Aalto University
School of Electrical Engineering

ELEC-E8126

Robotic Manipulation

Report: Exercise -4

Author:
Aravind Swaminathan

Student Number:
769383

February 27, 2020

1 The mathematical equation of torques computation:

1. The mathematical equation for computation of torque for feedforward controller is given by

$$\tau = J^T(\theta)F_d$$

where $J^T(\theta)$ is the Jacobian matrix and F_d is the Desired force of end effector and τ is the torque computed for end effector
In the above equation, the gravity term $g(\theta)$ is modelled in simulator as mentioned in the question, so didnt use it for computation

2. The mathematical equation for computation of torque for feedforward controller is given by

$$\tau = J^T(\theta)(F_d + K_pF_e + K_i \int F_e(t)dt$$

where $J^T(\theta)$ is the Jacobian matrix and F_d is the Desired force of end effector , F_e is Error in force(i.e desired- acual force), K_p and K_i are tuning constants for the controller and τ is the torque computed for end effector
In the above equation, the gravity term $g(\theta)$ is modelled in simulator as mentioned in the question, so didnt use it for computation

2 Plots of position/force profiles of the controllers for ideal/real robot:

2.1 Ideal Robot:

2.1.1 FeedForward controller results:

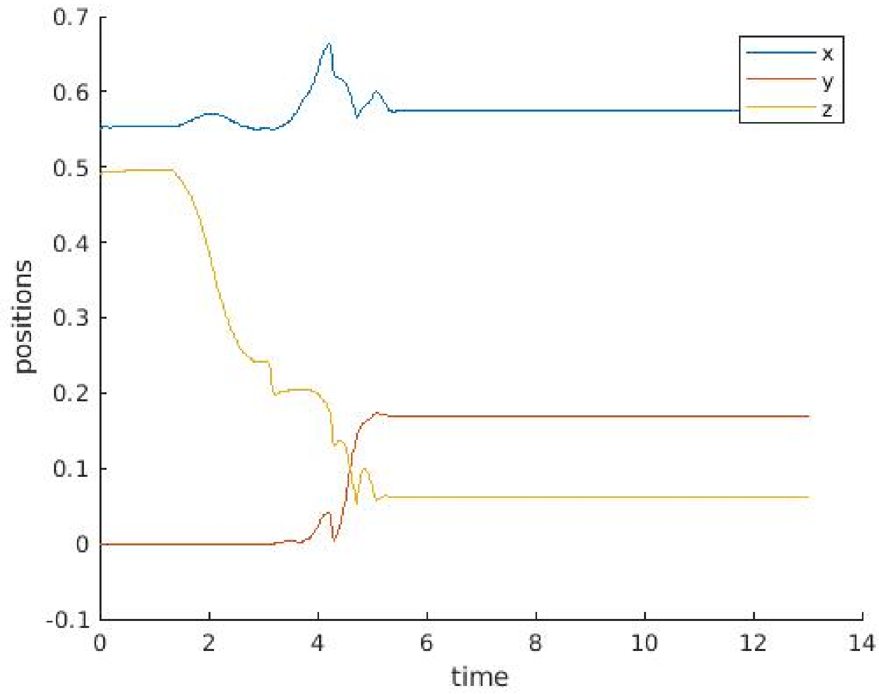

Figure 1: Ideal Robot with Feedforward controller position graph

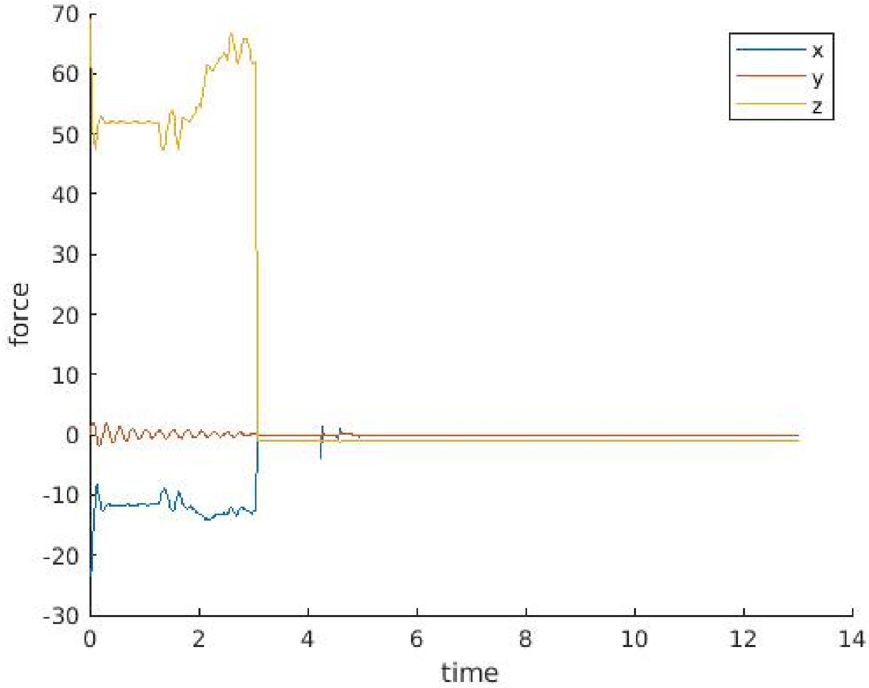

Figure 2: Ideal Robot with Feedforward controller force graph of end effector

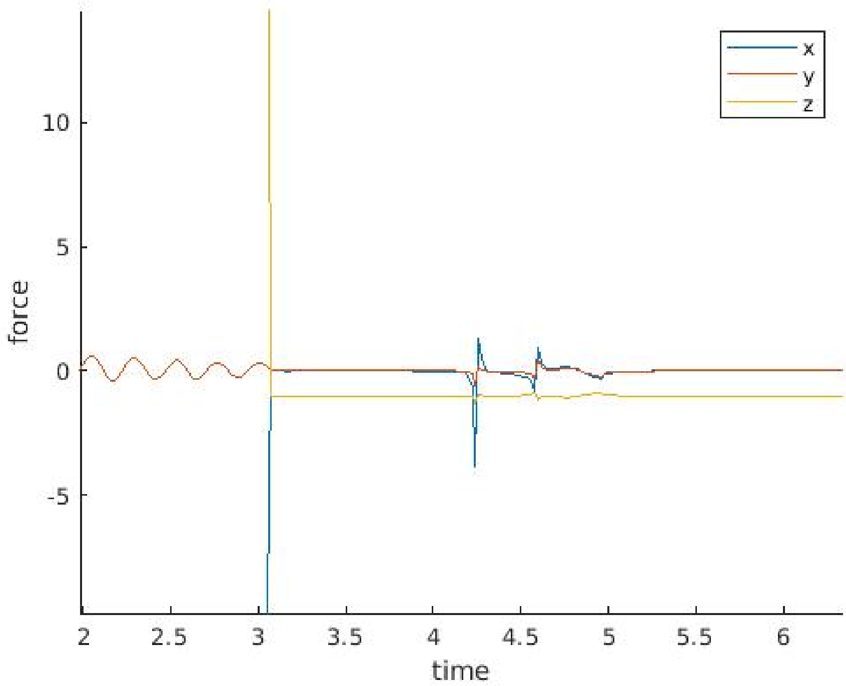

Figure 3: Ideal Robot with Feedforward controller force graph of end effector Zoomed for Steady state value

2.1.2 feedforward plus feedback-based (PI-controller) results:

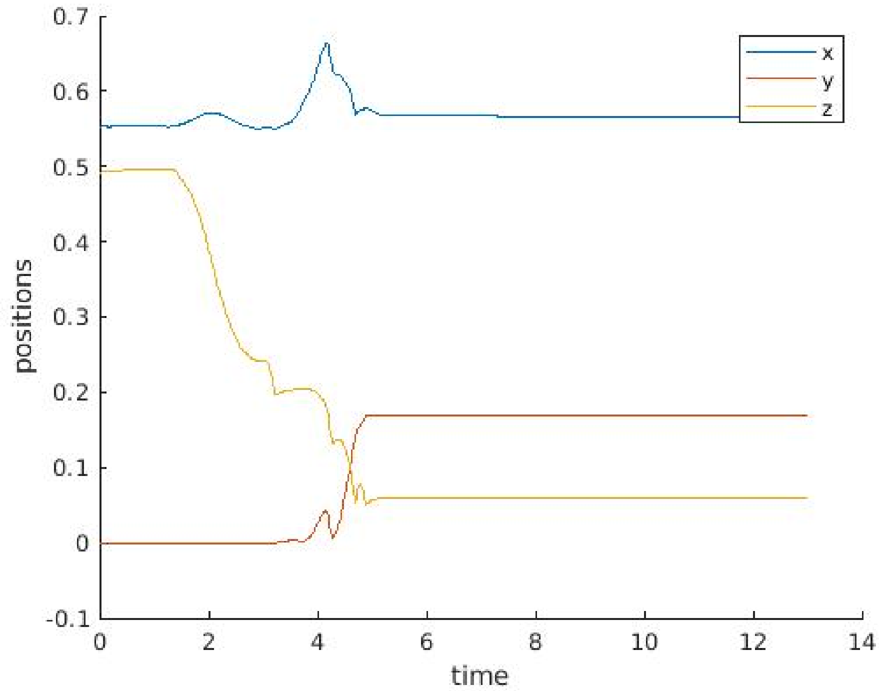

Figure 4: Ideal Robot with feedforward plus feedback PI controller position graph

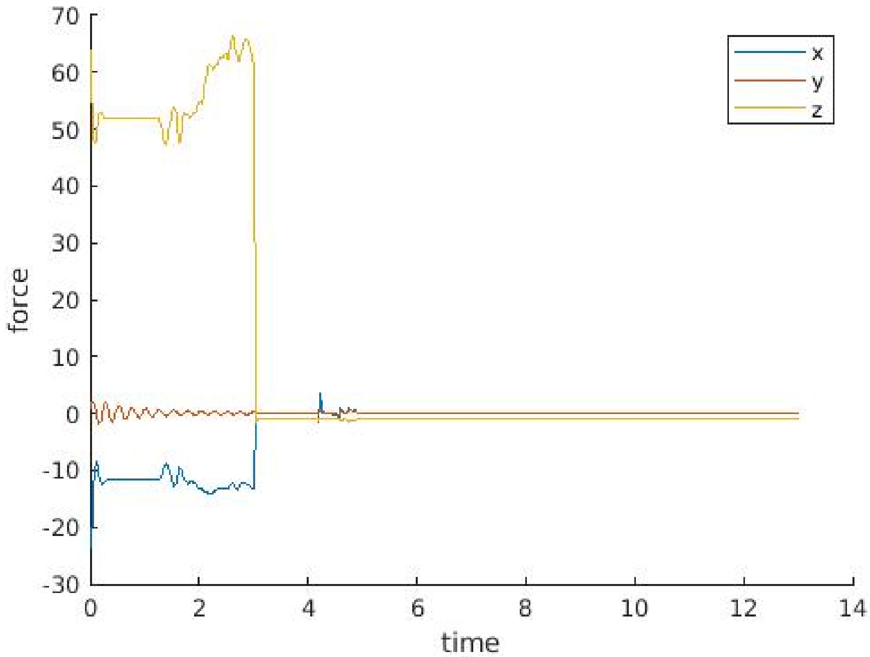

Figure 5: Ideal Robot with feedforward plus feedback PI force graph of end effector

