Lappeenranta University of Technology

BL40A2820 Motion Control and System Identification

Assignment

1 Introduction

The electromechanical system for this course's practical work contains a toothed belt that is driven by a servo motor connected directly to the driving pulley and the relevant control logic. The assignment work for this course is divided into four parts, which include modeling the 2-axis manipulator, analyzing its dynamics, implementing PLCopen Motion Control of 2-axis manipulator, and designing and identifying position control. Each of these parts is crucial in understanding and implementing the functioning of the electromechanical system effectively. This assignment aims to provide us with a comprehensive understanding of the theoretical concepts and practical applications of electromechanical systems in the real world.

Motion control is a critical area of study that plays a vital role in many industrial applications, including robotics, automation, and manufacturing. The primary objective of motion control is to achieve precise control of movement and positioning of electromechanical systems. System identification is another essential aspect of motion control that deals with the development of mathematical models for these systems.

2 Modeling the 2-axis manipulator

In the first assignment part task is to construct a flexible model of 2-axis manipulator.

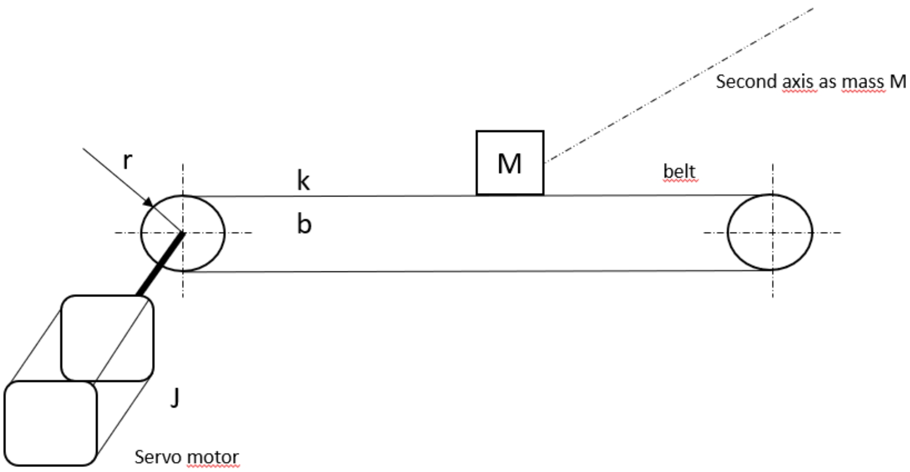

Figure 1: Mechanical system

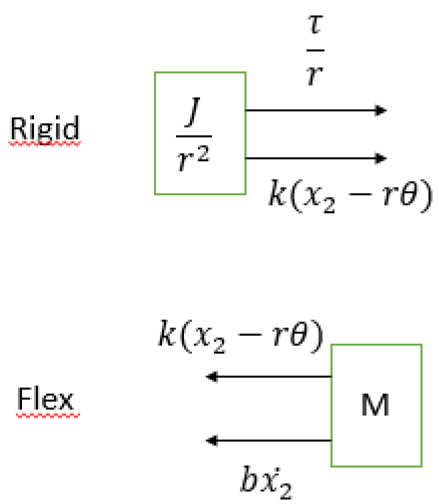

Figure 2: Free body diagram

Rigid: $\frac{\tau}{r} + k(x_2 - r\theta) = \frac{J}{r^2} r\ddot{\theta} => \frac{\tau}{r} + kx_2 - kr\theta = \frac{J\ddot{\theta}}{r} => \ddot{\theta} = \frac{\tau + krx_2 - kr^2\theta}{J}$

Flex: $M\ddot{x} = -(x_2 - r\theta) = b\dot{x_2} = -kx_2 + kr\theta - b\dot{x_2} => \ddot{x} = \frac{-kx_2}{M} + \frac{kr\theta}{M} - \frac{b\dot{x_2}}{M}$

State space: $\dot{z} = Az + Bu$
$y = Cz$

$$\dot{z} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x_2} \\ \ddot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-kr^2}{J} & 0 & \frac{kr}{J} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{kr}{M} & 0 & \frac{-k}{M} & \frac{-b}{M} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x_2 \\ \dot{x_2} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \\ 0 \\ 0 \end{bmatrix} \tau$$

$$y = Cz => y = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x_2 \\ \dot{x_2} \end{bmatrix}$$

3 Analysis of dynamics

The analysis of dynamics was done by simulating manipulator axis model in Matlab. Matlab has built-in functions for step response, impulse response and bode plots and these functions were used in analysis part. This part was done by changing parameter values of spring constant and friction coefficient and then looking at how it affected the response of the system.

First the value of spring constant k was changed, and results analyzed. The smaller the value of spring constant k is, the more elastic the system becomes and therefore it has a faster response to external forces. This also makes the system more sensitive to smaller changes. These conclusions were confirmed by plotting the step response of cart's position as it became more sensitive when the value of k was decreased. When value of spring constant k is decreased enough, the system will become unstable. When the value of spring constant k increases, the system becomes stiffer which means that a greater external force is required to deform the system. The system also becomes less sensitive to smaller changes when spring constant k increases.

Changing the value of friction coefficient b affects the amount of force required to move the belt along its axis. Increasing the value of b increases the resistance to motion. When the torque of motor stays the same and friction is increased, the pulley will move at slower speed. This also means that the position of the cart changes at a slower rate. These changes were confirmed by plotting step responses of the system. Increasing the value of friction coefficient also resulted in slower rising times. When the value is decreased, it will make it easier for the motor to move the pulley which results in faster rising times.

The resonance frequency of the system depicting cart's velocity is shown in Figure 3. The picture shows resonance frequency at 3.09e03 rad/s with a magnitude of 43.4 dB.

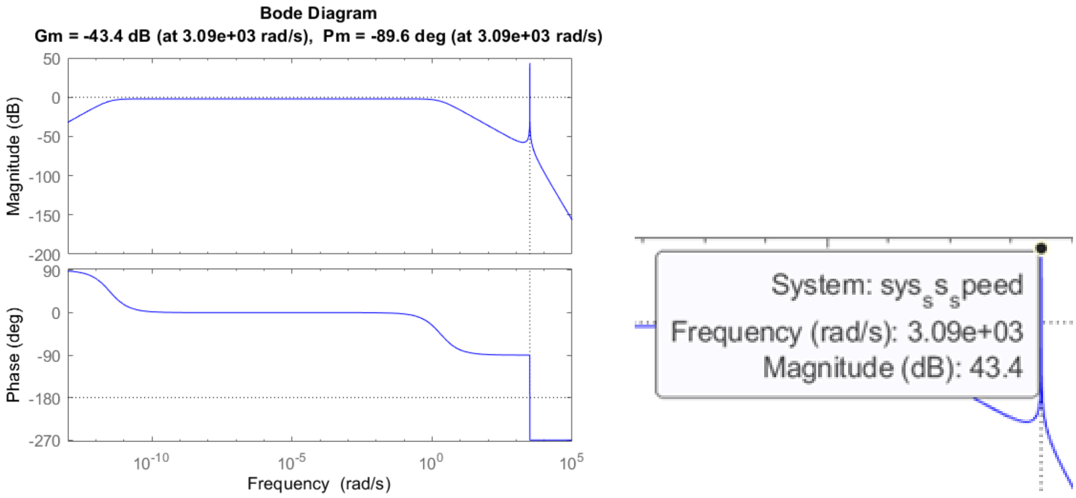

Figure 3. Cart velocity Bode diagram.

