

2019 FLYSET FTC Workshop

# Odometry Wheels

(8/24/2019)





**Presenter**



# Anthony - FTC 8565

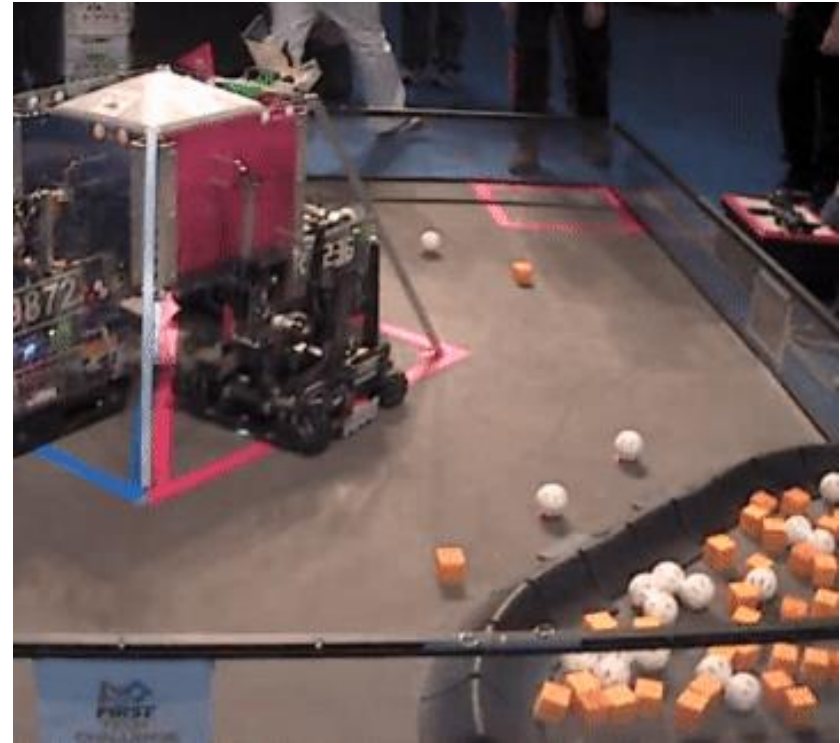
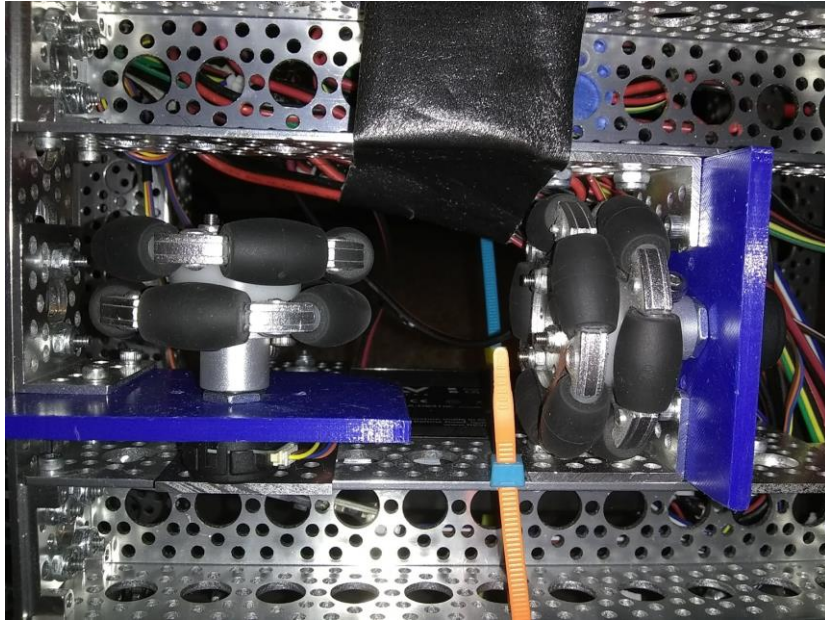
- 9th year in FIRST
  - 3 years in Jr. FLL
  - 3 years in FLL
  - 2 years in FTC
- Software Lead
- Enjoys Skiing and Video Games





