**上海交通大学 密西根学院**

**ENGR4901J/4902J/4903J本科科研课程 结题报告**

**单独工作报告**

**UM-SJTU Joint Institute**

**ENGR4901J/4902J/4903J Undergraduate Research Course Final Report**

**INDIVIDUAL CONTRIBUTION REPORT**

Student Name： Juntao Wu

Student ID Number：522370910038

Major: ME

Project Title: Robotic arm motion control for contact-rich scenarios

Instructor: Youyi Bi

Peg in the whole problem is a long-lasting problem in the field of robotics. It is worth researching mainly because of its extensive use in industry, for example, the assembly of shafts on aircraft and some precise insertion in small device assembly [1].

Before we started our research, I read many previous research works and got an overall understanding of the basic knowledge and research focuses. Long before the development of computer control, people had come up with a method called remote center compliance (RCC)[2]. It does not need positive adjustment but rather depends on the compliance of the “wrist”. Later, with the development of some force sensors (or visual sensors), another trivial way of controlling the robots was developed, which is the “rule-based” method. This simply makes the robot follow a “rule” written by programmers. This is very efficient in the sense that it doesn’t require the machine to make decisions and let physics take over. However, this method soon became undesirable as the complexity of the workspace exponentially grew [3]. Instead, the academia welcomed the latest technology hotspot- machine learning. Machine learning is put forward in 1959 by [Arthur Samuel](https://en.wikipedia.org/wiki/Arthur_Samuel_(computer_scientist)). Its concept is as straightforward as it sounds. It only needs the machine to “learn”. However, it requires much mathematical effort and formulation. In our research, we also decided to leverage the strong tool. We want to use reinforcement learning, which is one of the categories in Machine learning. By designing a punishment function, we can let the machine decide its response. But this is not the only way of machine learning. There are two main categories in using machine learning in peg-in-hole problems: Learning from demonstration and learning from environment [4]. They both have their unique advantages but also bury some flaws. Learning from demonstration means humans teaching the robot, it makes the robot more like a human and thus has a safer learning outcome. However, since it is impossible to cover all of the information and possible situations in the demonstrations, this method is not very generalizable. On the contrary, learning from the environment utilizes reinforcement learning and learns the strategy completely from the robot itself. This is very generalizable but it can be dangerous in the sense that it may develop some absurdly wrong strategy and lead to fatal misconduct.

By the way, I also want to mention about our choice of sensor. There are also many existing solutions when it comes to which kind of sensor to choose [4]. There are force torque sensors (F/T sensors) and visual sensors (cameras). In our research, we decided to use a recently developed tactile sensor – the Papill Array tactile sensor. The sensor consists of a total of 18 contact points and they are in contact with the object when measuring. It will output the force reaction in each contacting point with 0.01 precision and graphs. It is quick and precise compared to F/T sensors on the market. Also, since it has a total of 18 output data, the corresponding accuracy of our machine learning is guaranteed.

In our research, we tend to choose the learning form environment method and use reinforcement learning to train the robot. Needless to say, training a machine needs a huge amount of data and it can cost energy to acquire the set of data. To explore a way to eliminate the huge energy cost and time costs, we are to make the real-world problem into a digital one. There are several benefits of this strategy. First, we can minimize the energy in training the robot arm. Second, the measurement of the parameters in a simulation software turns out to be much easier to carry out. Third, with the strong applications nowadays, simulation provides a more generalizable method in machine learning.

To be more specific, there is some change of idea in our way. We had a shift in our decision of the precise modeling strategy. At first, I was told to learn Unity- a commercial game-making software. We chose it because it has a strong capability to model real-world physics situations like gravity and collision. I watched 80 episodes of tutorials online and managed to create a simulation environment. However, later I found that although the software is ideal for detecting collision and transformation in kinematics, it has its weakness in capturing quasi-static situations just like what we desire to model. And here is what FEA comes into play. Finite element analysis is a modeling method to decompose physical objects into some tiny “elements” and solve the systems of differential equations that relate all of the elements together. Using finite element analysis, force detection became convenient and plausible. And later results proved that we made a good decision. Moreover, I noticed that detecting force reactions in Unity is an unexplored area. We have animation makers working on the problems to visualize the movements of some squashy materials. And their research can be transformed into detecting force using the mathematical framework, to serve our purpose. Typically, I read some work about Professor \_\_, who is an expert in the field. This is potentially my next step target.

