

33778

Mechanical Error Spring

2026 Portfolio



Build

Mechanical Errors robot left from a simple structure with a basic design, inspired by other robots in our league, to a much more complex design able to properly play in the big leagues of New Mexico's FTC robotics competitions. Keeping the same wheels and motors, however altering the way the motors are attached to wheels changes everything by adding space, weight distribution, and better control over the design. The motors now reside in the center of the robot, connected to the mecanum wheels using a simple bevel-gear and belt system.



Build (Page 2)

Refining the Design for Efficiency, Durability, and Performance

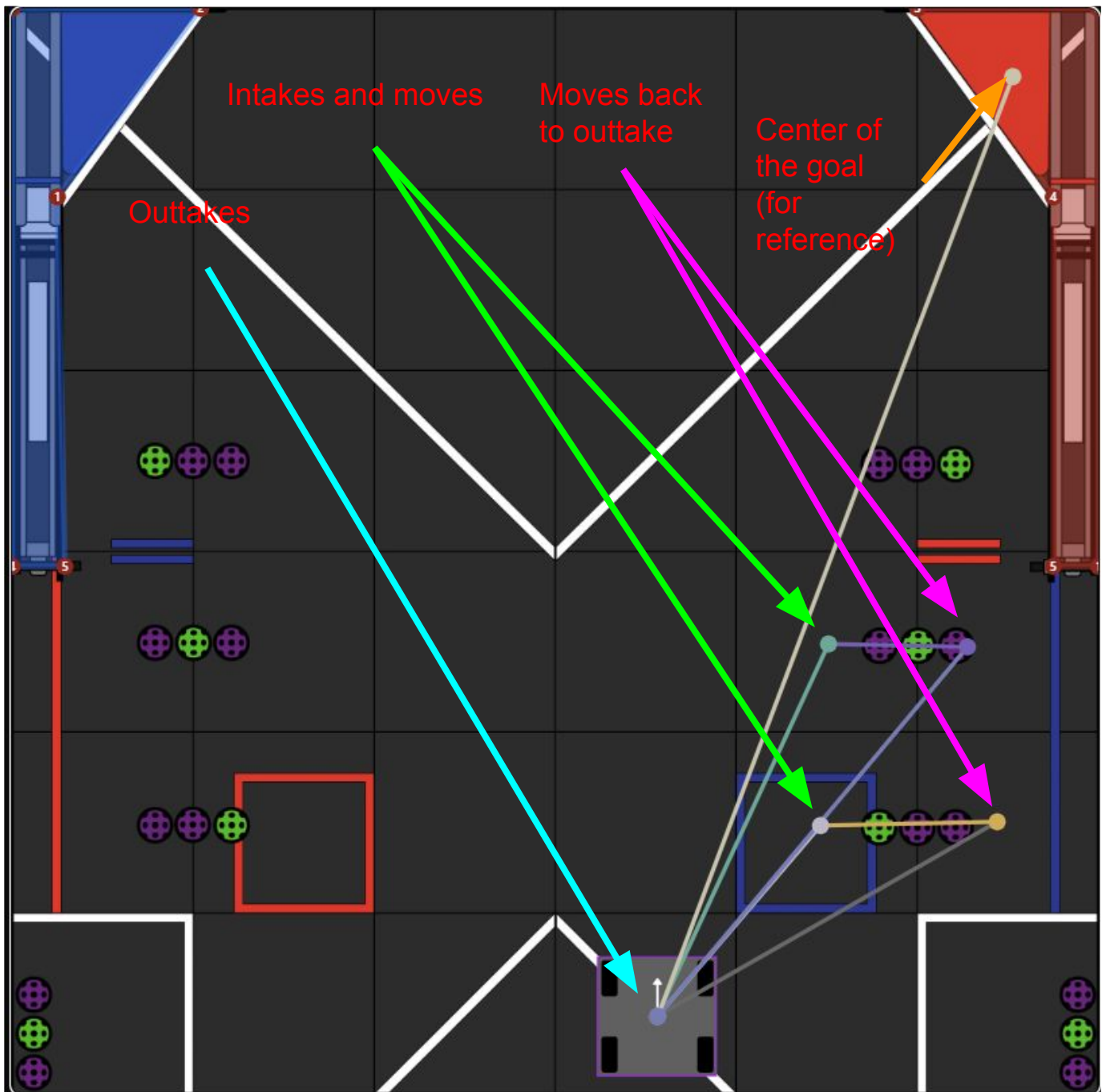
- **Improved Motor Protection**
 - Motors are now fully enclosed within the robot for better protection and durability.
- **Stronger Structural Foundation**
 - Polycarbonate side plates form the main chassis structure.
 - Reinforced with metal wheel guards for added stability and impact resistance.
- **Streamlined Ball Path**
 - Replaced the original two-part system and servo arm mechanism.
 - Added a single polycarbonate track that moves balls directly from intake to outtake.
 - More direct path reduces jams and increases reliability.
- **Rear-Facing Flywheel System**
 - Balls are shot out the back of the robot.
 - Flywheel spins opposite the intake direction.
 - Reduces unnecessary movement, improving scoring speed and overall efficiency.
- **Redesigned Intake & Outtake (CAD + 3D Printed)**
 - Newly CAD-designed, single-piece components.
 - Eliminates incompatible parts and inconsistencies.
 - Better ball fit improves ball control, shot accuracy, and shot consistency.
 - Multi-layer 3D printed construction increases strength and durability.



