FTC Pathing Libraries Comparison & Team Repositories

Introduction

This document compares the three main pathing options for FTC (RoadRunner 0.5, RoadRunner 1.0, and Pedro Pathing) and provides links to repositories of successful teams.

1. Pathing Library Comparison

RoadRunner 0.5 (Legacy)

The "classic" version of RoadRunner. Widely used for years but now effectively deprecated.

- **PROS:**

  * **Massive Community Support:** Years of tutorials, videos, and forum posts exist.

  * **Stable:** Known bugs are well-documented.

  * **Tools:** Many 3rd-party tools (like RRPathGen) were built specifically for it.

- **CONS:**

  * **Deprecated:** No longer actively maintained.

  * **Slower:** Generally slower and less accurate than 1.0.

  * **Harder Tuning:** Tuning PID coefficients can be tedious and difficult.

  * **No Actions:** Lacks the built-in "Actions" system for easy asynchronous commands.

RoadRunner 1.0 (Recommended)

The modern, official version. It is a complete rewrite focusing on performance and ease of use.

- **PROS:**

  * **Better Performance:** More accurate path following and smoother motion.

  * **Easier Tuning:** The tuning process is significantly faster and more automated.

  * **Actions System:** Built-in system to run commands (like "lift arm") while driving, without complex state machines.

  * **Live Dashboard:** Comes with FTC Dashboard integrated for real-time debugging.

* **MeepMeep Support:** Fully supported by the best visualizer (MeepMeep).

- **CONS:**

* **Documentation:** While good, it has fewer community tutorials than 0.5 (though this is improving).

* **Breaking Changes:** Code from 0.5 is NOT compatible with 1.0.

Pedro Pathing

A newer, high-performance library developed by FTC Team 10158 (Scott's Bots).

- **PROS:**

* **Extreme Speed:** Designed to be faster than RoadRunner in many scenarios.

* **Reactive:** Can correct for disturbances (like getting hit) in real-time better than standard followers.

* **Centripetal Correction:** Handles curves at high speeds very well.

* **Customizable:** Great for advanced teams who want deep control over the pathing logic.

- **CONS:**

* **Complexity:** Steeper learning curve than RoadRunner.

* **Tuning:** Can be difficult to tune perfectly.

* **Smaller Community:** Fewer resources and people to ask for help compared to RoadRunner.

* **Requires Odometry:** Strictly requires good odometry pods (dead wheels).

---

2. Which One Should You Choose?

🏆 **Recommendation: RoadRunner 1.0**

- **Why?** It strikes the perfect balance between performance, ease of use, and support.
- **For most teams:** RoadRunner 1.0 is the standard. The "Actions" system makes writing autonomous programs much cleaner.
- **If you are a beginner:** Definitely RoadRunner 1.0.
- **If you are an expert wanting to push limits:** You might experiment with Pedro Pathing for raw speed, but be prepared for more work.
- **Avoid RoadRunner 0.5** unless you are maintaining a very old codebase.

3. Successful Team Repositories (Into The Deep & Recent)

Here are repositories from high-performing teams to study. Note that many top teams keep their current season code private until after the season, so some links may be from recent past seasons or early releases.

- **Team 18840 Reynolds Reybots (Into The Deep Code)**
  * GitHub Repository
  * *Notes:* Good reference for current season structure.
- **Team 13085 Bionic Dutch (Into The Deep Experimental)**
  * GitHub Repository
  * *Notes:* Experimental code for the current season.
- **Up-A-Creek Robotics (Team 11260)**
  * GitHub Profile
  * *Notes:* Legendary team (World Champions). Look at their `ftc_app` repositories from previous seasons for high-level architecture examples.
- **BrainSTEM Robotics (Team 19746)**
  * GitHub Organization
  * *Notes:* World Championship winning alliance members. Excellent code structure references.
- **FtcRobotController (Official SDK)**
  GitHub Repository
  Notes: The base SDK. Always good to reference for the latest official samples.

---

4. RoadRunner 1.0 Specific Resources

Since you are using RoadRunner 1.0, these are the most relevant resources for you:

- **Official RoadRunner Quickstart (1.0)**
  * GitHub Repository
  * *Notes:* This is the base project you are likely using. It contains the official tuning OpModes and examples.

- **Into The Deep RoadRunner Quickstart (Unofficial)**

  * GitHub Repository

  * *Notes:* A community-made quickstart specifically for the "Into The Deep" season, often with pre-configured settings for common robot chassis.

- **FTC 6282 Simi Valley Robotics Tutorials**

  * YouTube Channel

  * *Notes:* Excellent video tutorials specifically for RoadRunner 1.0, covering installation, tuning, and the new "Actions" system.

- **Team Future 14473 (Quickstart Fork)**

  * GitHub Repository

  * *Notes:* An example of a team actively using the 1.0 quickstart.