Robotic Manipulation with a Human in the Loop

Mark Jouppi
Sebastian Schweigert
Jiewen Sun
James Su
Sunil Srinivasan

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-78
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-78.html

May 13, 2015

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2015

Electrical Engineering and Computer Sciences

Robotics and Embedded Software

Robotic Manipulation with a Human in the Loop

Mark Norman Lester Jouppi

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: ___________________________ Date _____________

Print Name/Department: Professor Ruzena Bajcsy, Electrical Engineering & Computer Sciences

2. Faculty Committee Member #2:

Signature: ___________________________ Date _____________

Print Name/Department: Professor Ronald S. Fearing, Electrical Engineering & Computer Sciences

Robotic Manipulation with a Human in the Loop

Master of Engineering Final Capstone Report

Table of Contents

Robotic Manipulation with a Human in the Loop

Sebastian Schweigert, Sunil Srinivasan, Jiewen Sun, Jimmy Su, Mark Jouppi

Final Capstone Report – Common Section

1 PROBLEM STATEMENT

We live in an age of increasing automation, but while we have machines that can open a can, pour a glass of water, or assemble a piece of furniture, the world does not have a machine that is versatile enough to do *all* of these tasks.

Normally, when people think of automation, they think of robots designed to accomplish a very specific task, such as lifting a car-door into a car-frame, over and over again. The newer generation of robots though is the class of general-purpose robots. While such robots have yet to materialize commercially, general-purpose is a great concept. Imagine if families could have a robotic assistant to take care of household tasks or run daily errands. In short, human life would be much more convenient and efficient.

Unfortunately, a major limitation towards reaching such a milestone is the engineering trade-off between cost and performance: with a limited budget of resources, it is almost impossible to add additional levels of complexity without decreasing performance. As such, it has traditionally been challenging to use robots that are both low-cost and versatile in domestic environments because the applications of these robots are limited by their low performance – specifically, their inaccuracy. This is where we decided that the human should come in.

To address this barrier of limited resources, our capstone team has developed a system that is designed around robot-human interaction, where human instructors train and work with cost-effective robots to accomplish a broad range of tasks with high accuracy. Using a set of algorithms that we have developed, the robot learns how to perform a task from a human who teaches it through a series of demonstrations. Following this learning process, the robot evaluates the task and identifies the precision requirements using a mathematical model. And when the robot detects that it is unable to achieve the accuracy required for a certain portion of the task, it requests human assistance. The final outcome is a system that excels across a vast range of duties, due to the combination of both the efficiency of robots working on a large-scale and the precision of humans working on a small-scale.

This revolutionary design of cooperation between man and machine succeeds at tasks that are otherwise impossible for the machine to accomplish alone. In essence, we added in the human as an additional resource to improve the overall performance of the system. This was the rationale behind our capstone project, for we saw an opportunity here to make an enormous technical stride in society’s current usage of commercial robots: we took an otherwise unimpressive commodity – the low-cost and inaccurate robot – and engineered commercial value from it in the form of robotic adaptability.

2 INDUSTRY AND MARKET TRENDS

Before examining any technical details though, we first wanted to scope out the business potential of our project. Consequently, in an attempt to analyze our strategic position in the market, we evaluated the competitive forces outlined by Porter (Porter, 2008) because we felt that an in-depth analysis of the intensity of these forces will influence our marketing strategies. In other words, analyzing these five forces enabled us to have a better understanding of our industry and shaped our strategies to sustain long-term profitability. Before we begin our analysis however, let us first clearly define both our market and our product.

2.1 Market and Product

We defined our market to be consumer households with the intent that our algorithms accomplish household tasks, such as assembling furniture. We chose to target the elderly and the disabled as buyers of our product because this is a large, growing population with critical and largely unmet needs. Simply put, the elderly population in the United States is growing. While the current number of senior citizens in the US is roughly 40 million, that number is expected to grow to over 80 million by 2050 (Ortman et al., 2014). Additionally, according to US 2010 census data, about 30%, 38%, and 56% of the population aged 70 to 74, 75 to 79, and 80 or over, respectively, live with severe disabilities (Brault, 2012). To further narrow our market though, we chose to focus specifically on affluent elderly-and-disabled individuals as our target customers. This is a reasonable objective because many elderly people have amassed a wealth savings and investments cultivated over their lifetimes. Indeed, according to a Pew Research study, the median net worth of senior citizens in the US is $170,000, which is 47 times greater than that of people aged 35 and younger (Censky, 2011).

The definition of our product is a more complex matter because, at its core, our capstone project involved the research and development of an algorithm that allows a robot to learn a task and cooperate with a human to perform that task; it is not a complete software – or hardware – solution. Unfortunately, while software solutions usually have commercialization potential, algorithms alone do not. In order to take our robot-learning algorithm and relate it to a commercial application, we had to decide what form that application should take and how to take such a product to market. One option was to simply license out our algorithm for others to utilize; we would receive royalties as a result of these sold licenses, and companies could make products or provide services using our algorithm. One major caveat, though, is that our algorithm incorporates ideas presented in externally-published research, so the intellectual property for this method may not lie entirely with us. We therefore chose not to investigate this option any further. Our next option to consider was to sell a software solution for users to install on devices that they already own. However, the “device” in this case would be a full-fledged robot, where, as a point of reference, a Baxter robot from Rethink Robotics – our current hardware-platform of choice – has a set price of approximately $35,000 (Rethink Robotics, 2015). Clearly, it would be ludicrous for people to purchase such costly technology without ensuring that it already comes with the necessary software to function. This left us with our final choice: a “full package”, in which we offer a robotic apparatus preloaded and set up with our software such that a consumer only needs to buy one product, with installation services if necessary. This way, we can market our product directly to our target consumers and eliminate the customer’s barrier-to-purchase that comes from setting up the technology. Thus, we decided on this “full package” as the form for our product: a physical robot bundled with software algorithms that we implement.

We must consider several factors with the decision to market this “full package”. The first is price, and this is largely influenced by the suppliers since we must obtain the proper robotic hardware or components externally. After all, according to an IBISWorld report, the cost of mechanical manufacturing is increasing as the expenditure of raw materials increases, so we opt to purchase a whole robot setup instead of building our own robot from basic components (Crompton, 2014:25). As a result, we would look to Rethink Robotics as a supplier of our Baxter robot, a hardware platform. With a markup from our software and services, selling our product at around $40,000, or at about a 15% markup, is not an unreasonable price point – especially if we were to get an Original Equipment Manufacturer (OEM) discount for Baxter. This provides us with a defined pricing model.

Lastly, we must discuss promotion and place/distribution. As O'Donnell points out, 50% of seniors are not using the internet, so marketing is better achieved through conventional channels such as mail, television, and newspapers (O'Donnell, 2015). Interestingly, O'Donnell also predicts an increased use of social media by seniors in 2020, making social-media campaigns a possibility in the near future (O'Donnell, 2015). Distribution of this product, however, is complicated; while we would like to be able to sell our product online, providing setup services would require a trained professional to be present. As such, we will most likely have to either distribute through local partners that provide such services or create a local presence ourselves, incurring additional costs. With our product, price, promotion, and place now defined, we have all the significant facets of a commercialization strategy. Note that we do not analyze the minimum viable product (MVP) in detail. This is because our research specifically investigates the Baxter robot's ability to learn the task of assembling a coffee table, at which point we will have a decent MVP that performs table assembly. Thus, we have established a viable (if hypothetical) commercialization strategy for our research efforts.

2.2 Competitive Forces Analysis

2.2.1 Power of Buyers

With the market and product definition out of the way, we can begin to evaluate Porter's five forces, the first of which is the power of buyers (Porter, 2008). We deduce this force to be relatively weak, since the large population of potential buyers means that individual buyers do not have much leverage or bargaining power with us in our product offering. Moreover, as we will address later on, there are few – if any – direct rivals in our industry. Thus, a scarcity of competing products only elevates our power, as options are limited for the buyer. Furthermore, the switching costs for complex, robotic solutions would be high; given that the price of these robots with our software would be roughly $40,000, it is not an expense to be made frequently. We imagine that a typical customer will only purchase one such robotic system in their life. Thus, it is not of great concern that customers would switch to using a competitor's domestic robot solution after purchasing our product. All in all, the power of buyers is assessed to be fairly weak, and we do not concern ourselves in mitigating this force.

2.2.2 Power of Competitors

Regarding rivalry within our industry, there are two main classifications of competitors: robotics companies and robotics research institutions. Some of these competitors offer products that are mildly similar to our envisioned product, and they also target similar markets. For example, Clearpath Robotics, a robotics company (Hoover's, Inc. "Clearpath Robotics Inc.," n.d.), offers support to the PR2 robot to perform household chores like fetching food from the refrigerator and cooking it. Alternatively, there are research institutions like the Information and Robot Technology Research Initiative (IRT) at the University of Tokyo working on developing software that allows the AR robot to accomplish household assignments such as cleaning floors, washing laundry, and so on. Fortunately, companies and research institutions like these will only indirectly compete with us because our product differs from theirs in the extent that humans are involved. The robotic systems these competitors are developing are meant to be fully autonomous – the robots execute their tasks independent of any human interaction – while our system is meant to be semi-autonomous, enabling a human to both work with and teach a robot to perform various tasks. This is an advantageously superior method because now the scope of the system is not limited to what the robot can accomplish independently; the scope is broadened to what the robot and human can accomplish together synergistically. Simply put, the generality of our method enhances a robot's utility and flexibility. Apart from offering a unique product though, we also have some advantages over our competitors in terms of hardware costs. To illustrate, a two-arm PR2 robot is priced around $400,000 (Owano, 2011) while a Baxter robot, as mentioned previously, is priced around only $35,000 (Rethink Robotics, 2015), a relatively far-cheaper option. To summarize, since we are working in a fairly new field, there are no true established rivals in this specific area yet. Thus, we can conclude that the force of competitors is weak.