Code (page 1)

Auto:

For our autonomous we use PedroPathing to make sure we will reach our set positions efficiently. Below is a diagram of one of our paths explaining what it does at each point.



Code (page 2)

Autonomous Logic, TeleOp Control, and Code Safeguards

Autonomous Pathing System

- Setpoints stored in an array of positions.
- Robot advances through setpoints using an index that increments upon reaching each target.
- An enum determines when the robot intakes and outtakes
- Multiple selectable starting positions (including red/blue alliance and near/far auto).

TeleOp Control

- Uses mecanum drive kinematics for precise directional movement.
- Two preset shooting setpoints for consistency.
- Outtake PID control for reliable, repeatable shots.

Backup Systems

- Created backup Autonomous and TeleOp programs.
- Exact copies of competition-ready OpModes at event start.
- Prevents unexpected issues when modifying code during competition.



Code (page 3)

Proportional Integral Derivative for outtake

- Derivative coefficient set to 0 because outtake works well with just PI
- Can be set to a near shot speed and a far shot speed

What is a Proportional Integral Derivative

- A proportional integral derivative (PID) takes an input wantedSpeed
- Every time the system loops:
 - The PID finds the current speed of the motor
 - This is then turned into error = motorSpeed - wantedSpeed
 - The proportional component P is set equal to error
 - The integral component I gets error added to it
 - The derivative component D is set to (error - lastError)/timeChange
 - The PID outputs a motor power of $K_p*(P + (I/T_i) + T_d*D)$ where K_p , T_i , and T_d are constants based on your control and physical systems

What does PID do?

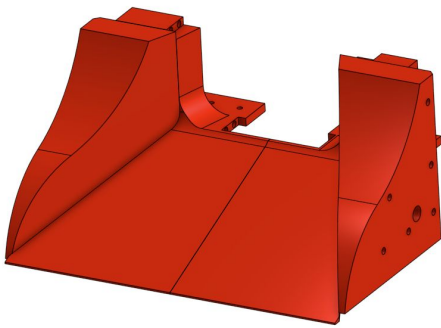
- PID (in our case) makes the outtake motors get up to speed quickly and makes our shots much more consistent
- Individually each component does:
 - P is just the component to make sure the system gets to the speed it needs to at all
 - I eliminates problems where the speed will be a constant speed off of your intended
 - D makes the entire system much faster at getting up to speed, but isn't recommended to tune if the system works without it

Other options

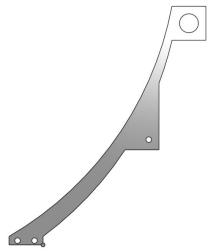
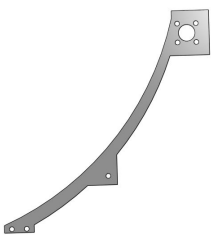
- PID isn't necessary by any means, you can just run without it and everything will work, but slower
- You can also use another variation of PID, such as PIDF
- From a little research, it seems when using another type of PID, you can mostly ignore the I part and just use another type of feedforward

