DEVELOPMENT OF A MULTI-DEGREE-OF-FREEDOM PRECISION STAGE WITH SOFT END EFFECTOR FOR 3-DIMENSION SINGLE CELL MANIPULATION

By

KUANG NIE

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

For the degree of

Master of Science

Graduate Program in Mechanical and Aerospace Engineering

Written under the direction of

Qingze Zou & Aaron Mazzeo

And approved by

____

____

____

____

New Brunswick, New Jersey

October, 2019

© 2019

Kuang Nie

ALL RIGHT RESERVED

ABSTRACT OF THE THESIS

DEVELOPMENT OF A MULTI-DEGREE-OF-FREEDOM PRECISION STAGE WITH SOFT END EFFECTOR FOR 3-DIMENSION SINGLE CELL MANIPULATION

By KUANG NIE

Thesis Director:

Qingze Zou & Aaron Mazzeo

This thesis is focused on the creation of a real-time 3-D single cell manipulation system based on pneumatic actuation-based soft end-effector. Single cell manipulation will benefit a lot to cell analysis. Separating a cell from the cell culture will give us a better observation of growth of one cell. Applying automatic robot in cell manipulation will make this process faster and more accurate. In this thesis, a hybrid-soft robotic manipulation system is developed that integrates a soft needle as the end effector for picking-up, translation, and release operation of micro beads with a three-DOF translation stage. This research work combines the exploit and exploration of soft robotics, vision-`based control, real-time control system and MIIC technique. Since our final goal is to manipulate cell by using the devices, resolution, dimension and precision are critical for the design of the device. A high precision 3-axis translational robot arm is selected.

To reduce and avoid damage of the cell membrane during the manipulation process, a soft needle made of PDMS and Ecoflex is designed and fabricated. Membrane attached to the top of the needle is designed to prevent liquid from entering the syringe, which would also be helpful to grasp the cell. Stresses are tested to get the most suitable deformation for grabbing a cell.
Image-based control and real-time control system are coupled together to achieve automatic positioning and moving without inputting any signal during every run. Using of high resolution camera gives the precise position of the target object. Real-time control system helps to cut error during processing and also improve operational efficiency. Also, the MIIC technique is used to achieve precision positioning during high-speed movement. Design, fabrication, motion simulation, motion control and finite element analysis are covered in this research work.

Table of Contents

1. Introduction

1.1 Main ideas

In this thesis, an image-based and real-time control-based device is developed to locate and manipulate small beads (with 3mm diameter). Recently, the cell analysis attracts more interest, and motorized microscopes are more widely used in cell observation and cell analysis. However, most motorized microscopes are only used to observe cells. They have no function of moving cells. Manual operation by using probes, electromagnetic and other methods are inefficient for moving cells since they all need to locate and move cells by person. The process for locating and moving a cell is slow by using manual device. An idea about creating an automatic motorized micro manipulator was motivated, which will improve the speed as well as the accuracy of manipulating a cell.

However, by using pipette tip will damage cells. During the process of connecting the tip with the membrane, the tip may damage the cell wall because of the pressure applied on it. For the So, we plan to apply soft robotics in our design. In this project, we plan to combine soft robotics, vision-based control, robot control and real-time control to create a device which is able to manipulate single cell. In this research, instead of using real cells, we build a prototype to locate and move small beads with 3mm diameters. By combining the computer vision control and real-time control system, the operation can be done more accurate and efficient. After we achieve fundamental capabilities, we applied MIIC technique to make the system faster.

The device we designed combine four main parts. Translation system, which combined with three linear stages, three stepper motors installed with three encoders. Positioning system composed of digital microscope and microscope support. Catching part consists of soft needle made with Ecoflex and PDMS, plastic syringe, vacuum pump and digital valve. The real-time control system we use is the Simulink Real-Time. It's a platform combined with two computers (host PC and target PC), and DAQ card.

1.2 Challenges

- How to catch up small objects by using relatively simple structure. Since the device we planned to design will do action in micro space. There will be great difficulties if we design the structure to be complex, like a soft gripper or mimetic hand.
- How to set up digital microscope to obtain the accurate coordinates of the small beads.
- The dimension of components designed should be fit for the small operation scale.
- How to control the movement of three linear stages and the pressure in the syringe precisely and in correct order in Simulink Real-time.
- How to transfer the coordinates obtained from digital microscope into signal to drive stepper motors.
- The relation between pixel coordinates, real coordinates and resolutions for stepper motors and linear stages need to be unified. The resolution should be precise enough since our final goal is manipulating cells (Diameters are around 50-100 micrometers).
- How to operate system in high precision and also in high-speed.
- How to apply the MIIC technique to this system (Transfer trajectories into signals).

1.3 Contribution

The main contribution for this research is the use of the Simulink Real-time to achieve the real-time control of robot arms to manipulate small objects. By combining computer vision control, real-time control and soft robotics. The device designed can successfully locate target and pick up small beads. And the use of MIIC technique helps the control system helps to make the system run more accurately and faster.

2. Design of the Single Cell Manipulation System

This chapter will show step by step about all the details about the structure to designed.

The figure 2.1 below shows the design of the whole structure.

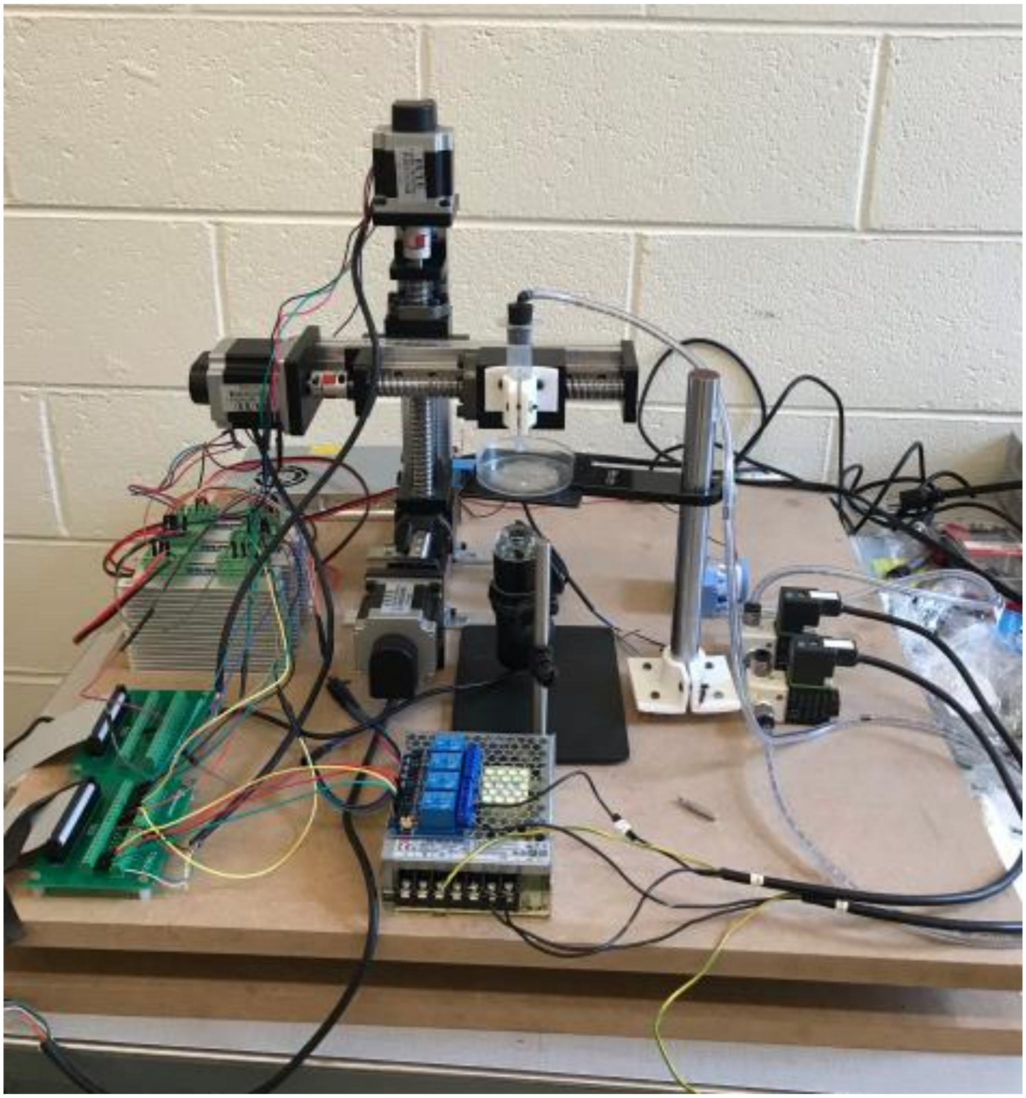


Figure 2.1 Final design of the structure

2.1 Overall Platform Design

Since our goal is to manipulate cells by using robot arms, the design has very high precision requirements. Vibration would affect the precision of moving and positioning. Initially the idea that we consider is to put all the translation components on a small suspend work platform, because according to our design, the sucking part will be above the Petri dish, and we need to put the camera upside down beneath the Petri dish since there’s no more space for the camera. But in this design, the center of the gravity will be high, thus will make the vibration amplitude larger during movement. So we changed our design to put the whole structure on a double layer wood platform with foam between them (As figure 2.2 shows).

Figure 2.2 Damping platform

2.2 Picking

For the picking part, we first consider using soft gripper to handle those beads. For example, in one of Prof Mazzeo’s research, they designed a soft gripper drive by pressurizing embedded channels (Figure 2.3). The diameter of this gripper is about 15

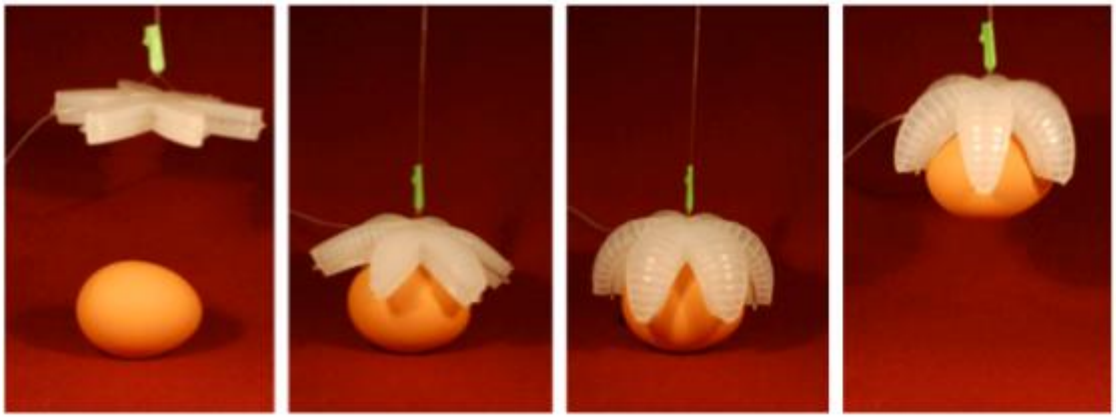


Figure 2.3 Soft gripper (Group Soft Robotics For Chemists)

cm, and the smallest object it could pick has diameter to be about 2.5 cm. And cells we finally want to pick up have diameters to be about 50-100μm, which are much smaller. If we try to pick up such small objects by using soft gripper, the diameter of the gripper should be about 150μm. The outer structure and inner pressurizing embedded channels are hard to be manufactured in such small scale. So we change to choose a simple structure. Sucking would be a simple method compared to grasping. So we design to use the syringe to suck up beads by reducing pressure, and release beads by increasing pressure in the cavity. Syringe will be attached onto the x-direction linear stage. In real situation, cells will be placed in liquid in Petri dishes, and cells are soft. So when we suck cells, they will be sucked into the cavity instead of stay at top of the needle. Liquid will also be sucked into the syringe as well. So it's necessary to design a membrane to prevent cells and liquid being sucked into the syringe. The material that used to make the membrane is Ecoflex-30, which has good extensibility and flexibility. Also in order to reduce the strength applied on the edge of the membrane, instead of directly warping the membrane on the needle, we attach the membrane onto a soft cube with hole made by PDMS (Figure 2.4). And I will talk about the FEA and fabrication of the membrane later.

