Trajectory Generation

Trajectory is the specification of robot configuration as a function of time. It is defined as the combination of a path and time scaling. Path is geometric description of a sequence of configurations achieved by the robot. Time scaling defines the time when the configurations are achieved.

Path $\theta(s)$ maps a parameter 's', where s has an initial value of 0 and final value of 1. The robot moves along the path, as 's' increases. Time scaling $s(t)$ assigns a value 's' to each time t ∈ $[0, T]$.

Together, a path and a time scaling define a trajectory $\theta\big(s(t)\big)$. The velocity and acceleration along the trajectory are defined as-

$$\dot{\theta} = \frac{d\theta}{ds}\dot{s}$$

$$\ddot{\theta} = \frac{d\theta}{ds}\ddot{s} + \frac{d^2\theta}{ds^2}\dot{s}^2$$

For the dynamics to well defined, each $\theta(s)$ and $s(t)$ must be twice differential.

Scripts are created in MATLAB to perform different task. The maximum speed of different joints on IRB 6620 are listed in table 1. Using the given values, feasible time duration between two consecutives via points is calculated for different types of trajectories. The study was done for point-to-point trajectory (final and initial velocities are zero). For the case of time scaling, Cubic, quintic and trapezoidal type were chosen.

Joint	Max Speed (deg/s)
J1	100
J2	90
J3	90
J4	150
J5	120
J6	190

Table 1. Joint velocity limits of IRB 6620

Cubic Time Scaling

In cubic order polynomial, the time scaling $s(t)$ is given by-

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

For initial conditions: $s(0) = 0\ ;\ s(T) = 1\ ;\ \dot{s}(0) = \dot{s}(T) = 0$

$$s(t) = \frac{3}{T^2}t^2 - \frac{2}{T^3}t^3$$

Quintic Time Scaling

In cubic order polynomial, the time scaling $s(t)$ is given by-

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

For initial conditions: $s(0) = 0\ ;\ s(T) = 1\ ;\ \dot{s}(0) = \dot{s}(T) = 0\ ;\ \ddot{s}(0) = \ddot{s}(T) = 0$

$$s(t) = \frac{10}{T^3}t^3 - \frac{15}{T^4}t^4 + \frac{6}{T^5}t^5$$

Trapezoidal Time Scaling

This time scaling is widely used in motion control. It consists a constant acceleration phase $\ddot{s} = a$ for a time period t_a. Largest velocity and acceleration in trapezoidal time scaling follow the following equation –

$$|(\theta_{end} - \theta_{start})v| = \dot{\theta}$$

$$|(\theta_{end} - \theta_{start})a| = \ddot{\theta}$$

For a three-stage trapezoidal profile, $\frac{v^2}{a} \leq 1$ must be satisfied. Then solving for 'T' we get-

$$T = \frac{a + v^2}{va}$$

Joint Space Trajectory

In joint space straight line path is given by –

$$\theta(s) = \theta_{start} + s(\theta_{end} - \theta_{start})$$

Straight line path from θ_{start} to θ_{end} offers simplicity. Also, the allowable joint configuration forms a convex set θ_{free} in joint space, so straight line between any two points in θ_{free} also lies in θ_{free}. The derivatives are given by –

$$\frac{d\theta}{ds} = (\theta_{end} - \theta_{start})$$

$$\frac{d^2\theta}{ds^2} = 0$$

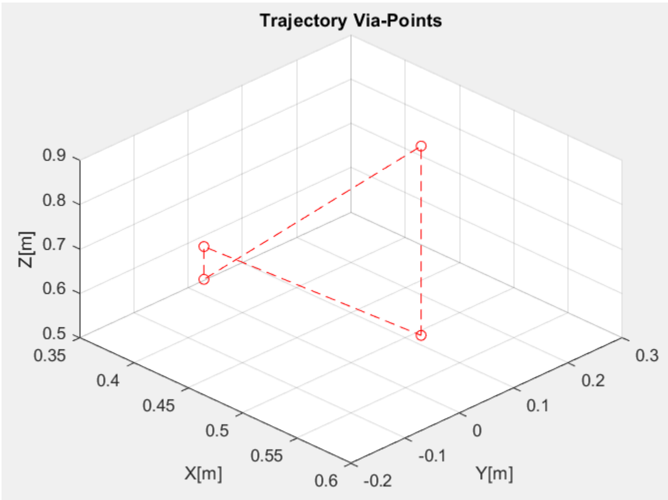

Fig.1 Via Points in the Trajectory

Moving on to our problem, taking the maximum speed values, time period for different time scaling is calculated which are then used for simulation. Via points passed by the robot are shown in the Fig.1. Upon calculation the following values are observed-