2.2.3 Power of Substitutes

Moving onto the next force listed by Porter, we realize that significant attention needs to be given to the force of substitutes since there are, broadly speaking, quite a number of substitutes to our product. For instance, alternative technologies, like the iRobot Roomba (Biesada, n.d.) – a popular floor cleaning robot, have existed in the consumer market for many years, and these established technologies have a large customer base. Customers are more comfortable with familiar products, so it will not be easy to encourage customers to migrate to a substitute product. Moreover, if we look past the technological substitutes, there are a variety of human-labor alternatives in regards to accomplishing household tasks, such as employing a live-in caretaker or residing in a nursing home. However, similar to our stance against the competitor force, we again have some advantages due to our functionality and low cost. Addressing the concern of alternative technologies, even though products like the iRobot Roomba are popular and functional, they tend to have a limited set of features, such as floor cleaning. Our product, on the other hand, is a more general solution which can be used to tackle a variety of household chores. Along that same line, for many tasks in this set, our robot can be more efficient than a human caretaker due to its autonomous nature. Furthermore, as mentioned previously, our pricing model markets our product at a cost of about $40,000 with an extensive lifespan, while most nursing homes cost up to $80,000 – and that is per year (Ellis, 2013). All of these arguments make our product competitive to existing substitutes, motivating us to divert attention from this force and concentrate on more pressing ones.

2.2.4 Power of New Entrants

In contrast to the mild nature of the forces mentioned previously, new-entrant competition looming over the horizon should be of great concern. For instance, some of the heavy-hitters in robotic research include companies like Clearpath Robotics (McMillan, 2015) and 3D Robotics (Bi, 2015), both of which were founded only six years ago in 2009. It seems that, unlike the issue of existing rivals and possible substitutes, there is indeed a strong force in regards to new entrants. To further illustrate this fact, large corporations with broader goals in the technological field can certainly seep into our industry, such as Amazon with its Amazon Prime Air drones or Google with its autonomous cars. Big players such as these would certainly have the resources to quickly create a new division within their company and fund research in alternative robotic avenues. Furthermore, even our suppliers can be considered possible new entrants, since they both already posses their own hardware and can additionally reverse-engineer our software algorithm that was, in large-part, acquired from public research papers. All in all, to summarize, we see that dangerous incoming players in this industry are: either startups or big companies with other additional focuses, or suppliers that provide our hardware. When combined with the fact that there are no true established rivals yet as mentioned previously, this danger reinforces both the notion that robotics is a relatively new field and that the threat of new entrants is high.

2.2.5 Power of Suppliers

The last of Porter's five forces to address is the threat of suppliers (Porter, 2008). This threat is a complex point that requires careful analysis in our business strategy. To first clarify, we envision robotic-hardware-platform manufacturers as our suppliers. As per our product description, we would take the robotic hardware platforms from companies like Rethink Robotics and Universal Robotics, customize the robots with our specialized software that gives them practical intelligence to work alongside humans, and then sell them to customers. In particular, we would purchase from companies that produce innovative, low-cost robotic hardware platforms upon which we can then build our solution. Our smart software would make up for the inaccuracies in the cheaper hardware with better algorithms and human-in-the-loop collaboration. Since there are currently only a few firms producing such low-cost platforms, these few suppliers have high bargaining power, as we are left with fewer alternate firms from which to choose.

2.3 Market Strategy

We see that presently, of Porter's five forces, both new entrants and existing suppliers hold the most power (Porter, 2008). Knowing this, we can establish our market strategy to mitigate these two forces, strategically positioning ourselves in a superior situation.

To mitigate the threat of new entrants from the suppliers themselves (see Section 2.2.4), we can generalize our software to work across multiple platforms and disincentivize suppliers to enter the market, as they would only be encouraged to produce software across their own single platform. Additionally, to discourage new and small startups from forming, we can both establish strong relationships with suppliers to gain a leg up on others looking to pursue our method of utilizing existing hardware and maintain a high fixed cost – such as a high R&D cost by developing more proprietary algorithms – to deter incomers that have a small amount of seed funding. Finally, we can address the threat of entry from large corporations by realizing that these companies have more overarching goals, so focus on their robotics branch will not be as heavy as on their other branches. As such, we can capture a niche market to detour focus and attention away from us. Fortunately, we have already positioned ourselves in such a situation, in which we target a niche group of customers – the elderly and the disabled. As a result, we see that our competitive landscape as it applies to new entrants can be classified as quite aggressive, but there are indeed routes we can take to dodge much of this aggression.

To mitigate the issue of being locked into a single supplier (see Section 2.2.5), the core strategy is still to generalize our software. This would considerably increase our power, since we would no longer be dependent on any one supplier. Note that as a trend, robotics startups are becoming increasingly common (Tobe, 2013), and we thus anticipate more suppliers coming into the market in the future. As of right now though, suppliers are a strong force that must be considered carefully in our strategy, and we must route efforts to ease this force.

2.4 Market Trends

With an evaluation of competitive forces complete, we end with a discussion of the major trends that influence our project strategy. Aside from the trends of both the changing age demographic of the US – affecting the power of both buyers and substitutes – and the increased interest in the robotic industry – affecting the power of both substitutes and new entrants, another trend to consider is the recent advancement in integrated circuit (IC) technology that has resulted in improved computing performance and reduced cost, resulting in reduced barriers-to-entry and thus further enhancing the threat of new entrants. IC technology has seen consistent improvements in computing power and consistent reductions in cost since their inception. Our industry is directly affected by these advancements; in recent years, more powerful computational devices have generated more robotic technology in the household arena, for engineers are allowed to easily incorporate computing power into the chassis of the robot. This design contrasts with industrial robots, where the computational power is often located in an external computer. The trend is summarized with a concept known as Moore's law, stating that the computational power of the average IC doubles nearly every two years. This trend has been relatively consistent since the early history of ICs. However, there is disagreement among analysts about how much longer this trend will continue (Hoover's, Inc. "Home Health Care Services.," n.d.). The trend has the effect of making our products more functionally efficient and versatile, which reduces the power of substitutes. However, the lower cost of computing technology also reduces the barriers-to-entry in the industry, which increases the power of rivals. Only time will reveal the overall impact that this trend will have.

To summarize, from our strategy analysis, we have deduced that while some competitive forces are certainly in our favor, a few forces bring cause-for-concern and need to be addressed. With adequate industry analysis, we can plan our strategy in order to leverage ourselves into a better position within the market. Summarizing our findings, we have identified within the market both the power of new entrants and the power of suppliers to be strong forces. Consequently, to dampen these threats, we would generalize our software to work across multiple platforms, disincentivizing suppliers from entering the market as well as taking away supplier bargaining power. We would also encourage people to use our product instead of substitutes by having features and functionality that other products do not, at a price point that is not prohibitively expensive.

3 IP STRATEGY

Aside from a business standpoint though, we must also consider which legal avenues to take in order to protect our intellectual property (IP): in particular, whether or not our idea is patentable. After all, in many research scenarios such as ours, a patent is the most feasible way to safeguard any IP that is developed. Unfortunately, as this section will argue, patenting our work may not be the most practical path to pursue; however, we do have an alternative strategy better suited to our purposes, in the form of copyright.

We feel that in our more specific situation, the costs of attempting to obtain and enforce a patent far outweigh the benefits, for a number of reasons. One consideration is that the mathematics behind the algorithms we employ are pulled from published research papers, particularly those that deal with robot learning-by-demonstration (Billard et al., 2008). Therefore, the proprietary essence of such research is not ours to claim. By the same token, we cannot patent the ROS (Robot Operating System) software platform upon which we develop because it is open-source and thus, once again, publically available. Most importantly, we do not feel that it is pragmatic to patent the software code itself. This is because software, at its core, is the manifestation of logical deductions, and another group or individual may take a different route of logical deductions to arrive at the same conclusion. Following this train of thought, it is ordinarily quite difficult to obtain and/or protect a patent when the end result can be reached in various ways. As explained by Anthony Klein, an IP attorney at Latham & Watkins LLP, pure software patents remain controversial since “what would constitute patentable subject matter is unclear” (Klein, 2015).

Before investigating an alternative means at protecting our ideas however, it is important that we have the foresight to research whether existing patents overlap with our results. We discovered that the closest patent to our project is entitled: “Method and system for training a robot using human-assisted task demonstration” (Bajaras, 2014). It describes a system for humans to train robots for pick-and-place tasks by moving the robot’s arm while recording trajectory-and-perception data through the robot’s sensory systems. At first glance, it may appear that our project directly infringes upon this patent. However, after delving into the details, this is not the case due to the limited scope of this patent. To give some background on the nature of patents, a patent consists of independent claims and dependent claims; if one does not violate the independent claims, then by definition, one does not violate the dependent claims (Brown & Michaels, PC 2006). Now, many of our project’s similarities with this patent lie in the dependent claims. However, if we can argue that our capstone project does not infringe upon any of the independent claims, then we can legally claim that we do not infringe upon the dependent claims as well – and thus the patent as a whole.

There are two independent claims mentioned in this patent. To quote the first independent claim (claim 1):

> A method for training a robot to execute a robotic task in a work environment, the method comprising: moving the robot across its configuration space … assigning, via the ECU, virtual deictic markers to the detected perceptual features (Bajaras, 2014).

We argue that we do not infringe this claim because our project does not use “virtual deictic markers” – markers are based on a representational paradigm that use “selective attention and pointers … to learn and reason about rich complex environments” (Ravindran, 2007). As for the second independent patent claim (claim 2):

> The method of claim 1, wherein moving the robot across its configuration space includes moving at least one of a robot arm and a robot manipulator attached to the robot arm (Bajaras, 2014).

Our project does not use a single “arm and a manipulator”, but rather a dual-armed Baxter-robot. Hence, our project does not violate any of the independent claims and thus none of the dependent claims. Therefore, while this is the closest patent to our idea, we do not infringe upon it and are therefore not required to license from it. Since other existing patents are even less related, a breach of IP is of no worry to us.

With the threat of similar, existing IP out of the way, we can now begin to pursue an alternative strategy of IP protection. After much consideration, we believe that copyright is the most appropriate option – in fact, this happens to be the choice for many software companies. Of course, copyright does indeed present a few risks since, in general, patents protect ideas while copyright only protects the expressions of ideas.

