Reinforcement learning for pole balancing based on haptic feedback

Hanjun Song
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195
hanjuns@cs.uw.edu

Abstract

Conventional pole balancing problem is getting feedback of tilting angle (and the angular velocity) of the pole and the position (and the velocity) of the balancer. Also, the pole is kinematically constrained to the hinge of the balancer on 2 dimensional surface. In this paper, model-free reinforcement learning is used to implement pole balancing controller based on contact force feedback in 3 dimensional space. Natural policy gradient algorithm is used to learn the control policy. Pole balancing can be thought as one kind of non-prehensile manipulation and this paper shows the importance of reinforcement learning and the haptic feedback in non-prehensile manipulation.

1 Introduction

1.1 Background

People use a lot of strategies to manipulate object. One common example is grasping objects and move around. However, most of the manipulation is non-prehensile manipulation, which means manipulation not involving grasping. Non-prehensile manipulation includes actions such as balancing, pushing, pulling, flipping, throwing, and so on. Non-prehensile manipulation is challenging in robotics because of many reasons. To mention a few of them, non-prehensile manipulation couples grasp planning and kinematic motion planning and increases uncertainty of feedback from sensors. For example, occlusion occurs due to the robot hand when the robot uses visual feedback from the camera on the its head. Because the robot is not grasping the object, the uncertainty increases more than when the robot is grasping the object. Therefore, haptic feedback is required to have more stable feedback than visual feedback.

Haptic feedback is the combination of kinaesthetic feedback and tactile feedback. For robotic manipulators, kinaesthetic feedback is obtained by forward dynamics based on the torques of each joints or force/torque sensor on the end-effector. Tactile feedback is acquired by sensors which are able to sense normal force, shear force, and vibration. For humans, the sense of touch is a key factor that enables manual dexterity for humans. Accordingly, there have been a lot of attempts to develop robust and multi-modal tactile sensors for robots and reliable results came out recently. For example, GelSight and FingerVision are camera-based tactile sensors and they give not only force and vibration data, but also proximity visual data. In addition to the tactile sensor, sensitive torque control became available for actuators. For instance, HEBI Robotics developed a series-elastic actuator with 0.01Nm torque resolution. In this paper, ATI Nano 25 force/torque sensor is considered as a kinaesthetic feedback due to the high performance of the sensor.

1.2 Statement of the Problem

Can robot balance a pole not based on the tilting angle, but based on the contact forces? What kind of reinforcement algorithm should be used to learn the control policy of this control problem? How can I transfer from simulation to physical system?

1.3 Objectives

The objectives of this research is to implement an algorithm for learning control policies of haptic-based pole balancing. This can be extended to non-prehensile problem as pole balancing is one kind of non-prehensile manipulation. The eventual goal of this research is to investigate the importance and possibility of haptic feedback in non-prehensile manipulation. Also, this research explores the way to transfer simulation to physical simulation by setting up the simulation environment as close as possible to the physical system.

2 Methodology

For the reinforcement algorithm, natural policy gradient algorithm is used to learn the control policy. Simulation environment with the similar parameters with the physical system is set up using MuJoCo. In MuJoCo simulation, a balancer (10cm x 10cm) is set to move on 2 dimensional surface and a pole (30cm x 1,2cm) is set to move in 3 dimensional space as a freejoint. On the balancer, 3 axis force sensor is added to sense the force at the contact between the balancer and the pole. The gear number of the motor in the simulation is adjusted from 7.0 to 3.0.

For the natural policy gradient algorithm, position, velocity, and acceleration of the balancer and the contact forces are observed as states. In addition to that, rewards are given as a sum of the rewards for tilting angle and distance from the center and the penalty on control cost. Action is given as a 2 dimensional position of the balancer. Basically, the natural policy gradient algorithm learns the mapping from the observed balancer states and the contact forces to the position of the balancer in order to balance the pole. For the convergence of the learning, learning parameters, such as the number of iterations and trajectories, should be adjusted depending on the physical parameters of the pole and the balancer.

States	Rewards	Actions
position, velocity, acceleration of the balancer, Contact forces	Tilting angle, Distance between the pole and the center, Penalty on control cost	Position of the balancer

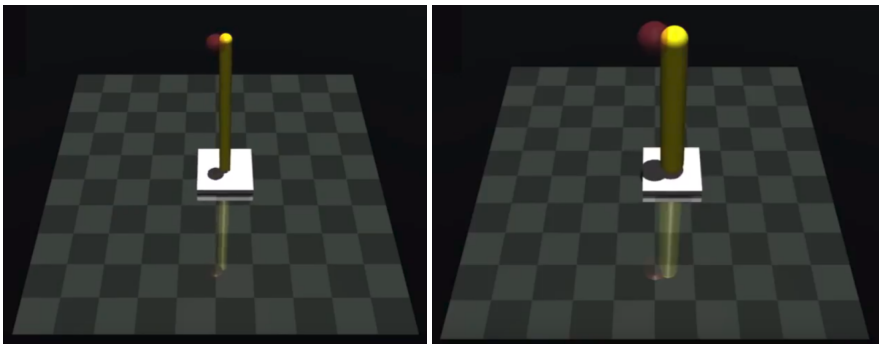
Table 1: States, Rewards, Actions of the algorithm

Eventually, the balancer should be able to balance the random objects with random inertia and shapes, but only capsule shapes of objects with different size and mass are considered in this paper to verify the feasibility of haptic based balancing. The adjusted parameters are the length, radius, and the mass of the capsule object in MuJoCo.

In order to transfer the simulation to a physical system, a 3DOF robotic arm with a force sensor are implemented. Also, the simulation model parameters are set as close as possible to the physical system. For example, the standard deviation of the force sensor in the simulation is set as 0.03N because the actual sensor in the physical system is 0.06N. Furthermore, the initial angle of the pole is uniformly distributed in a range of -10deg to 10deg as it is impossible for human to put the pole in a exact right angle.

3 Result

Learning parameters, such as the number of iterations, trajectories, and episodes are adjusted depending on the physical parameters of the pole and the balancer to make the learning converging. For the pole with 30cm length, 1,2cm radius, and 400g, the number of iterations is 600, the number of



(a) pole length:30cm, radius:1cm, mass:400g (b) pole length:30cm, radius:2cm, mass:400g

Figure 1: Simulation setup in MuJoCo simulation

(a) 3DOF arm for the actuation of the balancer (b) Installation of the force sensor

Figure 2: Physical system implementation

trajectories is 250, and the maximum episode steps are 800. The gear of the motor and the time step are adjusted as well. Simulation parameters are shown in the below table.

Simulation parameters	
Number of iterations	600
Number of trajectories	250
Maximum episode steps	800
Timestep	0.01
Skip	1

Table 2: Simulation parameters

Rewards also have coefficients to give different weights. For example, the pole with 1cm radius has 20 for the angle rewards, 3 for the distance rewards and 3e-1 for the control cost penalty. Those coefficients are found manually.

Even though there are some steady state error in the position, the balancer was able to balance the pole.

4 Discussion

There are several aspects to improve such as the way to find the optimal values of learning parameters and rewards coefficients. In this paper, those numbers are found manually and takes long time.

The physical system test is not performed in this paper due to the lack of time, this can be done in the near future. During the transfer from the simulation to the physical system, the latency of the system can be also an important factor to consider.