3R Planar Manipulator Design and Analysis

Venkata Sai Krishna*
Department of Robotics Engineering
Worcester Polytechnic Institute
vbodda@wpi.edu

UdayGirish Maradana*
Department of Robotics Engineering
Worcester Polytechnic Institute
umaradana@wpi.edu

Butchi Venkatesh Adari*
Department of Robotics Engineering
Worcester Polytechnic Institute
badari@wpi.edu

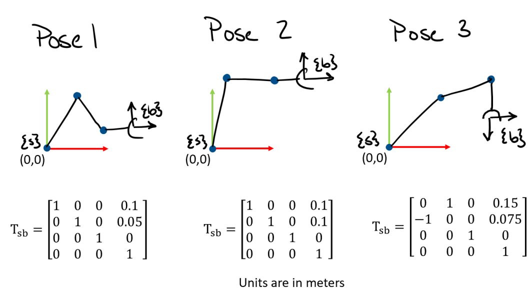

Fig. 1. Task Definition

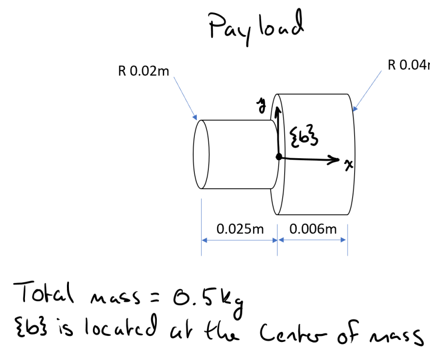

Fig. 2. Load

***Abstract*—Pick and place is a common task in Robot Manipulation. Even though it looks quite simple to pick and place an object for a human , or with a robot with current stack of libraries, it is a hard problem solved over decade's of research. In this paper we discuss about the nuances of the design of a 3R Manipulator which can pick a object and execute a certain trajectory without fail.**

I. Background

This work is a part of Coursework in RBE 501-Robot Dynamics at WPI. The task is to design a 3 DOF where all Rotational joints consists within same rotational axis resulting in a overall planar movement. Therefore, we call this manipulator a 3R Planar manipulator. Other considerations for the successful completion of the project is that the robotic arm should traverse through different poses (Pose1 $->$ Pose 2 $->$ Pose3) with a specified load of 0.5 kg. The Poses are explained in the Figure. 1

The robot should also carry a payload of 0.5 kg of specified dimension which is shown in Figure. 2.

II. Proposed Methods

The approach followed is based on the course learnings from calculating Forward Kinematics, Inverse Kinematics and Trajectory Generation plus the Force and Torque calculation. A brief picture of how the development flow looks like is represented in Figure. 3.

The steps are as follows:

- First we started a parallel exploration of design and implementing analytical calculations on Matlab.
- The kinematics of the robot are designed in such a way that the manipulator reaches all the poses mentioned in Figure 1 without colliding.
- By following above parameters, link one is set to 12 Cm's, link2 is set to 9 Cm's, and link 3 is set to 6 Cm's until the end of the gripper.
- The first joint of the robot can rotate freely between 180 degrees to -180 degrees without colliding. The second joint can rotate between -150 degrees to 150 degrees without colliding and the third joint can move between -130 degrees to 130 degrees without colliding(These approximations are made based on the CAD model of the manipulator, we dwell more into it below).
- The resulting workspace is

$$9.66 \leq \sqrt{x^2 + y^2} \leq 27 \tag{1}$$

- The poses defined in figure 1 are not near the extreme positions which may result in singularity. Thus we can definitely say that all the poses are within robot's workspace and these poses can be reached without collision between links.

Design Process:

- We started with the CAD Design of the links based on the common robotic arm we usually see.
- While designing links we design the end effector link such that the motor can easily fit . We initially chose some random servo without worrying about its specification, as most of the servos of a major brand are almost in same size.

