Delft University of Technology

Integration project S&C
SC52035

Qube/Rotary Pendulum

Authors:

Oscar He (4545451)

Wilbert ter Haar (5176476)

October 31, 2022

Contents

1 Introduction

The Qube/Rotary pendulum is a rotational inverted pendulum that is controlled by moving a brushed motor in the arm. This setup creates a challenging control problem since it involves two degrees of freedom with only one control input, one of which is unstable as well. This report will propose two controllers that enable stabilization and rapid reference tracking on the position of the arm with the pendulum in upright position. An extra controller is constructed to automatically swing the pendulum into upright position in a swingup maneuver.

2 Modelling the Qube

2.1 Schematic overview of the Qube

In Figure 1 the schematic overview of the setup is shown. The Qube consists of 2 parts, which is the rotary arm rod that is connected to the motor and the pendulum connected at the end of the rotary arm.
The angle of the rotary arm rod with respect to the middle of the cube is defined as α, the angle of the pendulum with respect to its downward equilibrium is defined as θ. The angles going into clockwise rotation is positive. Using these 2 angles as generalized coordinates, the positions of the Qube can be fully desicribed.
The cylindrical coordinate frame reference is used to describe the movement of the Qube, where e_ρ is a vector that is moving with the rotary arm rod and it is perpendicular to both e_z and the rotary arm rod. Lastly e_z is located in the middle of the Qube.

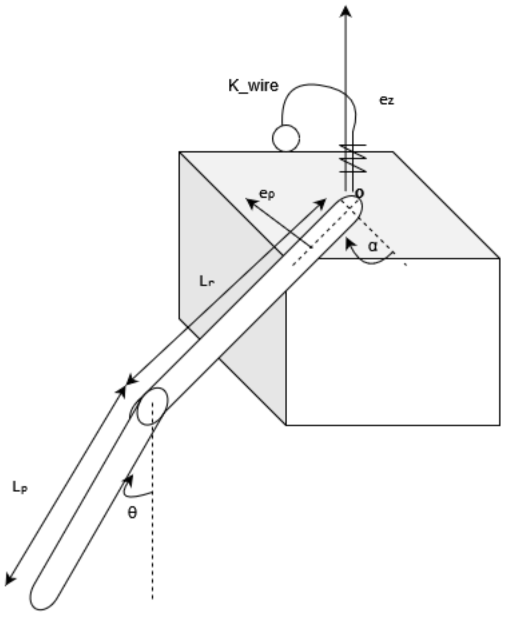


Figure 1: Schematic of the Qube

2.2 Euler-Lagrange method

In order to find the of the Equation of Motions (EoM), the general Euler-Lagrange equation will be used :

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} + \frac{\partial R}{\partial \dot{q}_i} = Q_i \tag{1}$$

Where q_i is the i^{th} generalized coordinate, R the Rayleigh's dissipation function, Q_i the generalized force and L the Lagrangian equation as seen on Eq. 2.

$$L = T - V \tag{2}$$

where T is the total kinetic energy and V is the total potential energy of the system.

2.3 Finding the Equation of Motion of Qube

The total kinetic energy of the system can be described as the sum of the rotational and translation energies of the rotary arm rod and pendulum, which is given as:

$$K = \frac{1}{2} m_p v_p^2 + \frac{1}{2} J_p \dot{\theta}^2 + \frac{1}{2} J_r \dot{\alpha}^2 \tag{3}$$

where m_p, m_r is the masses of the pendulum and rotary arm rod, v_p the pendulum velocity at its Center of Mass (CoM) and J_p, J_r the moment of inertia (MoI) of pendulum and rotary arm rod.

J_r is defined as MoI around the axle where the arm is attached. The pendulum is assumed to be a rod uniform in mass along its length, therefore its CoM lies in the center and its MoI (J_p) equals:

$$J_p = \frac{1}{12} m_p L_p^2 \tag{4}$$

where L_p is the length of the pendulum.

Using the schematic in Figure 1, the velocity at the CoM of the pendulum v_p can be represented as:

$$v_p = (L_r \dot{\alpha} + \frac{1}{2} L_p cos(\theta) \dot{\theta}) e_\rho + (\frac{1}{2} L_p sin(\theta) \dot{\theta}) e_z \tag{5}$$

where L_r is the length of the rotary arm rod.

The total potential energy of the system is the sum of the gravitational energy of the pendulum and the energy in the wire that is connected to the rotary arm that is modeled as a torsion spring. The total potential energy is then given by :

$$V = \frac{1}{2} m_p g L_p (1 - cos(\theta)) + \frac{1}{2} K_{wire} \alpha^2 \tag{6}$$

where g is the gravity constant and K_{wire} the torsion spring constant.

