# Control of a rotating inverted pendulum

Annelouk van Mierlo (4693736) & Yiting Li (5281873)[a]

[a] *University of Technology, Delft",*

**Abstract**

Report for describing the workflow of how to derive a control system for a rotation inverted pendulum using Matlab.

*Keywords:* Pendulum, Matlab, Simulink, System Control

**Contents**

## 1. Introduction

The objective is to design a controller that is able to stabilize the pendulum in the upright position and be able to recover from a small distortion. To achieve this objective the controlled system must have a small error and a fast reaction time.

The format of the report is structured in a similar way to the programming. First the calibration is explained Second the white box identification and the linearization of the system are obtained by the white box. After the white box identification the alternative approach, the black box is introduced and explained. Third the method of obtaining an observer is described. And as a final part, an explanation and results are discussed for two controllers. The report is then concluded with a discussion.

By the end of this project, a stable controller is designed and applied to the setup ROT 4.

## 2. Modelling and Calibration

### *2.1. Modeling*

The schematic drawing of the system is provided in figure 1. With the equations provided, the nonlinear system can be modeled via a differential equation solver in Simulink as shown in figure 2. The detailed equation of the subsystem `x_d = get_theta_dt_dt(u)` is in the code Appendix A. The setup will rotate in opposite direction compared to the provided equations under the same input, for mimicking that, the motor gain $k_m$ in the nonlinear model is assigned as a negative value.

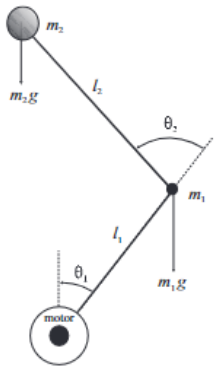


Figure 1: Schematic Drawing of the Rotational Pendulum

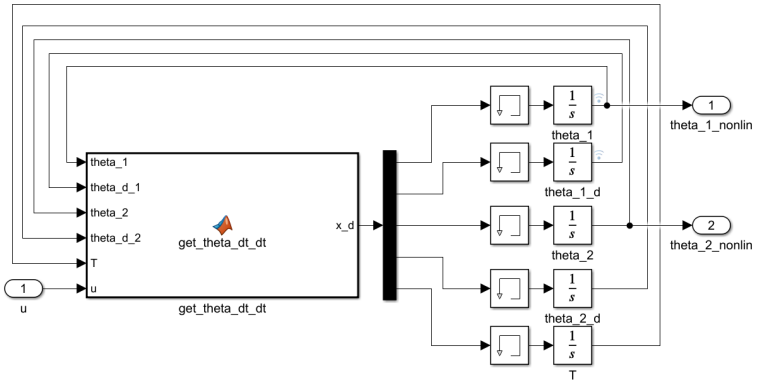


Figure 2: System Modeling via Simulink

### *2.2. Calibration*

The setup measurement is voltage but angle data is requested for control tasks, therefore, calibration is required to determine the relationship between angle and voltage. The relation between the link's angle (radians) and the sensors' output (voltage) is defined by 1, where indexes 1 and 2 are for Beam and Pendulum, respectively. Noticed that a challenge arose in the form of discontinuities or 'gaps' in the voltage measurement, which were attributed to the potentiometer's blind, where measurements cannot be taken.

$$\theta_i = \gamma_1^i (V_i + \gamma_0^i), \quad i = 1, 2 \tag{1}$$

The goal of calibration is to determine the offset and the gain, $\gamma_0^i$ and $\gamma_1^i$. This can be done by checking two positions of the setup. First, measurements $V_1^\pi$ and $V_2^0$ are taken when setup is placed at down-down-position, $[\theta_1, \theta_2] = [\pi, 0]$. Secondly for the up-down-position $[\theta_1, \theta_2] = [0, \pi]$, voltage measurements $V_1^0$ and $V_2^\pi$ are recorded. The offset and the gain can be derived as 2.

$$\gamma_0^1 = -V_1^0, \quad \gamma_0^2 = -V_2^0, \quad \gamma_1^1 = \frac{\pi}{V_1^\pi - V_1^0}, \quad \gamma_1^2 = \frac{\pi}{V_2^\pi - V_2^0} \tag{2}$$

## 3. White box Identification

A set of thirteen initial physical parameters of the model were given. However, eight of these could be incorrect and thus need to be optimized using the white box method. The white box estimation would first record a measurement $y_0$ from the setup and treats it as the reference, then it iteratively optimizes the cost function $J$ and searches for a suitable set of model parameters $\hat{x}$ that gives a smaller error between the estimated output from the nonlinear model $\hat{y}$ and the pre-recorded output from the setup $y_0$ following the given constraints. Once the optimization is finished, it has at least found a local minimum for which the parameters would make the nonlinear model similar to the setup. The cost function is donated as the error between the nonlinear model(shown in 2) and the setup.

To reduce the difficulty and dimension of this estimation problem and also cancel the correlation of both links, the white box estimation is split up into two parts: firstly the estimation of the pendulum parameters and secondly the estimation of the beam parameters.

### *3.1. Estimation of the pendulum parameters*

First, the parameters of the pendulum are estimated, which are the center of mass of the link($c_2$), the inertia of the link ($I_2$), and the dampening of the joint($b_2$). The pre-recorded measurement $y_0$ in this part is a 5-second length $\theta_2$ measurement while the pendulum is performing a free drop. One challenge faced here is that the backlash of the beam disturbed the measurement of the pendulum. To minimize this effect the beam was held still by hand during the measurement.

The cost function $J_1$ is defined as $J_1 = (y_0 - \hat{\theta_2})^2$, the nonlinear optimization solver is chosen as `lsqnonlin()` in MATLAB.

### *3.2. Estimation of the beam parameters*

Secondly, the parameters for the beam had to be estimated. However, since the parameters of the pendulum are now already known after the previous step, parameters related to the pendulum can be considered as a constant. The variables that are given to the cost function estimation are the center of mass of the link($c_1$), the inertia of the link ($I_1$), the dampening of the joint ($b_1$), the motor gain ($k_m$) and the motor electrical time constant($\tau_e$). In this part, the pre-recorded measurement $y_0$ is a 5-second length $\theta_1$ reading while the input $u$ is a ramping signal whose amplitude raises from 0 to 0.5.

The cost function $J_2$ is defined as $J_2 = \sum(y_0 - \hat{\theta_1})^2$, the nonlinear optimization solver is chosen as `fmincon()` in MATLAB.

