Robot Dynamics
&
Control

Assignment 3
Dynamic Robot Control

Subhransu Sourav Priyadarshan
5302091

Supervised by
Prof.Giorgio Cannata, Francesco Grella, & Giulia Baldini

Preface

The work is based on " Dynamic Control of a Manipulator " on Matlab and Simulink. Our aim is to give gravity compensation,linear joint and cartesian control and computed torque control to the manipulator.To facilitate these tasks, we need the help of some packages as well.

Acknowledgements

Under the supervision of Prof.Giorgio Cannata, Francesco Grella and Giulia Baldini, I learned a lot about the proposed work.Hence, I thank them for their continuous support.

Abstract

The assignment is about the Dynamic Control of a manipulator using Matlab and Simulink.The packages used for the work are "Robotics Systems Toolbox" , "Simscape Multibody" and "Robot Library Data Support Package".We are given with a simple simulation model of a UR5 robot and are supposed to control it providing the torque commands to the joints.

Keywords

UR5 Robot;Gravity Compensation; Linear Joint Control;Linear Cartesian Control;Computed Torque Control Joint Trajectory Tracking

Contents

1 Introduction 5

2 Methodology 5

2.1 Exercise 1 – Gravity Compensation 5

2.2 Exercise 2 – Linear Joint Control 6

2.3 Exercise 3 – Linear Cartesian Control 7

2.4 Exercise 4 – Computed Torque Control 7

3 Results and Discussion 8

3.1 Exercise 1 9

3.2 Exercise 2 10

3.3 Exercise 3 11

3.4 Exercise 4 12

4 Conclusion 12

5 References 14

1 Introduction

In this assignment, we are working on providing the torque commands to the joints to control a *UR5 robot*.The accuracy of dynamic parameters plays an major role in the precision, performance and robustness of these control algorithms. Moreover, a deep knowledge of the dynamic parameters is needed in path planning algorithms that take into account robot dynamics. Especially in the mechatronics area of robotics, model-based control is crucial for the increase of the system precision and reliability.The toolboxes used for the proposed work are Robotic Systems Toolbox , Simscape Multibody and Robot Library Data Support Package.This support package provides source mesh visualization and helps in getting a clear idea of the happenings.

Figure 1: UR5 Robot

2 Methodology

2.1 Exercise 1 – Gravity Compensation

In the first exercise, we are focused on giving a gravity compensation to the robot so that it doesn't fall due to gravity and remains intact without changing it's configuration.So our main focus is to make sure that we provide a block in the simulink file (as shown in the figure below) that will compensate the gravity and prevent the robot from making a free-fall due to gravity and remains intact. The only thing that we need to keep in mind is that the robot does not lose it's configuration.

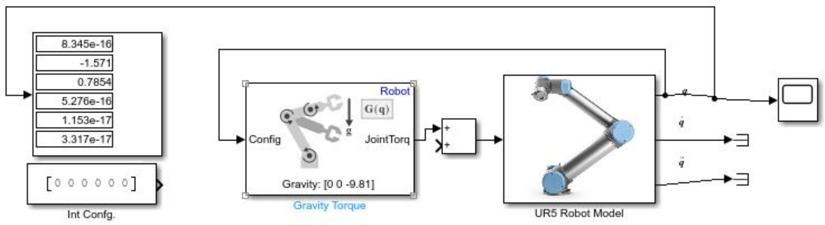

Figure 2: Initial values are zero

2.2 Exercise 2 – Linear Joint Control

Using the formula, $\mathbf{q^*} = \mathbf{q0} + \mathbf{q}$ we calculate the desired configuration(q*) of the robot on Simulink.After solving Exercise 1, we manage to get the initial configuration of the robot. Now we need to move it to a desired configuration.On Simulink, we add three blocks for calculating the transformation matrix, velocity and center of mass. Hence, this will provide the torque commands to reach the desired configuration.The blocks used are Joint Space Mass Matrix,Velocity Product Torque and Matrix Multiply.