# Project Background

# Background



(Courtesy of FTC Team 8393, Detroit 2019 Ochoa F2)



technibots **8565**



# Project Design

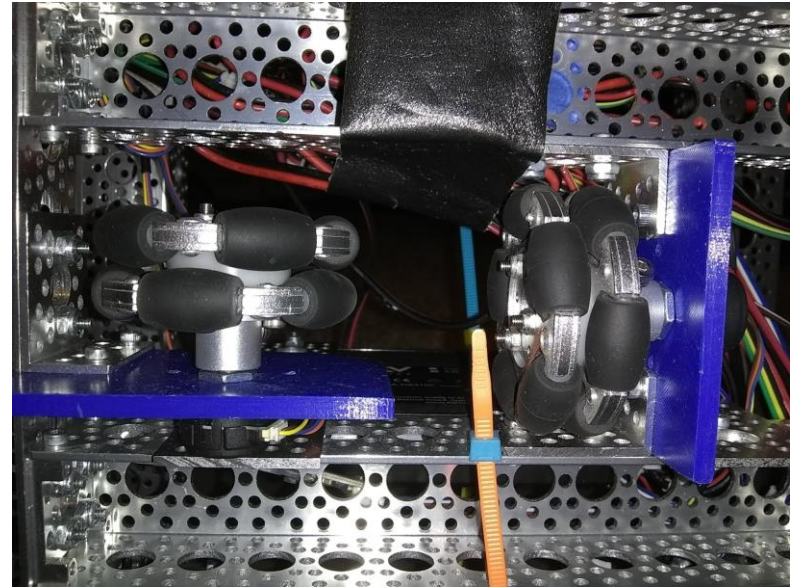
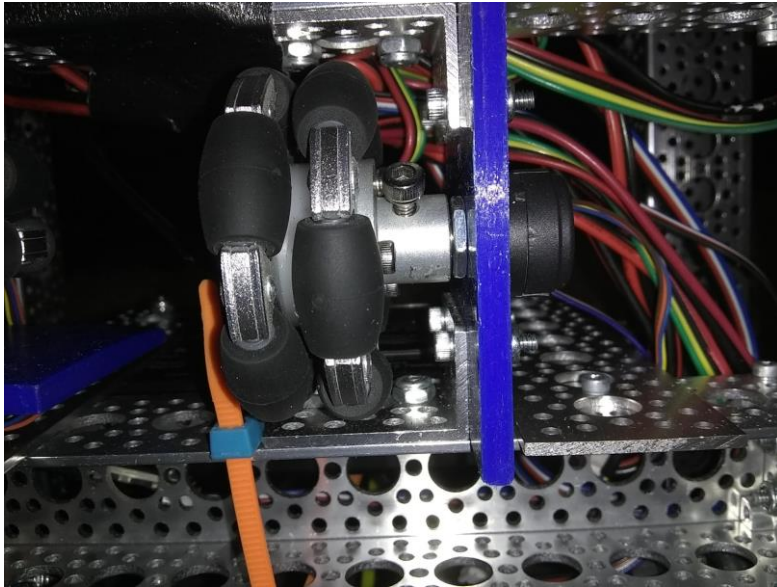


## Goals

- Learn how to construct an odometry wheel setup
- Compare odometry wheels to traditional motor mounted encoders



# Robot Setup







# Setup

- 15 trial runs for each of the encoders for strafing - moving 112 cm
- 13v battery
- Shared PID controller
- No load or 15 pound load
- 100% speed or 50% speed
- Counts\_per\_inch found experimentally





# Motor Encoder code

The code we used for the motor encoder:

```
public void mecanumSideDrive(double distance, double power) {
    distance *= COUNTS_PER_INCH;
    robot.defaultEncoderMode();

    int startDistance = robot.getEncoderPosition();
    float startHeading = robot.getHeading();
    timeoutTimer.reset();
    timeoutTimer.reset();

    while (Math.abs(robot.getEncoderPosition() - startDistance) < Math.abs(distance)) {
        double error_degrees = robot.getHeading() - startHeading;
        double correction = robot.gyroDriveController.findCorrection(error_degrees);
        robot.leftBackMotor.setPower(Range.clip(power + correction, -1, 1));
        robot.leftFrontMotor.setPower(Range.clip(power - correction, -1, 1));
        robot.rightBackMotor.setPower(Range.clip(-power - correction, -1, 1));
        robot.rightFrontMotor.setPower(Range.clip(-power + correction, -1, 1));
        dispEncoders();
    }
    robot.stopRobot();
    robot.defaultEncoderMode();
}
```



