Mobile Manipulation - Capstone Project for ME449 Robotic Manipulation

Overview:

The task of the Project is to write a program that enables a Kuka youBot to pick and place a cube from one position to another automatically. The youBot is a mobile manipulator with 4 Mecanum Wheels and has a 5 DoF robotic arm. The Program typically consists of three parts:

1) Trajectory Generation : The function TrajectoryGenerator will generate a reference trajectory for the end effector of Kuka youBot to pick and place a cube from one place to another. There are a total 8 segments for this trajectory.

2) Feedforward Plus PI Feedback Control: Function FeedbackControl is used to calculate the kinematic task-space feedforward plus feedback control law.

3) Finding the Next Configuration of the Robot: Function Nexstate is used to calculate configuration of the robot time_step later.

All the parts need to be put together to generate a trajectory that accomplishes the task of successfully picking up the block from one place to another

Instructions on how to use the program:

1) pip install modern_robotics
2) run Agresar_Bhagyesh_Final_Project.py
3) The program asks for input for Kp and Ki

Flow of the Program:

For every step of the reference trajectory:

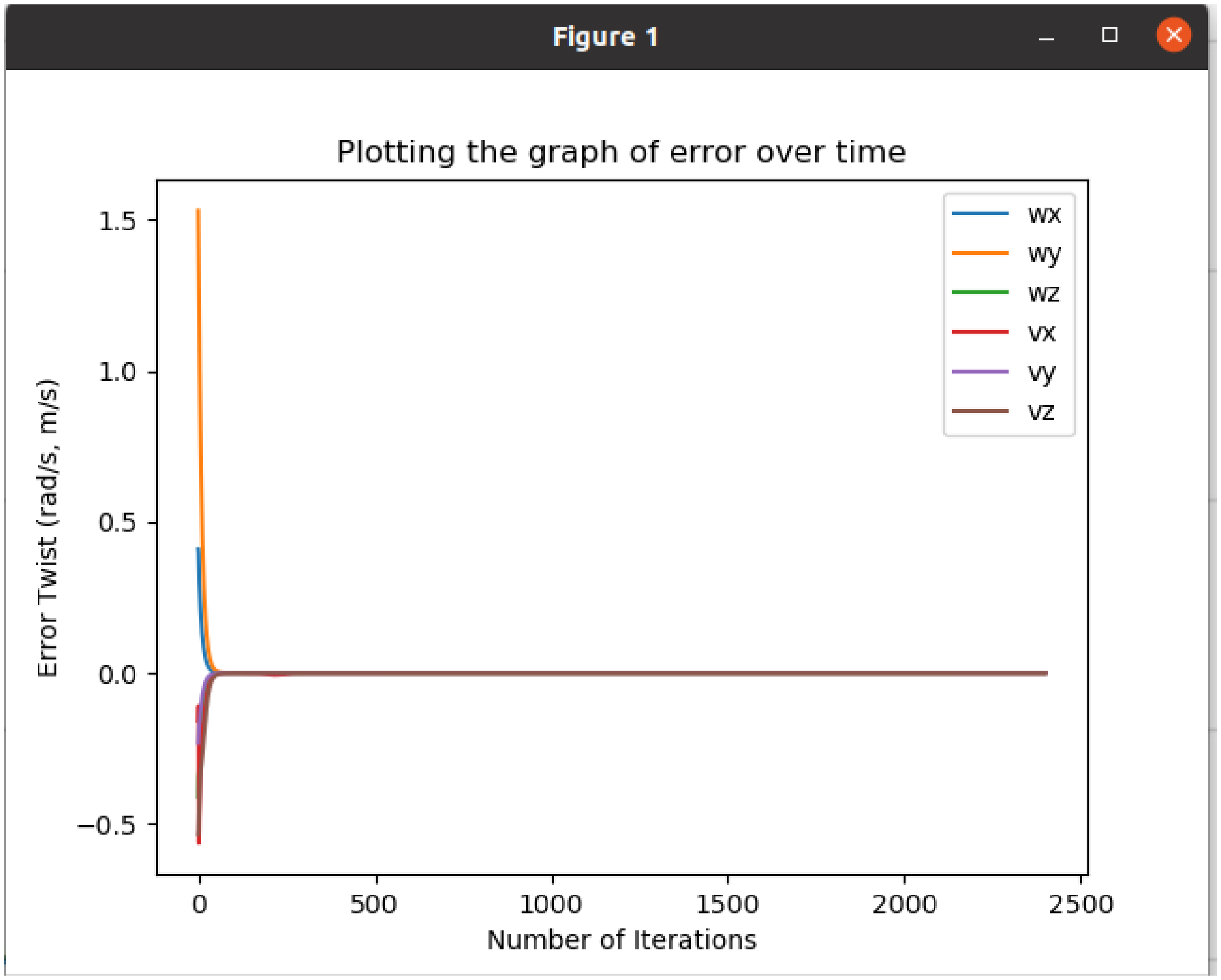
1) The current and desired end effector configurations are calculated and FeedbackControl function is used to find the commanded End-Effector Twist.
2) Since youBot is a mobile manipulator, the Jacobian of the base of the robot and arm are calculated and then the total Jacobian of the robot is calculated.
3) PseudoInverse of this Jacobian is used to get the wheel and arm velocities from the twist which is calculated in the previous step.
4) The current_configuration of the robot, the velocities of the arms and the wheels are then used by the NextState Function to calculate the configuration of the robot after time step later.

