other output is the smooth rectified signal. Since the organization already possesses more than a few MyoWare EMG sensors and they have a built-in amplifier, the sensor is considered a great choice for the patient's EMG testing, which will be recorded with smooth-rectified analysis for simplicity.

For the system, the team shall be creating an architecture of 3 EMG sensors that will be integrated in a single PCB that handles all the processing called the Brain. The idea for EMG integration will be to reuse a schematic for the design that the organization employed for its known prosthetic limbs with servo actuation, and select suitable components that will then be planned out on the brain PCB. Since the exoskeleton will originally be planned out as a wheelchair-bound model, the EMG units do not necessarily need to be placed on the patient and can thus be somewhat bulkier than what the team had in mind at first. Additionally, the cost of using integrated EMG sensors would make a big difference with respect to MyoWare's due to the fact that they are made up of simpler components that can be purchased in bulk quantities. *EMG Sensor 1* shows some of the aspects in which an integrated EMG sensor proves to be a better choice.

		MyoWare		Integrated	
Criteria	Wt. %	Rt.	Wt. Rt.	Rt.	Wt. Rt.
Power Efficiency	20%	3	0.6	3	0.6
Signal Quality / Bandwidth	25%	4	1	5	1.25
Maintenance	15%	3	0.45	4	0.6
Low Cost	15%	3	0.45	5	0.75
Availability	15%	3	0.45	5	0.75
Portability	10%	4	0.4	3	0.3
Total	100%	3.35		4.25	

EMG Sensor 1: Why Limbitless EMG's prove to be a better option for integration

III. SOFTWARE DESIGN

A. *Original Software Design*

The high level software design was something that we looked at early on when designing our system. We didn't know specific components, but knew general groups. With this information, we began to develop a layout for what would be connected to what and how the system was going to operate. There were a few iterations.

The component groups that we have can be broken down as such: Input sensors, feedback sensors, the output system, a reset switch, and the microcontroller(s). The idea is that the input sensors would read input and there would be some output to the rest of the system. Feedback sensors would stop the output if they detect anything that deems it necessary and a reset switch could be hit to do a hard reset of the system if needed. A control unit would be needed to run the logic and make all of this happen.

One factor that influenced our design was the fact that we would need a large number of pins to attach all of our components to. Originally, there were to be at least eight input sensor, at least 13 outputs, and also a few feedback sensors. Another factor was that this processing is very time sensitive. There cannot be unreasonable lag on this system or the system becomes unusable and therefore worthless. The combination of these two factors lead us to consider an SPI design.

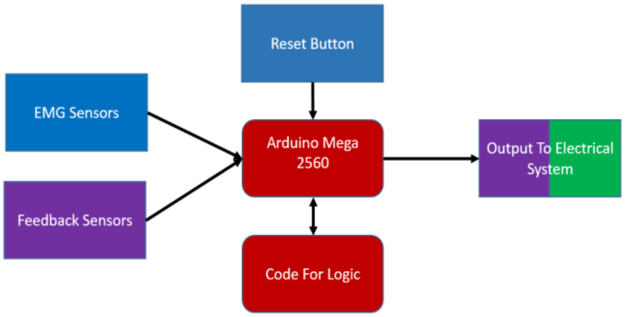
Our design was pretty straightforward. We were to have two Microcontrollers communicate with each other and co-process. This would solve the problem of having limited pins as now we would have the pins of two controllers to work with. This would also help with the processing speed as they could share the load.

The idea was to have the master (potentially a more powerful controller) take input from the input sensors as well as the reset switch and send corresponding signals to the output system. It would be coded to handle the logic for this. The slave would be responsible for just the feedback sensors. It would handle all of that logic and a send a signal up to the master. The master would read this and could factor the slave's response into its algorithm for deciding what to send to the output system.