For Rayleigh's dissipation function the sum of the friction only in the rotating pivots are taken into account and is given by :

$$R = \frac{1}{2} C_p \dot{\theta}^2 + \frac{1}{2} C_r \dot{\alpha}^2 \tag{7}$$

where C_p and C_r are the friction coefficients of pendulum and rotary rod.

Lastly for the generalized force there is a torque τ acting on the base of the rotary arm rod, which can be modeled as a simple DC motor with counter-electromotive force (back EMF). The equation for the motor will then be:

$$Q_{alpha} = \tau = K_t \frac{u - K_m \dot{\alpha}}{R_m} \tag{8}$$

where K_t is the motor torque constant, u the input Voltage, K_m the back EMF constant and R_m the motor resistance coefficient.

Finally using the Euler-Lagrange method in Eq. 2 with Eq. 1-8 and the generalized coordinates α, θ the following two EoM is obtained using matlab *syms toolbox* :

$$\begin{gathered} C_r \dot{\alpha} + K_{\mathrm{wirc}}\, \alpha + \ddot{\alpha} \left(m_p L_r^2 + J_r \right) - \frac{L_p L_r m_p \dot{\theta}^2 \sin(\theta)}{2} + \frac{L_p L_r m_p \ddot{\theta} \cos(\theta)}{2} = \frac{K_t \left(u - K_m \dot{\alpha} \right)}{R_m} \\ C_p \dot{\theta} + \left(\frac{L_p^2 m_p}{4} + J_p \right) \ddot{\theta} + \frac{L_p g m_p \sin(\theta)}{2} + \frac{L_p L_r \ddot{\alpha} m_p \cos(\theta)}{2} = 0 \end{gathered} \tag{9}$$

2.4 Linearizing the EoM

The obtained EoM in Eq. 9 are clearly non-linear and has to be linearized around the equilibrium points to be used for linear control methods.

To obtain the equilibrium points, the EoM will be first converted into state space equations and then set the state space equations to 0. Which is shown here:

$$\dot{x} = f(x,u), \quad x = [\alpha, \theta, \dot{\alpha}, \dot{\theta}]^T, \dot{x} = [\dot{\alpha}, \dot{\theta}, \ddot{\alpha}, \ddot{\theta}]^T \rightarrow \dot{x^*} = 0 = f(x^*, u^*) \tag{10}$$

where x^* and u^* are the states and input that puts the state space in equilibrium.

Using *syms* the following upward and downward pendulum equilibrium points were found :

$$\mathrm{x}^*_{down} = \begin{bmatrix} \alpha^* \\ \theta^* \\ \dot{\alpha}^* \\ \dot{\theta}^* \end{bmatrix} = \begin{bmatrix} \frac{K_t u^*}{K_{wire} R_m} \\ 0 \\ 0 \\ 0 \end{bmatrix}, u^*_{down} = u^*, \quad x^*_{up} = \begin{bmatrix} \frac{K_t u^*}{K_{wire} R_m} \\ \pi \\ 0 \\ 0 \end{bmatrix}, u^*_{down} = u^* \tag{11}$$

These equilibria correspond the pendulum hanging in downward position and balanced upside down respectively. The position of the arm α^* is still a free variable and can be chosen. For simplicity both upward and downward equilibrium will be selected with the arm in the center such that $\alpha^* = 0$ and $u^* = 0$.

Linearizing the nonlinear EoM around the downward equilibrium points using first-order Taylor series, the linearized downward EoM was obtained in Eq.12. And the upward linearized EoM is shown in Eq.13.

$$\begin{gathered} \left(C_r + \frac{K_m K_t}{R_m}\right)\dot{\alpha} + \left(m_p L_r^2 + J_r\right)\ddot{\alpha} + \frac{L_p L_r m_p}{2}\ddot{\theta} + \left(-\frac{K_t}{R_m}\right)u + K_{\text{wirc}}\,\alpha = 0 \\ \frac{L_p L_r m_p}{2}\ddot{\alpha} + \frac{L_p g m_p}{2}\theta + C_p\dot{\theta} + \left(\frac{m_p L_p^2}{4} + J_p\right)\ddot{\theta} = 0 \end{gathered} \tag{12}$$

$$\begin{gathered} \left(C_r + \frac{K_m K_t}{R_m}\right)\dot{\alpha} + \left(m_p L_r^2 + J_r\right)\ddot{\alpha} - \frac{L_p L_r m_p}{2}\ddot{\theta} + \left(-\frac{K_t}{R_m}\right)u + K_{\text{wirc}}\,\alpha = 0 \\ -\frac{L_p L_r m_p}{2}\ddot{\alpha} - \frac{L_p g m_p}{2}\theta + C_p\dot{\theta} + \left(\frac{m_p L_p^2}{4} + J_p\right)\ddot{\theta} = 0 \end{gathered} \tag{13}$$

Finally with the linearized EoM, the linear state space matrices representation of downward equilibrium is obtained in Eq. 14 and in Eq. 15 for the upward state space matrices representation. With both of the state space containing in total 10 parameters: L_p, L_r, m_p, J_r, C_p, C_r, R_m, K_t, K_m, K_{wire}.

