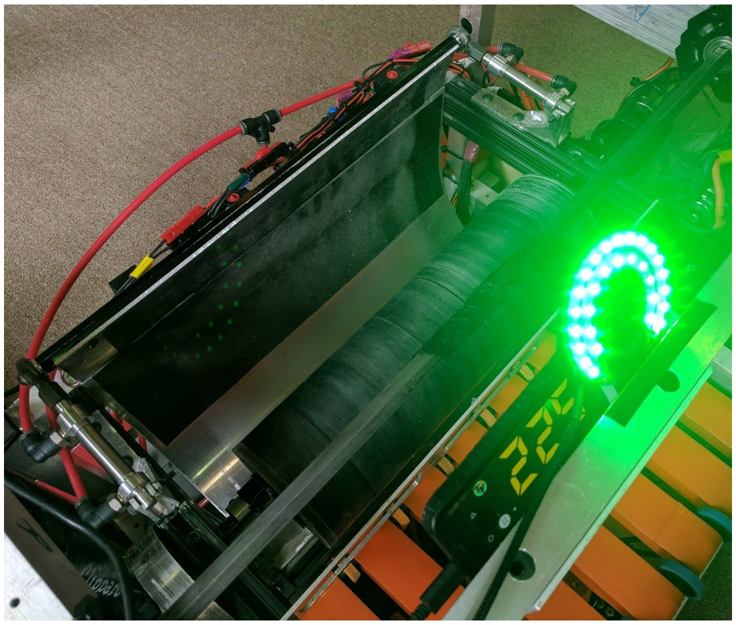
TECHFIRE 225

2017 CONTROL SYSTEMS

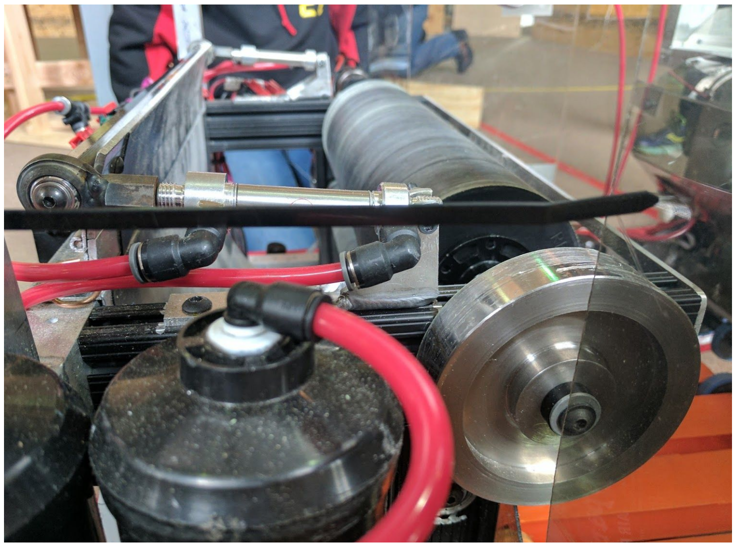
by Adam Wolnikowski - Team Captain, Driver, and Lead Programmer

Shooter Control

Although scoring fuel is somewhat of a secondary task in FIRST STEAMworks, our shooter is the most complex control system on the robot. After the announcement of this year's game, the team ultimately decided to pursue a "triple-wide" shooter for the sake of high-speed, high-volume shooting. Ultimately, both mechanical and programming design innovations came together to produce a shooter capable of shooting at about **80% accuracy** and scoring **40 fuel** in about **6 seconds**.

