EE/ME144 Analytical Project Report 2020

Bhavin Shah
Department of Mechanical Engineering
University of California, Riverside
Riverside, United States of America
bshah002@ucr.edu

Jason Weiser
Department of Electrical Engineering
University of California, Riverside
Riverside, United States of America
jweis012@ucr.edu

***Abstract* — This robot is designed to ease the accessibility of picking fruits or vegetables from their individual plant. Using different methods of forward, inverse, velocity, and statics analysis we can determine the moveability of this arm with respect to a constrained workspace. The following report explains how we went through each process methodically and implemented some of the ideas into code in order to understand how the robot is configured with respect to space and time.**

I. INTRODUCTION

As society struggles with questions of sustainability and fair resource distribution, technical development presents many opportunities to address these problems, often much faster than political consensus can be developed. We identify mechanized agriculture as one such opportunity.

Aeroponics offers a way to use 95% less water with mists of 20μm water droplets, a size which is optimized for better root uptake. Vertical stacking of systems saves space. Contained work areas better manage damaging fertilizer runoff and reduce pesticide use.

In this paper, we explore an inexpensive approach to reducing labor hours required to process these systems.

We implement our design in a fixed space for simplicity, but the natural extension of our work would slide along hundreds of meters of rows. Solutions to this problem require optimized energy use with each element of our set of movements. We understand these solutions through kinematics and statics modeling.

II. PROBLEM STATEMENT

Our robotic arm should be able to carry out efficient, repetitive actions in a 2 meter tall, 2 meter wide local task space. Consider the 5 DoF robot model shown in Figure 1. For our criteria, we proposed a 0.9 meter tall, 0.6 wide robot. It contains 2 prismatic joints and 3 revolute joints. $\theta 1 \in [0,1]$ *meters* denotes the first prismatic joint with limits between [0,1] meters. $\theta 2 \in S$ denotes the second revolute joint. $\theta 3 \in S$ denotes the third revolute joint with limits [0, π]. $\theta 4 \in S$ denotes the fourth revolute joint also with limits of [0, π]. Finally, $\theta 5 \in [0,1]$ *meters* denotes the fifth prismatic joint with limits between [0,1] meters.

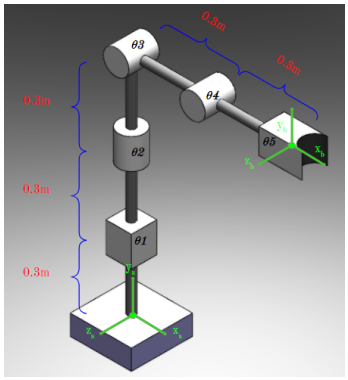

Fig 1. Robot Model

III. TECHNICAL APPROACH

To implement our proposed system, several derivations are necessary in order to analyze how the robotic arm works in space and time.

Forward Kinematics Derivation:

Knowledge of our end effector position through our actuator positions gives us a useful tool in constructing our PID controllers. It's also a basic tool in optimizing our models and in troubleshooting our equipment.

We implement calculations for forward kinematics through a product of exponentials. We use the space frame. We derive the axis of rotation (w) and the axis of joint extension (v), along with points (p) representing each joint shown in Figure 2. We represent the home position of our arm in M and calculate for *n* joints their transformation matrices $T_{i,i+1}$, $i \in n$.

TABLE I
SCREW AXES

JOINT	w	q	V = -w x q
1	(0, 0, 0)	__	(0, 1, 0)
2	(0, 1, 0)	(0, 0.6, 0)	(0, 0, 0)
3	(0, 0, 1)	(0, 0.9, 0)	(0.9, 0, 0)
4	(0, 0, 1)	(0.3, 0.9, 0)	(0.9, -0.3, 0)
5	(0, 0, 0)	__	(1, 0, 0)

Fig 2. Table of Screw Axes

$$T(\theta) = e^{[S1]\theta1} * e^{[S2]\theta2} * e^{[S3]\theta3} * e^{[S4]\theta4} * e^{[S5]\theta5} * M \quad (1)$$

Inverse Kinematics Derivation:

Our system requires visiting a constantly changing set of target points. Calculating joint configurations for desirable paths along these points is a fundamental part of our project.

The requirement of vertical movement at our arm's base necessitates angle constraints if we require unique solutions, which we do. Also, we want to have some sort of rail system guiding the robot on the base laterally so that is why the x-axis distance will be constrained to [0.3,0.6] meters, y-axis has a limit of [0.9,1.5] meters, and the z-axis would stay at a zero degree revolution.

This follows from considering that analytic solutions to inverse kinematics use coordinates at the end effector to influence the relationship between angles within the arm. A prismatic joint unilaterally changing an entire dimension's contributions makes our analytic solution have infinitely many solutions. Figure 3 shows the workspace of our θ values and how we constrained two of the revolute joints.

