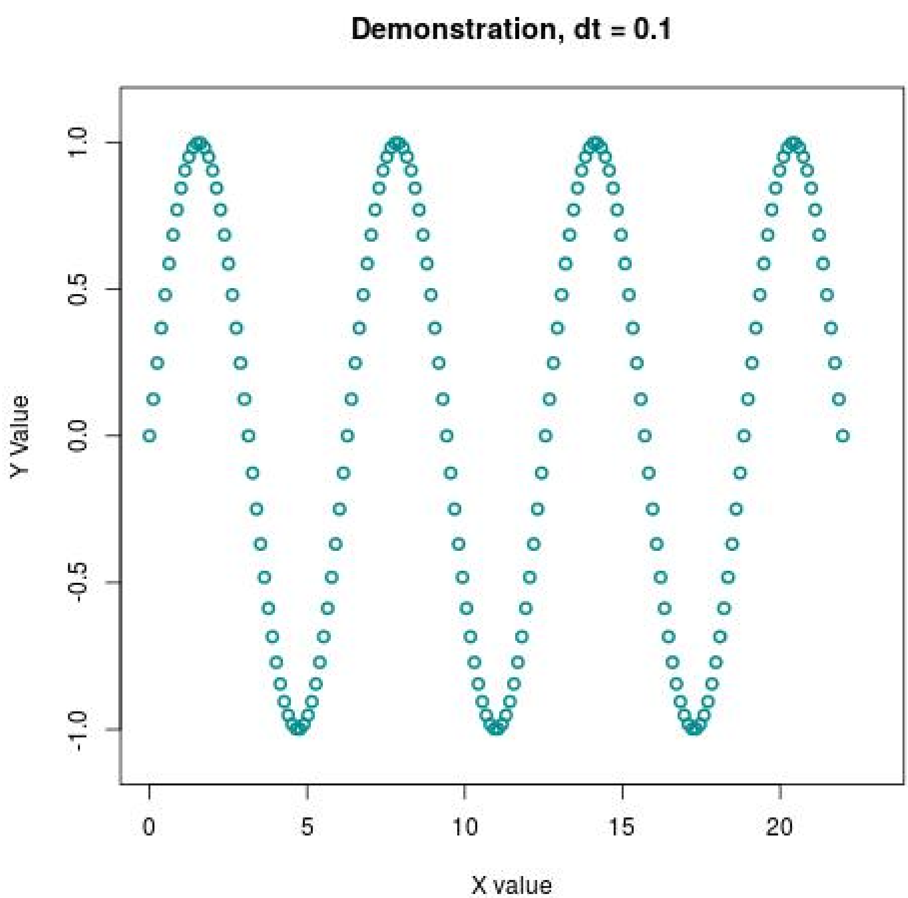
Robot Learning : DMPs

Prabhat Nagarajan

September 2016

Part 1

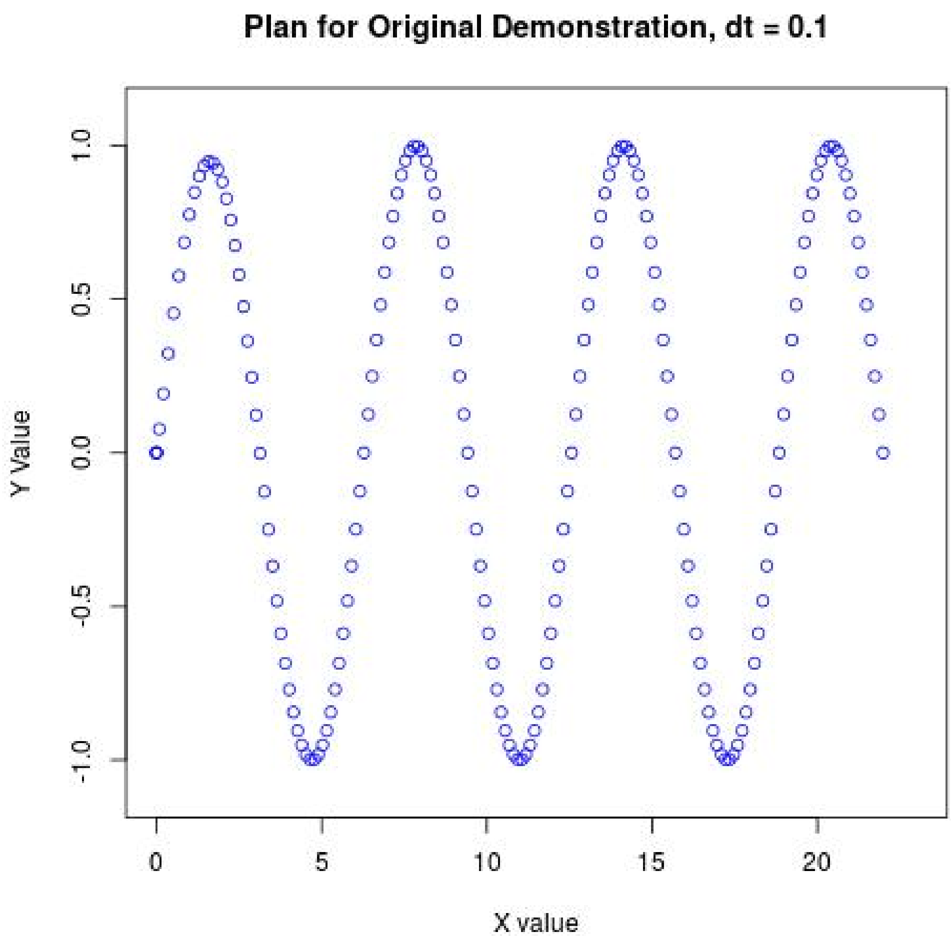
Figure 1: Demonstration



Part 2

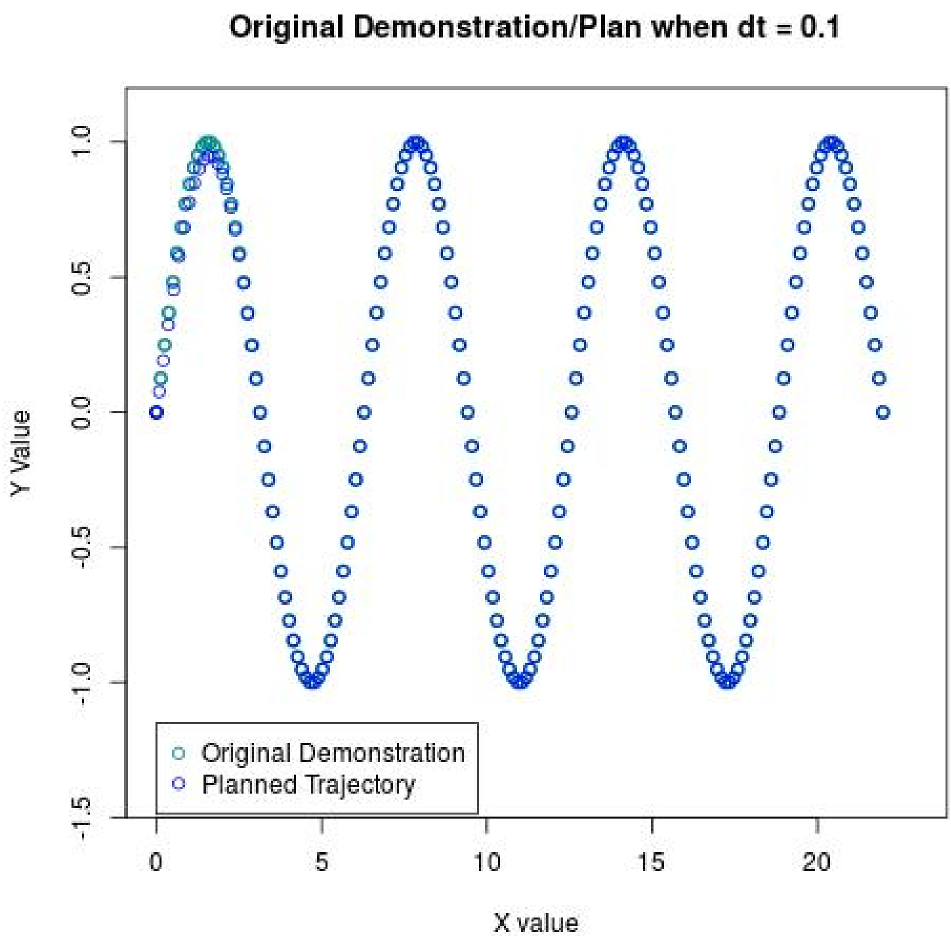
We first show the plan alone.

Figure 2: Plan for Original Demonstration



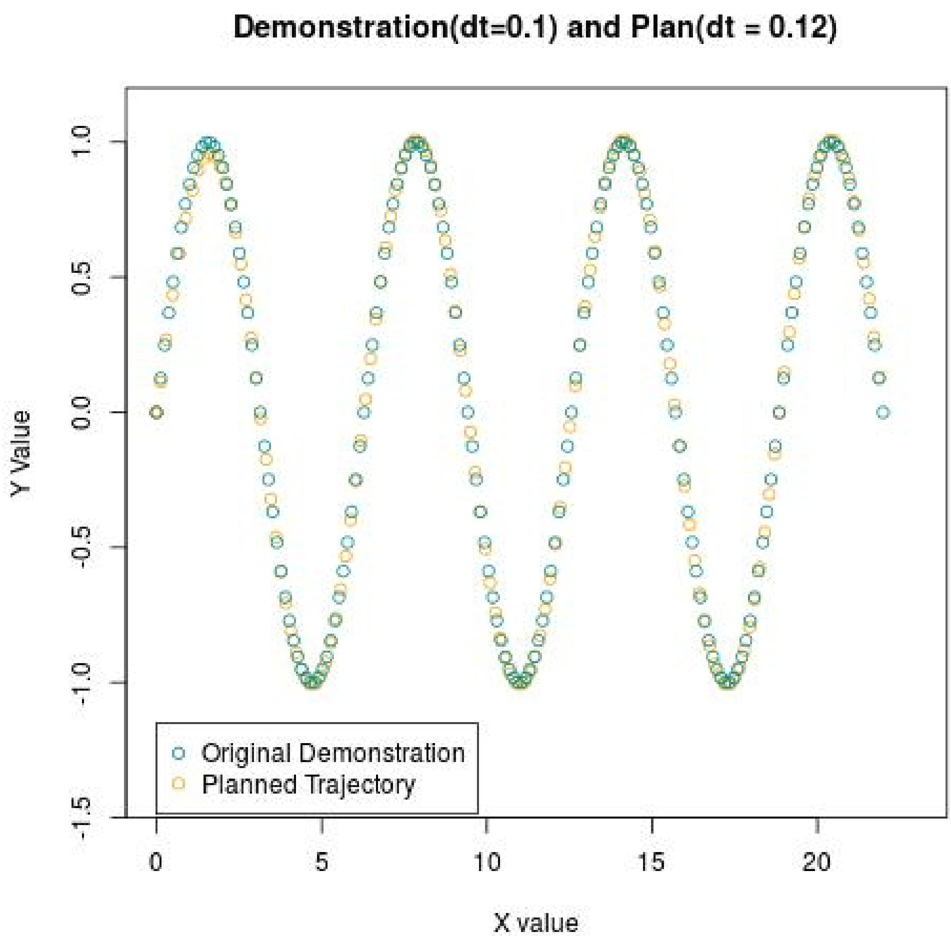
We now plot both the plan and the original demonstration, both with $dt = 0.1$.

Figure 3: Plot of Both Plan and Original Demonstration



As one can see, these are highly overlapping plots. To show the similarity more clearly, we plan the trajectory at time steps of 0.12, versus the original demonstration that had a time step 0.1.

