

README

The Python software has been used for this code.

The code is executed in a way bit different than the method stated. The method used three functions : NextState(), TrajectoryGenerator() and FeedbackControl(). These needed to be called during the program to allow looping over them and create csv files as output. Instead, there is just one function used here : TrajectoryGenerator() which has been called to create the csv file of output that contains the reference trajectory transformation matrices. The other two functions were not used and they are directly looped in to create the csv output as required. Hence, only the structure of the code would seem little different but the output remains the same.

Also, when the code is executed and the csv files are generated completely (this takes some time due to the number of lines to be written in the csv files), the first row and first column of the csv file 'configuration' needs to be deleted manually. After deleting both these things and saving the csv file, it can be executed by the CoppeliaSim simulator. This is because the simulator only understands numerical data and not the column names and row indices of the matrix (or DataFrame) and so it is required to delete those things manually before simulating that csv file.

The results show that using a feedforward-plus-P controller does not create any overshoot and it results in smooth motion. This controller was used for the 'best' motion. The feedforward-plus-PI controller created overshoot and the error slowly got reduced to zero after some time. This controller was used for the 'overshoot' motion.