Control Award Submission - KookyBotz 16379

Autonomous Objectives

High-scoring & reliable auto, that is partner adaptable

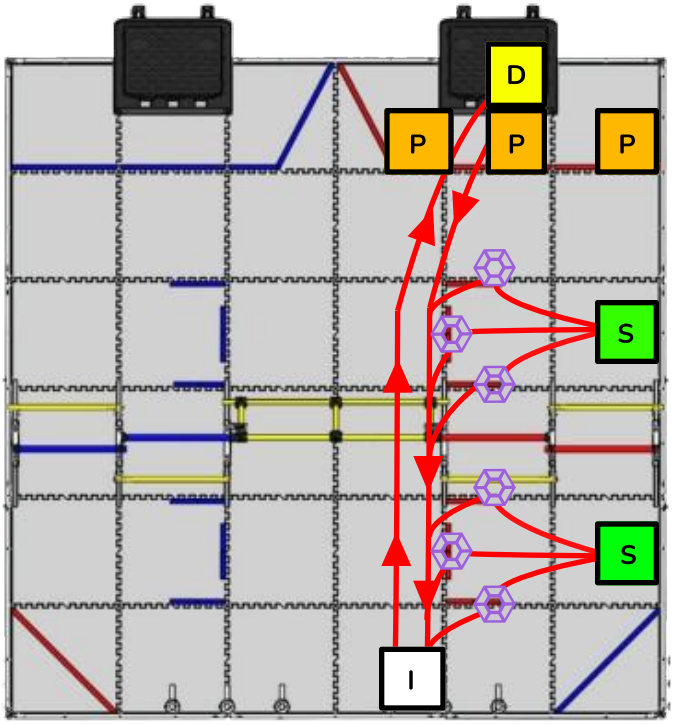
1. Detect the team prop using a **computer vision model**
2. Delay auto if needed, to work with our partner's auto
3. Rapidly scores the purple and yellow preloads
4. Driving by wall or through stage door, intakes 2 stack pixels
5. Approaches the backdrop and scores the pixels
6. Repeats steps 3 and 4 until 6 pixels are cycled
7. Parks in the 1 of 3 possible locations

Autonomous Enhancements

Uniquely during autonomous, our robot performs these functions to ensure accuracy, consistency, and performance.

- Detecting the partner pixel on the backdrop, we score in the other column, ensuring 50 points.
- Relocalizes using ultrasonic sensors and apriltags
- Detects robot collisions and ceases robot operations while attempting to re-align or re-score

Autonomous Map

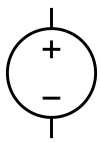


I Intake (3x) S Start (1 of 2) P Park (1 of 3) D Deposit (4x)

Sensors Used

Cameras (x2)
Read team prop for autonomous and relocalize using april tags.

Distance Sensors (x2)
Assist with redundancy of localization and alignment.



Voltage Sensor (x1)
Enables consistent power output (scaling power values according to battery charge).

Absolute Encoder (x1)
Allows us to track the position of our arm with 12 bit precision

Odometry (x3)
3 deadwheel odometry pods localize our robot.

Motor Encoder (x1)
We use a motor encoder to track the position of our box tube extension for consistent intaking.

IMU (x1)
Tracks the angular orientation of our robot.

AXON

After discovering an lack of sensors and actuators purpose built for FTC, our captain founded the company Axon-Robotics. We primarily use sensors from Axon, and our captain handles the manufacturing, and distribution.

Driver Enhancements

Finite State Machine Control
By modeling every possible transition and state the robot can be in, we queue certain behaviors and outputs in order to actuate and control our system efficiently and without error. We do this with a command based system, making additions of new states trivial.

Slew Rate Limiting
Scaling robot acceleration non-linearly through a piecewise cubic model reduces wheel slip and traction loss when autonomously driving.

Simple Input, Advanced Performance: Arm Inverse Kinematics
Presets on our controller streamline scoring on the nth row, by providing data to our arm inverse kinematics that move it to convenient scoring positions, separating the driver from manually controlling fine details such as deposit height or angle. Read more about how our kinematics work on the next page.