*3.3. Results and verification*

Based on the above setup, several experiments were conducted for white box estimation, the sampling time is chosen as 1 ms. The white box estimation comparison result is shown in figure 3 and 4. The estimated parameters for the system are shown in table 1, respectively. The nonlinear model output with the estimated parameters is close to the measurement, indicating that after the optimization, the white box estimation finds a set of parameters that are close to the real values, the normalized mean square error(NMSE) for Pendulum and Beam estimation are 2.391% and 0.0032%, respectively.

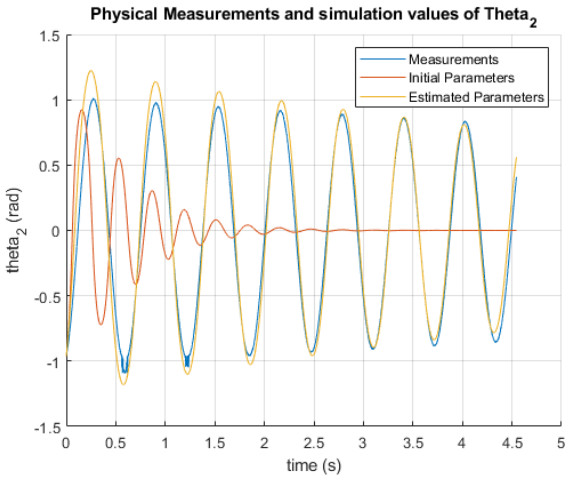


Figure 3: Parameters Comparison of the Pendulum

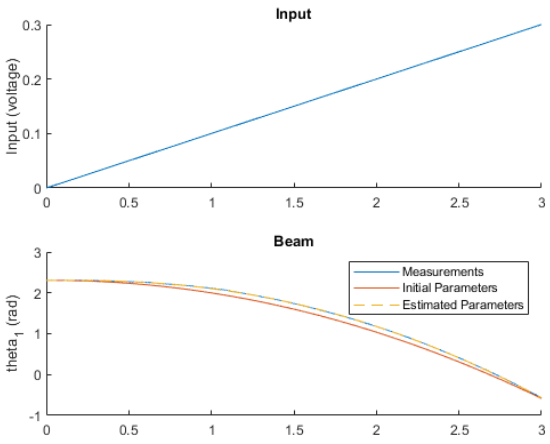


Figure 4: Parameters Comparison of the Beam

| parameter value | $c_1$ | $c_2$ | $I_1$ | $I_2$ | $b_1$ | $b_2$ | $k_m$ | $\tau_e$ |
|---|---|---|---|---|---|---|---|---|
| original value | -0.04 | 0.06 | 0.074 | 1.2e-4 | 4.8 | 2e-4 | 50 | 0.03 |
| estimated value | -7.8e-3 | 0.02 | 0.103 | 9.41e-5 | 5.75 | 2e-5 | 47.45 | 0.36 |

Table 1: Parameters Estimation for the Nonlinear Model

## 4. Linearization

Nonlinear systems are utilized in numerous real-world applications and they can be difficult to analyze and control directly. In order to be able to control these systems a linearization can be applied. This involves taking the first-order Taylor series around an operating point. The goal of the linearization is to get a model in a state space form as shown in equation 3. The system comprises five states in total, and the derivatives of these states are given the names of f(x,u) as shown in the equations 3. Due to the fact that the nonlinear equations are given, the system can be simplified. The functions for $f_1(x,u)$ and $f_3(x,u)$ can be written of functions of the state matrix itself and the function of $f_5(x,u)$ has been given in the provided material and thus can be written down as well. The functions for $f_i(x,u), i \in [1,5]$ can be found by rewriting the equation 4. With the $f_i(x,u)$ the matrix in the state transition function can be defined as shown in equation 5.

Finally, the symbolic toolbox function in Matlab has been used to calculate these derivatives to complete the state space model. This linearized model will only work around the operating point that is given. The accuracy of the linearized model under input $u(t) = 0.1\sin(2\pi t)$ is shown in

figure B.8(Note: the not aligned in left part just for better displaying, those two lines merge), the sampling time is chosen as 1 ms.

The NMSE between the linear model and the nonlinear model is computed under various sine wave input signals to test the linearized model's correctness. The linear model has a high accuracy when both of the links are near the operating point. Therefore, This state-space model can then later be used around multiple equilibrium points for the controllers.

| Amplitude of input | Frequency of input [Hz] | NMSE for $\theta_1$ | NMSE for $\theta_2$ |
|---|---|---|---|
| 0.1 | 1 | 0.05% | 1.98% |
| 0.9 | 1 | 0.46% | 40% |
| 0.1 | 5 | 0.21% | 0.85% |
| 0.9 | 5 | 1.85% | 13.9% |

Table 2: Linearized Model Accuracy Comparison

$$x = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \\ T \end{bmatrix}, \quad \dot{x} = \begin{bmatrix} f_1(x,u) \\ f_2(x,u) \\ f_3(x,u) \\ f_4(x,u) \\ f_5(x,u) \end{bmatrix} = Ax + Bu = f(x,u), \quad y = Cx + Du \tag{3}$$

$$\begin{bmatrix} f_2 \\ f_4 \end{bmatrix} = M^{-1} \begin{bmatrix} T \\ 0 \end{bmatrix} - M^{-1}C \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} - M^{-1}G, \quad f_1 = \dot{\theta}_1, \quad f_3 = \dot{\theta}_2, \quad f_5 = -\frac{1}{\tau_e}T \tag{4}$$

$$A_{5\times5} = \frac{\partial f(x,u)}{\partial x}, B_{5\times1} = \frac{\partial f(x,u)}{\partial u}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \end{bmatrix} \tag{5}$$

## 5. Black box Identification

An alternative to obtaining a linear model that behaves similar as a nonlinear model is using a black box identification. The benefit of this model is that it is not necessary to understand its structure. The model is only locally valid when employing black box estimation, as opposed to a non-linear model estimated using white box estimation, which can be globally valid. Therefore the black-box model can only be used for control tasks around the stable equilibrium.

### *5.1. Input Selection*

To properly excite the system and get enough information for model estimation, the added input signal should fulfill the following requirements.

First, the frequency should not exceed the frequency band of the setup. The `chirp()` command is used to test the maximum frequency of the setup, this resulted in the pendulum stopping for an input frequency that is larger than 30 HZ. Generally, the input signal should cover the whole frequency band, but in the setup, quick shifting of both links caused by high-frequency input will lead to a backlash, which should be avoided.

