ROBOT ARM PROJECT REPORT: EGT 320 ROBOTIC SYSTEMS AND MATERIAL HANDLING

Johnny Rivera
Northern Kentucky University
Kentucky, USA

Elijah Herron
Northern Kentucky University
Kentucky, USA

Jacob Weber
Northern Kentucky University
Kentucky, USA

ABSTRACT

Automation through programs to make machines do the heavier payload is becoming a prominent field in the industry. Though the machines require maintenance to maintain their uptime and troubleshoot overseen variables it is an efficient way to reduce the time it takes to complete a task without human intervention.

Keywords: Automation, programs, uptime.

NOMENCLATURE

α	alpha
θ	theta
a	y-displacement
d	x-displacement

1. INTRODUCTION

Our task for the past couple of weeks in our robotics class EGT 320 at Northern Kentucky University (NKU) was to incorporate some of our learned concepts. Some of these concepts include displacement, rotational displacement, and matrix calculations to find out how to reach out endpoint. In order to test these concepts, our group received a robotic arm from our professor to build and test.

2. MATERIALS AND METHODS

The following report will detail our findings and our methods for automating a simple task for our robotic arm. The simple task we automated was to program the robot to pick up small wooden blocks that had letters on them and set them in their designated location. The location would be determined by the word that was being spelled out, in our case we used the word "Robot". Since the robotic arm did not have a sensor to make an advanced version of the program where we would detect the letter it was picking we had to place them in designated starting locations as well.

The method we used to figure out the location of the scrabble board game letter pieces was "manual coding". This was an option embedded inside the Hiwonder's xArm Software that allowed the current position of each servo motor to be recorded as a direction in the program. These directions would be recorded in order, downloaded to the robot, and then tested by allowing the robot to follow the path we set through the recording.

The following list shows all the materials we used during this lab experiment:

- Hiwonder's xArm Programmable Robotic Arm[1]
- Scrabble Board Game Letter Pieces
- Scrap Paper
- Pencil/Pens
- Duct Tape
- Laptop with Hiwonder's xArm Software

2.1 Procedure

1. Cleared an area for the robot to be stationed and the environment to be set up.
2. Attached two papers on each side of robot and held them down with duct tape.
3. Placed scrabble letter pieces in a random order in set locations.
4. Marked locations with pen/pencil for test reference
5. Once the testing environment is set up we startup the Hiwonder's xArm Software and started programming.
6. For Each position we needed to record we would manually move the robot to that position and click "Read Angle".
7. Record all steps required and test if the robot would successfully place the blocks in their designated spots.

3. RESULTS AND DISCUSSION

3.1 Arm Simulation

To build up experience with robot arms we were tasked to do a simulation on the software Maplesim [2]. The simulation would need to demonstrate the degrees of freedom the arm had given its initial position and movement path. The following diagram shows the robot arm we had to construct on Maplesim and test:

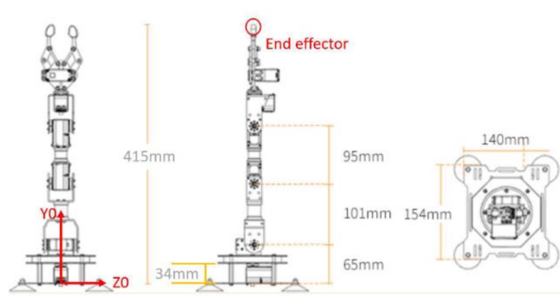


FIGURE 1: ROBOT ARM DIMENSIONS

The following video demonstrates our simulation of this open-mechanism robot arm.

