Inertia Cancellation via RL based System Identification

Matt McDermott

Introduction

As humans begin to live and work more closely with robotic actuators it is extremely important for such systems to remain as safe and predictable as possible. In some situations it is possible to make functional devices so weak that they pose a limited threat to human safety, however, generally actuators must find a balance between strength and safety unless special measures are taken to guarantee safety.

If the dynamics of a robot can be modeled with sufficient accuracy, feedforward linearization can be used to predict how the joints of a robot will move as a function of forces acting on the system and apply sufficient torques to cancel out a large portion of the internal forces of the robot. This is incredibly useful because it opens the door to new types of upper body exoskeletons. Current models either offer non-backdrivable joints which can be strong but are too clumsy for many useful tasks or operate using backdrivable joints with gravity cancellation and crude friction cancellation using the assumption of linear viscous damping and no consideration of dynamics. Although mobile and admittedly very cool looking, exoskeletons that operate using only gravity and viscous damping cancellation essentially give the same benefit that has been provided by overhead cranes for hundreds of years. While a user may not be required to exert the effort to fight against gravity when moving an object, the effect of inertia is still apparent in the object that is being moved and it will take significant effort to accelerate. If tuned correctly, inertia cancellation can potentially make a large object feel as though it has significantly less mass.

Background/ Related Work

System identification is a growing domain within the field of controls and there have been numerous publications focused on using reinforcement learning neural networks for tasks of system identification. One notable example can be found in *Reinforcement Adaptive Learning Neural-Net-Based Friction Compensation Control for High Speed and Precision* in which the authors develop a procedure to identify a stick-slip friction model for a sliding block system. While it is calculating stick-slip parameters for a 3 joint robot could be extremely difficult analytically, it is possible to obtain such parameters using DDPG.

Because neural networks are universal function approximators, it is possible for the dynamics of a complex multibody system to be determined to a sufficiently high resolution using a neural net with no prior knowledge of the system. That being said, because neural networks are a black box and do not give designers useful insights into the inner workings of a system, it is entirely possible that a network trained to work sufficiently well in one region of a robot's workspace could fail horribly to predict the dynamics in another region, an error that could be very dangerous for any human interacting with the system.

A more robust approach for determining the dynamics of a real world robot is to solve the equations of motion for the actuator using traditional analytic methods such as Lagrangian Dynamics using a symbolic manipulation solver like Maple or SymPy and leaving constants such as link mass, the values within each inertial tensor, and various friction model constants as unknown vairables. Reinforcement learning would then be used to determine the values of these constants. Using RL to determine values of constants once a closed form equation has been found should ideally extrapolate to new regions within the workspace much more consistently than a system characterized entirely by a neural network. Starting off with a baseline of closed form equations of motion means the model is guaranteed to follow the laws of physics for some system, even if it is not exactly the same as the real world robot. In practice, this would mean that an improperly identified system may have steady state drift and slightly inaccurate predictions for the forces acting on each joint which would lead to inertia cancellation not working smoothly, however there would be no sudden discontinuities or jumps in state prediction from one area in the workspace to another which is an issue that is likely to occur if a similar model were to be trained purely through a neural network without sufficient testing in all regions of the workspace.

Correctly implementing an RL system for modeling configuration dependent values would drastically reduce validation time as stability could be proven mathematically rather than through thousands of real world trials. Furthermore, even accounting for some of the benefits of transfer learning, a system modeled using this proposed strategy could be updated more quickly after a designer changes the physical characteristics of the robot (attaching additional sensor and cable mass to the robot, lubricating joints, changing end effectors, etc.) because the form of the baseline equations that govern the dynamics of the robot remain unchanged and only certain values are tweaked.

This strategy of RL based system identification allows solutions of greater accuracy to be reached with less time in simulation, less uncertainty about the stability of an obtained solution and less time spent doing manual trial and error.

Technical Approach

The equations of motion are determined analytically via Kane's method as:

$$\mathbf{M}(q,t)\dot{u} = \mathbf{f}(q,\dot{q},u,t)$$

Where the left side of the equation is the mass matrix and the right hand side is the forcing function. q represents generalized coordinates and u represents generalized speeds. Because most formulations do not include friction in the mass matrix since energy is being extracted from the system, it is included in the forcing function on the right hand side.

While it is possible that some of the system parameters evaluated in this project such as link mass and inertial tensors will (or at least should) remain consistent for all position and velocity states, other parameters such as constants associated with different modes of friction will become more or less prominent at different robot configurations. It may seem trivial to experimentally evaluate a simplified viscous damping model for friction of one joint at a given robot configuration, however, there is no guarantee that values determined in one neighborhood of the workspace of a 3DOF actuator will be applicable to another set of states far away. This stems from the fact that any real world robot's joints are going to behave very differently depending on which direction they are being loaded from. Bearings generally have lower friction when loaded radially rather than axially, and any pair of moving parts will have a higher coefficient of static friction than kinetic friction. Because of the high dimensionality of this problem, determining an accurate model of friction would be incredibly difficult using solely analytical methods.