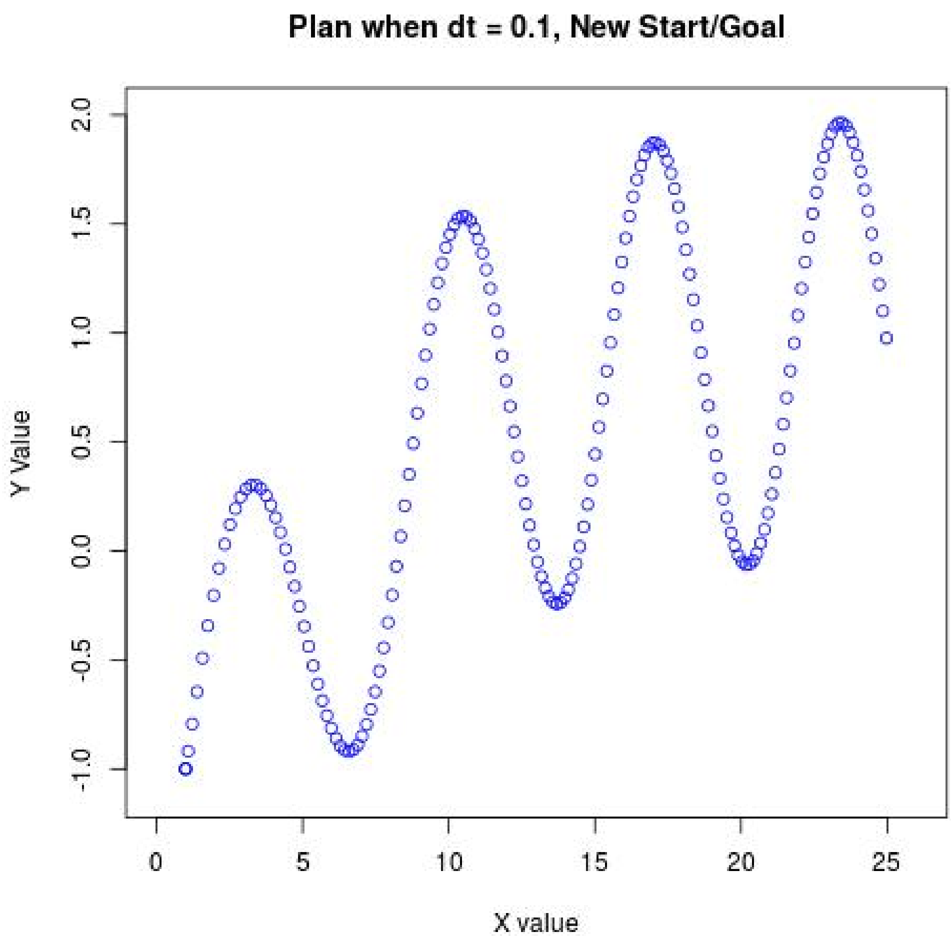
Figure 4: Demonstration and Plan



Part 3

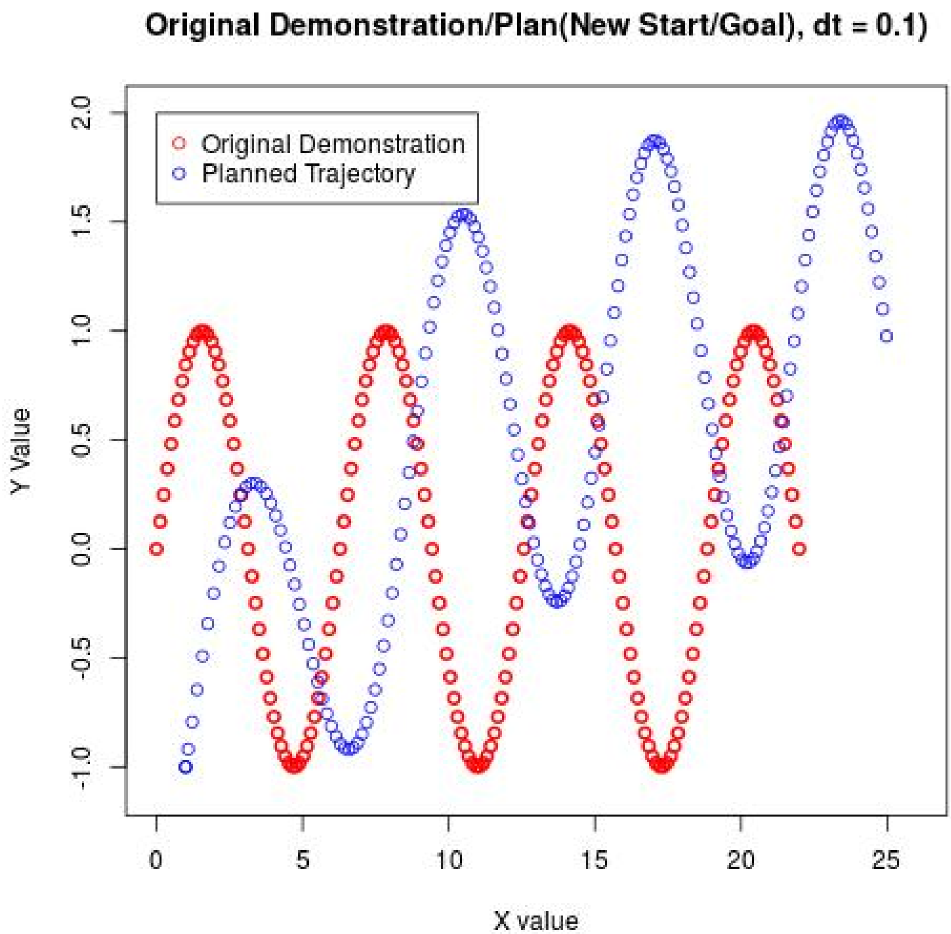
We assume that when the instructions ask for "the same with a significantly different start and goal", that we plan a new trajectory with a different start and goal.
The following plot plots the planned trajectory with the new start and goal. We use a start of $(1, -1)$ and a goal of $(25, 1)$

Figure 5: Plan with a New Goal



We now plot the new planned trajectory(with a new start and goal) along with the original demonstration.

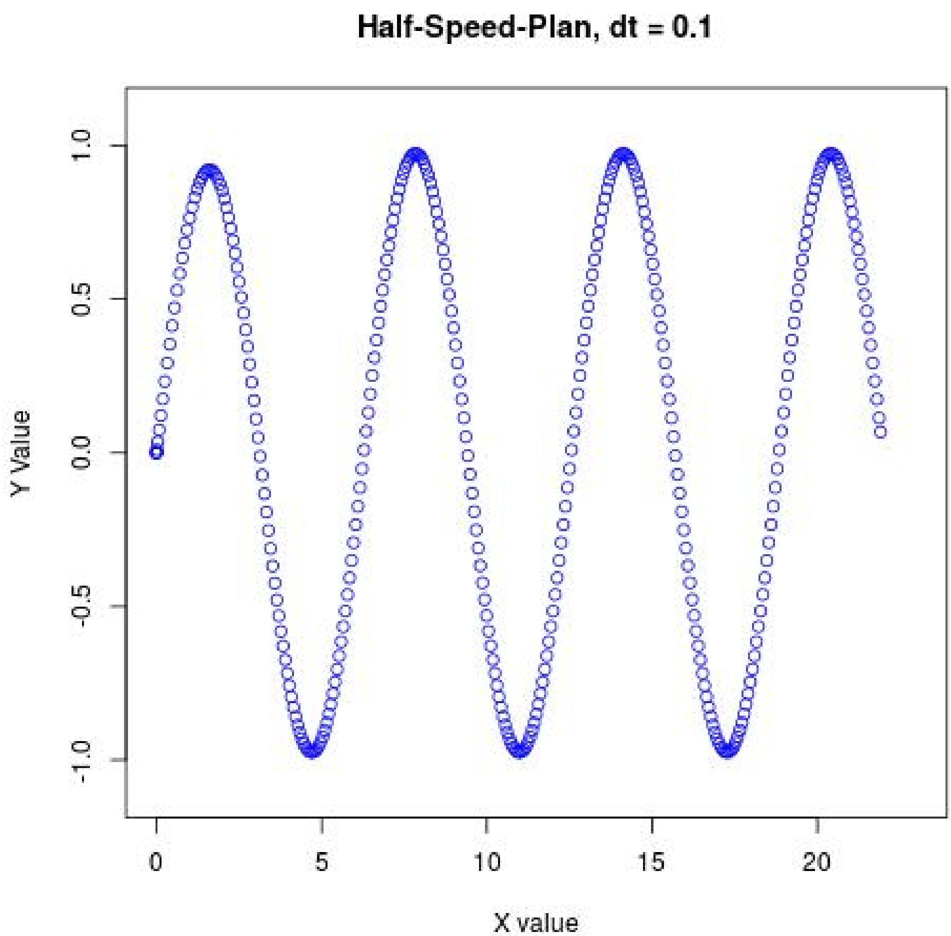
Figure 6: Demonstration and Plan(with New Start/Goal)



