Abstract

Throwing an object is a simple task in humans, often requiring only one or two examples to begin throwing an object efficiently, and yet in robots the task is much harder. In this experiment the goal was to determine an efficient policy for throwing paper airplanes in robots via the use of Baxter.

1 Introduction

The problem that at hand was to teach Baxter how to throw a paper airplane. Throwing paper airplanes is different from ordinary everyday throwing because of the different parameters involved (such as the shape of the plane, the weight of the plane, and so forth), and this made it a perfect candidate for a self learner. The goal for the semester was to try to teach Baxter how to learn a basic policy, and then to continually self improve on it by taking in inputs from the environment, and deciding which parameters to improve upon, essentially implementing Q-learning to solve the problem.

2 Methods, and Results

2.1 Q-learning and Deep Neural Networks

The initial idea was to read up Q-learning and learn deep learning for the project, which was much more complicated than intended. Most of the literature on the material is either too cursory, or too dense for a project such as this, targeting doctoral students. Furthermore, there are few tutorials about problems such as this. About a month and a half into the process (on top of learning how to use ROS and Baxter) I began to talk to Professor Akihiko and he suggested that I take a look at the breakdown that he had listed on his website

2.2 Professor Akihiko’s breakdown

Sub projects

- First, do not consider sophisticated tools. Just make a behavior, and find the problems.
- What are the difficulties? What are good (theoretical) tools to solve them?
- How to control the plane to throw as far as possible?
- How to control the plane to a target position?
- How to generalize your behavior to many types of paper planes?

This led to restarting the process, and starting from the beginning, first with considering different behaviors.
2.3 Manually searching the policy space

1) Modelling throwing after human throwing

Given the number of innovations that happen when we study nature and implement what happens, this was instinctively the obvious choice. The method in this case was to carefully study how the arm moves when throwing a paper airplane, and trying to code that into Baxter and then looking at the result and manually tweaking parameters that seemed wrong.

The main problem with this method are as follow:

1. Baxter doesn’t have any instinctive motor control, in that although we modelled it closely to human throwing, there are certain corrections that we as humans do when we throw planes, to allow the plane to fly further. The lack of such a control led to the plane often crashing straight into the ground due to the very high angular velocity from the throw.

2. The Angular velocity from the throw. Even after reducing the arc and trying methods to reduce the angular velocity we ended up with the problem of Baxter not being able to throw the paper airplane fast enough for it to fly. Reducing the arc by tweaking values in it led to problems with the arc ”jumping” in positions, and led to jerky movements at the high-point of the throw.

3. Timing issues. Tweaking the parameters to allow Baxter to get down the timing for the throw to coincide with what our predetermined optimal was very difficult because often-times it would require Baxter to release the gripper faster than it was capable of doing.

4. Inspiration, not rule. Looking back, to counter the problem of not having enough velocity after reducing the arc, we should have included the unused joint. We were too focused on modelling it after how we throw it to not see beyond using it as a starting point

2) Using the change in angles

This led to the next iteration of the project. To summarize, we obtained the ending position of the throw, and the change in the positions of the joints, $\Delta q$, and then we backpropogated the $\Delta q$ to find the optimal ”start” such that it wouldn’t face problems of arcing and jerkiness.

Problems faced:

1. Timing issues. Same as above

2. Not enough velocity. In theory, this method should have worked perfectly, but there were still problems with arcing. It essentially became an optimization on the parameters of ”speed of throw”, and ”amount of arc”, and although we tried many methods to find the optimal trade-off we still faced the same eventual problem of ROS either throwing the plane straight into the ground or not being able to release with enough force such that the plane would leave the gripper

3) Forward throw along the Z-axis

The next step was then to consider the case of what if we only threw the plane along one axis. If we used the $\Delta q$ method from earlier, and we tried incorporating all the joints Baxter might be able to launch the plane fast enough such that we overcome the problems from before.

We eventually reached the same problems as before.

4) Combining methods 2) and 3)

The main inspiration for this method was the fact that earlier on we couldn’t quite get any method to work fast enough because we either couldn’t incorporate all the joints in the throw so it wouldn’t be fast enough, or because it had too much of an angular velocity.

Essentially, what we did for this section was to allow Baxter to arc in the throw up to the highest point, and then to incorporate a forward throw method, moving only along the z-axis. By doing this, it should avoid the probe of arcing, and would allow the plane to move forward.

For the purpose of the analysis, we break down the movement into two parts: a, and b. Part a is the arc-ed throw up to the highest point, and part b is the z-axis throw.
The problems with this method are as follows:

1. Timing issues. Same as above
2. Jerky movements. The only way that we could think of to combine both movement a and b into one fluid movement was through the use of forward kinematics which is problematic in itself. We attempted to "bound" the forward kinematics to what we wanted by doing smaller and smaller time-steps of where the arm should be at each point, but unfortunately the system quickly grew complicated (to an un-generalizable point), and we were running out of time.

2.4 Innovating

At this point, Professor Akihiko suggested using a tool to help us accomplish the goal of throwing a paper airplane. After some back and forth we settled on using a slingshot which I constructed

![Figure 1: Slingshot, and a standard airplane we used](image1)

3 Results and discussion

We managed to get Baxter to throw the airplane!

4 What I learned

I learned that Machine Learning is still just a tool. You still have to think about how to solve a problem, and not rely solely on it.

I learned that sometimes I have to be creative in defining the problem and solving it. This was especially beneficial near the end when it seemed like there was no way to successfully throw the paper airplane using Baxter. This led to the use of multiple tools, eventually the slingshot.

I learned to not be ashamed to ask for help, even on a personal "project" like this. It took me a long time before I began to ask Professor Akihiko for his advice on the project, as well as to guide me in getting Baxter to throw the airplane.