**Moving Platform Documentation**

1. Introduction

The Moving Platform was constructed to simulate movement of pedestrians as well as cyclists in future road releases for nuTonomy’s autonomous vehicles. By being able to replace live pedestrians and cyclists, tests conducted during the road releases will be made safer.

This documentation was written to provide an understanding of the software and hardware implemented in the moving platform, as well as highlight its current limitations and the potential improvements which can be made to it in the future.

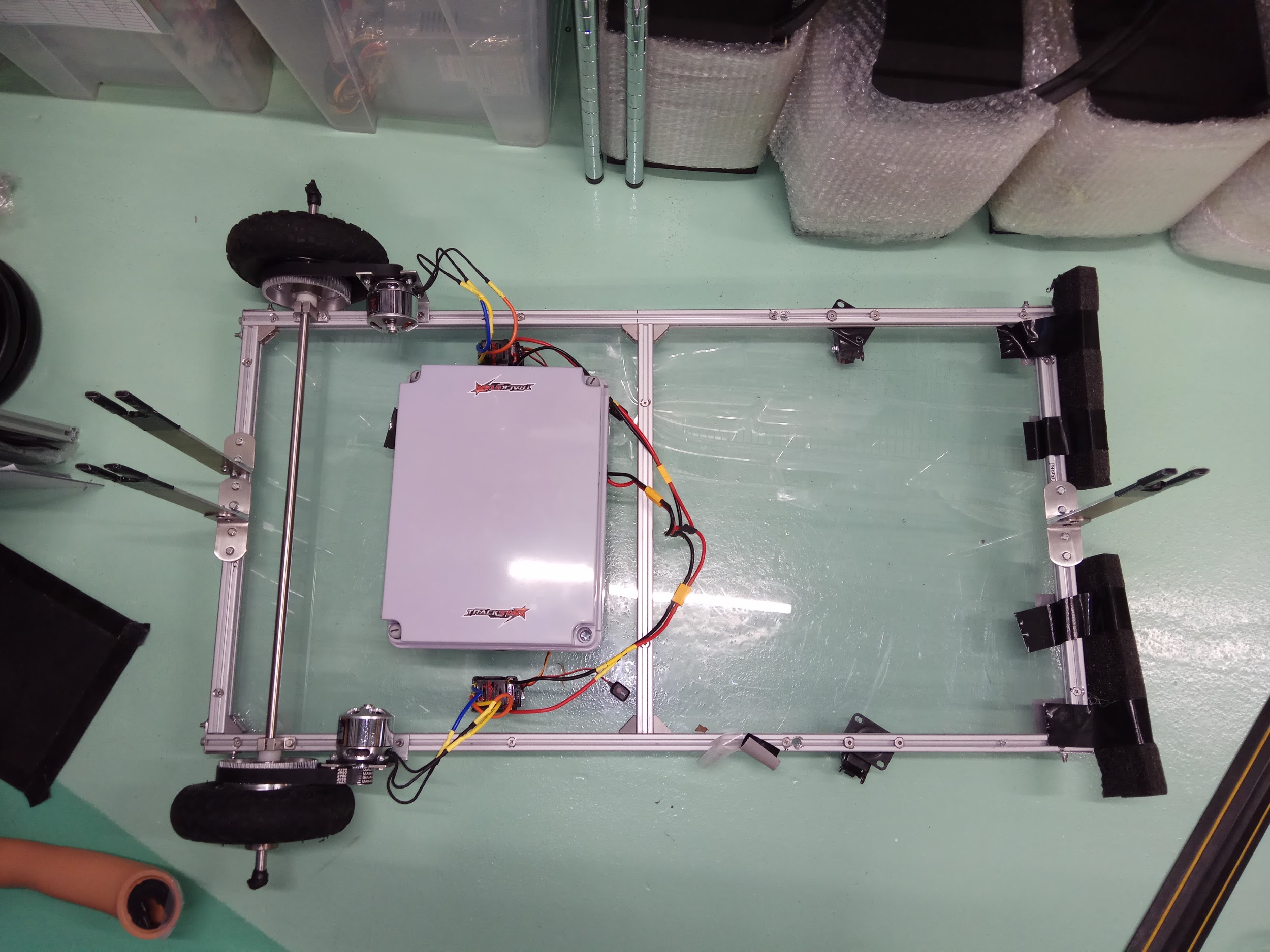
Where to find the code: <https://github.com/plumwithtangerine/moving-platform>

The main code used to control the cart is in **simple\_steering.ino.**

2. Moving Platform Overview

Ye old fashioned moving platform. Enough space to mount Bob or Matthias’ bicycle(it has a wheelbase of roughly 1m-1.02m). It uses caster wheels in the front for more responsive turning. I may have calibrated the current ESCs with 50% throttle trim setting on the radio controller by accident. It may/may not affect the power output by the motor so the ESC can be recalibrated if more torque is required to move Bob/the bicycle at a faster speed.