Part 4

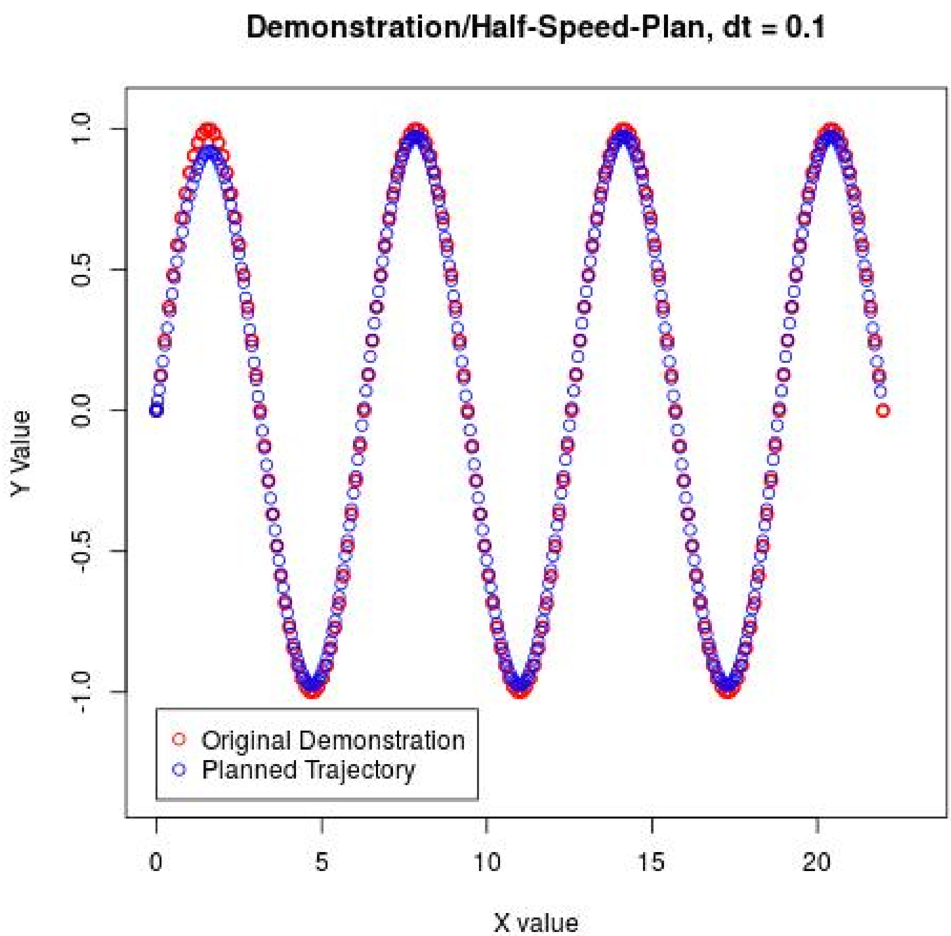
For this part, we assume that we are to plot the half-speed and 2x-speed trajectories alongside the initial demonstration.
The following plot shows the half speed trajectory.

Figure 7: Half Speed Plan



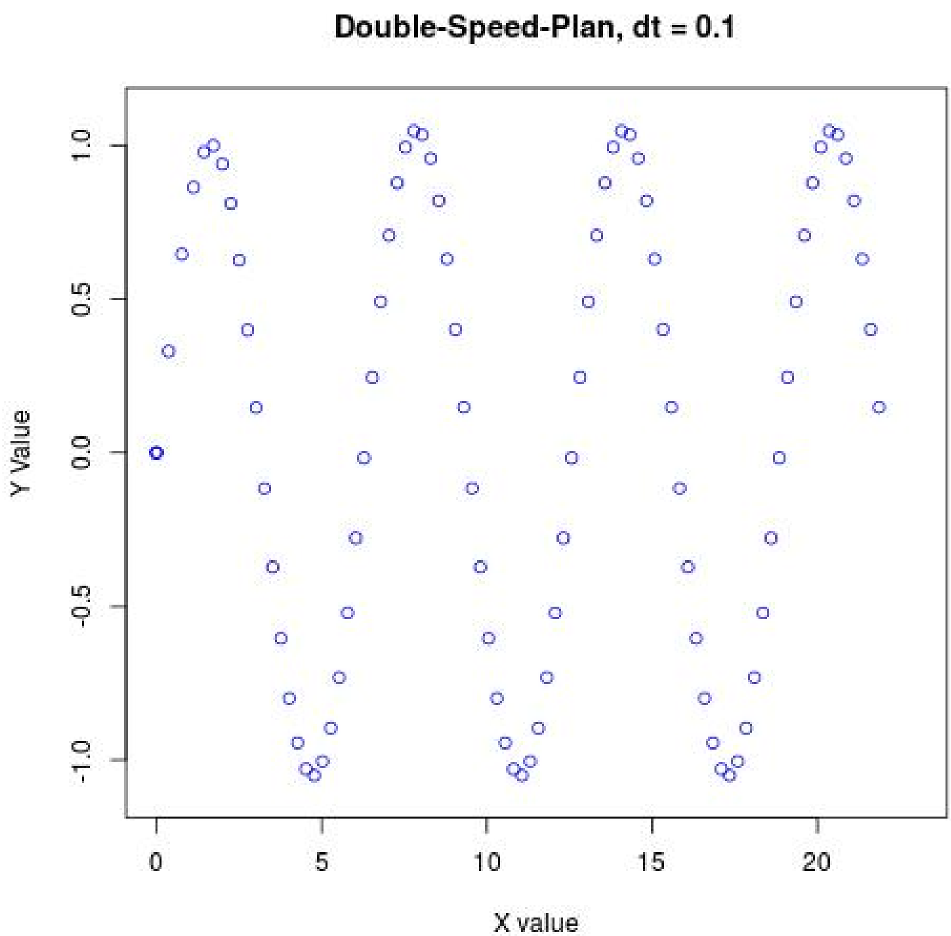
The following plot plots the half-speed trajectory and the original demonstration.