B. Final Software Design

After the creation of the initial design, some constraints were changed from external sources. The mechanical team had a meeting and adjusted some things. The main thing that changed as far as the software was concerned is that they reduced the number of outputs needed. This in turn, freed up some pins. We decided to adjust the software design to something more practical for the current needs.

The significant change that was made to the design was that we got rid of the second microcontroller and made it a single processor system. There were some thoughts behind this. The first was that we no longer needed the extra I/O pins because we no longer had so many outputs to deal with. Another thought was that because it did not have to do such heavy processing on deciding which output to activate, some of that processing power could be used for the for the feedback sensors. We figure that if we use a fast enough processor, we should be able to manage the delay time without needing an extra one. Lastly, using a single processor will reduce the cost of the system. This is always good as it allows us to allocate funds to other components and resources. Once we finalized the hardware design and components, we were able to update our software design to be a little more specific. This final design is displayed in figure High Level Software Design 2.



High Level Software Design 2. This is the block diagram for our final high level software design.

C. *Low Level Software Implementation*

The flow diagram displayed in Software Low Level Design 1 shows how the code will work for the system. The functions that convert analog signals to voltage are for debugging purposes and do not generate to output in the production environment, but everything else in the diagram will play a vital role in the success of this system. The first thing to do when the program starts is to enable the motor and turn on the compressor. After this we go into our main loop.

In our loop we will begin by reading the values from all of the EMGs. We then check if the wrist value was high enough to warrant a response, if it is we go to the wrist function. In this function we check if the hand is open or closed and set all appropriate variables accordingly. This means setting the pressure sensor we want to be reading, the solenoid we want to control etc. Next we open the solenoid and continuously read the pressure. Once the pressure we want is reached, we close the valve. This will give the effect of opening/closing the hand. If the wrist value is too low, we ignore it and skip this step. Next it will check if the biceps or triceps value is higher and we go to function of the leading muscle group. If they are equal, neither will be done. The biceps and triceps function are mostly identical. We check the EMG value; If the value is over the threshold we set the direction of the motor and move it at a preset speed. If the value is not above the threshold, we ignore it and do nothing.

After the biceps and triceps functions are complete, we do a small delay for stability and restart the loop. The program will continue to loop through and process input from the user.

Software Low Level Design 1. This is a flow diagram of how the system will be coded.

D. Microcontroller Selection

The selection of a microcontroller is something that had to be taken very seriously. It will be responsible for being the brain of the entire system and thus should be able to handle certain needs. There are a few metrics that we used to give quantitative values to some properties. A few of these include ease of use, low cost, number of I/O pins, processing speed, availability of external components, power requirements, and support community.

Ease of use is the first metric. This is important because we only have a limited amount of time to complete the project. We need to use a controller that can be learned and utilized within this time. Also, there are many variables in our project coming from outside forces, such as patient needs or last minute changes in the mechanical team's design. We need to be able to make quick code changes on the fly without having to restructure our entire code base. This can be difficult in certain situations like when the code implementation tries to take the programmer too low in level and does not abstract enough details.

Another factor in controller selection was cost. Though this project is sponsored by Limbitless Solutions, there is still a budget to follow. The ideal controller is not going to cost too much and blow through the budget. One important thing to keep in mind, however, is that while we are looking for a great price, we do not want to sacrifice quality in the process. We only looked at genuine parts from the manufacturer and not the cheap clones that are sold across the internet.

The next metric is number of I/O pins. The implementation of our design requires many inputs and outputs. The EMG sensors alone should take up quite a few. In addition to this, we want to have a few left over in case we need to add more components or are given extra requirements from the mechanical team. With the original SPI design, having two controllers would alleviate some of the pressure of supporting enough pins. However, when we decided a single controller would be better, pin count became a serious concern once again.

Processing speed is also a factor. Because our system is going to be attached to and used directly by a person, the response time needs to be very fast. We need a controller that can provide an adequate clock speed. It needs to be able to sample input at a reasonable rate and run calculations fast enough to process it to output before the user can experience noticeable lag.