[1]* Equal contribution

- Once the end effector link (Link3) is designed we started the design of the other two links.
- We assigned the material of the link in CAD software and Based on this we obtained initial Mass and Inertia parameters without considering the motors in the final assembly.
- We've set a co-ordinate system at the centre of mass of the link and the moment of inertia of the links we're obtained about this link.
- The Payload is assembled with link3 while calculating mass and moment of inertia of link 3 as they are fixed and can be assumed as a single body making it easier for calculations. The frame is assigned at the combined centre of mass.

Mathematical analysis:

- Initially we started working on the forward kinematics, By assigning frames for each link at the centre of mass of the link aligned with the principle moment of inertia axis.
- The inverse kinematics were performed to find the joint variables at each of the poses using which a trajectory can be generated. We used Newton Raphson method with six iterations method calculating inverse kinematics.
- An approximation is of the angles are drawn by geometrically solving for joint variables and these values are used for the first iteration in the Newton Raphson method to retired accurate and precise joint variables at each pose.
- Once this is done we derived Quintic polynomial equations for all the joint variables for trajectory 1(pose1 to pose2) and trajectory 2(pose2 to pose3) by keeping joint velocity and joint acceleration as 0 at each of these poses. The resulted Quintic polynomial trajectory is plotted and derived to retrieve Joint angles velocities and accelerations. The graphs are presented in the subsequent chapters of the paper.
- Post this, we used Newton-Euler inverse dynamics algorithm to find the torques required at each link to follow the trajectory.
- First newton-Euler forward iterations were applied to get the twists and derivation of twist at each link at every timestamp as the trajectory is being followed. The forward iteration equation we used is defined below, The first twist is defined as [0,0,0,0,0,0] and first twist derivative is defined as [0,0,0,0,g,0] as upward acceleration at base is in the same inertial space as gravity acting downwards.

$$\nu_i = AD_{T_{i,i-1}}(\nu_{i-1}) + \alpha_i \dot{\theta_i} \tag{2}$$

$$\dot{\nu_i} = AD_{T_{i,i-1}}(\dot{\nu}_{i-1}) + \alpha_i \ddot{\theta_i} + ad_{\nu_i} \alpha_i \dot{\theta_i} \tag{3}$$

- The transformation matrix used in the above equation is calculated using product of exponential method at each instance of the trajectory.
- Once link twist at each point in trajectory is calculated, we can go forward with backward iterations to calculate the wrench at each link at each instance of the trajectory.

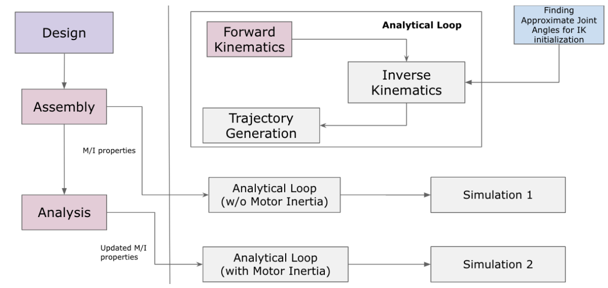

Fig. 3. Approach - Flow Diagram

The following equation is used for that purpose and Ftip is defined as [0,0,0,0,0,0] as the payload is intergrated with link 3.

$$F_i = AD^T_{T_{i+1,i}} F_{i+1} + G_i \dot{\nu_i} - ad^T_{\nu_i} G_i \nu_i \tag{4}$$

- Once the wrench of each link at each instance of the trajectory is calculated, The torque required to be acted at each joint to make the robot follow this trajectory is calculated using below equation.

$$\tau_i = F_i^T * \alpha_i. \tag{5}$$

- Usually, when choosing the actuators we have to take consideration of two main parameters along with generic parameters such as Angular Velocity and acceleration.
  - The two parameters are Continuous torque and Stall Torque.
  - **Continuous toruqe:** Continuous torque is the maximum torque that a motor can continuously deliver without overheating under normal operating conditions.
  - **Stall torque:**Stall torque refers to the maximum torque that a motor can produce when it is operating at standstill or rotating at a very low speed.
- Based on the above parameters we have choosen the actuator (A servo motor). From the design diagrams and a open source model library (**Grabcad**) we got the model of it and added it to the final design and got the updated Mass/inertia properties.
- Once this is done the updated M/I properties are added to the Matlab code and got the final graphs.
- Once we were convinced that the system was working,we connected entire plots to a custom simulation.