In Figure 4 the resonance frequency of the system depicting cart's position is shown. The resonance frequency is at 3.09e03 rad/s with a magnitude of –26.4 dB.

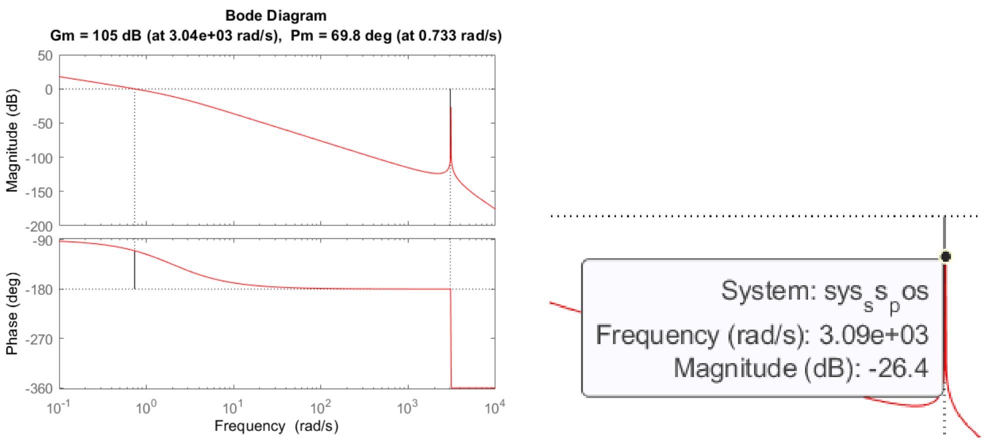

Figure 4. Cart position Bode diagram.

In both diagrams the value of resonance frequency is the same, but magnitude is different. The natural resonance frequency is affected by changes in mass or spring constant values of the system. The oscillating system's frequency would rise if the spring was firmer. The resonance frequency of the system would be lowered by adding mass to it [1]. The appropriate resonance frequency will depend on the specific application and requirements of the system. When the value of resonance frequency is stable the system and it smooth and the robot's response to external disturbances and its ability to maintain its position. Due to changing the resonance frequency of the control system for a system will affect the system's stability, response time, and susceptibility to overshoot [2]. If decreasing the resonance frequency; the system will vibrate more slowly in response to external disturbances. On the other hand, increasing the resonance frequency will result in a faster response to external disturbances, but it may also make the system more susceptible to overshoot and instability. It is important to design the control system to avoid resonance at the natural frequency of the system, as this can cause large vibrations that may damage the system or affect its performance [3].

4 PLCopen Motion Control of 2-axis manipulator

In this assignment, the simulated belt drive is operating using PLCOpen Motion Control blocks. In assignment the skeleton for blocks are given and connection to simulated belt drive is already completed. Assignment is implemented using ABB Automation builder (codesys 2.2). For making development easier the visual GUI are done to see system functionality.

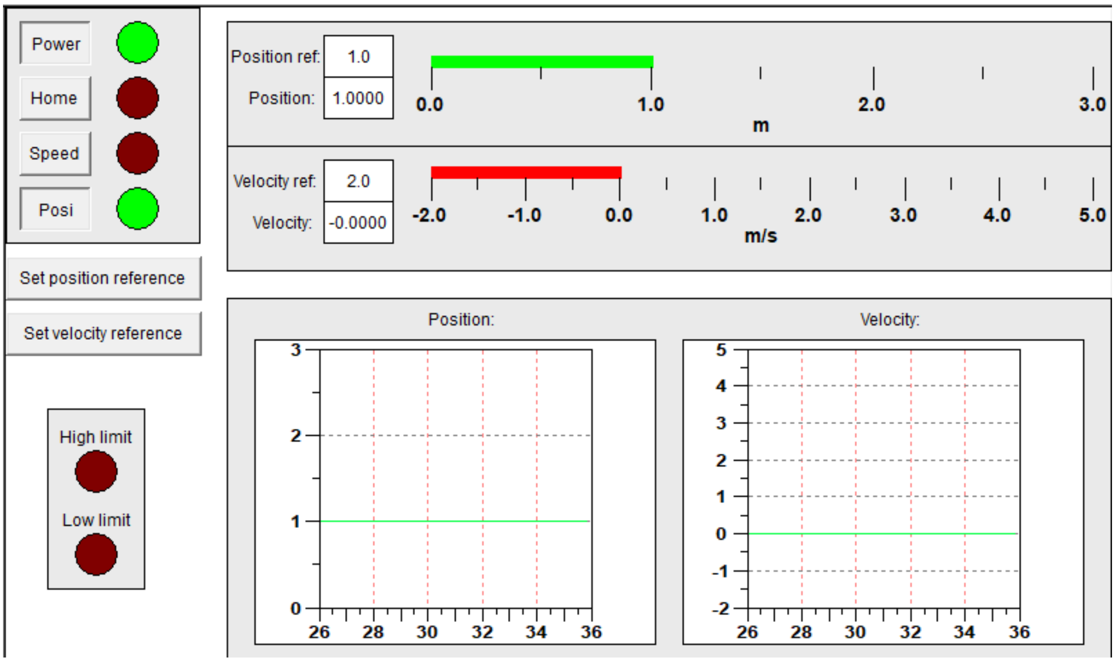

Figure 5. ABB automation builder GUI.

Following MC blocks are implemented:

MC_HOME

MC_MOVE_ABSOLUTE

MC_MOVE_VELOCITY

MC_POWER

MC_HOME:

Function Block for homing functionality.