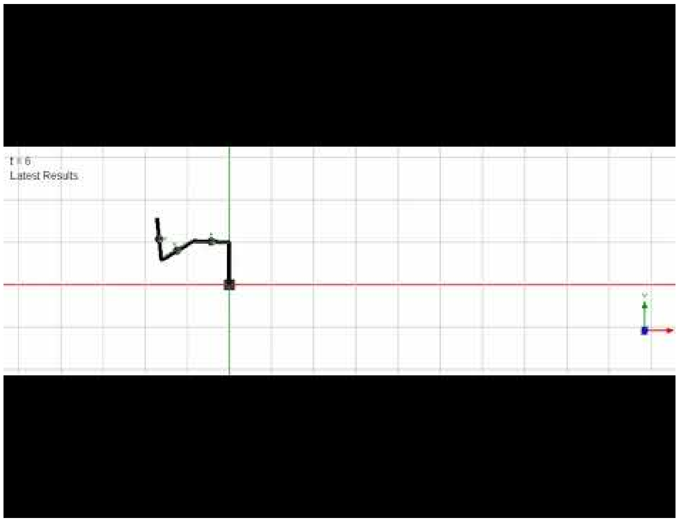


FIGURE 2: ROBOT ARM SIMULATION

Through the use of this simulation software, we were able to familiarize ourselves with setting initial conditions for robotic arms to use in our project.

3.2 Calculation

A week after discussing the simulation results we were tasked to build a program in Excel [3] that would facilitate the calculation of the movements of the robot arm. To calculate the movement and displacement of the robot we used Homogeneous Transformation Matrices (HTMs). The following picture shows the equations needed to find the HTMs for each joint:

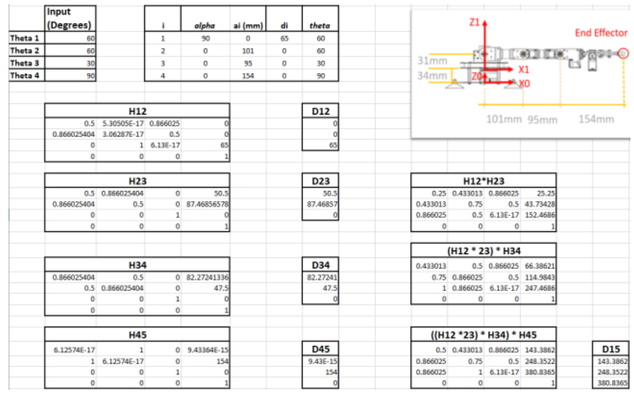
FIGURE 3 (BOTTOM-LEFT): DENAVIT-HARTENBER PARAMETER TABLE

i	α_i	a_i	d_i	θ_i
0				
1				
2				
3				

$$H^i_{i+1} = \begin{array}{c} \begin{array}{cccc} & \boldsymbol{R^i_{i+1}} & & \boldsymbol{d^i_{i+1}} \end{array} \\ \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

FIGURE 4 (TOP RIGHT): HTM TABLE

The calculations needed to find theta and alpha required us to use a Denavit-Hartenberg parameters table. This would allow us to get the rotational and displacement values as long as we give the current joint rotation denoted as theta for each joint.



	Input (Degrees)
Theta 1	60
Theta 2	60
Theta 3	30
Theta 4	90

i	alpha	ai (mm)	di	theta
1	90	0	65	60
2	0	101	0	60
3	0	95	0	30
4	0	154	0	90

H12			
0.5	5.30505E-17	0.866025	0
0.866025404	3.06287E-17	0.5	0
0	1	6.13E-17	65
0	0	0	1

D12
0
0
65

H23			
0.5	0.866025404	0	50.5
0.866025404	0.5	0	87.46856578
0	0	1	0
0	0	0	1

D23
50.5
87.46857
0

H12*H23			
0.25	0.433013	0.866025	25.25
0.433013	0.75	0.5	43.73428
0.866025	0.5	6.13E-17	152.4686
0	0	0	1

H34			
0.866025404	0.5	0	82.27241336
0.5	0.866025404	0	47.5
0	0	1	0
0	0	0	1

D34
82.27241
47.5
0

(H12 * 23) * H34			
0.433013	0.5	0.866025	66.38621
0.75	0.866025	0.5	114.9843
1	0.866025	6.13E-17	247.4686
0	0	0	1

H45			
6.12574E-17	1	0	9.43364E-15
1	6.12574E-17	0	154
0	0	1	0
0	0	0	1

D45
9.43E-15
154
0

((H12 *23) * H34) * H45			
0.5	0.433013	0.866025	143.3862
0.866025	0.75	0.5	248.3522
0.866025	1	6.13E-17	380.8365
0	0	0	1

D15
143.3862
248.3522
380.8365

FIGURE 5: EXCEL PROGRAM

The HTM tables allow us to see the change in each joint from position to position. It shows how they relate to one another and not just the initial positioning allowing for movements to be made from the ending of a previous one.

