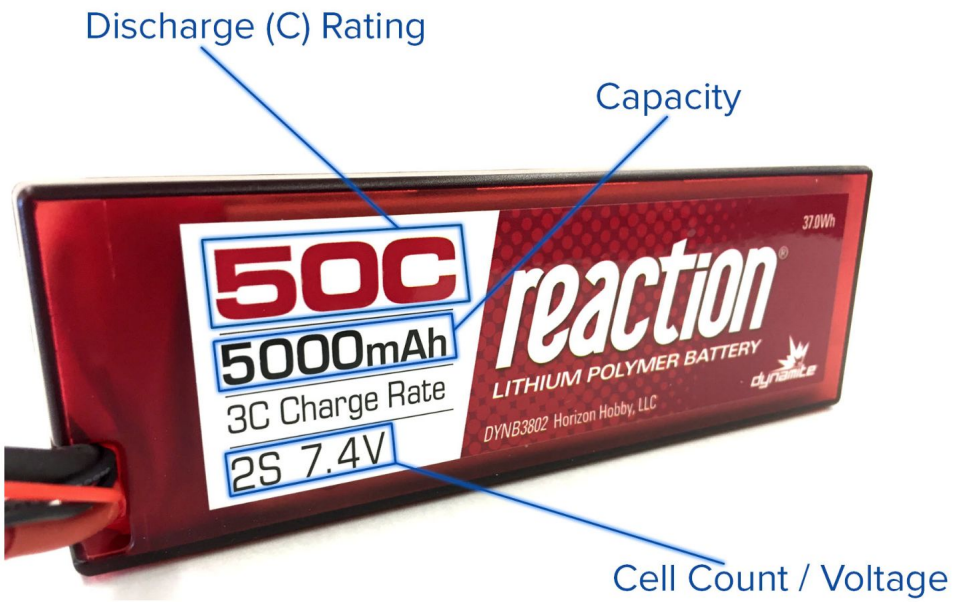
Setting up your vehicles

Penn Engineering

Lipo Batteries



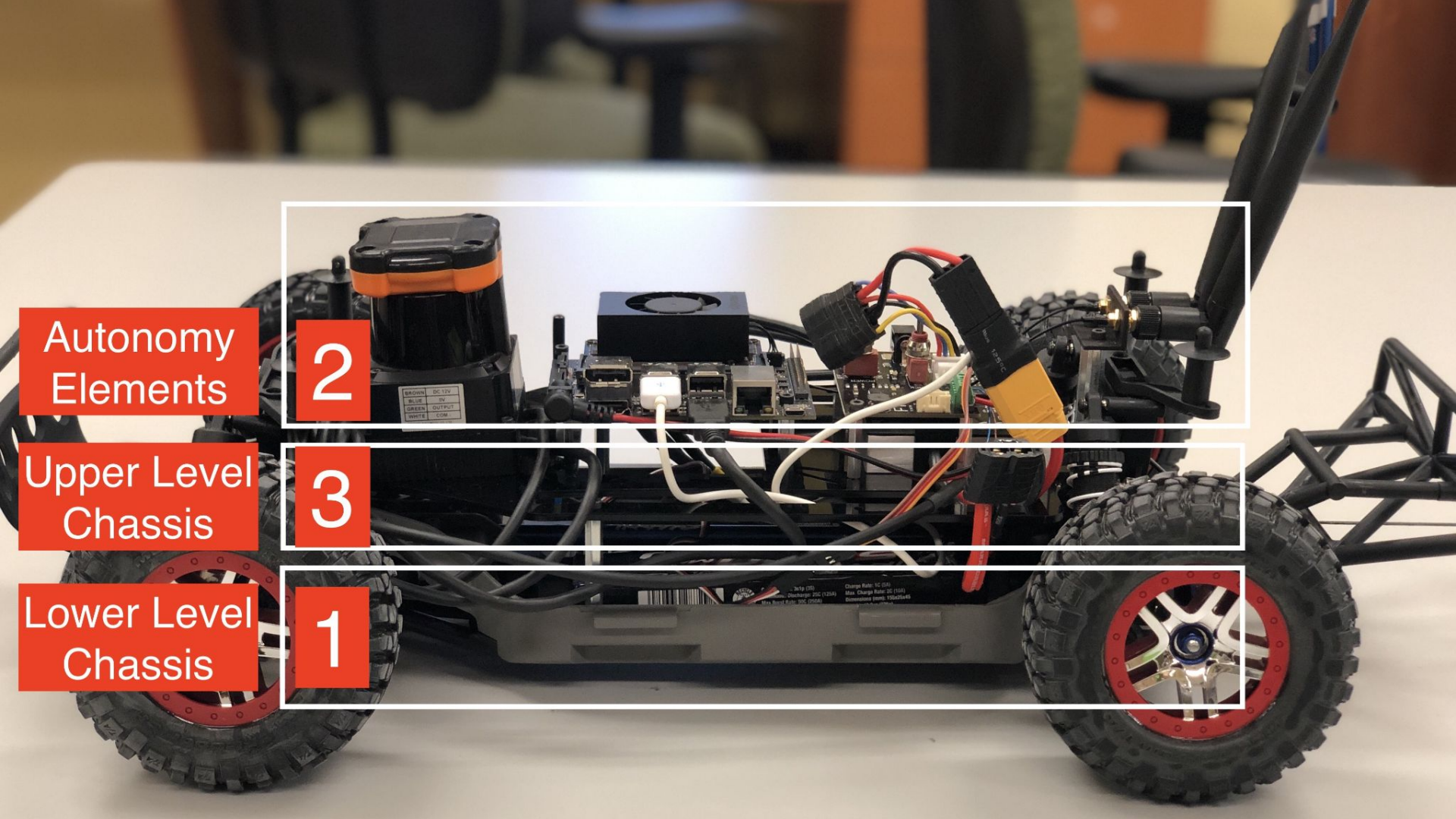
- **Capacity**: how much power a battery holds.
- **C-rating**: how fast the battery can safely discharge. Measures the burst current the lipo is capable of: C*Capacity. For example our battery can have a burst current of 50C*5Ah=250 A.
- **Cell count**: voltage per cell is 3.0-4.2 V (empty-full). Storage/nominal charge per cell is 3.7V.

