Estimating Object Inertia Properties for Trajectory Control

Aamir Rasheed (aamirar@stanford.edu)
CS 229: Machine Learning, Stanford University

Abstract

As robots become more ubiquitous, techniques are required to help robots deal with increasing uncertainty. In this project, I develop an approach for trajectory tracking of a manipulator that has successfully picked up an object of unknown inertial properties. My goal is to estimate the relevant inertial properties of an object so that the robot arm can continue following the desired trajectory. I framed the problem as a regression problem and used linear regression and a basic neural network to output predictions of the mass and location of the COM. The models did not work as expected, performing poorly, most likely due to a lack of meaningful training data.

Robotics Background

To track a trajectory, we use a PID control law:

$$F^* = \ddot{x}_d - k_v(\dot{x} - \dot{x}_d) - k_p(x - x_d)$$

evaluated at each time step, where x_d represents our desired acceleration, velocity, and position vectors and F* is the 6 x 1 acceleration vector we have computed to attempt to stay on the trajectory.

$$\Gamma = AF^* + b + g + {}^0J_{com}^T \begin{bmatrix} F_{mass} \\ M_{mass} \end{bmatrix}$$

Above is the basic dynamics equation that describes the motion of a robot, where A is the mass matrix of the robot, F* is the accelerations vector derived from a PID control law, b is the joint space centrifugal and coriolis vectors, g is the joint space gravity vector, J_com is the Jacobian of the robot at the COM of the object with respect to the base frame, and F_mass and M_mass force and moment values the end effector must exert in order to compensate for the object weight.

Problem Formulation, Data Collection, Learning Approach

To compute F_mass and M_mass}, we model the mass at the end effector as a point mass with mass m at its COM, with the vector r representing the position of the center of mass with respect to the end effector frame. Using this, we can then model F_mass and M_mass as

$$\begin{bmatrix} F_{mass} \\ M_{mass} \end{bmatrix} = \begin{bmatrix} m\hat{g} \\ -\hat{g} \times mr \end{bmatrix}$$

or, by decomposing the knowns from the unknowns,

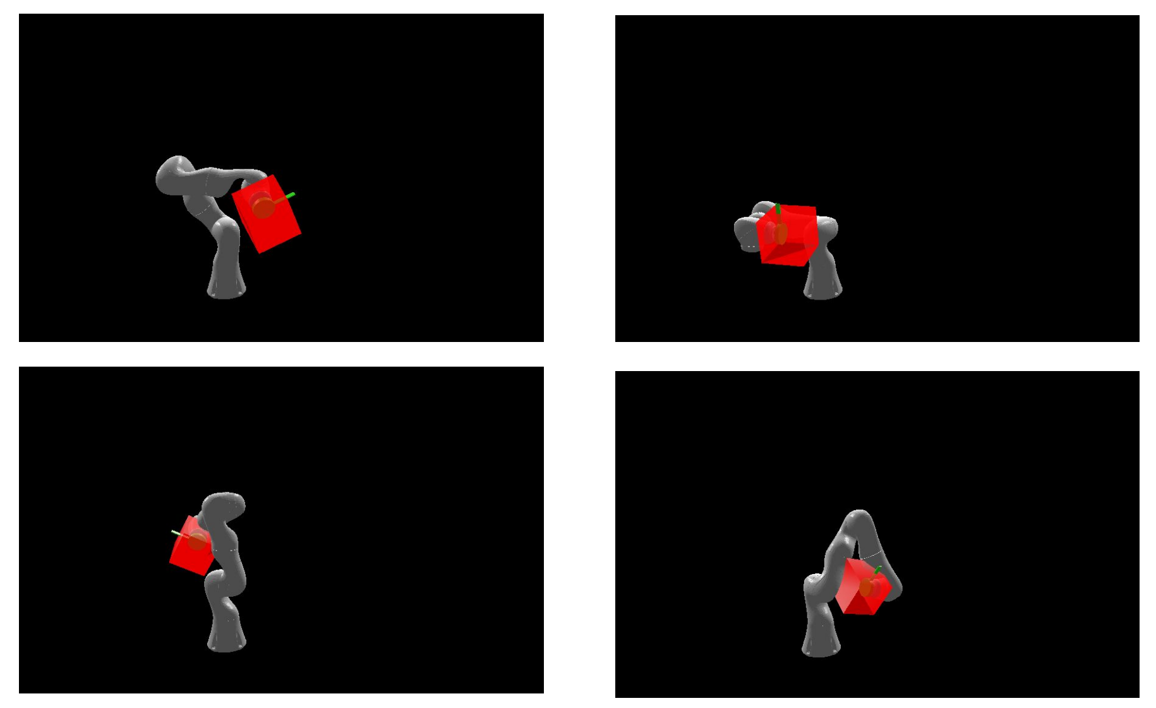
$$\begin{bmatrix} F_{mass} \\ M_{mass} \end{bmatrix} = \begin{bmatrix} \hat{g} & 0 \\ 0 & -[\hat{g}]_x \end{bmatrix} \begin{bmatrix} m \\ mr \end{bmatrix}$$