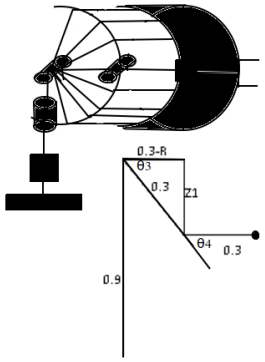

Fig 3. Inverse Workspace of Joint Connections

$$R = \sqrt{x^2 + y^2} \quad (2)$$

In our case, we can make $\theta3$ and $\theta4$ equivalent to create co-linear, knowably spaced joints in our upper arm, which leaves all of our vertical uncertainty on joint1 and the vertical contributions from the other joints dependent on their horizontal components.

$\theta2$ is simply a function of (xz). Our $\theta5$ is left unsolved for simplicity, as it is our end effector. It would be solved in the set of actions relating to picking or pruning.

$$\theta1 = z + z1 - 0.9 \quad (3)$$

$$\theta2 = tan^{-1}(\frac{z}{x}) \quad (4)$$

$$\theta3 = \theta4 = cos^{-1}(\frac{0.3 - R}{0.3}) \quad (5)$$

Velocity Kinematics Derivation:

Determining the joint velocities of our set of desired actions is an important part of engineering decisions like the voltage ranges of our motors, or in time considerations for task planning. We use a graphical Jacobian to obtain our joint velocities from necessary position orientations.

We built the Jacobian using individual S-frames, which examine sequential configurations of non-zero, through each frame shown in Figure 4.

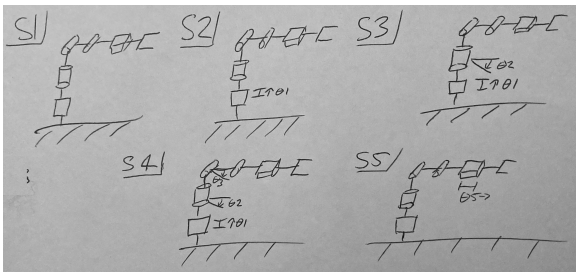

Fig 4. Frames S1,S2,S3,S4,S5 (Space Frame)

The equation of velocity kinematics is:

$$V = J_s(\theta) * \theta' \quad (6)$$

Using this formula, we can figure out the Jacobian matrix and substitute it back into (5). The final *[6x1]* matrix is a twist vector of the end-effector. We denote the sine and cosine function as *s1* and *c1* with the appropriate θ number.

$$V = \begin{bmatrix} 0 & 0 & s2 & s2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & c2 & c2 & 0 \\ 0 & 0 & 0.9c2 & c2(0.9 + 0.3c3) & 1 \\ 1 & 0 & 0 & 0.3c2c3 & 0 \\ 0 & 0 & -0.9s2 & -s2(0.9 + 0.3c3) & 0 \end{bmatrix} \begin{bmatrix} \theta1' \\ \theta2' \\ \theta3' \\ \theta4' \\ \theta5' \end{bmatrix} \quad (7)$$

$$V = \begin{bmatrix} s2\theta3' + s2\theta4' \\ \theta2' \\ c2\theta3' + c2\theta4' \\ 0.9c2\theta3' + 0.9c2\theta4' + 0.3c2c3\theta4' + \theta5' \\ \theta1' + 0.3c2c3\theta4' \\ -0.9s2\theta3' - 0.9s2\theta4' - 0.3s2c3\theta4' \end{bmatrix} \quad (8)$$

Statics Derivation:

Payload weights for our system range from 50 grams for a cherry tomato to 500 grams for an extra large onion, so

we tested for the maximum weight. Design of our joint strength and motor power requires an analysis of static forces. Using the equation of moments we can find the wrench vector and the Jacobian is found by the screw axes.

$$\tau = J_s(\theta) * \mathcal{F}_s \quad (9)$$

$$J_s(\theta) = [S1\ S2\ S3\ S4\ S5] \quad (10)$$

$$\mathcal{F}_s = [m_s\ f_s]^T \quad (11)$$

$$m_s = r_s\ x\ f_s \quad (12)$$

We use a non-zero configuration in order to calculate the space Jacobian as shown below in Figure 5 to get a symbolic result.

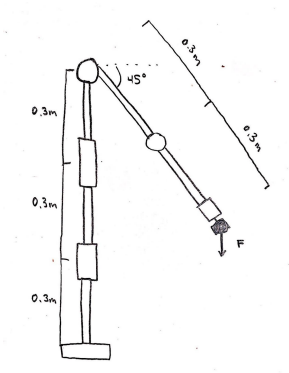

Fig 5. Non-zero Configuration

Using this angle with respect to the positive directional values, we came up with the following equation after substituting numerical values into the wrench vector (11) and plugging in the screw axes for each joint in the Jacobian (10):

$$\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.9 & 0 & 0 \\ 0 & 0 & 1 & 0.9 - 0.3\sin(-\theta) & -0.3\cos(-\theta) & 0 \\ 0 & 0 & 0 & \cos(-\theta) & -\sin(-\theta) & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ -2.49\cos(-\theta) \\ 0 \\ -4.9 \\ 0 \end{bmatrix} \quad (13)$$

Each of the columns in the first matrix of formula (13) are the respective screw axes, and the second matrix simply consists of the moment and force vector in one.