Lipo Charger

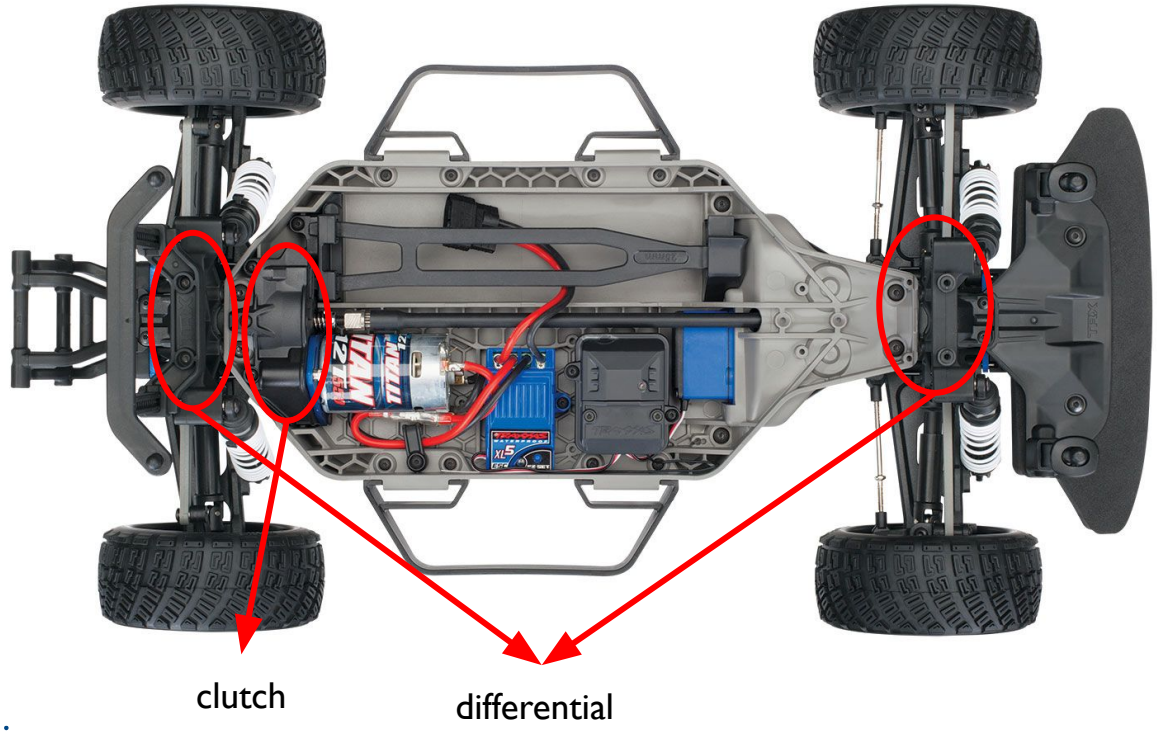
- **Battery Type**: Should automatically detect, but always use lipo.
- **Lipo Charge mode**: **Never** use fast mode, it'll cause unbalanced cell voltages. Always use **balance mode** when you're charging to full, and use storage mode when you're storing the battery.