# Dead Wheel Encoder code

The code we used for the dead wheel encoder:

```
public void mecanumOdoSideDrive(double distance, double power) {
    distance *= Constants.S4T_COUNTS_PER_INCH;
    robot.defaultEncoderMode();

    int starty = robot.leftFrontMotor.getCurrentPosition();
    float startHeading = robot.getHeading();
    timeoutTimer.reset();
    timeoutTimer.reset();

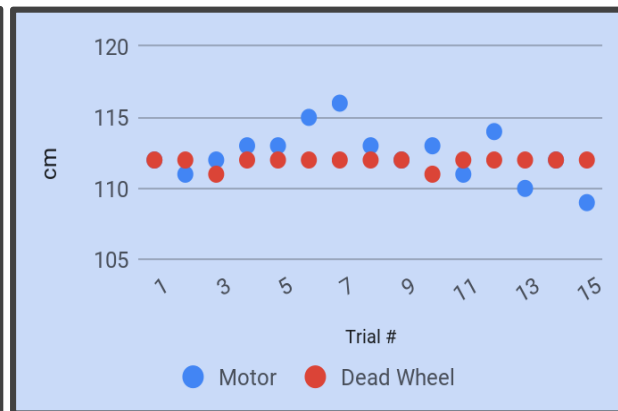
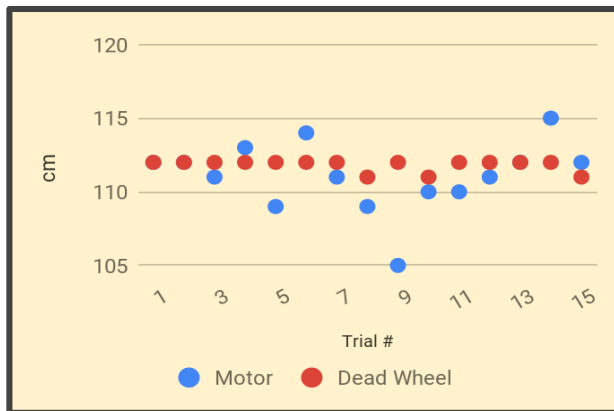
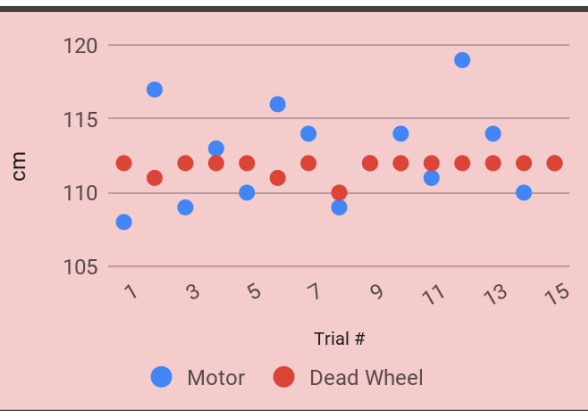
    while (Math.abs(robot.leftFrontMotor.getCurrentPosition()-starty) < Math.abs(distance)) {
        double error_degrees = robot.getHeading() - startHeading;
        double correction = robot.gyroDriveController.findCorrection(error_degrees);
        robot.leftBackMotor.setPower(Range.clip(power + correction, -1, 1));
        robot.leftFrontMotor.setPower(Range.clip(power - correction, -1, 1));
        robot.rightBackMotor.setPower(Range.clip(-power - correction, -1, 1));
        robot.rightFrontMotor.setPower(Range.clip(-power + correction, -1, 1));
        dispEncoders();
    }
    robot.stopRobot();
    robot.defaultEncoderMode();
}
```



# Project Results

# Results

	100% Speed	100% Speed	50% Speed	50% Speed
	Motor	Dead Wheel	Motor	Dead Wheel
0 pound load	<b>3.1818</b>	<b>0.5936</b>	<b>1.8047</b>	<b>0.3399</b>
15 pound load	<b>1.9952</b>	<b>0.3519</b>		





# Conclusions



## Conclusions/Observations

- Strafing with mecanum wheels has a significant amount of slippage. While strafing, **odometry wheels had 5.67 times smaller error than with traditional encoders.** Using motor encoders would have average error of 1.798 cm for every meter of strafing, and only 0.314 cm using odometry wheels.