Cubic Scaling

$$\dot{\theta}_{max} = \frac{3(\theta_{end} - \theta_{start})}{2T}$$

$$T_{max} \geq 0.8612$$

Quintic Scaling

$$\dot{\theta}_{max} = \frac{15(\theta_{end} - \theta_{start})}{8T}$$

$$T_{max} \geq 1.0765$$

Trapezoidal Scaling

Using, $\frac{v^2}{a} \leq 1$ we get $T \leq 2/v$

After taking everything in consideration, time period(T) was chosen as 4 secs and acceleration period in trapezoidal trajectory was taken as 1 sec. The results for joint space trajectory with trapezoidal trajectory are presented in Fig.2 and Fig.3. All the rest of the scripts can be viewed at **Github**..

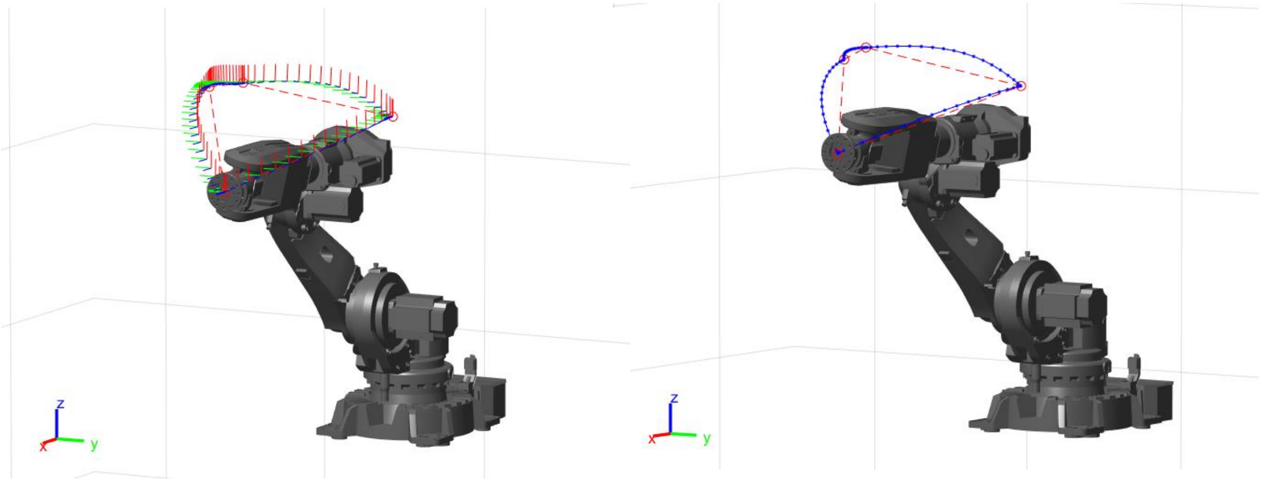

Fig.2 Trapezoidal Trajectory in Joint Space

In MATLAB scripts were created to simulated to different time scalings in joint space. Functions like trapveltraj(), cubicpolytraj(),quinticpolytraj() were used for trapezoidal, cubic and quintic trajectories respectively.