III. Results

Based on the above methodology we designed the system and analyzed it with two materials. Further plotted the results and simulated the system.
Before attaching Servo's to the model

- Torque Analysis - Figure. 4 (Torque Min needed - 0.9 Nm)
- Angular Velocity analysis - Figure. 5 (Angular Vel. Min needed - 2.5-3 rad/s => 30 RPM approx)

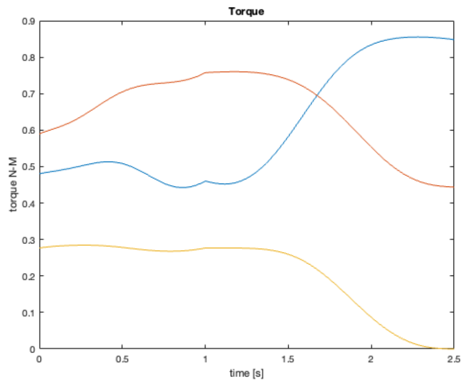

Fig. 4. Torque Analysis- Aluminium - Without Servo

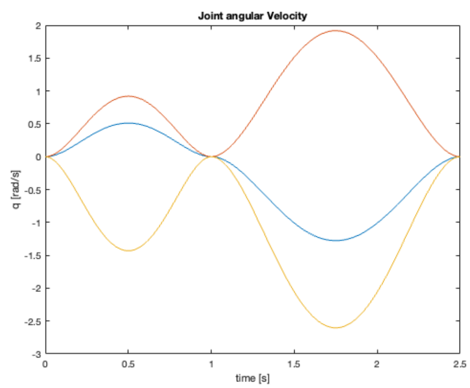

Fig. 5. Angular Vel Analysis - Aluminium - Without Servo

- Angular Acceleration analysis - Figure. 12 (Angular Accel.. Min needed - $5-6 rad/s^2$)
- Based on the above analysis we have found the minimum torque of 1Nm, may be atleast 1.5-2 times that as we add the motor weight and inertia. Based on this conclusion we tried to identify the best motor for our purpose.
- After reading a datasheets of few servo's we identified DYNAMIXEL XM540-W150-R as the perfect choice. As it is having a stall torque of 7.30 Nm and No load speed of 53 RPM. After reading throughly we also understood that it can give continuous torque of more than 3.5 Nm which is more than enough for our purpose. We can go higher with the choices but it will increase the cost. Dynamixel Servo is shown in Figure. 6
- Then the design is updated and the updated full CAD model is shown in the Figure. 7
- After updating the assembly , we again found individual links mass and inertia properties and updated the parameters in Matlab. Once done we found the final results for a aluminum material based design in the figures below:
  - **Torque Plot** - Figure. 8
  - **Angular Velocity plot** - Figure. 9
  - **Angle Plot** - Figure. 10
  - **Joint Position Plot** - Figure. 11

