Quintic Splines for FTC

1 Introduction

In this paper, we discuss the use of quintic splines for more sophisticated robot pathing in FTC. Traditionally, FTC robot autonomous motion consists of linear movements (including holonomic drive strafing) and point turns. And though these simple path primitives generally suffice, they can be inefficient, especially for motion-profiled nonholonomic drivetrains. That is, the fastest route between two poses (Cartesian position and heading combined) in these situations is not two point turns and a line. To address this, we propose quintic splines to achieve smooth, curved motion on the field.

In the first part of the paper, we will describe the problem in depth and provide motivation for splines. Next, we explore some of the mathematics behind quintic splines including interpolation and arc length parametrization.

2 Problem

2.1 Coordinate System

Although the methods presented here do not depend on a specific coordinate system, it's essential to establish a consistent reference frame for describing robot movement. Figure 1 shows an example coordinate system for the Relic Recovery field. Most importantly, it is right-handed, and its origin is in the center of the field. Many teams avoid an explicit coordinate-based representation of their robot motion, but this is a crucial tool for unambiguously describing more complex paths.

2.2 Pathing

In FTC, it's standard to have a sequence a poses that you want the robot to follow in autonomous (although pre-planned motions can also be utilized in TeleOp). For example, in the Relic Recovery game, a common movement task may be moving from one pose on the balancing stone to another in from of the cryptobox. This is traditionally handled with a series of straight lines and turns (two turns and a line can connect any poses with nonholonomic constraints). Additionally, these lines are typically followed using PID controllers with capped outputs (to prevent sharp accelerations that can cause wheel slippage and greater odometrical error).

For many game tasks and teams, this is a perfectly viable approach. However, in scenarios where speed is desirable (as in Relic Recovery with an effectively unlimited scoring potential in autonomous), First off, motion profiling can be used to achieve higher speeds without sacrificing accuracy by observing the robot's physical constraints (i.e., maximum velocity,

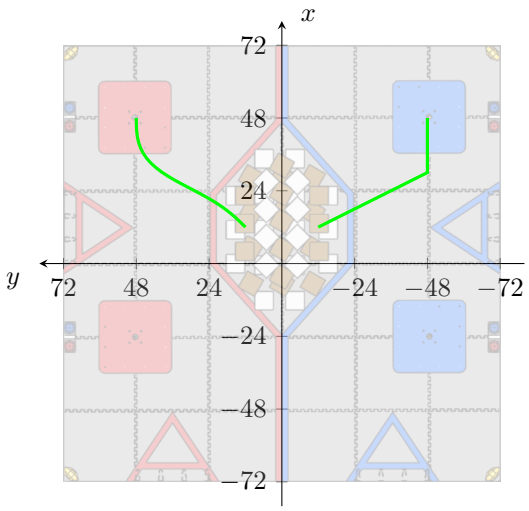

Figure 1: Example coordinate system for describing positions on the Relic Recovery field. This right-handed frame can be extended to three dimensions with the positive Z axis protruding from the origin. The path on the blue side is a typical autonomous route seen in FTC (don't forget there's a turn between the line segments — three profiles in total). The path on the red side is a quintic spline version of roughly the same move. The spline path is 1.6 times faster than the conventional path with reasonable drive constraints.

acceleration). For a good introduction to motion profiling, see the canonical talk by mentors from FRC teams 254 and 971[1]. Secondly, for nonholonomic drives (e.g., tank/differential drives), two turns and a line are required to navigate between two poses in the general case[2]. This separation of rotation and translation in turn demands multiple profiles and unnecessary decelerations and accelerations. To address this, it is desirable to combine the rotation and translation into a single, smooth motion.

This can be accomplished using a variety of path planning techniques. For simplicity, this paper only considers polynomial splines (which are well-suited for most FTC game tasks). Quintic splines in particular were chosen to guarantee C^1 heading continuity without introducing an unreasonable number of parameters (that, when chosen poorly, can lead to unnecessary extra curvature).