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cccc|c} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{K_{\text{wire}}\,a_{22}}{\sigma_1} & \frac{a_{13}\,a_{21}}{\sigma_1} & -\frac{a_{11}\,a_{22}}{\sigma_1} & \frac{C_p\,a_{13}}{\sigma_1} & -\frac{a_{14}\,a_{22}}{\sigma_1} \\ \frac{K_{\text{wire}}\,a_{13}}{\sigma_1} & -\frac{a_{12}\,a_{21}}{\sigma_1} & \frac{a_{11}\,a_{13}}{\sigma_1} & -\frac{C_p\,a_{12}}{\sigma_1} & \frac{a_{13}\,a_{14}}{\sigma_1} \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{array}\right] \tag{14}$$

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cccc|c} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{K_{\text{wire}}\,a_{22}}{\sigma_1} & \frac{a_{13}\,a_{21}}{\sigma_1} & -\frac{a_{11}\,a_{22}}{\sigma_1} & -\frac{C_p\,a_{13}}{\sigma_1} & -\frac{a_{14}\,a_{22}}{\sigma_1} \\ -\frac{K_{\text{wire}}\,a_{13}}{\sigma_1} & \frac{a_{12}\,a_{21}}{\sigma_1} & -\frac{a_{11}\,a_{13}}{\sigma_1} & -\frac{C_p\,a_{12}}{\sigma_1} & -\frac{a_{13}\,a_{14}}{\sigma_1} \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{array}\right] \tag{15}$$

where

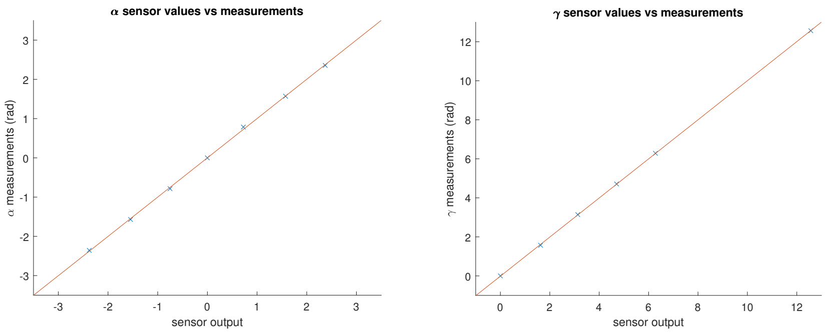
$$a_{11} = C_r + \frac{K_m\, K_t}{R_m}, \quad a_{12} = m_p\, {L_r}^2 + J_r, \quad a_{13} = \frac{L_p\, L_r\, m_p}{2}$$

$$a_{14} = -\frac{K_t}{R_m}, \quad a_{21} = \frac{L_p\, g\, m_p}{2}, \quad a_{22} = \frac{m_p\, {L_p}^2}{3}$$

$$\sigma_1 = a_{12}a_{22} - a_{13}^2$$

3 Sensor calibration

The Qube setups has two sensors, before using these values in any controller it is important to translate the raw sensor output to their real world interpretation. To achieve this, a simple experiment is performed for each sensor. The movable parts of the Qube are placed into various known positions by hand, the sensor output is then compared to real world measurements. The resulting values are plotted in Figure 2.



(a) measurements for α

(b) measurements for γ

Figure 2: Linear regression of sensor output and known locations

Both sensor experiments have nearly 1 to 1 correlation between output signal and measured angle in radians. This result implies that the sensors are already calibrated to accurately detect change in angle. The sensors do not have an inherent zero position and the first sample of each experiment is assumed to be 0, therefore it is important to start each test with the pendulum in equilibrium and the arm in the center of its rotation range.

4 Identification of the Linear system

In this Section the goal is to identify a good enough linear state space model to use it for controlling the real Qube system. To do this the prediction-error method (PEM) will be used, specifically using the linear grey box modeling. The linear grey box modelling method is used since we have the first principles and physical knowledge as a basis already for the model in PEM and this helps us converge to the right linear state space model better. The grey box model estimation will be done with the help of the *greyest* command in Matlab.