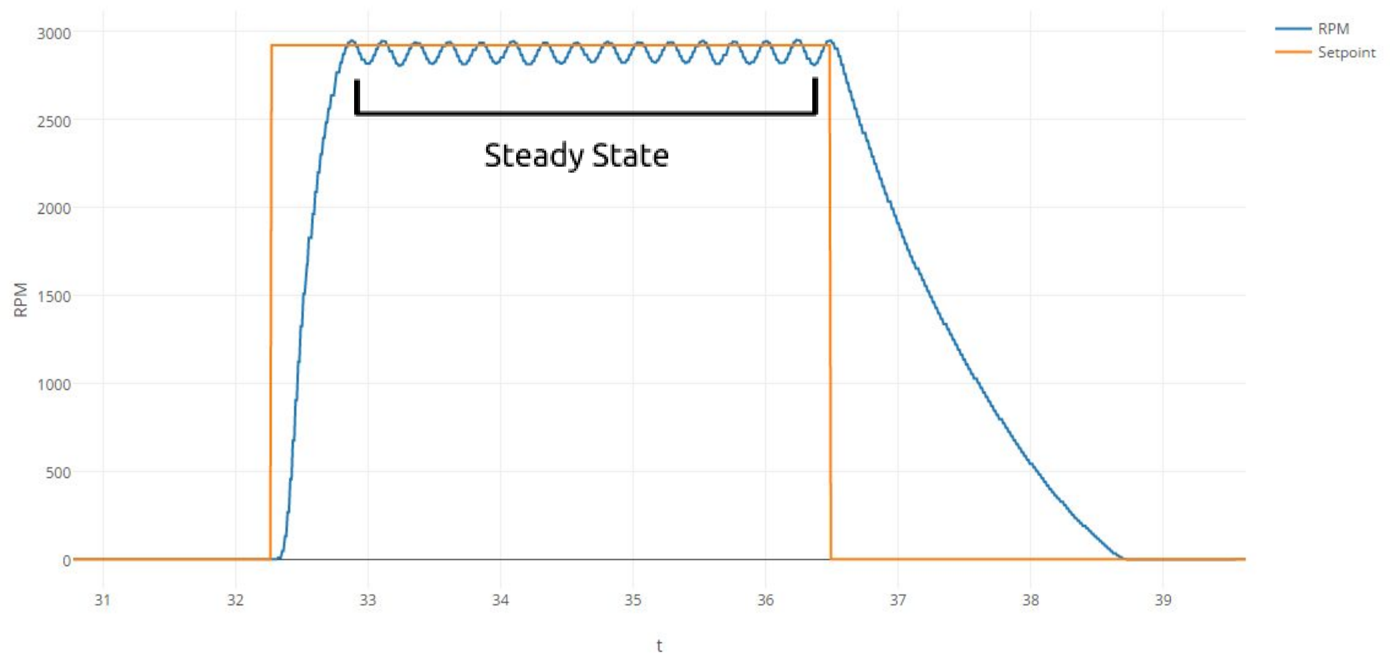


From a programming perspective, the shooter solves two problems: maintaining a constant RPM even while varying amounts of fuel are being fed through it, and automatically both aligning the robot with goal and selecting the correct RPM for the robot's distance from the goal.

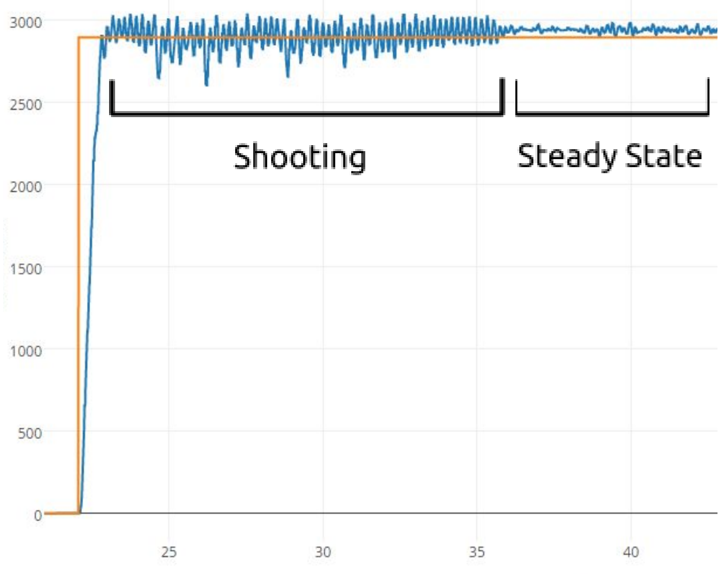


To maintain a given RPM, a Proportional, Integral, Derivative, and Feed-forward (PIDF) control loop is implemented. Originally, we attempted to implement this on a shooter without an auxiliary flywheel (i.e. just rubber wheels) and found the oscillation seen in the first graph. As such, we added a flywheel and were able to achieve the better results seen in the second graph. Finally, we further tuned the control loop and added a flywheel with greater rotational inertia in order to achieve the final results seen in the third graph.