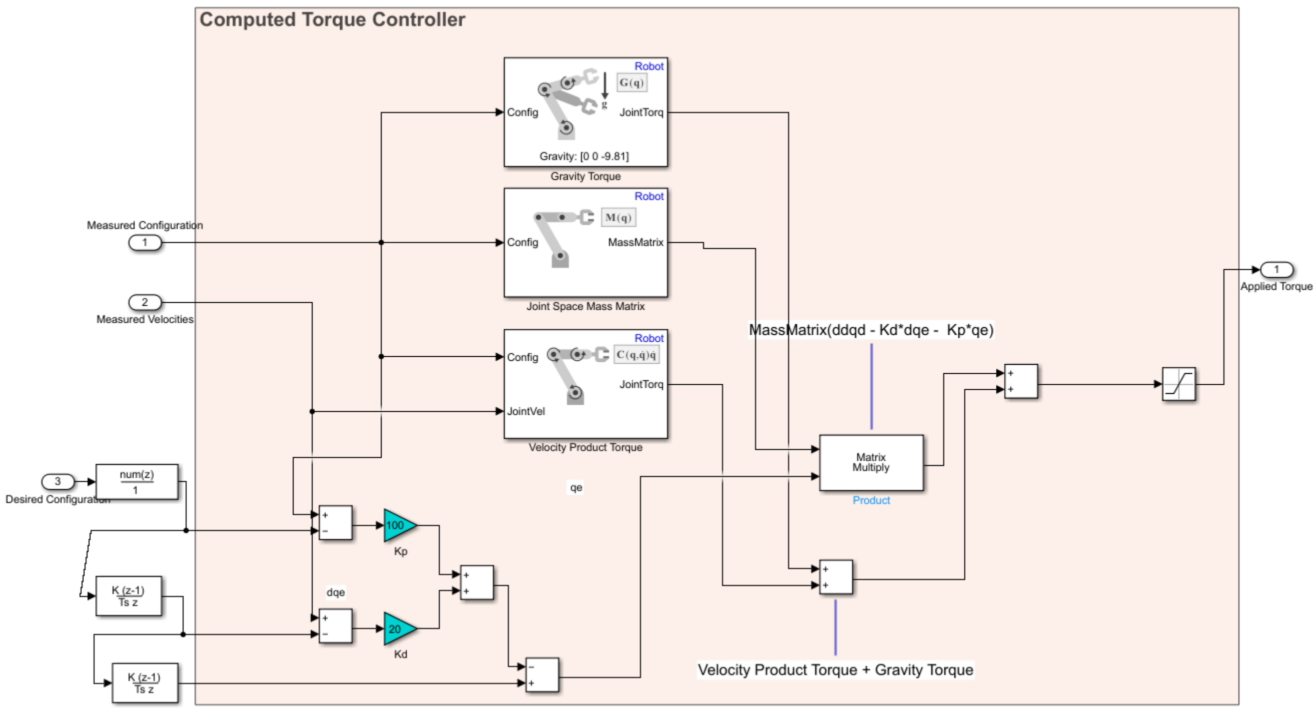

Figure 3: Block Diagram

2.2.1 Without Gravity Compensation

The robot will try to reach the desired position but will fail to do so because there is no gravity compensation and it will act as zero torque demo robot once again.

2.3 Exercise 3 – Linear Cartesian Control

Firstly, we need to determine the initial pose and orientation of the end-effector and we can get this information from "**Transformation matrix computation**" and "**Co-ordinate Transformation conversion"** blocks.The input is q from the robot model of the Ex-1 and then you save the values of the Euler angles x y z and the "**Trvec**" pose x y z values.With this blocks, before our PD controller we can get the desired pose of the end effector link.The input "**EulZYX**" is the vector of the last three components of x* and the input "**TrVec**" is the vector of the first three **(where x*** $=$ **x0** $+$ **x)**.