4.1 Initial estimates for the parameters

To find a good model, good initial estimates of parameters need to be used. This is because the cost function for PEM is highly non-convex and thus converges only to a local minimum.
Since only L_r and L_p can be measured from the Qube setup in the lab .The rest of initial parameters were obtained in the user manual of Qube servo site [1] and in paper [2].
This results in the following initial parameter values:

Table 1: Parameter values obtained

Parameter values	
Parameter	Value
L_p	$0.120\ m$
L_r	$0.085\ m$
m_p	$0.024\ kg$
J_r	$2.29 \cdot 10^{-4}\ kg \cdot m^2$
C_p	$0.0005\ N{\cdot}m{\cdot}s{\cdot}/rad$
C_r	$0.0015\ N{\cdot}m{\cdot}s{\cdot}/rad$
R_m	$8.4\ \Omega$
K_t	$0.042\ N \cdot m/A$
K_m	$0.042\ V \cdot s/rad$
K_{wire}	$0.01\ N \cdot m/rad$

Although some of the parameter values are from the Qube servo manual [1], it might still be different for our Qube setup in the lab. To verify some the parameters in Table 1, a simple experiment using only pendulum rod is done. First the pendulum rod is held at an angle and then let go while holding the arm rod still and without any input fed into the system. The result is shown in Figure 3.

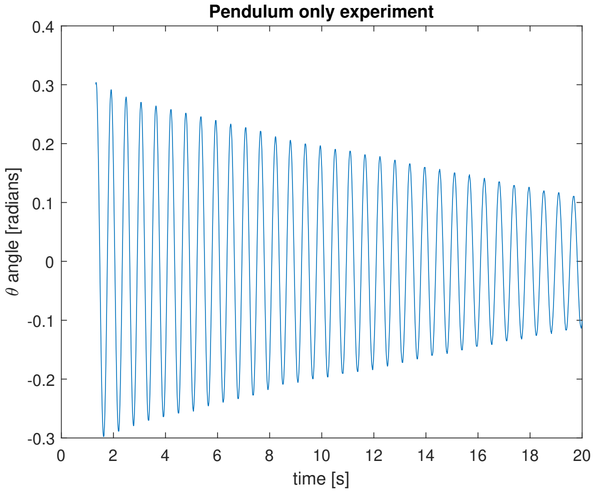


Figure 3: Pendulum only experiment left at a starting angle

Now by making a linear simulated model of the pendulum-only experiment with the previous parameters L_p, m_p, C_p and same initial conditions as the experiment. And then adjusting the 3 parameters to match the experiment data and simulated data, thus improving the values of the parameters.

In Figure 4 the simulated data and experimental data is compared with the new adjusted parameters. The state space model of pendulum-only system is obtained using the same method as in Section 2, the state space model is shown in Appendix 12.1.

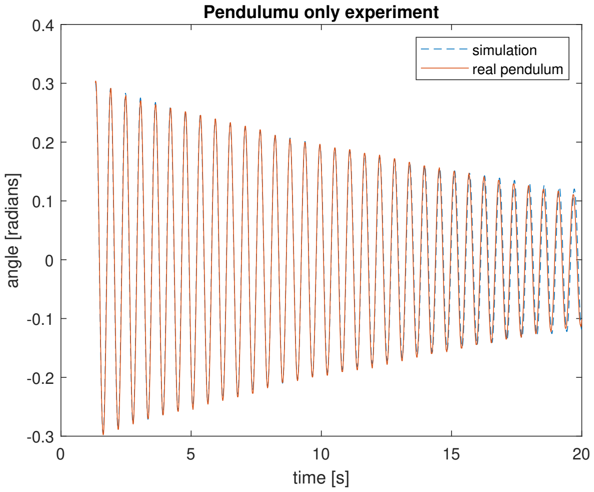


Figure 4: Pendulum-only experiment data compared against simulated pendulum-only model

In Table 2 the new parameters are shown. Since these parameters are found for our setup, they will not be used as a free parameter during the PEM. L_r will also be fixed, since it was measured. This means there are only 6 free parameters left for the PEM, which is: J_r, C_r, R_m, K_t, K_m, K_{wire}.

Table 2: New verified parameters

Parameter values	
Parameter	Value
L_p	$0.123\ m$
C_p	$1.2 \cdot 10^{-5}\ N \cdot m \cdot s \cdot /rad$

4.2 Grey box models

The grey box model used in this Identification section is linearized with the pendulum in the downward position. By doing identification around a stable equilibrium it is possibly to use open-loop identification.