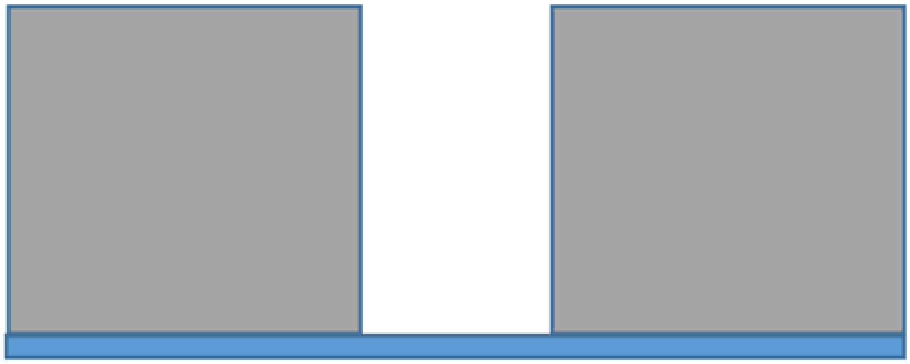


Figure 2.4. The cross section of the soft needle

2.3 Pressure Application Mechanism

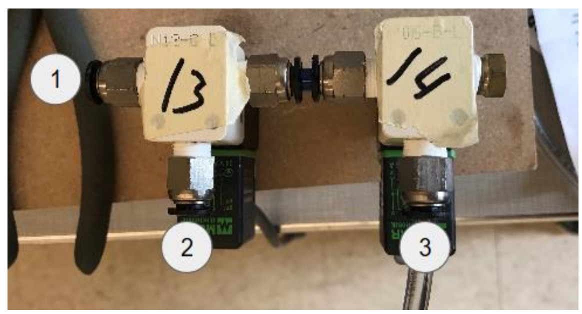


Figure 2.5 Valves to change pressure

Vacuum pump gives negative pressure in the syringe. And the membrane has two conditions: bending inward by negative pressure and flat by zero pressure. Switching the pump on and off may successfully apply two different pressures. But it may take some time for the pump to restart every time, and it may damage the pump as well. So we add one 2-way solenoid valve and one 3-way solenoid valve (figure 2.5) to control the air flow to change the pressure. No.1 pipe connected to the syringe. No.2 pipe goes to atmosphere. No.3 pipe connected to the vacuum pump. When applying negative pressure to the syringe, we open No.1 pipe and No.3 pipe, and close No.2 pipe. When increase the pressure to zero, we open No.2 and No.3 pipes, and close No.1 pipe.

2.4 Mounting Structure Design

Mounting parts for fixing the syringe to the y-stage and microscope support to the platform are made by 3D printing (Figure 2.6).

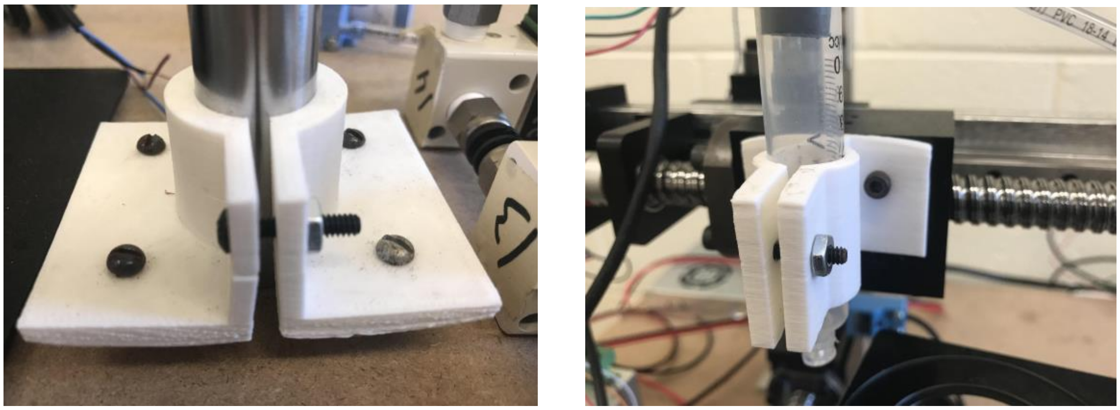


Figure 2.6 Mounting parts

2.5 Positioning Mechanism Design

Positioning part consists of a digital microscope and the microscope support. The camera is set upside down under the Petri dish support as Figure 2.7 shows. We put the digital microscope under the Petri dish because the manipulating mechanism will be above the Petri dish. There’s no more space to place a camera right above the Petri dish. If we still place the camera above the Petri dish, it has to be declining toward the Petri dish. Which we think will influence the accuracy and difficulty of positioning process. So, putting camera under the Petri dish is a good way to give camera a vertical vision of the Petri dish.

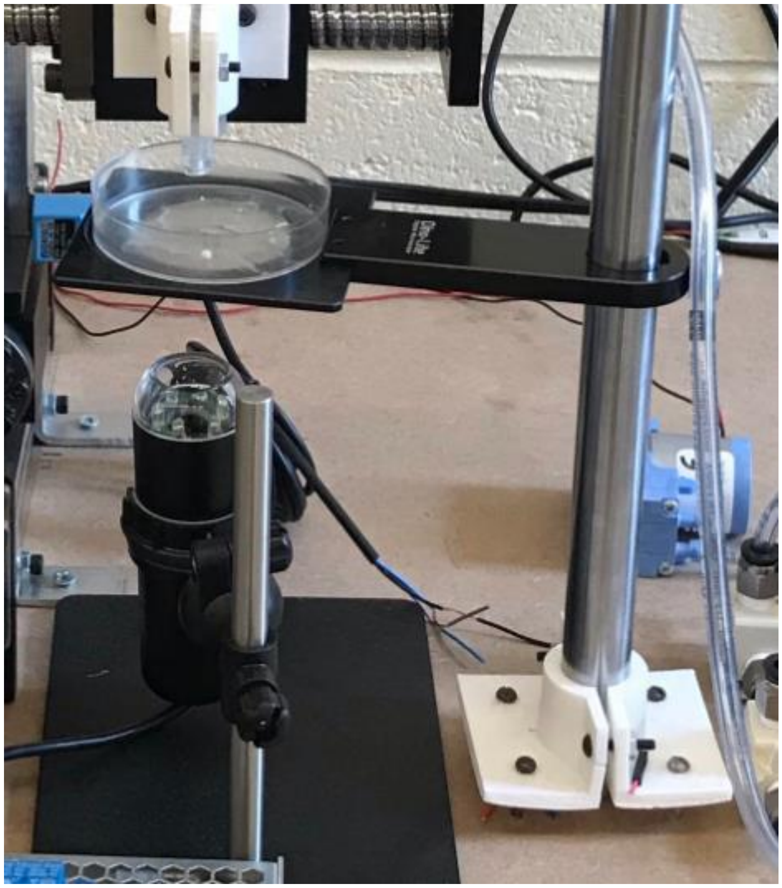


Figure 2.7 Positioning Design

2.6 Translation Mechanism Design

Translation part consists of 3 linear stages. First, we design set y-stage on x-stage (Figure 2.8a) thus we can get more moving space in y-direction. But for this design, the center of gravity will be on the right side of the y-stage, and the gravity force will apply more on the right of the connection than the left. Also, the vibration amplitude of the y-stage will be big. So, we changed our design to be like figure 2.8b.

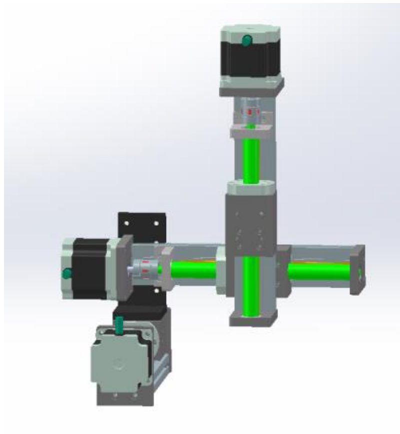


Figure 2.8a

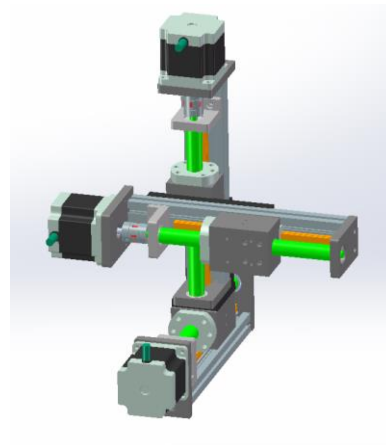


Figure 2.8b

We attached the middle part of the y-stage to the z-stage. This design will get less moving space in xy directions, but it still has enough space (100mm×100mm) for placing one Petri dish (with 70mm diameter). This design can help to move the center of gravity to be closer to the top of the xz mounting part, which will also reduce the vibration amplitude of the y-stage.

2.7 Control Hardware System

We use Simulink Real-Time as the platform for our control system. This real-time control platform consists of two computers: host PC and target PC. The host PC and target PC communicate with each other by an internet cable, and target PC is connected with Humansoft DAQ card (Figure 2.9) which has 8 digital input, 8 digital output, 8 analog input and 6 analog output. to input and output signals. It lets us to create real-time application from Simulink models and run them on the target PC. Figure 2.10 shows the working principle for Simulink Real-Time. We use Simulink Real-Time form Matlab 2017 because from Matlab 2018 vision, we can only use Matlab's PC as the target PC, which is much more expensive. And the screen ratio need to be set as 3:2 to show figures on it.

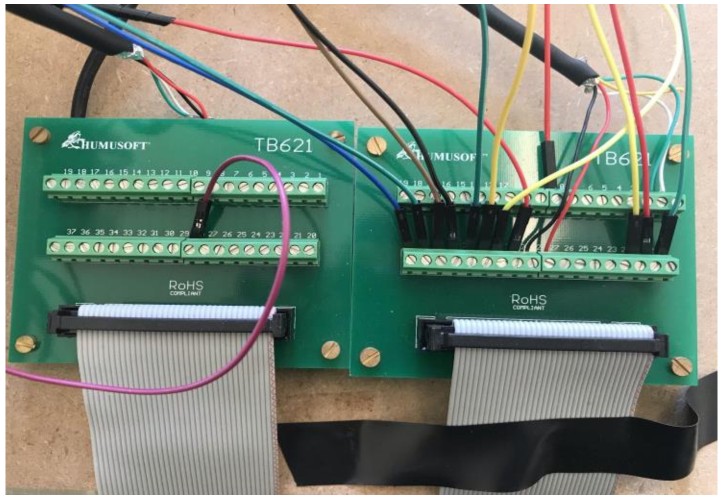


Figure 2.9 Humansoft DAQ card

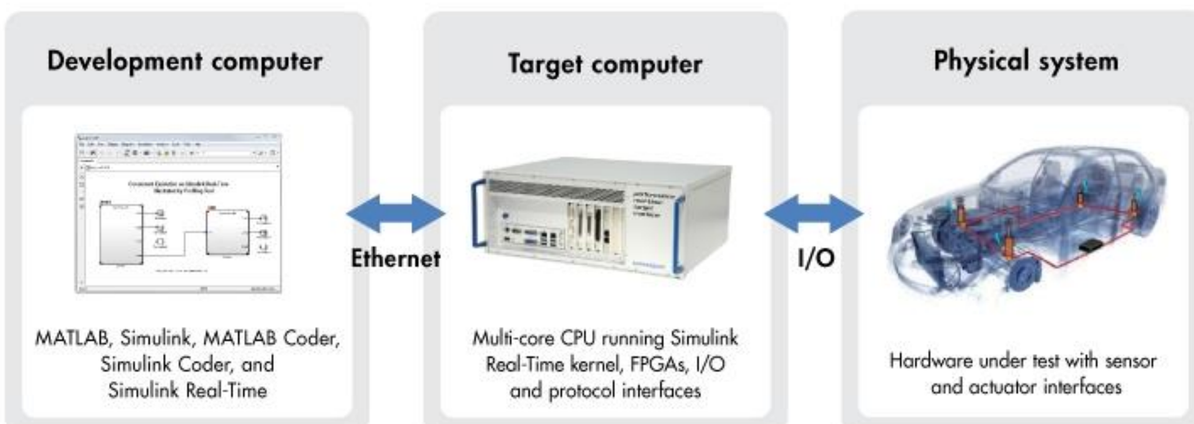


Figure 2.10 Simulink Real-Time working principle

3. Membrane and Soft Needle Fabrication

This chapter will talk about the finite element analysis and fabrication of the membrane and the soft needle detailly. FEA helps to make sure the pressure applying on the membrane makes it deform properly, and the stress will not exceed the tensile strength. A method to fabricate thin layer rubber membrane helps to make the membrane that thin enough for picking up small objects.

