

Improving end joint force sensing for robotic grasping (Using IMUs to estimate contact force)

Nao Ouyang

Rotation in Harvard Biorobotics Lab

This is my final report for my 299r rotation during Spring 2018 in Prof. Robert Howe's lab.

Abstract. *Despite decades of research, robots remain unable to reliably grasp objects in environments not designed for robots. Thus, they are not able to work alongside us in human environments or outside buildings. One approach is to use tactile sensing to predict whether a grasp will succeed or not as well as build simulations to evaluate potential grasps. In this lab, we use an underactuated three-fingered gripper, which makes it difficult to estimate the location of the fingers and the location of the force with respect to the robot frame of reference. I investigated the use of an inertial measurement unit to provide such information. By characterizing the stiffness matrix K of the finger, I could investigate whether using an inexpensive inertial measurement unit (IMU) to estimate θ deflection with the linear equation $\tau = K\theta$, I could discover information about the force applied. I did very preliminary work, limiting my experiments to the case where we only desire to estimate the change from directly before to second or two after contact. Additionally, I looked at using off-the-shelf machine learning approaches to reduce the errors in our model. Finally, I investigated adding more sensors to the fingertip in order to better estimate contact location.*

1. Introduction

State-of-the art robots are bad at grasping in unstructured environments, limiting their use to structured environments like factory floors. An active area of research in order for robots to expand to unstructured environments (such as kitchens or hospital environments), is the ability to grasp novel objects (objects the robot has not encountered before and may not have any prior information about). Grasp stability analysis plays a critical role in enabling robots to achieve high grasping success rates (in terms of not dropping objects). Through sensing, for instance tactile sensing, we can predict whether or not we will drop the object after grasping and prior to moving. If we find our grasp to be unstable, and assuming the object has not moved during grasping such that releasing our grip might e.g. cause it to tumble, we can simply release our grasp and try again.

Physical model approaches involve modeling the contact forces, contact location, and surface normals (which will differ from the force XYZ if the surface(s) are squishy). However, over the last few decades physical models have proven insufficient. This in part due to complex gripper-object interactions during grasping which are both difficult to measure and hard to model. Thus, one of the aspects of this investigation is to evaluate whether machine learning can be used to improve physical models. We can combine the good aspects of physical models (good generalizability, requiring significantly

less data than in pure computer vision machine learning approaches) with (not having to have extremely precise models). I tried to characterize the residuals in the physical models and determine if these residuals could be more accurately modelled by a black box machine learning approach rather than making the physical model more complex. This is an increasingly popular approach, which makes intuitive sense. We can leverage domain knowledge, instead of using an end-to-end approach for sensor data to stability prediction.

Although there are a few compact 6 axis force torque sensors, they cost multiple thousands of dollars and are finicky and fragile. Thus, we would like to use an inexpensive nine degree-of-freedom (three-axis accelerometer, gyrometer, and magnetic heading) IMU instead.

During the course of this rotation, senior graduate student Qian Wan discovered that the current tactile sensors are insufficient for estimating contact location accurately enough to develop a good grasp stability estimate. Thus, this project includes a brief exploration of a new sensor design using the same elements (barometric pressure sensors covered by a layer of elastomer, such as urethane), but more of them.

2. Theory

2.1. Model

For the model, we are making a few simplifying assumptions:

1. There is a linear relationship between force and deflection, or alternatively torque and angle of deflection
2. The center of the axis of rotation was at the tip of the proximal joint (in reality, likely it is two or three mm shifted outward, and changes as the flex increases)
3. We consider the z axis to be defined respect to the surface of the finger, so that $z=0$ is always at the tip of the finger

The consequence of #3 is that we cannot estimate z-axis torque.

2.2. Reference Frame

```

1 | Coordinates
2 | ----> + x (roll)
3 |
4 |
5 | v
6 | + y (pitch)
7 |
8 | + z (up out of page) (yaw)
9 |
10 |
11 | Finger Positions
12 | =====
13 | {CL}  ||  ---  || 15  12  9  6  3  ||
14 | {A}  [xy=0]-- || 14  11  8  5  2  ||
15 | {MP}  ||  ---  || 13  10  7  4  1  ||
16 | =====
17 |
18 |
19 |  [ [---] ]
20 | ||  WEB  ||

```

```

21 | |  CAM  | |
22 | |---- | |
23
24 | Points 13 through 15 are at x=2.6 cm, and each x is spaced 5 mm
    | apart.

```

2.3. Linear Model, 1D case

In the one-axis case, the math is straightforward.

$$\tau = k \times F \quad (1)$$

For example, one datapoint might be

$$k = F / \tau \quad (2)$$

$$k = (20g \times 9.8m/s^2) / 0.1 \text{ degrees} \quad (3)$$

where ‘k’ represents the stiffness of the finger. Let c represent the inverse of k .

Using least squares error, we may fit a line to find \hat{c} .

$$\hat{c} = \frac{1}{\hat{k}} \quad (4)$$

$$\theta = \tau \hat{c} + b \quad (5)$$

where ‘b’ be a constant determined by the line of fit.