Three difference models are tried during identification, each decreasing in complexity:

- **complex-model**, is Eq. 14 with the 6 free parameters.
- **intermediate-model** which is Eq. 14 with 6 free parameters being a_i values instead and with C_p=0 and $K_{wire} = 0.001$. C_p is 0, because in Tabel 2 it is a very small value. And K_{wire} kept the same value since it gives good results with *greyest*.
- **simple-model** which is shown in Eq. 16. The simple model model has 8 free parameters which is A_i and B_i. The entry A_{34} and A_{44} is 0, because C_p is also 0.

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cccc|c} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ A_{31} & A_{32} & A_{33} & 0 & B_3 \\ A_{41} & A_{42} & A_{43} & 0 & B_4 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{array}\right] \tag{16}$$

Both the simple and intermediate-models uses the same parameters initially such that the entries in the initial state space matrices of all the models are nearly identical at the start before PEM.

4.3 Excitation signals

First the sample rate T_s must be selected to make the input excitation signals. Using the parameters in Section 4.1 and the complex-model, an eigenfrequency of $13.35 rad/s$ was found. Which is about 2.12 Hz and then the sample rate is set to be about 25 times faster than the eigenfrequency. This results sample rate to be $50Hz$.

The excitation signals that will be selected are persistently exciting. There are 3 signals that are used: Pseudorandom binary sequence (PRBS), frequency sweep and doublet.

- PRBS is chosen since it is a signal that mimics white noise, but it has several improvements. To reduce strain on the hardware a minimum hold of two samples is added to the signal, this lowers the maximum sustained input frequency to $12Hz$. The generated PRBS signal is also the same for every experiment and thus repeatable.
- The frequency sweep goes through every frequency between $5Hz$ and $0.5Hz$ sequentially. The disadvantage is that the system has time to build up resonance beyond the linear region when it hits an eigenfrequency and therefore the duration and amplitude has to be lowered.
- The doublet is an impulse, immediately followed by an equal negative impulse. This introduces energy into the system while keeping the equilibrium α in the sample position.

The amplitude of each signal is designed in such a way that the output θ is always below 15° angle, to keep the Qube system in linear behaviour. The 3 excitation signals used in *greyest* are shown in Figure 5.

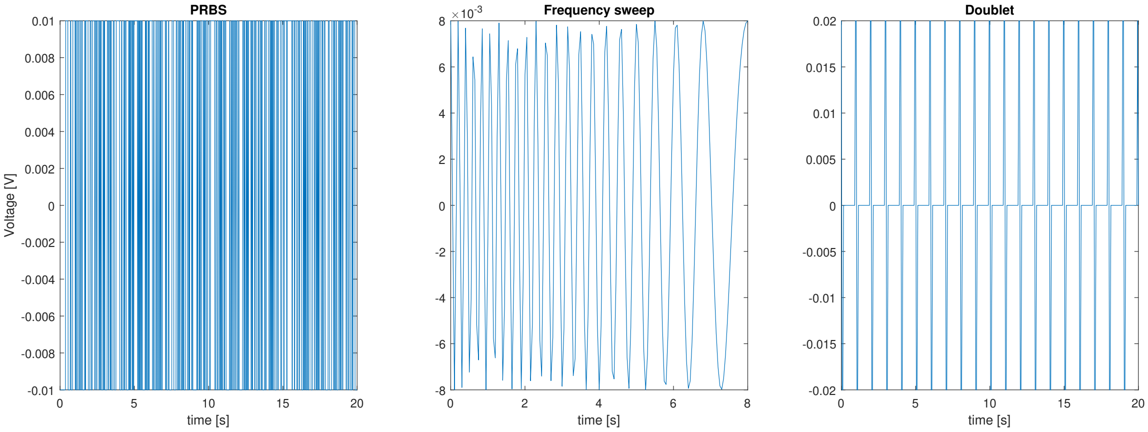


Figure 5: Persistently exciting input signals used for identification in grey box modelling

4.4 Finding the model with best fit

Upon inspection of the experiment data multiple problems are found.

- There is one sample of delay between input and measured output. This is solved by simply shifting the output data one sample forward.
- The measured output angle α does not always oscillate around 0 in all experiments. This can be explained by the very weak and inconsistent spring force from the wire that connects the sensor in combination with dry friction. To mitigate this effect, the mean of the measured α is subtracted from the signal.