Figure 8: Half Speed Plan Along with Original Demonstration



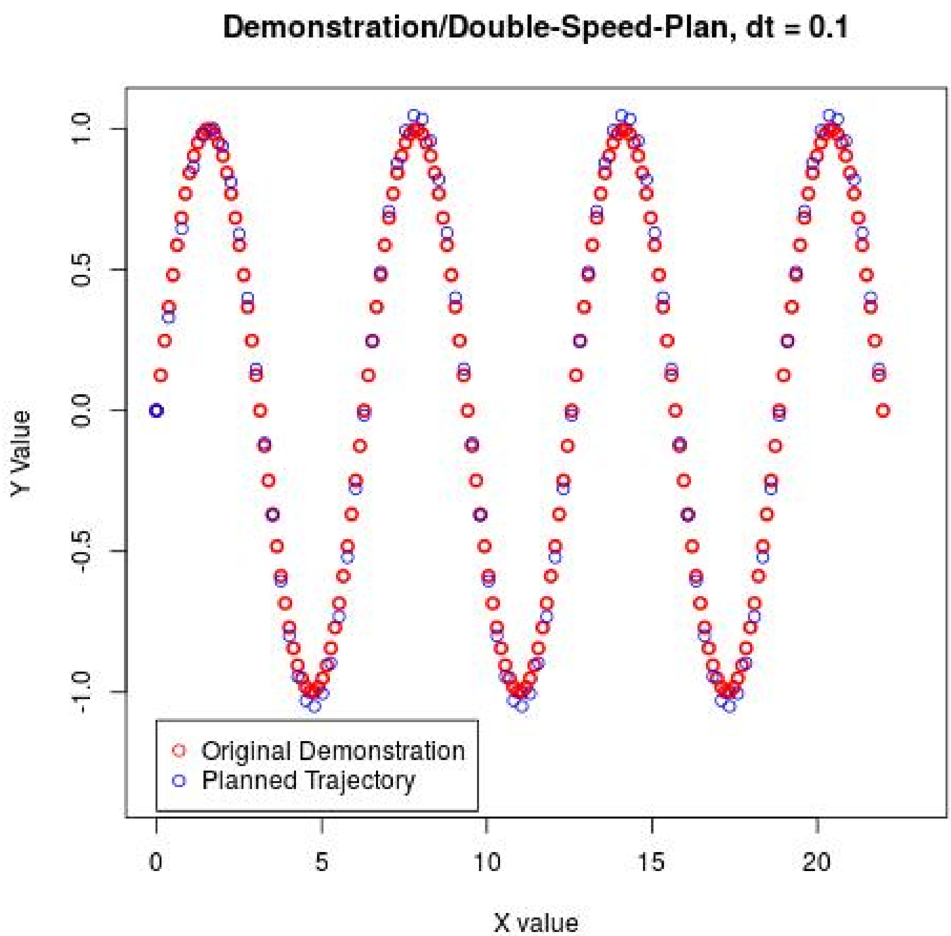
As one can see from the above plot, the blue dots are much more dense in the graph. This is because there are twice as many points in the trajectory, because the trajectory takes twice as long.
The following plot plots the double-speed trajectory.

Figure 9: Double Speed Plan



The following plot plots the double-speed trajectory and the original demonstration

Figure 10: Double Speed Plan Along with Original Demonstration



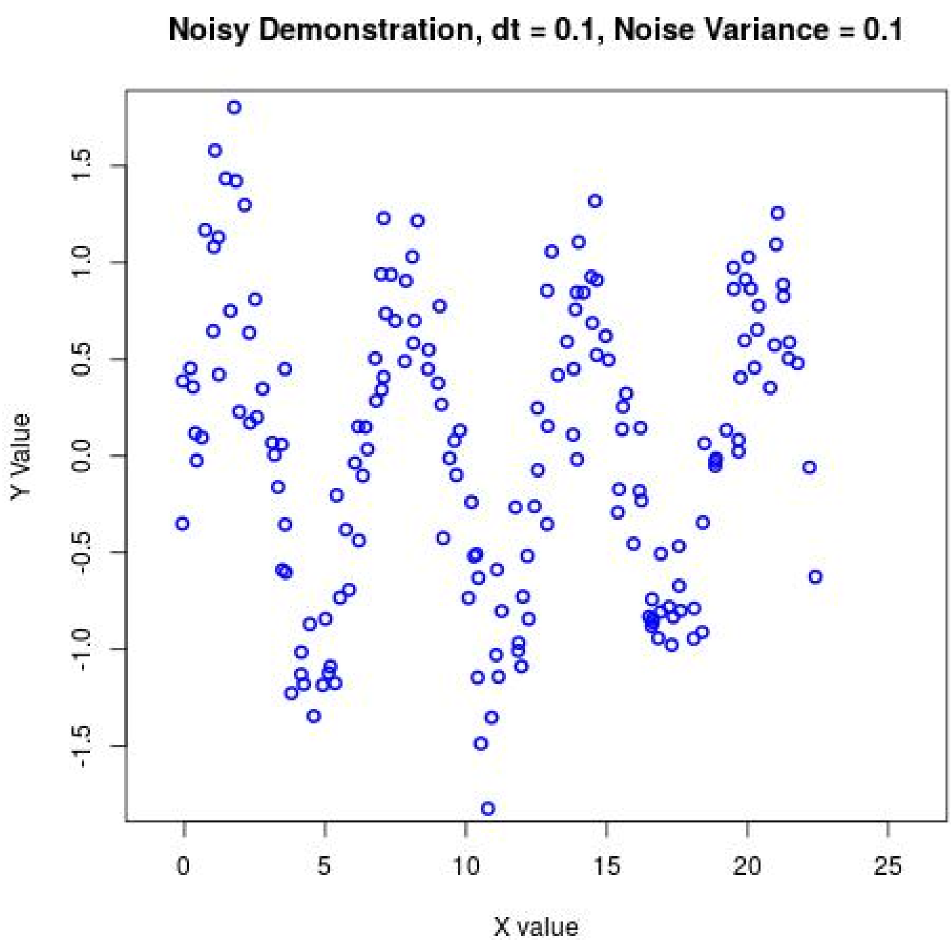
In this case, the planned trajectory is the blue colored dots, and as one can see, it is much more spaced out than the demonstration(red). This is because the trajectory is happening in half of the time that the demonstration takes, and as a result it has half as many points.

Describe how you did this: To change the speed of the trajectory, we simply change the τ used in planning. To plan for double the speed of the demonstration, we set τ to be equal to half the time taken for the original demonstration. To plan for half the speed of the demonstration, we set τ to be equal to double the time taken for the original demonstration.