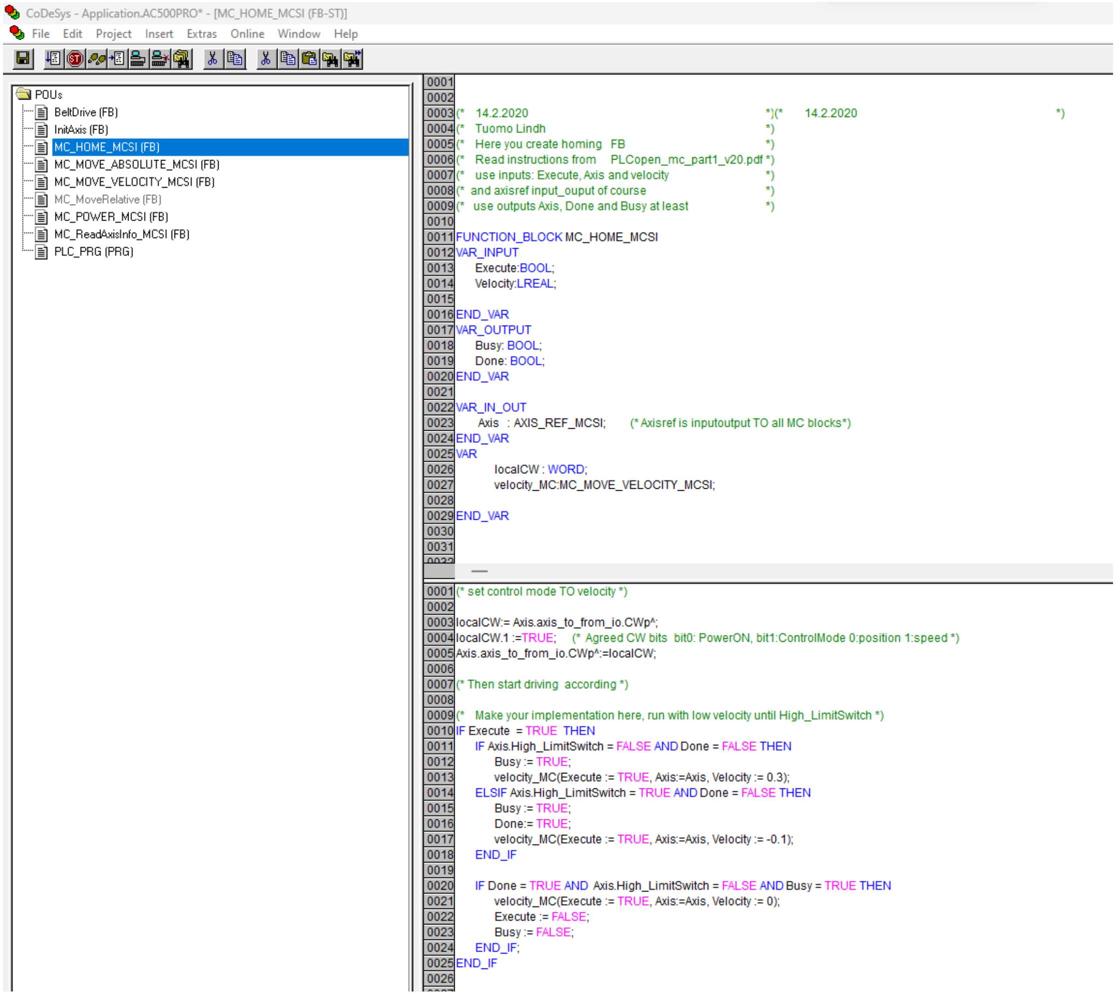

Figure 6. Homing functionality implementation.

MC_MOVE_ABSOLUTE:

Function Block commands a controlled motion to a specified absolute position.

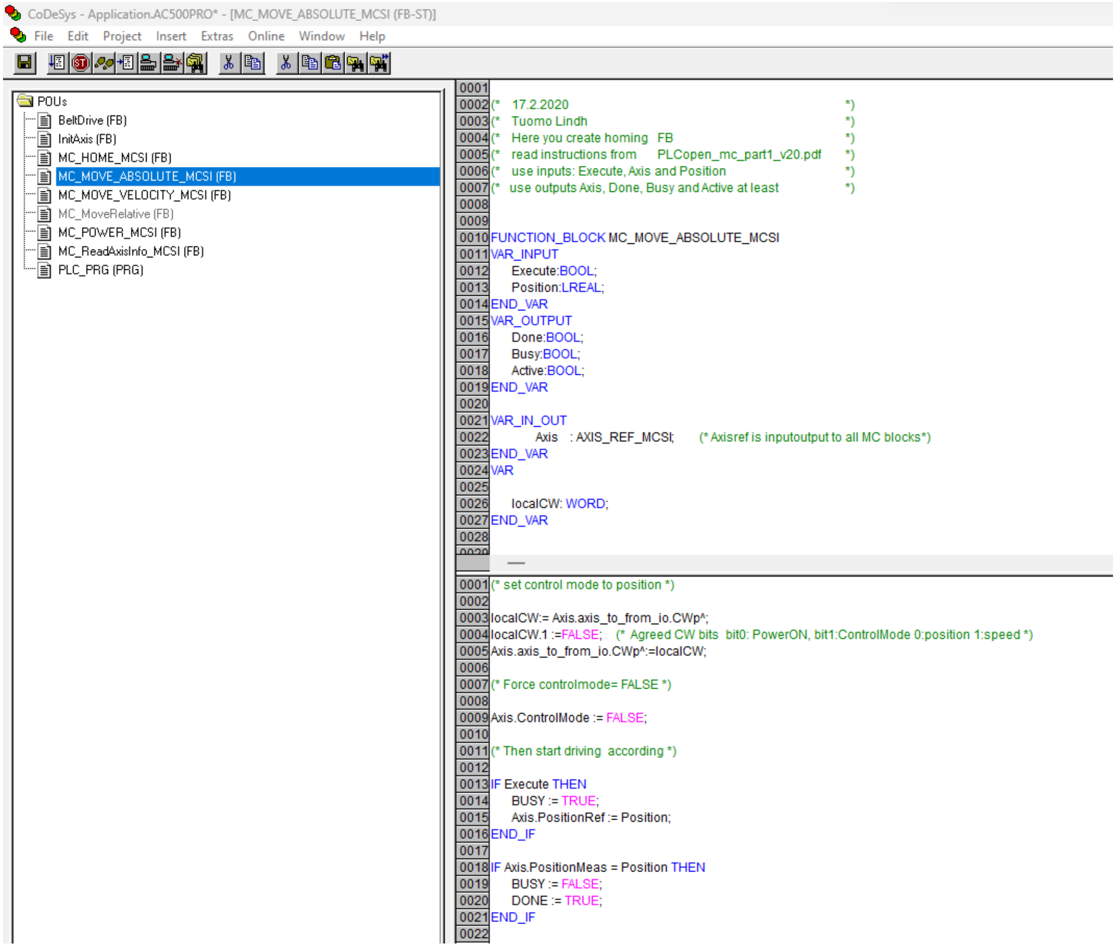

Figure 7. Move absolute functionality implementation.

MOVE_VELOCITY:

Function Block commands a never-ending controlled motion at a specified velocity.