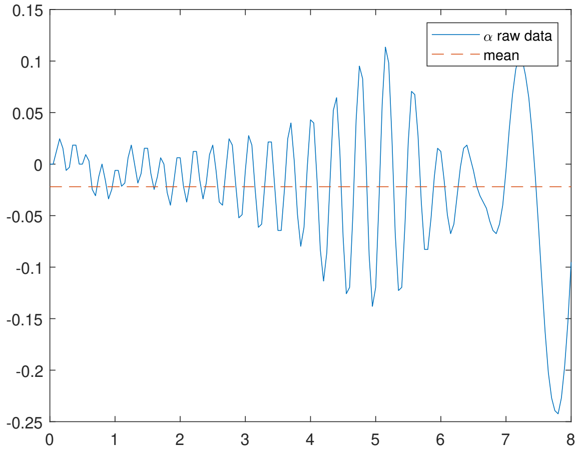


Figure 6: Raw output signal of α compared with its mean, in the frequency sweep experiment

To find the best fit for the model a scoring system will be used, where the score is the average and the standard deviation of the fit percentages of both training set and the validation sets. All three signals will be used, there are always 2 validation sets and 1 training set. In Eq. 17 the fit percentage calculation is shown.

$$\text{fit} \;= \left(1 - \frac{\|y-\hat{y}\|}{\|y - \text{mean}(y)\|}\right) \cdot 100\% \tag{17}$$

where y is the output data and $\hat{y}$ is the simulated output.

In Table 3 the scores for the different grey models with the corresponding training set signals. The simple-model has the highest score for all 3 training signals compared to the intermediate and complex models. In the end it was chosen for the simple model with PRBS training signal, since it was amongst the highest fit ratings, while having a lower standard deviation compared to the other training signal scores and thus more consistent fit result. The resulting state-space matrices that will be used for control upside-down pendulum is then in Eq. 18.

Table 3: Scores

Score values: mean ± standard deviation			
	PRBS	Freq. sweep	Doublet
Complex-model	74.96 ± 6.82	48.95 ± 4.21	69.66 ± 6.18
Intermediate-model	75.60 ± 7.97	73.47 ± 12.98	72.84 ± 8.01
Simple-model	78.92 ± 6.28	79.10 ± 12.11	79.11 ± 12.99

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cccc|c} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -84.16 & 114.8 & -7.017 & 0 & 1497 \\ -93.89 & 250.1 & -6.417 & 0 & 1635 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{array}\right] \tag{18}$$

5 Control objectives

From the site SC52035 info [3] it shows that the required control objectives are:

- The rotary arm angle α should follow a specified trajectory while the pendulum is stabilized in upright position
- The tracked reference should have zero steady state error
- Disturbance rejection from a small tick against the link.

6 Controller choices

There are 2 controllers that will be used in the following sections:

- LQR (Linear Quadratic Regulator)
- MPC (Model predictive controller)

The LQR controller, is chosen for its simplicity and intuitiveness in setting the weights on the states and input for determining the closed loop responses of the Qube.

The MPC is chosen for its ability to handle constraints on the states, which the Qube does have physical constraint on the angle α, the input Voltage and lastly on the angle θ to stay within the linear domain $|\theta| < 15°$.

7 LQR controller

7.1 Observer

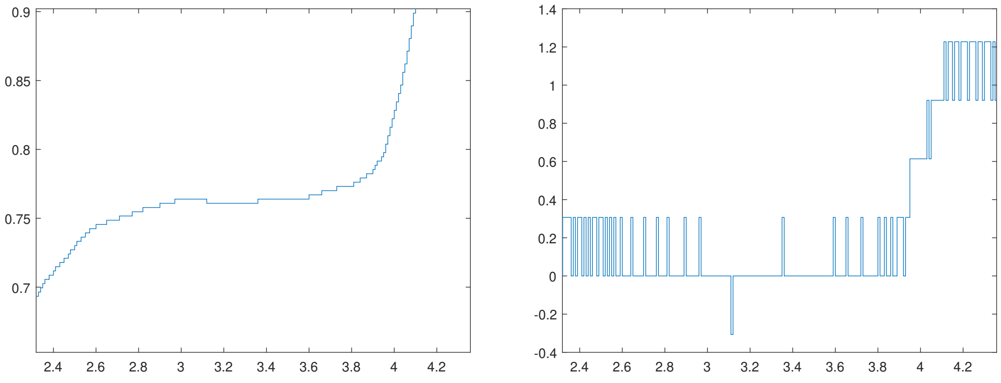
The sensors only detect the angles of the arm, whereas the model also has velocities in it. This could be solved by taking the derivative of the angles, however that introduces a large amount of noise and also some delay if filters are used. A discrete time Luenberger observer will be used instead as it is less susceptible to these problems.

The estimation error of the observer has the dynamics as shown below.

$$e(k+1) = (A - LC)e(k) \tag{19}$$

Using pole placement the poles of $(A - LC)$ can be picked to achieve arbitrarily fast error elimination, however real world limitations complicate this in practise.

The Qube setup uses two identical encoders for its α and θ measurements. Data from experiments shows that the sensors have very little noise and some slight inaccuracy due to quantization (Figure 8a). Since the noise is low, the poles for α and θ are chosen to be very fast.



