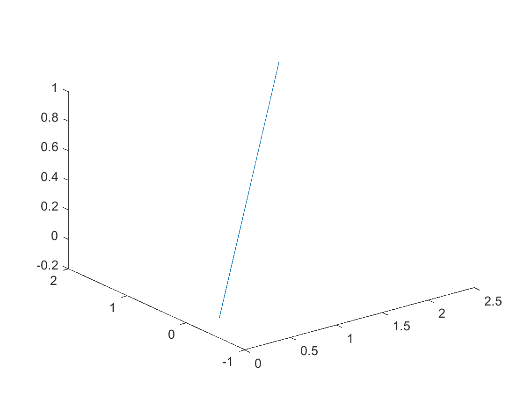
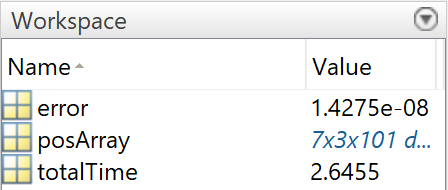
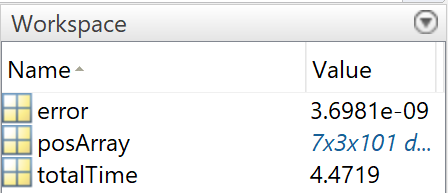
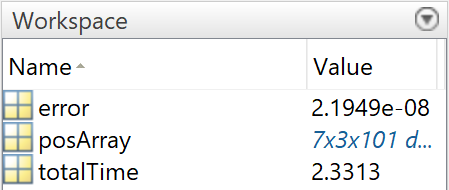
**ProblemSet5Report\_Fanuc\_JiaweiGe**

**The Line Trajectory**

I change the given line in the ik\_lab file because it’s partially out of the robot workspace. The line I chose is from the point (0.3, -0.1, -0.2) to the point (2.1, 1.7, 0.9). The plot is shown below. The root mean square error and the computation time are reported as shown in the screenshots. The first and second results are under BFGS algorithm and the third is under LM algorithm. The first result is acquired when the initial guess value is given by home configuration((0, 0, 0, 0, 0, 0)) and the second result is when I change it to the joint values that can reach the start point((-0.3218, -0.2821, -1.0148, -1.8629e-08, 1.2969, -0.3218)). According to the results, setting the initial guess to the last joint configuration makes good for calculation time while increase the error a little bit. What’s more, the LM algorithm is better than the default BFGS algorithm, as the error decrease apparently, but the time increase a lot.

When I maintain the 3 rotational weights and increase the 3 translational weights to 1000, errors drop to e-10 and time increase to around 4.9772 s (in LM algorithm). When I maintain the 3 translational weights and increase the 3 rotational weights to 1000, errors drop to 2.0512e-9 and time increase to around 2.3059s (in LM algorithm). According to the results, the translational weights will seriously impact the error while the rotational ones only slightly influence on it. They will both lead to an increase of calculation time.



**My Own Trajectory**

My own trajectory is a spiral line. The radius and height are set to be 2m and 1m respectively. In order to restrict the robot within workspace, I chose the (startRadian = 0.2\*pi), (endRadian = 1.8\*pi). The end effector will traverse my own spiral line trajectory well as shown in the second picture.