Design

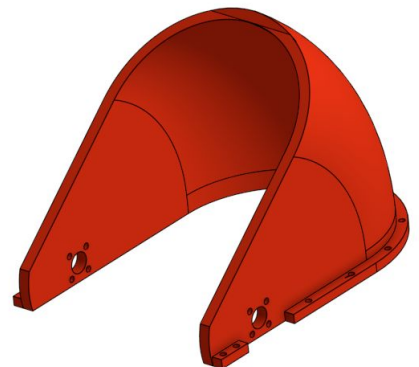
The Mechanical Error design is made off of Onshape. There is two main folders that we used to organize the CAD. One is called the “Intake” and the other is “Outtake.” Most of the robots structure is made from ¼ Inch thick polycarbonate that was CNC-ed. We have two side-plates, one top-plate, and one backplate.



The CADs that exist in the “Intake” folder is the intake ramp which consists of 4 parts that are connected by a heat press and screws. The index railings are 4 CNC-ed parts (2 top railings and 2 bottom railings) that carry the ball from the intake to the outtake.

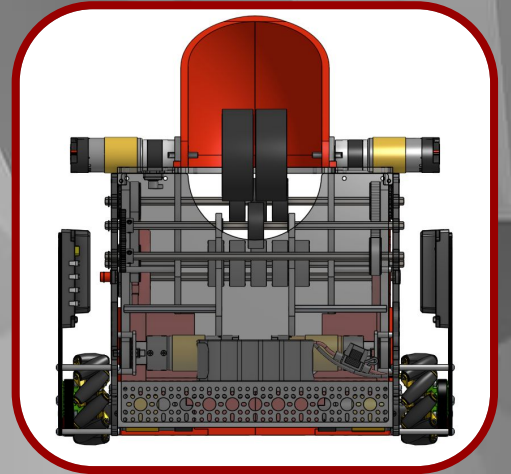


The Flywheel hood falls into the “Outtake” file and lays flat on the CNC top-plate. The ball comes at around 36 degrees so the ball can reach maximum distances with backspin included.



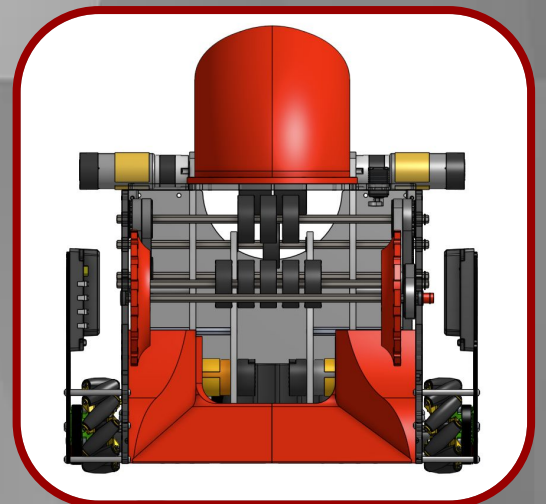
Design (page 2)

This is the view of the front face of the robot and the most important parts on this face is the flywheel, the control hubs, and the backplate.



This is one of the side views of the robot and it lets you see the robot's pulley system for the index and the intake. It also shows where the motors are mounted as well as the wheels.

This angle shows the front side which has the intake ramp, the rubber band geers for the intake, the and the index with all the wheels.



Outreach

We've done all this, but what have we done to spread and experience STEAM with others? Davy went to Gallup with ENIGMA to help another rookie FTC team set up their field and to teach them the basics of code, Sammy helped teach Hack City how to use android studio, and Asa helped Astro Viking with in-pit competition issues at Albuquerque comp. He also taught them how to properly route wires and create stable servo structures. His helpfulness goes beyond just that, including assisting La Academia de Esperanza finish cleanup after comp, being part of the last team returning things to rightful places and owners.