The first risk is the risk of knock-offs: there are ways around copyright such that people can make products very similar to ours but are not in violation of copyright law. This includes implementing our algorithm through a different tactic – one example is converting our code to a different programming language – as well as merely adapting functions from our program. The point is that copyright does not protect our ideas, making it incredibly easy for others to take our ideas and tweak them to look slightly different in their end product. We would need to mitigate this issue by implementing our algorithm across multiple programming languages to prevent the scenarios where someone claims credit on our ideas based on simple modifications.

The second risk is the risk of undetected duplication. It is the first risk in reverse, where certain competitors are indeed copying our code directly, but we have no way of detecting that they are doing so. The reason for this is that we will generally not have the source code of our competitors to compare to our own; all we will have is the compiled functionality that their code is capable of demonstrating. In that sense, it is near impossible to identify specifically if they have violated copyright. Consequently, it is quite difficult to mitigate this risk.

While copyright does offer less protection than patents, it is nonetheless more feasible and realistic to acquire. For instance, copyright is granted automatically when an original work is created, so registration is not required. This simplified procedure immediately eliminates the time and money that we would otherwise need to spend to obtain a patent. Moreover, the duration of copyright is the life of the author(s) plus 70 years, which is plenty of time for us given the short life cycle of software. Furthermore, copyright offers authors the exclusive rights to reproduction, protects against public displays and derivatives of their work, and establishes a public credibility that can attract investment and customers. Licensing can also present itself as a way to increase profit and expand a business.

All in all, it appears that pursuing a patent is not the route for us to go. Instead, a more practical approach at protecting our IP is for us to pursue copyright, due to both the more lenient restrictions and more efficient timeline at obtaining copyright.

4 SYSTEM WORKFLOW

Shifting gears now to the technical details of our capstone project, let us clarify once more that we developed a general-purpose robotic system that incorporates robot-human interactions. Yet it is difficult to implement generality without first implementing and testing lower-level components. Therefore, as a stepping stone for a starting point, we defined a specific task that we aimed to accomplish: having Baxter assembling a coffee table with the help of human. (Figure 1 shows the workflow of this system.) Having such an aim allowed us to physically manifest an implementation and test of our system.

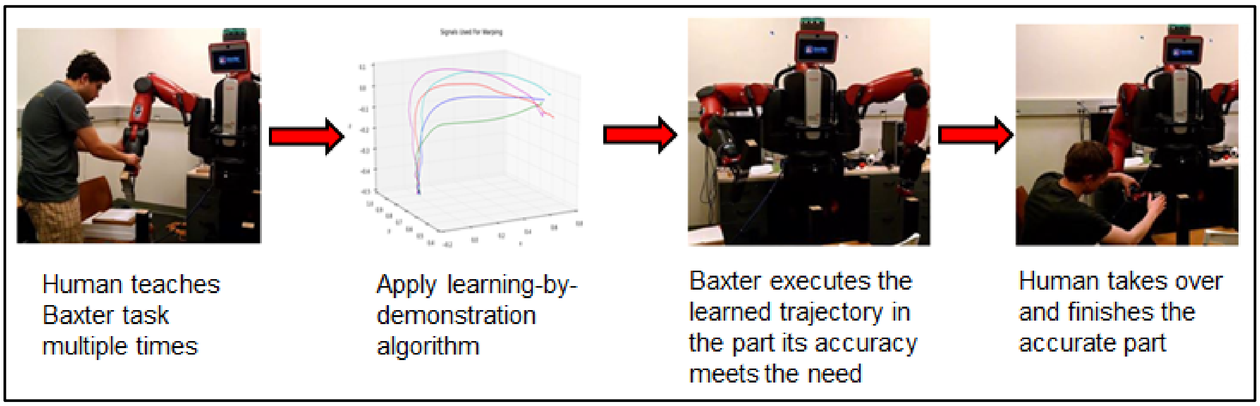


Figure 1: Workflow of System Process

5 TECHNICAL CONTRIBUTIONS / 6 CONCLUDING REFLECTIONS

Each team member has written a separate paper for these two sections. See the related documents to this report. (Figure 2 summarizes the scope of each paper and organizes the system components in a block diagram.)

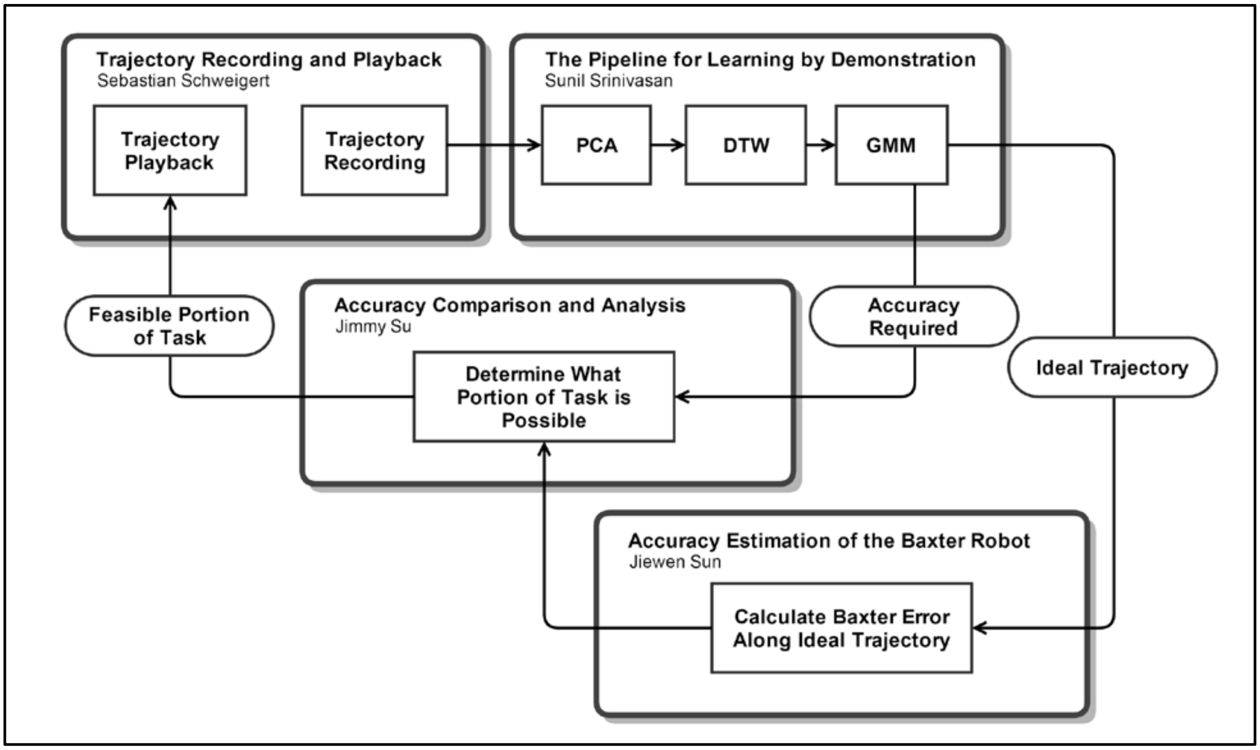


Figure 2: Paper Organization

References

(Strategy Paper References:)

Bi, F. & Mac, R. (2015). Drone Maker 3D Robotics Raises $50 Million in Latest Round. *Forbes*. Retrieved from http://www.forbes.com/sites/frankbi/2015/02/26/drone-maker-3d-robotics-raises-50-million-in-latest-round/

Biesada, A. (n.d.). Hoover's Company Profiles: iRobot Corporation. *Hoover's, Inc.* Retrieved from http://subscriber.hoovers.com/H/company360/fulldescription.html?companyId=132607000000000

Brault, M. (2012). Americans With Disabilities: 2010 - Household Economic Studies. *Economics and Statistics Administration, US Department of Commerce*. Retrieved from http://www.census.gov/prod/2012pubs/p70-131.pdf

Censky, A. (2011). Older Americans are 47 times richer than young. *CNN Money - The New American Dream*. Retrieved from http://money.cnn.com/2011/11/07/news/economy/wealth_gap_age/

Crompton, J. (2014). IBISWorld Industry Report 33399: Power Tools & Other General Purpose Machinery Manufacturing in the US. *IBISWorld.* Retrieved from http://www.ibis.com

Ellis, B. (2013). Nursing home costs top $80,000 a year. *CNNMoney (New York)*. Retrieved from http://money.cnn.com/2013/04/09/retirement/nursing-home-costs/

Hoover's, Inc. (n.d.). Hoover's Company Profiles: Clearpath Robotics Inc. *Hoover's, Inc.* Retrieved from http://subscriber.hoovers.com/H/company360/overview.html?companyId=244260399

Hoover's, Inc. (n.d.). Industry Report: Home Health Care Services. *Hoover's, Inc*. Retrieved from http://subscriber.hoovers.com/H/industry360/overview.html?industryId=1383

McMillan, R. (2015). Now We Can Build Autonomous Killing Machines. *Wired*. Retrieved from http://www.wired.com/2015/02/can-now-build-autonomous-killing-machines-thats-bad-idea/

Owano, N. (2011). Willow Garage slashes price (and arm) of PR2 robot for research. *PhysOrg*. Retrieved from http://phys.org/news/2011-08-willow-garage-slashes-price-arm.html

O'Donnell, F. (2015). Issues and Insights. *Mintel: Senior Lifestyles - US - December 2013*. Retrieved from http://academic.mintel.com/display/689700/

Ortman, J. M., Velkoff, V. A., & Hogan, H. (2014). An Aging Nation: The Older Population in the United States. *Economics and Statistics Administration, US Department of Commerce*. Retrieved from http://www.census.gov/prod/2014pubs/p25-1140.pdf

Porter, M. E. (2008). The Five Competitive Forces That Shape Strategy. *Harvard Business Review 86*(1), 25-40.