Battery safety

1. **DO NOT over discharge!** Use the low voltage buzzer **at all times** when you're using the battery.
2. Use proper storage. Store batteries in the safety cabinet in Detkin when you're not using. Discharge/Charge to storage voltage if you're leaving the battery for more than a week.
3. Avoid hard impact to the battery, we use soft cell lipos.


Autonomy Elements
2
Upper Level Chassis
3
Lower Level Chassis
1

Components of the car: Chassis

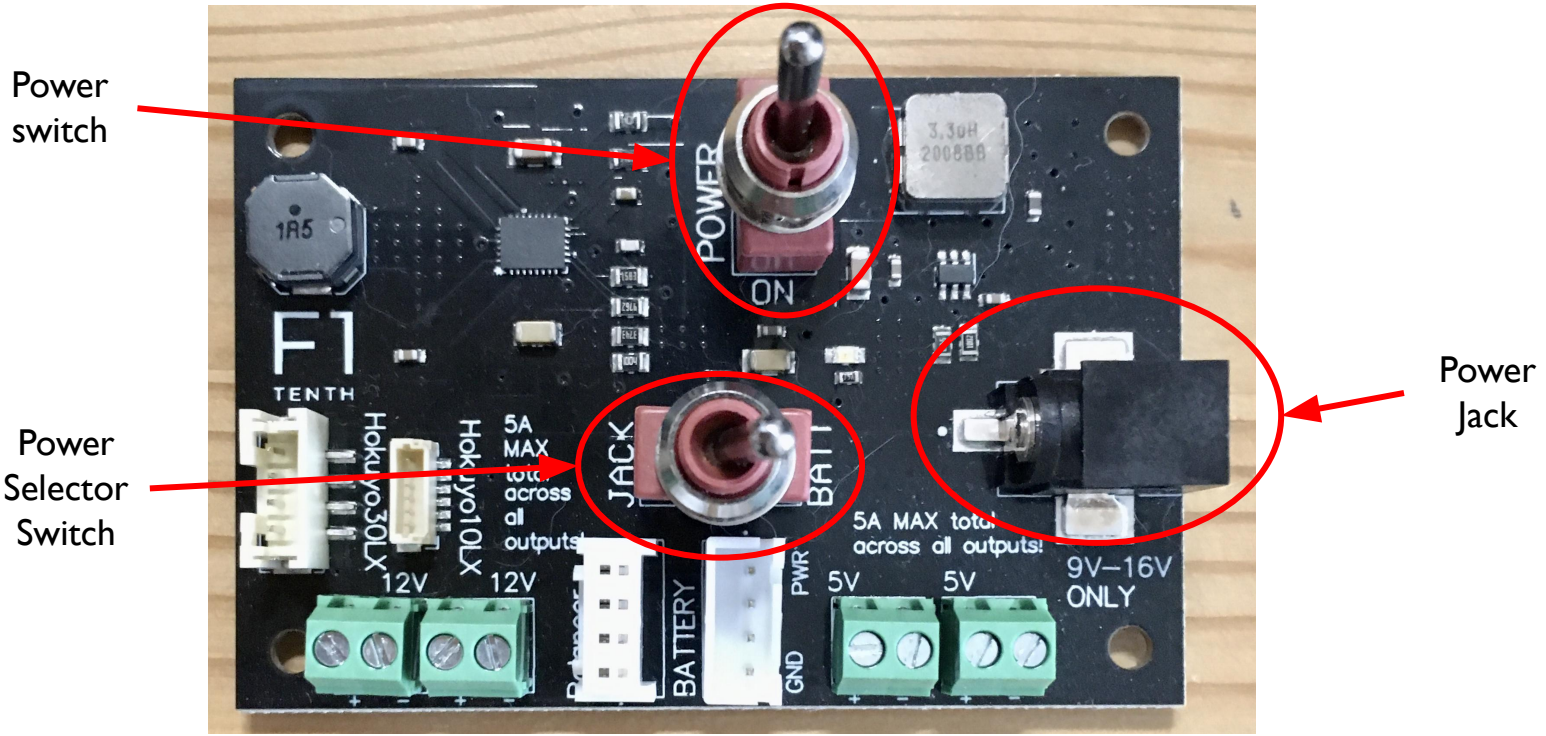


Components of the car: Jetson Xavier NX

Components of the car: VESC

Penn Engineering

Components of the car: Power Board



Penn Engineering

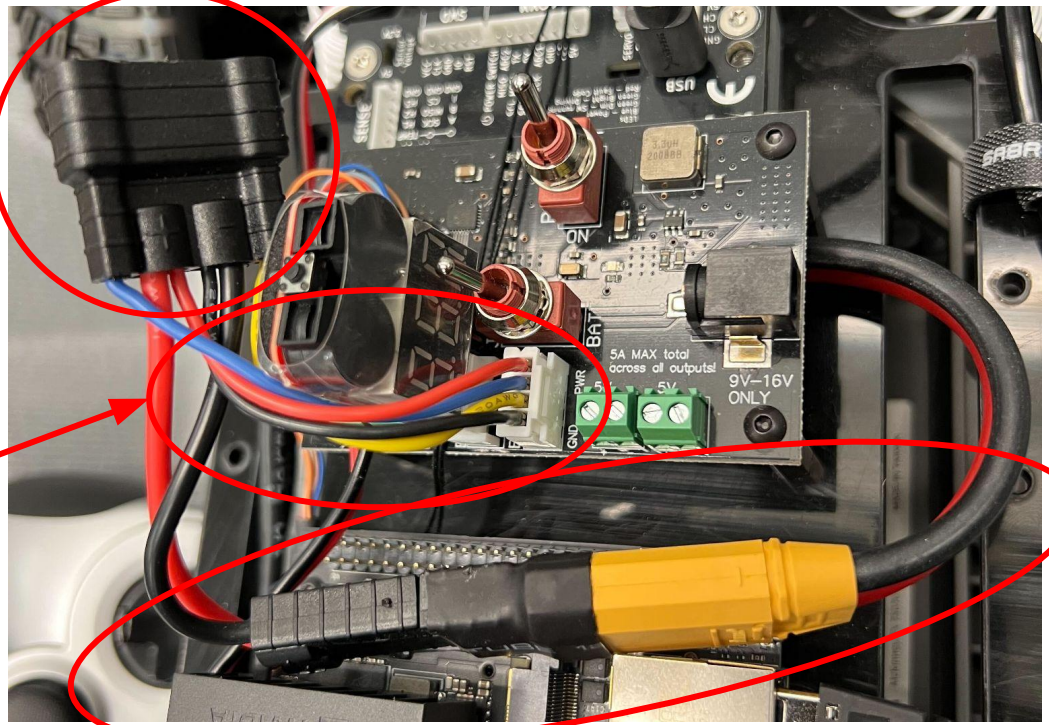
Components of the car: Power Board

Battery Balancer Port

Battery Connector

Components of the car: Battery Connection

Connects to the battery. Use **COLORS** to identify polarity.