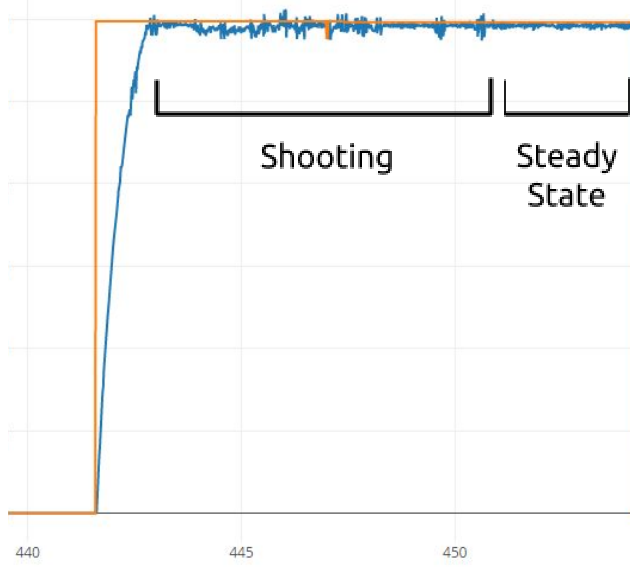
1. Initial attempt without flywheel



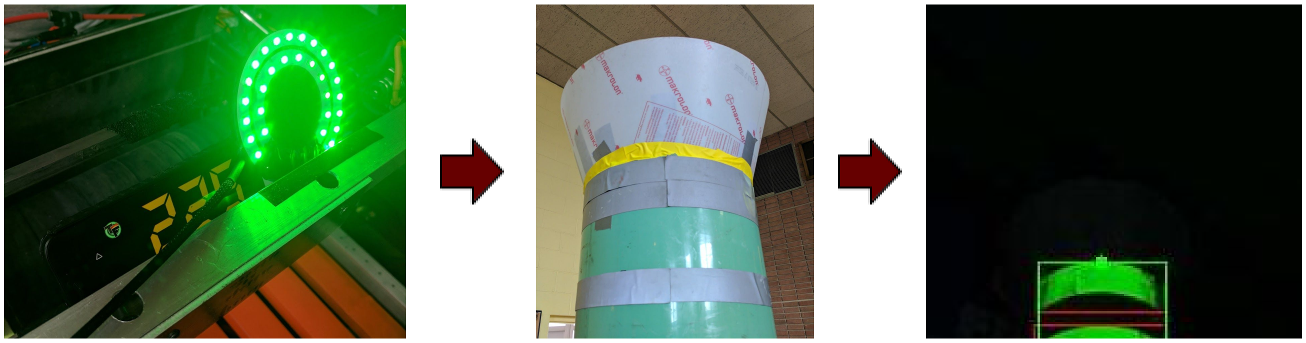
2. With flywheel



3. With more inertial flywheel and further tuning

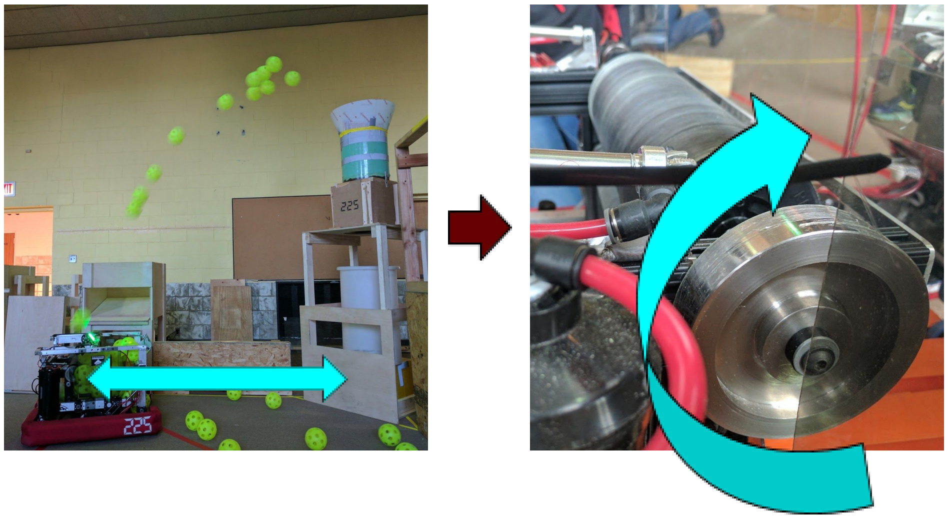


As for the automatic alignment and selection of RPM, a Nexus 5X Android phone is mounted in the center of the robot, running a custom application for vision tracking. The application uses OpenCV to analyze the picture that the phone's camera sees and select the target by seeing the reflection of the green LED ring in the reflective tape around the goal.