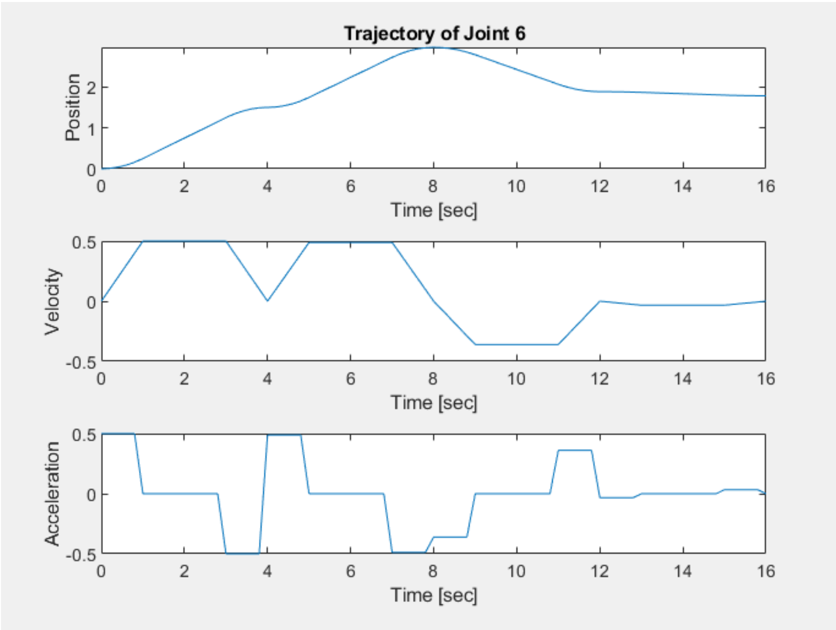

Fig.3 State of Joint 6 during motion

Task Space Trajectory

Straight line motion in joint space, doesn't yield straight line motion of the end effector in task space. To obtain task space straight line motion, two methods can be used-

1. Screw Motion from X_{start} and X_{end}

   This involves both rotation and translation of end effector about a fixed screw axis. This method provides a straight line for its constant screw axis. The path equation is as follows-

$$X(s) = X_{start}e^{(\log(X^{-1}{}_{start}X_{end})s)}$$

2. Decouple Rotation and Translation
   The first method doesn't yield straight line motion in cartesian space. To achieve that both translational and rotational motion are decoupled. The following equations are followed during the motion-

$$p(s) = p_{start} + s(p_{end} - p_{start})$$

$$R(s) = R_{start}e^{(\log(R^T{}_{start}R_{end})s)}$$

   Thus, end effector frame follows a straight line while having a constant axis of rotation.

Moving on to our problem, taking the maximum speed values, time period for different time scaling is calculated which are then used for simulation. Joint space and task space trajectories are compared in Fig.4. Upon calculation the following results are obtained-

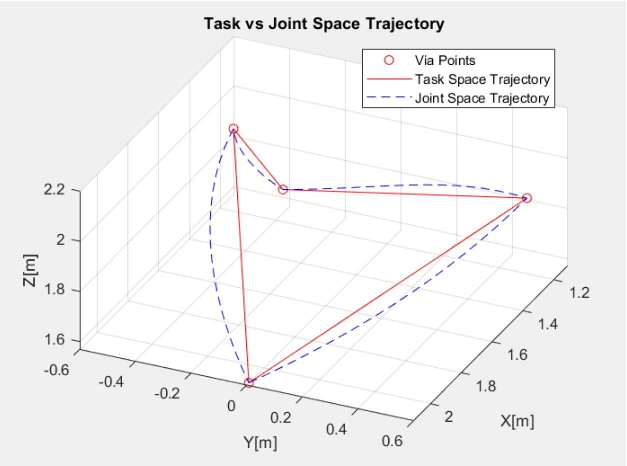

Fig.4 Joint vs Task Space Trapezoidal Trajectory

Cubic Scaling

$$\dot{p}_{max} = \frac{3(p_{end} - p_{start})}{2T}$$

$$T_{max} \geq 0.307$$

Quintic Scaling

$$\dot{p}_{max} = \frac{15(p_{end} - p_{start})}{8T}$$

$$T_{max} \geq 0.383$$

Trapezoidal Scaling

$$\text{Using, } \frac{v^2}{a} \leq 1 \text{ we get } T \leq 2/v$$

After taking everything in consideration, time period(T) was chosen as 4 secs and acceleration period in trapezoidal trajectory was taken as 1 sec. The results for task space trajectory with cubic trajectory are presented in Fig.5 and Fig.6. All the rest of the scripts can be viewed at **Github**.