3.1 Finite Element Analysis of the Soft Actuation

I used Comsol to did the finite element analysis. The material used for membrane is Ecoflex, and the material used for soft cube is PDMS. Table 3.1 shows the properties of Ecoflex and PDMS. We have

Property	Variable	Value	Unit
Poisson's ratio	nu	0.49	1
Young's modulus	E	100000	Pa
Density	rho	970	kg/m^3

Table 3.1a Ecoflex-30

Property	Variable	Value	Unit
Poisson's ratio	nu	0.49	1
Young's modulus	E	100000	Pa
Density	rho	1070	kg/m^3

Table 3.1b PDMS

two kinds of beads with different diameters. One has 0.3mm diameter, another one has 3mm diameter. The small bead has the diameter almost the same scale as cells (50-100 micrometers), so we first did the analysis about the small beads. I choose the diameter for the hole on the soft cube made by PDMS to be 0.5mm, which is a little bit larger than the bead that we need to pick up. The outer diameter of the PDMS cube is designed to be 2mm, and the thickness of The cube to be 1mm. The deformation of the membrane need to be half-spherical to put the bead into it, so I set the thickness of the membrane to be 1/10 as the diameter of the bead, which is 0.03mm. The sketch of the soft needle is shown in Figure 3.1. First, I set the pressure applied

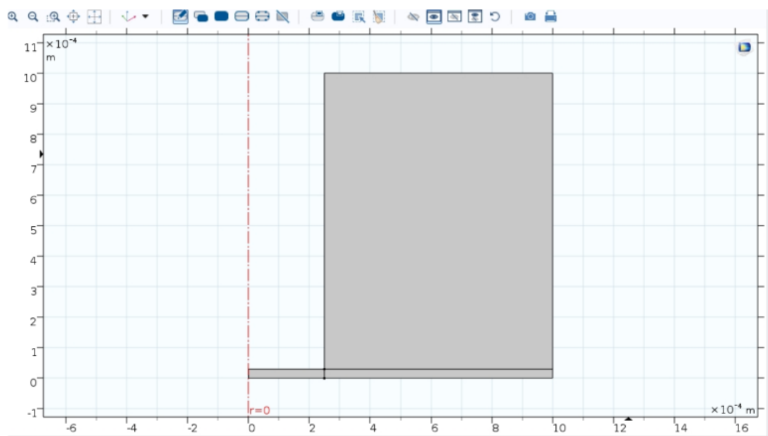
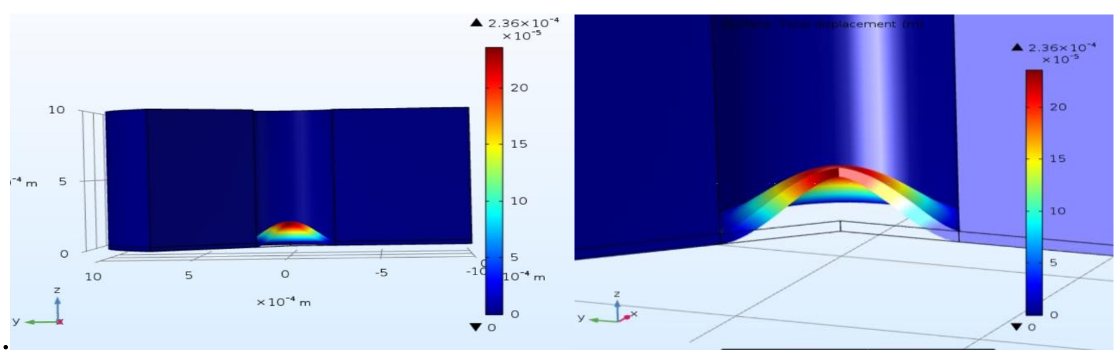


Figure 3.1 Axisymmetric sketch for the soft needle

at the inner surface to be 5000 Pa. The deformation and stress plot are shown as below (Figure 3.2a). The maximum deformation is 0.236mm. And the



.

Figure 3.2a Deformation plot

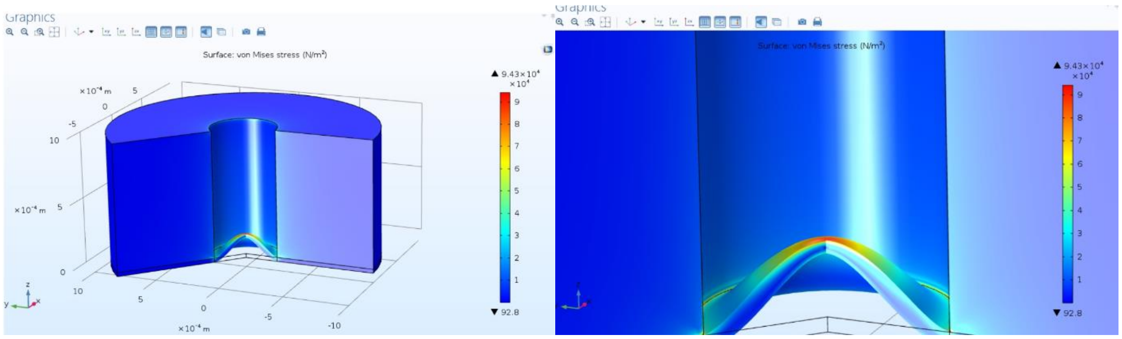


Figure 3.2b Stress plot

maximum stress is 94300Pa, which is far away below the tensile strength of Ecoflex. Then I choose the thickness of the membrane to be fixed (0.03mm), and change the pressure from 3500 Pa to 6000 Pa to see the relationship between the pressure applied on the inner surface and the maximum stress and deformation. And I got the result as table 3.2 shows.

Load (N/m^2)	Stress(N/m^2)	Displacement (mm)
3500	66000	0.165
4000	75400	0.189
4500	84800	0.212
5000	94300	0.236
5500	104000	0.259
6000	113000	0.283

Table 3.2 Relationship between pressure and stress & deformation

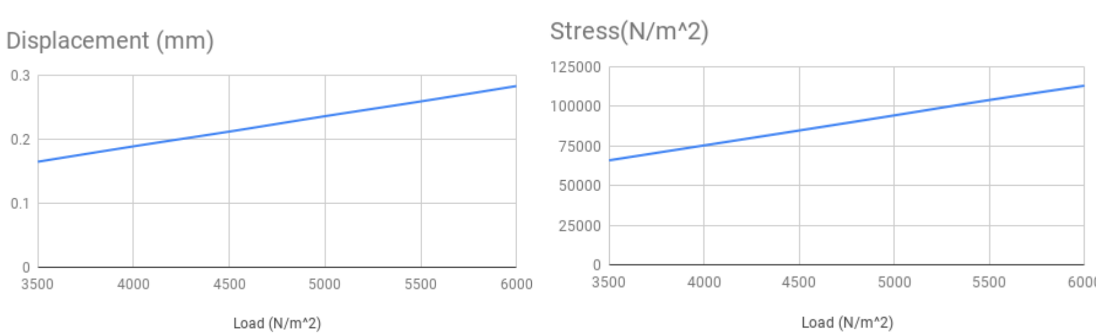


Figure 3.3 Plot about stress & deformation VS. pressure

By drawing the plot about stress & deformation VS. pressure (Figure 3.3) we can see that it's a proportional relationship. And all the stresses are far below the tensile strength of Ecoflex. And the best deformation for the membrane is about 0.2mm (half spherical), when the inner pressure to be about 4000 Pa. After determine the appropriate pressure, I add the bead into the simulation model to see if the deformation of the membrane can actually catch the bead with the pressure change from 4000Pa to 0Pa. As we can see from Figure 3.4, the top surface of the bead and inner surface of the membrane contact with each other when the deformation is half spherical, and this surface force is large enough for catching up the bead because of the little weight the bead has.

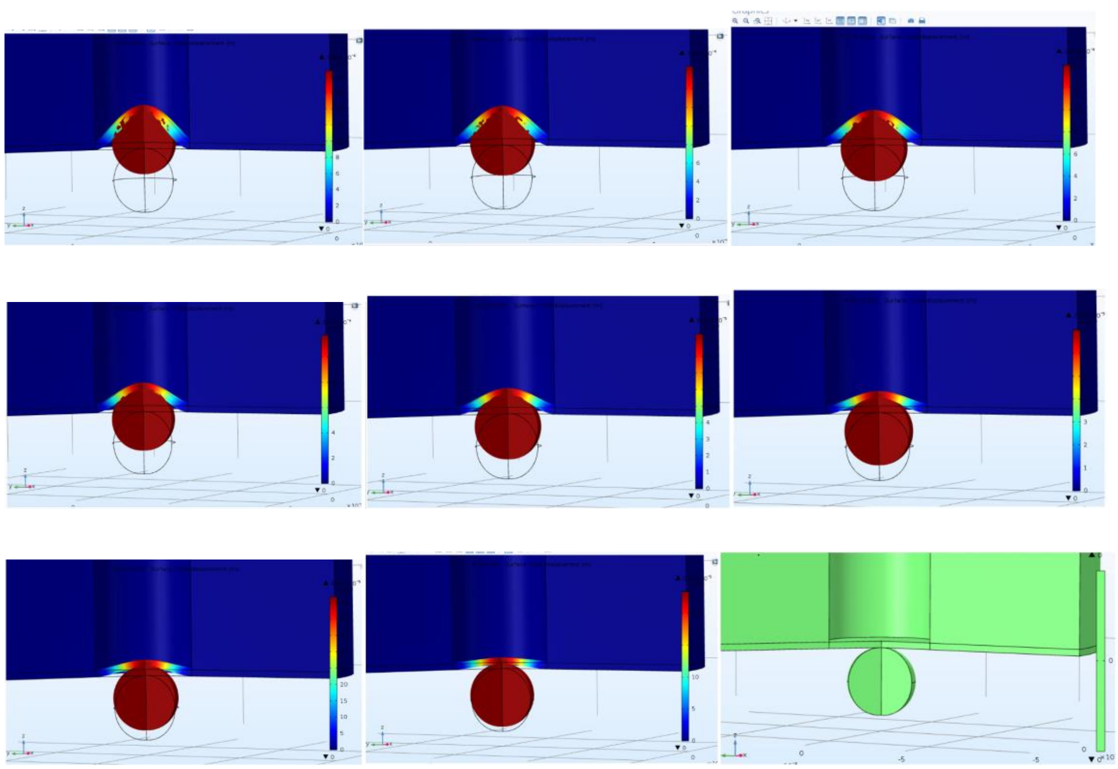


Figure 3.4 Add bead to simulation model

Then I went through the same procedures for the larger beads (3mm diameter). Figure 3.5 shows the deformation when the applying pressure is about 1 PSI, which is half-spherical. So we choose the range of the vacuum to be about 1 PSI.

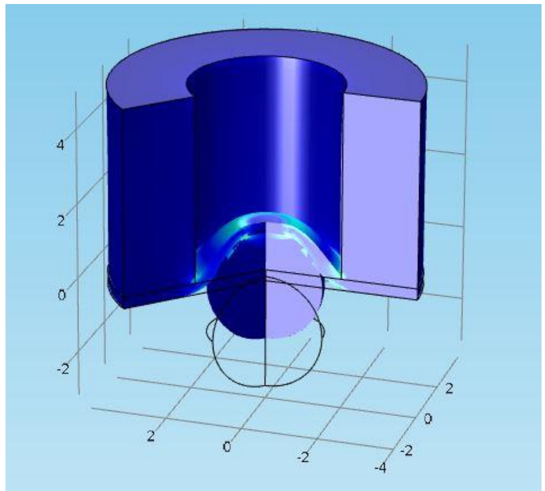


Figure 3.5 3mm bead simulation

3.2 Fabrication of the Membrane and PDMS cube

Fabrication of the thick layer membrane is much easier than the thinner membrane. Fabrication of the thicker membrane is simpler compared to the thin layer membrane. For fabricating the thicker membrane, I pour a small amount of the Ecoflex into a Petri dish. And use a small stick to stir it gently to spread Ecoflex over the bottom of the Petri dish evenly. After waiting overnight, Ecoflex will be curd. The thickness of the membrane made by this way is about 0.1mm. Which is good for picking up the large bead (3mm diameter). Method for fabricating PDMS cubs is the same as fabricating thicker membrane. The difference is to pour more amount of PDMS into the Petri dish to make the thickness of the PDMS layer is about 5mm. After it is curd. I use punching to punch the PDMS to get cylinder cubes with different diameters from 1mm to 8mm. But using punching to cut PDMS cube only works for making the larger size of the PDMS cube because the smallest diameter of the punching is 0.5mm. And it’s hard to cut PDMS cubes in good shape by using punching. So for making soft needles with small diameters, we need to use other method like laser cutting.