Rethink Robotics (2015). Build a Baxter. *Rethink Robotics.* Retrieved from http://www.rethinkrobotics.com/build-a-bot/baxter/

Tobe, F. (2013). A glimpse at our robotic future: 235 start-ups reviewed. *Robohub*. Retrieved from http://robohub.org/a-glimpse-at-our-robotic-future-235-start-ups-reviewed/

(IP Paper References:)

Barajas, L. G., Martinson, E., Payton, D. W., & Uhlenbrock, R. M. (2014). Method and system for training a robot using human-assisted task demonstration. *U.S. Patent No. 8,843,236*. Retrieved from https://www.google.com/patents/US8843236

Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Robot Programming by Demonstration. *Springer Handbook of Robotics*, 1371-1394.

Brown & Michaels, PC. (2006). How do I read a patent? - the Claims. *Brown & Michaels*. Retrieved from http://www.bpmlegal.com/howtopat5.html

Klein, A. R. (2015). Intellectual Property Basics for Technology Companies. *Latham & Watkins*.

Ravindran, B., Barto, A. G., & Mathew, V. (2007, January). Deictic Option Schemas. In *IJCAI* (pp. 1023-1028).

Robotic Manipulation with a Human in the Loop

Master of Engineering Final Capstone Report

Mark Jouppi
mark.jouppi@berkeley.edu

May 9, 2015

Abstract

While humans can naturally feel forces as they manipulate objects, robots need special instrumentation to provide this capability. Force and torque sensing is crucial to enable robots to perform manipulation tasks such as picking up delicate items, screwing in parts for assembly, and handling deformable objects. In this paper, I present a force sensing data acquisition system that I implemented and used with the Baxter Research Robot. I further detail how I have characterized the accuracy of these sensors and experiments I have conducted with the sensors instrumented on the robot. Finally, I describe software and algorithms to make use of the data in controlling the robot.

1 Overview

While people have long dreamed of having robotic assistants helping them in their homes, this vision has seen only limited realization in recent years. For example, robotic vacuum cleaners like the iRobot Roomba have seen significant market adoption due to affordable pricing and meeting customer needs [Jones, 2006]. However, these robots in service of the homes of today have very narrow functionality. While they may excel at one particular task such as vacuuming, they cannot do anything else.

Figure 1: The Baxter robot; image courtesy [Guizzo, 2013]

In the past, general purpose robot platforms have been much too expensive for the vast majority of people to buy for their personal use. For example, the Willow Garage PR2, a large mobile robot with dual 7-degree of freedom arms, costs $280,000 [Willow Garage, 2014]. Furthermore, even if one could afford the robotic hardware platform, one would also need software providing the intelligence to enable the robot to perform a broad range of tasks, operate autonomously, and seamlessly interact with people. Creating such software is an incredibly difficult challenge on the forefront of cutting edge research; it is not by any means a solved problem or ready to be deployed on a commercial product.

Although the software remains a complex problem, there is hope for significantly cheaper hardware. A new paradigm is emerging in which companies are using novel manufacturing processes, parts, and techniques to offer complex robots at significantly cheaper than the status quo. For example, Rethink Robotics offers the dual-arm Baxter robot for only $25,000, an order of magnitude less than the PR2 [Rethink Robotics, 2015b] (see Figure 1). Baxter is made with many innovative, cheap parts. For example, its gears are made of compressed metal powder and it uses series elastic actuators (SEAs) [Guizzo and Ackerman, 2012]. However, as a consequence, Baxter's arms allow for limited precision compared to more conventional, expensive robotic arms. Baxters ability to perform highly sensitive or delicate tasks requiring fine motor skills is thus limited.

Now that the idea of people being able to afford robotic hardware platforms for their homes is a step closer to reality, the question becomes, how do we create software that allows these robots to intelligently help people in a variety of tasks while overcoming any limitations of their cheap hardware? That is, even with innovative, low-cost hardware, better software is still needed. Herein lies the motivation for the capstone project overall. The overarching goal of the capstone project is to develop software that will make low-cost robots like Baxter truly useful to people in the home.

The main idea of the capstone project is to develop a framework in which Baxter can learn how to perform tasks via human demonstrations, which it will then analyze to determine the accuracy needed at each step. A simple way of doing this would be to analyze the variance in task trajectory signals across each demonstration and conclude that task segments with tiny variance required greater accuracy. For task segments that required greater accuracy than the robot possess, the robot could then summon a human to position its end effector under "virtual fixtures" [Abbott et al., 2007]. Alternatively, the robot may be able to resort to more traditional techniques such as visual servoing to gradually hone in on the target [Espiau et al., 1992].

Other members of the capstone team will detail various components of the project such as the trajectory recording system, learning framework, and Baxter accuracy determination. This paper will focus specifically on force sensing. The main learning system of the project has thus far utilized training trajectory signals consisting of position and orientation of the end effector over time, which are used to synthesize a Cartesian trajectory to execute on the robot.

However, while position control is effective for moving a robot through free space, it is often not adequate for tactile or contact tasks [Craig, 2005, p. 317]. For example, as our primary demonstration application, the capstone team uses the task of assembling IKEA furniture. Consider the task of a robot screwing in a table leg onto the main table body (see Figure 2). In this case, the robot must press the table leg down into the main piece while twisting it. That is, it must apply a downward force orthogonal to the table surface while also applying a torque about the axis of the table leg. Clearly, relying on position control in such a task would result in failure as this is fundamentally a task constrained by forces and torques.

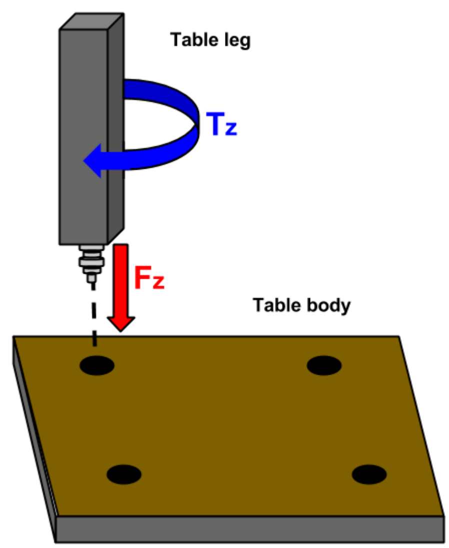
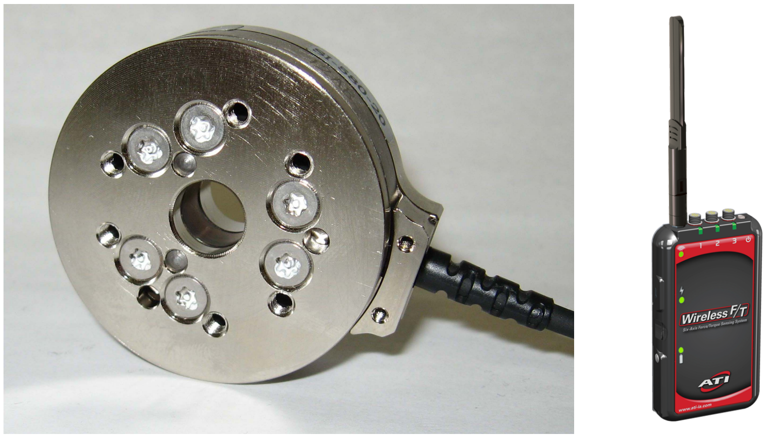


Figure 2: Screwing in a table leg

Therefore, I argue that the trajectory recording system developed for this project must be augmented to record measured forces and torques throughout a task training demonstration, and the learning system must be modified to use additional features in its trajectory signals consisting of these forces and torques. This would enable the robot to learn not only the end effector configurations over time it should generalize from, but also the forces and torques important to carrying out a task. Then, at test time, the robot would use a closed loop controller to make sure that at each task segment, the forces and torques it is exerting are equal to those it had learned from training.

In this paper, I present a force and torque sensing data acquisition system I have created to satisfy these needs. I use ATI Industrial Automation force/torque sensors in addition to



(a) Mini45 Force/Torque sensor

(b) Wireless sensor hub

Figure 3: ATI Industrial Automation force/torque sensing hardware; images courtesy [ATI Industrial Automation, 2015a], [ATI Industrial Automation, 2015b]

a wireless hub device, so the sensors can be instrumented on the robot without unnecessary wiring to the external control computer (see Figure 3). I proceed to discuss experiments I have conducted with the robot and sensors. Finally, I detail software and algorithms to make use of this data in controlling the robot.

2 Literature Review

In this section, I present a brief overview of relevant literature to give the reader context and background for the project specific content that follows.

2.1 Force sensing background

Force sensing for robotics applications goes back to the 1970s and 1980s, during which researchers were primarily interested in developing new force sensing technologies and devices [Tegin and Wikander, 2005]. Early applications included force sensing for industrial robots performing grinding and deburring manufacturing tasks [Craig, 2005, p. 318]. Recent focus in the field has shifted more towards better understanding and modeling force-torque data as well as exploring new applications such as robotic surgery [Tegin and Wikander, 2005]. The field is still in its relatively early stages with much work to be done; thus far, more research has gone into other perception schemes such as vision [Tegin and Wikander, 2005]. Over time, work in the field has split into two approaches; force sensing, which focuses on measuring and using overall net forces at the end effector, and tactile sensing, which focuses on instrumenting end effectors with multiple fine-grained sensors for more detailed analysis of the object such as its shape and other properties [Bicchi et al., 1993].

There are many different types of force sensors, including joint sensors, wrist sensors, and finger sensors [Craig, 2005, p. 253]. Joint sensors measure the torques that motors at each joint apply as the robot moves.

Wrist sensors are mounted between the end effector and the wrist, and generally provide three measurements for each direction of force, and three measurements for each direction of torque [Craig, 2005, p. 253]. This is useful for determining the forces between the robot's end effector and the object it is contact with. Wrist force-torque sensors are generally constructed using several internal highly-sensitive strain gauges placed around the center of the device, which can be used to figure out the forces and torques for each axis [Garcia et al., 2009]. I employ one such type of wrist force-torque sensor in this project.

