Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

Second Prototype Testing Plan

MOSS Chair: 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

I. Required Materials

1. *For Motor Controller:*
    A. Hardware
        - Wheelchair
        - 2 50 W 24 V Brushed Motors
        - 2 Motor controllers
        - 2 24 V AGM Batteries
        - 2 Single-ended quadrature wheel encoders
        - 2 Single-ended to differential quadrature signal converters
        - Joystick
        - Arduino
        - NVIDIA Jetson Nano

    B. Software
        - Simple Arduino script
            - Controls RPM of the motor through the motor controllers
        - Simple NVIDIA Jetson Nano Script
            - Uses PySerial library
            - Communicates with Arduino to send commands to move forward, backward and stop

2. *For Object Recognition Software:*
    A. Hardware:
        - NVIDIA Jetson Nano
            - Flashed microSD card
            - Power adapter (2.5 A 5 V)
            - WiFi card
            - WiFi antennas
        - Raspberry Pi SC0024 IMX219 NoIR Camera

    B. Software:
        - Python 3.6
        - Jetson Inference
        - PyTorch with CUDA suppport
        - TorchVision
        - GStreamer

II. Set-Up

The equipment and setup are divided into 2 parts: The motor controller, and the object recognition software.

Motor Controller: There are two separate controlling systems we will test across a variety of loads. The first controller is the joystick which requires 3 connections: Power, Left Motor, and Right Motor. (Power is always last to plug in and first to unplug) The second controller will be an Arduino microcontroller. In this setup, each motor drive gets connected to Power and its respective motor, and the Arduino gets plugged into the NVIDIA Jetson Nano to receive speed inputs from the serial monitor. The reason for the Jetson Nano is that, unlike the Arduino, it does not have to be directly plugged into a laptop and it can rather be controlled remotely through SSH. We however still need an Arduino because the Jetson Nano only has 2 PWM output pins, while we currently need 6 to control the wheelchair. We therefore used PySerial to make the Jetson Nano communicate user inputs to the Arduino from a remote laptop, and the Arduino then relays the commands to the rest of the hardware. The Arduino sends the speeds to each motor drive, and the motor drive regulates the speed from 4 rpm to 16 rpm for whichever load it is currently tuned for. To switch loads, as there is no current weight measurement to dynamically change the tune, you must upload the corresponding file to both motor drives one at a time.

Object Recognition: The Jetson Inference library that we are using already contains a pre-trained Inception-V2 model trained on Microsoft's COCO dataset, which contains multiple instances of common objects. Today, we will import this model and use it to detect objects in real time from the Raspberry Pi SC0024 IMX219 NoIR Camera connected to the Jetson Nano and classify whether these objects were detected correctly or not.

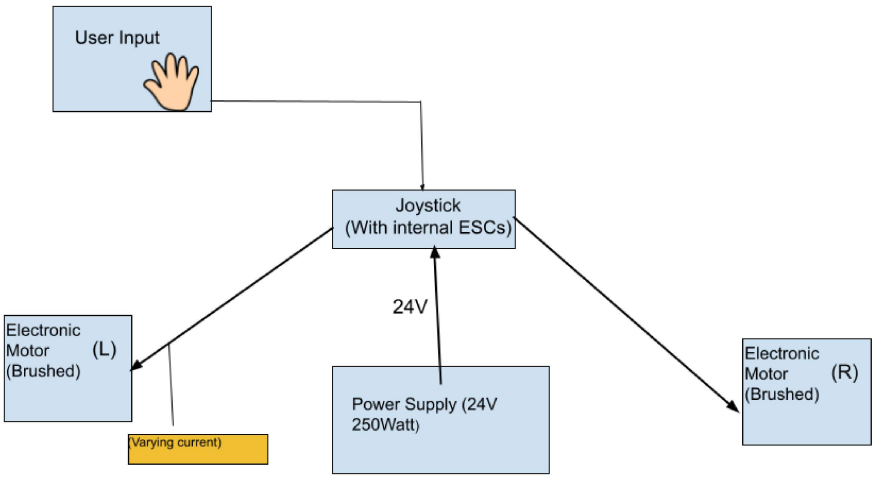
III. Pre-testing Setup Procedure

Motor controller:

Note: Power is always last to plug in and first to unplug

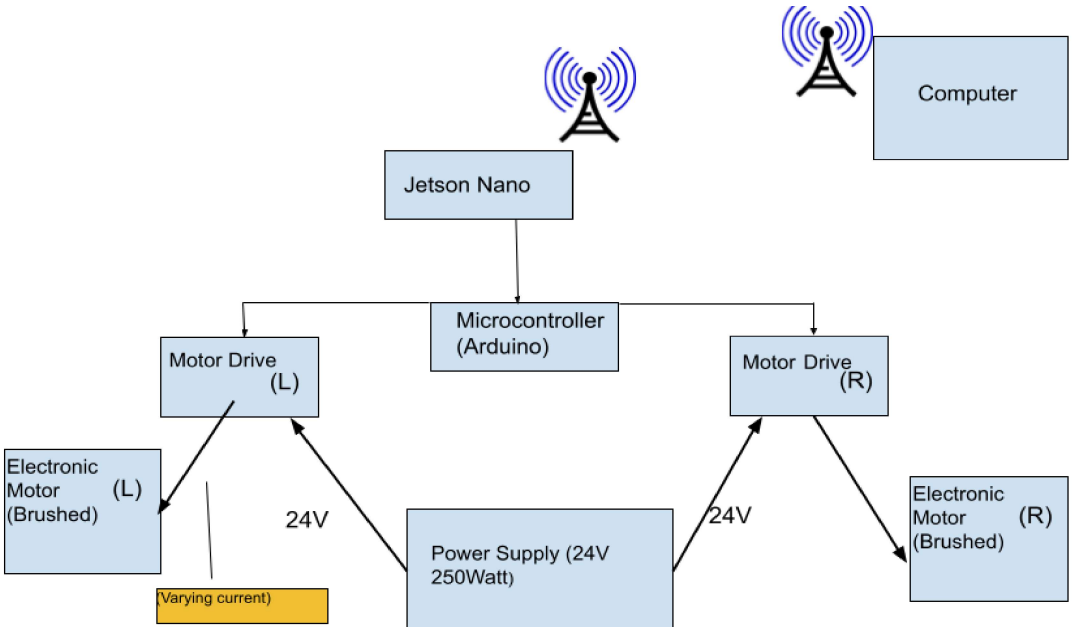
Joystick*:*

1. Plug in the joystick's 3 connections to the corresponding component. (Power, Left Motor, and Right Motor)
2. Establish a navigation course to test controllability.



Arduino:

1. Plug in motor controller's 4 connections(Power, Left Motor, and Right Motor, Arduino).
2. Upload corresponding file for tested load, starting with no load.
3. SSH into the Jetson Nano
4. Activate the virtual environment containing the required PySerial library
5. Navigate to the directory containing the control_arduino.py script
6. Run the control_arduino.py script using the command: *python3 control_arduino.py*
7. Apply load to chair, starting with no load (chair propped on)
8. Repeat steps 2-7 for each tested load: Chair propped up, weight of chair, and weight of human on chair



Object Recognition:

1. Make sure you have a version of Python 3 (preferably python 3.6 to avoid dependency issues) installed on your Jetson Nano
2. Build the jetson-inference module from source by cloning the following repository and following the instructions listed on it: https://github.com/dusty-nv/jetson-inference
3. Make sure your Raspberry Pi SC0024 IMX219 NoIR Camera is connected to your NVIDIA Jetson Nano via a ribbon cable
4. Find the index of your camera by running the following command on your Jetson Nano: *v4l2-ctl --list-devices* . In our case, the index of our camera is "0"
5. Download the pretrained InceptionV2 model
6. Run the script with the command: *python3 test_inference.py*

IV. Testing Procedure

Motor Controller:

Joystick:

1. User gets on wheelchair
2. Power on using joystick's button interface
3. Increase speed to level 2 using joystick's button interface
4. User navigates to the course using joystick control.
5. Power off
6. Switch user, repeat steps 1-5 for each user

Arduino/Jetson Nano:

1. Enable motors
2. Display smooth acceleration and deceleration in each direction using serial monitor commands
3. **(For No Load only)** Measure the wheel full rotation time at the high and low end of input commands (4-16rpm)
4. Repeat for each load

Object Recognition:

1. Power on the Jetson Nano
2. Navigate to the directory that contains the test_inference.*py* Python script and the pre-trained Inception-V2 model
3. Run the script: *python3 test_inference.py*
4. Wait for the screen to show your real-time camera recordings
5. Place objects to detect in front of the webcam
6. Check whether there is a bounding box around the object to determine whether the model recognizes an object is present in the frame
7. If a bounding box exists, check the label of the bounding box to determine whether the object has been classified correctly

Object	*Detected Object in Frame?*	*Classified Object Correctly?*

V. Measurable Criteria

Motor Controller:

1. Joystick control system should be usable by all users under 150lbs
2. Arduino control system should be able to regulate speeds of wheels as low as 4 rpm and as high as 16 rpm
3. Arduino control system should be able to move forward and backward with up to 120 lbs load applied

Object Detection:

1. The model should detect that there is an object in the frame at least 80% of the time
2. The model should correctly classify the object at least 60% of the time
3. The model should detect the object's presence within a second