# Robotics: Odometry and Pure Pursuit

Jack Hughes
2022
Kent Denver Innovative Scholars Program

Engineering, Robotics
Alainey Hellman

## MOTIVATION

In order to be more competitively successful, I wanted more consistent autonomous routines on my competition robots. In order to graduate from timers and basic tracking, I pioneered the creation of a full position tracking system and a complex motion algorithm for smooth and consistent movement at least as good as a human driver..

## RESEARCH/DESIGN

The first step was improving my tracking algorithms to be more accurate. Building on my previous year's work, I improved the modeling of robot motion by representing translation through arcs instead of straight lines, and I improved the build quality of the tracking wheels, resulting in nearly perfect tracking.

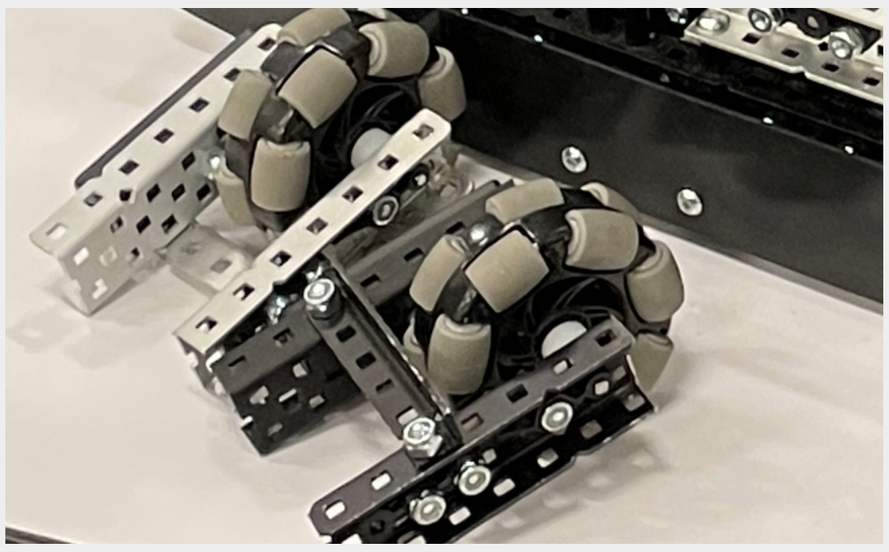

FIGURE 1: Improved Tracking Wheels

Most of my work was on the second phase of the project: an autonomous routine management app and the robot-side code needed to execute routines made by this app. This necessitated the research of multi-language code integration, standard and hermite cubic parametric splines, the pure pursuit algorithm, and file management.

## THE APP: Ender Operating System

I created this app from scratch in order to quickly  plan and adjust autonomous routines to decrease the amount of time spent on the testing and creation of autonomous routines before every tournament, as well as to provide a visualization of these routines instead of simply looking through hundreds of lines of code. This improved useability and allowed teammates not well-versed in code to create and tune routines without me present which was a huge productivity boost.

The most difficult part of the app was making it "talk" to the robot; the app was made in Java and runs on my laptop, meanwhile the robot runs in C++ and does not run through my laptop. Ultimately, I had to painstakingly program the app character by character to write directly to the source files of the robot's code and then upload it to the robot.

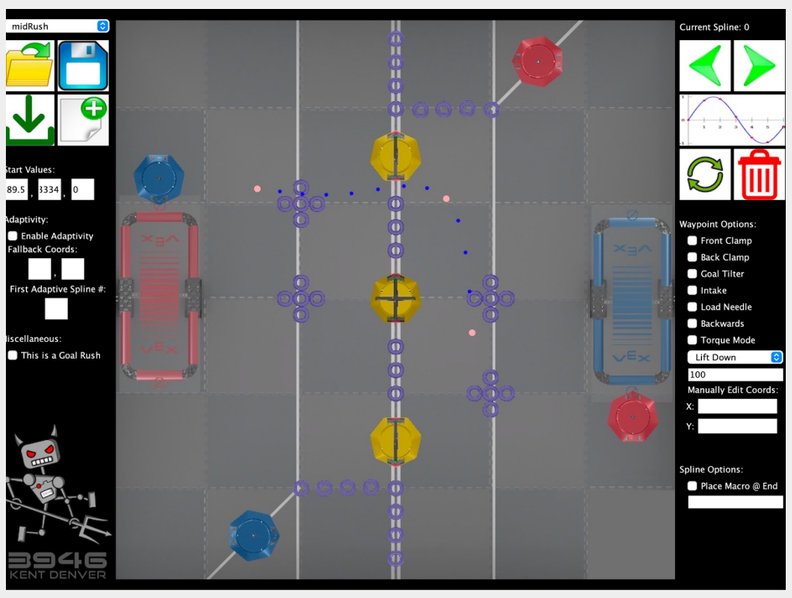


FIGURE 2: A Screenshot of EOS

## PURE PURSUIT, RESULTS

The last portion of the project was the motion algorithm—generating a series of waypoints along a smooth curve with instructions for what the robot should do at those points is one thing, making the robot follow that curve and do those actions at the right time is another.