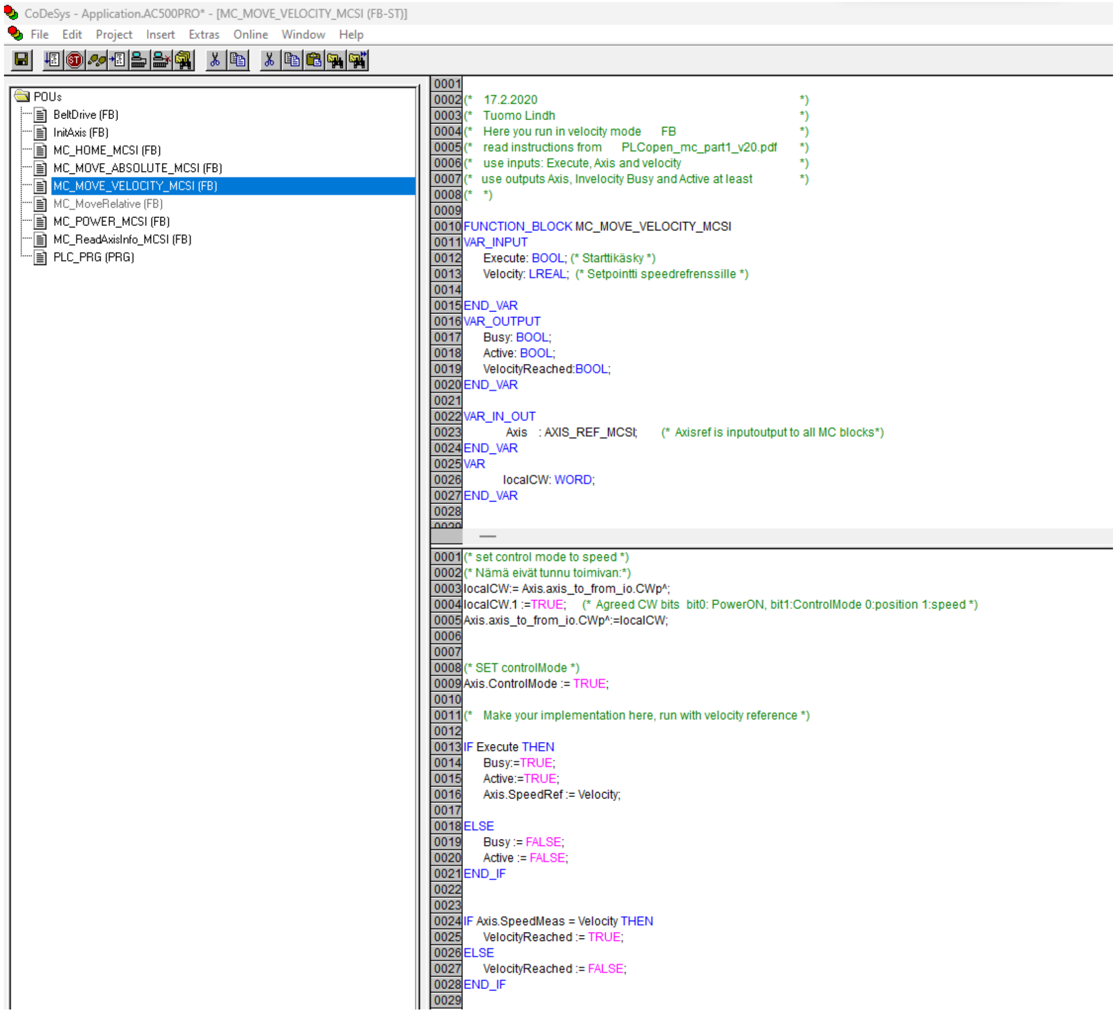

Figure 8. Move velocity functionality implementation.

5 Position control design and identification

The final part of the assignment was system identification and control design. The calculations and results can be found from MATLAB & Simulink files: assignment_part5.m, Model_posCTRL_SS_2019a.slx

First created controllers for model. P-controller for position and velocity controller (PI) with feedback loop.

The next parameters were found to velocity PI controller:

$Kp_{speed} = 0.12$ and $Ti_{speed} = 3.5$

P-controler to Position controller is

$Kp_{position} = 0.55$

Feedforward gain is set to

$Kp_{feedforward} = 0.720843$

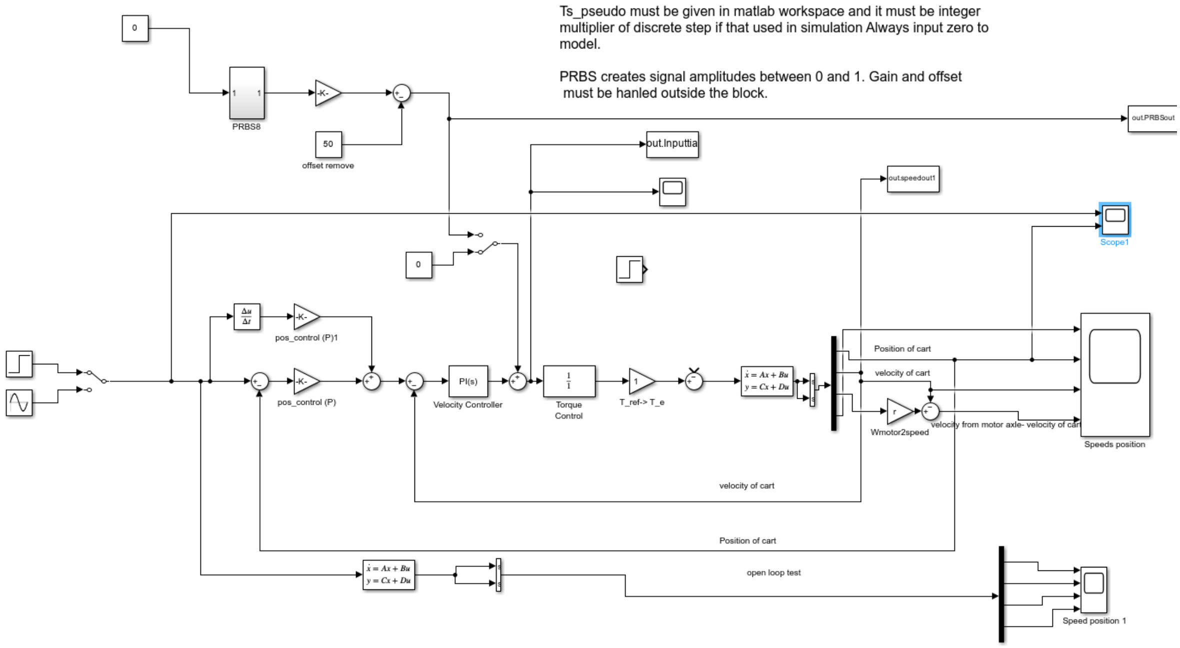

Figure 9. posCTRL_SS_2019a.slx Simulink diagram.

