RBE501 Course Project Report

Jennings, Liam
ldjennings@wpi.edu

Laman, Benjamin
bclaman@wpi.edu

Rooney, Samuel
srooney@wpi.edu

December 3, 2023

Abstract

The purpose of this project is to design a robot arm that can move a 500 gram payload between three different known positions in its planar workspace. The design analysis includes not only the basic ability of the robot to reach the three positions, but also the torques, inertia, and trajectories (including velocity and acceleration) to reach the positions in a defined amount of time. Finally, motors must be selected that are rated for the speeds and torques we will provide. The target positions, shape and mass of the payload, and kinematic constraints to reach our target positions are all given in the assignment description.

1 Background

The three positions are defined in our assignment by the resulting transformation matrices. These matrices are as follows (with units in meters):

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0.05 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$T_3 = \begin{bmatrix} 0 & 1 & 0 & 0.15 \\ -1 & 0 & 0 & 0.075 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The frames are defined with the X axis parallel to the floor and the Z axis facing vertically away from the floor. The frames are numbered from left to right as Pose 1, Pose 2, and Pose 3.

The time taken to move between Pose 1 and Pose 2, and from Pose 2 to Pose 3, are defined. The motion between Pose 1 and Pose 2 will take 1 second, and the motion between Pose 2 and Pose 3 will take 1.5 seconds. A smooth trajectory must be generated between these poses, and the dynamic torques evaluated to ensure they don't exceed the capabilities of our motors.

Our payload is a single part defined as two cylinders end to end with a total mass of 500 grams. The "payload frame" is the center of frame b, meaning it's defined by the above transformation matrices. A drawing of the payload can be found in Figure 1.

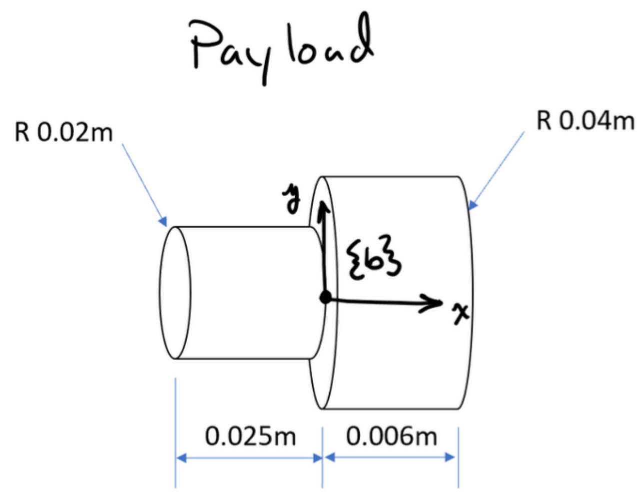

Figure 1: Drawing of the payload from our project assignment

An interesting side note is that the density of this payload can be calculated to determine what material it might be made of. This was done using Solidworks, and came out to approximately 8000 kilograms per cubic meter, or the same density as steel.

To reiterate the above in a more concise format, we have five requirements for this project. 1. Kinematic design and analysis of a robot arm which can reach the three poses without the robot colliding with itself 2. A CAD model of the robot links and joints in order to find the mass properties and inertia matrices of the links 3. Smooth trajectory generation between the positions (quintic-time linear-path trajectories) 4. A dynamic analysis of the robot arm to find the joint torques required to follow the trajectories 5. Selection of motors which are capable of producing the required torques, accelerations, and velocities.

2 Methods

2.1 Kinematic Design and Analysis

The first property of our robot we must consider is it's required task space. Based on the target poses assigned in the project, we only need a planar robot to complete the task. That is the primary constraint surrounding our analysis here and will simplify our analysis.

The second objective is to find the link lengths for the robot. This can be done by sketching out all three of our target positions in CAD, then creating a three-segment chain which follows our payload constraints (position and angle) and adjusting the lengths of the segments until we are satisfied.

Third, we need to find the forward kinematics for our robot. This is done by finding the zero configuration M, as well as the screw axes. With these pieces of information we fully define our forward position kinematics model under the Product of Exponentials (PoE) formulation.