The thickness of the thinnest membrane can be made from by pouring Ecoflex is about 0.1mm. But this thickness is still too thick for picking up small beads (0.3mm). So we need to use another method to fabricate the membrane. First, we consider using spin coating to make thin layer membrane with the thickness about 30 **µ**m. On website I only found the relationship between the spinning speed and the thickness for PDMS (Figure 3.6). Since

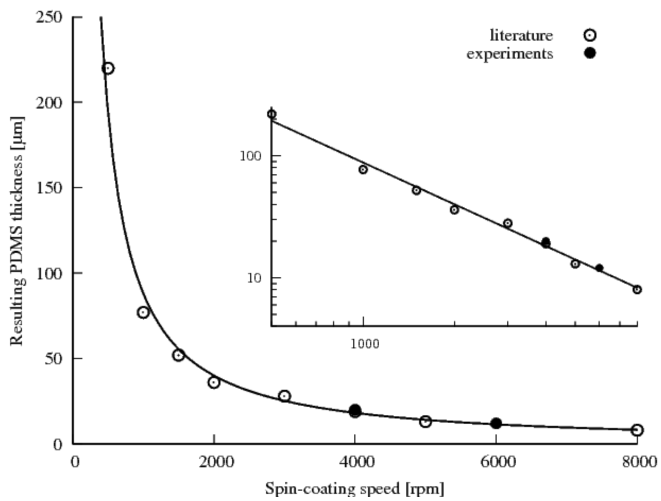


Figure 3.6 Spinning speed vs thickness (http://willem.engen.nl/uni/intern-mbx/material/Sylgard-184-spincoat.php)

Ecoflex and PDMS have similar properties. I assume Ecoflex has the similar relationship for spinning speed and thickness. From Figure 3.6, we can see that when the thickness is about 30µm, spinning speed need to be about 2000 rpm. However, after we fabricated the membrane on the silicon wafer, it's hard to detach the membrane from the silicon wafer base without damaging it. Since Ecoflex has high stickiness, and peeling a thin layer (30µm or less) of any substrate in air could result in stretching, tears, and membrane collapsing on itself. I tried to put the membrane into the ultrasonic water tank to wash with DI water, then peel with tweezer. And this method still damaged the membrane. So we need to search for another way to fabricate the membrane. Prof Mazzeo suggested a way to drop one or two drops of Ecoflex onto the surface of water. The Ecoflex will spread quickly on the surface of the water like oil. I did some experiments about making membrane by using this method. One thing that we need to care about is that during dropping Ecoflex, we need to make the drop to be filamentous, and drop it very gently onto the surface of the water. If we drop it quickly, Ecoflex won't spread out. And there's another factor we need to consider about. By using plastic Petri dish or glass Petri dish may cause difference in making the membrane, because it's different when water contact with these two different materials. As figure 3.7 shows, there will be a concave meniscus when water close to the surface of the glass Petri dish. So, when we drop Ecoflex to the glass Petri dish to make the membrane, Ecoflex may not spread to contact with the edge of the glass Petri dish. Here are two typical results from those experiments for using plastic Petri dish and glass Petri dish. For the first one I used the plastic Petri dish to fill the water. And the edge of the membrane attached to the wall of the Petri dish (Figure 3.8a). I scooped it out of the water by using silicon wafer. And the membrane can keep in flat shape (Figure 3.8b). For the second experiment I used the glass Petri dish to fill the water. And the edge of the membrane didn't touch the wall of the Petri dish (Figure 3.8c), which is easier to be scooped out of the water.

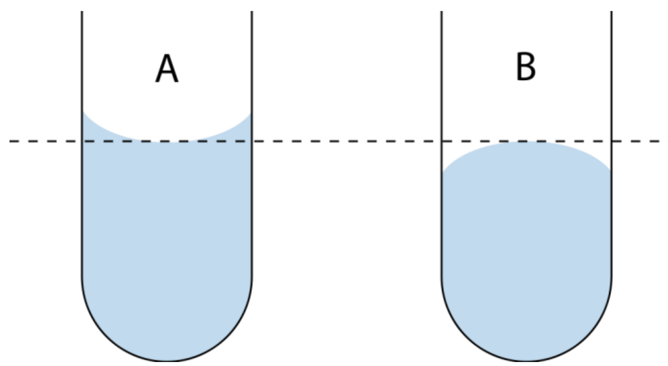


Figure 3.7 Concave meniscus

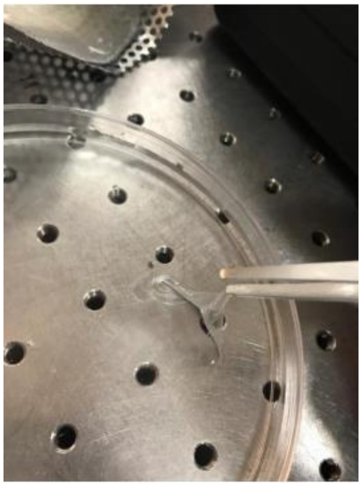


Figure 3.8a *Figure 3.8b* *Figure 3.8c*

3.3 Membrane and PDMS Cube Integration

After fabricating the membrane and the PDMS cube, next step is to attach the membrane to the cube. The main idea is to contact PDMS cube to Ecoflex membrane before the membrane become dry. After Ecoflex become dry, the membrane will be attached to the PDMS cube. And methods for attachment are different for thicker membranes and thinner membranes.

As I mentioned above, the thicker membrane made by pouring Ecoflex into the Petri dish. So, after Ecoflex spreads evenly on the bottom of the Petri dish, I place the PDMS cube onto the Ecoflex substrate. After waiting overnight, the membrane will be attached to the PDMS cube. And the thicker membrane soft needle is shown below (Figure 3.9).

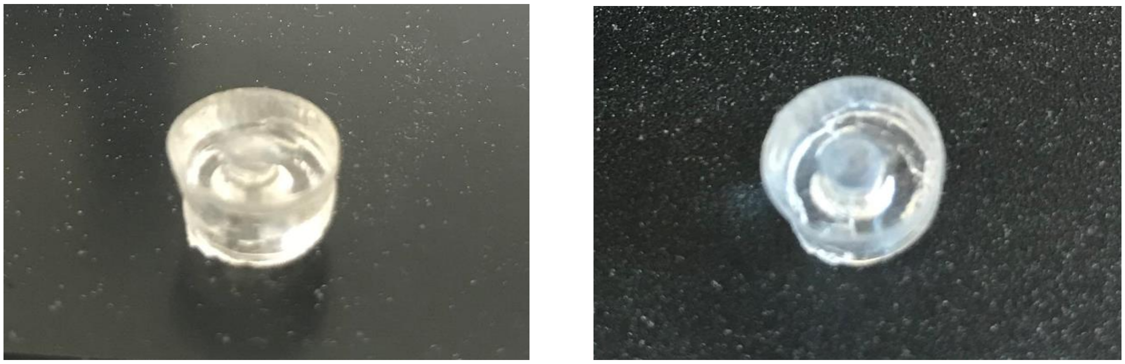


Figure 3.9 Thicker layer membrane needle

The method for attaching the thinner membrane is complex compare to the thicker membrane, and I tried three different ways to attach the membrane.

For the first way, I moved the membrane out of the water first, then attached the membrane to PDMS cube by using the Sil-poxy glue. To prevent sealing the hole of the PDMS cube, I used small stick daubed small amount of Sil-poxy around the hole, then pressed the PDMS cube onto the membrane lightly until they stick together. The result shown below (Figure 3.10). The result seems good. However, daubing the glue on PDMS cube or

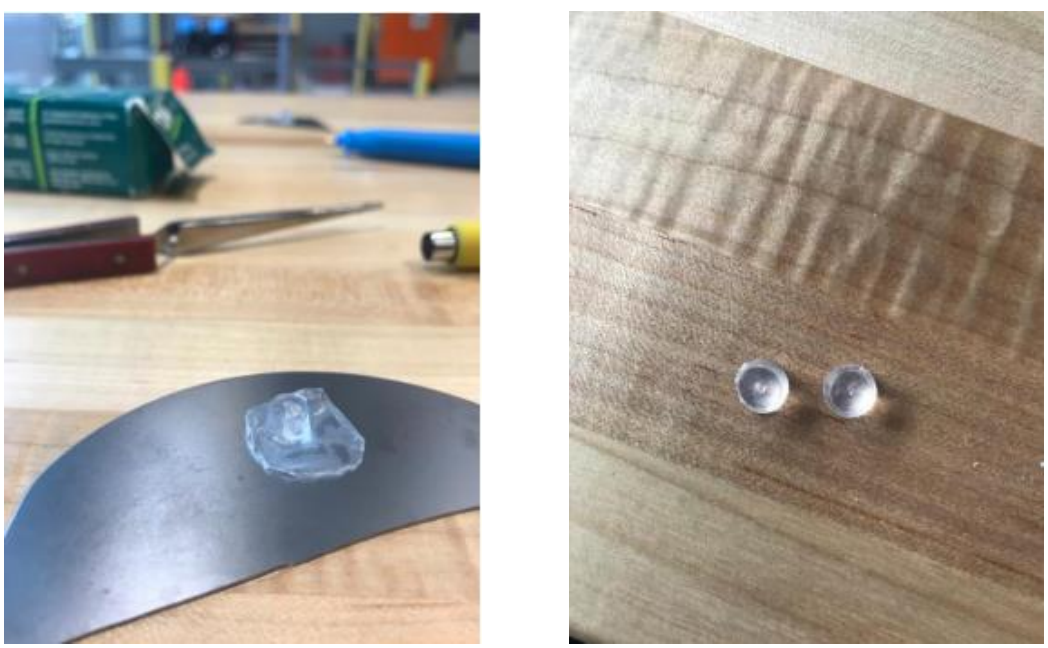


Figure 3.10 Attach membrane by using Sil-poxy

pressing PDMS cube onto the membrane will have the risk of sealing the hole by Sil-poxy. For the second way, I didn't move the membrane out of the water first. After daubing some Sil-poxy around the hole on PDMS cube, I directly contact the bottom of the PDMS cube onto the membrane when it's still resting on top of the water. And then press lightly until the PDMS cube attached with the membrane.

Figure 3.11 Second method

And the result shows below (Figure 3.11). It looks similar to the first way. But the membrane sometimes will crinkle during the process when I stick the cube onto it.

For the third way, I directly attached the membrane to the PDMS cube without using the glue. I punched 0.5mm holes on PDMS piece first, and passed three small wire through the PDMS cube (Figure 3.12a). Then put the PDMS piece into the Petri dish. And then I poured water into the Petri dish until the surface of the water just over the top of the PDMS piece (Figure 3.12b). After this was prepared, I make the membrane on top of the water by drop Ecoflex. The key idea of this method is that when using the pouring method, the top surface of the membrane become dry after waiting overnight, but the bottom surface of the membrane stays wet because it contacts with water. After the membrane was made, I pulled the membrane out of the water, and the membrane clung to the top of the PDMS piece (Figure 3.12c). And after the membrane became dry, It attached to the PDMS piece well. Then I could cut the PDMS piece into cylinder shape with 0.5mm holes in the middle. From figure 3.12c you can see there are some small bubbles between the membrane and the PDMS piece. To avoid having bubbles, I need to be careful when pulling PDMS out of water. Although there are some bubbles, there are still some part of the membrane above the hole is flat that can be use.

Figure 3.12a

Figure 3.12b

Figure 3.12c

For the three ways above, the third way should be the best way for the attachment. This method doesn’t require the use of glue, which eliminate the risk of sealing the hole by the glue. Also the attachment can be done when the membrane is still on water, which can eliminate the damage caused by getting the membrane out of the water.

4. Control the 3-Degree-of-Freedom Precision Translation Stage

In this chapter, I'll talk about the controlling of three stepper motors for the moving of three axes. Two methods will be talked about in this chapter. One is based on image-based control. We need to determine the order of the movement of each axis for this method. And the input may change based on time. Another method is based on MIIC technique. We control the moving of three motors by directly input the trajectory for each axis, which is simpler compared to the first method.

4.1 Overall Control Scheme

First, I'll talk about the whole process for controlling the system by using real-time control and vision control. The device that we designed is to put a few messy targets in the desired position. For moving one target, the process can be divided into three parts (Figure 4.1): 1. Move from the original position to the target's position, and catch up the target bead.