Part 5

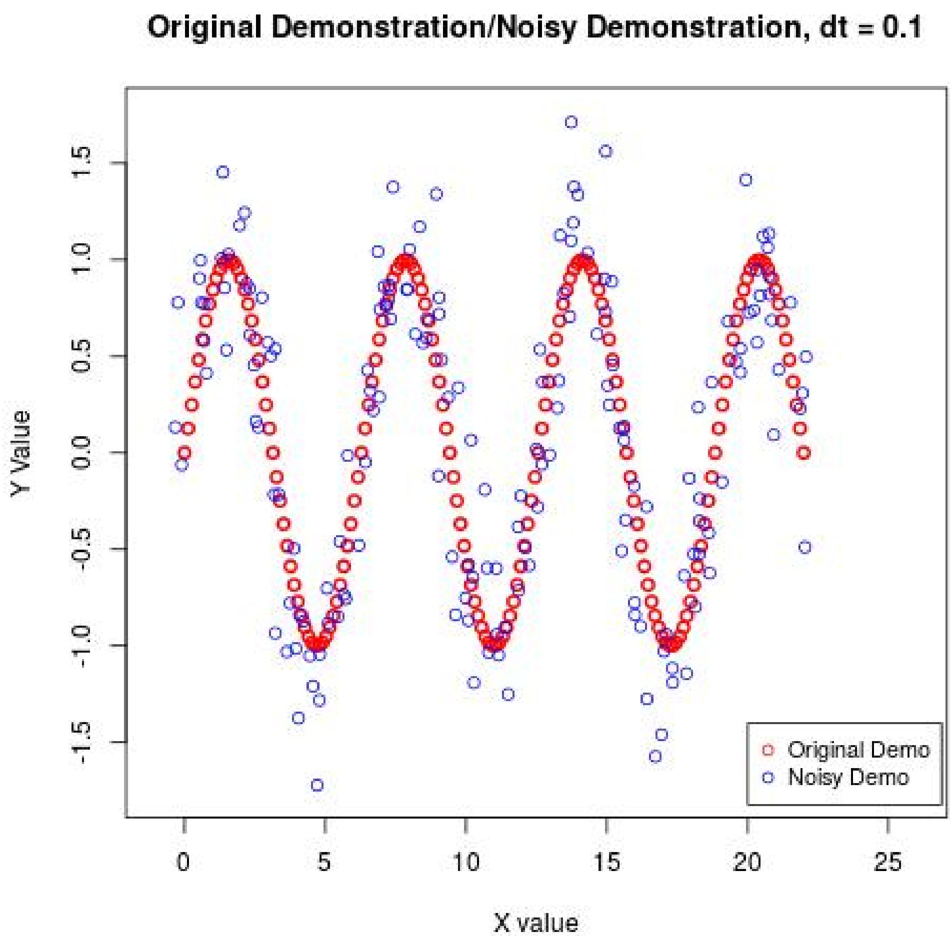
The following plot plots the original demonstration with 0-mean Gaussian noise, with a variance of 0.1.

Figure 11: Demonstration with Gaussian Noise



The following plot plots both the Noisy Demonstration and the Original Demonstration.

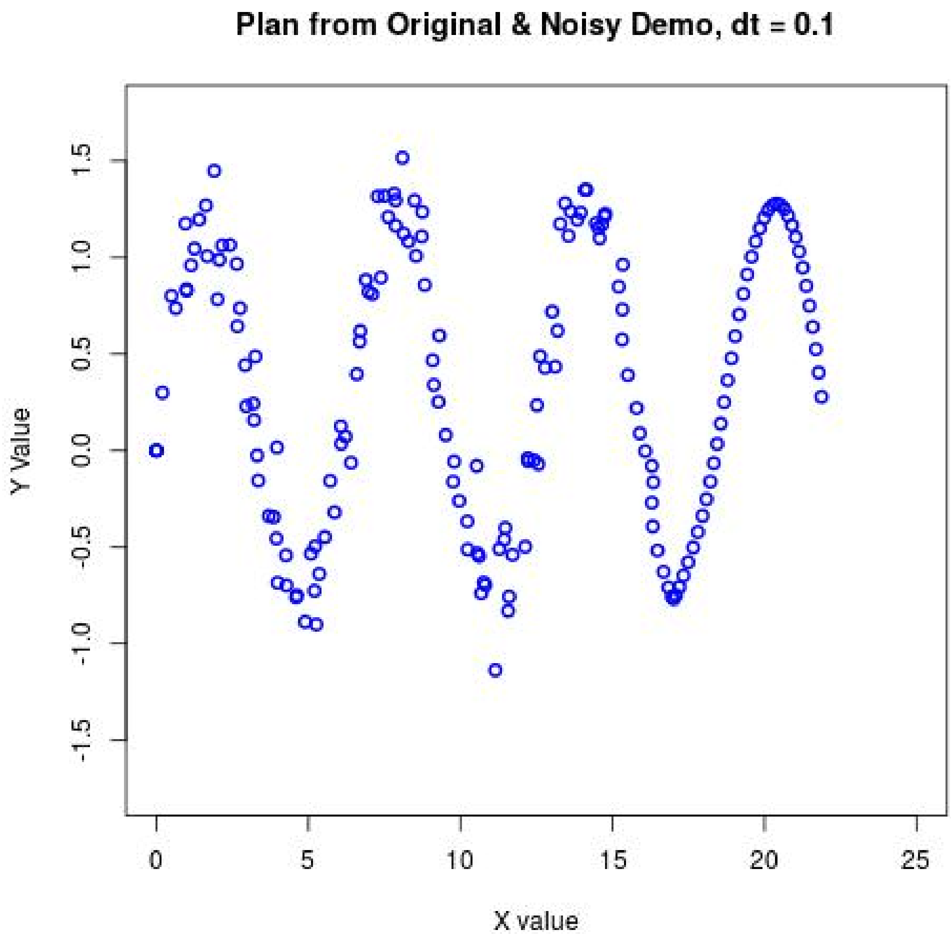
Figure 12: Original Demonstration Along with Noisy Demonstration



Part 6

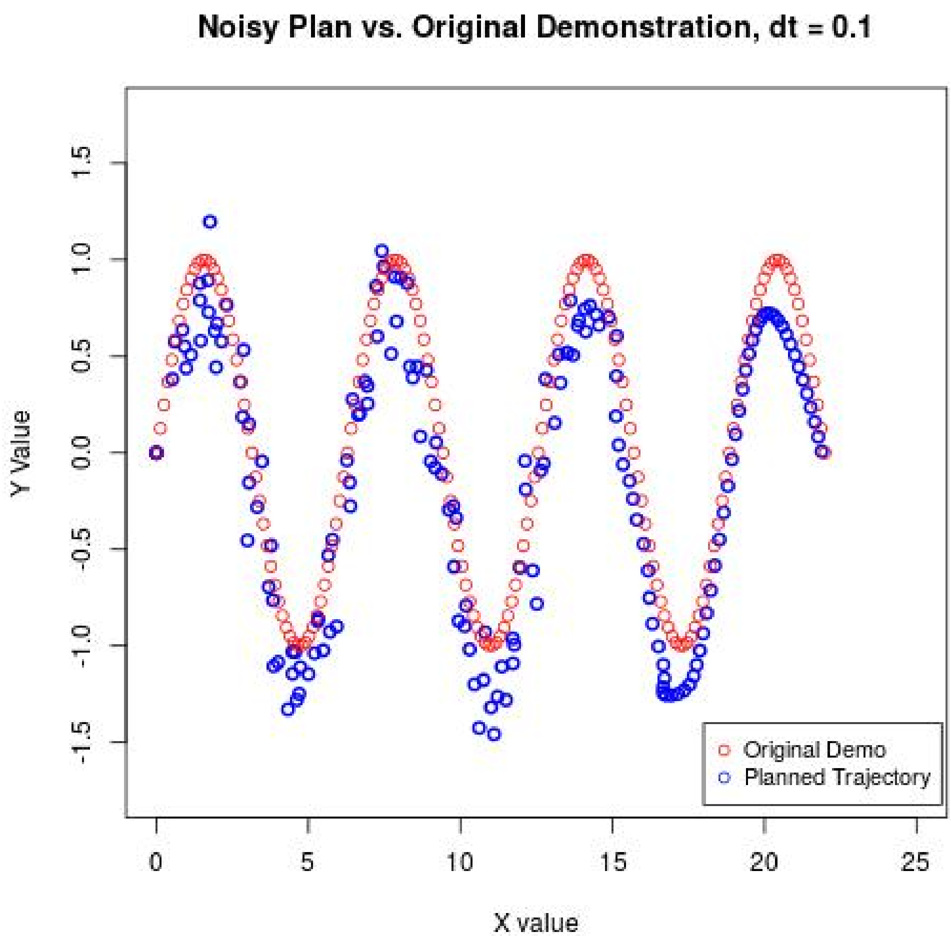
The following plot plots the planned trajectory when using both the original demonstration and the noisy demonstration for learning.