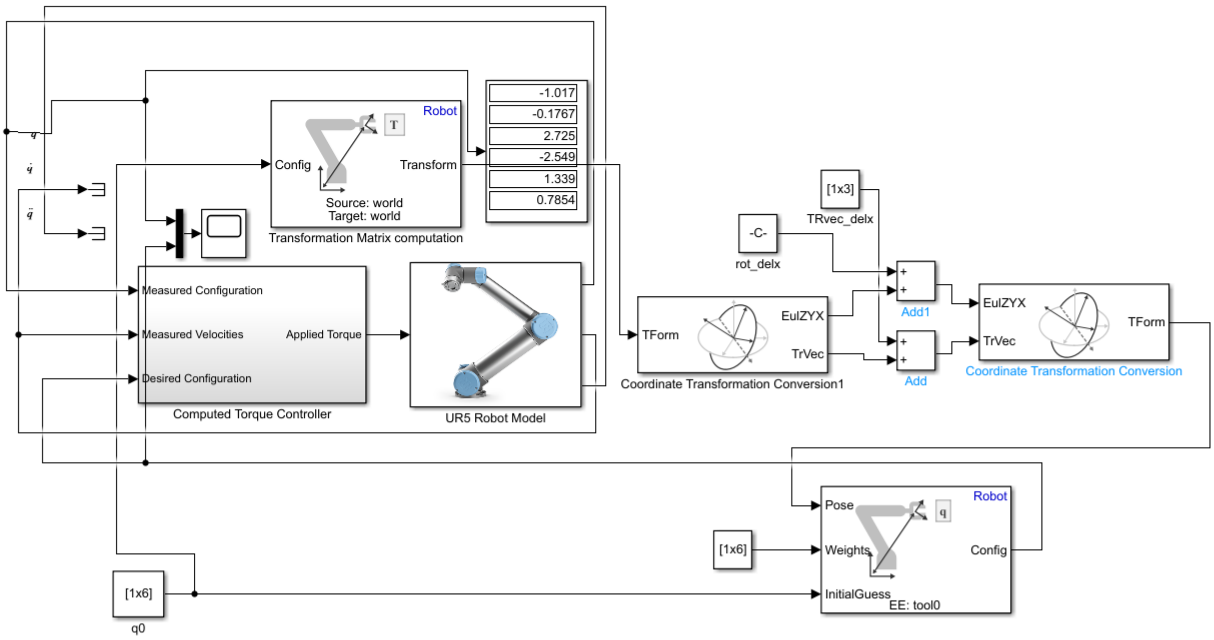

Figure 4: Block Diagram

2.4 Exercise 4 – Computed Torque Control

We need to use the block of "polynomial trajectory" to get position,velocity and acceleration and then need to proceed forward by connecting it to the clock.

If we remove the damping coefficient from each joint, then the shocks received at each joints are minimized and the robot becomes robust and the robot works efficiently.

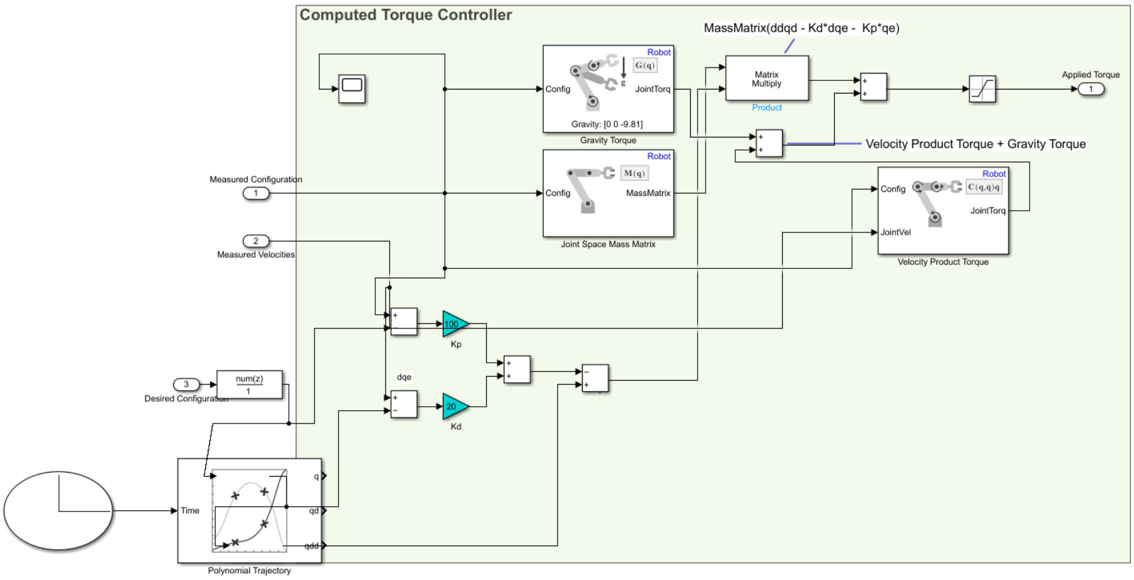

Figure 5: Block Diagram

3 Results and Discussion

The UR5 robot is initially in a zero-torque (see in the figure below)