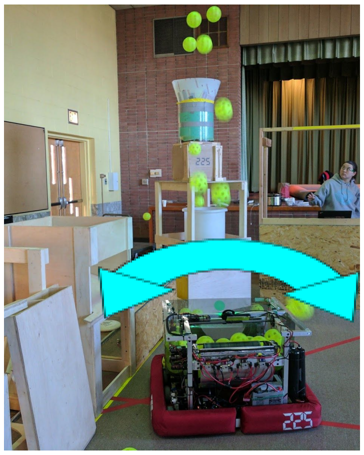


Once the application identifies the target, it then uses some simple trigonometry along with stored information about the focal length of the camera, the height of the camera, and the pitch of the camera, to determine the distance and angle that the robot is away from the target. The Nexus 5X is able to reliably obtain this information at ~30 frames per second and send it to the robot's main controller, the roboRIO. To align the robot with the target, the roboRIO then runs a PID control loop on the robot’s angle, the setpoint for which is determined by comparing the error that the camera reports with the roboRIO’s history of angular positions. To set the shooter to the correct RPM, the roboRIO inputs the robot’s distance to the target into a lookup table of predetermined distance-RPM value pairs, finds the correct RPM based on linear regression of these predetermined points, and passes the RPM to the shooter control loop, allowing us to obtain accurate and automatic shooting.