Fourth, we will define the inverse kinematics for our robot so we can find the proper joint angles to reach a given point. Because our robot has a small number of degrees of freedom, we can do this geometrically using the Law of Cosines and the Law of Sines. These equations do result in two valid solutions (commonly referred to in literature as "elbow up" and "elbow down" configurations), but we only include the "elbow up" configuration due to our limited set of target positions.

2.2 Robot Arm CAD Model

The robot's CAD model is created based on the link lengths and zero position we designed previously, and with some consideration of practical problems around manufacture and assembly. Pictures of the model can be found in the Results section.

2.3 Smooth Trajectory Generation

Smooth trajectories can be generated in either the joint or the task space by simply defining the endpoints and creating a polynomial function which meets the constraints given. In our case, we have constraints on starting time, ending time, starting position, ending position, starting velocity, ending velocity, starting acceleration, and ending acceleration. This is a total of 8 constraints for each movement.

These constraints can be met by defining a quintic polynomial. We can write out the quintic polynomial in matrix format, which allows us to multiply a "time matrix" (see below) by a column vector containing the polynomial coefficients, resulting in a new column vector containing the position, velocity, and acceleration constraints.

We define t as time, p as position, v as velocity, and a as acceleration. The polynomial coefficients are defined as A, B, C, D, E, and F.

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} f \\ e \\ d \\ c \\ b \\ a \end{bmatrix} = \begin{bmatrix} y_0 \\ y_0' \\ y_0'' \\ y_f \\ y_f' \\ y_f'' \end{bmatrix} \tag{4}$$

Equation 4 is analogous to the following symbolic representation in Equation 5.

$$TA = Y \tag{5}$$

By left-multiplying both sides by the inverse of the time matrix, we can solve for the quintic coefficients in A based on our known time and kinematics constraints in T and Y.

$$A = inv(T)Y \tag{6}$$

The inverse operation can be efficiently done by MATLAB or any other linear algebra library, including NumPy, so there is not a need to find the inverse ourselves. This process must be repeated for each coordinate we traverse through; we choose to define our intermediate joint configurations, so each full trajectory will be based on 3 quintic polynomials following the same time constraints but different start and end value constraints.

For our problem, we set the constraints on both starting and ending velocity and acceleration to zero. The time taken for both movements is defined, so we can set the starting time to 0 and the ending time for the defined duration. Finally, the starting and ending positions are defined in the joint space with inverse kinematics.

We choose to define our trajectories in joint space only because it decreases the number of equations we will show in this report. It is trivial to apply the same trajectory generation process to the task space: you simply define the trajectory in terms of your spatial coordinates (x, y, z, alpha) and perform inverse kinematics on the result of the trajectory generation step.

To find the value of the intermediate position in the trajectory, use the next equation.

$$\begin{bmatrix} 1 & t_n & t_n^2 & t_n^3 & t_n^4 & t_n^5 \end{bmatrix} \begin{bmatrix} f \\ e \\ d \\ c \\ b \\ a \end{bmatrix} = y_n \tag{7}$$

2.4 Dynamic Analysis

A dynamic analysis of the robot involves finding the torques required to be generated by the motor to follow our smooth trajectory. To do this, we applied the Recursive Newton-Euler (RNE) algorithm[1], which has two steps.

For each time-step, we analyze the torque needed to hold the motors up against gravity, and separately how much torque is required to accelerate the robot arm against its own inertia. Then we sum the torques, the result of which is a valid solution to the dynamics problem due to the law of superposition of linear systems.

Note that because the payload is rigidly attached to the third link, and there is no outside force acting upon it, it has been abstracted as part of the third link.

