Robotic Arm Contact Operation Task Using Reinforcement Learning Assistance

Chen Chen, Xu Dai and Jozsef Kovecses

June 5, 2024

Robotic Arm Contact Operation Task Using Reinforcement Learning Assistance

Chen Chen*, **Xu Dai, József Kövecses**

* Department of Mechanical Engineering
Centre for Intelligent Machines
McGill University
817 Sherbrooke St. West, QC H3A 0C3, Montreal, Canada
chen.chen12@mail.mcgill.ca
xu.dai@mail.mcgill.ca
jozsef.kovecses@mcgill.ca

Abstract

Robotic contact manipulations of objects are central elements in many applications. In such tasks, a robotic arm applies forces to an object at certain contact points to move the object to a desired position or along a trajectory. The aim of this work is to devise and implement an algorithm based on Reinforcement Learning (RL) to move a rectangular object to a desired configuration along a designed trajectory by employing a three-link robotic arm. Traditional control methods are challenging to implement due to the complex geometry of the object and the unknown contact dynamics, particularly friction. Reinforcement Learning overcomes this by eliminating the need for sophisticated control gain tuning, allowing the robot to adapt to randomized scenarios. Residual policy learning [1] is utilized as a joint torque augmentation to ensure the end effector has the appropriate contact points and forces throughout the contact operation. The results indicate that residual policy learning significantly improves the accuracy of the object's translation and rotation during the pushing process.

Methodology

Pushing a rectangular object to a desired location along a specific trajectory by a three-link robotic arm can be modeled and simulated as a contact dynamics problem. By considering unilateral constraints between the end effector and the object, contact dynamics can be expressed by a linear complementarity problem (LCP) formulation:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^{+} \\ h\boldsymbol{\lambda}^{+} \end{bmatrix} + \begin{bmatrix} -\mathbf{M}\mathbf{v} + h\mathbf{c} - h\mathbf{Q}_a \\ h\dot{\mathbf{A}}\mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{w}^{+} \end{bmatrix} \tag{1}$$

$$\mathbf{0} \leq \mathbf{w}_n^{+} \perp \boldsymbol{\lambda}_n^{+} \geq \mathbf{0}$$

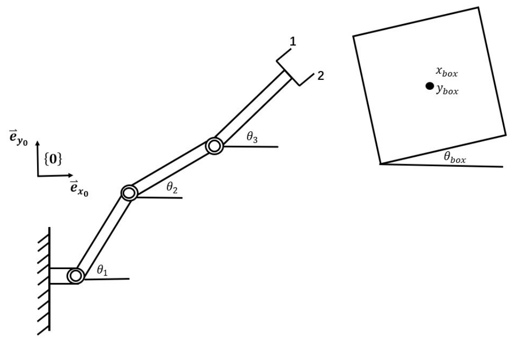
$$\mathbf{w}_t^{+} \perp \boldsymbol{\lambda}_t^{+} \in [-\boldsymbol{\mu}\boldsymbol{\lambda}_n^{+}, -\boldsymbol{\mu}\boldsymbol{\lambda}_n^{+}]$$

where $\mathbf{M}$ is the mass matrix of the system including the robotic arm and the object; $\mathbf{c}$ represents the Coriolis and centrifugal terms; h is the time step; $\mathbf{v}$ is the array of generalized velocities; $\mathbf{Q}_a$ represents the applied forces including the joint torques (τ_1, τ_2, τ_3); $\boldsymbol{\lambda}$ are contact forces composed of normal forces $\boldsymbol{\lambda}_n$ and friction forces $\boldsymbol{\lambda}_t$; $\mathbf{A}$ is the constraint Jacobian transforming generalized velocities into constraint velocities $\mathbf{w}$ which consists of normal $\mathbf{w}_n$ and tangential $\mathbf{w}_t$; $\boldsymbol{\mu}$ is the contact friction coefficient; superscript $+$ indicates the terms in next time step.