With these values we got a step response for both speed and position controller:

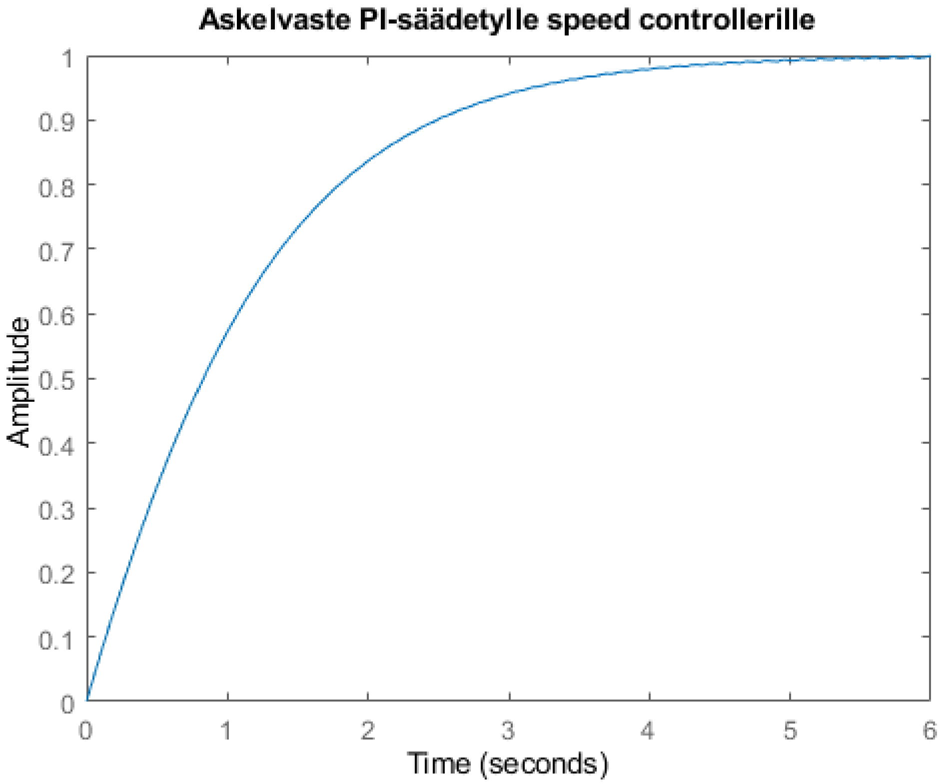

Figure 10. Step response to speed controller.

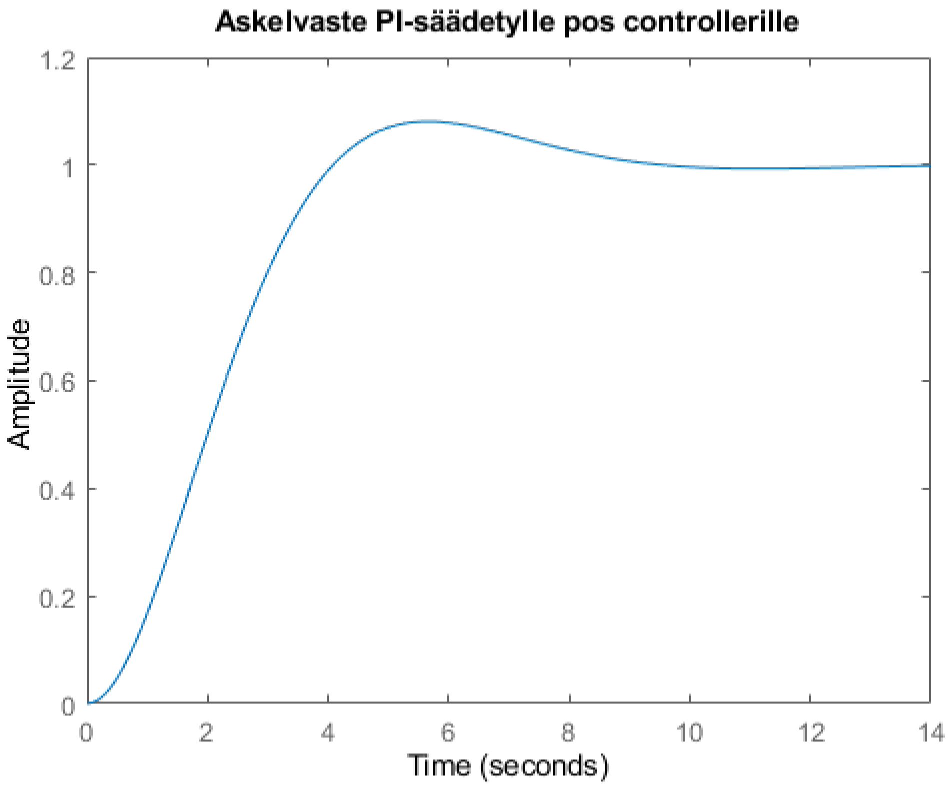

Figure 11. Step response to position controller.

Speed controller are set a little overshoot for purpose. It can be avoided set to $KTi_{speed} = 3$. Reason for this is that Simulink model posCTRL_SS_2019a.slx position step response is faster (risetime).

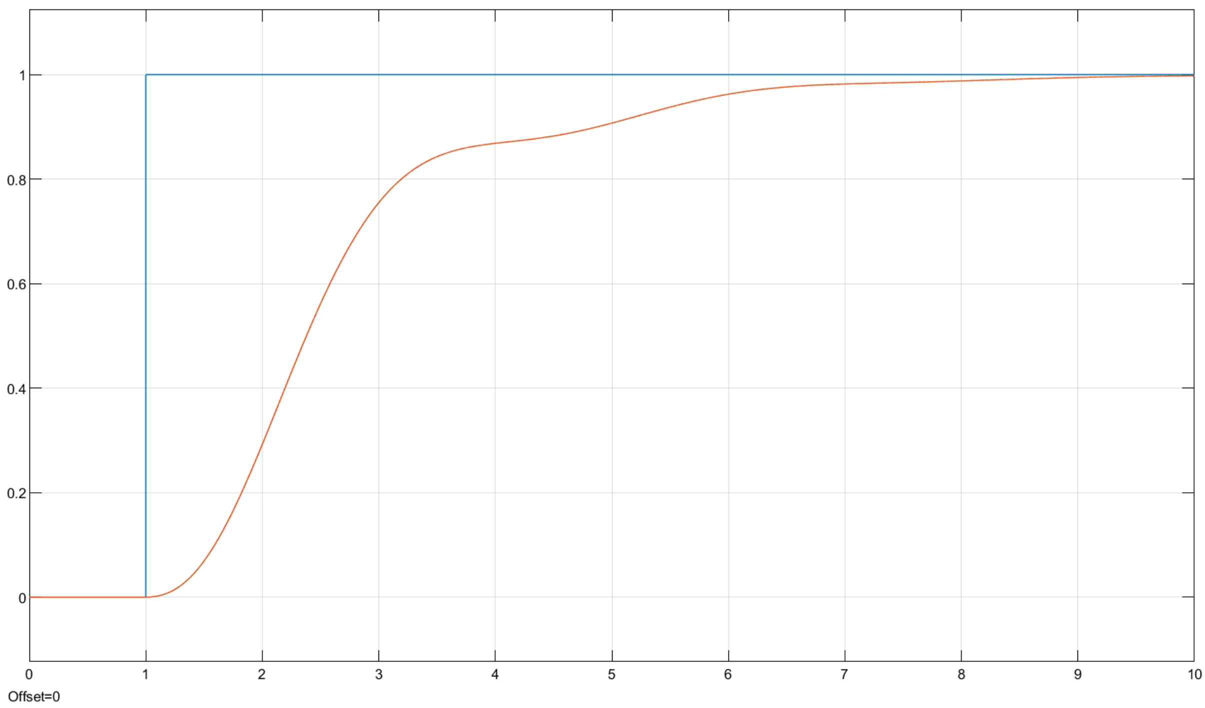

Figure 12. System step response.