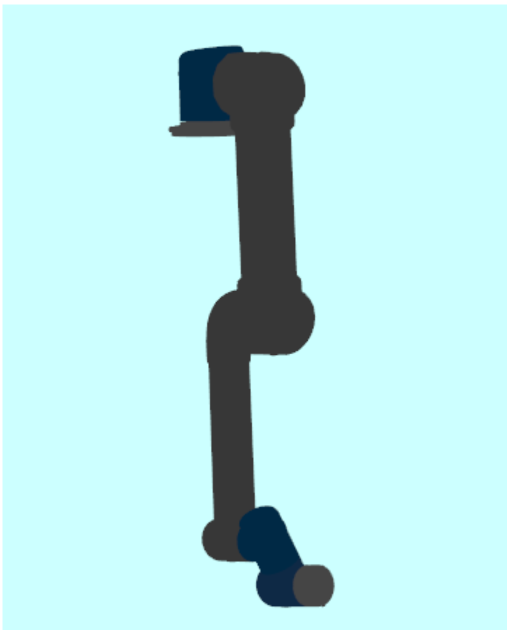

Figure 6: Zero-torque Robot

The scopes attached in the figures compares the desired position,velocity and acceleration with the initial ones.

3.1 Exercise 1

The robot stays intact with the anti-gravity effect provided to it in the z-direction.

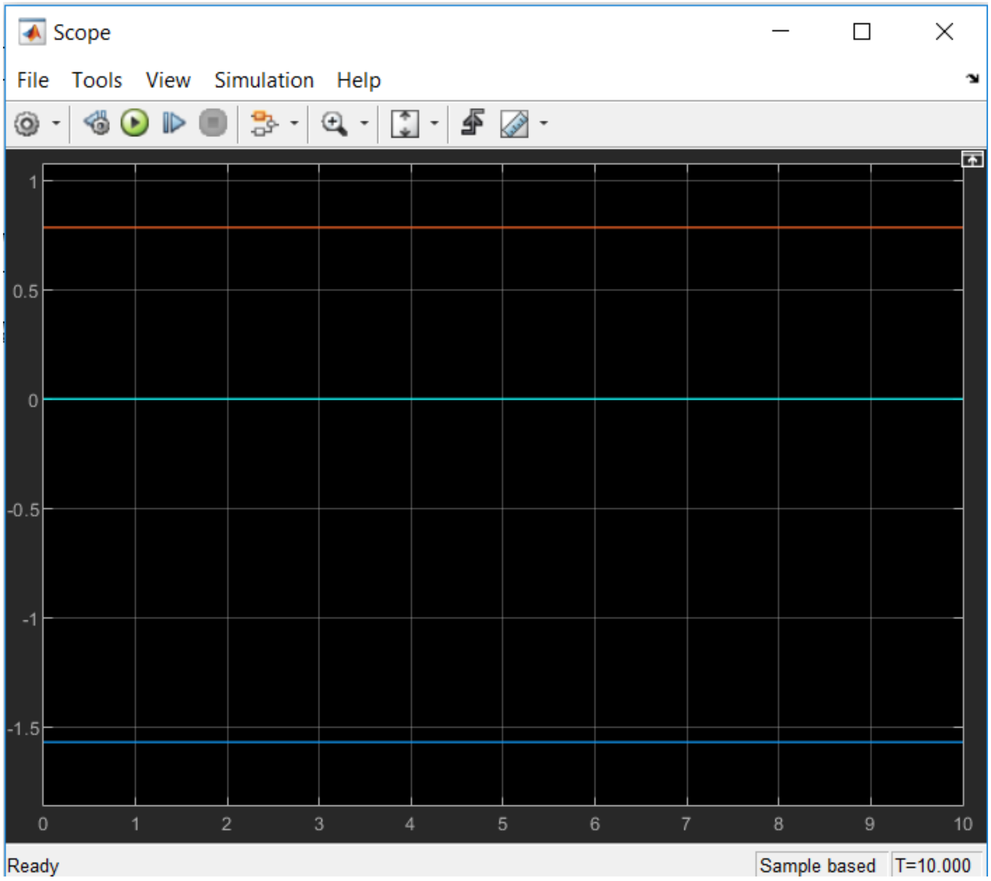

Figure 7: Gravity Compensation

Figure 8: Gravity Compensation

3.2 Exercise 2

With the help of the given parameters, we were able to achieve the desired configuration.The output is being displayed in four views.