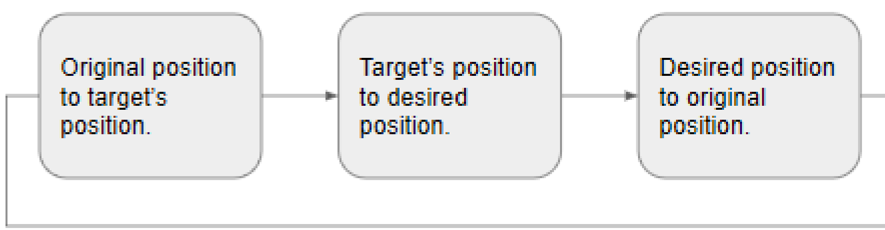


Figure 4.1 Process for a whole run

2. Move from the target's position to the desired position, release the target. 3. Move from desired position to the original position. We can see from the process that the position of the picking part is only related to the x, y axes. The z axis does the same movement every time, so we only care about the movement for the x, y axes, and add z axis to the system after we create the control system of x and y axes. We set the coordinates of the picking

part for the original point to be (x_0, y_0), and the target's position to be (x_1, y_1) and the desired position to be (x_2, y_2). The reason why we not directly cut off the third step, and move the target from the desired position to the next target's position is that it's convenient for writing the code. Because we can run the same process for every round. The third step acts like a transition to the next round. Since it moves back to the original position every round. We can calculate the relative distance by minus the same point ((x_0-x_2, y_0-y_2) from desired position to original position, and (x_1-x_0, y_1-y_0) from original point to the target's position). There's another important reason that we add the important reason that we add the third step is that we can reduce the error in the third step for every round. Every time when the picking part is back to the original, it's probably not exactly on the actual original point because there's some error (Δx, Δy) during the movement in a round (Figure 4.2) And we can reduce the error by either placing the picking part back to the actual original position by using camera positioning or by setting the new original position to be the original point. I think placing the picking part back to the actual original position could be a better way to reduce the error. Because we only need to add one step at the end of every round. If we set the new original position to be the original point, we need to change the coordinates of all the rest positions at every beginning of each round by camera positioning. For our device, we didn't consider this error because by now it's difficult to use the digital camera to correct the original point. In real situation, the camera is set below the Petri dish, and the soft needle is set over the Petri dish. There's a Petri dish and fluid in the Petri dish between the camera and soft needle. The fluid and Petri dish will infect the

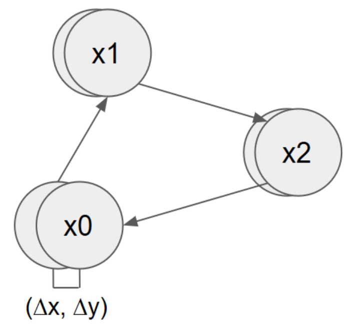


Figure 4.2 Moving error

position of the soft needle shown in camera. So now we don't apply real-time vision control during the movement. We only apply real-time vision control for getting the original position for each target in real-time.

4.2 Target Localization Mechanism

The work of computer vision control is done by Fan Wu. I'll briefly talk about the steps for locating targets and the coordinates reading in Simulink Real-Time.

As the figure 4.11 shows, first we compress the file and transfer to target PC. Then we decompress the file to the RGB image. From the RGB image file, we can know its red, green and blue intensity. After we exchange the RGB image to intensity image, we compute the complement. In the output image, dark areas become lighter and the light areas become darker. We add it because the color of the target that we need to locate is black, but the Bolb analysis can only locate the white target. So, if the color of the target is darker than the background, we need to do the reversion of the color by using the image complement. Then we set a threshold to the image to exchange the RGB image to binary image. In the binary image, we calculate the boundary for each target, then find their centers by using the Blob analysis. However, there may be some small dots that have close color with targets that will influence the results, from figure 4.3a we can see that there's a small white dot at the bottom of the right side. So we add erosion before the Bolb analysis, which cut the edge of each Bolb to delete the small white dot as figure 4.3b shows.

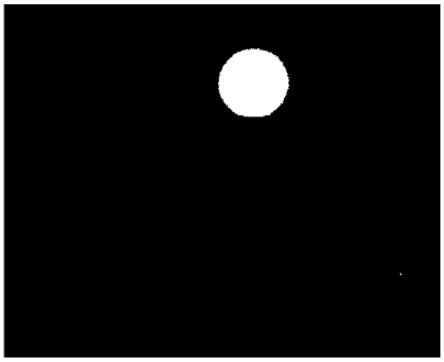


Figure 4.3a Before erosion

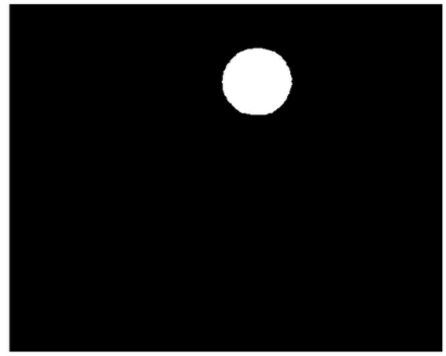


Figure 4.3b After erosion

There are some limitations for the vision control. First, now we are using the host PC to take pictures of targets, then do the image processing on target PC. Part of the process are not done on the target PC, which means this method are not exactly Real-Time. The speed for analysing a picture is limited by the transfer speed of the cable. And also the speed of the camera's USB cable also limits the speed for analysing a picture. Right now we are using the USB 2.0, the transfer rate is about 300MB/s, the transfer speed could be 10 times faster if we use a USB 3.0 camera.

Another limitation is that labels of the targets are ranged from right to left. Which means the order of moving a target and its desired position cannot be random. The

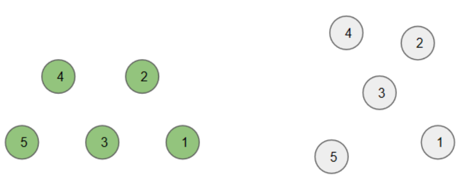


Figure 4.4a Right order

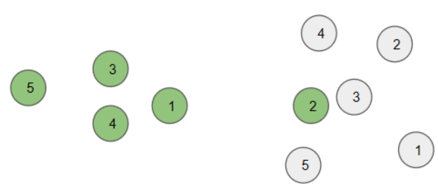
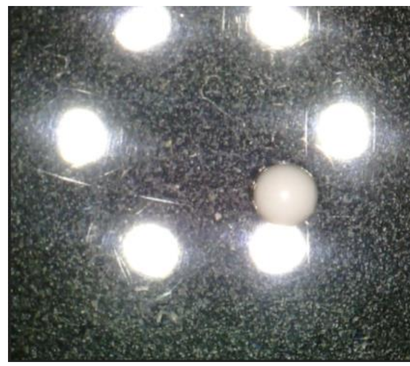


Figure 4.4b Wrong order

coordinates we get from the image analysis is a 2×n matrix, n is the number of targets. From the first row to the last row is the coordinate for the first right target to the last right target. So we always move the first right target in the scope to the first left position as the desired position (Figure 4.4a). If we do not move the target to the left of the first left target position, we may read one target more than one time. For example, from figure 4.4b, we didn't move the second target to the right of the fifth target. So after it finish moving the fourth target, the system will move the second target again instead of moving the fifth target. The requirements for color contrast are a bit strict for locating right coordinates of targets. We use the binary image to determine boundaries and centers of targets. But when colors of targets are close to the background color, it's hard for setting a threshold to separate targets and the background. Also lights will also influence the reading of the coordinates. From figure 4.5, we can see that there is only one target, however from the reading of workspace, there are more than one target because of the light reflection. So we cannot use light that vertically illuminated the Petri dish. But when we use the fluorescent lamp from the lamp, we still need to adjust the position of the camera to successfully read positions of targets.



2.5714	152.2857
99.4086	178.1327
12	218
13	183.5000
13	213
39.2414	223.8276
39	9
43	113
43	124
58.5556	254.8889
64.4286	94.5714
73	113
78	91
80	319
84	305
86.6000	98.7000
95.5946	294.8108

Figure 4.5 Light interference

4.3 Control of Stepper Motor

For controlling stepper motors, first we need to know how stepper motors work. Stepper motors rotate continuously when DC voltage is applied to their terminals. We control the moving of stepper motors by control the motor driver. There are two inputs for controlling a stepper motor. One is for direction, and another one is for voltage signal. A stepper motor rotates one step when there’s a voltage change from high (higher than 4V) to low (lower than 0.5V). For the direction, the stepper motor rotates clockwise when the input signal is at low voltage (lower than 0.5V), and it rotates counterclockwise when the input signal is at high voltage (higher than 4V). For this system, it’s convenient to use PWM signals or sine wave signals to control the movement of the stepper motor, because the stepper motor rotates one step for every one period of a input signal. Thus we can also control the speed of a stepper motor by change the frequency of the input signal. Changing the frequency is one way to control the rotation speed of a stepper motor. It controls how many steps a stepper motor will rotate in one second. The rotation speed of a stepper motor is also affected by the motor driver in another way. The driver have different divisions for running a motor. Division means the number of steps a stepper motor needs to rotate one round. For example, if we set the frequency of the input signal to be 100, and the division to be 500. It will take 5 seconds for the stepper motor rotating one round (Frequency of stepper motor to be 0.2s). The driver we use is fmdd50d40nom stepper motor driver, and it has divisions as Figure 4.6

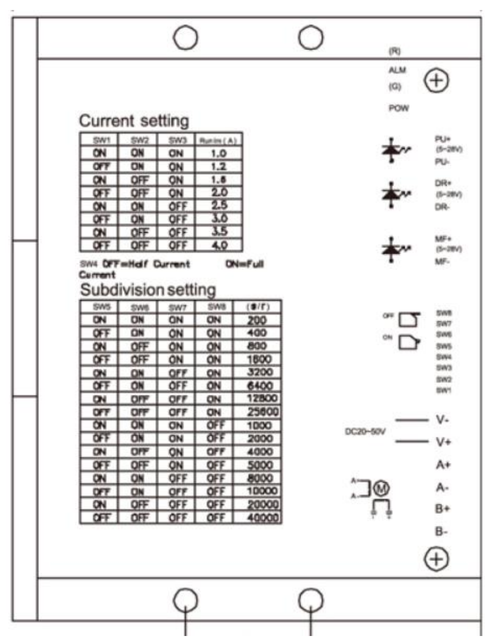


Figure 4.6 Motor driver

shows. We can see that this motor driver has 8 switches. The sw4 to sw8 control the division of the driver. By turning the 4 switches on and off, we can get 16 different divisions from 200 to 40000. The sw1 to sw3 control the working current, for our stepper motors, 1A working current is good enough. Sw4 controls the number of pulses. To make the input simpler, we turned sw4 off to the single-pulse control mode. Division of the driver also influence the resolution for each axis. The resolution for each axis is the distance for moving one step. So, the resolution is determined by the division and the thread pitch (Figure 4.7), and It equals to $\frac{Thread\ pitch}{Division}$. For our device, the thread pitch is 5mm, and the division that we choose is 1000, so the ideal resolution for each axis is 5**μ**m, which is good enough for locating small beads. The frequency for input signal for all three axes is 500/s. So the moving speed for three stages are $\frac{500}{1000} \times 5 = 2.5$mm/s.

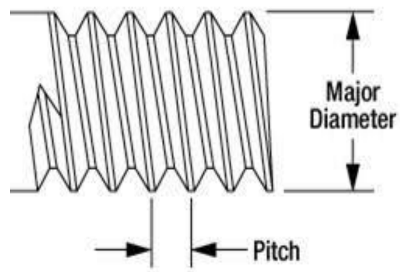


Figure 4.7 Thread pitch

After determining the moving speed and resolution for each stages, we move on to control the movement for each stepper motor. Since we use the Simulink Real-Time platform to design our system. There are few things that we need to care about. First, the real-time system has its own sampling time, which means that the whole system will run one time after every fixed time. It can be set on the Configuration Parameters >Solver (Figure 4.8)

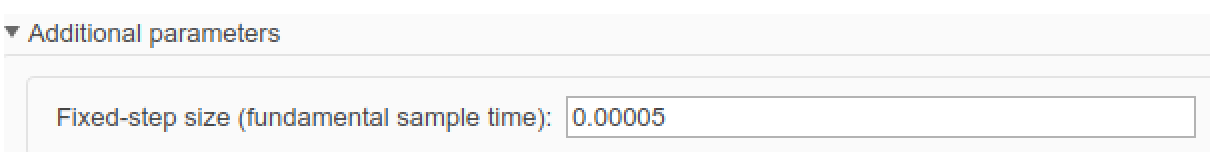


Figure 4.8 Sample time

So, the frequency of the sampling rate should be higher than twice the frequency of the output signal sent to stepper motors, or there will be steps missed.

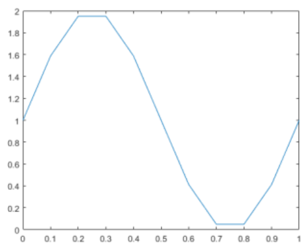


Figure 4.9a

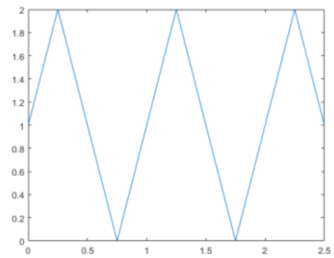


Figure 4.9b

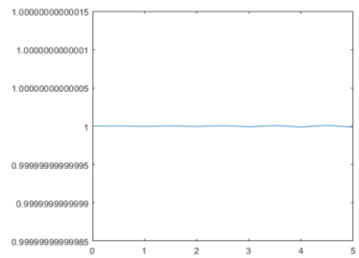


Figure 4.9c

From figure 4.9, we can see that when the sampling rate is 10 times the frequency of the output signal, the signal can have a complete waveform (Figure 4.9a). When The sampling rate is 2 times the frequency of the output signal (Figure 4.9b), the signal still can generate the high and low voltage. When the sampling rate is less than 2 times the frequency of the output signal (Figure 4.9c). The signal cannot generate the high and low voltage, thus cannot drive stepper motors.