**Turning on/off the ESCs:** To turn them on, press both ESC power buttons **once.** To turn them off, press both power buttons and hold until the light on the ESC changes. Make sure the circuit breaker and the circuit containing the LiPo battery and the ESCs is closed.



*Figure 1: Photo of (near) completed moving platform.*

3. Setting Up/Configuring the Moving Platform

This section will explain how to configure the moving platform in the event where parts require replacing or upgrading.

3.1 Calibrating Your New Radio Controller

Somehow you mess up and have to buy a new radio controller. No worries! Just plug the radio receiver(make sure the receiver has at least 2 usable channels) into the Arduino board and run the **RC\_Controller\_Calibration.ino** code. Use it to note down the min and max values of the controller for each channel when the signals get mapped into ESC values.

3.2 Calibrating Your New Electronic Speed Controller(ESC)

Somehow you caused your brand new ESC to explode as well(**\*cough\*** Yilin **\*cough\***). Don’t try to repair it, but buy a new one. Firstly, get ESCs that are able to rotate the motors both forwards and backwards. Being able to reverse your moving platform will be a great help when it runs into walls or curbs. Just plug your ESC with the battery into the relevant pin on the Arduino board and run the **Calibrate\_2way\_Motor.ino** code, following the instructions needed to calibrate the ESC.

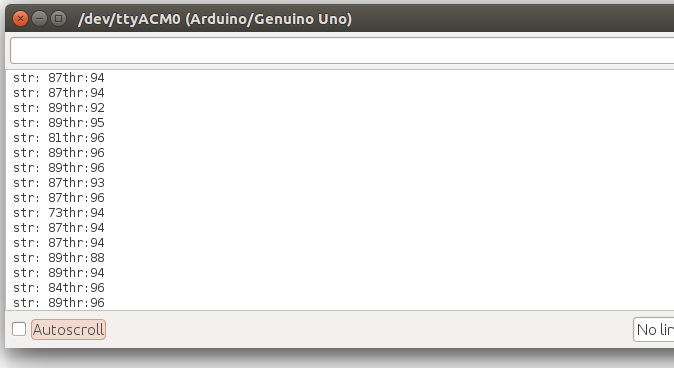
NOTE: don’t plug anything into the positive terminal of the ESC as it can fry your computer’s USB port.

3.3 Type of Connectors to Use

XT60 connectors or 4mm male/female connectors. Can be bought from HobbyKing.

4. Current Limitations and Future Work

At the moment, the platform does not have a proper mount for Bob and relies on the mannequin’s own base stand for support. The stand will be secured to the platform using strong adhesive tape.

Also, the radio signals being transmitted from the radio control to the receiver experience a fair amount of noise, which means that the signals received by the radio receiver are not consistent. Using a multitude of if else statements can rectify this problem fairly well. However, the controls become less sensitive and the remote controlled platform may not move accordingly to the user’s desires.

*Figure #: Inconsistent radio signals displayed on Arduino IDE when the controls were in the ‘neutral position’. ‘str’ stands for ‘steering’ and ‘thr’ stands for ‘throttle’.*

Another limitation are the caster wheels on the moving platform. The advantage of caster wheels is that it makes turning easier and simpler for the platform. However, it is also a disadvantage as it restricts the platform from turning and moving at higher speeds due to ‘caster flutter’. It is not significant problem right now, but when road releases require bicycle simulations to be tested at higher speeds, this problem may crop up. Another problem with the caster wheels occurs when the moving platform accelerates forward after it has reversed. This causes the caster wheels to rotate 180 degrees to move forward probably due to the weight placement of the wheels. If the wheels do not rotate fully, this could cause the platform to move undesirably. ~~(why did i write so much help me)~~

To overcome these limitations, possible future work could include:

* Signal averaging through the Arduino IDE to increase remote control responsiveness. This has been tried before but the translated signals became worse than the original. I did not have sufficient time to check my code so I may have made mistakes while calculating the signals.
* Improving the code which controls movement in Arduino IDE. My coding skills are awful so I consider my implemented code to be pretty primitive at the moment. If you can find a better algorithm to translate the radio signals into smooth and buttery motor movement that would be great.
* Creating a new Bob. Current Bob is very tall and heavy so he may not be stable enough for the platform if road releases include running pedestrian tests.
* Replacing current ESCs with VESCs. Expensive(used for electric skateboards!) but they are much better in terms of performance compared to regular ESCs.
* Waterproof the ESCs and motor somehow.