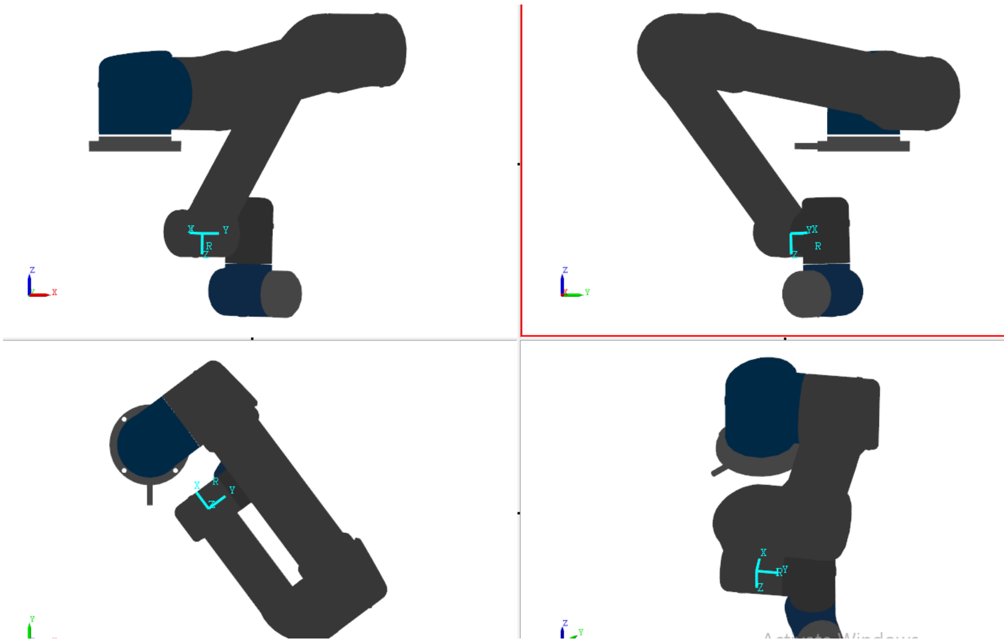

Figure 9: Robot at max stretch

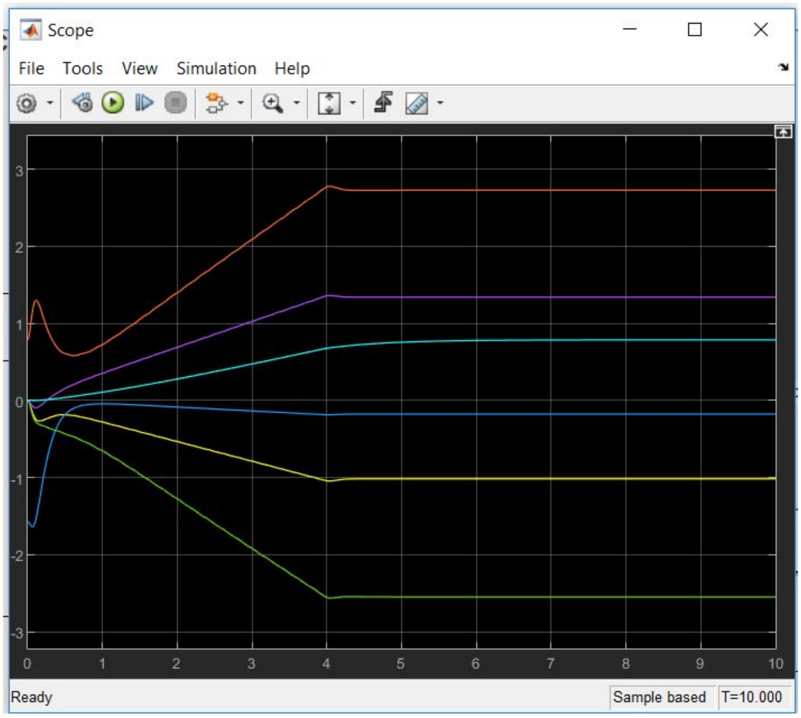

Figure 10: Scope

3.3 Exercise 3

With the help of the given parameters, we were able to achieve the desired cartesian configuration of the end-effector.