3 Results

3.1 Kinematic Design and Analysis

3.1.1 Link Lengths

Link 1: 10cm Link 2: 10cm Link 3: 5cm (to COM)

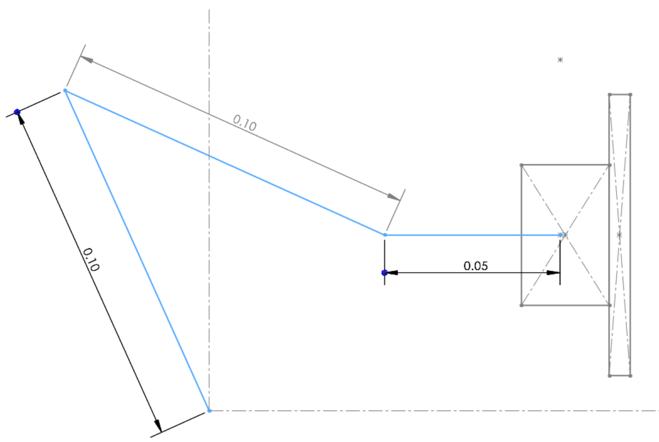

Figure 2: Sketch of arm at pose 1 with link lengths defined

3.1.2 Forward Kinematics

We compute the forward kinematics of our arm using the Product of Exponentials formulation[1]. Our home configuration is defined by the transformation matrix M, and our screw axes are defined by the row vectors in the matrix S. We take the center of the axis of rotation for Joint 1 to be the robot's origin.

$$M = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0.25 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$S = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & 0.2 & 0 & 0 \end{bmatrix} \tag{9}$$

3.1.3 Inverse Kinematics

Rather than using a numerical approach like the Newton-Raphson algorithm, we chose to follow the geometric approach for the inverse kinematics of the arm. There are only two possible solutions for most frames, so we felt that it was advantageous to use the closed form solution.

As there is no control of the orientation, the position of the third joint can easily be found by moving backwards along the target frame's orientation by the length of the third link:

$$T = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}$$
$$P_3 = P - R\,l_3$$

From here the problem is identical to a 2R planar manipulator, with a caveat of the angles being measured from the y axis. In the calculation of the angle of the first link from the y axis, this can be accounted for by switching the order of coefficients in the tangent calculation:

$$\alpha = \mathrm{atan2}(P_{3x}, P_{3y}) \tag{10}$$

Care was taken to capture both the elbow up and elbow down configurations. As the inverse kinematics of a 2R manipulator is a well documented process, it is omitted from the report.

Once the first two angles are calculated, the third can be found by calculating the heading of the end effector:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$
$$\gamma = \mathrm{atan2}(\cos\theta, \sin\theta)$$

And subtracting the other angles:

$$q_3 = \gamma - q_2 - q_1 \tag{11}$$

Using this method, we found these angles for the elbow up position for each frame:

Table 1: Frame Joint Angles

Frame	q_1	q_2	q_3
T_1	0.4240	-2.4189	0.4240
T_2	0.5139	-1.9552	-0.1296
T_3	-0.6578	-0.4365	-2.0473

