**LDRD – Maximizing Energy Efficiency in Quadrupedal Robots**

**Project Summary**

Christopher LaBorde

Over the course of 10 weeks, we are tasked with designing and developing a small, ambulatory “bug” robot. The goal of the internship period was to design the robot – then to develop a machine learning algorithm that reduces the amount of energy consumed for the robot’s walk. We want the algorithm to allow the microcontroller to receive feedback from power and distance sensors, and to enable it to learn what the correct hardware parameters are for the walk. The walking is performed by a set of servos, with each one controlling a pair of legs.

The work is being done as a Lab Directed Research and Development initiative under the direction of Dr. Steve Xiao in the Research and Development Engineering group at the Savannah River National Laboratory.

**Methods**

The electronics of the robot are fairly simple. We have the PKCELL LP503562 3.7V 1200mAh battery, an ESP32 Microcontroller, an Adafruit INA260 Power Sensor, a VL53L1 ST Microelectronics Time of Flight Lidar Range Sensor, and MG995 Servomotors.

First we had to build the hardware and familiarize ourselves with the development Environment for the programming of the microcontroller. We downloaded the Arduino IDE and ran tests for programming the ESP32 controller. We practiced programming input/output pairs to operate the servo motors and get readings from the power sensor and Lidar.

Before we were able to have our robot walk, we had to assemble the separate electronic components of the robot. We had to combine various circuit boards, specifically the boards referenced above. To do this we consulted the equipments’ technical data sheets to identify pins and their functions . We were able to learn and practice the art of soldering in the electronics and instrumentation lab on site.

Once the robot is assembled, and the software for the sensor components has been tested and debugged, we are ready to collect the data needed for the algorithm. We begin the walking process by feeding the program predetermined values for the parameters. In this case we choose center +- (15 degrees) sweeps for both the front and rear servos. For the time between movement of the front and rear servos we set the value to 300ms. The robot walks with these parameter inputs and we get a reading of the distance traveled and the power used over the course of the walk.

In order to train the algorithm, we must record the data on both the input and the output side. For the input variables we have 4 parameters, the angle change from center for the front and rear servos, the timing between steps for the servos, and the time period of the stepping loop.

A successful machine learning algorithm is comprised of several necessary components. It must have a set of **data** on which to train. It has a prediction **model**, a mathematical model that relates a set of input variables to a set of output variables. It has an **objective function** AKA loss function, a function that compares the outputs from the prediction model to the actual output values. Lastly it has an **optimization algorithm**, a way to update the parameters of the prediction model.

**AI method**

The AI part of the project was the most challenging for me, as it was difficult to figure out even where to begin with the programming.

To be transparent, I was not able to test the following method. I had gotten to the point of being able to collect reliable data when the internship period came to an end. The following is an untested and relatively undeveloped method of implementing AI.

Initially, my vision was to sweep through combinations of parameters manually, saving the specific energy/distance values for each combination. This would then provide a “map” of reasonable or expected values for the output depending on the specific input parameter combination. Then, using a supervised machine learning algorithm, the microcontroller could be programmed to update a simple multi-variable model, using a variable for each input parameter. The predictions produced by this model could be checked against the known output values, and the model could be refined to it’s most accurate point.

The goal of the algorithm is to have the microcontroller identify the set of parameters which results in the lowest energy/distance ratio. As the controller manually or programmatically sweeps through combinations of input values, we could keep track of the lowest obtained output (energy/distance) value. Then, once the model is made, we could program the instruction “if a change in this parameter results in an increase in the energy/distance value, then change this parameter in the other direction, or revert the parameter value back to the previous value”. This way, ideally, the input parameter values would only change in a way that resulted in a decrease of the output value.