Fig. 6. DYNAMIXEL XM540-W150-R

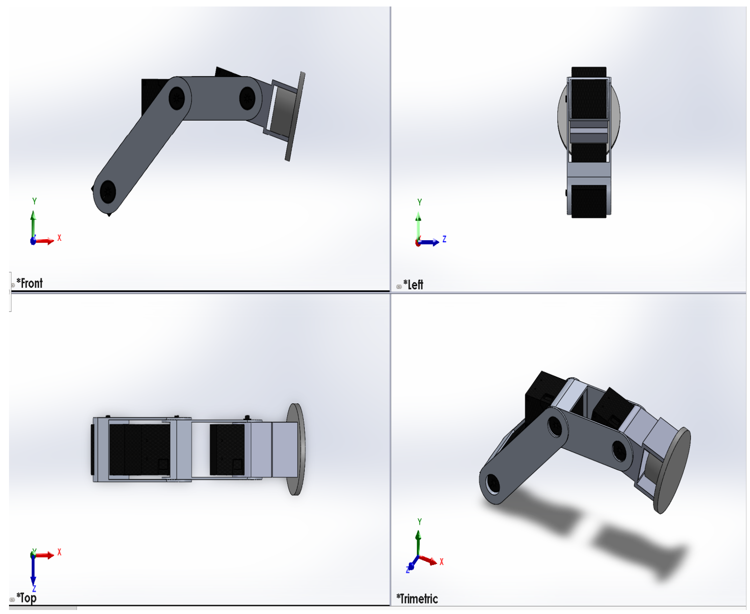

Fig. 7. Final 3R Robotic Arm Assembly

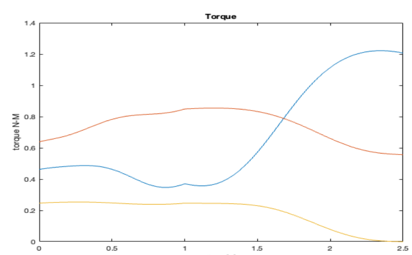

Fig. 8. Torque Plot with Servo in CAD

  - **Angular Acceleration plot** - Figure. 12
  - **Simulation** - Figure. 13

IV. Discussion

- After obtaining the result , we tried to read a bit literature on what is the perfect material to use in Robotics for robotic arm manufacturing , we found that Steel Alloys and Aluminium alloys are pefect choice. Out of that a steel alloy with molybdenum gives more performance. So we tried to compare the results with aluminium alloy