Figure 13: Planned Trajectory using Original Demonstration and Noisy Demonstration



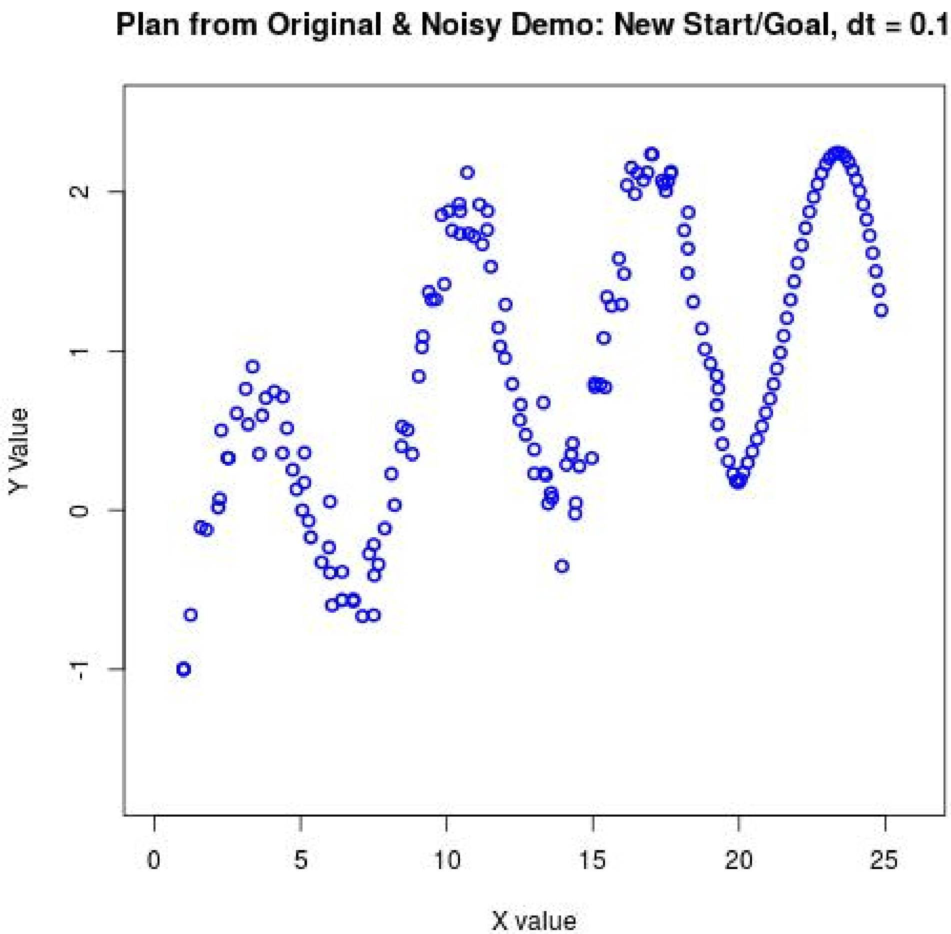
The following plot plots the planned trajectory along with the original demonstration. Note the similarity between the two curves, despite the fact that the DMP parameters were learned with the noisy demonstration as well.

Figure 14: Planned Trajectory vs. Original Demonstration



The following trajectory is the planned trajectory with a new start of $(1, -1)$ and a new goal of $(25, 1)$.

Figure 15: Planned Trajectory using Original Demonstration and Noisy Demonstration with New Goal



Final Parameters

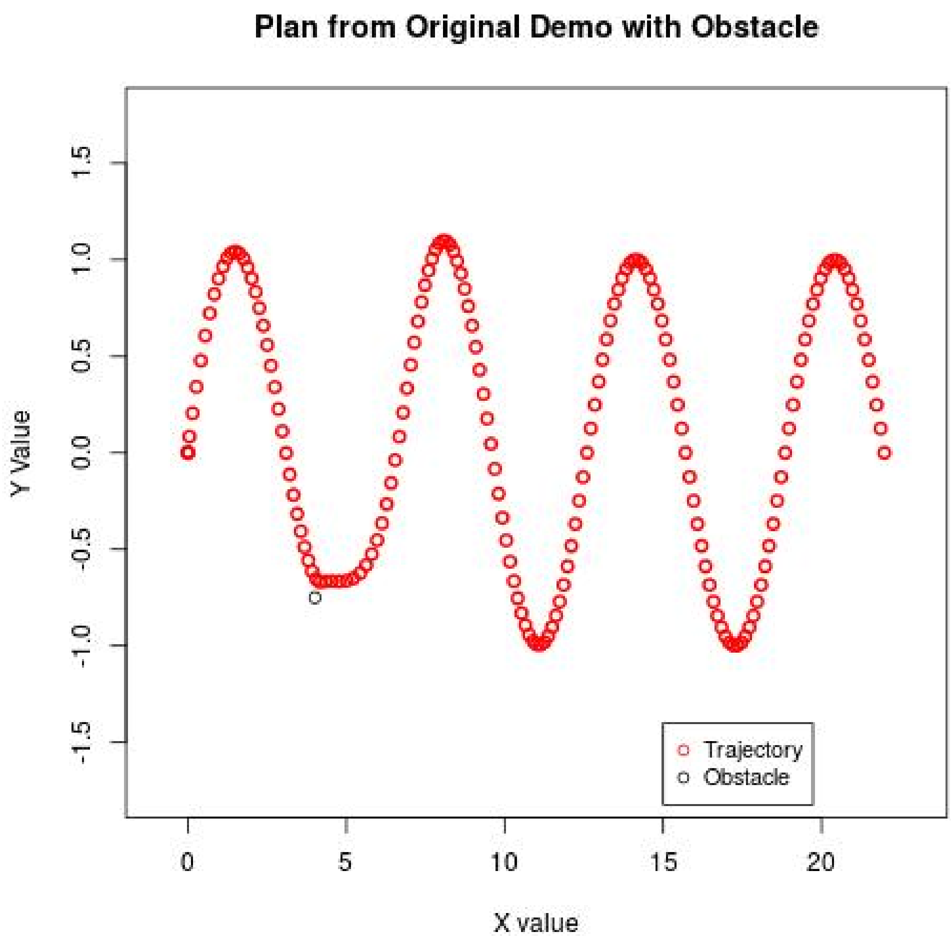
I ultimately chose the number of basis functions to be equal to the number of s data points we have gathered. The general idea is that one should have more basis functions centered at lower values of s, since there are more time (and as such, data points) is spent at lower values of s. For each s, we have a basis function centered at that value of s. By choosing the number of basis functions to equal the number of s's we observe, and by defining them this way, we satisfy this general requirement.
To choose the h values, the widths, we base it off the idea that basis functions