One possible challenge with this method is that the algorithm might find a local minimum and label it as a global minimum. To address this potential pitfall, the input parameter values could be initialized at different random combinations (within reasonable ranges, so that the robot could still walk). Starting at these different locations, and following a gradient-descent-like method, the algorithm would find different local minima. Repeating this enough times, we would obtain many local minima, and could compare them to obtain a global minimum.

The above method is a form of supervised machine learning. Supervised machine learning means we already know the outputs that we want the model to be able to recognize. The outputs in this case we have from our “map” of input-output relationships that we obtained from sweeping over the field.

We were encouraged to use a technique called backpropagation in our algorithms. Backpropagation, put simply, is observing how the incremental change of a parameter value affects the accuracy of the model overall (the backpropagation of error through a model). In the limited research that I’ve done, it seems to me that backpropagation is a method used only in supervised machine learning. Using a different approach, it is not clear to me how backpropagation would be implemented.

Other methods of implementing machine learning algorithms should be explored. Henry Goff sounded pretty confident about a method called Q-learning, which might warrant further research.

I am thankful to have participated in this research, even if our stated goal was not reached. It’s been an honor working under Dr. Xiao, and Matt Folsom and Caleb Scott in the Advanced Technologies Laboratory at the Savannah River National Laboratory.

**Challenges –**

Over the course of the summer, we experienced small impediments to progress. One such recurring impediment was that the wires making connections between boards often failed. Either the solder joints themselves would fail, or the wires would break at points due to high stress. I frequently saw interns carrying the soldered boards by the wires, or setting the boards up on the chassis in ways that strained the joints. A reminder or a notice up front to avoid handling the robots by the wires in ways that causes undue strain might be a low cost way to save headaches down the road.

Another challenge was experienced in the realm of logistics. The intern group was quite large this summer. There were 7 of us, in a lab with 2 full-time employees and frequent visitors and we often had 10 or more people in the lab. Sometimes deskspace available on which to work was difficult to find. This could sometimes make it difficult to have physical surface on which to place a laptop to program, or research online, and keep a lab notebook, and build or experiment with the robot at the same time. It is easy to overlook, but I believe that attention to this basic need would facilitate the experience substantially.

Reliable Internet access was another issue faced over the summer. There was a LTE Hotspot put in the lab, but it only supported 4 devices at a time. This meant that only 4 people could be connected to the internet on the Wi-Fi at one time. This proved frustrating when many of us were trying to research solutions for issues we were encountering and methods of going forward or implementing AI on the microcontrollers.

The large number of people also could be distracting at times. Communication in teams is so important, but I noticed over the summer that oftentimes I was distracted by communication around me that seemed irrelevant to the project. I enjoyed making friends and building relationships with the other interns, but sometimes it seemed that the amount of people was more of a hindrance than a help. Having a designated quiet space like an office or another area could have helped. (I was not able to obtain a clearance for telework over the summer, but I think that option was a good solution to any overcrowding in the lab)

Ok, I’m done griping. These are not meant to be complaints but rather hindrances that I perceived over the course of the summer.

**Lessons Learned:**

* **Read the documentation -** There was so much rich information online. Although it could seem overwhelming, sifting through documentation was one of the most effective ways to find a solution to a problem. Watching Matt with 12 years of experience troubleshoot a problem was an enlightening experience. He regularly referenced datasheets, online forums, etc to find solutions to technical problems.
* **Don’t be afraid to ask for help** - Often, asking for help from a more experienced mentor would solve a problem in minutes that I had been struggling with for hours. Then, watching the solution process was also enlightening and informing as an observer. The value of experience is immense.

**Successes:**

TCP/IP Wireless communication between ESP32 Server and computer client

Coded simple Linear Regression “Neuron” in both C++ and Python programming languages

Successfully assembled bug robot with dual servo action and lidar and power sensors.

Practiced the skill of soldering connections between integrated circuit boards

Achieved multi-core processing on microcontroller, having 2 functions run simultaneously

Effectively managed control flow of logic in program, integrating constant sensor reading with intermittent servo actuation