Another benefit of using RL for system identification tasks like this is that it does not limit the solution to a single type of model for friction. For example, stick-slip may accurately model frictional forces at forces at which a surface begins to move but fails to accurately model friction at high speed when compared to another method. In the formulation of this problem, both models of friction can be included in the equations of motion and the constants determined by the RL agent for each will change to reflect how important each mode of friction is based on the current states of the system. Being able to easily “mix and match” different models of friction and other parameters is quite useful because it does not require the designer to be an expert in friction to be able to model the process.

Tasks

The primary objective of this project is to identify certain physical parameters of the robot so that gravity, friction and inertial cancellation can be performed as accurately as possible. Unlike past iterations of this project, this implementation will not use viscous damping to approximate the energy lost to friction but instead will use a combination of more complex models including dry and stick-slip friction. All models will be included with separate constants so that different modes of friction can be weighted more or less importantly as a function of system configuration.

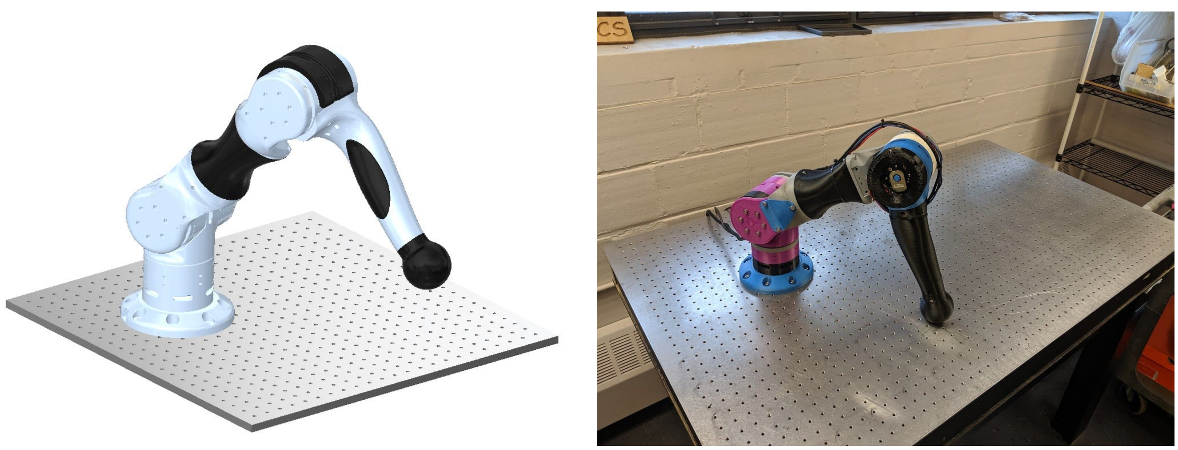


Fig. 1: 3DOF Backdrivable Actuator

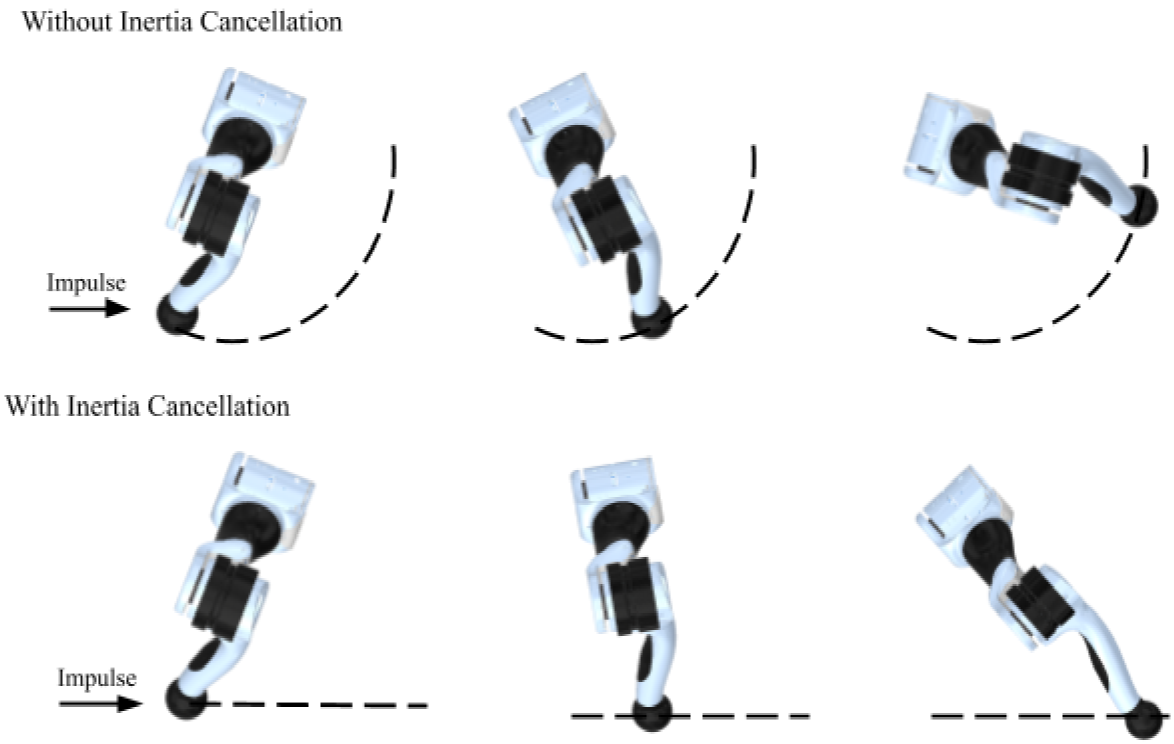


Fig. 2: Visualization of movement after inertia of robot is removed

Evaluation

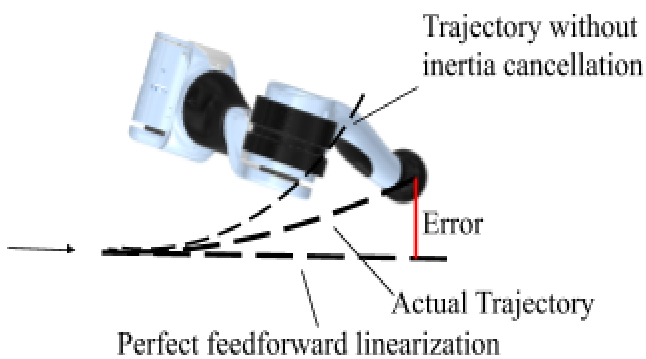
Overall success of the implementation and "transparency" of the actuator will be evaluated through two metrics:

1. Steady state position drift
2. Deviation from linear behavior in feed-forward linearization used in inertia cancellation

The first metric means that the experimentally determined value for the mass of each link should be accurate enough that when the arm is moved to any configuration the torque applied to each joint (as a function of link length, link mass and joint angle) is just enough to prevent the arm from falling over as a result of being overpowered by gravity, and also not too much such that the joints accelerate