Second there should be a requirement for the amplitude. Because a two-level signal is insufficient for a nonlinear system a multi-sine signal has been chosen. The maximum amplitude has been

selected in a way where the error between $sin(\theta)$ and $\theta$ is less than 2%. The amplitude should maintain an SNR that is higher than 50 dB.

Based on the above requirements, the input signals, SNR = 64.3896 for the first link and SNR = 50.2 for the second link, have been chosen for the black box estimation and is shown in figure B.9. The corresponding output is shown in figure B.10.

### *5.2. Model Selection*

There is no clear trend in the signal when observing the output data. Therefore the BJ(Box-Jenkins) model is a suitable option. This model works well for the noise that is caused by the non-linearity around the operating point.

By iterating the order combination, the most suitable order that achieves the maximum fitness between experiment data and model for the $\theta_1$ is $[n_b, n_c, n_d, n_f] = [2, 3, 3, 4]$, the accuracy is 92%, as shown in the figure B.11. Increasing the order would be unnecessary since it won't greatly improve the accuracy but will increase the computation complexity.

Unfortunately the BJ model doesn't work well for estimating the pendulum. The alternative model ARMEX will give a better fit. The order for $\theta_2$ is $[n_a, n_b, n_c, n_k] = [6, 5, 5, 7]$, the accuracy 84% as shown in B.12. A possible reason for this could be that the pendulum only has one source of input which comes from the movement of the beam, as a result, both noise and input are applied to the same system. The structure of ARMAX can represent this property since both input and noise terms share the same A polynomial. While in the estimation of the beam, the noise could come from either the beam itself or the effect of the pendulum movement, so the BJ model is suitable because the polynomial for noise and input are different.

### *5.3. Model Validation*

The choice of the model and its order are validated by checking the accuracy of both the pendulum and the beam. In addition to that there are visualisations of the validation that are shown in figure B.13 and B.14.

The average accuracy of the estimated model for Beam is 79.58%, however, the average accuracy of the Pendulum model is 34.53%, which is low for control purposes. Therefore there is no implementation of this black box estimation for a controller.

## 6. Observer

Since not all states can be measured an observer is applied. The observer takes the same input $u(k)$ and output $y(k)$ as the process. Based on the available input/output measurement, the observer then gives an estimation of the states $\hat{x}(k)$ as an output that is given to the controller. The structure of the observer is shown in figure B.15. Pole placement is a technique commonly used for choosing a suitable observer gain while designing observers in control systems, the pole selection is based on the desired performance of the observer.

Generally, the observer should have a short settling time to ensure that the observer responds quickly and accurately to changes in the system being observed because delays in the observer's response can lead to errors or instability in a real-time system. Under this requirement, the chosen poles should be further away from the imaginary axis and have a large observer gain. However, if the observer gain is too large, the observer can amplify the measurement noise and lead to inaccurate estimates.

The poles are chosen as $P = [-100, -40, -60, -40, -20]$ by applying the heuristic method and multiple tests. The observer performance is shown in figure 5. The angle estimation is close to the measurement from the setup, with a NMSE of less than 2e-4 for both the estimation as the measurement. Furthermore, the velocity estimation gives a smoother results. This is due to the fact that the observer can help to filter out the noise in a control system by using feedback to adjust the estimated state of the system based on the difference between the predicted output and the measured output. Figure 5 shows the performance of the observer. The estimated state of the observer can be used for the controllers.

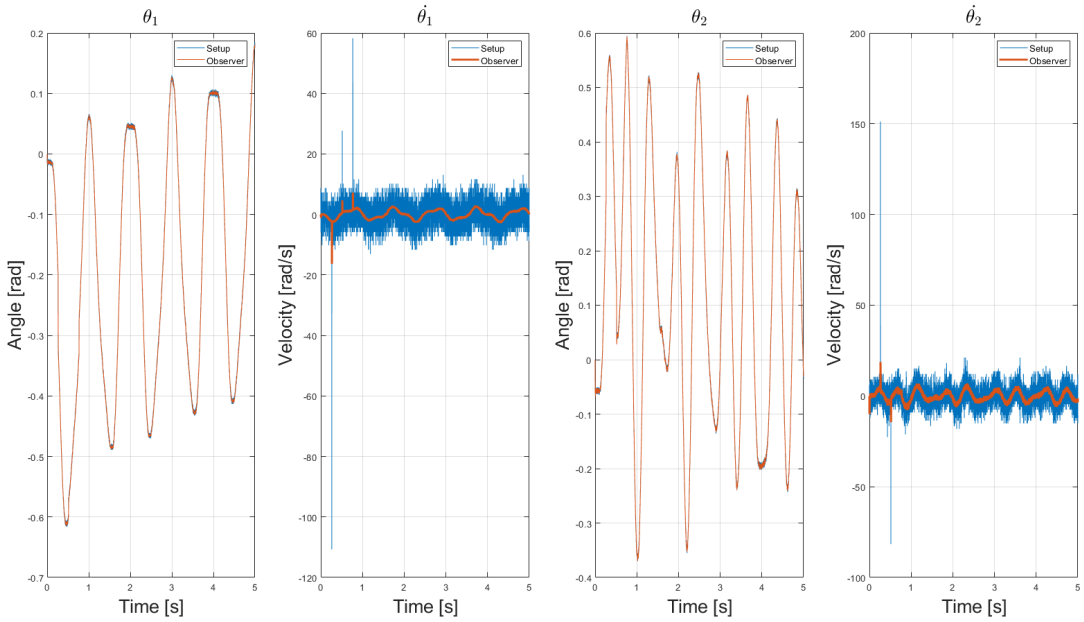


Figure 5: Observer Performance

The influence of the poles' position on the observer performance is shown in table 3. Ts is the settling time for impulse response, and the SNR value is used to evaluate the observer filtering ability for velocity smoothing, it's calculated by taking the observer output as the "pure" signal and the setup output as the noised signal, by calculating the improved SNR, we can know how much noise is filtered out with the observer. Table 3 shows that, with smaller poles value, the settling time of the impulse response is large, and the observer couldn't track the variation of the angle and velocity, leading to a larger NMSE value and a blank improved SNR. As the poles are away from the imaginary axis, the observer tracking ability increases, but the SNR decreases indicating the observer amplifies the noise and causes inaccurate estimations, which should be avoided in the control task.