Next, we examined availability of external components. We wanted something that would be easy to attach things to, especially for testing purposes. Some of the issues faced were, whether or not the screen would be able to display information. Also, if there were adapters and relays. Then there was the question of whether there are libraries supporting the code for this project. Some microcontrollers had better information regarding these questions than others.

The power requirements of the controller is important because we need to limit the drain on the battery. This device needs to be available for use hours at a time. We didn't want something that would consume too much battery life and make it harder to reach this objective. We looked in controllers that supported things like "Low-Power-Mode" or that drew minimal amounts of energy in standard usage.

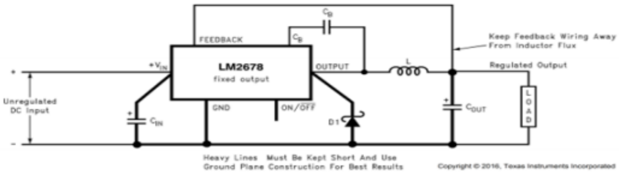
The last factor we looked at was community support. This factor could easily be forgotten or overlooked, but we found it to be very important for a few reasons. In order to keep down the price of this system and stay within budget, we aren't able to purchase extensive documentation (data-books, manuals, standards, etc.). Also, the time it could take to read through everything and pull out the useful information could be very lengthy. Lack of support would also mean that simple things could become very difficult when we have to use specific bit codes and implement things at a low level. A large support community means that things have been done before. If we hit an obstacle, we want to be able to find solutions relatively quickly. There is also expected to be a large number of support libraries to assist with implementing common things, such as reading to an output screen.

In the end, for the reasons listed above, we end up going with Atmel's Atmega2560. We used the Arduino Mega 2560 development board for testing.

IV. HARDWARE DESIGN

A: Voltage Regulator

From a huge selection of voltage regulators that are sold by multiple companies, the team's choice was the LM2678. The LM2678 is a step down voltage regulator, which has all of the characteristics that were needed for this project. This component is manufactured by Texas Instruments, which is known for its wide selection of electrical components. Some of the reasons why this voltage regulator was chosen by the team is because the LM2678 voltage regulator can reach up to an efficiency of 92%, is simple and really easy to design. Its design easiness comes from the fact that the components needed to complement the design are easily accessible. This voltage regulator has many advantages that others cannot replicate. This step-down voltage regulator has a real wide input voltage which gives us some buffer in case we have to increase or decrease the power source that we will be using, in this case our 25.6 V battery. This voltage regulator has all the active functions of a switching regulator which is capable of driving loads that are up to 5 A. This buck converter could be set up to be used to get multiple output voltages. Therefore, this component can be used to control multiples circuits in the design. By just replacing some of the components values in the adjustable version, we can change the output to be higher or lower. Once we knew the exact output values that we were looking for we were able to find the same regulator but with the fixed values. By using the fixed value regulators we were able to use two components less per regulator circuit. The fixed value regulators that were used for this project were: the 12 V and the 5 V. In Voltage Regulators 2 shown below, the typical circuit for a fixed output in an LM2678 model is shown. This same circuit with different values will be used to regulate different components in the system. To illustrate, the team will regulate the voltage that will be feeding the MCU to a max of 5 V. For the air pump, and the solenoids the team will be setting the circuit to allow an output of 12 V. By doing this, the team is making sure that every component is getting a constant input signal to avoid any fluctuations when in operation. Below in figure Voltage Regulators 2 we can see the diagram for the switching regulator LM2678.

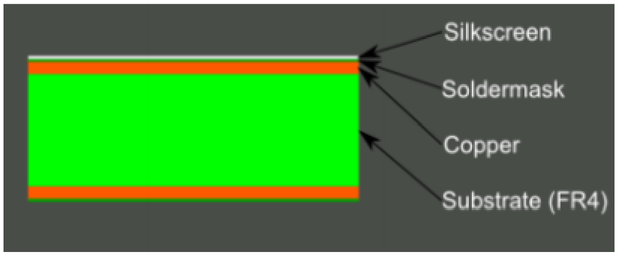


Voltage Regulators 2: Diagram for a fixed LM2678 buck converter.

