Introduction:

We made a program to dynamically generate autonomous trajectories for FRC robots in order to avoid obstacles. We did this by using A* and rejecting all paths that pass through obstacles.

Methods:

We had to optimize this algorithm in order to get it to generate paths quickly on the limited resources of an FRC robot. We tested our system by running it inside of AdvantageScope a simulation software built for FRC.

For the path generation, used A* algorithm to find the shortest path between the robot’s initial location and the location we want to be going. We organized the algorithm to reject the neighbor nodes inside obstacles which made us avoid obstacles along the way. Because the field is not a grid, we rewrote the getNeighbors method to return neighbor nodes using the coordinate system that the robot moves on. We used a heuristic of distance to the final position plus the distance from the start position, this made it generate the shortest possible path while avoiding all obstacles.

For the robot's movement, we used a command called the "Ramsete command" which combines the feedback from the feedforward controller with the robot's current pose to calculate appropriate control signals, such as wheel speeds or motor voltages. By using this command, the robot was able to follow the trajectory we created.

There is also a simulation code that enables the users to see how the robot moves using the WPILIB shuffleboard app.

Results:

Our implementation worked, and avoided all of the obstacles, but generated some fairly wiggly paths due to an issue in the trajectory generation system.

Discussion:

In the future, we are going to improve this program by trying to fix the issue with trajectory generation and format our code as a library so others can use it.

Video of the robot simulation