Robot-to-target distance determines the shooter’s RPM



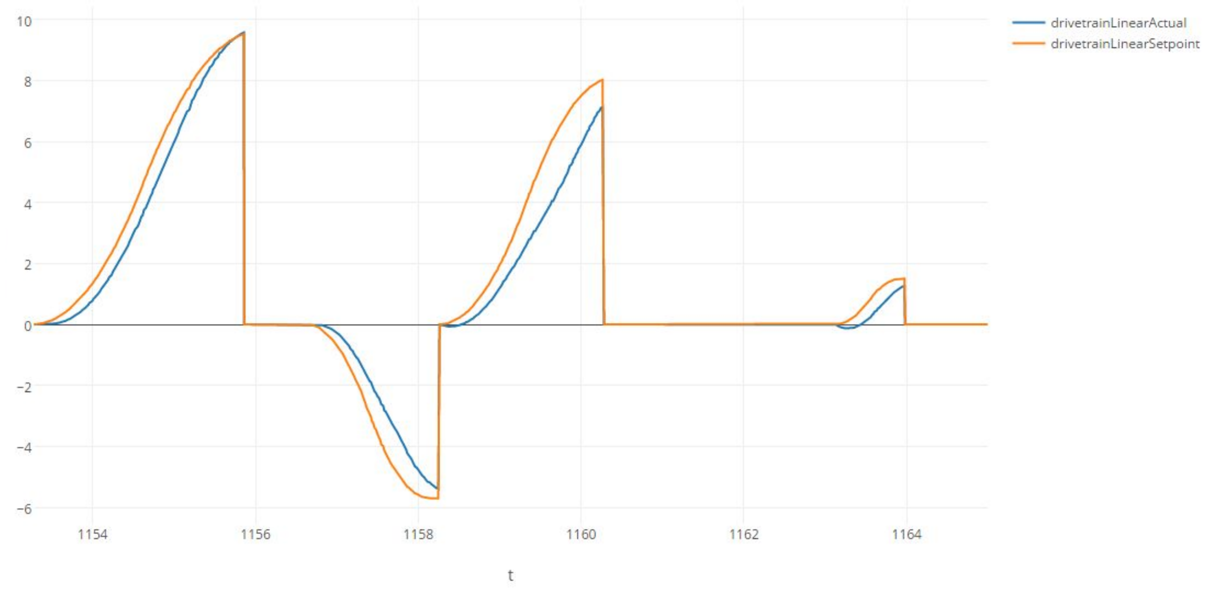
Angular alignment



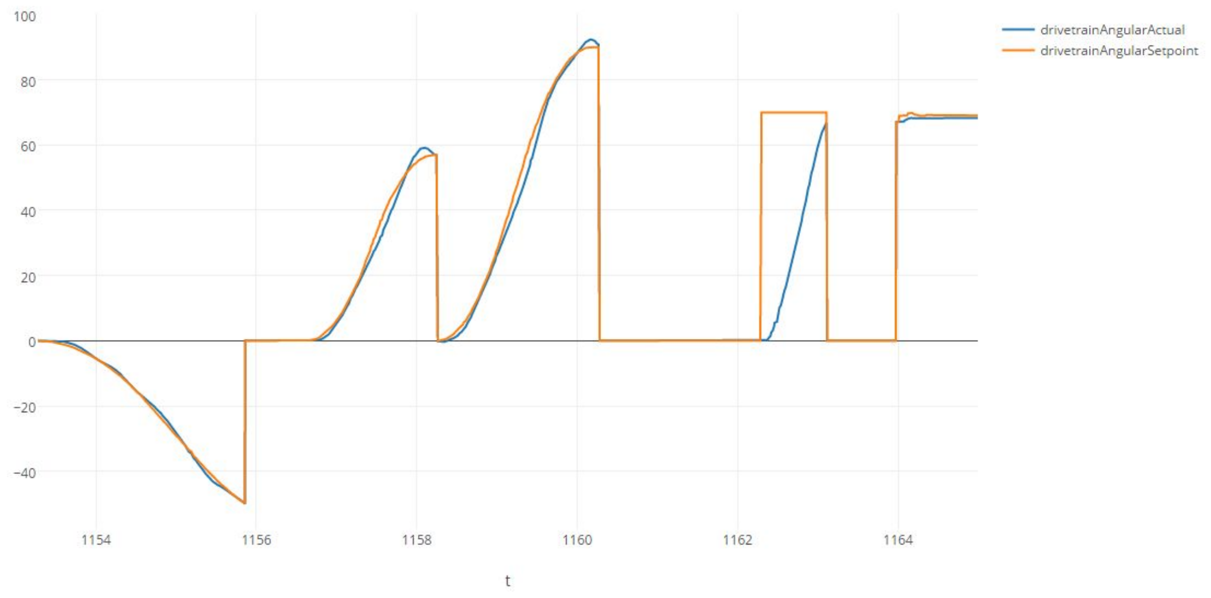
Drive Control

For precise and repeatable driving in autonomous, we use simultaneous profiled controllers on the linear and angular positions of the robot. That is, the roboRIO generates profiles for the robot's position and angle, dictating the distance/angle, velocity, and acceleration for each point in time during the autonomous routine. The roboRIO then holds the robot to these profiles with the use of PIDF control loops. Below are visualized the linear and angular profiles for our autonomous routine that places a gear on the lift, collects fuel from the hopper, and shoots.

1. Linear autonomous profile (in feet)



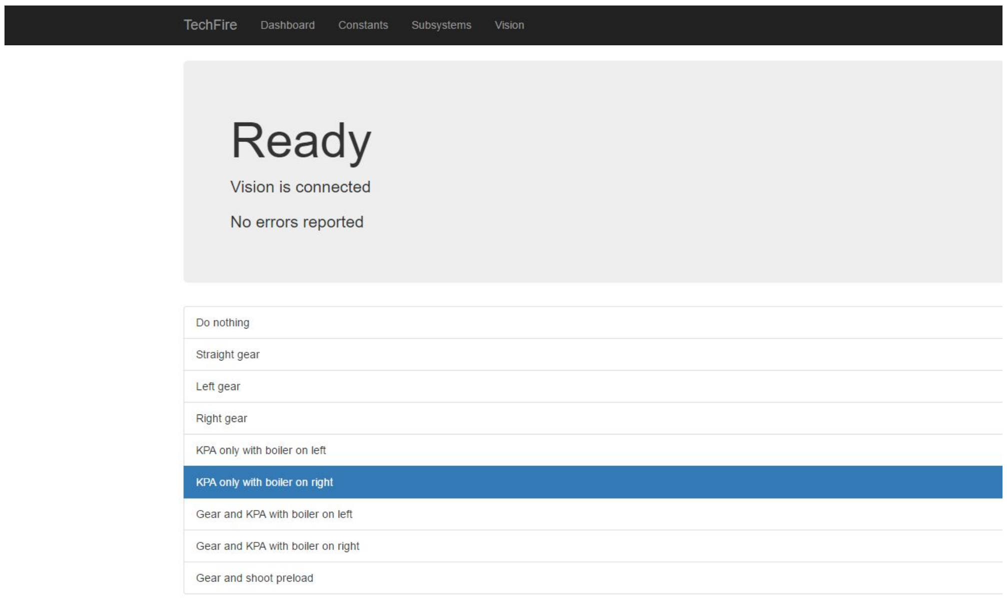
2. Angular autonomous profile (in degrees)