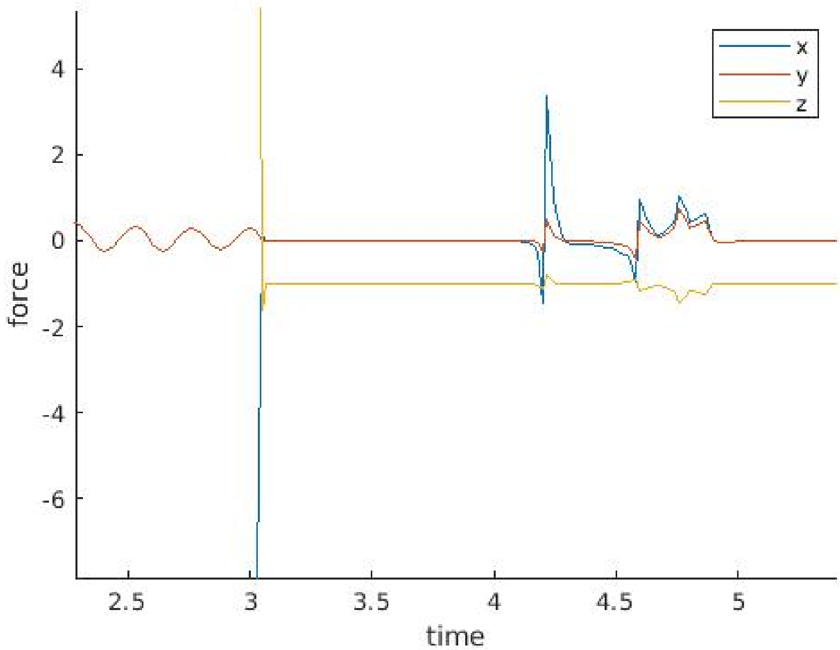

Figure 6: Ideal Robot feedforward plus feedback PI force graph of end effector Zoomed for Steady state value