The next thing that we need to care about is that we need to build our system basing on time. The main variable for the system should be time. So we need to change every variable to time unit. So, for our system, we need to change the distance change to time change. Since we only care about beginning positions and final positions, we can use time length to represent the displacement length for each moving part. For example, for the first moving part (Form original point to the target's position), we use $x_{01} = \frac{x_1 - x_0}{2.5}$ to represent the distance change for the first part on x-stage. And the input sine wave for each axis is $signal = 2.5 \times sin(2 \times pi \times 500 \times t) + 2.6.$

4.3 Motor Control System Integration

After we coding the movements for all three parts for three stages, we combined them together with the right order. During coding in the real-time platform, there are few points that we need to care about.

First, since the real-time system has its own fundamental sample time, it has the “for loop” for the whole system itself. We don’t need to write other “for loop” in our block.

Second, during the coding, we find there’s an interest thing that different from the normal coding. When we add the time interval with upper and lower limit to divide one movement to several parts (eg. t_1< t< t_2, $t_2 <= t < t_3$…), it will stay on the first time interval and not move on to the next time interval. But, when we change the time interval only with the upper limit (eg.t< t_1, t< t_2, $t < t_3$…), it can move on to the next time interval. The reason why this happen I think could be that the lower limit is more first than the upper limit in Simulink Real-time. So, when there are lower limit and upper limit at the same time, it will only read the lower limit.

The third point that we need to care about is that to make all variable persistent in every round, we need to write “Persistent variable_name” at the beginning of every block. And for our system, we run each part as an individual part, so we need to set all things to be zero at the beginning of each part, or the data will start accumulating from the previous one instead of zero.

As I mentioned before, for our system, we only do the vision based real-time control for the original points. So the original input coordinates will change with time. To achieve this function, we change the input coordinates every time the coordinates of the original point change. But this would lead to recalculate the relative distance between the original point and the target's point. So every time the coordinates change, there will be a new sine wave signal output. If the output signal changing too fast, it won't output a full period of a sine wave. For example, the sampling rate for our system is 20000, and the frequency for the output signal is 500. We need 40 sample times to output a full period of a sine wave, so the coordinates of the original points cannot change within 40 sample times (0.002s). Thus we need to set an interval for this change to prevent the signal from changing too fast. We noticed that for our system, the last 4 digits of the coordinates changes really fast with time, so we get the approximate number by adding an interval to the fourth last digit.

To run each part alone, we used switches to open and close each parts, which is convenient and simple to control. And we use signal for label changing at the end of each round. The flow chart is shown below (Figure 4.10). The important thing for designing

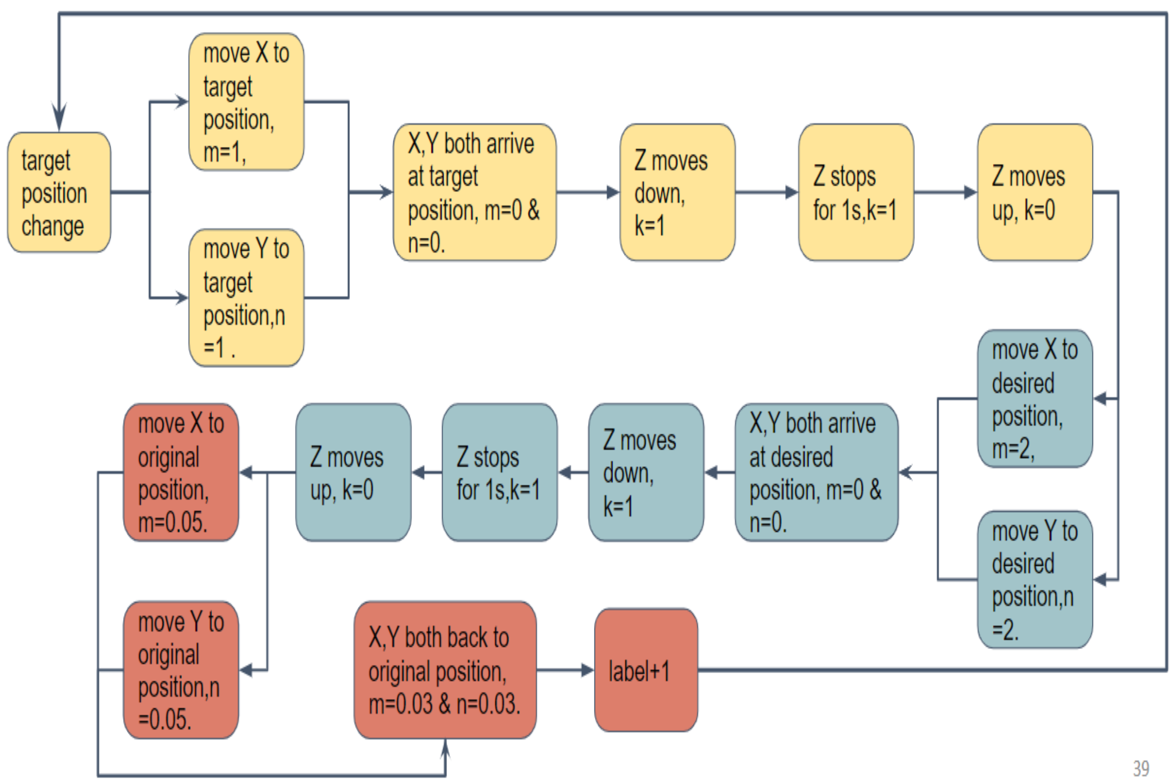


Figure 4.10 Flow chart for the whole system

this system is that for running each part, we need to make sure the previous one is completely over, which means that the x-stage and y-stage both arrived to the destination. So we need command from both x and y to indicate the next move. At the end of each round, we use the analog output to output two signals to indicate that the round is over and the label will increase. And it will proceed to the next target.

Figure 4.11 is the part for the image processing. We have a problem for reading image from the target PC, and we tried many ways to solve the problem and still cannot figure it out. So we change from reading image from the target PC to reading image from the host PC, and then send images continuously to the target PC to make it to be real-time based input. And run the image processing on the target PC.

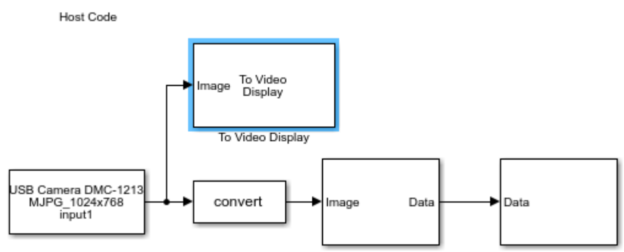


Figure 4.11a Read image from host PC

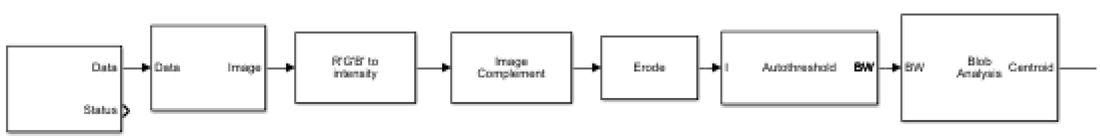


Figure 4.11b Image processing on target PC

Figure 4.12 is the block for the sampling rate transition. The resolution for a image is 640×480, and the size for a 640×480 image is 300KB. The transfer speed for the camera cable is about 300MB/s. So, the highest transfer rate for image is about 1000/s. After adding the time for processing one image, the whole process for reading and analysing one image is longer than 0.001s. So we set the sample time for catching one image from host PC to be 0.05s. However, the system cannot run when the sampling time are different for each part, so we add a block to converts rate of input signal to specified sample time. Figure 4.13 shows the principle for sample rate transitions.

Figure 4.12 Rate transition

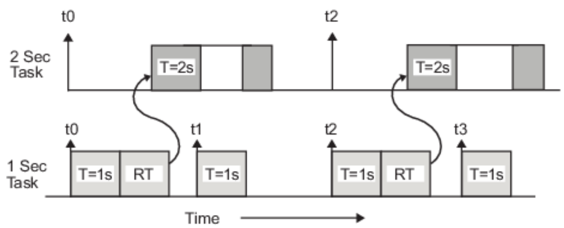


Figure 4.13a Faster to slower

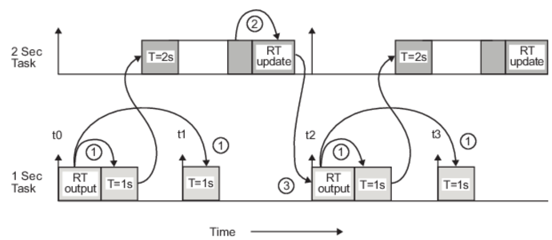


Figure 4.13b Slower to faster

Figure 4.14a is the Simulink model for the movement control of Y-axis. Since the X-axis and Y-axis both are static during the picking process, we only add the movement of Z-axis and the valve control to the Y-axis part. The movement control for X-axis (Figure 4.14b) is similar to the Y-axis.

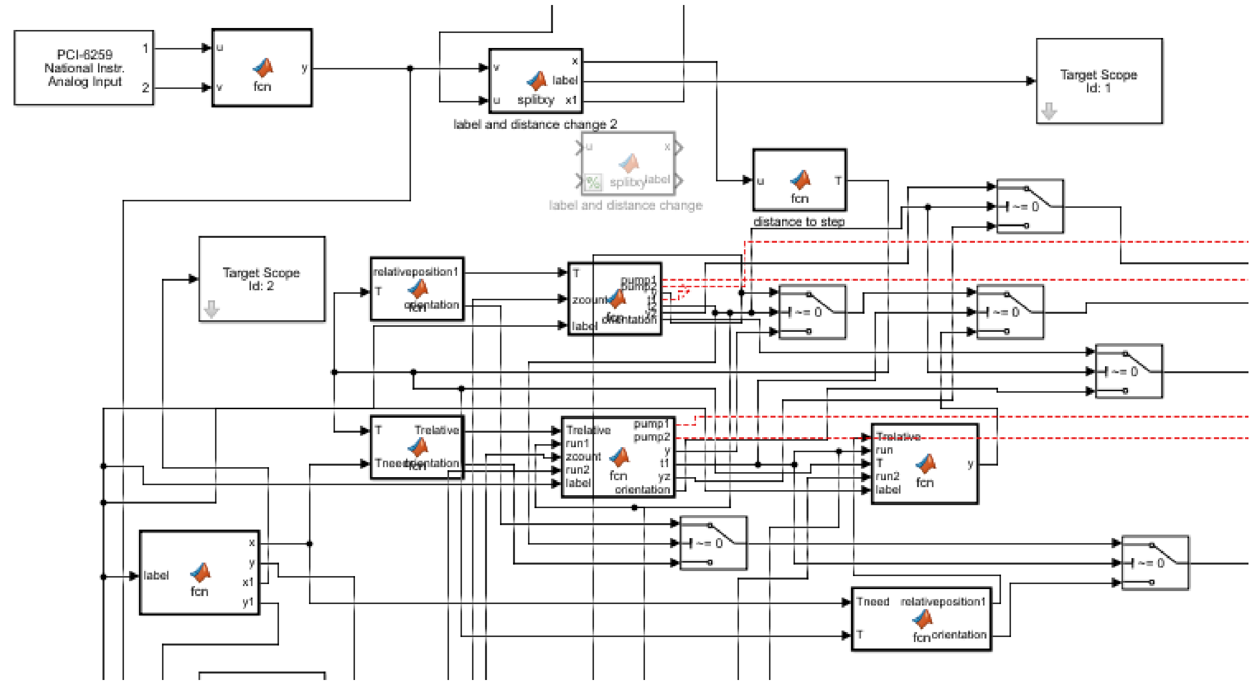


Figure 4.14a The movement control of the Y-axis

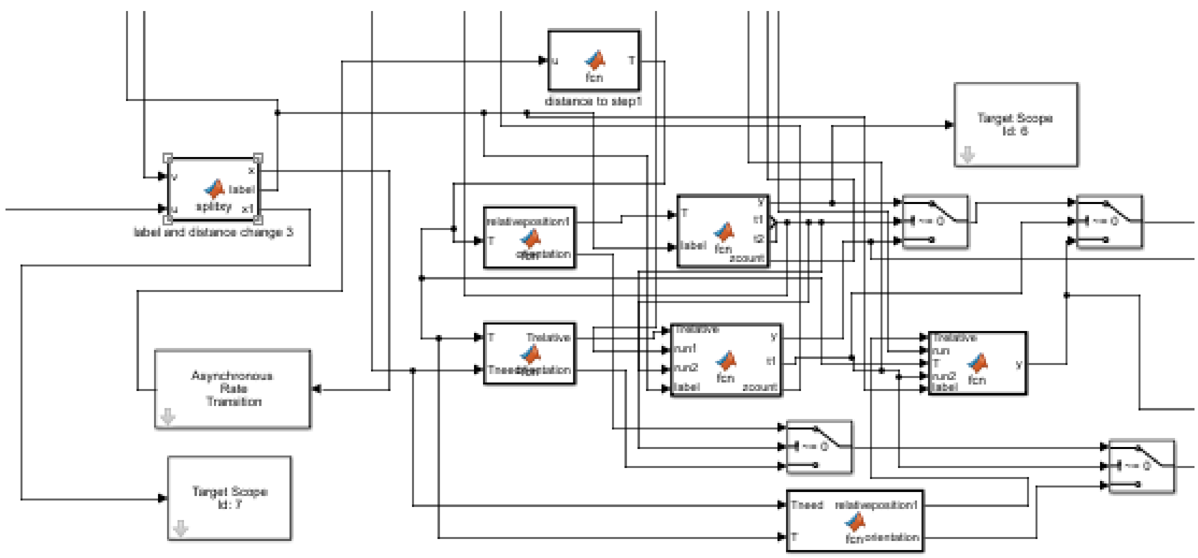


Figure 4.14b The movement control of the Y-axis

4.4 Limitation of stepper motor

Stepper motor may miss some steps when the frequency of the input signal is too high, it because the signal applied to the stepper motor gives more torque than the required by the application for the stepper motor. I analysed the output steps by using encoder. And I get the step missing for different frequency of the signal (Table 4.1). As we calculated before, the resolution for the linear stage is 5µm, and the ideal resolution for our system is less than 20µm, so the idea frequency for the input signal is equal or less than 1000. But for the future work about using MIIC technique, we need the motor doesn't loss any step, so we set the frequency to be 500.