Step change settling time is quite long. PI controller values affects the rise time but not a settling time.

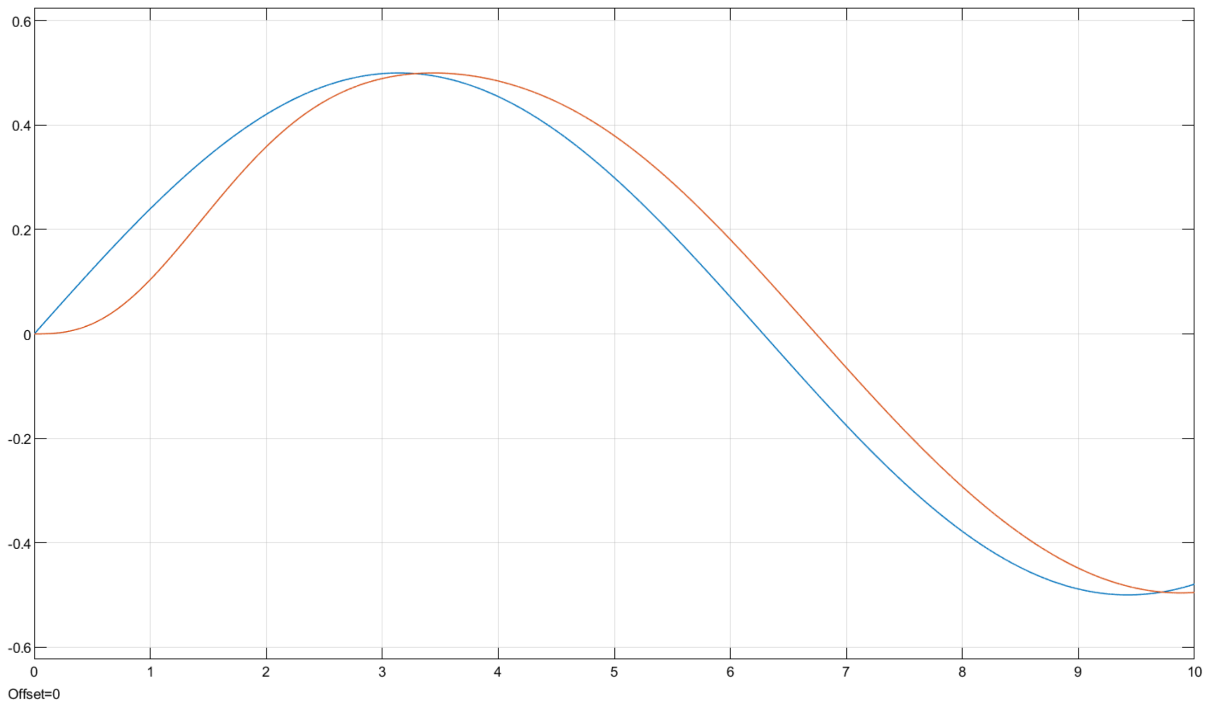

Figure 13. Sinusoidal signal input and system output.

Adding feedforward gain $Kp_{feedforward} = 0.720843$ the position response to follow sinusoidal signal seems to get better but it don't affect step changes so much.

System identification

This part is divided two parts. First, we test identification methods to compare (Assignment I) model to use MATLAB created PRBS excitation signal and secondly we use already recorded real measured data and use it to identification that includes real 2-axis system dynamics that are not included Assignment I model.

Manipulator real measurement data for different excitation signals are given ready for assignment. Used excitation signals are PRBS and sweeps for different start and end frequencies. We test PRBS and all sweep signals. It seems that sweep is giving a better identification result.

Assignment I model

Manipulator test data size are 835 records. So we create with Matlab function idinput function same size excitation signal. Test data are divided for model (300) and validation part (535).

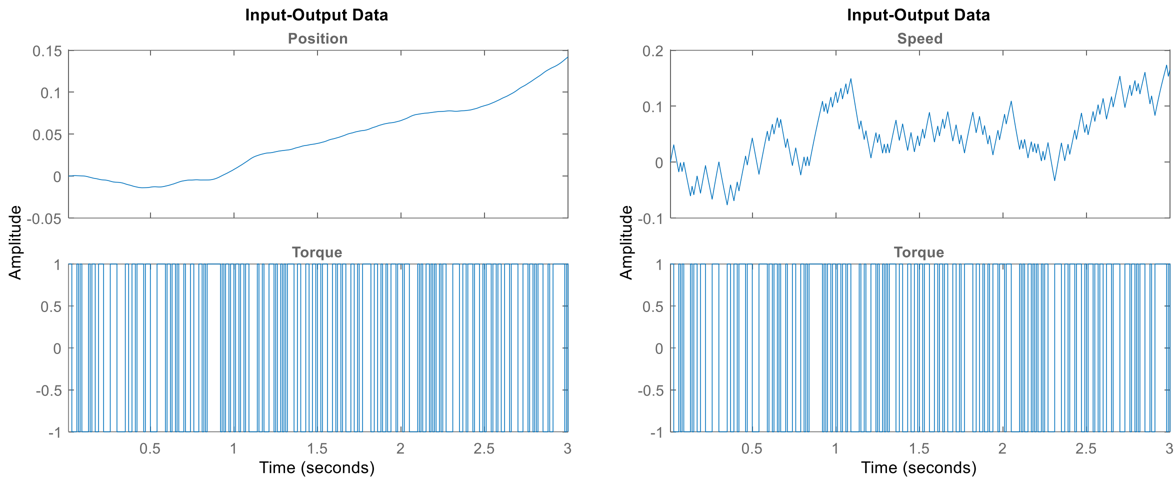

Figure 14. Excitation signal (PRBS) generated idinput.

Excitation signals was tested to modelled (Assignment I) space state for speed and position.

We tested identification methods OE, TF and impulse to own model. Created PRBS signal feed to identification methods and then compared to our model. Models parameters need to be found by try and error but finally we found a good estimation that corresponded closely to own model.

Using OE identification method, we managed to get high percentage (100 %) match for position and speed. OE model parameters polynomials nb=5, nf=3 and delay nk=1. Values are found by just testing different values.