IV. RESULTS

The resulting values we received from our data were mostly as expected. Inverse Kinematics is the only analysis that seemed to be challenging due to the fact that our last joint is prismatic. Using proper constraints helped us minimize the error of going out of our domain range. The following results show how each process was done using python code all with respect to space frame $[x_s, y_s, z_s]$.

Forward Kinematics: using formula (1), input values for $[\theta1, \theta2, \theta3, \theta4, \theta5]$ were given from the user and the script calculator the end-effector position. Figure 6 shows results.

```
Enter translational value for joint 1 [0, 1m]: 0.1
Enter rotational value for joint 2 [0, 2pi]: 0
Enter rotational value for joint 3 [0, pi]: 0.8
Enter rotational value for joint 4 [0, pi]: 0.8
Enter translational value for joint 5 [0, 1m]: 0.2
The position of the end effector is:
0.194412251654
1.71499362879
0.0
```

Fig 6. Python Output for Forward Kinematics

Inverse Kinematics: using formulas (2), (3), (4), and (5) we can solve for the joint angles given a random coordinate for the end-effector frame $[x_b, y_b, z_b]$ by the user. We tested several values due to the fact that our z axis was constraint to zero. Figure 7 shows results.

```
Enter x,y,z values:
[0.6, 0.9, 0]
The positions of theta are:
[0.0, 0.0, -0.0, -0.0]
Enter x,y,z values:
[0.3, 1.2, 0]
The positions of theta are:
[0.6, 0.0, -1.5707963267948966, -1.5707963267948966]
Enter x,y,z values:
[0.3, 1.3, 0]
The positions of theta are:
[0.7000000000000001, 0.0, -1.5707963267948966, -1.5707
963267948966]
Enter x,y,z values:
[0.3, 1.4, 0]
The positions of theta are:
[0.7999999999999999, 0.0, -1.5707963267948966, -1.5707
963267948966]
```

Fig 7. Python Outputs for Inverse Kinematics

Velocity Kinematics: using formula (7), we assumed a time of *[0, 3] seconds* and values such as $\theta(t) = [0, 0, 0, 0, 0]^T$ and $\theta'(t) = [0.03, 0.174, 0.12, 0.135, 0.03]^T$ were inputted to find the end-effector's twist vector. Figure 8 shows results.

$$V = J_s(\theta)\,\theta' = \begin{bmatrix} s2\theta3' + s2\theta4' \\ \theta2' \\ c2\theta3' + c2\theta4' \\ 0.9c2\theta3' + 0.9c2\theta4' + 0.3c2c3\theta4' + \theta5' \\ \theta1' + 0.3c2c3\theta4' \\ -0.9s2\theta3' - 0.9s2\theta4' - 0.3s2c3\theta4' \end{bmatrix} = \begin{bmatrix} 0 \\ 0.174 \\ 0.255 \\ 0.3 \\ 0.0705 \\ 0 \end{bmatrix}$$

Fig 8. Hand Calculations for Velocity Kinematics

Statics: using formula (13), we can calculate the torques required to hold a maximum mass of 500 grams when the robot is in the configuration shown in Figure 5. Figure 9 shows results.

$$\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.9 & 0 & 0 \\ 0 & 0 & 1 & 0.9 - 0.3\sin(-45) & -0.3\cos(-45) & 0 \\ 0 & 0 & 0 & \cos(-45) & -\sin(-45) & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ -2.49\cos(-45) \\ 0 \\ -4.9 \\ 0 \end{bmatrix} = \begin{bmatrix} -4.9 \\ 0 \\ -1.7607 \\ -0.7212 \\ -3.4648 \end{bmatrix}$$

Fig 9. Hand Calculations for Statics

V. CONCLUSIONS

This paper should offer the reader several tools to implement a harvesting arm as a part of their mechanized agricultural system. Due to some constraints in our workspace, the inverse kinematics was very limited to what type of solutions would arise. However, an understanding of concepts such as forward, inverse, velocity and statics analysis should be explicitly shown.

In regards to future remarks, logical next steps of this process include adapting your arm to operate in a warehouse designed to maximize plant density. Our revolute joint near the base would certainly be replaced with a system of rails for movement throughout your system. Optimally, plant output is transported mostly by ramps and belts, and moved there directly from the plant. Additionally, efficient arm pathing requires controllers derived from convex optimization.

ACKNOWLEDGEMENTS

Professor Karydis Konstantinos from University of California, Riverside gave interesting and detailed views of concepts about robotics through recorded lecture sessions. The TA and classmates during lecture and lab hours were helpful in regards to asking keen questions at some times.

REFERENCES

[1] K. Karydis, Lecture Notes in EE/ME144: Introduction to Robotics Course, University of California Riverside, 2020.

[2] Lynch, Kevin, and Frank Park. *Modern Robotics: Mechanics, Planning, and Control*. 1st ed., Cambridge University Press, May 2017.