frequency(steps/s)	step missing for every 500 steps
1000	3-4
666	2-3
500	0
200	0

Table 4.1 Step missing

5. Sucking System Integration and MIIC Technique Application

In this chapter, I'll talk about control of vacuum pump and solenoid valve to apply negative pressure and zero pressure. The use of the MIIC technique and its working principle will be briefly presented. And I'll also present some experiments about applying the MIIC technique to our system and problems that we are facing right now

5.1 Sucking System Integration

For our design, the pressure we need to pick up 3mm beads is about 1.2PSI, So I need to choose vacuum pump with low pressure range. Parker Hannifin - L045B-11 vacuum pump with working pressure range 0-10 PSI is a good choice for us (Figure 5.1). And we use the solenoid valve combined with digital relay to give the negative pressure and the zero pressure. As I mentioned in the 1st chapter, No.1 pipe connected to the syringe. No.2 pipe goes to atmosphere. No.3 pipe connected to the vacuum pump (Figure 5.2).

Figure 5.1 pump

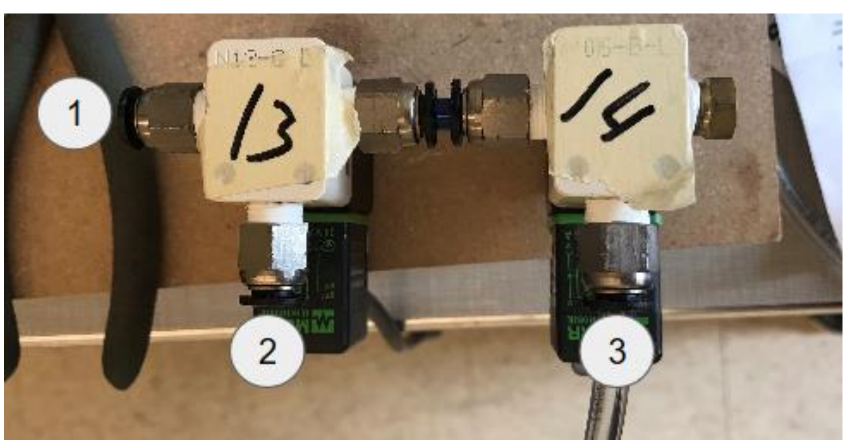


Figure 5.2 Solenoid valve

5.2 Vacuum-based Sucking Control

Negative pressure and zero pressure can be provided by open and close valves. For providing negative pressure, we open No.3 pipe and close No2. pipe, then the syringe will connect with the pump. For providing zero pressure. We open No.2 pipe and close No.3 pipe, then the syringe will connect to the atmosphere, and the pump will be closed (Figure 5.2).

5.3 Working Principle of MIIC Technique

In this section, I'll talk about the use of the MIIC technique and its working principle. And I'll also present some experiments about applying the MIIC technique to our system and problems that we are facing right now.

MIIC technique is an approach to improve the accuracy of the high-speed system. It's an inversion-based iterative control technique. The working principle for MIIC technique that applying on the multi-axis motion control is simple. The algorithm of MIIC is shown in figure 5.3. α is the estimated DC-gain that get from the slow speed condition. y_d(jω) is the desired output trajectory, and u_k(jω) and y_k(jω) are the input and output from the kth iteration. The diagram for MIIC process is shown as Figure 5.4. As you can see from the diagram that there's a noise from the kth iteration, but we don't consider the noise for our system because of the low speed for our prototype device. The main part for the MIIC for our system is the input trajectory and the

$$u_0(\mathrm{j}\omega) = \alpha y_{\mathrm{d}}(\mathrm{j}\omega)$$

$$u_{k+1}(\mathrm{j}\omega) = \frac{u_k(\mathrm{j}\omega)}{y_k(\mathrm{j}\omega)} y_{\mathrm{d}}(\mathrm{j}\omega),$$

$$(\text{for } u_k(\mathrm{j}\omega) \neq 0,\ \ y_k(\mathrm{j}\omega) \neq 0,\ \ k \geqslant 1)$$

Figure 5.3 Algorithm of MIIC

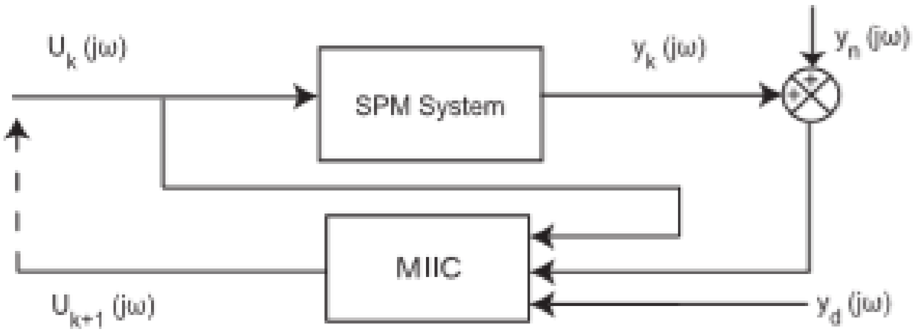


Figure 5.4 Diagram for MIIC

output trajectory. For our system, we need to transfer input trajectories to input signals to three stepper motors. And after we run the program one time, we need to read three trajectories from three stepper motors. Reading trajectories can be done by using encoders. But there are some problems with readings from encoders that we use, and I'll talk about it later.

5.4 Transfer Trajectories to Output Signals

The way to use MIIC to control stepper motors is different from the previous method. First we need to divide the trajectory for the movement of the whole system to x,y,z axis individually. For example, if we have a trajectory for the whole system as figure 5.5a shows. The movement for the whole system would be like: From point (0,0) to (-1,1), pick up the target at (-1,1), then move to (0,2), and release the target at (0,2), then move to (1,1), then back to original point (0,0). And we can get the trajectory for x,y and z axis as Figure 5.5b, 5.5c and 5.5d show. For designing the desired trajectory for our system, we need to care about the maximum moving speeds for all three axes don't exceed 2.5mm/s for not losing any step.

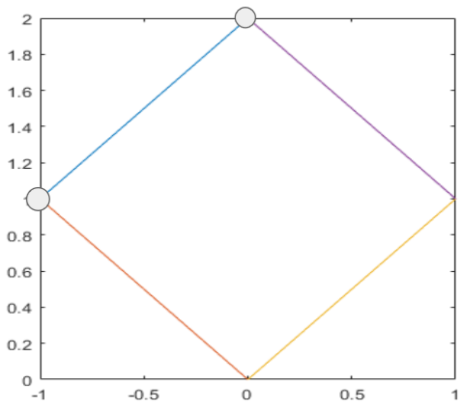


Figure 5.5a

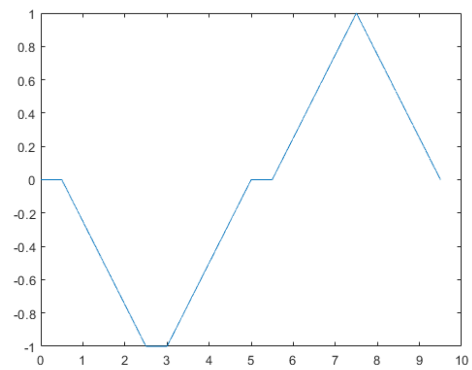


Figure 5.5b

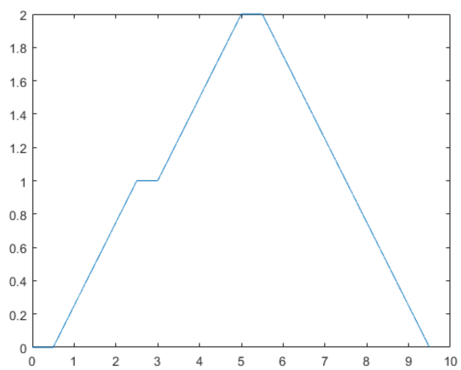


Figure 5.5c

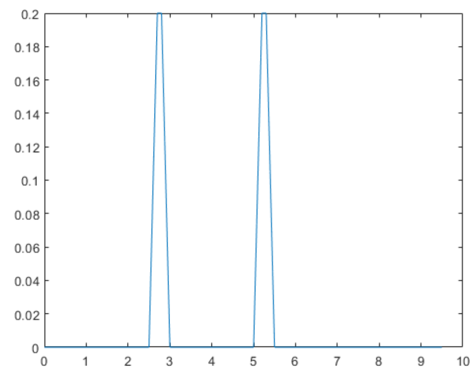


Figure 5.5d

The way I use to transfer trajectories to output signals is to output one step for every one step displacement. First I considered to compare the coordinates every constant time. But for this method, there will be time delay between the time that we need to output signals to motors and the time actually send signals to motors. So the best way I think is to send one step for every one step displacement, which won't have any delay for output signals. Calculating displacement can be done by compare the coordinates between two points. But we cannot directly compare coordinates of two points every sample time, because the displacement change will be really small between two sample times, which is not big enough to determine if there's one step displacement. So what we need to do is to increase the displacement that we read every time to have at least one step displacement. But we cannot compare the coordinate to the previous coordinate where we output a step signal, because that will cause the error accumulated. From figure 5.6, the first row is the displacement that we read, and the second the row is the actual displacement. We can see that there's an error (orange) because the first displacement that we read that is a little bit longer than the actual displacement. To avoid this error, I compare the number of steps instead of the displacement to the previous point. First I transfer the distance to the original point to be the number of steps, then compare the number of steps to the number of steps for every 40 sample times (0.02s), because the minimum time to output a step is 0.02s. If number is different, we can output a step signal to move the stepper motor one step forward or backward depend on the number increase or decrease.

Figure 5.6 Displacement error

When I was doing experiments about output signals, I noticed that the output signal wasn't complete (Figure 5.7). I wrote a program to output a trajectory in x-direction, which move backward for 1s, and stop for 1s, then move forward for 1s. The speed for the stepper motor is 50steps/s. After I run the program, the result shows as Figure 5.7a. The steps for the first part movement are only about 45 steps. But the second part movement doesn't lose any steps. So it's probably because of the delay of the Simulink Real-Time output. When I add 100 sample times (0.005s) as the making up. The results show better as Figure 5.7b, which lose about 2 steps. When I add 200 sample times (0.01s) as the making up (Figure 5.7c), there's no step missing.