A variety of finger sensors have been developed including tactile array sensors that measure forces over a finger's surface, finger joint angle sensors, and force-torque sensors mounted at the tip of each finger [Howe, 1993]. These sensors are vital to approaches focusing on tactile sensing to determine the shape, deformation, and rigidity of objects the robot is grasping.

2.2 Force sensing computations

Suppose that we want to direct the Baxter robot to press down with $-c$ N of force in the Z direction and twist with k Nm of torque about the Z axis to screw in a table leg (Figure 2).

Given our desired vector of end effector forces and torques, $f = \begin{bmatrix} f_x & f_y & f_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & -c & 0 & 0 & k \end{bmatrix}^T$, how can we compute the torques, $\tau_{joints} = \begin{bmatrix} \tau_1 & \cdots & \tau_7 \end{bmatrix}^T$, Baxter's motors at each joint would need to apply to generate these desired forces at the end effector? Henceforth, I will refer to these six dimensional force-torque vectors as "wrenches" [Murray et al., 1994, p. 61].

To solve this problem, we can use the Jacobian. In general, the Jacobian is the gradient of a multidimensional function. For robotics in particular, the Jacobian refers to the derivative with respect to time of the forward kinematics map [Murray et al., 1994, p. 115]. One can define a spatial or body Jacobian depending on if one wants to consider the base frame or end effector frame perspective, respectively. For this project in particular, we are interested in the body frame since we want to be able to reason about the forces the end effector experiences. The following equations show how to compute the body Jacobian from a forward kinematics map represented in product of exponential twists form, in the notation of Murray, Li, and Sastry:

$$g_{st}(\theta) = e^{\hat{\xi_1}\theta_1} \cdots e^{\hat{\xi_n}\theta_n} g_{st}(0) \tag{1}$$

$$J_{st}^b = \begin{bmatrix} \xi_1^\dagger & \cdots & \xi_n^\dagger \end{bmatrix} \tag{2}$$

$$\xi_i^\dagger = Ad^{-1}_{(e^{\hat{\xi_1}\theta_1} \cdots e^{\hat{\xi_n}\theta_n} g_{st}(0))} \xi_i \tag{3}$$

where $g_{st}(\theta)$ is the transformation between the tool (end effector) frame t and base frame s with joint angles $\theta = \begin{bmatrix} \theta_1 & \cdots & \theta_n \end{bmatrix}^T$, $g_{st}(0)$ is the same transformation but at initial configuration (i.e., the robot's arm is stretched out with each joint angle $\theta_i = 0$), ξ_i is the twist for joint i, and Ad^{-1} is the inverse adjoint [Murray et al., 1994, p. 115-117]. Equation 1 defines the forward kinematics map (for a given set of joint angles θ, how to transform between tool frame and base frame). Equations 2 and 3 define the body Jacobian.

In general, the Jacobian J is a d by j matrix where d is the dimension of the robot's workspace (i.e., up to 6 degrees of freedom), and j is the number of joints the robot has [Craig, 2005, p. 150]. Note that if the robot has $j > d$, as is the case for the Baxter robot with a 6 degrees of freedom workspace and 7 joints, it is "kinematicaly redundant", which means that there may be multiple solutions for the robot to achieve a given end effector wrench; this added flexibility is important for working around obstacles, avoiding singularities, and choosing motion plans that optimize given criteria (e.g., minimize energy expended over motion) [Chiaverini et al., 2008]. The Jacobian is important because it is often used as a mapping from velocities in joint space to velocities in end effector space. Furthermore, as utilized in this project, the Jacobian is useful in static force analysis as it establishes a mapping between an end effector wrench and a set of corresponding joint torques.

Once we have the Jacobian, we can answer the question initially posed. That is, given some desired end effector wrench, we can calculate the joint torques needed to produce this at the end effector:

$$\tau = (J_{st}^b)^T F_t \tag{4}$$

where $\tau = \begin{bmatrix} \tau_1 & \dots & \tau_n \end{bmatrix}^T$ is the vector of joint torques, $(J_{st}^b)^T$ is the body Jacobian transposed, and $F_t = \begin{bmatrix} f_x & f_y & f_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^T$ is the end effector wrench [Murray et al., 1994, p. 121]. In the case of Baxter, τ is 7x1, J_{st}^b is 6x7 (so the transpose is 7x6), and F_t is 6x1. Thus, the dimensions all agree.

2.3 Force sensing and control

Suppose the robot computes the joint torques it needs to produce a desired end effector wrench and then applies those joint torques. How should the robot then respond if the subsequently measured end effector wrench is still a bit off? Control algorithms are necessary to make sure the robot uses its force sensing measurements in real time as feedback to adjust its joint torques. Closed loop control refers to systems such as this that use feedback to adjust the output. Open loop control refers to systems that make no use of feedback.

One simple technique that could be implemented for the purposes of the project would be a Proportional-Integral-Derivative controller (PID controller), which computes an error signal from the difference between the desired output and the current output, and then uses the current error, the sum of past errors (integral), and rate of change of error (derivative) as feedback to modulate the control variable [Aström and Hägglund, 1995]. In this project, such a scheme would amount to computing the error signal between the desired end effector wrench and the measured end effector wrench, of which the current error, past errors, and rate of error change would be used to update the joint torques to get closer to the desired end effector wrench. Open loop control for this project would entail repeatedly sending joint torque commands, but not using feedback from the measured forces and torques to alter those commands.

More advanced controller architectures used in robotics include methods that combine both position control and force control [Raibert and Craig, 1981]. These hybrid controllers can be used to establish control based on forces in directions where forces matter most (e.g., along screwing axis in Figure 2) and position control for everything else [Garcia et al., 2009].

3 Methods

Now that the conceptual groundwork and relevant background has been established, I will describe specific hardware and software tools I use for the project.

3.1 Baxter Research Robot

Baxter is a robot designed by Rethink Robotics as an alternative to traditional huge, expensive, and dangerous industrial factory robots. Baxter was created specifically to be low-cost, safe around people, and easy and flexible to setup and train for different tasks, aimed at factories and manufacturing roles [Rethink Robotics, 2014a]. The robot has two arms, each of which has seven degrees of freedom. The robot possesses cameras in its wrists and head, as well as sonar sensors around its head. The robot is stationary as it does not possess any mobile drive system. The Baxter Research Robot is a variant of the original Baxter robot that is specifically aimed at research labs, academia, and educational users [Rethink Robotics, 2014b]. The Baxter Research Robot comes with an open source software development kit (SDK) and utilizes the Robot Operating System (ROS) to give researchers and students a great foundation of core software to build upon [Rethink Robotics, 2014b].

3.2 Force sensing hardware

For force sensing instrumentation, I use ATI Industrial Automation Mini45 force-torque transducers, which use internal strain gauges to compute the forces and torques on the sensor (see Figure 3) [ATI Industrial Automation, 2015c]. I also use the recently released ATI Industrial Automation WNET device, which is a wireless hub that connects to the transducers via cables and streams their data to a network. The WNET also provides a serial interface to change lower level device parameters, calibration data, and transducer settings [ATI Industrial Automation, 2015c].

3.3 Robot Operating System

The Robot Operating System (ROS) is an open source robotics framework that enables natural development of software modules that communicate between one another for robotics applications [Quigley et al., 2009]. ROS applications take the form of a *computation graph* consisting of *ROS nodes*, which are running processes that communicate over edges of the graph, called *ROS topics* [Cousins, 2010]. ROS contains many libraries ranging from low level robot hardware and sensor drivers to high level path planning algorithms and as well as software build tools [Cousins, 2010]. ROS was very useful to the team as it provided a solid foundation of core software and existing functionality to build upon and provided a means to program the Baxter robot, where many lower level details were abstracted out. As is often the case with open source software, there were sometimes problems with ROS and a lack of adequate documentation to address them online, but these issues were solved eventually. Overall, ROS was a crucial piece in the project and definitely went a long way towards giving the team a solid foundation to start from.

4 Results

I now present the results of the project as they pertain to force sensing, and discuss challenges and outcomes.

4.1 Data acquisition system

A core piece of this project, especially from an engineering work standpoint, was just to get the data pipeline from the force-torque sensors to ROS on Baxter's developer workstation established. The ATI sensors are third party devices, not supported by Rethink Robotics, so there is no built-in or default way of installing them on Baxter. To circumvent this issue, we bypass the Baxter onboard control computer. This embedded computer interfaces with all

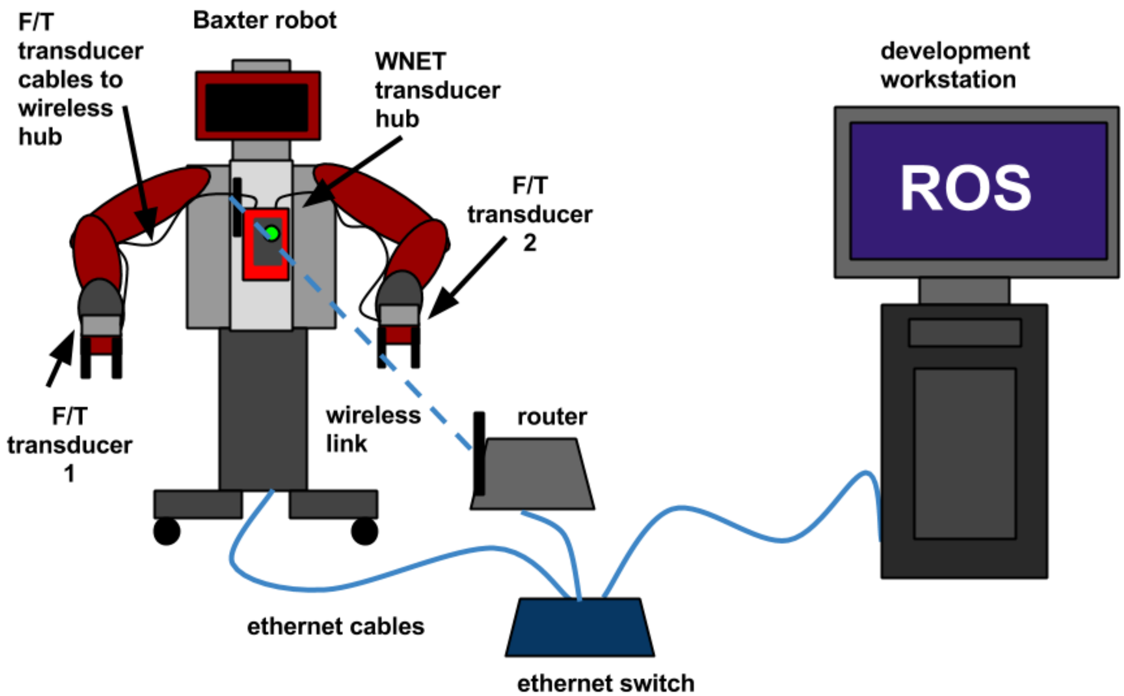


Figure 4: Physical/networking setup of data acquisition system

of Baxter's built-in sensors and actuators and connects to the workstation via an Ethernet link. This computer is bypassed using the WNET device onboard Baxter, which connects to the force-torque sensors and streams sensor data to the workstation over WiFi (see Figure 4).