B: Printed Circuit Board Design

Before we even consider to start a PCB for this project, all the components were evaluated and tested on a breadboard. Having a prototype is essential because once the schematic is sent out for fabrication, there is little to no room for additional changes on the PCB. Additional to the breadboard, we also solder some of the surface mounts circuits to simulate how would they behave once in the pcb.

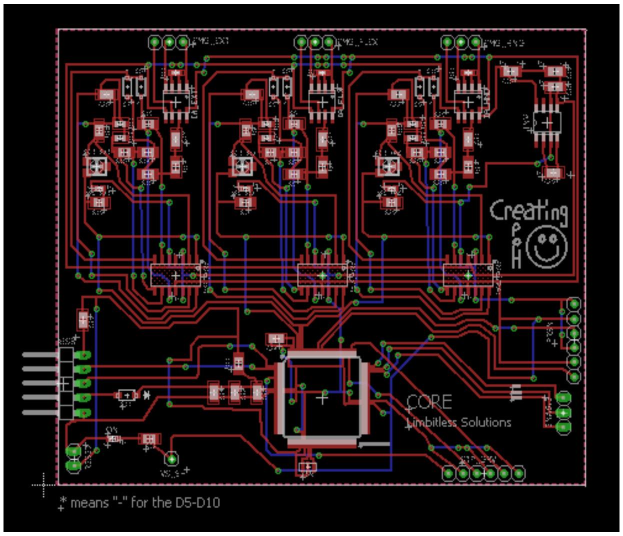
Once initial testing phase is finished, the team can then move to design the circuits in CAD software for PCB design. There are multiple programs that allows for building a PCB; but EAGLE will be used. This is a free software that has many tutorials and a big community online on how to use it. This software has many libraries with parts that will be needed in order to create the wiring schematics for our circuits. For those parts that are not in the library of EAGLE, this program has the capabilities of creating a library in which components can be added by designing them. Since this program is free, its license is limited as it only allows building a double sided PCB. A double-sided PCB should be more than enough to create the design. In the figure below Printed Circuit Board 1 there is an example of how a dual layer pcb looks like.



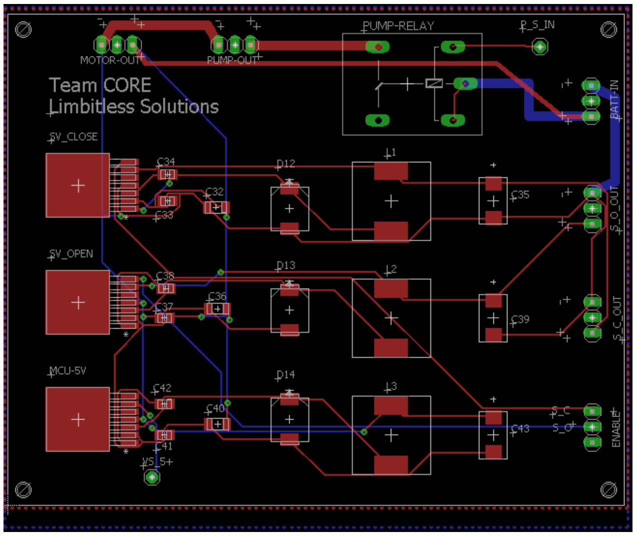
Printed Circuit Board 1: Simple construction of a double sided PCB.