## Lessons Learnt

- Wheel height sensitivity- Sometimes the wheel can't touch the ground. We are going to spring load.
- Set screw hub coming loose
- Motor encoder port



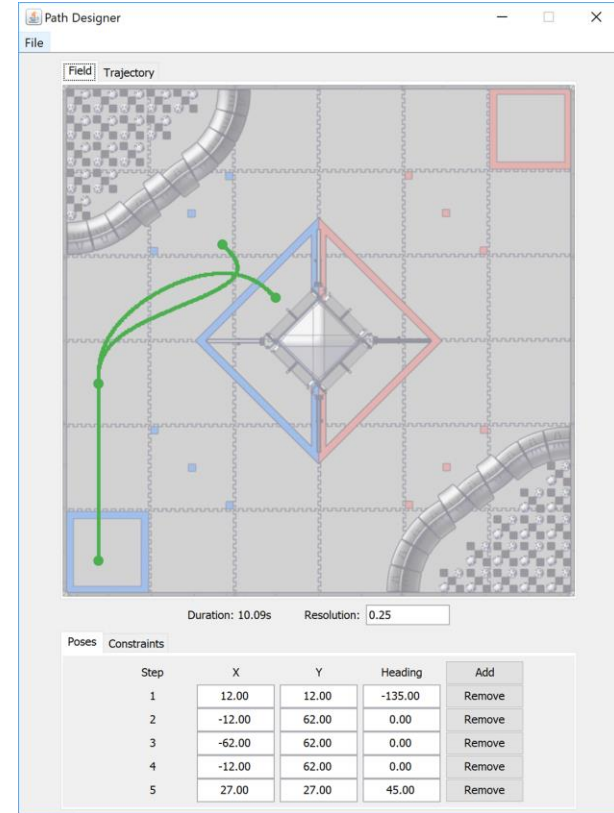
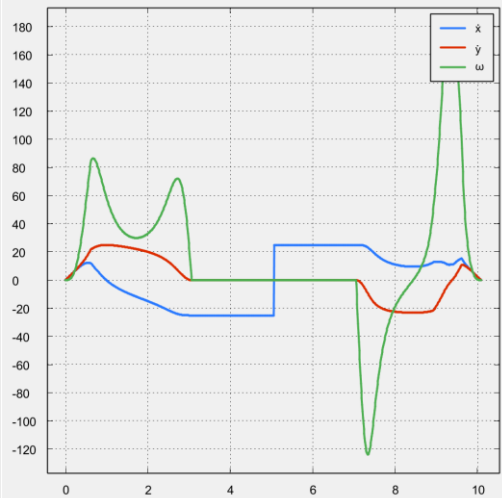
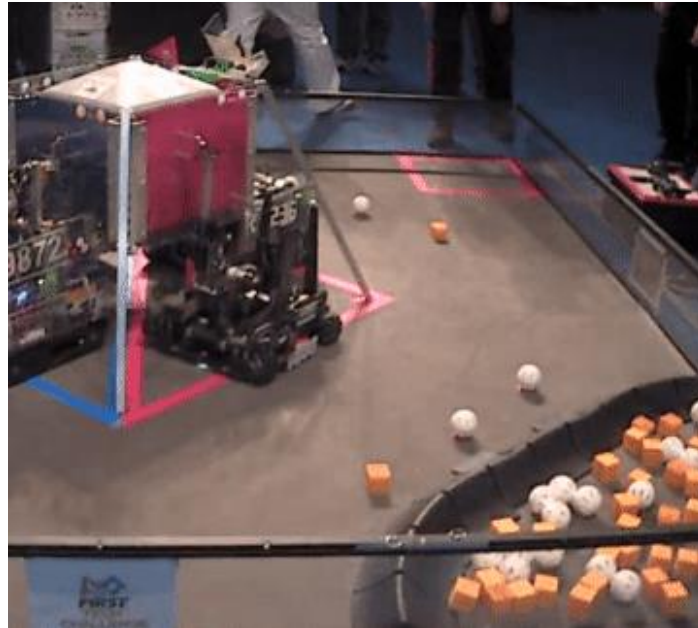
## Sensor



- Encoder damage- we had several encoders break from pressing against the ground too hard. This problem stopped once we switched from s4ts to e4ts though.
- S4t is generally better as it already comes with a shaft, so it is easier to put on and so it is harder for dust to get in and ruin the encoders.
- They are the basically the same thing, the only difference is the shaft

# Next steps

- Roadrunner
- Spring loading





**Questions?**