that are close together can have small widths, since for any value of s in a dense area, it can be expressed more accurately at a linear combination of the basis functions near that point. Since s decreases exponentially with time, we have many basis functions for lower values of s, and fewer for higher values of s. From this, we increase the width exponentially as s increases. We start with $h = 0.003$ (we found this through experimentation to be a good choice). Then, we go through each value of s in sorted order, and multiply h by 1.3, and as such h grows exponentially in s.
We plot the planned trajectory alongside the original (not-noisy) demonstration, as shown above. As one can see, the plan is quite similar to the demonstration. Additionally, when we change the the start to be $(1, -1)$ and the goal to be $(25, 1)$, we achieve good performance as well, and the generalization still works.

Part 7

We assume that when the instructions say to plot two graphs, we may plot graphs for two different obstacles.
The following plot shows the obstacle along with the avoidance trajectory.

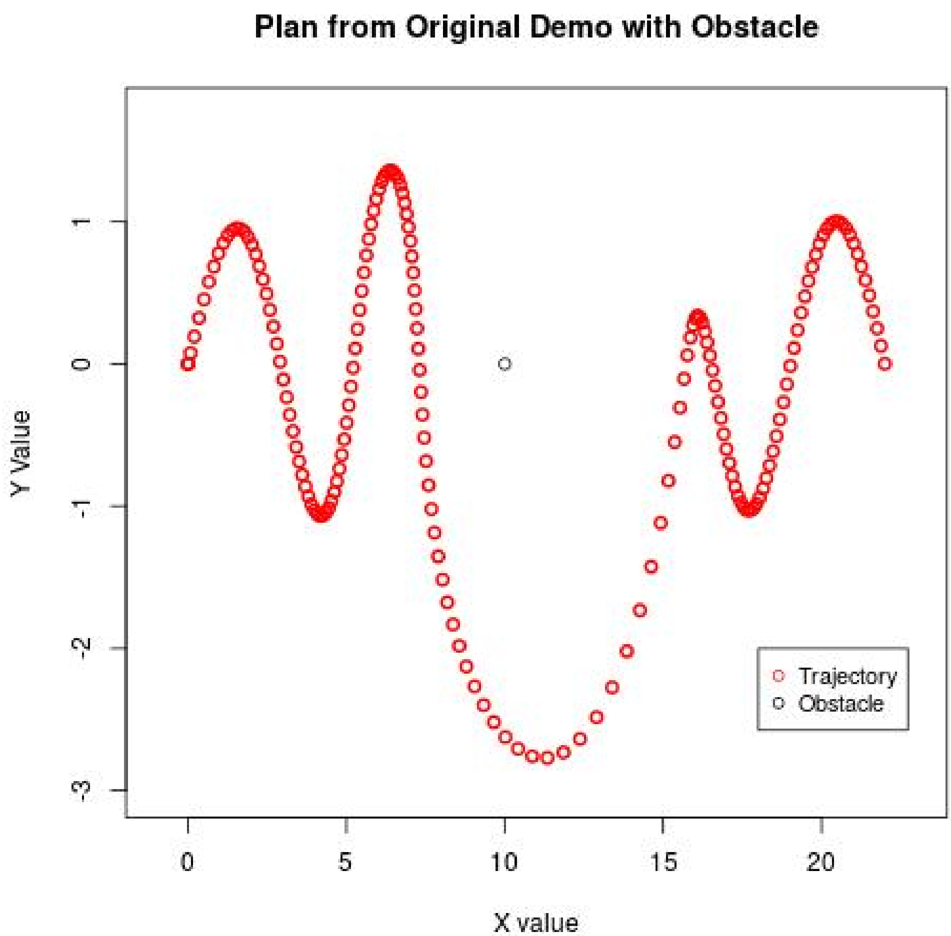
Figure 16: Obstacle Avoidance



As one can see, this obstacle is directly in the path of the trajectory. The trajectory was one track to go down to -1, but the obstacle causes an upward trajectory, after which the normal trajectory resumes, as the influence of the obstacle is no longer negligible.

The following plot, shows an obstacle that exerts an acceleration 10 times that of the obstacle in the previous plot. We show this to contrast how the plot above is able to resume it's trajectory in a normal fashion, whereas the following plot greatly changes the trajectory.

Figure 17: Obstacle Avoidance



This obstacle exerts a much larger force, and its radius of strong influence is much larger that the obstacle in the previous plot. As a result,this causes the trajectory to veer farther down, before coming back up, outside the strong influence of the obstacle.

Question 1

Our DMP generalized well for both the linear interpolation approximator and for the Gaussian Basis Function approximator. It seemed for the demonstration to be able to generalize to new goals (and even perform reasonably for the original goals), we required a sufficiently high K. Further, we ensured that our basis functions were set so that $f(s)$ was a reasonable function approximation. Has we had incorrect parameter values for the basis functions, the results both generalized poorly to new goals, as well as performed worse in general. However, in our search, we did not explicitly find any particular parameters that succeeded in planning with the original start and goal while failing at other start and goal locations.

Question 2

To compute trajectories on the fly, we would plan new trajectories every few time steps. Potential perturbations that the robot could face include, for instance, an obstacle. We have found that when facing obstacles, higher values of K cause the least perturbation in the trajectory. This makes sense in equation (9) of the paper, all terms involve K or D, save the coupling term. Thus the higher K is, the lesser the affect of the obstacle on the trajectory. This extends on to perturbations in general.

Question 3

We believe that the parameters to perturb are the widths of the Gaussian basis functions. Other components of system are not as volatile and are simple enough to set by a human. However, setting the widths of the basis functions requires much intuition, and hand-tuning, and even after that it is not perfect. These are parameters that must be near perfect for optimal trajectory planning. Policy search could, in the long run, converge to the optimal values of the widths. Furthermore, if we wanted to create a cost or reward function, we could measure the estimated $f(s)$ and compare it against $f_{target}(s)$ and use this comparison (e.g. squared error) as the cost.