## 7. Controllers

After having the linearized system and the observer estimate the states, a feedback controller can be used to apply state feedback to the system, the structure of the system is shown in figure B.16. Two control designing methods are tested, LQR and Pole Placement. The methods and results are discussed for both controllers in three equilibrium states, including stable equilibrium (down-down),

| poles value | Ts [s] | NMSE for $\theta_1$ | NMSE for $\theta_2$ | SNR for $\dot{\theta_1}$ [dB] | SNR for $\dot{\theta_2}$ [dB] |
|---|---|---|---|---|---|
| 0.5P | 0.3 | 16% | 56.57% | - | - |
| P | 0.07 | 0.0288% | 0.12% | 27.2 | 10.0 |
| 5P | 0.02 | 0.0075% | 0.013% | 26.3 | 10.1 |
| 10P | 0.007 | 0.0095% | 0.016% | 25.0 | 8.7 |

Table 3: Poles Influence on Observer Performance

and two unstable equilibriums (down-up and up-up). Observer poles for down-down and down-up is set as $P = -[100, 40, 60, 40, 20]$, observer poles for up-up is set as $P = -[70, 65, 60, 55, 50]$.

While both controllers work quite well, an offset of the angle of the beam still exists. To overcome this, an integrator term over $\theta_1$ is added to the output of the state feedback. This integral term is making corrections to the positions of the beam, leading the mean move back to the intended centering during runtime.

### *7.1. LQR controller*

An LQR controller uses quadratic cost functions to find a good control input for the linearized system. The Q-matrix gives a penalization for each state and the R-matrix gives a penalization for the input, these costs are combined into a cost function $J$ as shown in equation 6.

$$J = x^T Q x + u^T R u \tag{6}$$

#### *7.1.1. LQR for different configurations*

As explained in Chapter 6 the observer poles differ for the different configurations(refers to the equilibrium point). For each configuration, the Q and R matrices should be tuned.

Since the Q and R matrices stand for the penalization of states and input, the adjustments of Q and R will be based on the desired system's performance. The detailed Q, R matrix for three configurations are defined in equation 7 and 8. In the Q matrix, five diagonal terms are a penalization term directly to the states $x = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T]$, and elements next to the diagonal of the Q matrix are the cost added for the combination of the position of the beam and the pendulum in the different configurations. Equation 9 donates gain for the integral term for the three configurations.

Firstly for the down-down configuration, in the $Q_{DD}$ costs of the positions of the angles are chosen to be relatively large because the dampening of the pendulum is supposed to be maximized. There is no integral gain for this configuration because this configuration is in equilibrium and the gravitational force is enough to guide the system to the reference. This reason also explains the large penalization of input, because the system is in a stable equilibrium.

Secondly, for the down-up configuration, in the $Q_{DU}$ matrix, the penalization of the beam angle is larger than the one for the pendulum. A possible reason could be: the overshooting of the beam has a large effect on the pendulum, and as a result, it should be penalized. Also, the cost for the input $R_{DU}$ is a larger value than for the down-down configuration because this can minimize the overshoot and oscillations of both links. Without an integral gain $z_{DU}$ the system would not stabilize the pendulum in the reference point but with an offset. For a larger integral gain the system would correct this error faster.

Lastly, for the up-up configuration, the Q matrix is a lot simpler compared to previous cases. this is also due to the difference in the observation poles. The cost for the input is even lower than

that for the down-down, and up-up configurations. This is chosen because it needs more correction to stabilize, this is most likely due to the fact that the center of mass of both the pendulums are further apart than in the down-up configuration.

In figure 6 the angle of the beam and the pendulum are plotted for a real-time simulation for all three configurations. A video demo of these results can be found in Appendix A. In these simulations disturbances that are visible in the figure are added by hand. The figures also show that for the more difficult configuration, up-up, the are more oscillations.

$$\mathrm{Q}_{DD} = \begin{bmatrix} 1100 & 0 & 100 & 0 & 0 \\ 0 & 0.001 & 0 & 0 & 0 \\ 100 & 0 & 1100 & 0 & 0 \\ 0 & 0 & 0 & 0.001 & 0 \\ 0 & 0 & & 0 & 0.01 \end{bmatrix} Q_{DU} = \begin{bmatrix} 300 & 0 & 100 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 100 & 0 & 200 & 0 & 0 \\ 0 & 0 & 0 & 0.001 & 0 \\ 0 & 0 & & 0 & 0.01 \end{bmatrix} Q_{DU} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

$$R_{DD} = \begin{bmatrix} 200 \end{bmatrix} R_{DU} = \begin{bmatrix} 50 \end{bmatrix} R_{UU} = \begin{bmatrix} 10 \end{bmatrix} \tag{8}$$

$$z_{DD} = \begin{bmatrix} 0 \end{bmatrix} z_{DU} = \begin{bmatrix} 0.3 \end{bmatrix} z_{UU} = \begin{bmatrix} 0.3 \end{bmatrix} \tag{9}$$

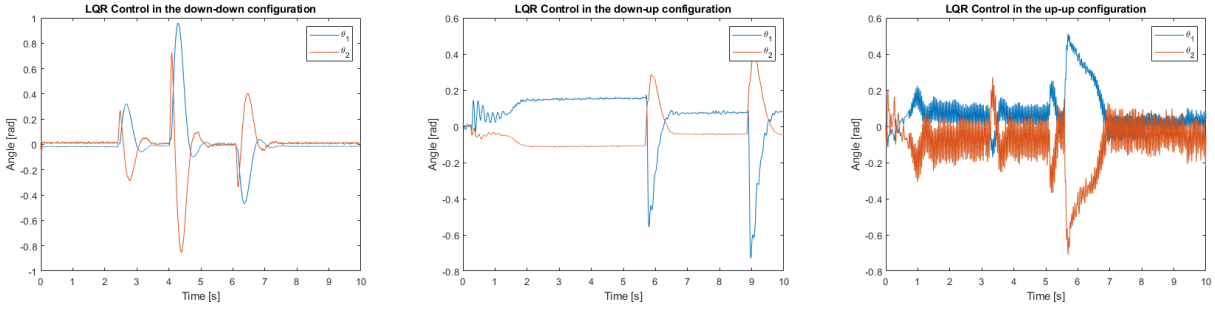


Figure 6: LQR control with disturbances in three configurations

### *7.2. Pole Placement Controller*

Pole placement is a popular control design technique that involves placing the poles of the closed-loop system in a desired location to achieve a desired performance. Pole placement uses a more intuitive tuning than the LQR controller because the poles are rewritten in terms of the eigenfrequency dampening and settling time.

#### *7.2.1. Pole Placement for different configurations*

As a starting point the poles of the LQR controller are extracted for each configuration as shown in equation 14. After that the poles are used in the equations for the eigenfrequency 11, the dampening 12, the torque and the settling time 13.

