LEARNING FORCE CONTROL POLICIES FOR COMPLIANT MANIPULATION

By: Nikhil Aggarwal

1) INTRODUCTION: In-order to make really assistive robots, they should be able to do fine manipulation skills. Complex contact forces with environment need a controller that can handle forces in meaningful way. Rigid body dynamics can give robot motion parameters. But once robot is in touch with the environment than it requires resulting contact interactions. Reinforcement learning has been an approach to learn manipulation skills. But those approaches consider position that need to be controlled. These are potentially dangerous as no consideration for object position is given. In this paper we present a new approach in which we learn the forces and torque to be controlled at the end effector combined with kinematic demonstration. An initial knematic demonstration is given. But so doesn't contain information about force and torque. So we do reinforcement learning through trial and error. "The contributions of this paper are two-fold:
(1) we demonstrate that learning force control policies enables compliant execution of manipulation tasks with increased robustness as opposed to stiff position control, and
(2) we introduce a policy parameterization that uses finely discretized trajectories coupled with a cost function that ensures smoothness during exploration and learning."

2)POLICY IMROVEMENT WITH PATH INTEGRAL(PI^2): Policy improvement with path integral optimizes control parameters based on a cost function i.e given a parameterized function and a cost function depending on state we need to minimise cost of the path.

3) LEARNING FORCE FEEDBACK CONTROL: The policy is initialized using user provided kinematic demonstration. The PI^2 reinforcement learning algorithm is used to optimise policy and to achieve right profile of force/torque through trial and error. Some of the steps involved in improving the policy are:

- **Demonstration:** Force and torque are set to zero initially as during kinematic demonstration they can't be observed correctly. Robot is maximally compliant i.e. gives easily to the contact forces.
- **Cost Function:** A automated or user provided cost function must to assigned to improve policies.
- **Execution:** PI^2 reinforcement learning algorithm is model free reinforcement learning algorithm so it just optimises control parameters subjected to the cost function treating the intermediate controller as black box.
- **Rollout reuse:** We preserve some rollouts from previous iteration so that we keep learning from those. But it may be case that in actual rollout can't be generated again. So, in-order to overcome this effect we keep re-evaluating such rollouts.

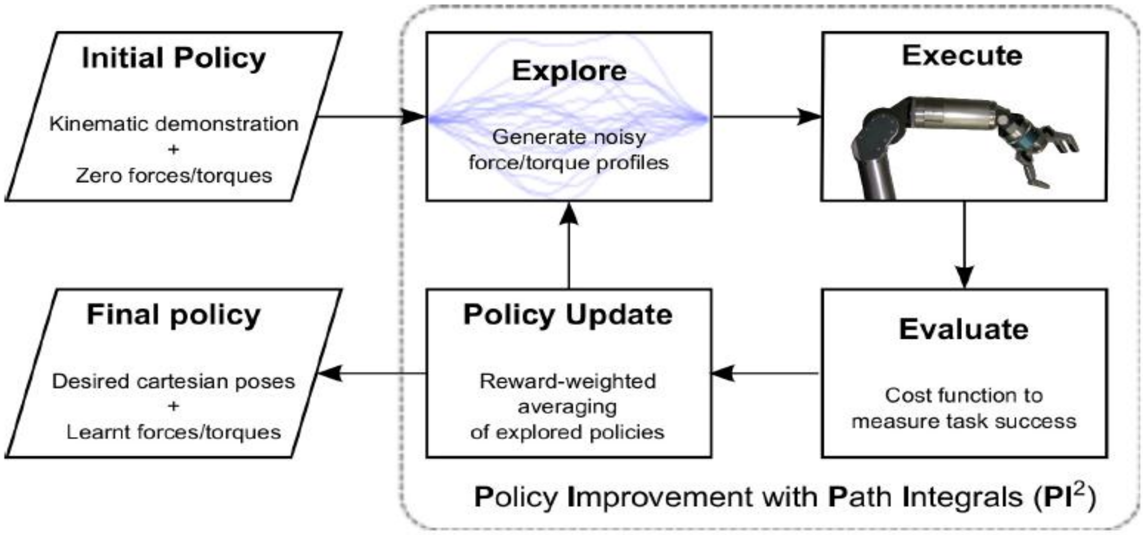

Fig. 3. A high-level overview of our approach to learning force control policies for manipulation.

4) <u>EXPERIMENT:</u> We tested our approach on two manipulation tasks. Both tasks were performed using a 7 degree of freedom(DOF) Barrett WAM arm, equipped with a three-fingered Barrett Hand and a 6-DOF force-torque sensor at the wrist. Our control law for 7 DOF arm is as following and as given in fig 4.

$$T_{arm} = T_{inv.dyn} + T_{joint} + T_{force}$$

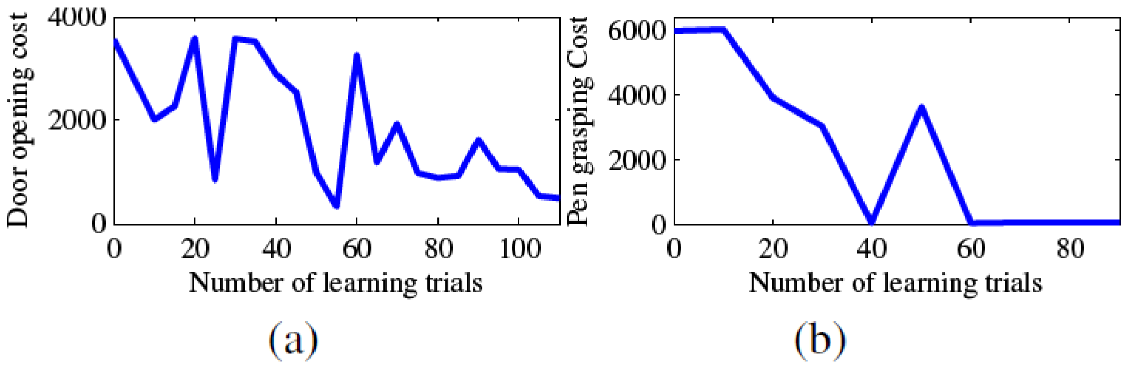

Fig. 6. Evolution of cost functions during learning for the two manipulation tasks: (a) door opening, and (b) pen grasping.