Demonstration of Different Results of the program:

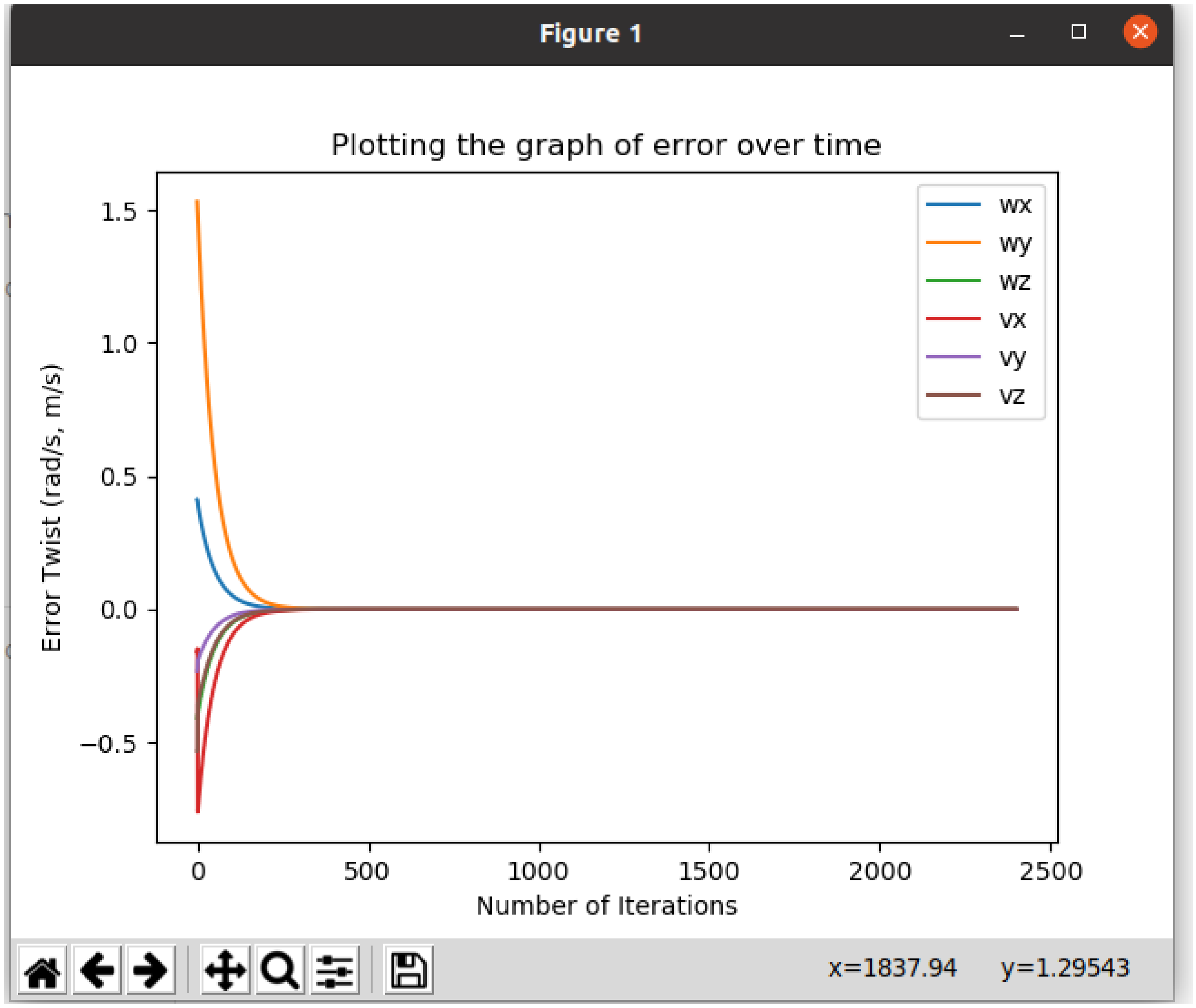
To test the controller, the chassis was given a 30 degree orientation error and 0.2 m positional error in x and y direction.