So, in order to determine the force necessary to maintain the trajectory of the object (as well as compute the Jacobian at the com) we must find out what the mass and COM are with the use of the inbuilt force and moment sensor in the end effector of the robot. We can model the end effector mass and force as

$$\begin{aligned} F_s &= F_{mass} + F_{bias} \\ &= m\hat{g} + F_{bias} \\ &= \begin{bmatrix} \hat{g} & I \end{bmatrix} \begin{bmatrix} m \\ F_{bias} \end{bmatrix} \end{aligned}$$

$$\begin{aligned} M_s &= M_{mass} + F_{bias} \\ &= r \times m\hat{g} + F_{bias} \\ &= \begin{bmatrix} -[\hat{g}]_x & I \end{bmatrix} \begin{bmatrix} mr \\ M_{bias} \end{bmatrix} \end{aligned}$$

We now have expressions that relate knowns (sensor readings) to our unknowns (bias of the sensors, mass, and COM). We gather sensor reading data of each mass using a robot that followed a trajectory while grasping the object. Below are images from the simulation



Taking a variety of readings, we can formulate our labels as

$$\begin{aligned} Y &= \{(m_1, mr_1), (m_2, mr_2), \ldots, (m_n, mr_n)\} \\ Y_0 &= \{m_1, m_2, \ldots, m_n\} \\ Y_1 &= \{mr_1, mr_2, \ldots, mr_n\} \end{aligned}$$

and our features as

$$\begin{aligned} X_i &= \{(F_{s,1}, M_{s,1}), (F_{s,2}, M_{s,2}), \ldots (F_{s,d}, M_{s,d})\} \\ X_{i,0} &= \{F_{s,1}, F_{s,2}, \ldots F_{s,d}\} \\ X_{i,1} &= \{M_{s,1}, M_{s,2}, \ldots M_{s,d}\} \end{aligned}$$

This allows us to model our problem as a learning problem

$$\begin{aligned} h(X_{i,0}) &= Y[i]_0 \\ g(X_{i,1}) &= Y[i]_1 \end{aligned}$$

where h and g represent functions that map from the sensor readings to the mass and center of mass. For our linear regression model, these were weight matrices, and in our other model it was a network.

Results and Discussion

Linear Regression Approach

Our linear regression approach was framed as the following:

$$W_f[F_s] = \begin{bmatrix} m \\ F_{bias} \end{bmatrix}$$

$$W_m[M_s] = \begin{bmatrix} mr \\ M_{bias} \end{bmatrix}$$

this approach estimated the mass of the object incorrectly by about 3kgs on average, and the COM by about 0.3m on average.

Neural Network Approach

the architecture consisted of an input layer of size 2 * 3 * d, a number of hidden layers, and an output layer of size 4. We found that the best result was around two hidden layers of 150 nodes each. The mass of the object was estimated incorrectly by about 3.5kgs on average, and the COM was estimated incorrectly about about 0.5m on average.

Discussion & Future Work

Unfortunately, the results obtained during the course of this paper was not enough to be legitimately useful for the problem this approach was attempting to tackle, which was disappointing. My hypothesis is that this is due to the training data not having enough variation to generalize well, so my future approach would include tuning the trajectory to get more varied training data.

References

[1] Altuncu, M. Tarik, et al. "Content-Driven, Unsupervised Clustering of News Articles through Multiscale Graph Partitioning." ArXiv:1808.01175 [Cs, Math], Aug. 2018. arXiv.org, http://arxiv.org/abs/1808.01175.