Custom Web Application

This year, to aid in prototyping, tuning, match preparation, and even match play, we developed a custom web application that runs on the roboRIO and is accessible from the driver station. Below are pictured the various functions of the web application.

1. Dashboard: Status of robot and vision system and autonomous selection



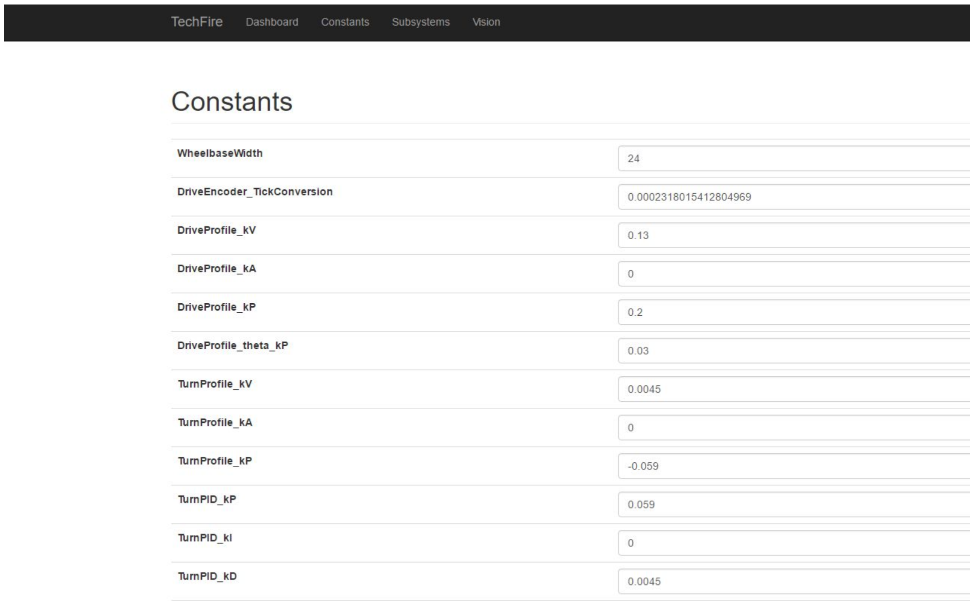
Ready

Vision is connected

No errors reported

- Do nothing
- Straight gear
- Left gear
- Right gear
- KPA only with boiler on left
- [✓ KPA only with boiler on right]
- Gear and KPA with boiler on left
- Gear and KPA with boiler on right
- Gear and shoot preload

2. Constants: Real-time editing of values for fast tuning



Constants

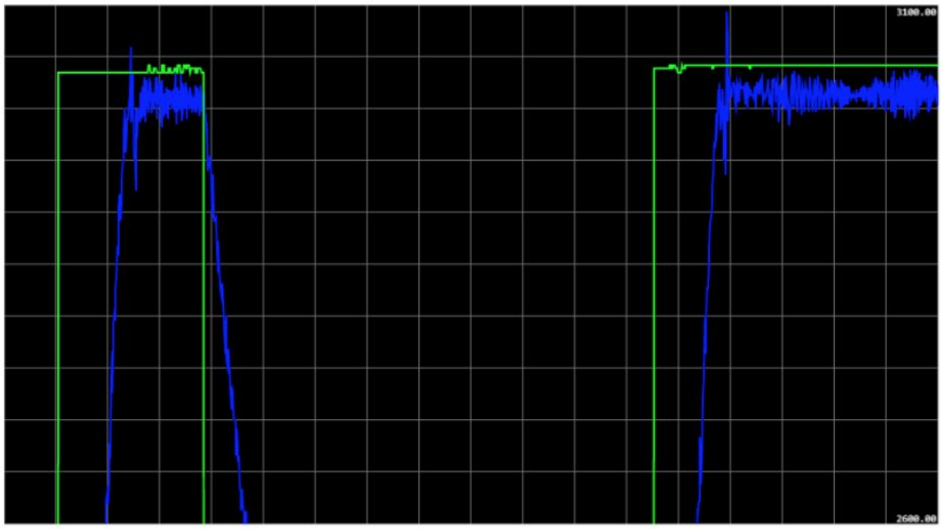
WheelbaseWidth	24
DriveEncoder_TickConversion	0.0002318015412804969
DriveProfile_kV	0.13
DriveProfile_kA	0
DriveProfile_kP	0.2
DriveProfile_theta_kP	0.03
TurnProfile_kV	0.0045
TurnProfile_kA	0
TurnProfile_kP	-0.059
TurnPID_kP	0.059
TurnPID_kI	0
TurnPID_kD	0.0045