Once the schematics were done, EAGLE gives the option to create a file for both the schematic and the board layout. We made sure to find the company requirements before we send out the files. The files wanted by the supplier were the Gerber files. The Gerber files have specific information about the board layouts, which include traces' width, copper wires, where holes must be drilled, etc. These files contain the necessary information that companies need in order to build the boards in a language that their machine can understand. It is crucial that these files are revised by the team members before they are send out to the manufacturer. When receiving the PCB, attention must be paid not to install components or other parts onto the board right away. First, the team must verify and compare to current design prints. Second, the team shall analyze the board to make sure everything is where it is supposed to be. Third, the team is to physically test point to point connections by the use of continuity with a multimeter. The university offers various locations where circuits can be built and tested. These facilities have to be used to the team's advantage accordingly. If any issues are found with the board, the team will have to quickly figure out what is wrong with it. For that reason, the prototype that was made on the breadboard should stay there for as long as possible. It will be the team's best reference.

For spatial reasons, the team decided to go with a design that comprises 2 PCBs. The PCB labeled "Brain" holds the EMG structures as well as the MCU, the pressure sensor input and the outgoing traces to the "Heart" PCB. The latter PCB will hold the regulators that distribute voltage to the different electro-mechanical parts as well as the MCU. The Brain shares 2 signal connections with the Heart's enable pins for the regulators that control the solenoid valves. When the hand is actuated, these pins will turn on the regulators that will allow current to run through the solenoid valves, thus turning them on and allowing for airflow. The reset button was placed outside the PCB to allow the patient to easily reset the system in case of instability. In a similar manner, the Heart provides constant power to the Brain PCB, which in turn provides power to the MCU, EMGs and pressure sensor. Additionally, the Heart contains the relay that turns the pump on and off with help from the pressure switch that is integrated in the pressure tank. *Printed Circuit Board 1* and *Printed Circuit Board 2* show the board layout for both Heart and Brain PCBs



Printed Circuit Board 2.: Brain PCB with chip, EMGs, pressure sensor, and driver controls.



Printed Circuit Board 3.: Heart PCB with regulators to solenoid valves and MCU as well as a relay for the pump.

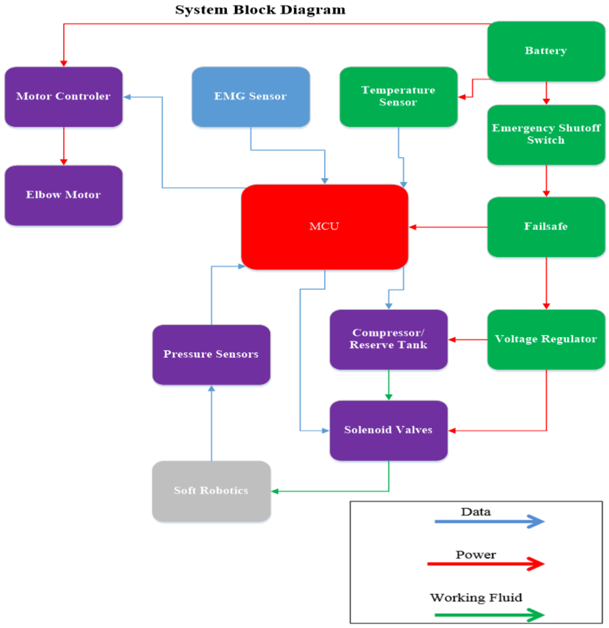
V. ELECTRO-MECHANICAL DESIGN

A. Electro-Mechanical System Overview

The electro-mechanical system for the exoskeletal arm will be comprised of two subsystems; one for the actuation of the hand and another for the actuation of the elbow. These systems will be controlled by the MCU which will receive signals from the EMG sensors to then be interpreted into movement. For the hand actuator the pneumatic system will only perform two actions either being open or closed. The pneumatic hand system will be using a series of compressed air tanks, valves, and pressure sensors to control the soft robotics finger actuators which will comprise the hand subsystem. The actuation of the elbow will be based on a fixed amount of rotation, rotating either clockwise or counterclockwise for each of the EMG signals received. The elbow will rotate through the same range of motion as a typical elbow would. However, there would be mechanical stops for the elbow such that overextension would not be a possibility. A stepper motor will be used to articulate the movement of the elbow and the user will visually verify the position of the elbow. Below is the Electro-Mechanical System Overview 1 figure of the overall design of the exoskeletal arm as a system.