(a) Arm angle α

(b) Derivative of arm angle $\dot{\alpha}$

Figure 7: Zoomed in plot of measured α and its naive derivative.

The derivatives $\dot{\alpha}$ and $\dot{\theta}$ are more complicated due to the noise that the derivative action introduces as seen in 8b.

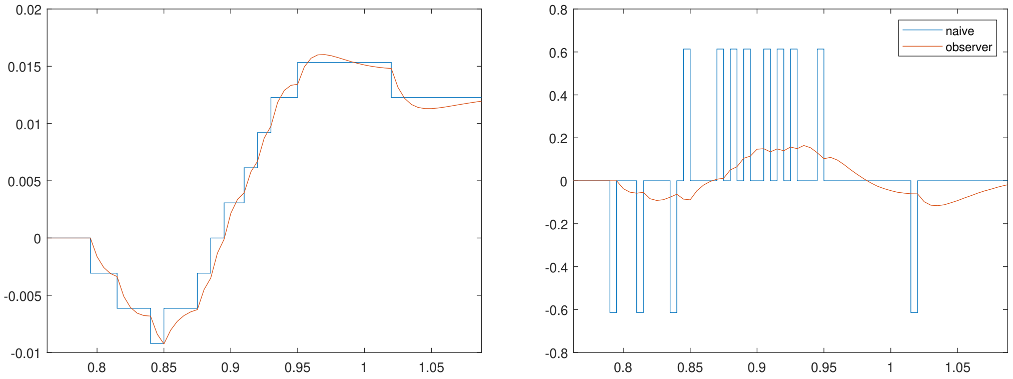
Based on this image the time constant of the observer poles for $\dot{\alpha}$ and $\dot{\theta}$ is chosen to be 4 samples, which is 0.02 seconds which correspond to poles at -50 in the continuous domain.

These choices result in the following continuous and discrete time poles for the observer:

Table 4: Observer poles

Parameter values		
State	Continuous pole	discrete pole
α	-80	0.6703
θ	-81	0.6670
$\dot{\alpha}$	-50	0.7408
$\dot{\theta}$	-51	0.7371

Observer verification



(a) Arm angle measurements and observer. α

(b) Derivative of arm angle $\dot{\alpha}$

Figure 8: Zoomed in plot of measured measured and observed α and $\dot{\alpha}$

7.2 Controller design

First a diagram of the full-state feedback LQR controller with the plant is shown in Figure 9. In the diagram there is also a feed-forward gain K_r that makes sure that the angle α tracks the reference r. This gain is obtained from the $dcgain()$ matlab and the gain is calculated by $K_r = \frac{1}{dcgain(\alpha)}$. The gain K is the LQR gain and the system is the plant QUBE.

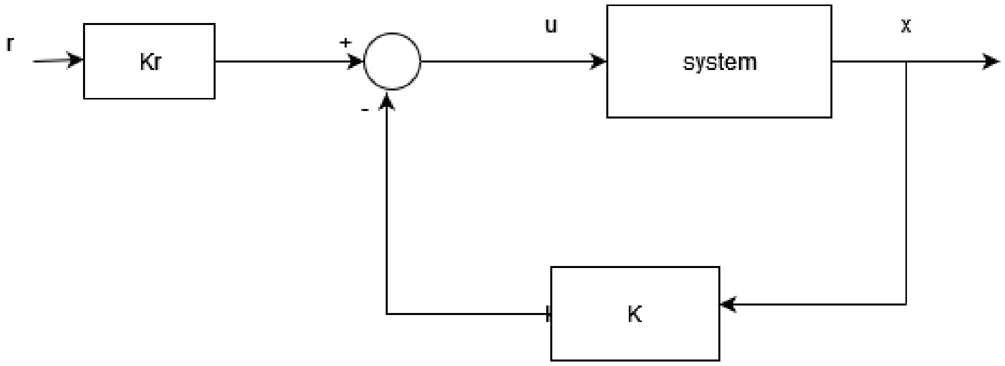


Figure 9: Full-state feedback LQR for reference tracking

From the full-state feedback LQR different weight on the states has been tested to see the effect it has on the step-response, as seen on Figure 10. The weights represented are the diagonal elements on the weight matrix Q_i of the states. The input weight R_i is 1 for all the tests, since it has very little effect on the step response compared with the weights on the states.

From the step responses it can be seen that the weight on the angle α has the most effect on the settling time compared to the other weights. This means that the step responses can be tuned by only tuning the weight on angle α. For the final weights Q_{final} is chosen , since going beyond 30 on state α has almost negligible improvement on the settling time with a higher input and higher θ angle that might destabilise the pendulum.