2.2 Real Robot:

2.2.1 FeedForward controller results:

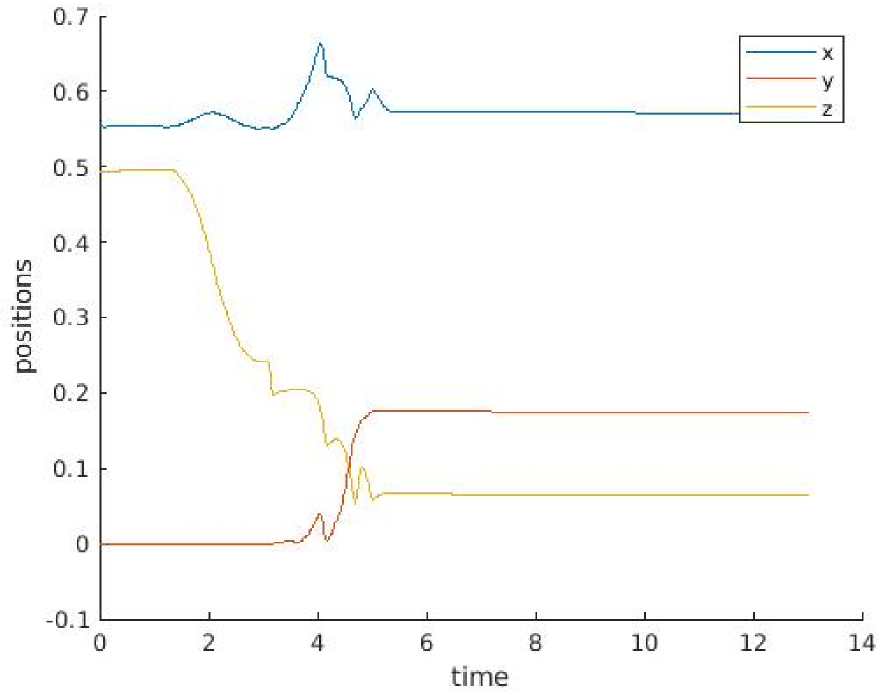

Figure 7: Real Robot with Feedforward controller position graph

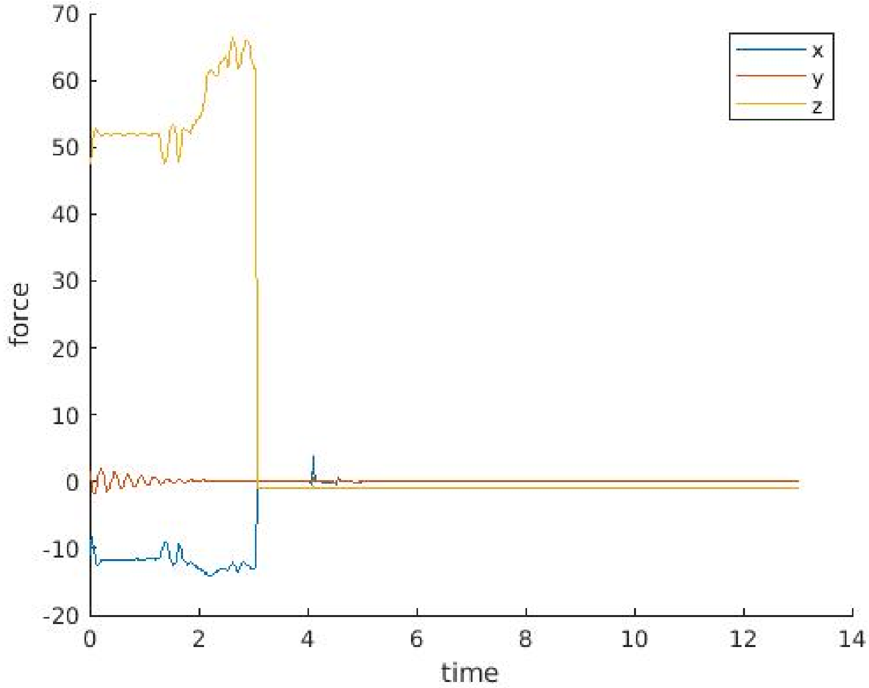

Figure 8: Real Robot with Feedforward controller force graph of end effector

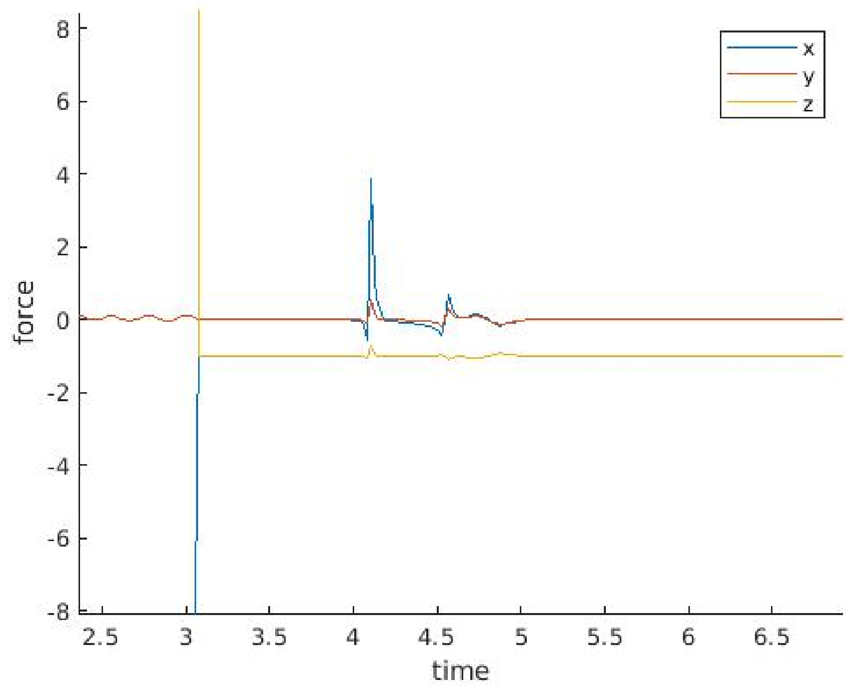

Figure 9: Real Robot with Feedforward controller force graph of end effector Zoomed for Steady state value

2.2.2 feedforward plus feedback-based (PI-controller) results:

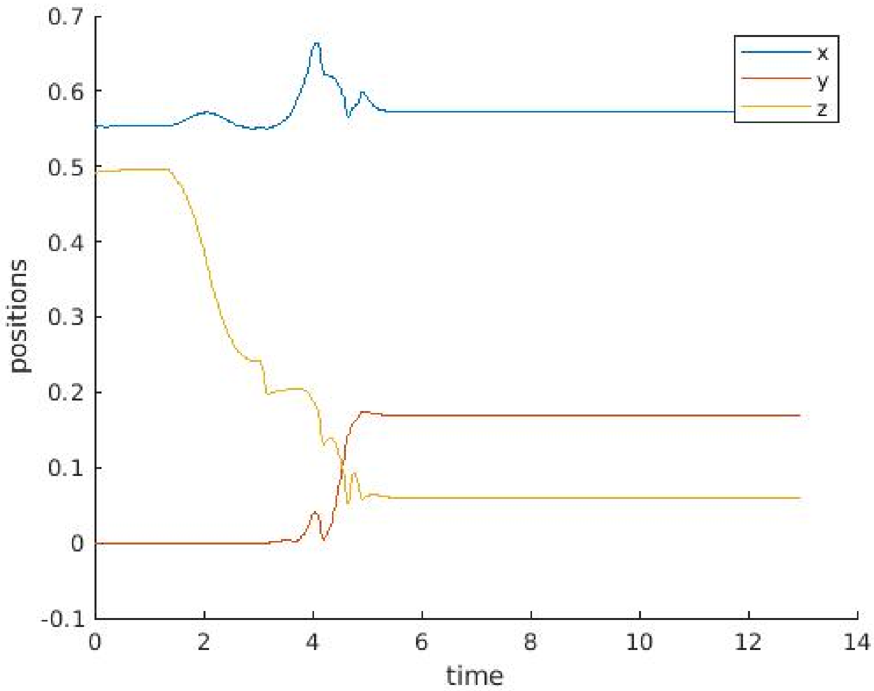

Figure 10: Real Robot with feedforward plus feedback PI controller position graph

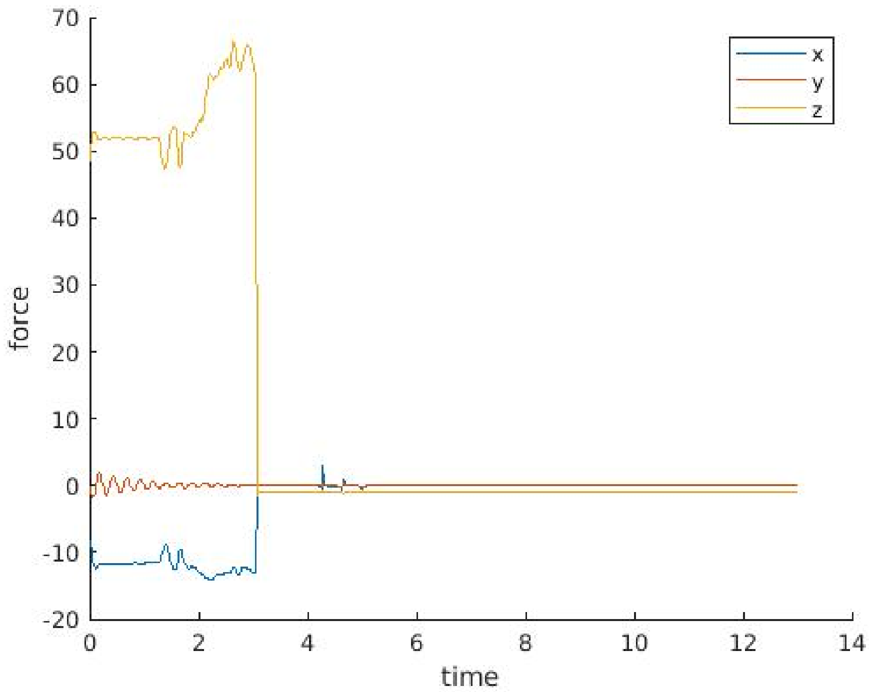

Figure 11: Real Robot with feedforward plus feedback PI force graph of end effector