3.2 Robot Arm CAD Model

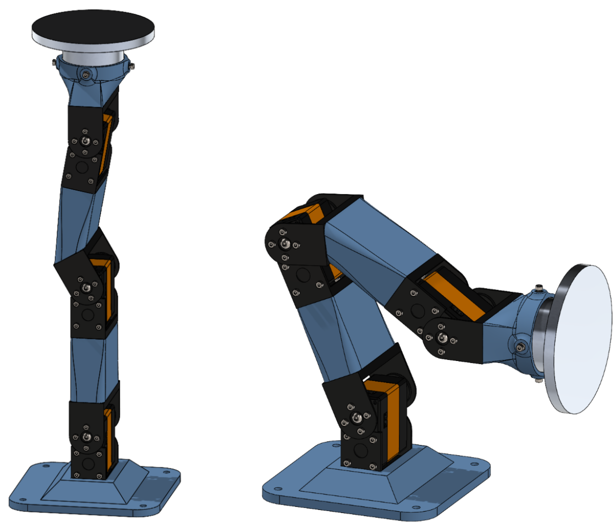

Figure 3: Robot CAD model in Home configuration & Pose 1

3.3 Generation of Smooth Trajectories

By applying Equation 6 to the robot's joint space target positions, we find the coefficients of a quintic trajectory in joint space. We have been assigned to find the trajectories between Position 1 and Position 2 in 1 second, then between Position 2 and Position 3 in 1.5 seconds. The trajectories define intermediate joint configurations and are defined by quintic coefficients. The coefficients are stored as sets of column vectors in one matrix, following the same format as in Equation 4. The time matrices are written here as well for reference. We treat both trajectories as if they begin at "time zero"; that is to say, we start a timer when we begin each movement and pass the elapsed time into the trajectory polynomial. This timer resets between each trajectory.

$$T_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 2 & 6 & 12 & 20 \end{bmatrix} \tag{12}$$

$$A_{12} = \begin{bmatrix} 0.424 & -2.419 & 0.424 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.899 & 4.637 & -5.536 \\ -1.349 & -6.956 & 8.304 \\ 0.539 & 2.7822 & -3.321 \end{bmatrix} \tag{13}$$

$$T_{23} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1.5 & 2.25 & 3.375 & 5.0625 & 7.59375 \\ 0 & 1 & 3 & 6.75 & 13.5 & 25.3125 \\ 0 & 0 & 2 & 9 & 27 & 67.5 \end{bmatrix} \tag{14}$$

$$A_{23} = \begin{bmatrix} 0.514 & -1.955 & -1.296 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -3.472 & 4.500 & -5.682 \\ 3.472 & -4.500 & 5.682 \\ -9.258 & 1.200 & -1.515 \end{bmatrix} \tag{15}$$

3.4 Dynamic Analysis

After performing the dynamic analysis described above, the following figures were obtained:

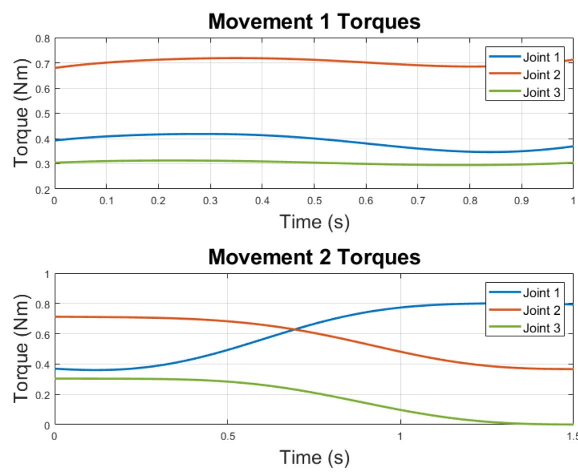

Figure 4: Joint Torques throughout required movements

Using this analysis, we found the maximum torque requirements, detailed in the next section.

3.5 Motor Selection

We found our required motor parameters, then did market research on a number of different motors and came to a final conclusion.

Motor Name	Number	Price
Waveshare ST3215	1	\$22
Hiwonder HTD-45H	2	\$25
Lynxmotion LSS-ST1	3	\$75
HerkuleX DRS-0602	4	\$396

Table 2: Map of motor names to their number in Table 3

No.	Range	Torque	Velocity	Acceleration
Req.	138°	0.8 N*m	2.4 r/s	4.9 r/s/s
1	360°	2.95 N*m	4.71 r/s	>9.42 r/s/s[1]
2	240°	4.4 N*m	5.8 r/s	>11.6 r/s/s[1]
3	360°	1.37 N*m	6.28 r/s	>12.56 r/s/s[1]
4	900°	7.5 N*m	6.38 r/s	>12.76 r/s/s[1]

Table 3: Motor parameters as compared to our project requirements. We refer to velocity and acceleration in radians per second or per second squared, written with r to reduce table width.

We decided that Hiwonder HTD-45H motors fit our application best. While all of the motors we investigated exceed our requirements, these motors have the best price to performance ratio.

Additionally, all of our dynamics calculations were made with the HTD-45H motors as a placeholder, so selecting this motor minimizes additional analysis.