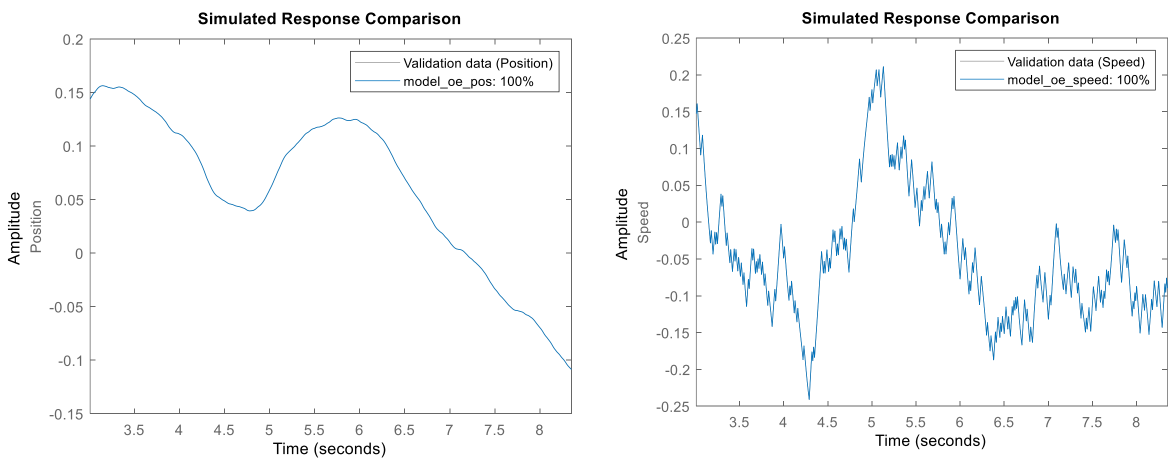

Figure 15. OE model for PRBS signal.

Real measured data

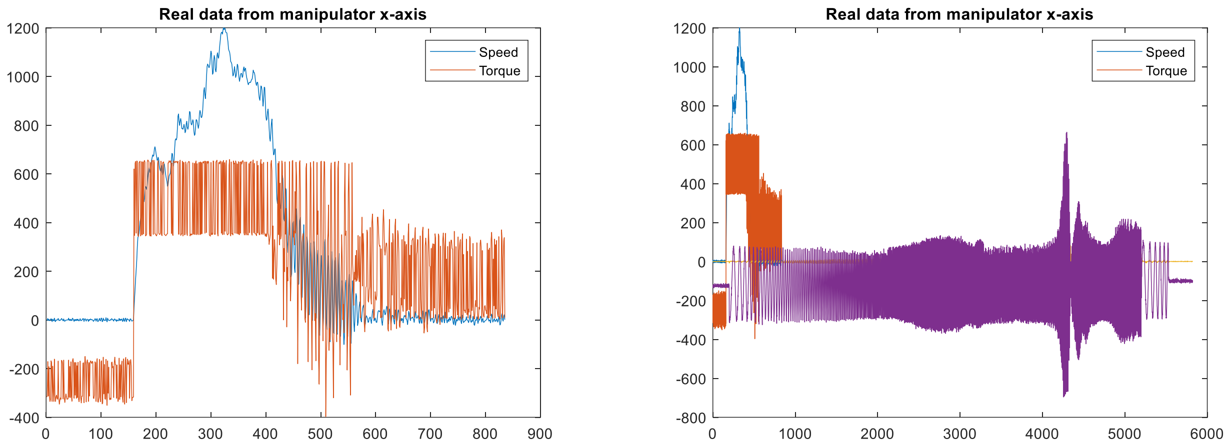

Figure 16. Measurement data outputPRBS_1500 (left) and outputSWEEP_20_1_2000 (right).

We load different test data to MATLAB using load function. Measured data are divided model data (300) and validation data (535).

Measured data are detrended with using MATLAB detrend function. It seems to get better result for identification. Used identification methods OE, TF, SS, Impulse, ARX and NLARX.

For OE identification method we found the best correspondence with sweep.

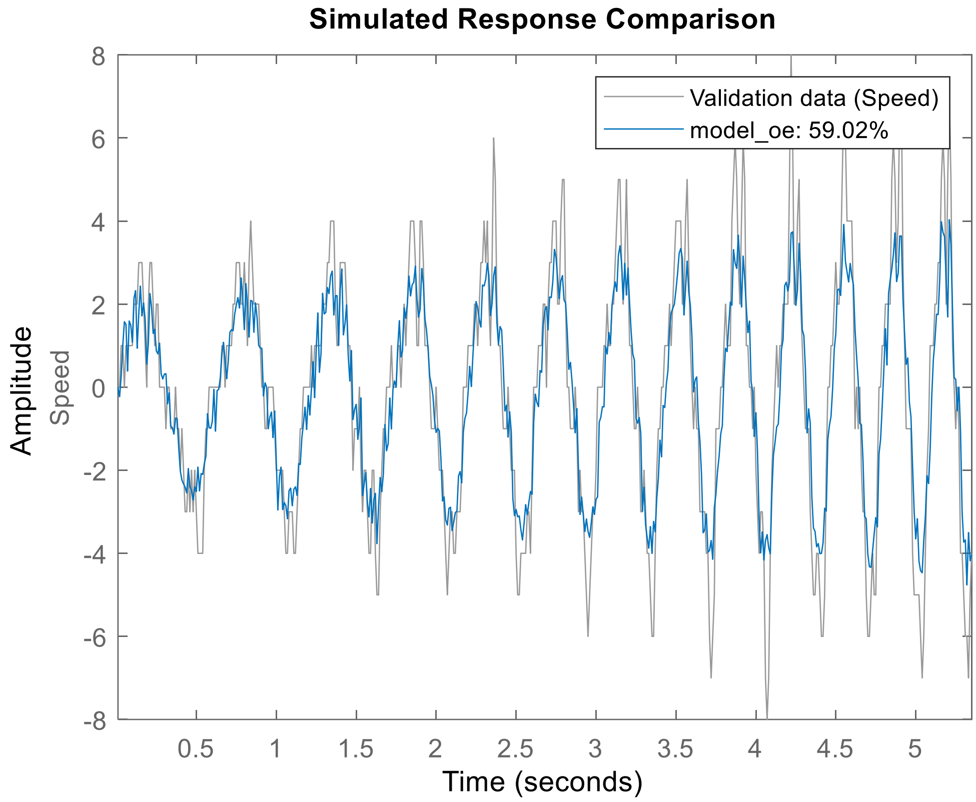

Figure 17. outputSWEEP_20_1_2000 signal and OE model.