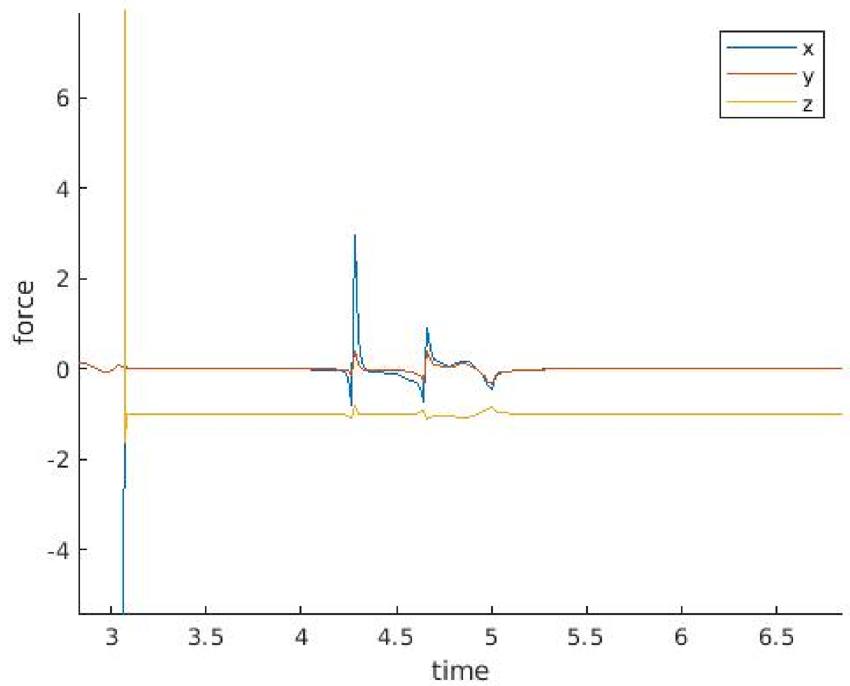

Figure 12: Real Robot feedforward plus feedback PI force graph of end effector Zoomed for Steady state value

3 Discussion of the results:

1. On the analysis of all the results, we can see that the desired force in the end effector is attained after few cycles(4-5 secs) of operation. The steady state error is reached to

$$f = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

2. The same is the case for Ideal robot and Real Robot with both the feedforward controller and feedforward plus PI feedback controller.

3. But in all the scenarios, the robot keeps falling, because we are controlling only the force of the end effector and not the dynamics of the robot

4. The robot falls after exeucting the desired trajectory at around 4-5 secs for each case.(Can be seen clearly at the zoom versions of all graphs above)

5. PI controller performance:

a.) The results above are computed with $K_p = 0.8$ and $K_i = 0.6$

b.) With the bad tuning, for example with high values of tuning parameters like $K_p = 2.0$ and $K_i = 1.6$, we can see the robot behavior is very bad and it fails to attend the desired force in the steady state. Also we have this error from mujoco as with the high values of tuning parameters, the simulation is not able to perform the computed operation with the current Robot dynamics. "Nan, Inf or huge value in QACC at DOF 0. The simulation is unstable."

