Jose Corona

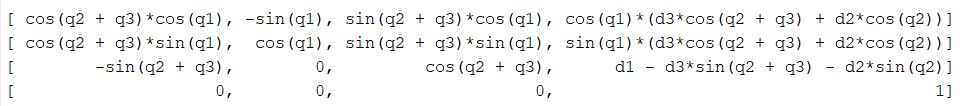
Home Task 4

To run the file with name “Program.m “ in MATLAB.

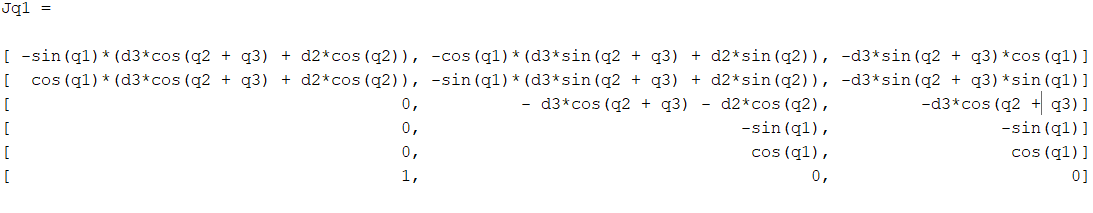
1. Calculate Jacobian (numeric method)





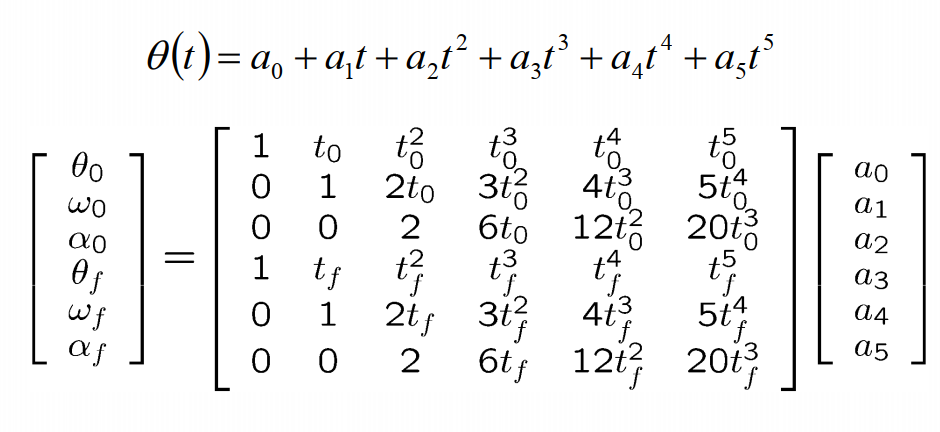


I used the numeric method to get the jacobian.



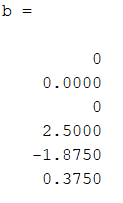
2. Joint trajectory q(t) from q(0) = (0, 0, 0) to q(2) = (2, 3, 4) with null initial and final velocities and accelerations. (polynomial)

A polynomial solution for each joint is solved as like in presentation, since it has 6 constrains, is needed a polynomial of fifth solution.



For the joint 1:

So the coefficients for the polynomial are:

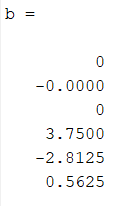


q1= (3\*t^5)/8 - (15\*t^4)/8 + (5\*t^3)/2 + t/562949953421312





For the joint 2:

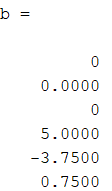


q2= (9\*t^5)/16 - (45\*t^4)/16 + (15\*t^3)/4 - t/562949953421312



For the joint 3:



q3= (3\*t^5)/4 - (15\*t^4)/4 + 5\*t^3 + t/281474976710656

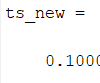
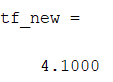




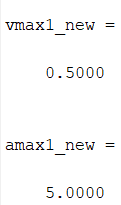
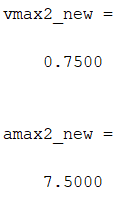
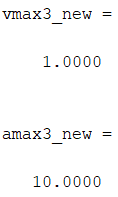
3. Joint trajectory for the following commands: PTP – q1 = (0, 0, 0) to q2 = (2, 3, 4) (trapezoidal)

Fist we have to calculate the times “ts” and “tf” for all trapezoidal velocity for each joints, taking into account the controller command interpretation frequency of 10hz.