upwards against the force of gravity. This steady state error will be relatively easy to quantify using data from joint encoders as well as physical observations of the actual robot. These values will be determined first with a relatively simple parameter sweep starting with j2 and moving backwards. Once mass values are determined, they will be used as constants for the second more difficult problem.

The second metric involves determining the 3D inertial tensors with a sufficiently high degree of accuracy. If the inertial tensors and friction modes are modeled perfectly, feed-forward linearization should result in any impulse on the end effector causing behavior similar to pushing around an object in outer space. Not only would the end effector of the robot feel like it is unaffected by the force of gravity, but disturbing the end effector would result in the end effector moving in a straight line in the direction opposite to the application of force. This would mean that when closing your eyes and grabbing the hand of the robot it would feel like you are pushing around a styrofoam robot, rather than one that weighs 35lbs.

100% inertia cancellation of a system which has no outside loads acting upon it is inherently unstable (applying any force to an object with zero inertia results in infinite acceleration), however, even an 80% reduction in inertial forces is enormously beneficial in many applications. For the second goal, the error between perfect and actual trajectory will be used as the error function for the agent.



Timeline and Individual Responsibilities:

Task:	***Date Due***
Design & Fabricate v9 Hardware	Done
Project Proposal	11/6
Wire new robot to control box	11/15
Determine Friction Modes	11/15
Solve EOM with Friction	11/22
Complete Task 1	11/27
Complete Task 2	12/10
Reach Goal: Cancel inertia of additional DOF on the end effector (robot acts as underactuated system)	-

References

1. Young Ho Kim and F. L. Lewis, "Reinforcement adaptive learning neural-net-based friction compensation control for high speed and precision," in *IEEE Transactions on Control Systems Technology*, vol. 8, no. 1, pp. 118-126, Jan. 2000, doi: 10.1109/87.817697.

2. Cerrada, Mariela & Aguilar, Jose. (2008). Reinforcement Learning in System Identification. 10.5772/5273.