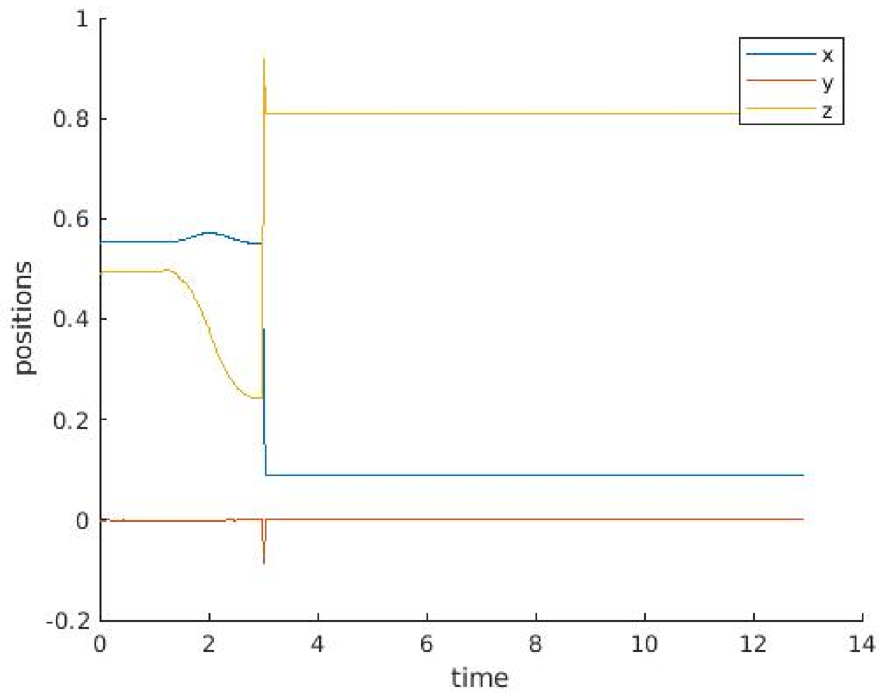

Figure 13: Real Robot with feedforward plus feedback PI pos graph of end effector with $K_p = 2.0$ and $K_i = 1.6$

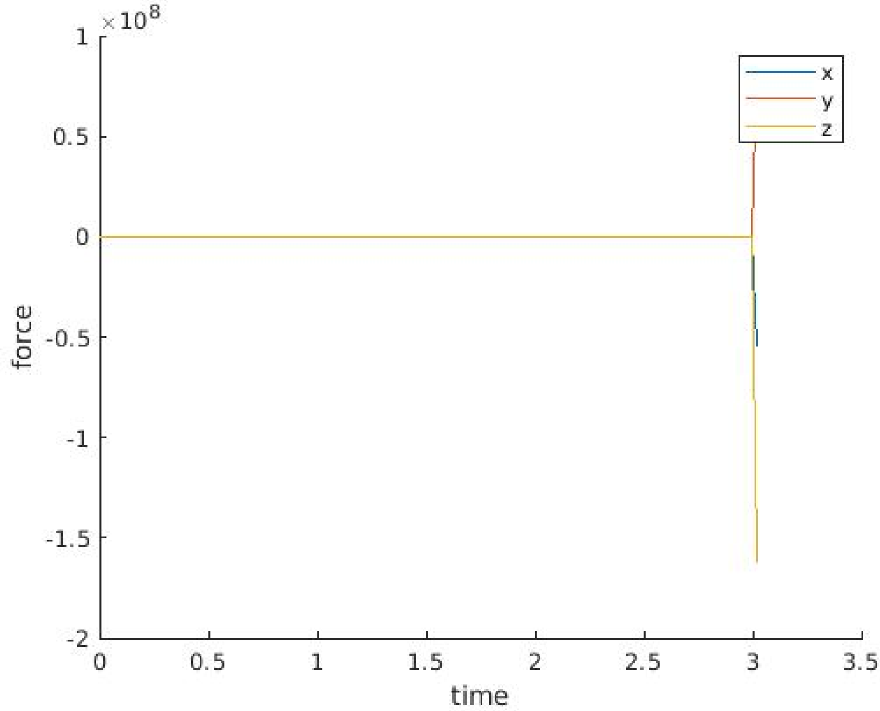

Figure 14: Real Robot feedforward plus feedback PI force graph of end effector with $K_p = 2.0$ and $K_i = 1.6$

4 Answers to the questions:

4.1 In which coordinate frame is the force/torque specified?

The coordinate frame in which the force or torque is specified is in Robot's end effector frame.

4.2 Is your force controller stable such that the robot remains in single stable robot configuration for a long time? If not, what technique can be used to stabilise it?

No, The Robot is not stable for a long time. The reason is we are controlling only the force of end effector. We need to control the position of Joints (including the dynamics of Robot). Maybe we should use a Hybrid controller controlling the force and the position of joints as well to make the robot stable for a long time.

4.3 What is the benefit of the feedforward plus feedback-based controller compared to the feedforward only?

Feedforward plus feedback-based controller will defenitely improve the stability of the system and increase the performance whenever there is a major disturbance that will affect the system. Feedforward control works better mostly on the Ideal cases, but on the practical scenarios, it is always better to have a feedback control to improve the stability.

4.4 In which cases the integral term of feedback based controller is necessary?

The integral term is used to bring the control to desired steady state value. It will try to reduce the oscillations caused by the P-controller. It will have history of past errors to be integrated over a period of time and try to apply the integral action on the control value.