4 Discussion

4.1 Kinematic Design

The kinematic design we came up with for our robot is a fairly standard one. As a group, we decided to set the link lengths to relatively round numbers (0.1 or 0.05 meters), which has an advantage. The advantage is that it makes analysis of the arm slightly easier, especially handwritten analysis.

These easy-to-work-with numbers make validating a computer's calculations a slightly simpler process. However, it's important to note that because we did not choose to minimize the link lengths we do have to apply higher torques with our motors. In our case, this is not an issue, but this could become problematic when designing closer to the rated limit of a motor's torque.

4.2 CAD Model

After we decided on our link lengths, we immediately decided to estimate the maximum torque the robot arm must hold. This allowed us to begin designing around the Hiwonder HTD-45H motors before we technically proved that they were capable of applying the torque we needed. We estimated the torque by doing static calculations in the worst-case configuration for the arm; holding the payload exactly parallel to the ground as far away as the arm could reach.

By estimating link masses, motor mass, and payload mass, we determined that the HTD-45H motors were a reasonable tentative choice, and luckily they ended up arguably the best choice of motor in our final analysis.

By doing this preliminary analysis first, we were able to design an appropriate CAD model for the robot arm with placeholder motors. This gave us the advantage of being able to produce a physical model of the arm, although we did this at the very real risk of the motors not being rated for our final maximum torque and velocity. In the end, we felt it was more valuable to have the physical model and be incorrect.

4.3 Smooth Trajectory

We decided to generate our smooth trajectories in the joint space, which doesn't necessarily make sense for all circumstances. The advantage of generating trajectories this way is that the dynamics analysis is made easier; there are less steps to find joint accelerations and velocities.

Our choice has the significant downside of not allowing us to determine the path we want to follow through the task space. This is important for a number of reasons, two of which are chiefly important. First, we have to perform inverse kinematics anyway to precisely determine our endpoints, so we have an additional step to define our path which could be avoided if we simply perform inverse kinematics at every time step. Second, we are unable to follow complex trajectories, which makes it far more difficult to, for example, design a robot for computer-aided machining. It is very often important to not only define the endpoints a robot finds itself at, but also the intermediate points it passes through.

4.4 Dynamics Analysis

This section is short as we did not make any special decisions about the methods we used to analyze the torques on each joint. It is important to note that these calculations were made with the Hiwonder HTD-45H motors as placeholders. We checked that these motors have comparable mass, size, and torque to most motors in the same market segment, so this was a reasonably well-founded decision.

4.5 Motor Selection

While it's hard to say that the final motor selection was completely unbiased (we have repeatedly noted our use of the Hiwonder HTD-45H motor as a placeholder in many of our calculations), it's a fine choice of motor. With regards to our analyzed trajectories, the HTD-45H motor has a safety factor of approximately 5.5, which means the motors never exceed 25% of their rated torque under normal use; this meets the design recommendations from WPI's own Professor Bertozzi (and also means the motors operate near peak efficiency).

It is notable that the usable range of the motor is less than its competitors, only being able to use 240°rather than the market-typical 360°. However, it still far exceeds the requirement for our problem.

[1] An estimate of the implied acceleration given by the motor's finite position range and maximum velocity. The manufacturers did not provide rated acceleration figures.

5 Conclusion

Our robot arm design project is successful by all stated technical metrics. First, it is capable of reaching all three of our target positions. Second, we created a CAD model which both holds the payload and uses the motors we selected. Third, we designed trajectories which move the payload to its assigned positions within our given time frames and without the robot intersecting itself. Fourth, we analyzed the motor torques required to follow that trajectory. Fifth and finally, we selected motors which are capable of producing those torques and of meeting our velocity and acceleration requirements. The mass of these motors is accounted for in our dynamics analysis.

References

[1] Kevin Lynch and Frank Park. *Modern Robotics: Mechanics, Planning, and Control*. English. 1st ed. Cambridge University Press, May 2017. ISBN: 9781107156302. URL: `modernrobotics.org` (visited on 11/01/2023).