3. Subsystems: Real-time graphs of important variables and ability to record CSV of all robot variables and subsystem states, as well as raw display of variables

TechFire Dashboard Constants Subsystems Vision

t = 1118.559229

Shooter FAR_VISION



100rpm/y section, 1000ms/x section

Recorder

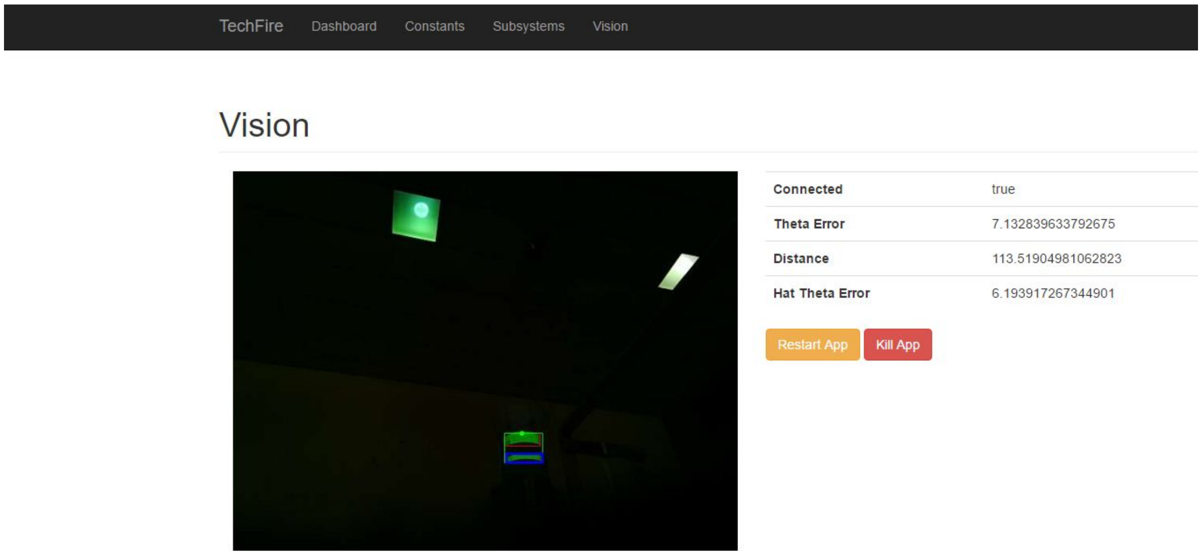
Start Recording

Download CSV

Raw Values

drivetrainMode	OPEN_LOOP
drivetrainDistance	1.376669353664871
drivetrainLeft	1.3963724846737133
drivetrainRight	1.3569662226560288
drivetrainAngle	68.0436375
compressorCurrent	0.5
drivetrainLinearPos	0
drivetrainLinearActual	0
drivetrainLinearError	0
drivetrainAngularPos	68.99433401251042
drivetrainAngularActual	68.2594375
drivetrainAngularError	0.814325855084661
gearHolderMode	CLOSED
hopperMode	OFF

4. Vision: Live feed of vision camera view, display of vision variables, and ability to either restart or kill the vision app remotely.



Vision

Connected	true
Theta Error	7.132839633792675
Distance	113.51904981062823
Hat Theta Error	6.193917267344901

Loading Station Alignment

Finally, perhaps the most creative part of our design is our method for aligning the robot with the human player loading stations during the teleoperated period. Because of the visual obstruction that the airships cause, we realized the need for a method of automatic alignment. We ultimately implemented the strategy of using our shooting vision tracking system to also track reflective tape sewn onto our human player's hat.