This section of the exoskeletal arm will be where the electrical design and the mechanical design join together. Creating a device which meshes seamlessly together is one of the most noticeable, or unnoticeable if done correctly, attributes a product can have. Without all of the components for the device working properly together then the entire device will be a failure. In a way a product is much like an orchestra, each section has its own part to play and when combined the symphony comes alive. However, if that orchestra was missing the woodwinds section then it would be incomplete and your experience would be less than the full symphony. This is why a product must have all of its components working together in unison to product the full symphony. Many times we find devices which have excellent hardware but terribly design software and vice versa. Those are devices which typically fall behind in the markets. The products that tend to rise to the top are usually the same ones that have a coherent flow between the different sections as if each were designed with the other in mind. This is why the electro-mechanical subsystem of the exoskeletal arm is crucial to the success of the entire project. While is it easy to come up with ways to solve problems, the hard part is coming up with good ways to solve problems which mitigate risk and improve performance.

The design intent of this system is to make it simple while retaining function. This particular subsystem should be designed it the least bulky manner such that it does not cause user to avoid utilizing this tool simply because it is more of a hassle than a help. Even if the design is minimal but remains something which will reap additional use then that is an ideal design. The goal is to make something which will be used not something that is extensively overbuilt and complex. The design intent is to make this system function in a failsafe manner, much like regulators used in SCUBA diving. The valves on these regulators are designed in a manner such that if the first stage regulator should fail, thus over pressurizing the system, then the second stage regulator, which is the part located in the diver's mouth, will free flow allowing the diver to still breath air from the system. SCUBA is normally viewed as a recreational activity, however the gear is life support equipment and is designed with multiple modes of failure in mind. For the exoskeletal arm similar modes of failure must be identified and mitigated to make this a viable product for the patient to utilize on a daily basis.



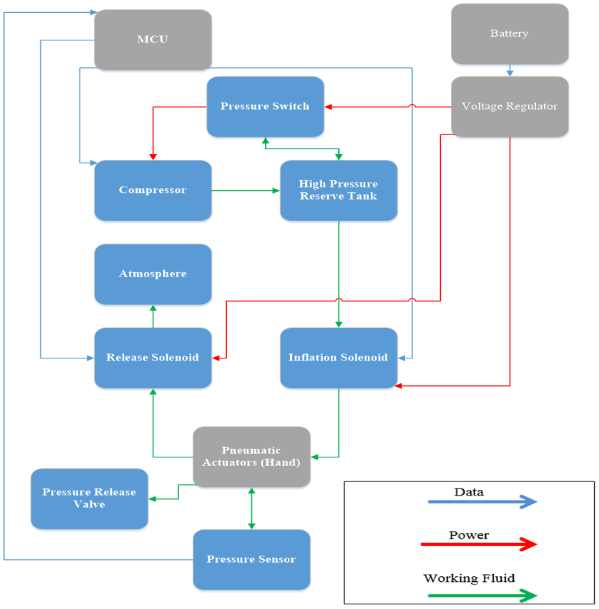
Electro-Mechanical System Overview 1: Exoskeletal Arm System Block Diagram

B. Hand Actuation Subsystem

The actuation of the hand will be accomplished through the use of a pneumatic system. This pneumatic system will be comprised of a compressor, tank, two solenoid valves, pressure sensor, pressure switch, pressure regulator, and two pressure relief valves. The use of the reserve tank system will allow for the hand actuator to react quickly and not be limited by the power of the compressor to constantly be either inflating or deflating the hand actuator. The compressor will compress the air from the atmosphere into the high pressure tank. The working fluid will be simple air direct from the surroundings.