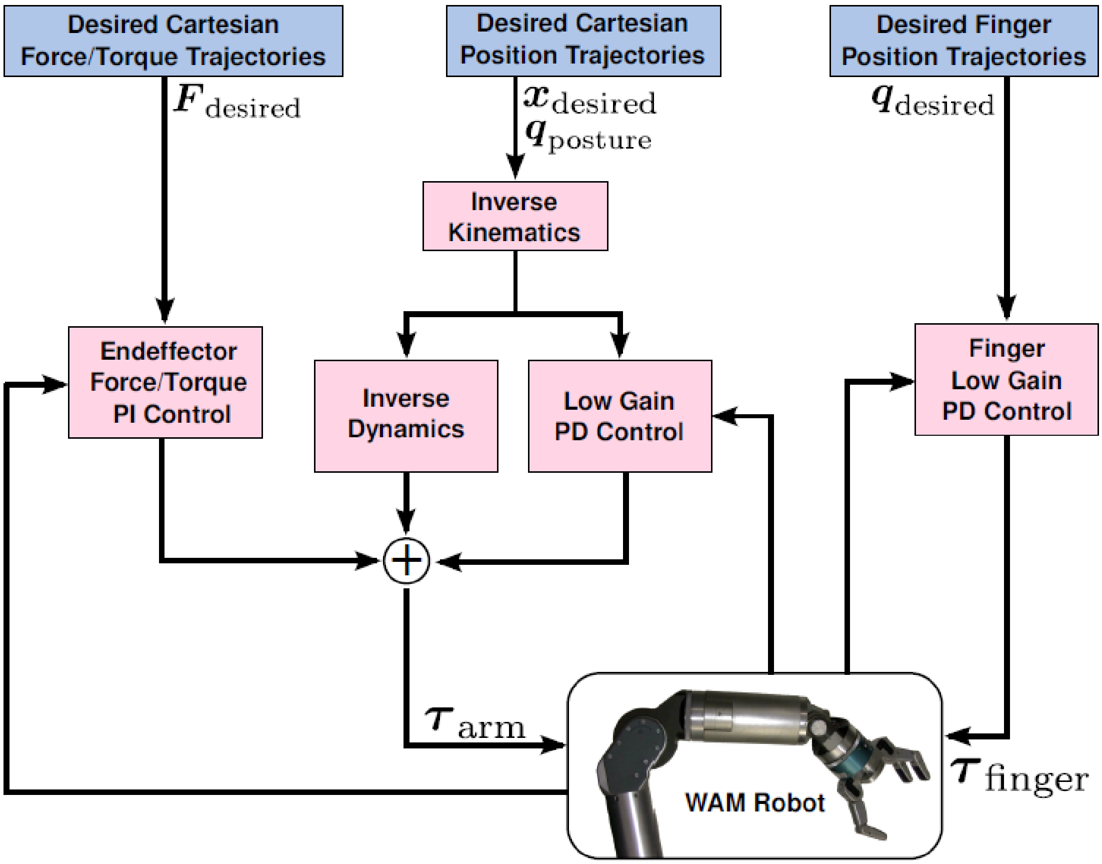

Fig. 4. Overview of the controllers used in our experiments.

- **Opening a Door:** The aim of the experiment is to learn a control policy for successfully operating a lever door handle and open the door. The trajectory was 10 second long and descritized into 100 steps. The cost function used was The immediate cost function at time t is:
  r_t = 300q_{door} + 100q_{handle} +100q_{pos} +10q_{orient} +0:1q_{fmag} +0:02q_{tmag} +0:02q_{ttrack} + 0:01q_{ftrack} + 0:0001X^TRX
  where q_{door} and q_{handle} are the squared tracking errors of the door and handle angles respectively, q_{pos} and q_{orient} are the squared tracking errors of the position and orientation of the hand, q_{fmag} and q_{tmag} are the squared magnitudes of the desired forces and torques, q_{ftrack} and q_{ttrack} are the squared force and torque tracking errors, and X^TRX is the control cost.
  After 110 trial we achieved policy that achieved 100% success.

Grasping a pen: The task is to pick a pen kept on table. More tougher as pen might slip from hand.
The immediate cost function at time t is:
r_t = 100q_{pen}+1:0q_{ftrack}+0:5$q_{fingertrack}$+0:1q_{fmag}+ 0:0001X^TRX,
where q_{pen} is an indicator cost which is 1 if the pen has slipped out of the hand (as described above), q_{ftrack} is the squared force tracking error, $q_{fingertrack}$ is the squared finger position tracking error, q_{fmag} is the squared force magnitude, and X^TRX is the control cost
After 90 trials we achieved policy with 100% success. Although uncertainty in orientation was not taken to consideration but it could handle some error in orientation as well.

(All figure are taken from the paper.)