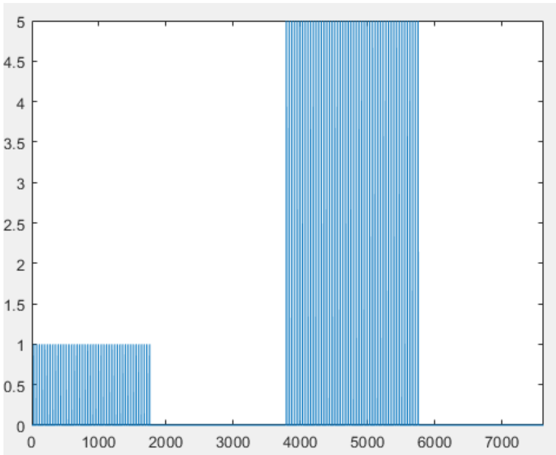


Figure 5.7a

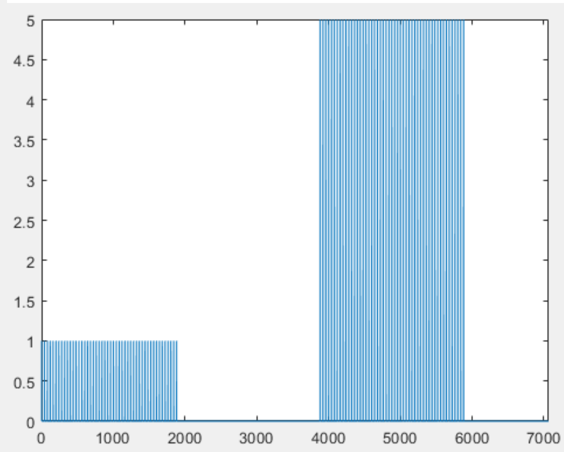


Figure 5.7b 100 delay

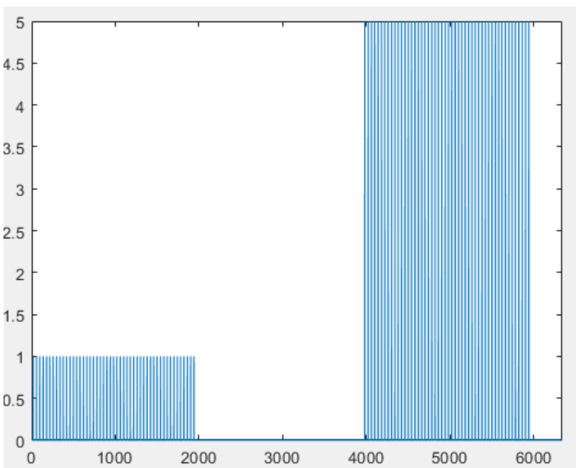


Figure 5.7c 200 delay

5.5 Calculate Trajectories from Encoder Readings

An encoder is an electromechanical device that can measure motion or position. Most encoders use optical sensors to provide electrical signals in the form of pulse trains, which can be translated into motion, direction, or position. The encoder we use is HS30A Optical Kit

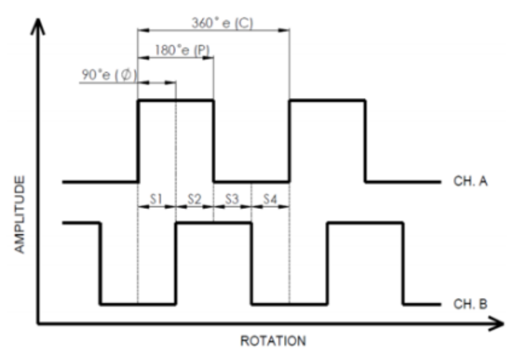


Figure 5.8 Output waveform

Encoder, which has 4 wires, and we only need two of them to get the number of steps and moving direction, Channel A and Channel B. Figure 5.8 shows a typical output waveform for Channel A and Channel B. The number of steps can be read from the total amount of the edges. And the phase (Number of electrical degree between Channel A and Channel B as a result of the transition in the output state) can determine the rotation direction of the stepper motor. For example, if A leads B (Phase is positive), the motor is rotating in clockwise direction. If B leads A (Phase is negative), the motor is rotating in counter-clockwise.

To read the number of steps that each motor rotates is simple. I can add one step every time there's an edge appear. But I faced some problems when I was trying to add direction to draw a trajectory. The outputs of Channel A and Channel B for our encoders are irregular. As you can see from figure 5.9, the phase of Channel A and Channel B are changing with time. So it's different to determine the rotation direction by using phase. So I change to count the number of steps to measure the rotation direction, which can

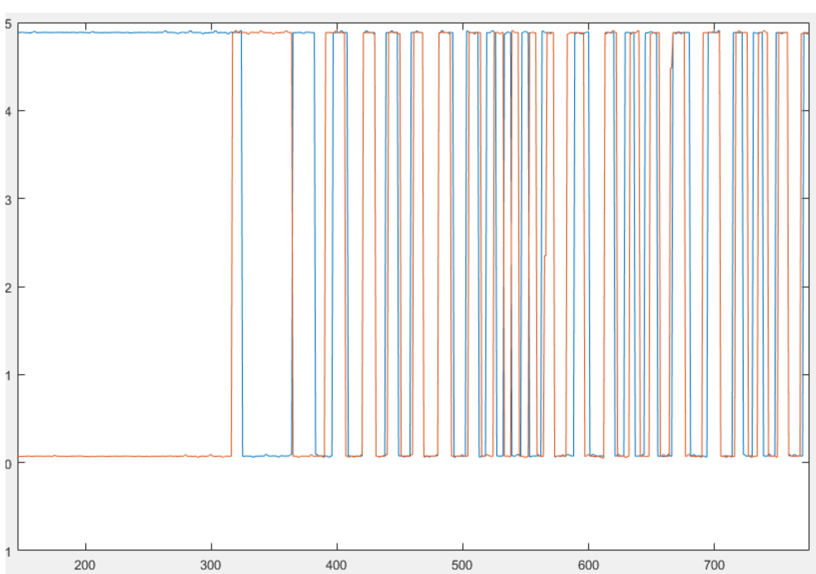


Figure 5.9 Encoder readings

also output the trajectory at the same time. For example, if the number of steps of Channel A is one more than Channel B, the system will add 0.005mm(one step) to the total displacement. If the number of steps of Channel B is one more than Channel A, it will minus 0.005mm to the total displacement. However, the start voltage of Channel A and Channel B will influence the result of the output trajectory. Because of the random start voltage of Channel A and Channel B, it's impossible to output the right trajectory. Also I noticed that when one movement is over, Channel A and Channel B sometimes will not back to zero voltage immediately. This also will influence the result of the output trajectory. All three encoders present the same problem, so I think it might be not the problem of the encoder. It might be the problem of motors. So, in future research, I plan to replace motors to see if there're still such problems, or we need to replace encoders.

6. Discussion and Conclusion

6.1 Discussion

In this paper we designed a device to manipulate small beads. About designing the whole structure, we choose to use Cartesian 3-axes linear stages as the main body of the device. The advantage of using this structure is that we will have an easier coordinates system to control the movement of each axis. But the linear stage we bought doesn't have high resolution and may not have high quality. We might replace the linear stage in the future.

And in the third section, we designed a soft needle to suck small beads. For my design, there still are some limitation. For example, I used punching to punch holes and to cut PDMS cubes. The shape of those cubes are not perfect cylinder. So we might need to fabricate PDMS cubes by using other tools which are more meticulous in fabricating, like laser cutting.

Now our design is for picking up small beads, which have spherical shapes. When the shape of the target is flat or other irregular shape, we might not pick up the target by using the soft needle, because the membrane may not have a chance to touch with the top surface of the target. And this will not give enough force to pick it up. So in the future, we might change the design of the soft needle a little bit for each target with different shape. For example, for picking up the flat target, we can make the soft needle flat as well to fit for the flat shape. And we might not use the pressure to suck the cells up in the future. Because the dimension of a cell is relative small. For picking up targets with diameters from 50-100μm, we might change from using pressure to use magnetic force combine with soft needle, like adding the magnetic field right above the membrane.

When I was doing experiments about picking up beads by using soft needle, I can easily pick up those beads, but it’s hard to release them. I think it’s because the Ecoflex has high viscosity, which makes the membrane sticky. So when I reduce the pressure, the bead is still stuck to the top of the needle for most times. This situation can be slightly relieved by immerse beads under water. And I don’t know if cells will be stuck to the soft needle too.

For the image processing, I’ve talked about the limitation about the order of the label that we give to each target. And the light intensity and light direction also influence results a lot. It’s inconvenient to adjust the light every time we do experiment. Also, for this method now, we need to separate the target from the background first, then the Bolb analysis could be processed. But when colors of targets are similar to the color of background, it’s hard to separate targets and the background. For our future work about locating cells by using computer vision. It’s challenging to locate cells by using such method. Because cells are transparent, and the boundaries are not clear even under microscope.

And there will be more elements we need to consider about when we pick up smaller objects. When we doing the binary image process, there will be small dots as I mentioned before, when the diameter of target become smaller, the dimension of the target will be close to the dimension of those dots. So when we eliminate those dots by using erosion method (by cutting edges of targets), the target will be eliminated too. So, we need to optimize the image processing method, or avoid these interferences.

6.2 Conclusion and Contributions

A 3-axes Cartesian robot arms with soft needle is designed to pick up small beads with 3mm diameters. We designed the controller for 3 stepper motors and computer vision control in Simulink Real-Time platform. Simulink Real-Time enable us to run all the process on target PC without inputting any new input. With the use of soft robotics, we are able to use soft needle with membrane to pick up soft objects and prevent liquid been sucked in. Combined with the computer vision, we successfully move the top of the syringe to target points and then move to the desired points we want. By using of the vacuum pump, digital relay and solenoid valve, we are able to pick up small beads.

6.3 Future Work

1. Change the design of the soft needle to fit for different objects with different shapes.

2. May replace the linear stage with new one which has higher resolution, for locating and picking up smaller objects.

3. Searching for other methods to successfully read the image on the target PC (Maybe use interruption to grab pictures), makes the system exactly be real-time.

4. Replace the USB 2.0 camera with USB 3.0 camera to increase the transfer speed for sending pictures.

5. Improve the image processing method to label each target with any number that we want. Also make it more efficient for locating targets with any color and smaller dimension. Furthermore, apply the image processing to the real-time control of the movement of three stages.

6. Fix the motor or encoder by replacing the motor or encoder to be able to draw trajectories by using readings from encoders. Replacing to a better motor can increase the moving speed

7. Apply the full MIIC technique to the system to achieve a high-speed motion control.

Reference

[1] Yan Yan1, Qingze Zou1,3 and Zhiqun Lin2. A control approach to high-speed probe-based nanofabrication. 2009 American Control Conference Hyatt Regency Riverfront, St. Louis, MO, USA June 10-12, 2009.

[2] Dr. Filip Ilievski Dr. Aaron D. Mazzeo Dr. Robert F. Shepherd Dr. Xin Chen Prof. George M. Whitesides. Soft Robotics for Chemists. Angewandte Chemie International EditionVolume 50, Issue 8.

[3] Anis, Y. H., Holl, M. R., & Meldrum, D. R. (2010). Automated selection and placement of single cells using vision-based feedback control. IEEE Transactions on Automation Science and Engineering, 7(3), 598-606.

[4] Kyong-Soo Kim and Qingze Zou. A Modeling-Free Inversion-Based Iterative Feedforward Control for Precision Output Tracking of Linear Time-Invariant Systems. IEEE/ASME TRANSACTIONS ON MECHATRONICS, VOL. 18, NO. 6, DECEMBER 2013.

[5] Moeto Nagai *, Keita Kato, Kiyotaka Oohara and Takayuki Shibata. Pick-and-Place Operation of Single Cell Using Optical and Electrical Measurements for Robust Manipulation. Micromachines 2017, 8(12), 350.

[6] Jae Hyeok Choi, Adebola O. Ogunniyi, Mindy Du, and Minna Du, Marcel Kretschmann, and Jens Eberhardt, J. Christopher Love. Development and Optimization of a Process for Automated Recovery of Single Cells Identified by Microengraving. DOI 10.1002/btpr.374 Published online January 8, 2010 in Wiley InterScience.

[7] Mariangela Manti, Taimoor Hassan, Giovanni Passetti, Nicolo` D'Elia, Cecilia Laschi, and Matteo Cianchetti. A Bioinspired Soft Robotic Gripper for Adaptable and Effective Grasping. SOFT ROBOTICS Volume 2, Number 3, 2015 a Mary Ann Liebert, Inc. DOI: 10.1089/soro.2015.0009.

[8] Wei Wang, Sung-Hoon Ahn. Shape Memory Alloy-Based Soft Gripper with Variable Stiffness for Compliant and Effective Grasping. SOFT ROBOTICS Volume 4, Number 4, 2017 ªMary Ann Liebert, Inc. DOI: 10.1089/soro.2016.0081.

[9] Artificial cilia for microfluidics- Sylgrad-184 spin-coating

http://willem.engen.nl/uni/intern-mbx/material/Sylgard-184-spincoat.php

[10] https://www.mathworks.com/products/simulink-real-time.html

Acknowledgement

Thanks for instruction from Prof Qingze Zou, Prof Aaron Mazzeo.
Thanks for all the help from Fan Wu, Jingren Wang, Xiyue Zou, Chirstopher Di Paola, Parth Patel, John Petrowski, Tianwei Li, Jiarong chen.