DoNRS: Homework #4

Due on 17 October

Alexandr Klimchik

Albatoul Yaseen

In this task, forward and inverse kinematics equations were derived for RRR robot and Jacobian matrix using geometrical method.

1 Task1

```
Jg =

[ -sin(q1)*(cos(q2 + q3) + cos(q2)), -cos(q1)*(sin(q2 + q3) + sin(q2)), -sin(q2 + q3)*cos(q1)]
[  cos(q1)*(cos(q2 + q3) + cos(q2)), -sin(q1)*(sin(q2 + q3) + sin(q2)), -sin(q2 + q3)*sin(q1)]
[                                 0,          - cos(q2 + q3) - cos(q2),         -cos(q2 + q3)]
[                                 0,                          -sin(q1),              -sin(q1)]
[                                 0,                           cos(q1),               cos(q1)]
[                                 1,                                 0,                     0]
```

2 Task2

In this task we should build the joint trajectory (polynomial) between two points in joint space, q(0)=(0,0,0) and q(2)=(2,3,4) with null initial and final velocities and accelerations.
Since we have 6 constrains, our polynomial should be of degree 5. for each joint we compute the coefficients of the polynomial trajectory using the joint constrains.

```
A =

[ 1, t0, t0^2,   t0^3,    t0^4,    t0^5]
[ 0,  1, 2*t0, 3*t0^2,  4*t0^3,  5*t0^4]
[ 0,  0,    2,   6*t0, 12*t0^2, 20*t0^3]
[ 1, tf, tf^2,   tf^3,    tf^4,    tf^5]
[ 0,  1, 2*tf, 3*tf^2,  4*tf^3,  5*tf^4]
[ 0,  0,    2,   6*tf, 12*tf^2, 20*tf^3]
c = [q0;v0;acc0;qf;vf;accf];
b = inv(A)*c;
```