3.3 Robot

Now with further understanding of the programming and mathematics behind making the arm rotate and translate objects we were tasked to have the robot create a shape or word with blocks it would move independently. The task was initially simple with the robot's software allowing movement to be performed off of several manually determined saved positions; it was not reliable. We had drawn out spaces and created a stable environment for the robot to have repeatable trials by taping pieces of paper down and drawing the potions the blocks the robot would be manipulating would sit on.

FIGURE 6: PROJECT VIDEO

The jittering of the servos in the arm was making it difficult to replay the task with successive working trials. Several manual modifications to the code allowed us to eliminate the most problematic issues like pieces dropping or turning when moving.

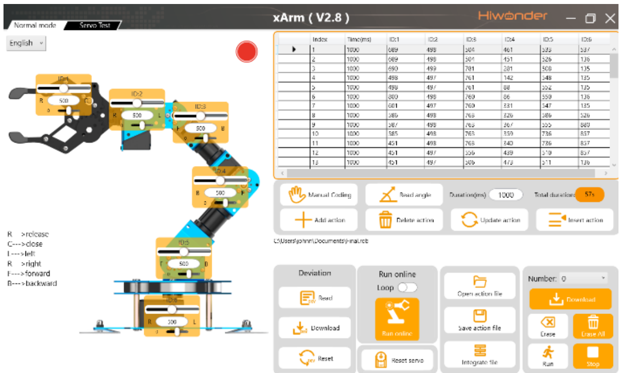


FIGURE 7: HIWONDER'S xARM SOFTWARE

Results

For our final result, we had the robot arm pick up scrabble pieces from one table and move them to another table spelling out the word robot. This was achieved by manipulating the position of the robot manually and using the Hiwonder's xArm Software. This allowed for great control of the robot's movement and allowed for the robot to complete its task. Along the way, we encountered multiple issues involving the movement of the robot's arm and grip. We ran into issues including erratic arm movement and grip strength. The first issue was regarding erratic. The arm was moving unpredictably which resulted in issues with the robot being able to complete its function. We were able to solve this issue by isolating the robot's movements and breaking them down into simpler movements the robot was able to better understand and execute. We found that giving the program smaller steps would allow for greater control and precision for the robot arm. Another issue presented by the program was the grip of the robot, during many trials runs the robot would pick up the scrabble piece but drop it as it was moving the letters. We experimented with different angles and placements for our pieces to allow for the smoothest pickup and release. We also made sure to increase the grip strength of the robot to keep it from falling or spinning while in motion. Through these many different trials and experiments, we were able to successfully program the robot to pick up scrabble pieces from one table and spell out the word robot on the other table.

4. CONCLUSION

Through working on the simulation, calculation, and hands-on labs we were able to gain a better theoretical and practical understanding of how robots can function. The simulation built a good foundation on what to expect from the robot. The calculation showed what was allowing the robot to function. The hands-on lab brought it all together to execute programming on an actual robot arm.

REFERENCES

[1] Nearing, Anne G. *ASME Style Guide*, 2018

[2] *Engineering simulation software*. Maplesoft. (n.d.). Retrieved May 12, 2022, from https://www.maplesoft.com/solutions/engineering-simulation-software/?p=TC-5531&gclid=Cj0KCQjw4PKTBhD8ARIsAHChzRKsWe-SryfhPCrpQImnXBvEGTLSsskcy7ylGWyiYaWtAcVvNafpiFgaAtqVEALw_wcB

[3] Microsoft Corporation. (2018). *Microsoft Excel*. Retrieved from https://office.microsoft.com/excel