2.4. Residuals

From the above, we may calculate the residuals of any of our estimated variables. For instance, from our actual data we may obtain an estimate for ‘k’.

$$\tau_{data} = \hat{k} \cdot \theta_{data} \quad (6)$$

$$(7)$$

Using this k we can go back and calculate estimates for the ”true”torque, assuming our linear model was correct.

$$\hat{\tau} = \hat{k} \cdot \theta_{data} \quad (8)$$

We would then calculate our torque residuals as

$$\varepsilon_\tau = \hat{\tau} - \tau \quad (9)$$

If we plot a graph of (torque residuals) vs (estimate residuals) and find that our points are randomly scattered around a straight line, then our model well-approximates reality as sensed by a noisy sensor.

However, our residuals may instead follow a parabola, in which case we would want to amend our model to have higher-order terms.

$$\hat{\tau} = \hat{k}\theta_{data} + c_1\theta_{data} + c_2\theta_{data} \quad (10)$$

and so forth.

We may eventually use machine learning techniques to fit higher-order terms.

2.5. 3D case

The 3d case is exactly the same, except now each of the variables are vectors or matrices.

$$[\vec{\tau}]_{3 \times n} = [K]_{3 \times 3} \cdot [\vec{\theta}]_{3 \times n} \quad (11)$$

where

$$\vec{\tau} = \begin{bmatrix} 3 \times 1 & 3 \times 1 & \dots & 3 \times 1 \\ \tau_1 & \tau_2 & \dots & \tau_n \end{bmatrix} \quad (12)$$

$$(13)$$

$$\vec{\theta} = \begin{bmatrix} 3 \times 1 & 3 \times 1 & \dots & 3 \times 1 \\ \theta_1 & \theta_2 & \dots & \theta_n \end{bmatrix} \quad (14)$$

2.6. Sanity Check: Simplify to 2D case

We know that $\tau = r \times$

$$[\tau_x \ \tau_y \ \tau_z] = \begin{bmatrix} K \end{bmatrix}_{3 \times 3} [\theta_x \ \theta_y \ \theta_z] \quad (15)$$

Further, we can use some of the simplifying assumptions we made about to model our system, in order to have an idea of what values our math should result in for k .

We may also see that we will only ever have x and y components for r ; z components for F ; and therefore (by how cross products work) only ever have x and y components for τ .

Off-axis

$$r \approx \begin{bmatrix} r_x \\ r_y \\ 0 \end{bmatrix} \times f \approx \begin{bmatrix} 0 \\ 0 \\ f_z \end{bmatrix} = \tau = \begin{bmatrix} \tau_x \\ \tau_y \\ 0 \end{bmatrix} \quad (16)$$

$$(17)$$

On-Axis

In the even more simplified case, if we do not apply the force off-axis and there is only a pitch deflection

$$r \approx \begin{bmatrix} r_x \\ 0 \\ 0 \end{bmatrix} \times f \approx \begin{bmatrix} 0 \\ 0 \\ f_z \end{bmatrix} = \tau = \begin{bmatrix} 0 \\ \tau_y \\ 0 \end{bmatrix} \quad (18)$$

$$(19)$$

We **could** have enough nonzero values for the xyz components of θ that we will be able to calculate a nondegenerate estimate for k . This is thanks to applying the force off-axis, causing the finger to roll and create non-zero elements for θ_y (our roll) and θ_z (our yaw). However, after solving for K, I found that I ended up with a bottom row consisting entirely of zeroes.

3. Methods

For the setup, I used a triple-beam balance as a way to apply force while retaining a vertical angle. I hot-glued a pumpkin awl to the bottom of the plate and put standard weights on top of the scale. This caused the awl to contact the surface of the finger, and the finger would then deflect. In the 1D case, I simply placed a ruler behind the finger and used a webcam to take pictures. By mapping pixel counts to millimeters, I could then estimate the deflection of the finger. The linear fit from these very imprecise measurements proved linear enough that I then relied on the IMU going forward and skipped the step (which should be completed eventually) of characterizing the accuracy of the IMU versus a gold standard, such as a stereo camera (the Microntracker available in lab) or other methods (such as using Apriltags with a calibrated webcam).

I then created a grid of 15 points on the finger. For each of the 15 points, I applied several known amounts of force, with three samples at each force-point combination. Each sample consisted of a "zero" measurement pre-contact and a measurement post-contact. In post-processing, I then subtracted the two to obtain my final sensor data.

Complications arose from the limited travel of the triple beam balance, which meant I could only measure small deflections (initially on the order of 4 mm; after discovering that I could take the end stop off the back edge of the triple beam balance, about 10mm). At the tip of the finger, I could only less than a hundred grams of "force" (about 1 N), while 4N is a normal amount of force applied by humans to grasp things. This remained a source of frustration throughout the data collecting and ate up many hours of adjusting the experimental setup.