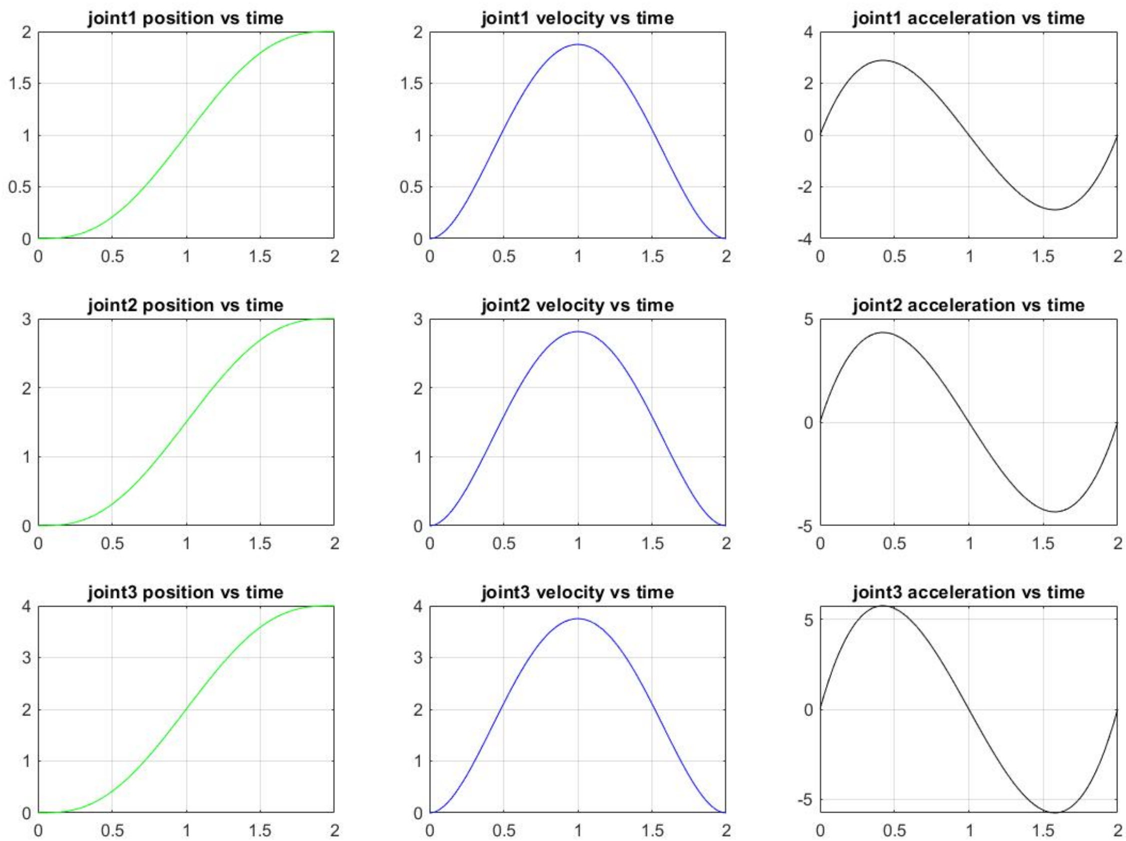

Figure 1: joints position, velocity and acceleration

3 Task3

In this task we have to compute joint trajectory(trapezoidal) between two points following PTP(point to point). q1=(0,0,0) to q2=(2,3,4) with the following constrains:
o Controller command interpretation frequency – f = 100 Hz (dt=0.01)
o Maximum joint velocity – 1 rad/s
o Maximum joint acceleration – 10 rad/s^2
we start by computing ta and tf for each joint using the given constrains by taking into account the controller constraint and then make sure that all joints start moving at the same time (same ta) and finish at the same time (same tf). then we compute the new maximum velocity and acceleration for each joint.
To make sure the velocity profile is trapezoidal, we divide our path into three parts :

$$q(t) = \begin{cases} a_{10} + a_{11}t + a_{12}t^2 & , t \leq t_a \\ a_{20} + a_{21}t & , t_a < t \leq t_f - t_a \\ a_{30} + a_{31}t + a_{32}t^2 & , t_f - t_a < t \leq t_f \end{cases}$$

and then we differentiate to get the velocity and acceleration function. The coefficients can be computed for each joint using the initial constrains.