So, the values of “ts” and “tf” for all the joins is :

And the new values of max velocity and acceleration for each joint are:

And the plot of the position, velocity and aceleration are:

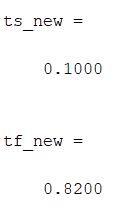






4. Joint trajectory for the following commands: LIN – p1 = (1, 0, 1) to p2 = (√2/2, √2/2, 1.2) (trapezoidal)

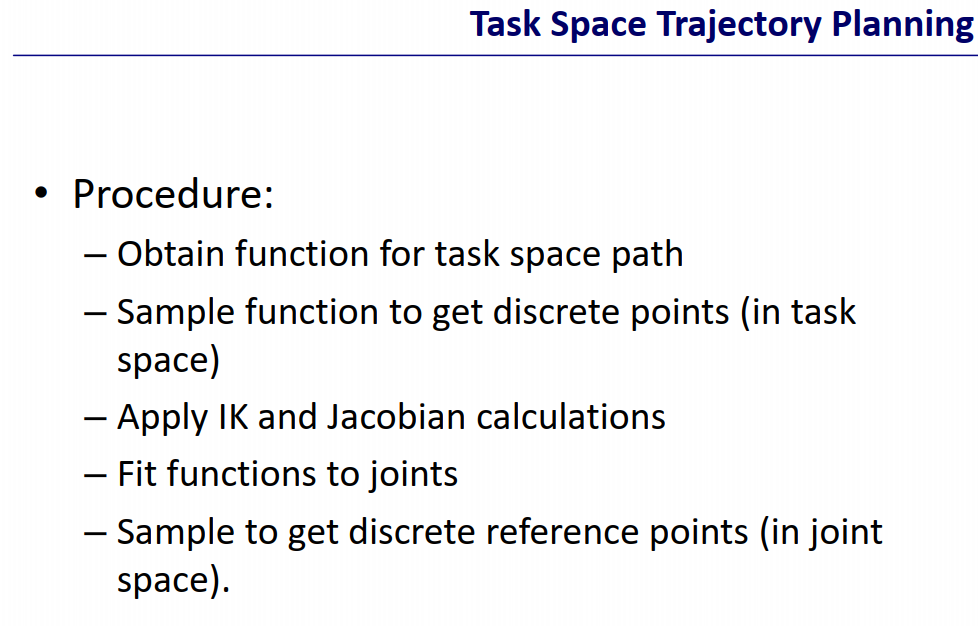
4a. Trapezoidal trajectory in task space (without constrains in joints space)







To get the joints position, velocity and acceleration, since que have the formulas for velocity in x,y,z for each section of the trajectory, we have to multiply it with the inverse of the Jacobean, to get the function of velocity in the joints. Then we could try derivate it to get the function of acceleration in the joints.



4b. Trapezoidal trajectory in task space and constrains in joints space too.

Fist we have to calculate the joints position for p1 and p2, so we use the inverse kinematics. I develop a function for IK (file named “IK.m”).

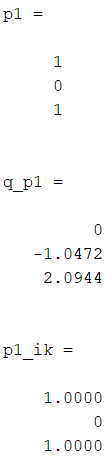
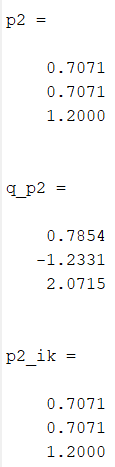




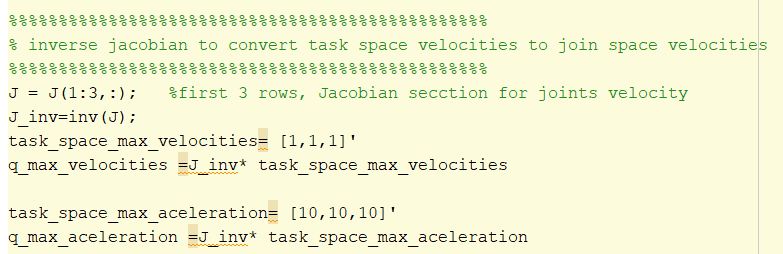




So, for point p1 and p2, I get the joins position and checked they were correct, with my forward kinematics:

Then,



Link Github:

https://github.com/Jose-R-Corona/HomeTaks4