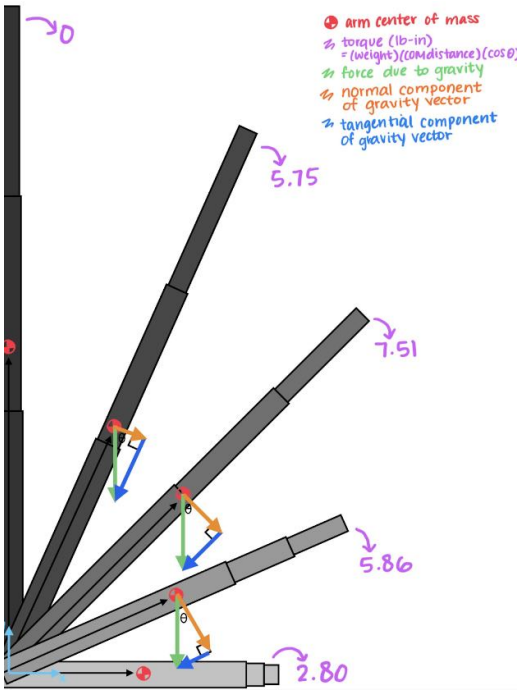
Drone FSM:	Fired		Armed
Claw FSM:	Drop	Closed	Open
Arm FSM:	Scoring	Stored	Intake
Pivot FSM:	Scoring	Stored	Intake

Control Award Submission - KookyBotz 16379

Key Algorithms

Arm Physics and Inverse Kinematics

Our robot's extremely large and fast arm presented a unique opportunity to introduce physics modeling and driver enhancements- control that could show a considerable performance benefit over robots of the same archetype. This also presented a learning opportunity for us, applying math and physics to FTC.



Inverse Kinematics

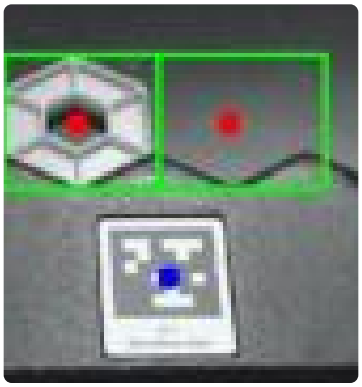
Using a mathematical model we take in target row of the backdrop and distance from the backdrop. The model then translates them into angles the robot can understand and travel to, which are arm and wrist positions. We derived this model using trigonometry and dimensions from CAD, and it has trivialized our deposit code. This mathematical arm control allows higher accuracy and precision, compared to arbitrary scoring height presets.

Arm Feedforward Gravity Compensation

Calculating the center of mass of our arm and the effects of gravity on it enable us to calculate the exact gravitational effects exerted on our arm motor. Based on that force, we can then apply a modeled feedforward to our PID control which will keep the arm balanced at whatever angle it was sent to.

Partner Pixel Detection

Detecting partner's preload and scoring in the adjacent column, ensures 50 preload points, lessening alliance coordination requirements.



April Tag Relocalization

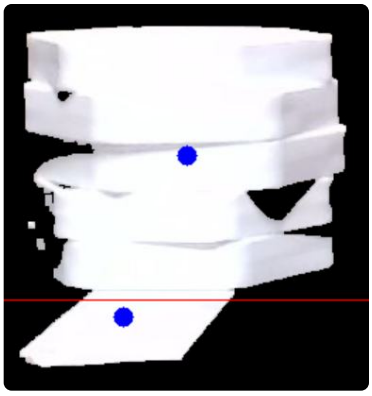
Every time our robot scores, we use the april tags on the backdrop to relocalize.

Loop Time Optimizations

A mix of performant code, publisher/subscriber format lynx reads and write, and UUID hub identification fixes, our teleop improved from running at 50hz to running at 450hz (+700%!).

Stack Detection

Using a CV model, we analyze the centroids of visible white contours, and based on each contour's width and height, we use a linear regression to determine the distance and angle of the stacks from our camera, allowing for error-less realignment and intaking.



CSV Data Logging

We timestamp and log robot data, such as robot positions or extension ticks, allowing for a seamless debugging process.

Custom Pure Pursuit Pathing

Our custom pure pursuit pathing gives us smooth trajectories compared to rigid point to point paths. Our following system performs movements tangential to the path, which is the most efficient and fastest for mecanum characteristics, which results in a higher velocity on average and quicker path completion (one such test shown on the right).

Arm data over time.

2300ms Path	1640ms Path
Old PID to Point Pathfinding	Our Custom Pure Pursuit

Engineering Portfolio References

- Arm Inverse Kinematics (pg. 14)
- Dynamic Feedforward System (pg. 14)
- Robot Control Architecture (pg. 12)
- Vision Localization (pg. 13)
- Open-Source Github Repository (pg. 3)
- Autonomous Strategy (pg. 4)