Following Bezier Curves
Cameron Neville
Malden Catholic High School
cjn5962@g.rit.edu

Following Bezier Curves

1 Following a Bezier Curve

The purpose of this project is to program a robot to follow a bezier curve. The robot is passed a series of *x* and *y* coordinates and generates a path. It is written for robots using an iCreate base for the sake of standardization.

2 Bezier Curves

A bezier curve is a smooth curve that approximates a path through an ordered series of points. A bezier curve does not always pass through the points that define it, but each point "pulls" the curve towards it, as seen in Figure 1.

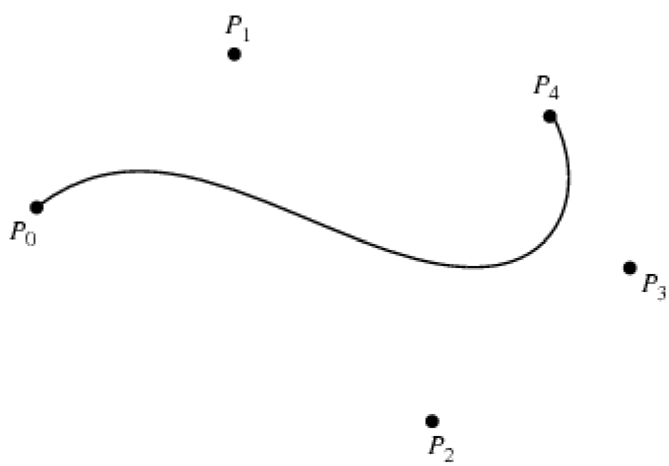

Figure 1: An example of a bezier curve

2.1 Why Bezier Curves?

Bezier curves are ideal for this project both because they are relatively simple for users to approximate and because they are easy to code for. Because bezier curves are defined by a series of points, it is easy for users to approximate a path. This is essential in situations where extensive testing of paths in impossible. In addition, their starting and ending points are the first and last points respectively, so it is simple to know where the robot will stop.

The other main benefit of bezier curves is their mathematical properties. A bezier curve can be defined by a pair of parametric equations with *x* and *y* as polynomial of *t*. This makes calculus involving these curves easier and more precise for computers. The power rule can be applied to quickly take derivatives, and calculations are easy to perform.

3 Following Curves

The bezier curve is approximated by a series of 1000 arcs. Each arc has a differential between the left and right wheel speeds appropriate to the curvature at that section. The length of each arc varies based on the section of curve. The robots progress is regulated by the variable t, which progresses from 0 to 1 in intervals of .001. All calculations are performed using parametric equations of x and y.

3.1 Arc Length

The robot finds the appropriate length of the current segment by using the arc length formula. The driving distance is regulated by calculating the net robot velocity, which is the average of the left and right wheel velocities, and driving for a variable amount of time. Once the arc is completed, t is increased by .001 and the next arc is started.

3.2 Angles

The robot finds the angle it must reach by the end of an arc using $\arctan((dy/dt)/(dx/dt))$. It subtracts this from its current angle to find the necessary change in angle over this arc. The differential between the left and right wheels is determined based on the change in angle over the arc length to be traveled.