

The code folder contains all the common functions and the 3 individual scripts for the 3 tasks. For the best tuned, the file name is best_tuned.m, for overshooting it is overshoot.m and for new block configs it is new_tuned.m.

The block location for the new case is

Start : (1.5,0.5,0)

End : (0.2,-1.2,(-pi/2))

Difference Between Overshoot and Best :

Refer to Figure1 showing the error plot for best case, while figure2 for overshoot. As can be seen there is a small overshoot in the second figure during the start which is absent from the first plot. The error settles down. This jerk is also observed in the video file uploaded

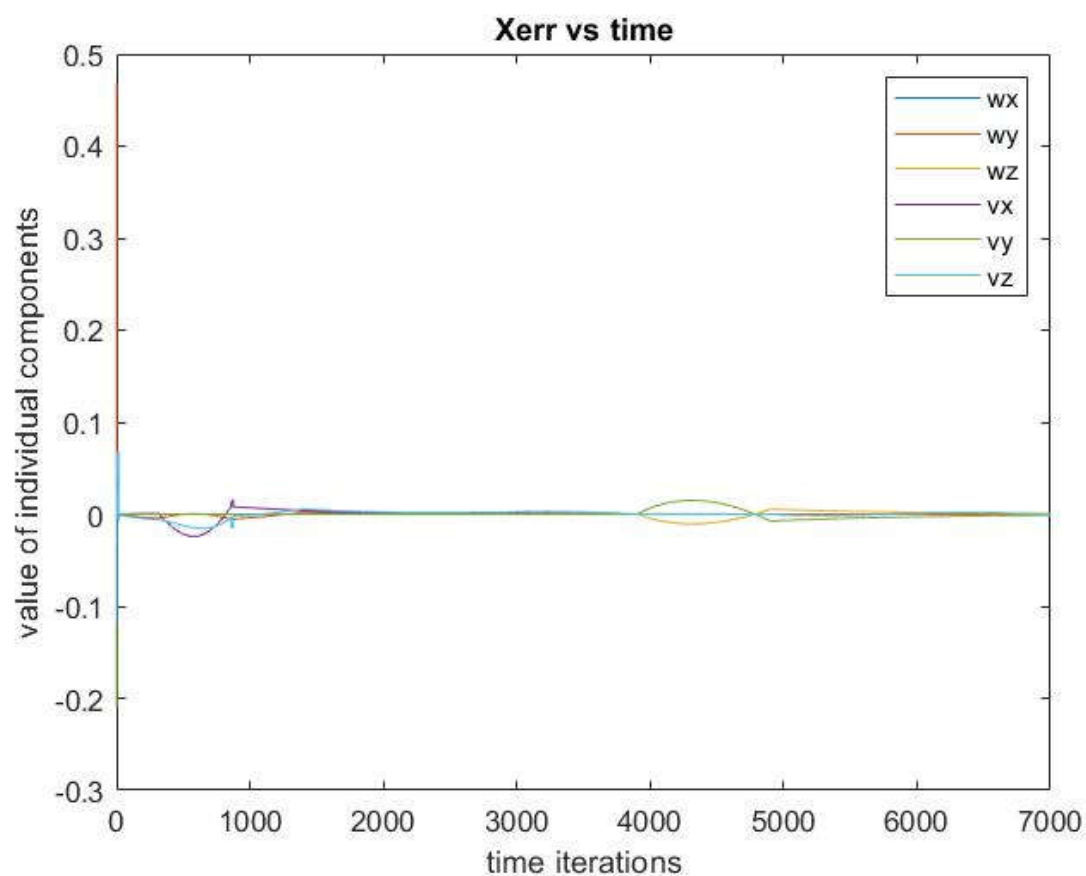


Figure 1

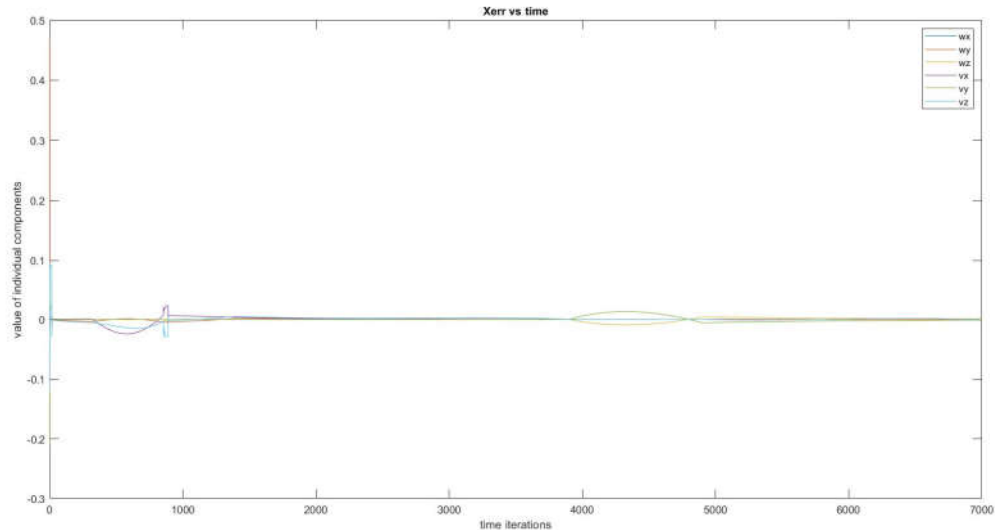


Figure 2

Controller tuning for Best case:

$K_p = 1.1$ (across all velocities)

$K_i = 0.01$ (across all velocities)

Controller tuning for overshoot :

$K_p = 1.3$ (across all velocities)

$K_i = 0.01$ (across all velocities)

Implementation of Joint limits:

I think implementing joint limits not only helps in preventing intersection, but also helps move the arm joints in a 'non' shaky way by preventing the arm reaching singularity and/or avoiding unnecessary movements in a non-useful direction. In the start positions of the particular case as done here, there was no self-intersection as such. Hence the joint limits put, are only to prevent jitters and singularities. Thus they are local to the positions I chose and may not standardize over all cases. The joint limits I have specified are as follows:

Joint 2 of the robot arm is restricted between 0 and -3 radians

Joint 3 of the robot arm is restricted to below 0 radians

Joint 4 of the robot is restricted to be below 0.5 radians

In a folder labelled joint_limits_videos, two videos showing a before and after are shown.

The way I have implemented joint limits is by following the procedure on the project page. A function called testjointlimits is created, which is fed the speed values from the inverse of Jacobian. These values are then used to

Troubles / doubts: Either because of a programming error or because of the nature of the problem, I am still unsure of the exact procedure of tuning, as there are so many variables which are playing a

part. If I change the 'time periods', the entire tuning parameter changes. Similarly adding joint limits, giving tolerance to the pseudo-inverse, changing starting configurations, all lead to changes. I had hoped that there were certain general tuned parameters which would hold true for a larger variety of cases.

In case the code doesn't run please use the folder 'supported files and code' which has all the supported functions.