First split powers the power board

Second split powers the VESC

Powering the car on

The power board can use two power sources:

1. Using the battery. Plug the battery into the connector (with correct polarity). Make sure the power selector switch is at the battery position, then switch the power on.
2. Using the provided power supply. Plug the power supply into the wall and the jack on the power board. Make sure the power selector switch is at the power supply position, then switch the power on. The power board ONLY SUPPORTS 9-16V. Do NOT use any other power supply except for the one provided.
3. Turn off everything before you switch the power source.

The VESC will not be powered on if only using the provided power supply. You can still power the vesc alone with the battery by unplugging the connector from the battery to the power board.

Connecting to the car

1. Connect the mouse, keyboard, and monitor to the Jetson
2. Power on the system

User: **nvidia**, pwd: **nvidia**

3. Connect to the provided router in the network settings

SSID: **NETGEAR57** pwd: **grandowl947**

Install ROS 2 Foxy

1. Follow the instructions on https://docs.ros.org/en/foxy/Installation/Ubuntu-Install-Debians.html to install foxy-desktop.

Installing rosdep

Follow https://docs.ros.org/en/foxy/Installation/Alternatives/Ubuntu-Install-Binary.html#installing-and-initializing-rosdep

Checking udev setup

1. When the vesc is powered on (with the battery), run:
   `ls /dev/sensors/`
2. You should see `/dev/sensors/vesc`
3. And if you have a 30LX (USB LiDAR), you should see `/dev/sensors/hokuyo` also.
4. If you can't see these, follow https://f1tenth.readthedocs.io/en/foxy_test/getting_started/firmware/drive_workspace.html#udev-rules-setup to setup udev rules for the vesc and/or the LiDAR.

Set up the F1TENTH stack

1. Create a ROS workspace:

   ```
   cd $HOME && mkdir -p f1tenth_ws/src
   ```

2. Clone the F1TENTH stack repo:

   ```
   cd f1tenth_ws/src
   git clone https://github.com/f1tenth/f1tenth_system.git
   ```

3. Update the git submodules

   ```
   cd f1tenth_system
   git submodule update --init --force --remote
   ```

Set up the F1TENTH stack

4. Install dependencies with rosdep

```
cd $HOME/f1tenth_ws
rosdep update --rosdistro=foxy
rosdep install --from-paths src -i -y
```

5. Build the workspace

```
colcon build
```

LiDAR configuration

- If you have a 30LX (USB LiDAR), comment out the `ip_address` line and uncomment the `serial_port` line.
- Parameter file is located at:
  - `$HOME/f1tenth_ws/src/f1tenth_system/f1tenth_stack/config/sensors.yaml`

Choosing A ROS_DOMAIN_ID

To prevent cross talk from other car's ROS topics, set a unique ROS_DOMAIN_ID for your car before you run anything. https://docs.ros.org/en/foxy/Concepts/About-Domain-ID.html explains what the domain id is.

To set it, use:

```
export ROS_DOMAIN_ID=<your_team_number>
```

How to start teleop and the LiDAR

1. Sourcing overlays and underlays

   ```
   source /opt/ros/foxy/setup.bash
   cd $HOME/f1tenth_ws
   source install/setup.bash
   ```

2. Launch the bringup

   ```
   ros2 launch f1tenth_stack bringup_launch.py
   ```

You should then be able to launch RVIZ with `rviz2` in the command line, add a scan topic, and change the frame to `/laser`.

Teleop, Deadman's switches with Logitech Joystick

Use 'D' mode on the logitech joystick and the mode light should be off. You might have to remap the axis/buttons depending on your joystick.

Connecting the joystick

1. Open the bluetooth settings, filter by joystick devices
2. Hold the **share** and **PS** button on the joystick at the same time until the light bar flashes
3. Pair the device (look around before you pair the joystick other teams might also be doing it at the same time)

Teleop, Deadman's switches with PS4 Joystick

Hold to use message on /drive topic.

Hold to use joystick teleop.

Axis for throttle

Axis for steering

Remapping Joystick Controls

1. Echo the /joy topic to see what the mapping from joystick to a joy messages axis is.
2. Identify the indices of buttons and axis that you want to use.
3. Modify `f1tenth_system/f1tenth_stack/config/joy_teleop.yaml` to change the mapping.