To simulate the real-time environment and utilize FEA, I chose to use ANSYS, which is another commercial FEA software expert in FEA modeling. I started with exactly zero foundations and I watched hundreds of hours of tutorials and was finally able to make a simple model using ANSYS workbench. The first difficulty soon arose. The “static structural” mode of the software is specified to solve static analysis. This means that the overall velocity of the object after you add force on it has to be exactly zero. You also cannot change the model too much just to satisfy the requirement for equilibrium and lose the [realisticity](https://www.google.com/search?sca_esv=587451553&sxsrf=AM9HkKne94u3ai3k6XUrcpow3VZolmPO6g:1701605216182&q=realisticity&spell=1&sa=X&ved=2ahUKEwjxjNWznfOCAxXUJEQIHTtoAJwQBSgAegQICBAC). Finally, after I analyzed the model over and over again and tested out several different versions, I came up with the model that “hid” the hole and added force to the peg manually. This is sufficient to reveal the real-world condition and simplify the model. Another difficulty is the difference in the results of the two sides of the sensors. Theoretically, the resulting force on the two sides of the sensors should be identical due to symmetry. However, they had very little difference at the beginning. Although the error is small, it was also intolerable because when you do normalization, some become a maximum and others become minima. After consulting my academic brother, I finally solved this problem.

After the model was complete, now came the question of how to automatically let the computer solve the model and repeat with different conditions applied. There are three solutions for me to choose. The first way is to use the special programming language in Ansys, APDL. It has a different grammar from all the languages I learned and learning it will be very difficult. My second choice is to combine the Ansys workbench and the Python scripting in it, which is known as Ansys scripting. The third way is to utilize the package Pymapdl in Ansys and work the problem without opening the workbench. I finally chose the second method. The modeling is a little complicated with only code. I eventually decided to use SolidWorks to generate the model and workbench automation to collect data.

Finally, I was able to use the software to generalize 1000 data and produce a framework to extract the data. I can now proudly say I have got command of a technique! As it turns out, the struggle and effort I put in is worthwhile. This model I built has a nice generalizability. Especially when it comes to different materials and shapes. With this portal, I can generalize the data for any different situations I want.

I learned so many things in this process, academically and mentally. It is a process from zero to one and may go to infinity. At first, I had no idea what conducting research was like. I never used Ansys and Unity. I didn’t hear of the “peg in the hole” problem at all. However, I managed to survive and made my model in Ansys a framework to extract data, and plenty of experience. I have asked to further my research in Professor Bi’s lab. I want to learn more about utilizing machine learning in robot control and using finite element analysis to do simulations.

References:

[1] Abdullah MW, Roth H, Weyrich M, et al (2015) An approach for peg-in-hole assembling using intuitive search algorithm based on human behavior and carried by sensors guided industrial robot. IFAC 48(3):1476–1481

[2] Chen H, Wang J, Zhang G et al (2009) High-precision assembly automation based on robot compliance. Int J Adv Manuf Technol 45(9–10):999–1006 Park H, Park J, Lee

[3] Tang T, Lin H-C, Tomizuka M (2015) A learning-based framework for robot peg-hole-insertion. In: Proceedings of the ASME 2015 dynamic systems and control conference: Columbus. American Society of Mechanical Engineers, New York, pp 1–5

[4] Ehlers D, Suomalainen M, Lundell J et al (2019) Imitating human search strategies for assembly. In: 2019 IEEE international conference on robotics and automation (ICRA). IEEE, Piscataway, pp 7821–7827