3 Interpolation

Quintic splines are essentially piecewise curves composed of quintic polynomials. In the context of mobile robot pathing, these splines are usually split into two parametric components $x(t)$ and $y(t)$. Therefore, a quintic spline of n segments can be represented by n pairs of

[1] `https://www.youtube.com/watch?v=8319J1BEHwM`

[2] A keen reader will realize that this is not necessarily the case for holonomic drives. All pose-to-pose movements can be executed with a combination of strafing and rotating. Nevertheless, splines can still be of use. For one, traveling along the tangent can reduce odometrical errors accrued from translating on the lateral axis. Additionally, splines can help navigate around obstacles in situations where a piecewise linear path would normally be required.

quintic polynomials with the following form:

$$\begin{cases} x^{(i)}(t) = a_x^{(i)}t^5 + b_x^{(i)}t^4 + c_x^{(i)}t^3 + d_x^{(i)}t^2 + e_x^{(i)}t + f_x^{(i)} \\ y^{(i)}(t) = a_y^{(i)}t^5 + b_y^{(i)}t^4 + c_y^{(i)}t^3 + d_y^{(i)}t^2 + e_y^{(i)}t + f_y^{(i)} \end{cases} \qquad \text{where} \quad 0 \le t \le 1$$

Now the goal of interpolation is to "fit" these polynomials between a series of $n+1$ points (commonly referred to as knots) labeled (x_i, y_i). To accomplish this, we can impose the conditions $x^{(i)}(0) = x_i$ and $x^{(i)}(1) = x_{i+1}$ (and correspondingly for $y^{(i)}(t)$). Additionally, to ensure greater continuity, we also match the first and second derivatives at each knot point[3]:

$$\left.\frac{dx^{(i)}}{dt}\right|_{t=0} = x_i'$$
$$\left.\frac{dx^{(i)}}{dt}\right|_{t=1} = x_{i+1}'$$
$$\left.\frac{d^2x^{(i)}}{dt^2}\right|_{t=0} = x_i''$$
$$\left.\frac{d^2x^{(i)}}{dt^2}\right|_{t=1} = x_{i+1}''$$

(and analogously for y)

Collectively, these constraints guarantee that the overall spline will be (by construction) C^2 continuous. They also fully define the polynomial coefficients for each spline segment. To actually compute the coefficients, we simply need to solve the linear system with following matrix representation:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 4 & 3 & 2 & 1 & 0 \\ 20 & 12 & 6 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \\ e_x \\ f_x \end{bmatrix} = \begin{bmatrix} x_i \\ x_i' \\ x_i'' \\ x_{i+1} \\ x_{i+1}' \\ x_{i+1}'' \end{bmatrix}$$

This equation can then be solved to yield the coefficients with your favorite matrix library (or put in row echelon form to yield a set of back-substitutable equations).

4 Arc Length Parametrization

Now that we've interpolated the spline segments, we have to join together all of the spline segments into a single piecewise function of a unified variable. More specifically, we're going to reparametize each function from $t \in [0, 1]$ to the arc length parameter s. Although it sounds complex, s is just the true displacement along the path (mathematically, $\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2 = 1$).

[3]This is generally the best choice for splines in the context of path planning although other schemes are occasionally employed. For instance, one can instead force the third and fourth derivatives to equal zero.

To shift from t to s, we first have to define $t(s)$. This is difficult to represent analytically although the inverse function $s(t)$ is relatively simple:

$$s(t) = \int_0^t \sqrt{\left(\frac{dx}{d\tau}\right)^2 + \left(\frac{dy}{d\tau}\right)^2}\, d\tau$$

In practice, $t(s)$ can be obtained by progressively evaluating the above integral and stopping when the integral sum reaches s.

By computing $t(s)$, we can determine $x(s)$ (and $y(s)$) by composition: $x(s) = x(t(s))$. Next, differentiating yields the new parametrized derivative:

$$\begin{aligned}\frac{dx}{ds} &= \frac{dx}{dt}\frac{dt}{ds} \\ &= \frac{\frac{dx}{dt}}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}}\end{aligned}$$

It's important to note that $\frac{dx}{ds} \neq \frac{dx}{dt}$ here. It may seem that this messes up the C^2 continuity at knots, but it isn't an issue because $\frac{dt}{ds}$ is the same for both splines (since $\frac{dx}{dt}$ and $\frac{dy}{dt}$ are equal). This expression can then be differentiated once more to get the reparametrized second derivative, $\frac{d^2x}{ds^2}$ (and so on ad infinitum).

5 Conclusion

This paper discussed the motivation for splines in FTC and some basic mathematics for generating basic quintic splines. The author hopes this will help further the proliferation of advanced motion planning techniques in FTC.