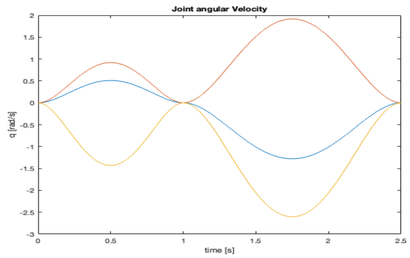

Fig. 9. Angular Velocity Plot with Servo in CAD

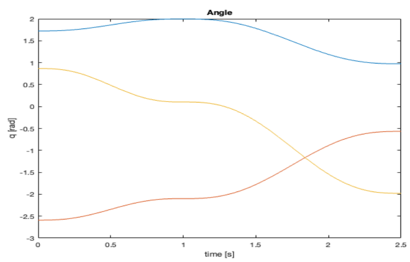

Fig. 10. Joint Angle Plot with Servo in CAD

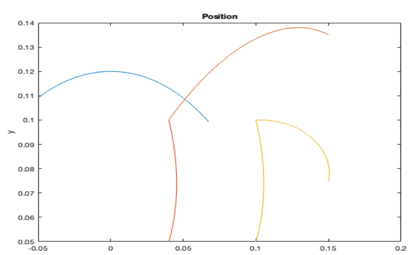

Fig. 11. Joint Position Plot with Servo in CAD

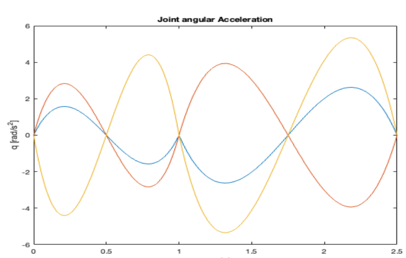

Fig. 12. Angular Acceleration Plot with Servo in CAD

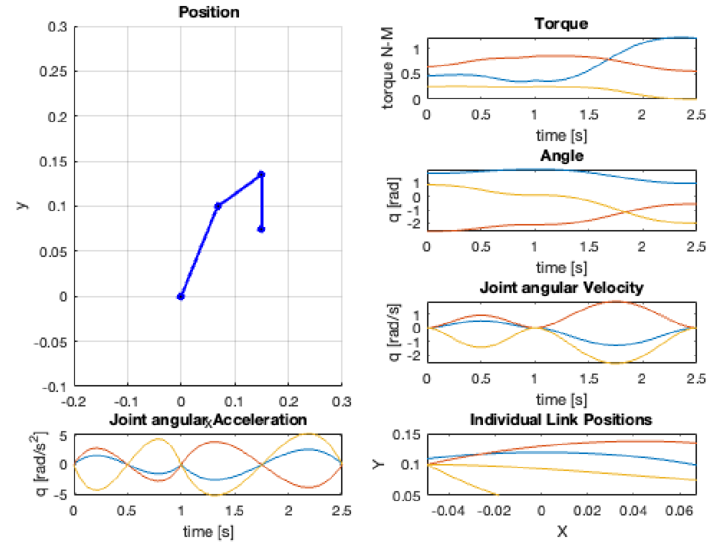

Fig. 13. Simulation

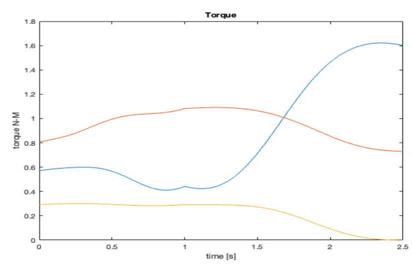

Fig. 14. Torque Plot with Servo in CAD - Steel Alloy

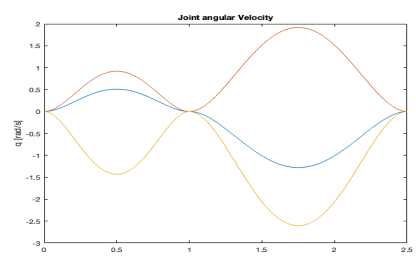

Fig. 15. Angular Velocity Plot with Servo in CAD - Steel Alloy

((1100-O Rod (SS)) and tried to understand whether our current actuator works with new material or not. Surprisingly, even though there is a increase of weight , the current actuator choice actually works.

- Some of the results - Torque and Angular Velocity with the Steel Alloy (AISI-4340 Steel - Annealed).
  - **Torque Plot** - Figure. 14
  - **Angular Velocity plot** - Figure. 15
- If we observe the graphs we observe that the robot is going from Pose1 to Pose2 in 1s and having a stop which is why its angular velocity is zero at that instant and then it slowly moves to Pose3 in next 1.5s. When it reaches the final position we can see the angular velocity and angular acceleration is zero.
- A better view of simulation is present in the Presentation Video and Slides.

V. Limitation of the Project

- Currently, we haven't implemented a control algorithm to optimize the trajectory. But this is currently out of scope for this project as our main aim in this is to learn

about designing a system, modelling its kinematics and dynamics, further achieving the desired trajectory.
- CAD design is not completely optimized and still it is just a prototype version.

VI. Conclusion

- Overall the learnings in the project almost covered most of the topics we are learning throughout the course. The project can further be enhanced by adding a control simulation. One more thing which can be done is adding the CAD model as STL and import it to MATLAB to show the simulation which makes it more realistic. Also , some more things have to be implemented for collision based stopping and impedance control which are more control level topics. Overall, this is a pure full-fledged learning experience.

Acknowledgment

We are thankful to Dr. Christopher Nycz for giving us an opportunity to work on real-world task where we got to apply all our learnings in the course.This project, inspired us to start one more project for fun. We are further planning to work on cool concept to have a Python-based web based GUI where user can place obstacle and enter mass the robot should simulated and show torque at every point.

References

[1] Lynch, K.M. and Park, F.C. (2021) Modern Robotics: Mechanics, planning, and Control. Cambridge: University Press.

[2] GrabCad

[3] Dynamixel Website for Choosing motors.