These rewritten equations give an idea of how to tune the controller for the poles. Since if there is more dampening of the beam required $damp_{beam}$ can be increased. For each configuration, this value is adjusted accordingly and the results of these values and the integrator gain are shown in equation 10. If there was more time the fine-tuning of this controller could have been better. After fine-tuning, the values are written back as poles and places inside the controller gain.

In figure 7 (the enlarged version can be seen via B.18)the angle of the beam and the pendulum are plotted for a real-time simulation for all three configurations. The same video demo of the LQR shows also the results of the pole placement, which can be found in Appendix A. In these simulations disturbances that are visible in the figure are added by hand.

$$\text{Tuning}_{DD} = \begin{bmatrix} \omega_{eig_{beam}} & 22.30 \\ damp_{beam} & -0.962 \\ \omega_{eig_{pend}} & 7.16 \\ damp_{pend} & -0.64 \\ t_{ss} & 0.0098 \\ z_{gain} & 0 \end{bmatrix} Tuning_{DU} = \begin{bmatrix} \omega_{eig_{beam}} & 27.28 \\ damp_{beam} & -3.04 \\ \omega_{eig_{pend}} & 9.16 \\ damp_{pend} & -0.88 \\ t_{ss} & 0.0098 \\ z_{gain} & 0.9 \end{bmatrix} Tuning_{UU} = \begin{bmatrix} \omega_{eig_{beam}} & 20.21 \\ damp_{beam} & -1.21 \\ \omega_{eig_{pend}} & 5.28 \\ damp_{pend} & -1.20 \\ t_{ss} & 0.0098 \\ z_{gain} & 0.5 \end{bmatrix} \tag{10}$$

$$\omega_{eig_{beam}} = \sqrt{beam_1 * beam_2}, \omega_{eig_{pend}} = \sqrt{pend_1 * pend_2} \tag{11}$$

$$damp_{beam} = (beam_1 + beam_2)/(2 * \omega_{eig_{beam}}), damp_{pend} = (pend_1 + pend_2)/(2 * \omega_{eig_{pend}}) \tag{12}$$

$$\tau = -1/T, \quad t_{ss} = 4 * \tau \tag{13}$$

$$\text{Poles}_{DD} = \begin{bmatrix} pend_1 & -4.57 + 5.51i \\ pend_2 & -4.57 + 5.51i \\ beam_1 & -21.45 + 6.09i \\ beam_2 & -21.45 + 6.09i \\ T & -406.94 \end{bmatrix} Poles_{DU} = \begin{bmatrix} pend_1 & -8.02 + 4.43i \\ pend_2 & -8.02 + 4.43i \\ beam_1 & -20.45 \\ beam_2 & -36.38 \\ T & -406.80 \end{bmatrix} Poles_{UU} = \begin{bmatrix} pend_1 & -2.84 \\ pend_2 & -9.82 \\ beam_1 & -12.76 \\ beam_2 & -32.00 \\ T & -406.96 \end{bmatrix} \tag{14}$$

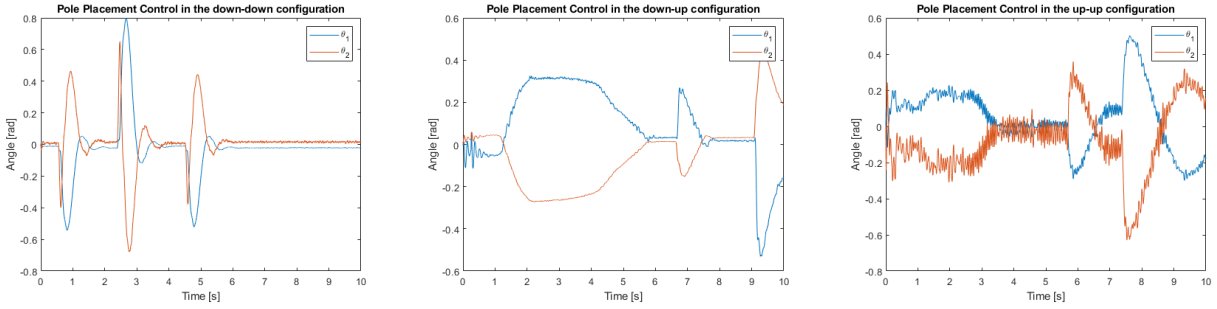


Figure 7: Pole placement control with disturbances in three configurations

### *7.3. Pole placement vs. LQR*

When comparing the performances of the LQR with the Pole-placement as shown in table 4 and in table 5. It shows that the LQR controller is better than the pole placement controller in all cases, the latter has a higher settling time, especially for the up-up case. But the vibration is less violent than the LQR controller, also the LQR controller has a drift effect even corrected by the integrator, while the pole placement controller can better force the system back to reference points.

| | $Down-Down$ | $Down-Up$ | $Up-Up$ |
|---|---|---|---|
| NMSE for $\theta_1$ | 13.6825% | 36.0881% | 10.4194% |
| NMSE for $\theta_2$ | 13.9954% | 16.8933% | 21.6232% |
| Ts for $\theta_1$ [s] | 1.1 | 1.05 | 1.5 |
| Ts for $\theta_2$ [s] | 1.15 | 1.2 | 1.5 |

Table 4: Performance Comparison in each Configuration of the LQR Controller

| | $Down-Down$ | $Down-Up$ | $Up-Up$ |
|---|---|---|---|
| NMSE for $\theta_1$ | 1.5896% | 24.2053% | 21.3746% |
| NMSE for $\theta_2$ | 1.5123% | 15.4734% | 26.6431% |
| Ts for $\theta_1$ [s] | 1.7 | 1.4 | 3.6 |
| Ts for $\theta_2$ [s] | 1.8 | 1.3 | 3.1 |

Table 5: Performance Comparison in each Configuration of the Pole placement Controller

## 8. Discussion

By applying feedback controllers, the rotating double pendulum could be stable at three equilibriums, the system is disturbance rejecting around the operating point. The system's performance on down-down and up-down configurations is better than the performance on up-up configurations, the trajectories are smooth, and the settling time is quite small and satisfying. When the setup is placed in the up-up configuration, despite the good disturbance rejecting ability, two links have some vibrations which reduce the total performance.

To overcome this oscillation, some advanced methods could be used. Applying some nonlinear control and robust control such as sliding mode control and H-infinity control could be a possible option to reduce the vibration. Moreover, some advanced observer design strategies can be applied. Using machine learning and data-driven approaches might also be able to develop more effective and robust control strategies.

To reduce the drift effect to the beam, an integrator is applied to the feedback controller, The easily implemented PID controller only works for the down-down configuration because the tuning is too hard for other configurations, it's a trade-off between easy implementation and easy tuning.