This is where the pure pursuit algorithm comes in. Essentially, a circle is virtualized around the robot's tracking center, and the robot steers itself towards the intersection of that circle and the path. The velocity of each wheel is calculated with inverse kinematics, and the robot slows down around turns and at the end of paths because of a different set of inverse kinematics equations.. This part of the project took the longest because there is no way to debug this code as it is buried in the robot's brain in a strange format, so it was guess and check for multiple weeks.

Ultimately, this was a successful project; the robot followed any given path and executed any inputted actions at points along that path with great consistency and reliability. We unfortunately did not get the chance to use this system at the world championship due to hardware-related robot issues that cropped up a week before the tournament (a nice way of saying we completely rebuilt the robot in a week).

I have taken the time to teach the younger students in our robotics club the math and code behind the pure pursuit algorithm and the app, and I can say with confidence that this code will be looked to as an example and a huge resource for our students for years to come.

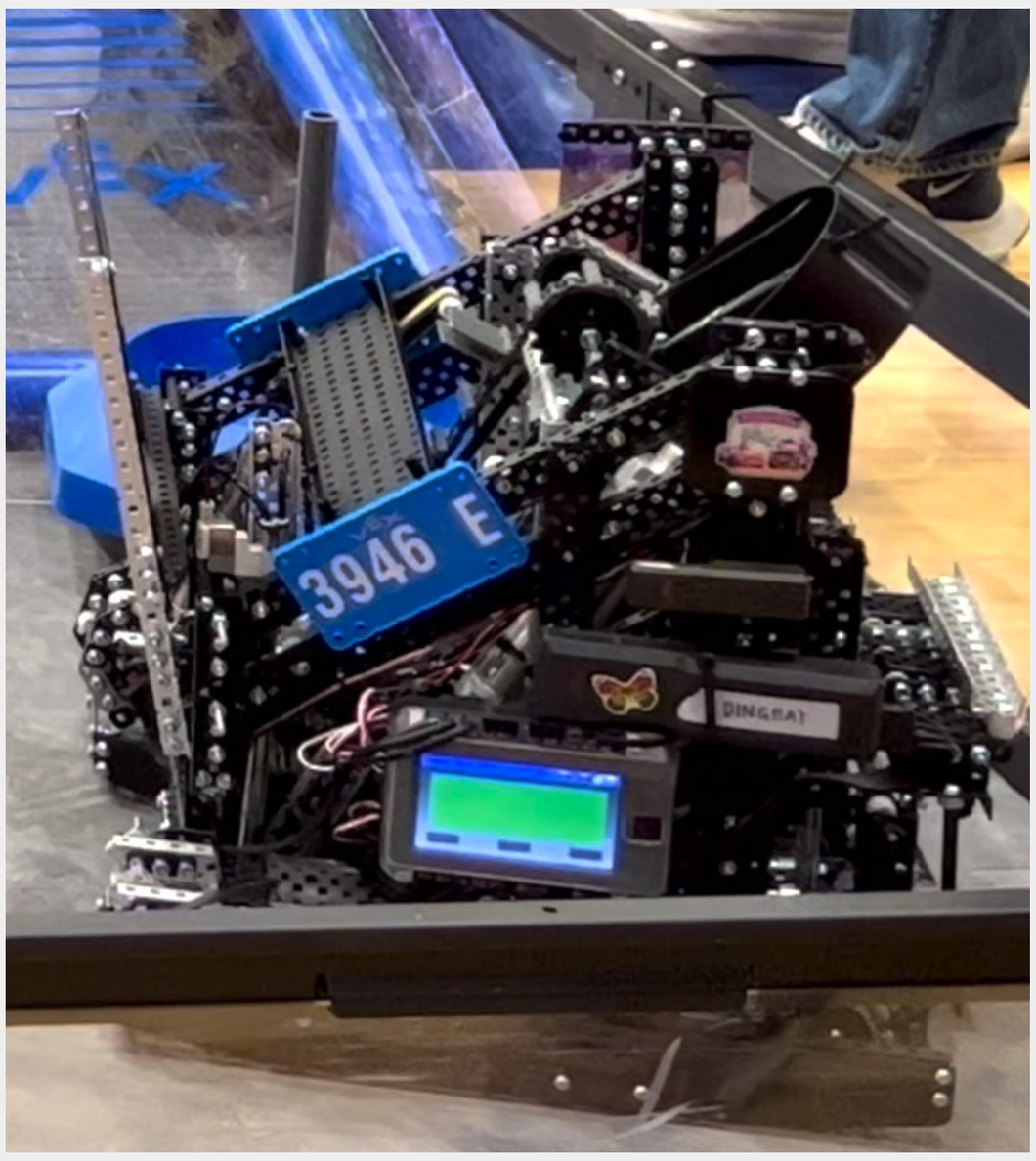


FIGURE 3: Our States Robot (On Which the System Worked)

## CONCLUSION

I learned a lot about robots and complex motion and tracking math in this project, it was a worthwhile experience!

I hope future robotics students can benefit from our learning journey and all the resources we are leaving behind for them!