MAE263B Project 2 part 2

Name: Yuan Hung Lo

1. **MATLAB Script**

   The MATLAB script can be found here:
   https://github.com/garyloee/MAE-263B/tree/edace558150616819c4cff2a87fdc01d627744f4/Project2
   To generate trajectory, input the desired destination tag, such as "reference" and "chip1" or its coordinate with the form: (r_z, p_x, p_y, p_z). The Trajectory Generation function will then output the set of joint positions that completes the whole trajectory.

```
% Feeder center at
Feeder=[0,60,400,100];

% Chips at four corners of PCB
chip1=[0,-45,445,100];
chip2=[-90,45,445,100];
chip3=[-180,45,355,100];
chip4=[90,-45,355,100];
```

```
% trajectory planning
% input goal position with the first variable of the function
[q,dq,ddq]=trajectoryGeneration(viaFeeder,q,dq,ddq);
[q,dq,ddq]=trajectoryGeneration(Feeder,q,dq,ddq);
[q,dq,ddq]=trajectoryGeneration(viaFeeder,q,dq,ddq);
```

   Combining all the generated q (joint configuration) can lead to the complete trajectory and can be used for animation.

2. **Trajectory Animation**

   Both the joint space trajectory and task space trajectory animation is uploaded to canvas. You can also find the online videos in the link here:
   https://drive.google.com/drive/folders/1EwznBC8EDrZoytocMqwcmKZf878NPU--?usp=sharing
   The location and orientation of the chips and feeder are indicated with markers in both animations.

3. **Joint Monitoring**

   To depict a robot arm's dynamic characteristic more accurately, I increased the maximum acceleration for rotational joint 3 and prismatic joint 4. These two joints are at the wrist of the robot arm. While they often experience the most amount of displacement, they endure the least amount of inertia and weight load. Increasing their speed makes the trajectory smoother and more realistic. This can be easily reversed by changing the maxAcc array variable in the script.

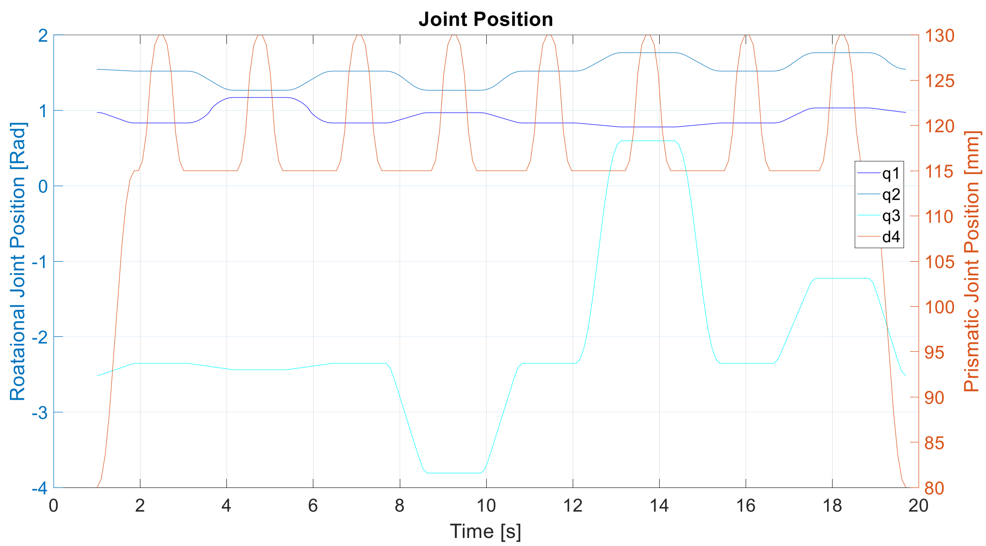
```
maxA=50*pi/180; % maximum acceleration
maxAcc=[maxA,maxA,10*maxA,150];
```

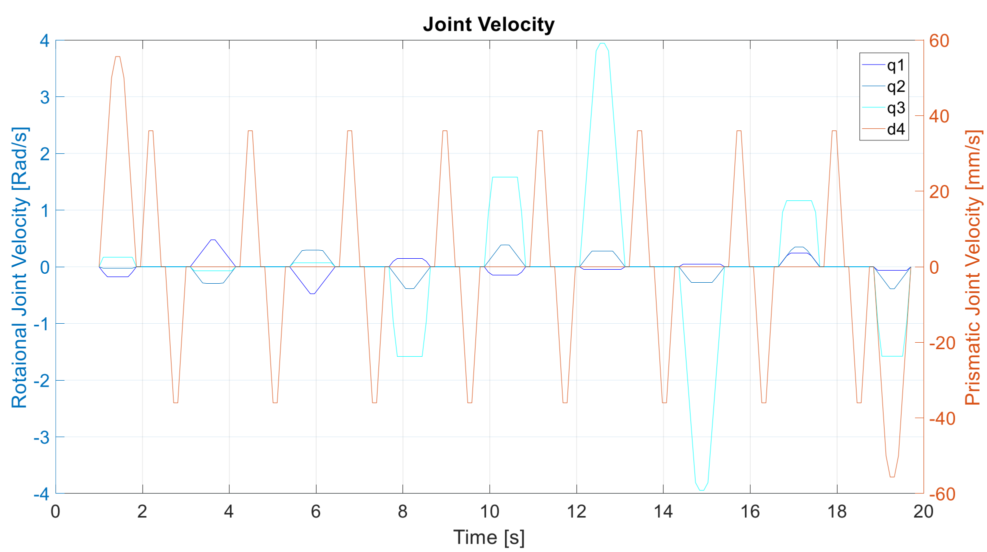
   With the maximum acceleration set, the total time of each trajectory must be precise. Too much time would lead to zero acceleration in the middle of the trajectory, wasting potential speed. While too little time would inhibit the joint's ability to reach its correct destination.

