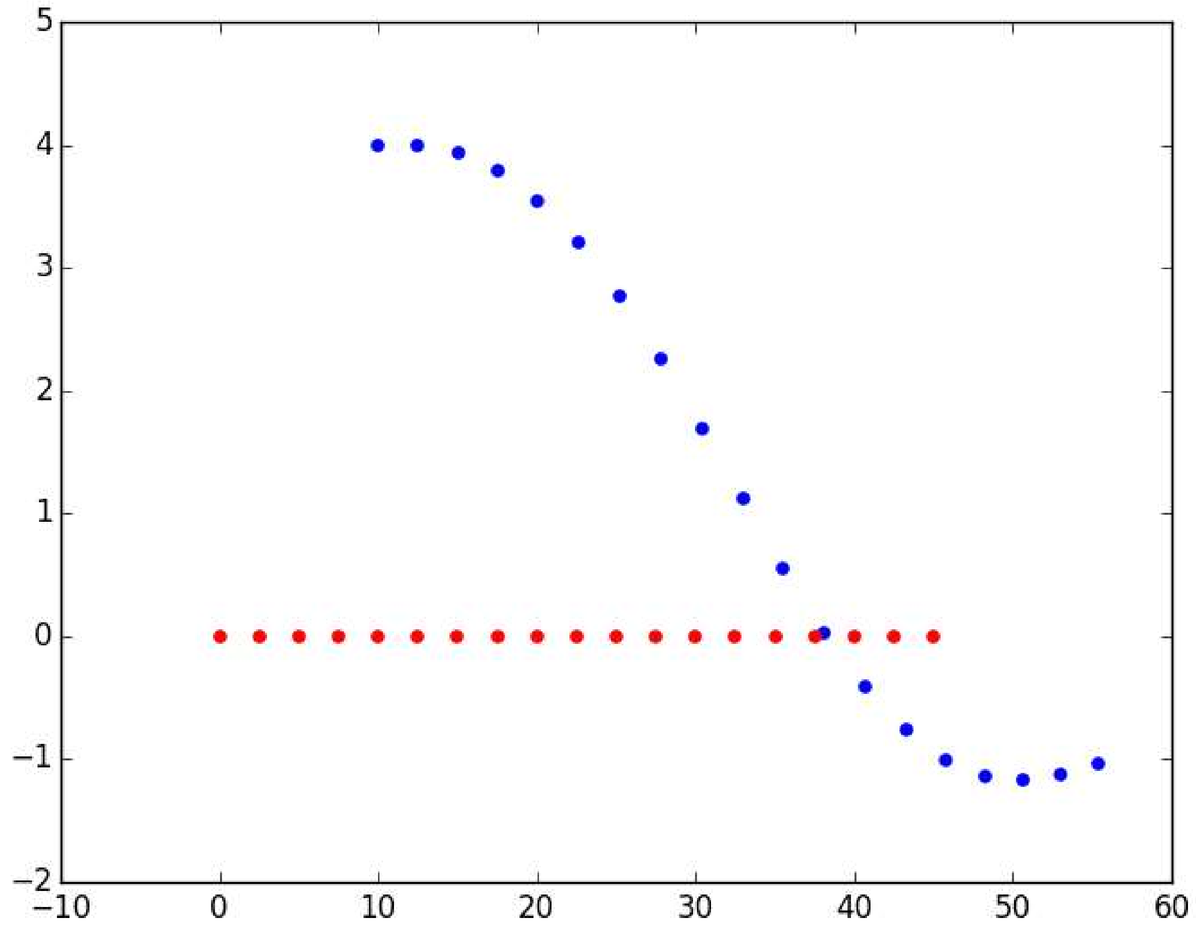
Mentor Help
Ask a mentor on our Q&A platform

Peer Chat
Chat with peers and alumni

Getting Started

1. Download the project code by clicking **TrajectoryExercise2** at the bottom of this page.
2. From the project's directory, run `python evaluate_ptg.py`. You should see a plot similar to the one below. This plot shows the s (x-axis) and d(y-axis) trajectories followed by a vehicle in traffic (red) and a self driving car (blue).



In this situation, the self driving car was trying to get behind the target vehicle, but the cost functions it was using weren't weighted appropriately and so it didn't behave as expected.

Fixing the Problem(s)

There are 5 files in the provided code. You'll probably want to start by modifying cost function weights in `ptg.py` but may also want to add cost functions of your own. As a bonus challenge try implementing this same code in C++.

File Descriptions

1. `ptg.py` - The primary code for generating a polynomial trajectory for some constraints. This is also where weights are assigned to cost functions. Adjusting these weights (and possibly adding new cost functions), can have a big effect on vehicle behavior.
2. `cost_functions.py` - This file contains many cost functions which are used in ptg.py when selecting the best trajectory. Some cost functions aren't yet implemented...
3. `evaluate_ptg.py` - This file sets a start state, goal, and traffic conditions and runs the PTG code. Feel free to modify the goal, add traffic, etc... to test your vehicle's trajectory generation ability.
4. `constants.py` - constants like speed limit, vehicle size, etc...
5. `helpers.py` - helper functions used by other files.

Supporting Materials

- Trajectoryexercise2
- Trajectoryexercise2_python3

NEXT