The results of the program are divided into 3 parts:

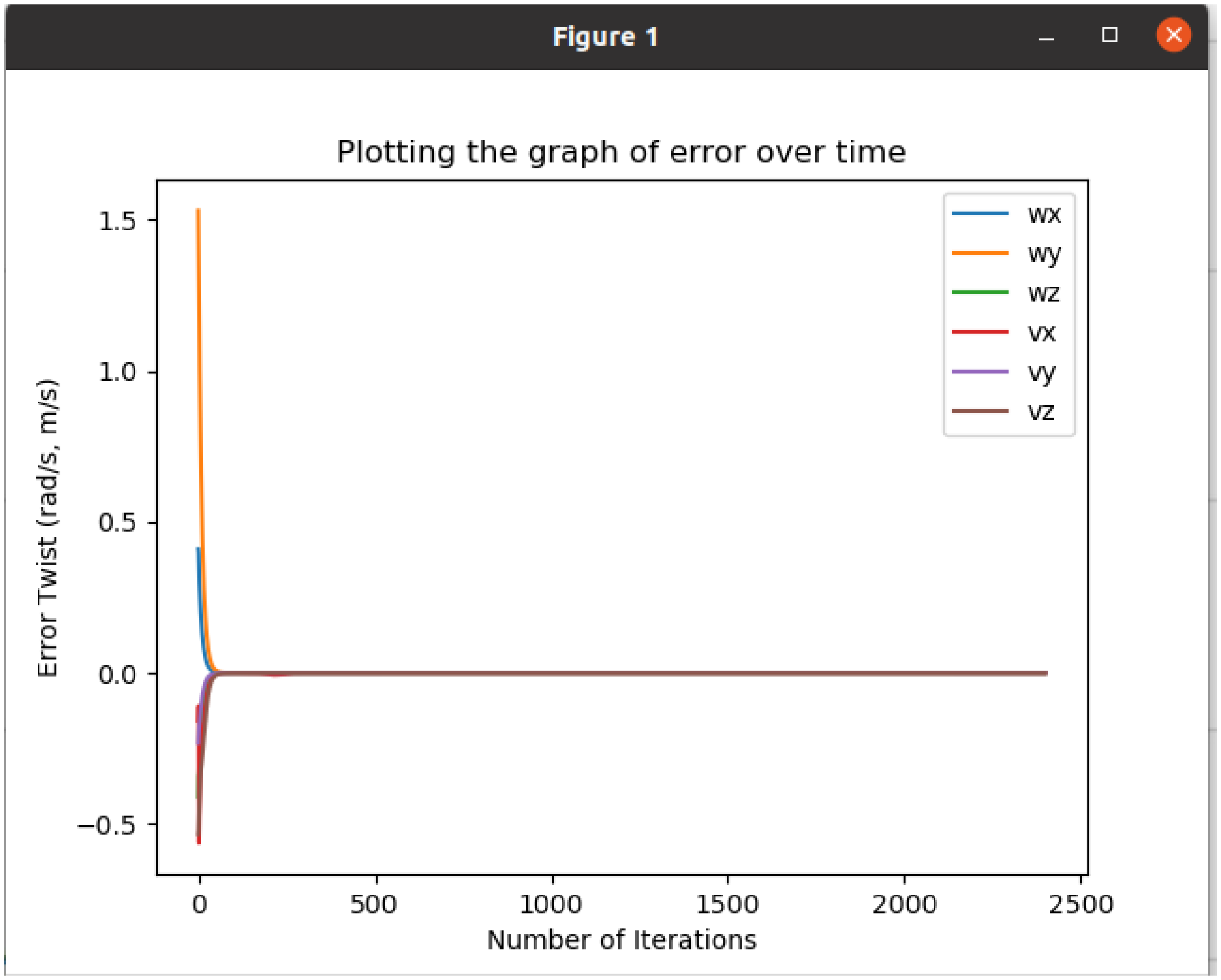
a) Best Run: In this run a feedforward plus PI controller is used. The proportional and integral gains are set to : kp = 10, ki = 0.5



b) Overshoot: Due to some numerical errors in the code, a perfect overshoot condition couldn't be achieved. Here kp =2, ki = 0.89



c) New_Task : In this run, the cube's initial and final configurations are changed to test the robot controller. In this case, due to some numerical errors, the robot doesn't perfectly pick up the cube. In this case the kp = 10 and ki = 0.5.



Conclusion:

The robot corrects its position initially, to correct the error, after that the error twist converges to zero as can be seen in the figures above.