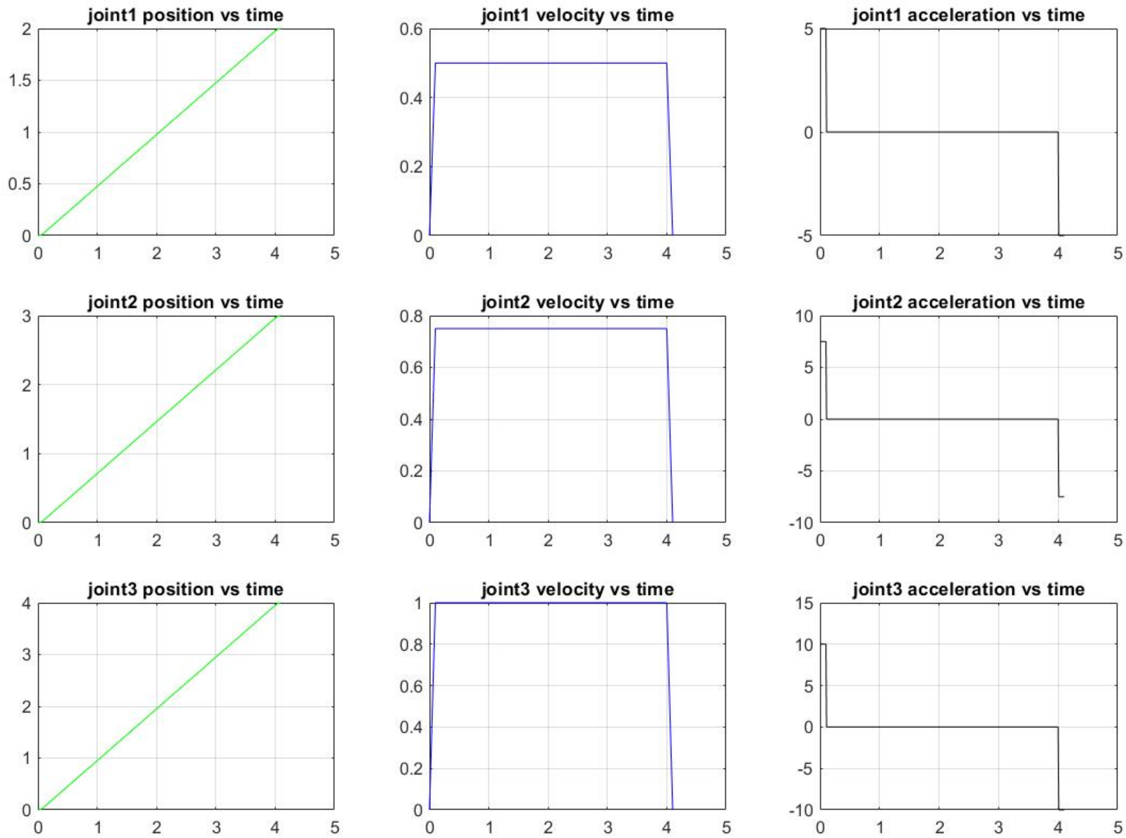

Figure 2: joints position, velocity and acceleration

we see that our joints start moving and finish at the same time while still achieving the desired final joint position.

4 Task4

In this task, we have two points, $p1 = (1, 0, 1)$ and $p2 = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 1.2)$ in the task space and we were asked to find the straight line path between these two points taking into account the following constrains :
o Controller command interpretation frequency – f = 100 Hz
o Maximum linear velocity – 1 m/s
o Maximum linear acceleration – 10 $m/s2$
First of all, we need to find the inverse kinematics for this task.
we start by finding a straight line between our two points then we compute ta and tf taking into account the controller constrain. then we find the new velocity and acceleration. then we loop on our points in task space to find postions and velocities in joint space using the IK and Jacobian.
after that, for each two consecutive points in joint space we build a path.
Finally we apply FK on the computed joint positions and compare it with the desired path.

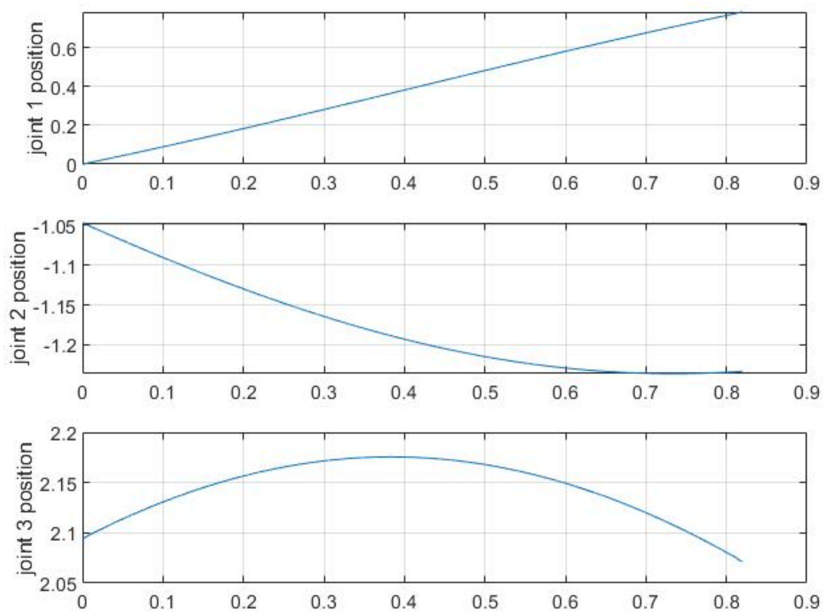

Figure 3:

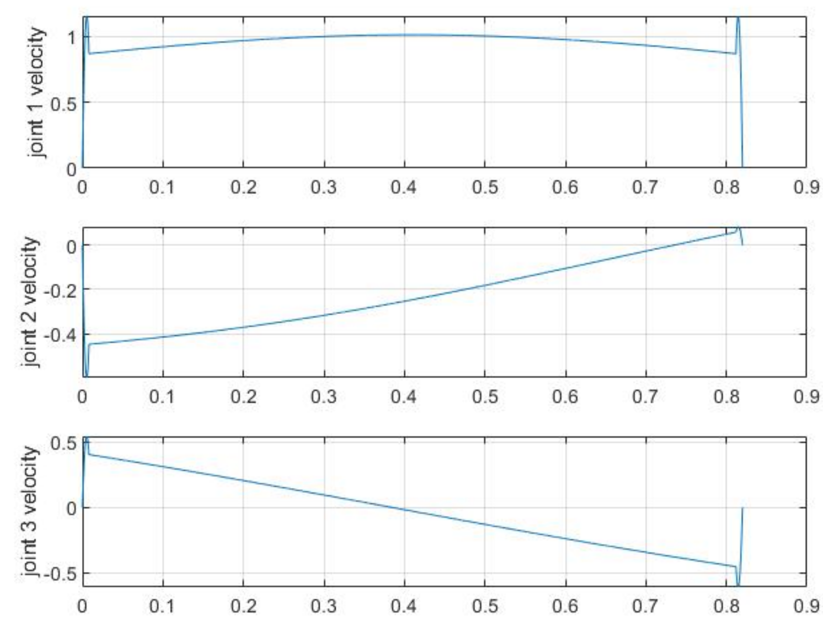

Figure 4:

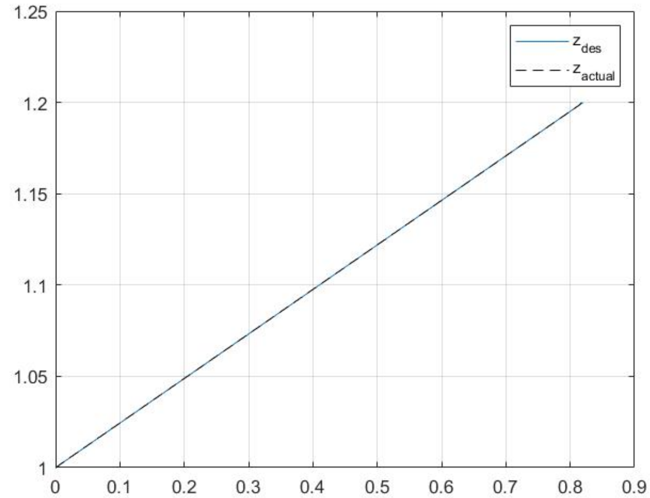

Figure 5:

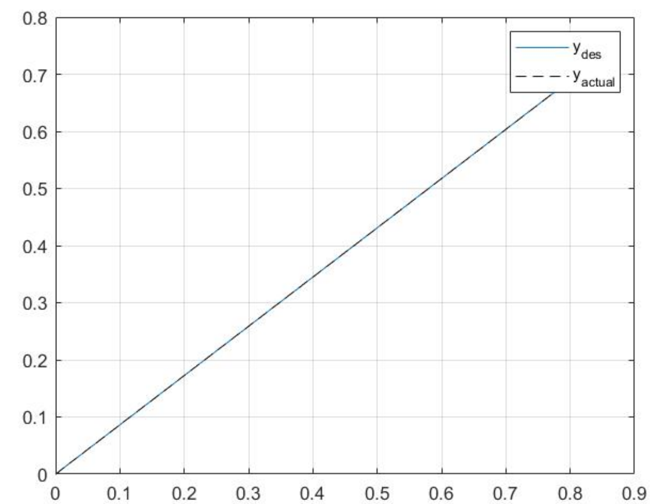

Figure 6:

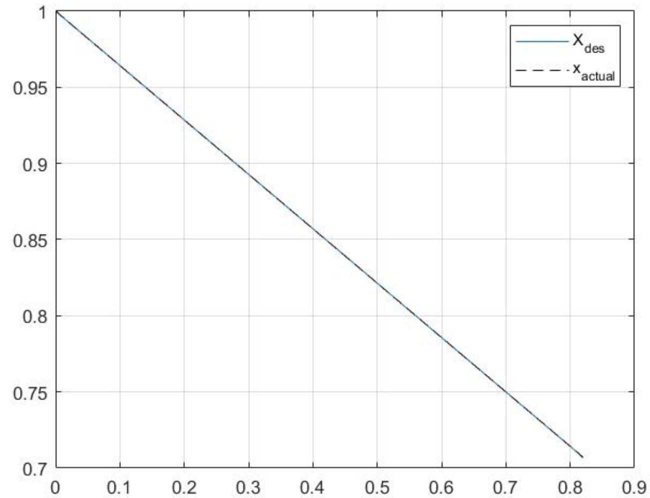

Figure 7:

5 GitHub

Link to assignment : HW4