The WNET device sends UDP packets containing data samples from each time frame to the network. ATI provides sample Java code to run on the workstation that handles reading UDP packets and parsing/unpacking the data into a readable form. I started with a stripped-down version derived from this sample Java code that read packets and wrote the data to CSV files. I made modifications to further eliminate unnecessary functionality and to only send data I was interested in (e.g., leave out the health status info of the device, just keep the force-torque measurements). I also wrote networking code so that instead of writing to a CSV file, the program writes the force-torque measurements in strings of a predefined

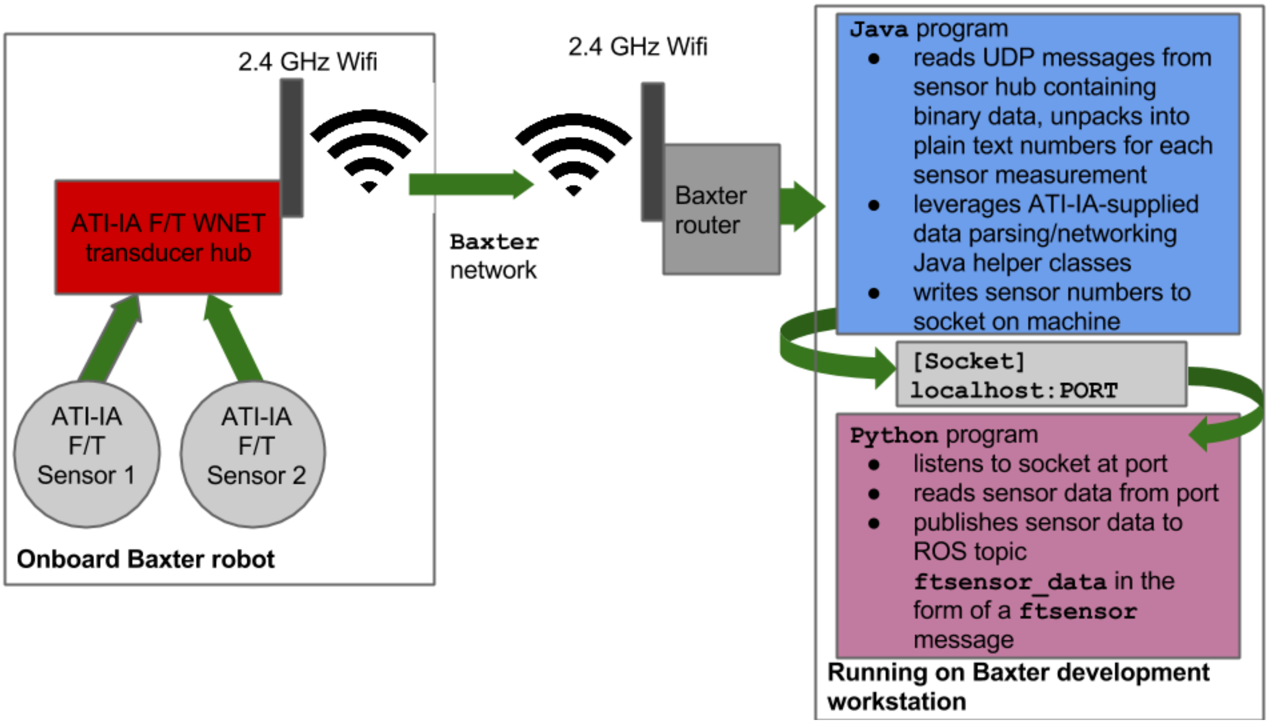


Figure 5: Software setup of data acquisition system

format to a socket on the machine at a fixed port.

I wrote a Python program that listens to this socket at this port, reads the data strings, and using the established convention, parses the string to obtain the sensor values. The program then publishes the sensor values to the ROS topic `ftsensor_data` in the form of a `ftsensor` ROS message, which contains the six force/torque measurements for each of the two transducers. Other ROS nodes can now subscribe to the `ftsensor_data` ROS topic and use the information in making decisions to determine the robot's actions. A diagram of the overall software system is shown in Figure 5.

While the above data acquisition system is conceptually simple, in practice, it took significant work to get running reliably, primarily due to problems with the ATI force sensing hardware. In particular, getting the WNET device to work properly required extensive debugging and interfacing with the device over a serial connection to send different lower level commands to change various hardware settings. For example, initially, there were a myriad of issues getting the device to connect to a wireless network. This was compounded by ATI demo applications that crashed upon launching as well as a lack of troubleshooting documentation. Other problems included the device not recognizing its transducers and the force-torque sensors giving wildly strange numbers due to lack of calibration. All these issues and more had to be solved via hours of debugging the device over the lower level serial interface with much trial and error due to lack of adequate documentation. The author has since written his own documentation and troubleshooting guide so that future generations of graduate students in the lab will be spared the same fate.

4.2 Experiments with force sensors

After completing the force sensing data acquisition software, I physically installed the sensors on the robot and performed experiments to verify the correctness of the data. I show an F/T sensor installed on Baxter's wrist in Figure 6. While originally, the base of Baxter's wrist would connect directly to the gripper, a module consisting of the transducer sandwiched between two custom mounting interface plates now lies between the wrist base and the gripper. The UC Berkeley Tele-immersion lab designed these plates and had them machined as no commercial product existed.

After loading all the calibration data specific to each sensor unit, I tested the accuracy of the transducers using known calibration weights. For simplicity, this test procedure only measures force in the Z direction as a weight is placed on each transducer, with the weight vector pointing in the negative Z axis of the transducer. Figure 7 shows the results of these tests. The transducers are very accurate with the largest error I saw at around 0.15 N.

However, it is important to note that the transducers are noisy, so when I obtained the measurements, I used an average of 8 samples to help decrease the effect of random noise. Also, the sensors are affected by temperature, so putting a cold metal weight on them can produce measurements that are off. To rectify this problem, I placed a thin plastic layer

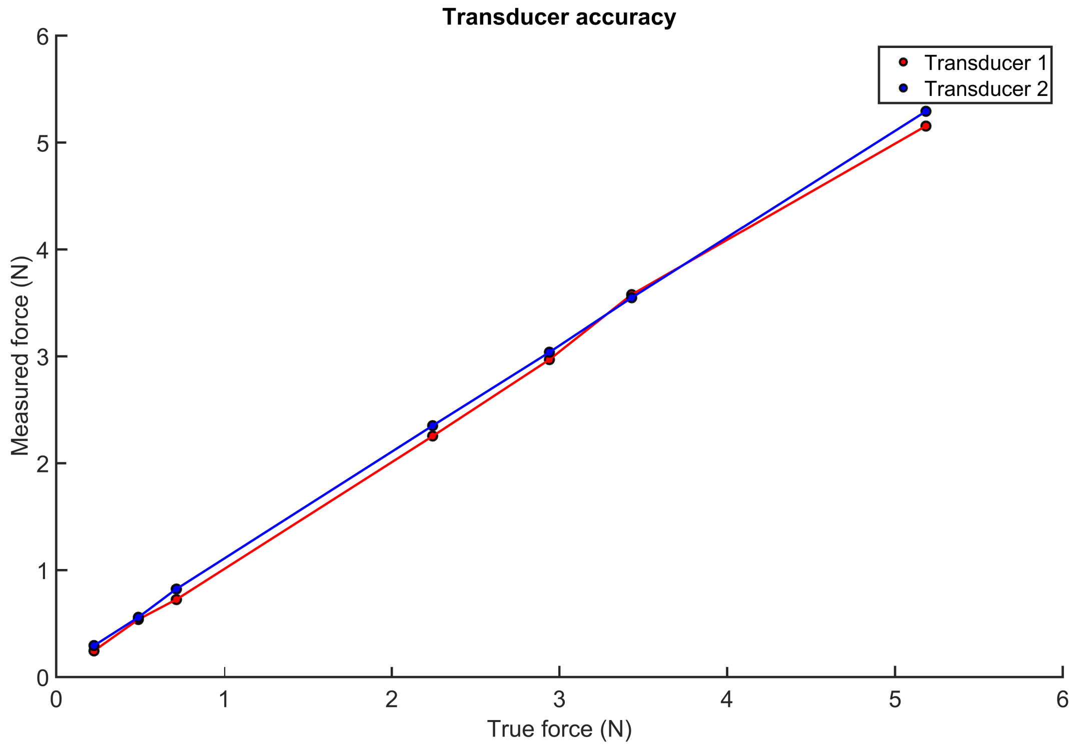
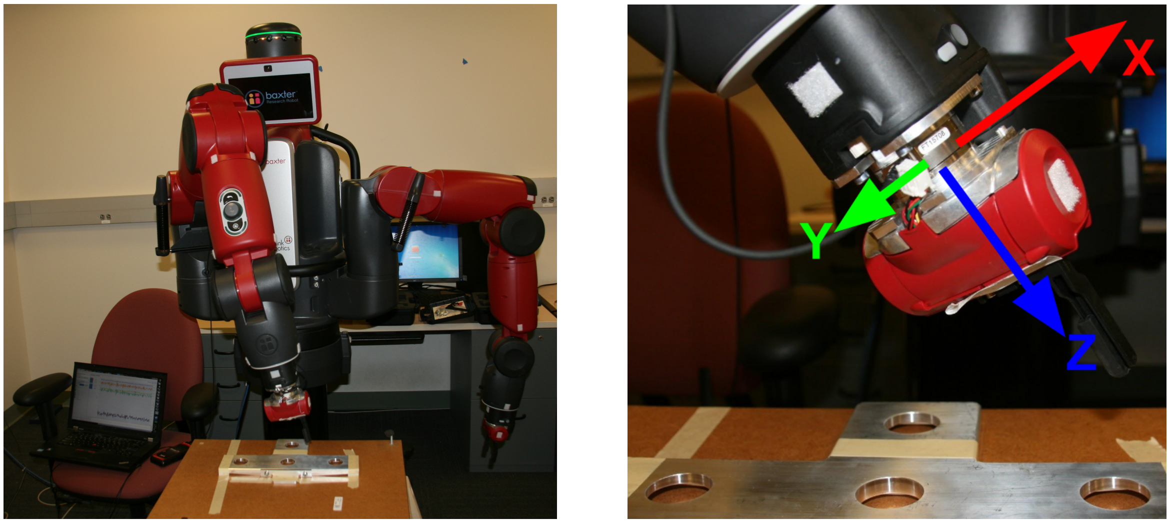


Figure 7: Transducer accuracy tests

between the weight and the transducer and subtracted the known weight of the plastic piece.