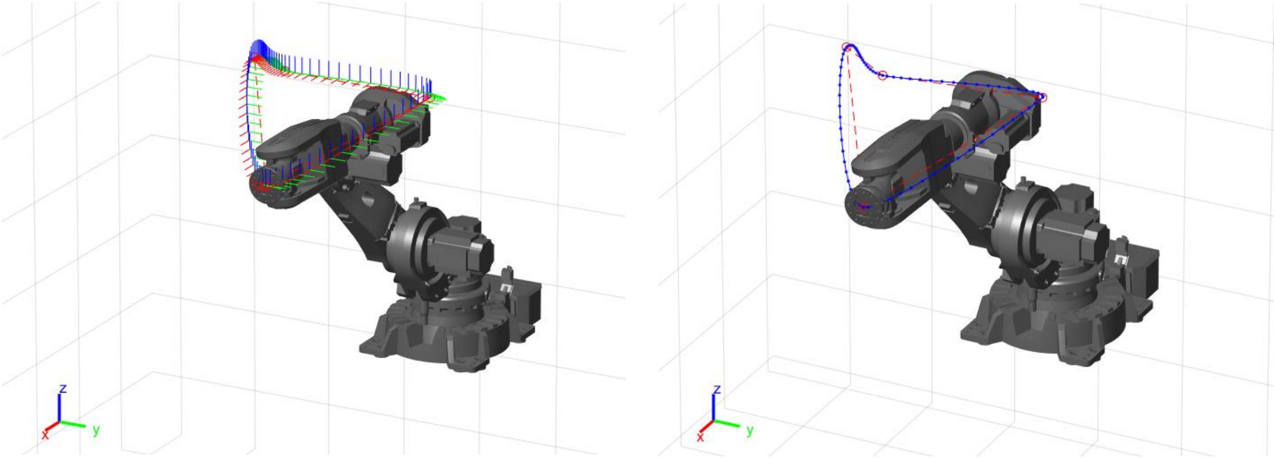

Fig.5 Cubic Trajectory in Task Space

In MATLAB scripts were created to simulated to different time scalings in task space. Functions like trapveltraj(), cubicpolytraj(),quinticpolytraj() were used for trapezoidal, cubic and quintic trajectories respectively.

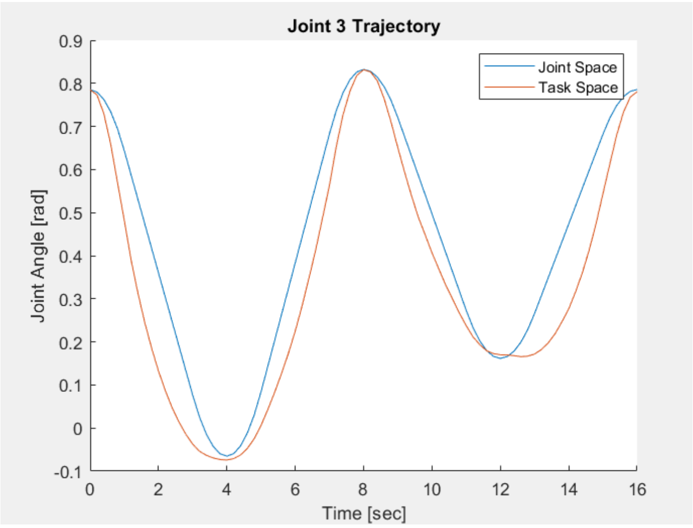

Fig.6 Comparison of Joint angle of 3rd Joint

Conclusion

In the above project, Joint space and Task space trajectories were compared in MATLAB. Comparing the two cases, each one has its own pros and cons. Joint space planning method executes faster since Inverse Kinematics is only calculated at the via-points. Also, the actuator motion is smooth and easier to validate. But in joint space, intermediate points are not guaranteed to be collision free or respect joint limits.

In task space planning, Motion is predictable and offers better handling of obstacles and collisions. But the method is slower since Inverse Kinematics is calculated at every time step. Also, Actuator motion is not necessarily smooth.