I glued an IMU to the undersid of the finger. The IMU I used was the Bosch BNO055 sensor. A breakout was available from adafruit.com for \$35, which is relatively inexpensive. I chose this sensor not only for the built-in processing, and also because of the extensive beginner-friendly documentation developed by Adafruit for all of its products. It was recommended to me by at Right Hand Robotics, which is a spinoff of this lab.

"Bosch is the first company to get this right by taking a MEMS accelerometer, magnetometer and gyroscope and putting them on a single die with a high speed ARM Cortex-M0 based processor to digest all the sensor data, abstract the sensor fusion and real time requirements away, and spit out data you can use in quaternions, Euler angles or vectors."(source: adafruit.com)

I also attached an apriltag (which looks like a QR code) and a micontracker tag (which has a checkerboard like cross), both of which can simply be printed out on paper, and calibrated the webcam for use with the Apriltag C++ script. Although I have collected microntracker and apriltag data for comparing to the IMU deflection data, I have not had a chance to analyze this data yet,

Additionally, the IMU acclerometer would consistently refuse to calibrate if the sensor did not start parallel to the ground. The significant slope of the finger versus the clamp and 90 degree angle of the 80/20 meant that I had to approximate a slanted 80/20 setup by increasing the degrees of freedom on the L brackets (removing t-nuts). This led to a lot of worry on my part about the creakiness and variability in the data collection as I could not guarantee angle consistency, although in retrospect this is almost completely mitigated by the fact that I am zeroing before every reading.

As I progressed, I trended toward collecting coarser and coarser data as I grew confident that measurements still provided enough data for me to see overall trends in the data (such as the linearity of the data). I shifted from 2g increments, to 20g, and in the end 50g increments. This meant that for the finger when stiffness was applied, I had to change the setup so that the finger started at an angle and became parallel to the floor when load was applied to the tendon. After my deep grievances with producing angles on the 80/20 setup, I decided to use legos with popsicle stick props on one end in order to produce the correct angles.

Finally, I discovered at the very end that I could calibrate the scale to have a constant force offset. Thus, instead of collecting a zero measurement with the tip starting just above the surface, which meant that I had to wait for the oscillations to settle, I could put an offset which made the scale keel all the way to up when the weight was removed. This made zero measurements 2-3x faster, which reduced the cognitive load to remember which measurement I was at, which both reduced data cleanup and needing to recollecting data. This led to significant time savings.

4. Data Analysis

5. Sensor Design

6. Knowledge

In the process of this research, I learned about how accelerometers are calibrated, which I found very interesting (although ultimately, thanks to the built-in sensor fusion and

calibration routines of the BNO055 sensor, I did not need to write my own calibration algorithms). The common presence of the gravity vector across measurements at different angles makes this calibration possible.

I learned the most about exploratory statistical data analysis. Where previously I had not even heard of residuals (only scalar measurements such as mean-square-error), I learned about how they could be compared to a horizontal basis against which they would be randomly distributed assuming random noise around a proper physical model.

Finally, I learned more about the I2C and SPI protocols for receiving (and writing) to multiple sensors. I began work on creating a custom printed circuit board to create a 3x4 array of sensors, and decided to switch and learn to use a newcomer in the PCB design space, which has CERN support: the free and open-source KiCAD tool.

7. Conclusions

The stiffness of the finger was remarkably linear, which meant that machine learning and higher order terms had little to contribute to improving the fit between measured theta deflections and force estimates.

However, I was only able to apply relatively small loads to the finger. It's possible that with large loads, the stiffness becomes less linear, in which case linear fit residuals may become more prominent or structured. If this were the case, machine learning approaches might have more to contribute to the force estimates. As it were, the data I collected did not benefit much from the addition of non-linear models fitted with machine learning techniques such as random forests.

Most crucially, Qian demonstrated to me the process by which grasps fail. By observing the grasping, I discovered that when the tendon was pulled taught (as needed to grasp anything), the y-axis was now the stiffest axis! Thus, characterizing the finger with no tendon load applied and primarily in the y-axis direction, while a great learning experience, does not generalize at all to practice and becomes a toy example.

Additionally, the finger relied extensively on mechanical mechanism intelligence (as opposed to full actuated parallel grippers, commonly used in computer science labs). I saw that complete failures tended to occur because the finger had failed to detect contact when contact had already occurred. This suggests the better instrumentation of the finger is critical. The IMU by itself can only suggest contact force, and the tactile sensors are needed to tell contact location. To evaluate the contribution of the IMU to force estimates and not just torque estimates, we must build better location sensors than the current 1x4 array.

7.1. Future Work

The microntracker should be used to evaluate the IMU. Evaluating the use of Apriltags with the microntracker would help reproducibility by others (or future grad students), as webcams are cheap and readily available, and documentation for Apriltags is freely available online.

Additionally, a full sensor array on a custom PCB, manufactured into a finger unit, should be developed and characterized. By collecting data and then fusing this with the IMU data, we could physically evaluate the grasp stability predictions, producing an

important tie between theory and practice to ensure the two complement each other. The important of this has been mentioned above. I