The theory is quite different from the real experiment, controlling a rotating double pendulum involves lots of knowledge and hard-working. We spent lots of time going through the hard-core project, the most important thing is that, after each step, a validation process should be proceeded to prevent building the structure on the wrong base. Both of us had done enough jobs and get a working controller in the end.

## Acknowledgements

Thanks to all people attending this course :), thanks to all the help besides the two of us! Few words but truly sincere.

## Appendix A. Demo Video & Code Link

Click here to see a demo of the working of the controllers
Click here for the code used in the demo
For looking for the nonlinear model function, please refer to the path: `rotating_pendulum/2.NonLinearSystem/NonLinFunc.mat`

## Appendix B. Pictures

### *Appendix B.1. Linearization Comparison*

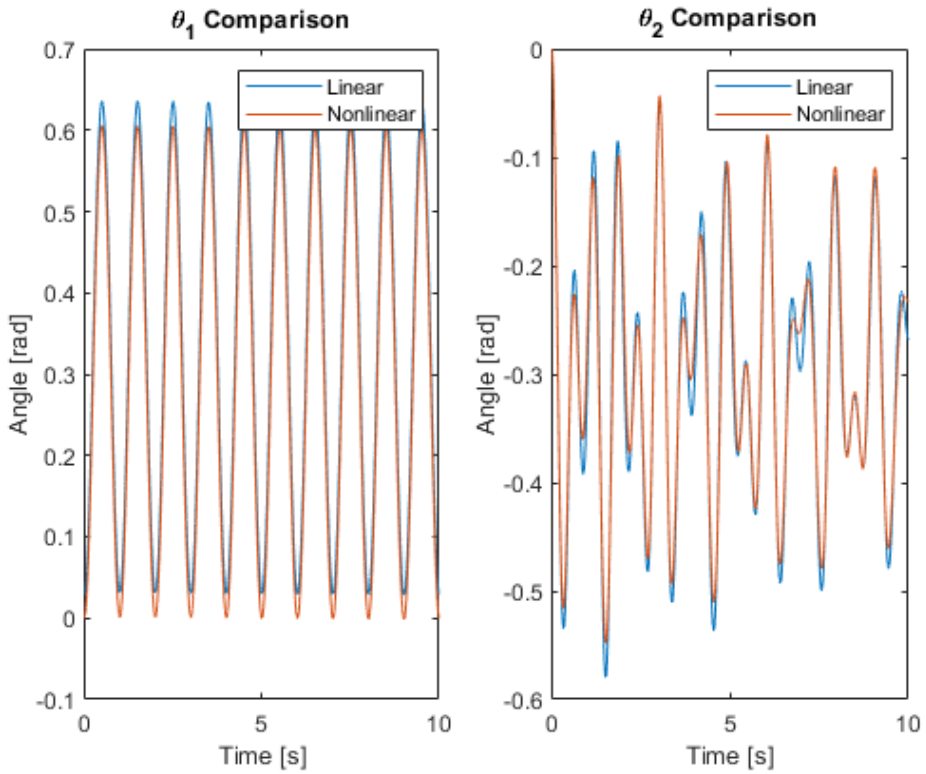