There will be two solenoids on the tank. The solenoid on the high pressure tank will be opened to inflate the pneumatic hand and to deflate the hand the solenoid open to the atmosphere will be opened to deflate the pneumatic hand. The pressure sensor will be placed such that it can read the pressure in the hand actuator. The sensitivity of the pressure sensor will need to be enough to detect a pressure to 1 psi or better. The pressure sensor will then be able to read the pressure and relay the information the MCU. Then the MCU can decide which valves need to be opened or closed. The solenoid valves themselves will be the type which remain closed until energized.

There will be a pressure relief valve located on the tank as well as on the pneumatic actuators themselves. The pressure valve will maintain the safety aspect of the design; this will ensure that no portion of the pneumatic system will reach a pressure that could become dangerous. Below is the figure Hand Actuation Subsystem 1 which define the actuation subsystem for the pneumatic hand in terms of hardware.

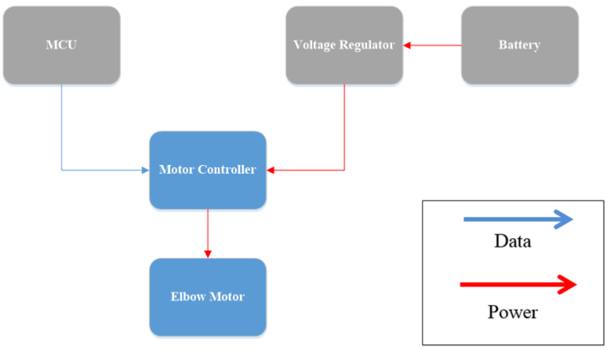


Hand Actuation Subsystem 1: Electro-Mechanical Hand Subsystem Block Diagram

C. Elbow Actuation Subsystem

The actuation of the elbow will be accomplished utilizing a stepper motor. This system will be comprised of two main components; the stepper motor and the motor controller. The system will read the inputs from the EMG sensors placed on the subject's arm, then the MCU will interpret the signals to control the motors position. A stepper motor will be used due to its ability for fine position control. Typically, the position of a stepper motor can be controlled within one degree of angle or better on a repeatable basis. The user will provide feedback of the actual position of the motor and adjust accordingly if necessary. Below is the figure Elbow Actuation Subsystem 1 which define the actuation subsystem for the electrical elbow in terms of hardware.

Electro-Mechanical Elbow Subsystem Block Diagram



Elbow Actuation Subsystem 1: Electro-Mechanical Hand Subsystem Block Diagram

VI. BIOGRAPHIES

Daniel is an undergraduate studying a double major in Computer and Electrical Engineering. He has accepted a job offer at Lockheed Martin as a Systems Engineer starting in 2017. Daniel plans to pursue his Masters' degree after graduation and his main interests are biomedical applications and nanotechnology.

Gavin Bell is currently completing his bachelor's degree in Electrical Engineering at UCF, from which he has already graduated with degrees in both Aerospace and Mechanical Engineering. His areas of interest include space exploration, renewable energy systems, and biomedical engineering applications.

Kelvin Feliciano is 31 years old and will be graduating the University of Central Florida with a Bachelor's degree in Electrical Engineering. Upon graduation he will pursue a career in the engineering field with the hopes of returning to school and obtaining his Master's degree. His areas of interest include power electronics, electric vehicles, renewables and their interface to power systems.

Brandon Johnson is a student at UCF, completing a Bachelor's degree in Computer Engineering. He has accepted a Software Engineering position with Galatea Associates and plans on obtaining a Master's degree in the future. His interests include software development and database management.

VII. REFERENCES

[1] Texas Instruments, "LM2678 SIMPLE SWITCHER® High Efficiency 5-A Step-Down Voltage Regulator," 2016.

[2] Sparkfun, "Using EAGLE: Board Layout," August 2016. [Online]. Available: https://learn.sparkfun.com/tutorials/using-eagle-board-layout.