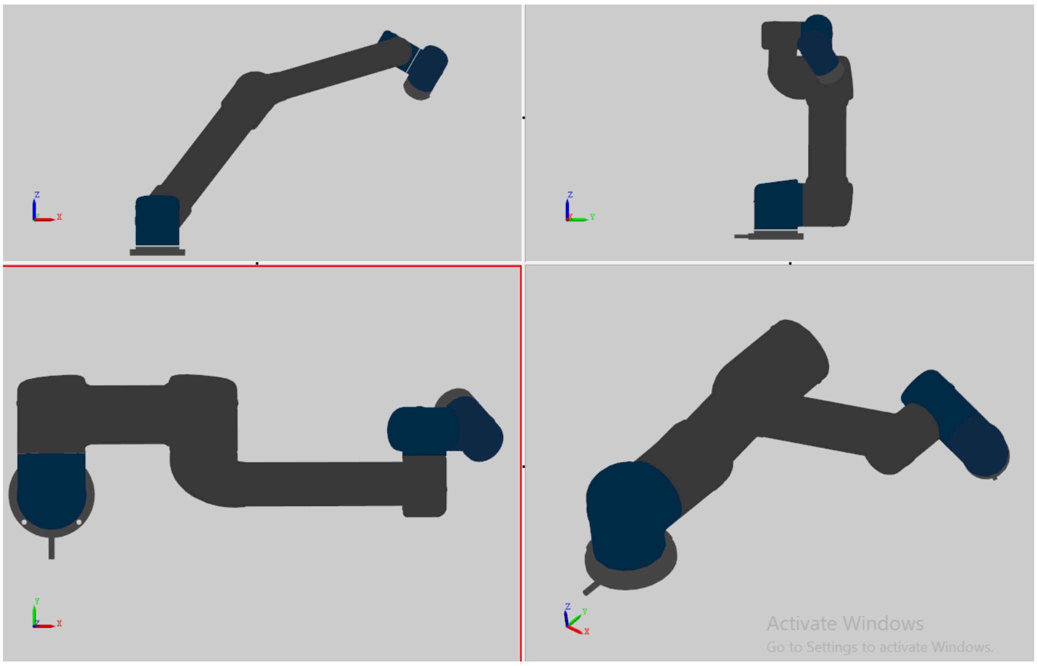

Figure 11: Robot at mid stretch

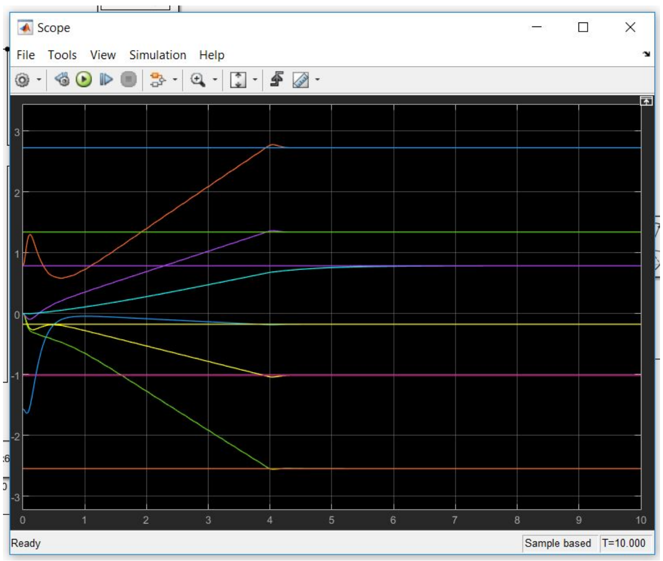

Figure 12: Scope

3.4 Exercise 4

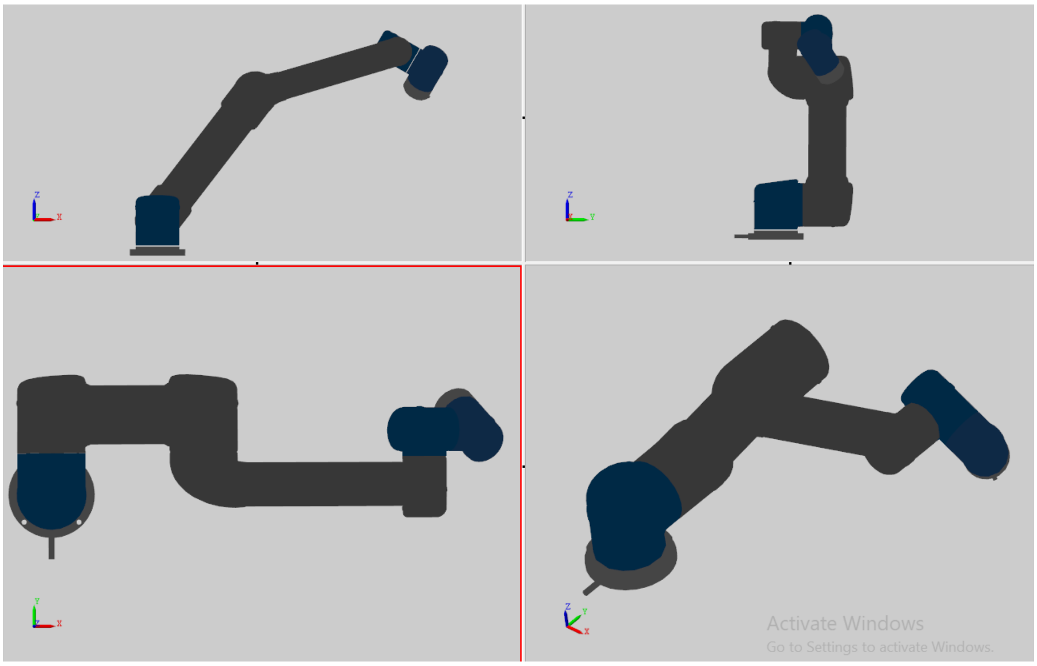

Figure 13: Robot at mid stretch

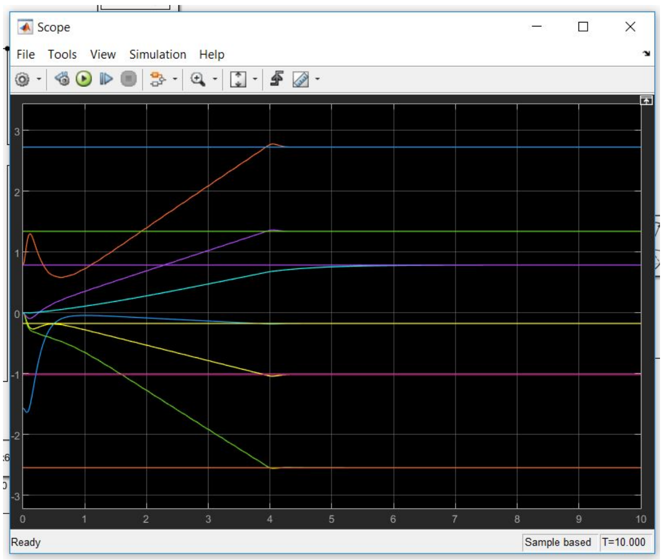

Figure 14: Scope

4 Conclusion

The Universal Robots UR5, a highly flexible robotic arm enables safe automation of repetitive, risky tasks.The UR5 is very easy to set up and program. This gives it one of the fastest payback times on the market. This robot can be operational in less than half a day thanks to the simple way of programming with a 3D visualisation.In the assignment, we analysed and compared the parameters on different situations.Our entire work is based on analysis through graphs and reading the outputs on Simulink.

5 References

1. M. Gautier and W. Khalil. Exciting trajectories for the identification of base inertial parameters of robots. International Journal of Robotics Research, 11(4):362–375, 1992.

2. Li Ding, Hongtao Wu, Yu Yao, and Yuxuan Yang. Dynamic model identification for 6-DOF industrial robots. Journal of Robotics, 2015, 2015.

3. Jun Wu, Jinsong Wang, and Zheng You. An overview of dynamic parameter identification of robots. Robotics and Computer-Integrated Manufacturing, 26(5):414–419, 2010.

4. The Stm, S T Mems, S T Mems, and U S B Otg. UR5 User Manual. (January):1–42, 2014.

5. Martin Grotjahn, Bodo Heimann, and Houssem Abdellatif. Identification of friction and rigid-body dynamics of parallel kinematic structures for model-based control. Multibody System Dynamics, 11(3):273–294, 2004.