Figure B.8: Comparison between Linearized Model and Nonlinear Model

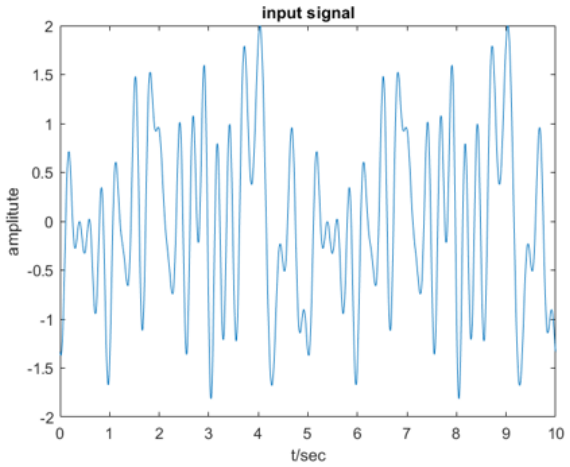


Figure B.9: Input Signal for Black-box Estimation

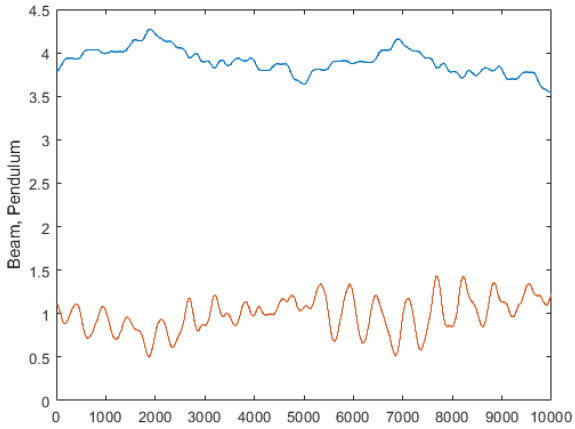


Figure B.10: Output from the Setup

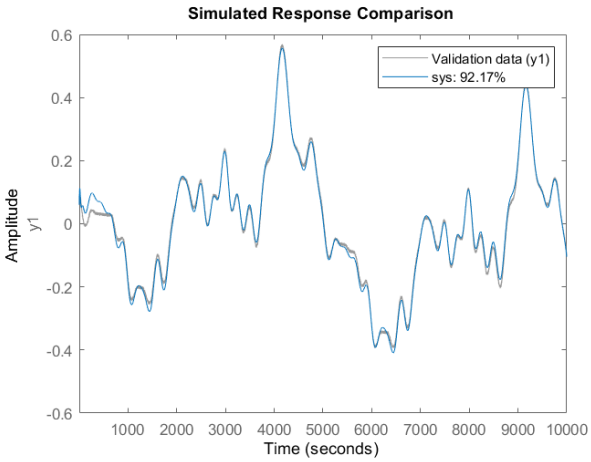


Figure B.11: System Response Comparison for Beam

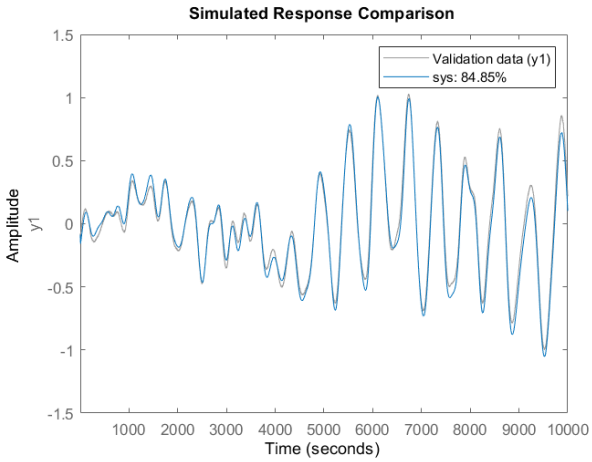


Figure B.12: System Response Comparison for Pendulum

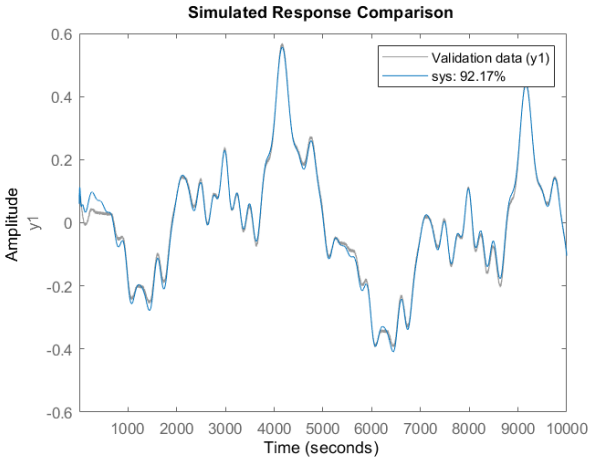


Figure B.13: Validation for Model Estimation of Beam

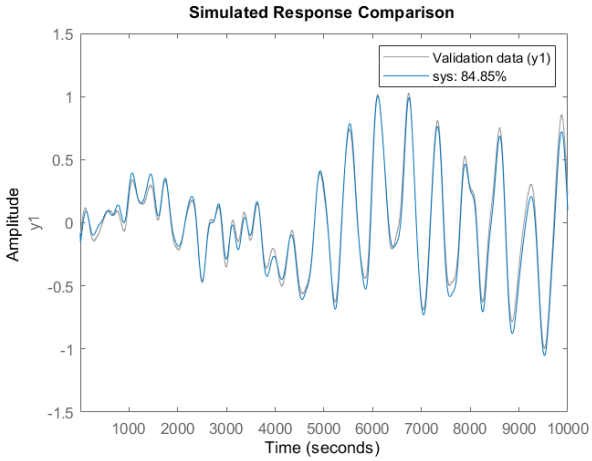


Figure B.14: Validation for Model Estimation of Pendulum

*Appendix B.3. Observer*

Observer structure:

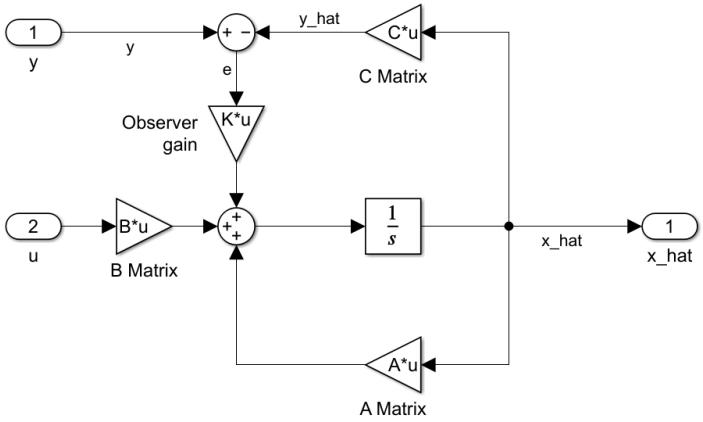


Figure B.15: Observer Structure

*Appendix B.4. Controller*

Controller structure:

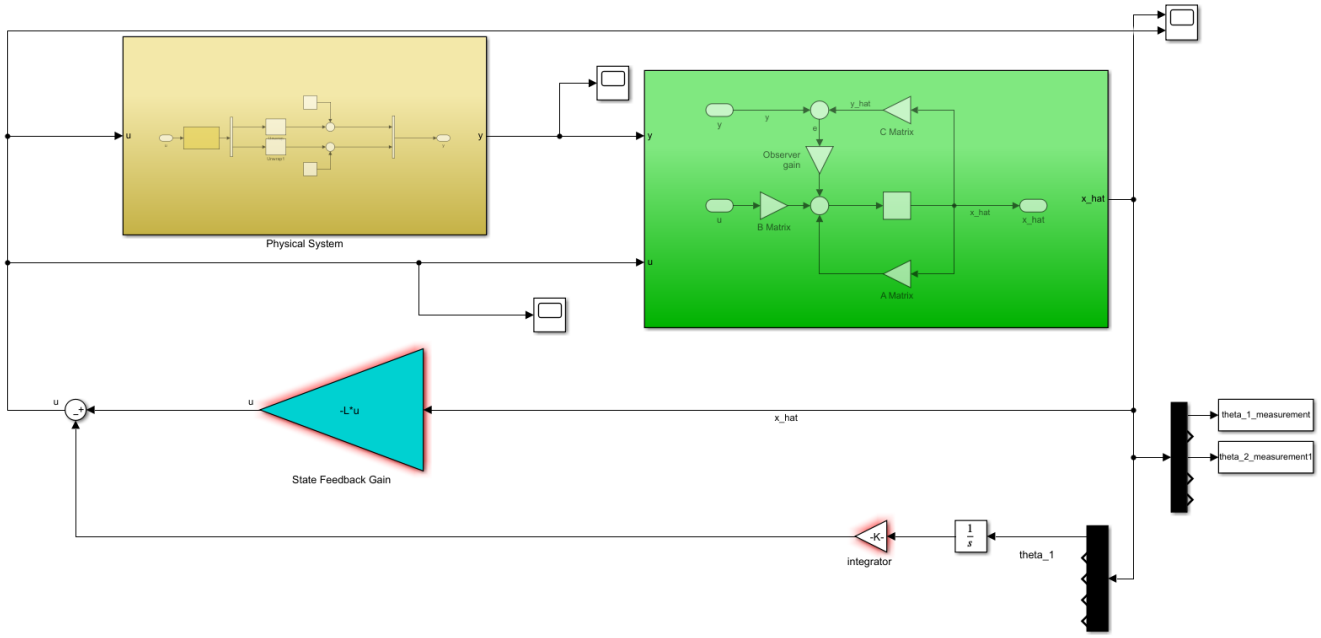


Figure B.16: Controller Structure

Controller performance:

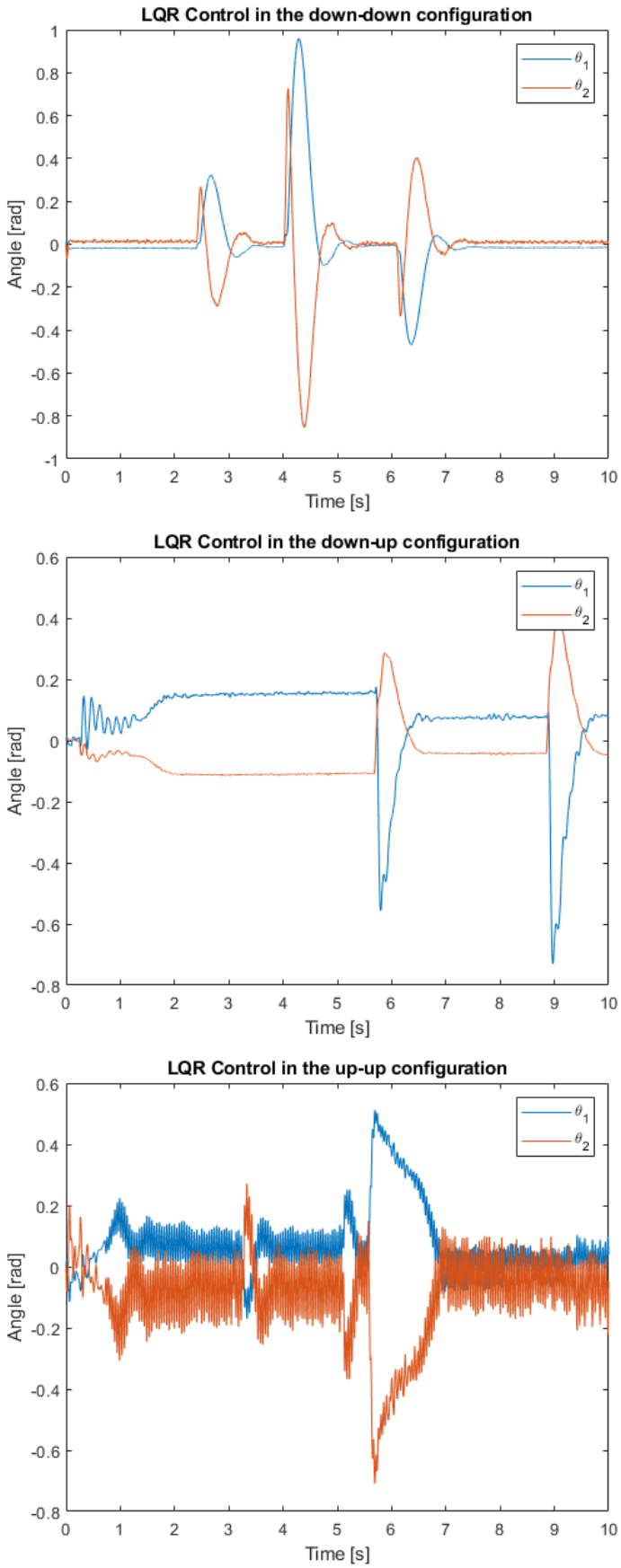


Figure B.17: LQR control with disturbances in three configurations

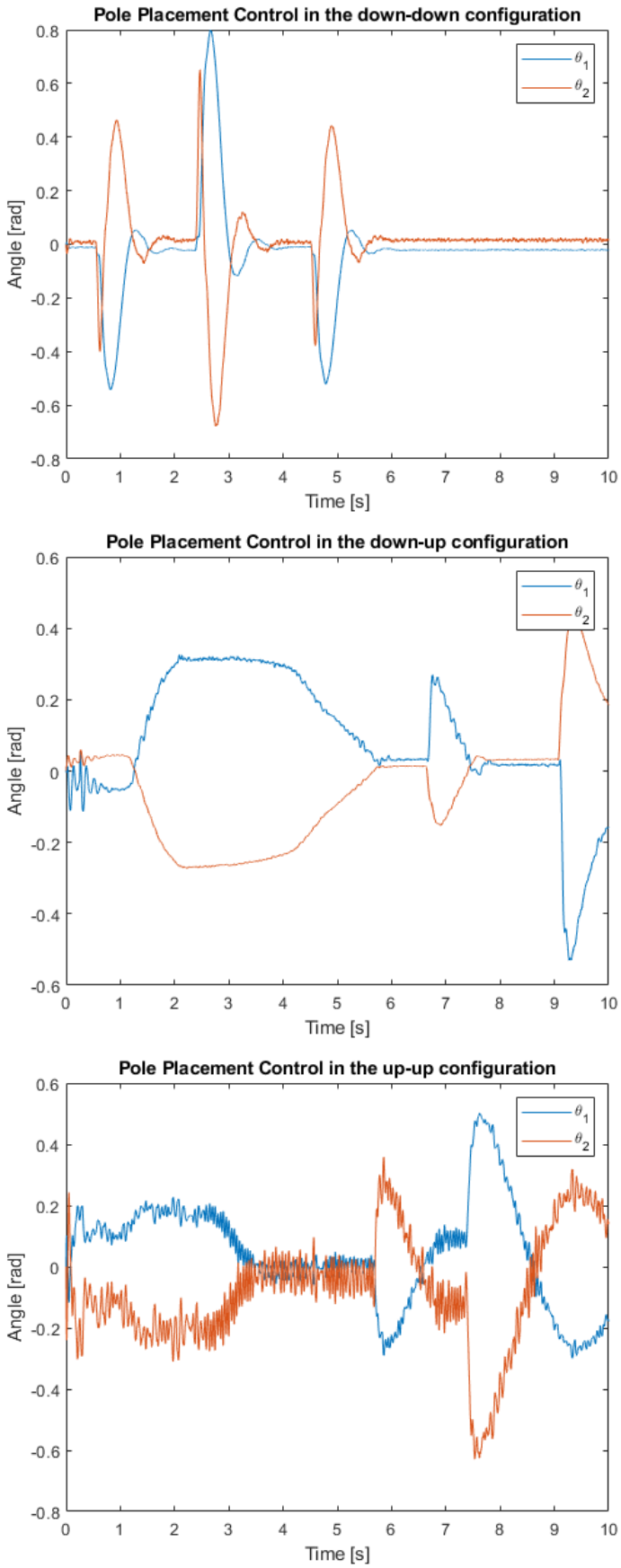


Figure B.18: Pole placement control with disturbances in three configurations