Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

First Prototype Test Report

MOSS Wheelchair: Modular Open-Source Smart Wheelchair

by

Team 11
MOSS Wheelchair

Team Members

Sebastian Gangemi sgangemi@bu.edu
Aya Kassem ayak@bu.edu
Norbert Malik norbert@bu.edu
Maha Noor mahakdn@bu.edu

Equipment and Setup

Motor Controller:

- 110VAC-24VDC power supply
- Brushed Motor
- Arduino Uno
- Electronic Motor Speed Controller
  - (regulates the voltage from the power supply into the motor according to the analog signal sent by the Arduino)
- Breadboard for connections and Turn-On button

Object Detection:

- Laptop with webcam activated
- In the same directory on that laptop:
  - The pretrained YOLOv8 model, *yolov8_tuned.pt*
  - The object detection Python script that instantiates the pre-trained model above, *detect_objects.py*
- Objects that you want to detect. Because the model we trained was pre-trained on the COCO dataset (common objects) and then fine-tuned on a dataset of doors and door handles (inspired by the fact that we would want the robotic arm to detect doors and door handles to open doors), we wanted to test both groups of objects the model was trained on.

  For this reason, detection was done on objects lying around us at the time of testing, such as:
  - People (us)
  - A bottle
  - A backpack
  - Calibers

  As well as:
  - A door
  - A door handle

  To test the fine-tuning of our model on these objects.

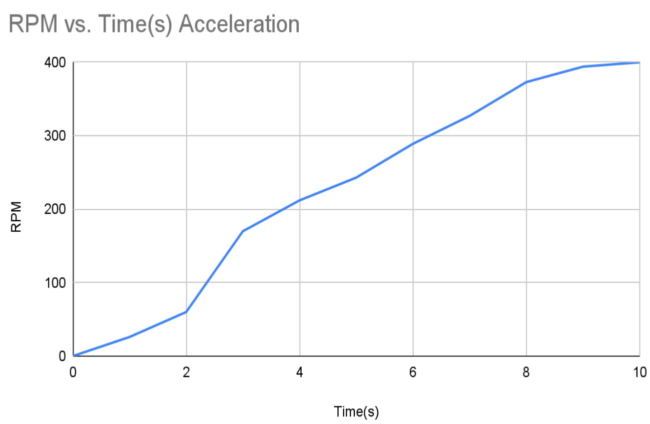
Robotic Arm:

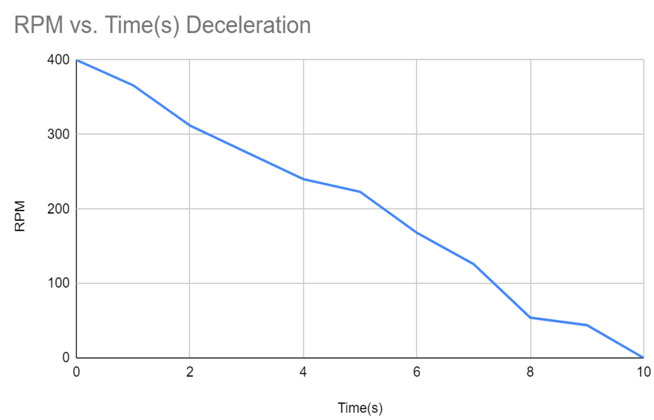
- 3D printed arm (hand, forearm, and bicep) from inmoov.fr
  - Fasteners and adhesives to hold the arms together
- Strings for facilitating the movement of the arm
  - Running string through finger and hand to be pulled on, causing the finger to bend one way or the other

Measurements Taken

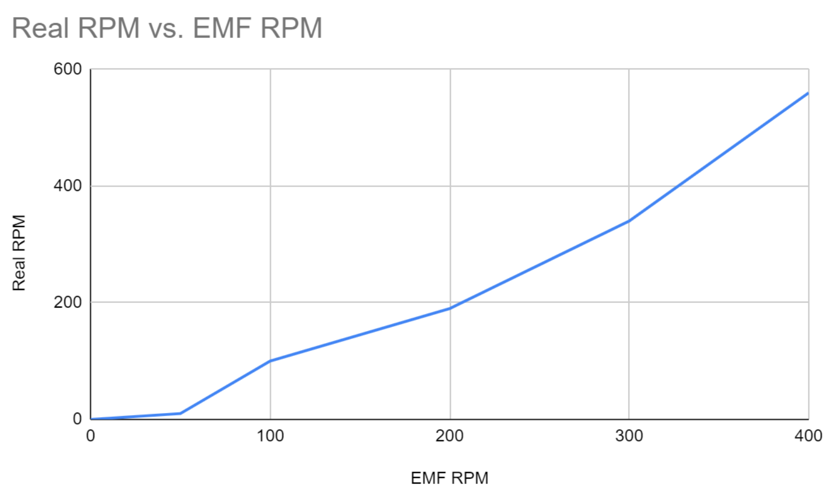
Motor Controller:

- Time vs (EMF estimated)RPM to visualize smoothness of acceleration and deceleration in the clockwise direction:





- EMF estimated RPM vs Actual RPM to test the accuracy of the Electronic Speed Controller:



Object Detection:

- For each object, we recorded:
    - Whether the model detected the presence of an object (even if it was labeled wrong). The reason we were interested in this measurement was that in the case of obstacle detection and collision avoidance during navigation, detecting the presence of an object, no matter what the object is, is often enough to avoid

obstacles or collisions. For example, we can stop the wheels' motors as long as an object is detected in front of the camera sensors, no matter what the object is. If the presence of an object is not detected at all, however, the wheelchair might end up colliding with that object.
    - Whether the detected object was labeled correctly or not. In the case where we want to detect objects for everyday tasks such as picking objects up, pushing elevator buttons, opening doors, etc., the object must be detected correctly since the task executed by the wheelchair's robotic arm will be specific to the label of the object detected.
    - Whether the object was detected in less than 1s.

- The following results were collected:

Object	Detected Object in Frame?	Labeled Object Correctly?	Label Given by Model (if labeled incorrectly)
Person	Yes	Yes	N/A
Bottle	Yes	Yes	N/A
Backpack	Yes	Yes	N/A
Calibers	Yes	No	Cellphone
Door	Yes	Yes	N/A
Door Handle	Yes	Yes	N/A

- Approximate detection time for each object: 300 ms

Robotic Arm:
- Each joint angle was taken to find cumulative finger bend
    - Knuckle Joint: 110 degrees
    - Proximal Finger Joint: 90 degrees
    - Distal Finger Joint: 90

Conclusions
Motor Controller:

While the acceleration seemed to be very constant based on the EMF feedback, we discovered that there are some ranges of EMF read RPM that are not sufficiently accurate. The test showed that the EMF feedback is close to 100% accurate in some ranges of RPM, however, it can be somewhat inaccurate in other ranges of RPM. At very low RPM (0-50) the EMF overestimates it greatly (500%), but this should be negligible as it only holds true for very small RPM. The lower and middle range of the speed controller are fairly accurate, but as we approached the top 20% of the RPM range in which the speed controller is programmed to allow (400RPM+), it began to underestimate the true RPM (as low as 70%). By adjusting the EMF constant in the speed controller, and the max RPM scale value, the range in which the EMF RPM reading is sufficiently accurate changes (lower EMF constant, more accurate at higher ranges). While we obtained sufficiently accurate results up to 300 RPM in this motor with the weak power supply (1 Amp max), when we receive the actual motor and power supply we will be operating in the range up to 2000 RPM. This will call for a greater max RPM scale value and a lower back EMF constant. It is important to note that EMF readings are generally more accurate in this range as PWM speed controllers are designed to operate linearly in such higher ranges. With the test results combined, we are confident that we will be able to accelerate sufficiently smoothly at the desired ranges of RPM.

Object Detection:

The test showed that the model detected the presence of an object in the frame 100% of the time and that it both detected an object and labeled it correctly 83.33% of the time. Despite these numbers being satisfactory for our testing plan's measurable criteria, the sample size of this test was quite small because of the time limit of our test. As part of our next steps, we would like to get a better estimate of the capabilities of our fine-tuned model by experimenting with a much bigger and more realistic sample size. This first prototype test also allowed us to test the training process that we used for our model: we imported a model that was pre-trained on Microsoft's COCO dataset, and we then fine-tuned it on a custom-made dataset that contains instances of objects that we need to detect specifically for the purpose of our project such as doors, door handles, and elevator buttons. Successfully detecting objects from both the COCO dataset as well as the custom-made dataset validated our training method. Our next step is to add more objects to our custom dataset (street signs, for example) and attempt to train our next model with more training epochs to see whether it would do an even better job at accurately detecting and labeling objects.

Robotic Arm:

The test showed that the fingers are capable of closing and keeping a closed shape similar to that of a human hand. There was an issue however with the rigidity of the finger and getting it to open and close fully without getting stuck. We will be looking into how to improve the fluidity of finger motion to decrease the force required to close it. We will mainly accomplish this by decreasing the friction between joints in the finger. The idea of a redesign to decrease the number of joints and overall parts to decrease friction and make general processing and fabrication easier is being considered. However, we will continue to build and test the rest of the robotic arm as a potential contender for our final design.