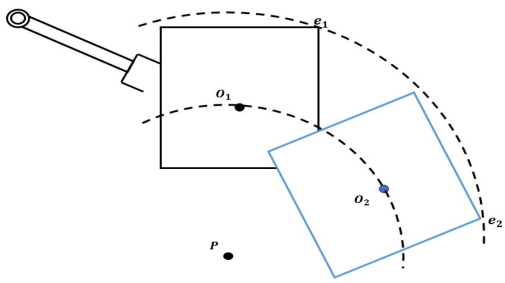
To attain the goal of displacing the object, our approach involves applying specific joint torques to the robotic arm. The initial policy joint torque, denoted as $\boldsymbol{\tau}_{ctl}$, is determined through end effector trajectory planning and position control. This planning aligns the end effector position with the desired position of applied forces, derived from the designed position of the object using the center of rotation. Following this, the contact position is precisely defined for each time step. Nevertheless, the exact contact forces cannot be predetermined and planned in advance. As a solution, a residual policy learning-based approach is implemented to complement the torques on selected joints, ensuring effective handling of both desired contact points and forces. The resulting torques are a combination of the controlled torques $\boldsymbol{\tau}_{ctl}$ and the augmented torques $\boldsymbol{\tau}_{aug}$ learned through RL:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{ctl} + \boldsymbol{\tau}_{aug} \tag{2}$$

The chosen RL algorithm, actor-critic Deep Deterministic Policy Gradient (DDPG), involves the actor network selecting actions based on augmented joint torques. Simultaneously, the critic network estimates



(a) Three link robotic arm push task configuration.



(b) Push an object to the desired location by instant center of rotation approach.

Figure 1: Push task configuration.

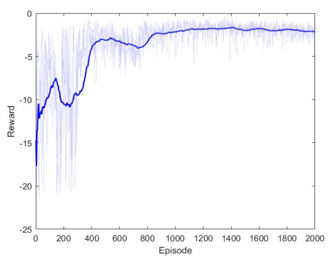
the value of different states or state-action pairs. The states include contact forces, translation, and rotation of the object. Both networks comprise three fully connected layers with 32 neurons, and ReLU activation layers transform the weighted sum of inputs in each neuron into an output signal. The learning rate for both the critic and actor networks is set at 0.001. The training process involves a total of 2000 episodes, each consisting of 500 time steps. The reward function is designed based on the error between desired applied forces and contact forces of each time step.

Results

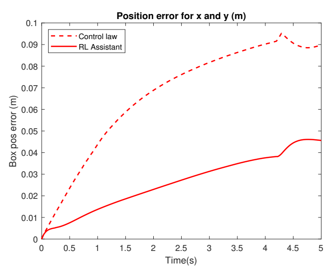
Two distinct policies were developed and evaluated for pushing an object with final randomized positive and negative angles at a constant angular velocity:

- Policy1: Push the object from $[-0.1m \;\; 0.5m]$ to $[-0.5m \;\; 0.35m]$ but angle from 0 deg to the randomized between 20 deg and 30 deg.
- Policy2: Push the object from $[-0.1m \;\; 0.5m]$ to $[-0.5m \;\; 0.7m]$ but angle from 0 deg to the randomized between -30 deg and -40 deg.

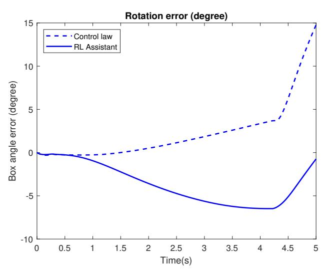
We conducted comprehensive tests on DDPG agents, evaluating their performance based on 100 randomly selected final angles for each policy. The results demonstrate a substantial improvement in both translation and rotation accuracy compared to the position control law at each time step. The final translation error is successfully reduced from 25% to below 10%, and the final rotation error exhibits a significant decrease from 80% to below 10% for both agents.



(a) Training reward versus 2000 episodes for Policy1 by RL-assisted approach .



(b) Translation error comparison between position control law and RL-assisted approach for angle 21.6 deg.



(c) Rotation error comparison between position control law and RL-assisted approach for angle 21.6 deg.

Figure 2: Training results by position control law and RL-assisted approach.

References

[1] Silver, Tom and Allen, Kelsey and Tenenbaum, Josh and Kaelbling, Leslie: Residual policy learning. ArXiv preprint arXiv:1812.06298, 2018.