After verifying the calibration of the sensors, I performed an experiment inspired by the canonical peg in hole robotics problem, in which a robot is tasked with inserting a peg into a hole, typically by using force sensing to determine how it must adjust the peg to fit into the hole (e.g., [Qiao et al., 1993]).

For my experiment, I drag the robot's end effector across a metal plate with holes, and plot the forces and torques measured during this process. I then show that from the data, one can tell when the end effector runs into a hole versus when it is moving along a smooth surface. The experiment setup is shown in Figure 8a, with the coordinate axes of the F/T sensor shown in Figure 8b. For this experiment, I have removed one of the fingers, such that only one finger is attached to the gripper. In the experiment, I drag the end effector laterally across the metal plate. I present the force and torque data gathered from this experiment



(a) General setup (b) F/T sensor axes

Figure 8: Surface with holes experiment

in Figures 9a and 9b, respectively.

The following time windows summarize what I did in each part of the experiment, with samples and time window numbers corresponding to those shown in Figure 9. The absolute sample values I list in the ranges are approximate.

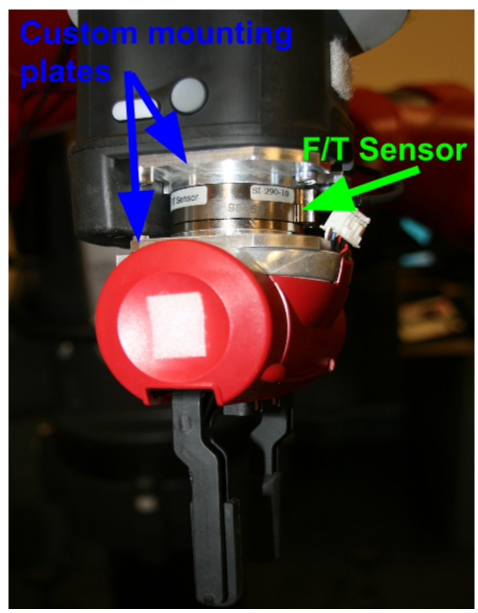


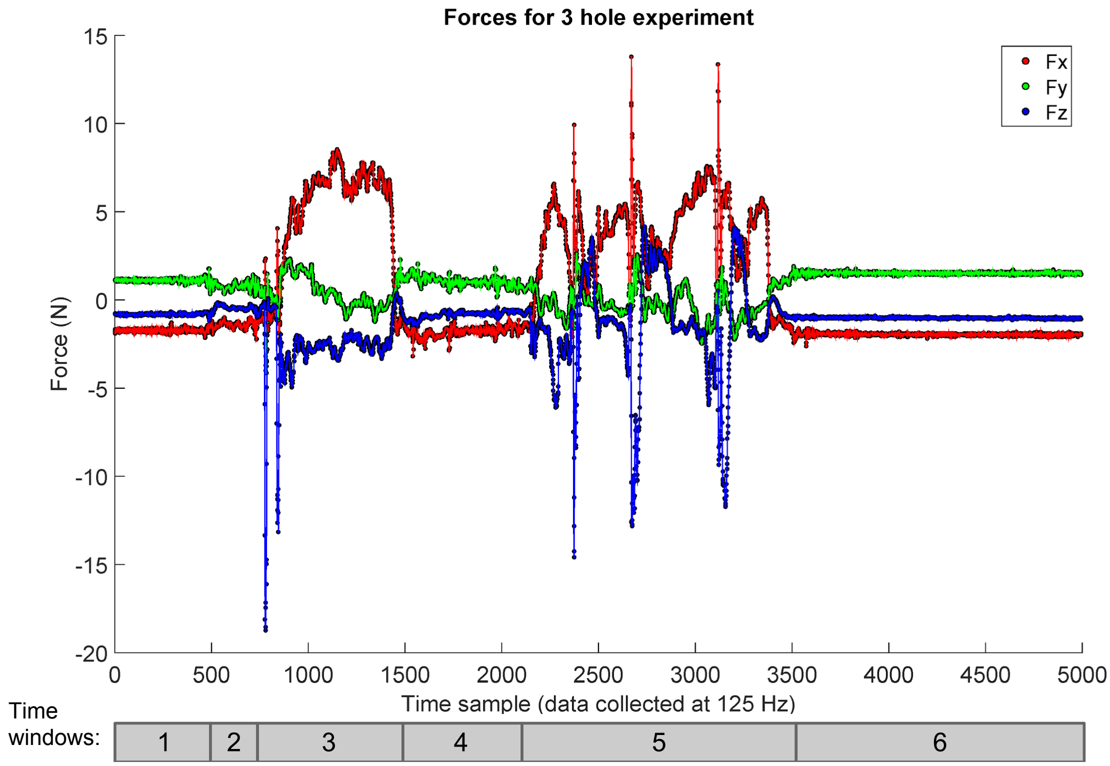
Figure 6: F/T sensor installed on Baxter

Time windows:

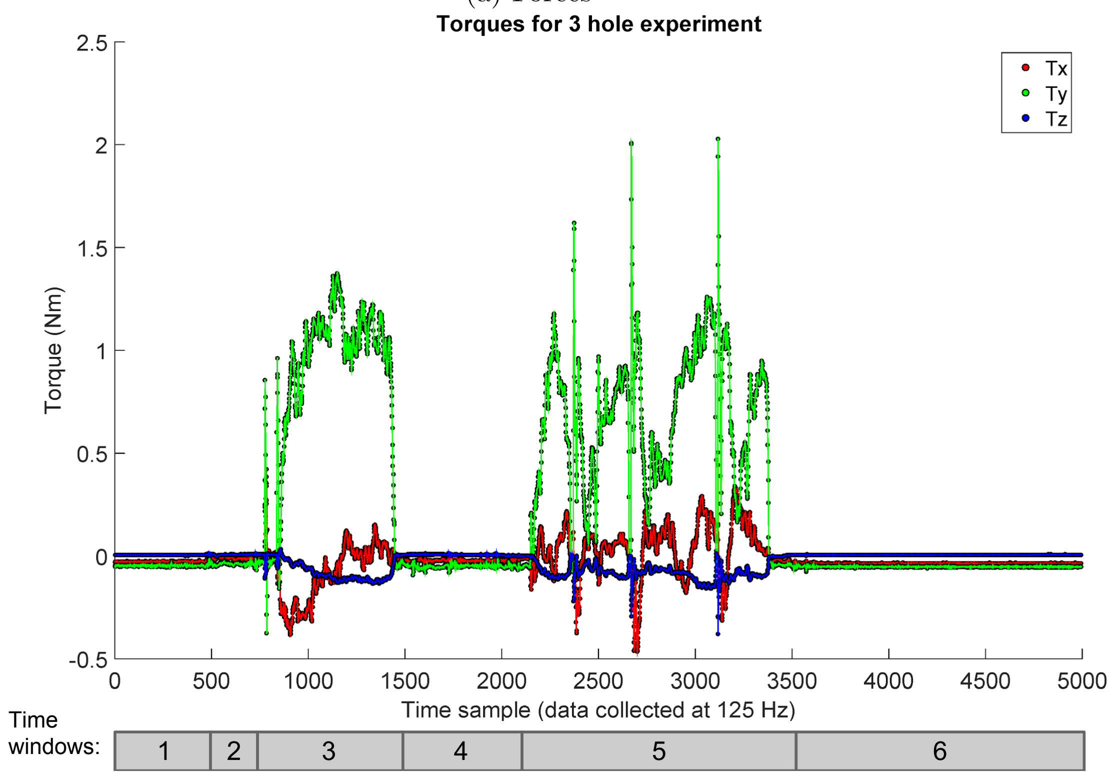
1. Sample 0 to 500: End effector static, floating in free space

2. Sample 500 to 750: End effector moved in free space to hover over start position

3. Sample 750 to 1500: End effector dragged



(a) Forces



(b) Torques

Figure 9: Surface with holes experiment data

across flat surface *without* holes

4. Sample 1500 to 2100: End effector moved in free space to hover over start position

5. Sample 2100 to 3500: End effector dragged across surface *with* three holes

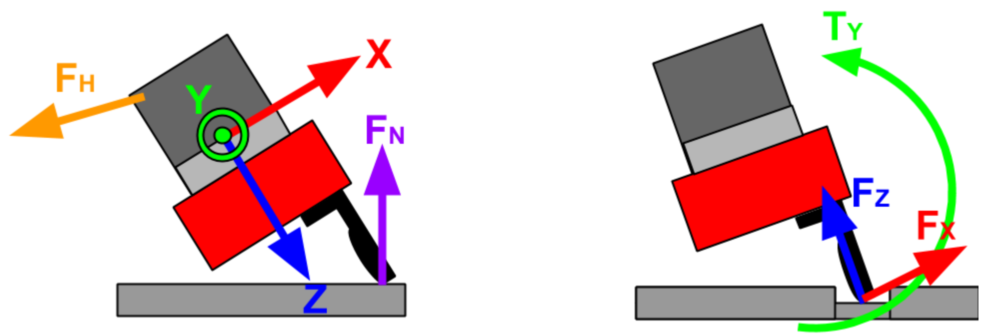
6. Sample 3500 to 5000: End effector static, floating in free space

I will now explain the data and how it shows the above actions. For time windows 1 and 6, the end effector is just floating still in free space without anyone touching it, so as expected, the forces and torques are all about 0 during these periods. I set the bias on the sensor before running the experiment while the end effector was in a similar configuration to ensure that this was the case, although the forces are likely 1 or 2 N off from 0 due to moving the end effector a bit after the bias was set and before running the experiment.