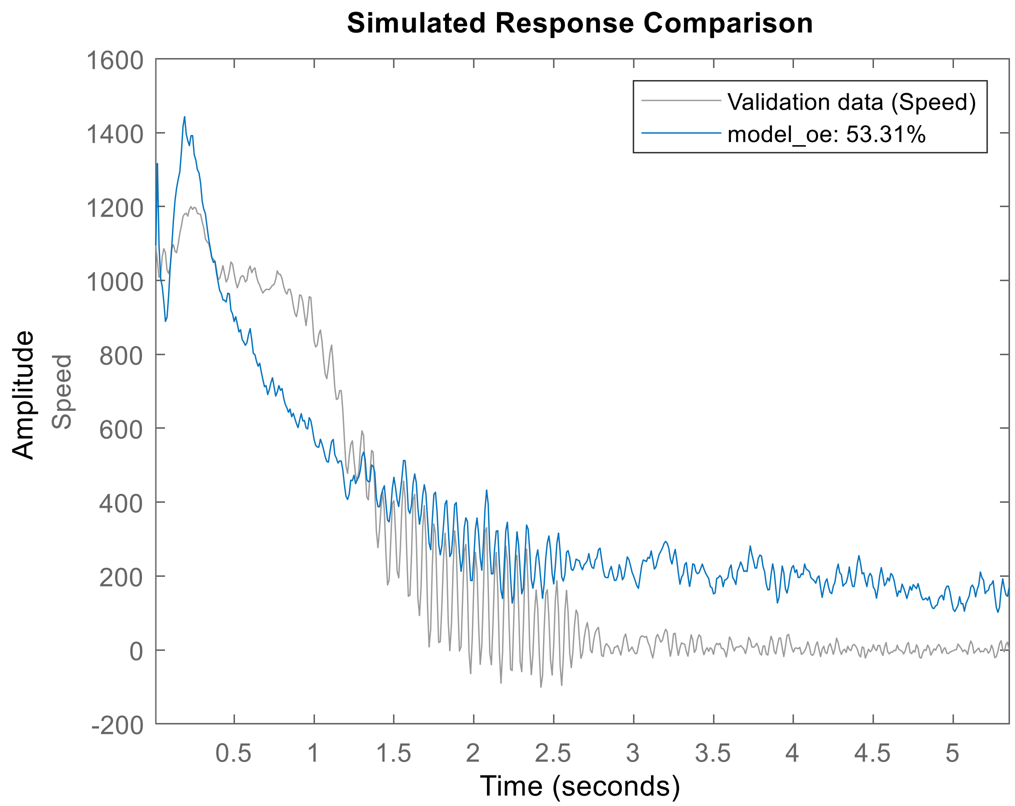

Figure 18. outputPRBS_1500 signal and OE model.

Finally, we compare OE model for real data and our Assignment I model and simulated both using our own PBRS signal.

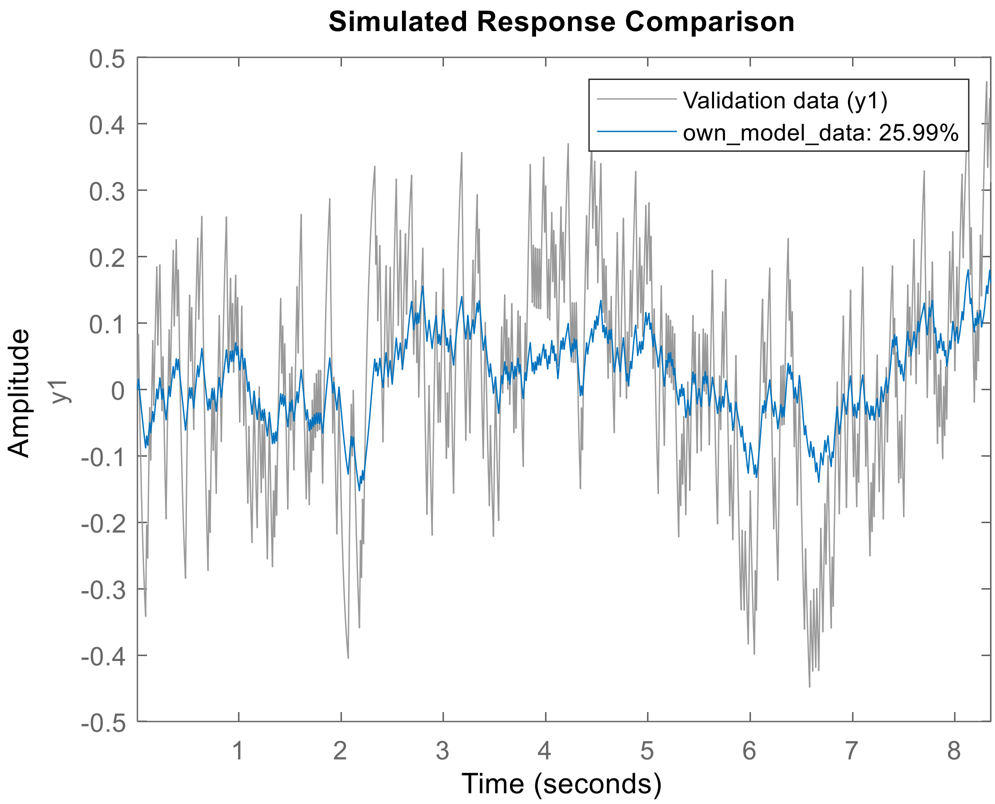

Figure 19. Comparison of models. Best modelled OE of real data and Assignment I model with same matlab generated PRBS input data.

Comparing OE and TF method estimation to space state model using same PRBS input signal. The result of OE and TF is quite far from perfect. OE and TF can follow somehow but the model is missing some dynamics that causes oscillation. That may cause by the another axis dynamics.

6 Conclusion

Motion Control and System Identification course assignment consists of four different parts.

1. Modelling the 2-axis manipulator
2. Analysis of dynamics
3. PLCopen Motion Control of 2-axis manipulator
4. Position control design and identification

Assignments was challenging and educational. All assignments are done separately and needed Matlab, Simulink and Codesys files are included one compressed file.

System identification with real measured data we noticed that sweep signal outputSWEEP_20_1_2000 and using OE method with polynomials nb=8, nf=7 and delay nk=1we get a model than accuracy is 59%. Comparison between first assignment model and OE model based on sweep excitation signal can be seen that second axis cannot be modelled as a point of mass. And second axis has it own dynamics. This axis had attributes that caused some sort of resonance with the first axis. So that can be for example flexing of the shaft, bearings, machinery that the axis controlled and etc.

Comparing identifications methods and choosing parameters (polynomials) takes time and there may be a better chooses what we found.

References:

[1] Arizona State University.Title: Hooke's Law and Simple Harmonic Motion. Available: https://www.webassign.net/labsgraceperiod/asucolphysmechl1/lab_8/manual.html

[2] Sallam A. Kouritem, Mohammed I. Abouheaf, Nabil Nahas, Mohamed Hassan, A multi-objective optimization design of industrial robot arms, Alexandria Engineering Journal,Volume 61, Issue 12,2022,Pages 12847-12867, ISSN 1110-0168, https://www.sciencedirect.com/science/article/pii/S1110016822004355

[3] Newman, Michael & Lu, Kaiyue & Khoshdarregi, Matt. (2020). Suppression of robot vibrations using input shaping and learning-based structural models. Journal of Intelligent Material Systems and Structures. 32. 1045389X2094716. 10.1177/1045389X20947166. https://www.researchgate.net/publication/343860635_Suppression_of_robot_vibrations_using_input_shaping_and_learning-based_structural_models/citation/download