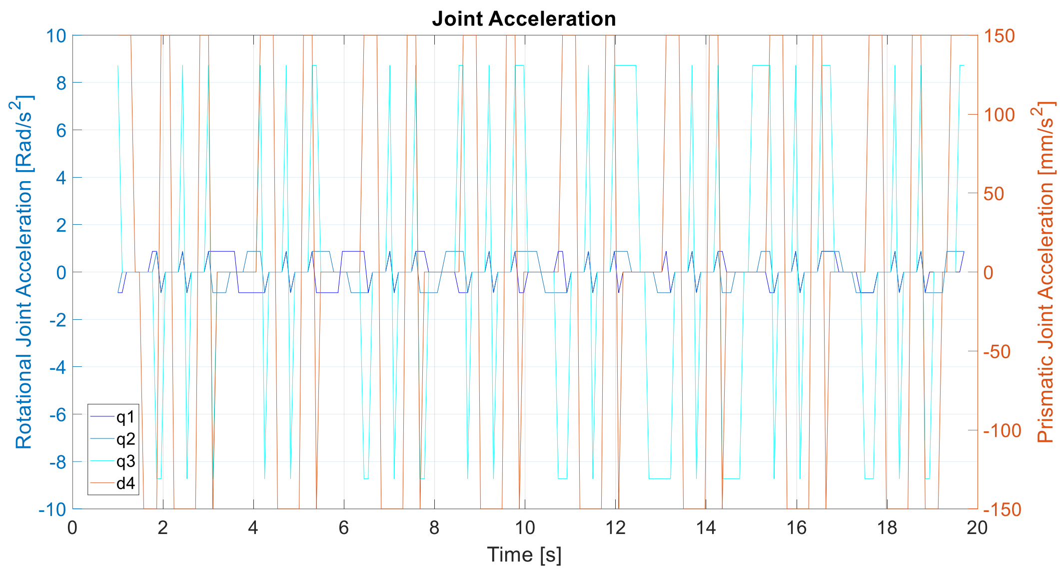
The joint position, velocity and acceleration are plotted below. To ensure robust control of acceleration, deceleration, and minimizing constant speed phase, I implemented a total time calibration code that adjusts the time spam of the trajectory according to the distance that it needs to travel.

```
%S=max(abs(qf-q0)); % calculate maximum rotational joint displacement
S=abs(qf-q0);
t0=0;
tf=max(2*sqrt(S./maxAcc));
```

Using this equation, we can ensure that the motors are working at peak efficiency to increase the overall speed. This can also be reflected in the velocity graph, where we can see that each trajectory has one motor that accelerates than decelerate immediately after reaching the middle point, meaning that no time is wasted, and the motor is working at highest acceleration all the time. At the same time frames, we can also see other motors that require less displacement moving slower and less efficiently to essentially wait for the ones that have the largest relative displacement. The difference in acceleration between each motor can also be observed as the slope of each line is different.







Since the acceleration is set differently in q3 and d4, their respective acceleration curves are drastically different. However, we can still see the maximum acceleration of each joint to be exactly as we determined.

4. End Effector Position and Orientation

The end effector position and orientation are shown in the animation in previous sections.

5. Verification

From the joint position and acceleration plot above, we can see that the acceleration never exceeds the designed limit and is always working at the highest efficiency. The position plot also shows that all joints are working within their range of motion.

6. Discussion

In the project, the implementation of a trajectory generation function for a robotic arm successfully demonstrated the capability to navigate between specified points, such as "reference," "chip1," or custom coordinates, with precise joint configurations. The adjustments made to the maximum acceleration for the rotational joint 3 and prismatic joint 4 significantly enhanced the robot arm's performance, making the motion smoother and more realistic while ensuring the system operates within its dynamic constraints. The trajectory animations provided clear visual feedback on the system's efficiency and the effective placement of markers indicated the target locations accurately. Through careful calibration of the total trajectory time, the project achieved an optimal balance between speed and accuracy, ensuring peak motor efficiency without sacrificing the accuracy of the end effector's position and orientation. The verification process confirmed that the acceleration limits were adhered to, validating the system's design for high-efficiency operation within safe operational parameters.