Time windows 2 and 4 consist of moving the end effector through free space to hover over the surface start point. As expected, the forces and torques are about 0, although some small variation is evident, likely as a result of how I pulled on the arm a bit to get it in place.

Now let us examine time window 3, during which the end effector is dragged in a line across part of the metal plate *without* any holes in it. I have drawn a free body diagram of this flat surface scenario in Figure 10a, where F_H is the force the human (i.e., me) applies during the experiment, F_N is the normal force from the surface, and X, Y, and Z are the transducer axes as shown. Note that the human must apply force laterally that is greater than the force of friction so that the end effector will drag along the surface. The force the human applies must also have a downward component to make the end effector press down into the surface a little and to counteract the normal force of the surface. Note that Baxter's arm is gravity compensated (the joint controllers automatically exert the necessary torques to counter the effect of gravity on the arm such that the arm is static under gravity and does not sag), so I do not consider the weight of the arm. From this scenario, we would expect that the transducer would measure force in the positive X direction since the end effector is tilted as shown in Figure 10a, and the surface is pushing back against the gripper in the X direction. We would also expect to measure force in the negative Z direction since a component of the vertical normal force is along this direction. The force along the Y direction should be negligible. We would expect to see negligible torque about the X and Z axes, although there should be a positive torque about the Y axis because there is force being applied to the end of a "lever arm" connected to the transducer origin about the Y axis. The length of this lever arm would be the distance from the transducer to the end of the gripper finger. All these conditions are indeed evident from the data in Figure 9 during time window 3, as expected.

Now let us examine time window 5, during which the end effector crosses over three holes. A diagram of this part of the experiment is shown in Figure 10b. The reasoning for this part of the experiment is similar to that for the flat surface. While the end effector drags over the flat part in between holes, we would expect identical results to the flat surface. When the finger falls into a hole, we would expect to see results with similar direction, but *exaggerated* magnitude. That is, when the finger falls into a hole, we would expect to see a spike in the force along the positive X direction and negative Z direction, since the forces after falling into a hole and bumping the surface should be more severe than usual. There should also be a spike in torque about the positive Y axis for the same reasons. These expectations are indeed clearly met as shown in the data in Figure 9. Furthermore, the three spikes corresponding to each of the three holes are clearly visible in the data for the X and Z forces in Figure 9a and the Y torque in Figure 9b. One can easily imagine building an automated system that uses such force sensing data to determine when the robot's end effector has run



(a) Flat surface

(b) Surface with hole

Figure 10: Free body diagrams for experiment

over a hole.

4.3 Force sensing using data

The next piece of the project was to implement software with the functionality described in Section 2.2 to determine required joint torques for a given desired end effector wrench. I wrote a ROS node that performs this computation and commands the robot with the calculated joint torques. I had to be very careful, as joint torque control is an "advanced control mode", in which the usual safety and collision avoidance features are bypassed through lowest level control access [Rethink Robotics, 2015a]. I implemented an open loop controller that repeatedly computes a Jacobian based on the current arm joint angles, uses that to determine what joint torques it needs to generate a desired end effector wrench, and then commands the arm motors to apply those torques. I performed tests such as specifying a desired end effector wrench with force in the Z direction, and verified that the arm moved up as the end effector exerted force in the Z direction. I also blocked the end effector with my hands and verified that it exerted force against me in the desired direction. My job was made easier thanks to the Baxter SDK, since it already included making adjustments to my joint torque absolute command values to account for gravity compensation [Rethink Robotics, 2015a].

While I was advised to just tackle open loop control for now due to time constraints of the project, the next step would be to implement a closed loop controller such as that introduced in Section 2.3. Additional testing as well as integration with the rest of the capstone project including the trajectory recording system and learning framework would follow.

5 Concluding Reflections

The goal of this particular piece of the project was to build a data acquisition system for force-torque sensing, conduct experiments with this system, and implement open loop force control. These objectives have been accomplished. As this force sensing component relates to the overall capstone project, using this force sensing system to augment the trajectory recording and learning components will allow the robot to learn and execute contact tasks such as screwing in pieces for assembly and determining properties of surfaces through touch.

Overall, this capstone project covered many disciplines including human-robot interaction, sensing, controls, and learning. It was a fun project as well as a very valuable learning experience. Through the project, I have become familiar with many of the exciting advances and new trends in modern robotics, while also building a better understanding of the great many challenges and body of future work that remains to be solved.

While the dream of having intelligent, general purpose robots helping people in their homes is still a distant one, I hope that this capstone project as a whole takes a small step towards bringing that vision to a reality.

Acknowledgments

The author would like to thank advisers Professor Bajcsy and Aaron Bestick for all their advice, guidance, and patience, as well as providing resources such as the Baxter robot and force sensors. The author would also like to thank Sarah Seko for providing an initial stripped-down version of the ATI Java sample code as a starting point and for her help in diagnosing and fixing problems with the force sensing hardware. Thanks to Robert Matthew for designing the custom Baxter transducer mounting plates.

References

[Abbott et al., 2007] Abbott, J. J., Marayong, P., and Okamura, A. M. (2007). Haptic virtual fixtures for robot-assisted manipulation. In *Robotics research*, pages 49–64. Springer.

[Aström and Hägglund, 1995] Aström, K. J. and Hägglund, T. (1995). Pid controllers: theory, design and tuning.

[ATI Industrial Automation, 2015a] ATI Industrial Automation (2015a). F/t sensor: Mini45. `http://www.ati-ia.com/products/ft/ft_models.aspx?id=Mini45`. Accessed: 2015-04-14.

[ATI Industrial Automation, 2015b] ATI Industrial Automation (2015b). New product - wireless f/t. `http://www.ati-ia.com/products/ft/ft_WirelessFT.aspx`. Accessed: 2015-04-14.

[ATI Industrial Automation, 2015c] ATI Industrial Automation (2015c). *Wireless F/T Installation and Operation Manual*. Pinnacle Park, 1031 Goodworth Drive, Apex, NC 27539. Document: 9610-05-1031.

[Bicchi et al., 1993] Bicchi, A., Salisbury, J. K., and Brock, D. L. (1993). Contact sensing from force measurements. *The International Journal of Robotics Research*, 12(3):249–262.

[Chiaverini et al., 2008] Chiaverini, S., Oriolo, G., and Walker, I. D. (2008). Kinematically redundant manipulators. In *Springer Handbook of Robotics*, pages 245–268. Springer.

[Cousins, 2010] Cousins, S. (2010). Welcome to ros topics [ros topics]. *Robotics & Automation Magazine, IEEE*, 17(1):13–14.

[Craig, 2005] Craig, J. J. (2005). *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River.

[Espiau et al., 1992] Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *Robotics and Automation, IEEE Transactions on*, 8(3):313–326.

[Garcia et al., 2009] Garcia, G. J., Corrales, J. A., Pomares, J., and Torres, F. (2009). Survey of visual and force/tactile control of robots for physical interaction in spain. *Sensors*, 9(12):9689–9733.

[Guizzo, 2013] Guizzo, E. (2013). Rethink robotics opens up baxter robot for researchers. `http://spectrum.ieee.org/automaton/robotics/humanoids/rethink-robotics-baxter-research-robot`. Accessed: 2015-04-14.

[Guizzo and Ackerman, 2012] Guizzo, E. and Ackerman, E. (2012). The rise of the robot worker. *Spectrum, IEEE*, 49(10):34–41.

[Howe, 1993] Howe, R. D. (1993). Tactile sensing and control of robotic manipulation. *Advanced Robotics*, 8(3):245–261.

[Jones, 2006] Jones, J. L. (2006). Robots at the tipping point: the road to irobot roomba. *Robotics & Automation Magazine, IEEE*, 13(1):76–78.

[Murray et al., 1994] Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*. CRC press.

[Qiao et al., 1993] Qiao, H., Dalay, B., and Parkin, R. (1993). Robotic peg-hole insertion operations using a six-component force sensor. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 207(5):289–306.

[Quigley et al., 2009] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5.

[Raibert and Craig, 1981] Raibert, M. H. and Craig, J. J. (1981). Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133.

[Rethink Robotics, 2014a] Rethink Robotics (2014a). *Baxter Cutsheet*. 27 Wormwood St, Boston, MA 02210.

[Rethink Robotics, 2014b] Rethink Robotics (2014b). *Baxter Research Robot Datasheet*. 27 Wormwood St, Boston, MA 02210.

[Rethink Robotics, 2015a] Rethink Robotics (2015a). Arm control modes. `http://sdk.rethinkrobotics.com/wiki/Arm_Control_Modes`. Accessed: 2015-04-15.

[Rethink Robotics, 2015b] Rethink Robotics (2015b). Build a baxter. `http://www.rethinkrobotics.com/build-a-bot/baxter/`. Accessed: 2015-02-15.

[Tegin and Wikander, 2005] Tegin, J. and Wikander, J. (2005). Tactile sensing in intelligent robotic manipulation-a review. *Industrial Robot: An International Journal*, 32(1):64–70.

[Willow Garage, 2014] Willow Garage (2014). Order a pr2. `https://www.willowgarage.com/pages/pr2/order`. Accessed: 2015-02-15.