Remote Desktop

1. Install NoMachine on the Jetson Xavier NX. Follow https://knowledgebase.nomachine.com/AR02R01074 . You don’t have to install Xfce4

Remote Desktop

2. Install NoMachine on your laptop. Follow https://www.nomachine.com/download for your specific OS.

Remote Desktop

3. Connect both your laptop and the car to the provided router. Find the ip of your car with `ifconfig`.

Remote Desktop

4. On your laptop:

Give your connection a name

Put the car's ip here

Penn Engineering

Parameter Tuning

- The guide can be found at
  - https://f1tenth.readthedocs.io/en/foxy_test/getting_started/driving/drive_calib_odom.html

Parameter Tuning

- You'll have to find the right speed to erpm gain, and the right steering angle to servo offset so that the odometry is somewhat accurate.
- Parameter file is located at:
  - `$HOME/f1tenth_ws/src/f1tenth_system/f1tenth_stack/config/vesc.yaml`
- The three parameters you'll need to tune are:
  - `speed_to_erpm_gain`
  - `steering_angle_to_servo_offset`
  - `steering_angle_to_servo_gain`

vesc.yaml

```
/**:
  # general parameters shared by all vesc-related nodes
  ros__parameters:
    # erpm (electrical rpm) = speed_to_erpm_gain * speed (meters / second) +
speed_to_erpm_offset
    speed_to_erpm_gain: 4614.0
    speed_to_erpm_offset: 0.0


    # servo value (0 to 1) =  steering_angle_to_servo_gain * steering angle
(radians) + steering_angle_to_servo_offset
    steering_angle_to_servo_gain: -1.2135
    steering_angle_to_servo_offset: 0.5304
```

Tuning the steering offset

Tuning loop:

1. Start teleop
2. Drive the car in a straight line a few times. It'll never be perfectly straight but if it drives straight most of time it'll be fine.
3. Adjust `steering_angle_to_servo_offset` in `vesc.yaml`, call `colcon build`, and go back to step 1.

Tuning the steering gain

This gain takes a steering angle and converts it into a servo position. We use the kinematic model to find this gain. The turning radius is calculated as `R=L/(2sin(beta))` where L is the wheelbase of the vehicle (around 33cm), and beta is `arctan(0.5tan(delta))` where delta is the maximum steering angle of the car (around 0.36 radians). The turning radius comes out to be around 0.947 meters. Thus, when turning a half circle, the diameter of the half circle should be 1.784 meters.

Tuning the steering gain

Before tuning:

Set up the tape measure again like you did when tuning the odometry.

Tuning loop:

1. Place the car at the 0 of the tape measure where this time the car's rear axle lines up with the tape measure, and the x-axis (roll axis) of the car coincides with 0.
2. Start teleop.
3. Set steering angle to the maximum to one side depending on your setup. For example, if the measuring tape is to the left of the car, turn left all the way.
4. Hold the steering and drive the car slowly and steadily till it runs over the tape measure and the rear axle re-aligns with the tape measure again.
5. Take a measurement with the car's x-axis. The goal is 1.784 meters. If it overshoots, increase the gain slightly (0.1 at a time). If it undershoots, decrease the gain.
6. Don't forget to call `colcon build` after you adjust your gain values.

Tuning Odometry

Before tuning:

Extend your tape measure to around 4-5 meters on the floor, make sure it won't shift around during tuning.

Tuning loop:

1. Place the car at the 0 of the tape measure. Make sure the rear axle of the car is lined up with 0.
2. Start teleop
3. Open another terminal, use the following command to echo the odometry messages. We want to pay attention to pose/pose/position/x:

```
ros2 topic echo --no-arr /odom
```

Tuning Odometry

4. This should be 0.0 at the start when the car hasn't moved yet. If it's not 0.0, kill teleop and restart.

5. Drive the car forward for around 4 meters. Make sure you're only driving forward without steering.

6. Record the distance traveled. Take the measurement from the rear axle.

7. Compare the measurement to the output of the value you're echoing. If the distance reported by echo is larger than the actual measurement, decrease the gain. Otherwise increase the gain. The gain should be on the order of thousands.

8. Stop teleop, go back to step 1 if the values are not close enough (within 2-3 cm). Don't forget to call `colcon build` after you adjust the value.