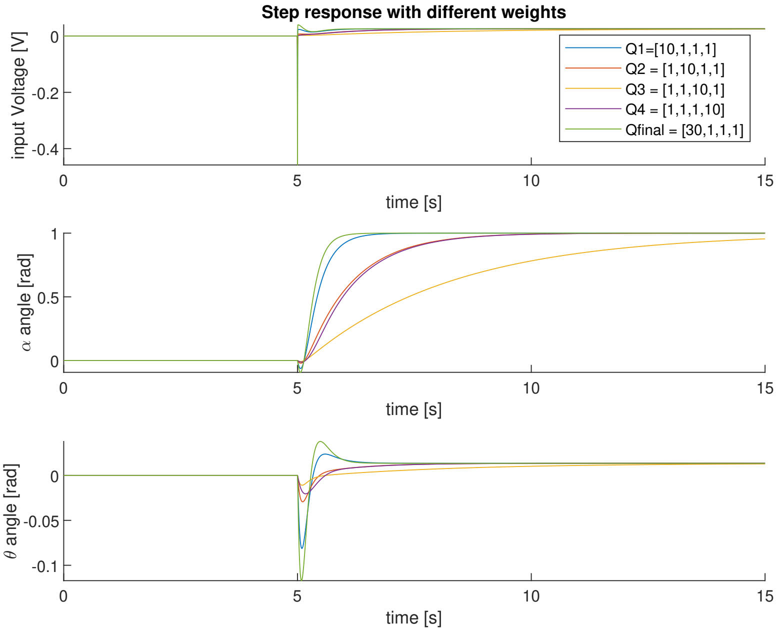


Figure 10: Step responses with different weights using full-state feedback LQR and all input weight $R = 1$

In Figure 11 the step response is simulated using the weight Q_{final} with $R = 1$, but now with the Luenberger observer designed in Section 7.1.
It can be seen that the non-linear model and the real simulation has nearly the same response behaviours, meaning the non-linear modeled correctly.
Also both the nonlinear and real simulation can't track the reference angle α perfectly, which is expected since it is a feed-forward gain for the linear model that does not have perfect fit on the input-output data.

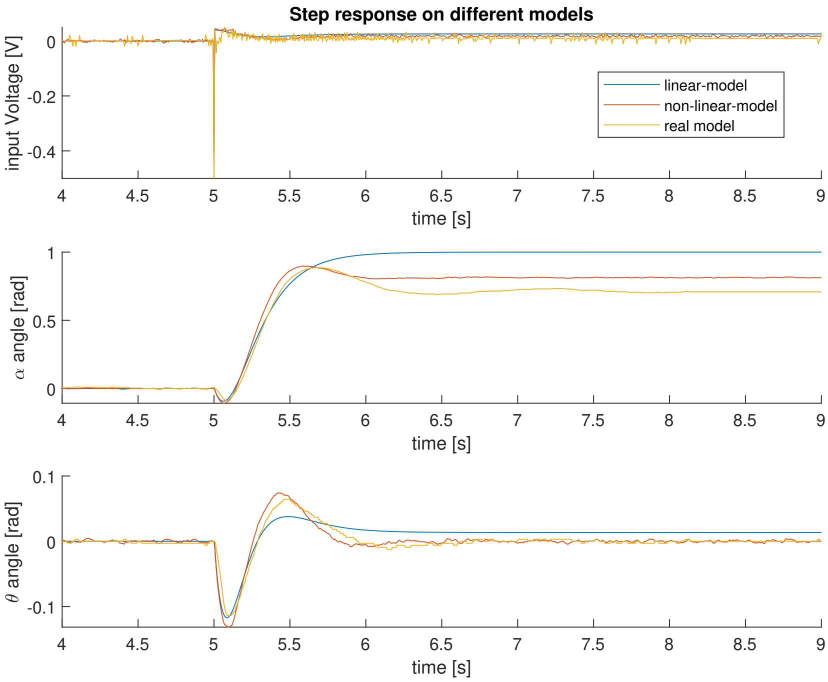


Figure 11: Comparison of step response with the final designed weights

To fix the steady-state error on the real setup, the Linear Quadratic Integrator (LQI) can be used. The LQI diagram used for our setup is shown on Figure 12. Here the gain K is the same LQR gain, the gain K_e is the gain for the integrated error e of the reference and output.
Since the only angle α has the tracking error, only the α error is integrated and multiplied with K_e. It can be seen from Figure 13 that increasing the gain K_e increases the settling time but also introduces oscillation. The gain $K_e = 0.5$ has been chosen for its acceptable settling time while having almost no oscillation.

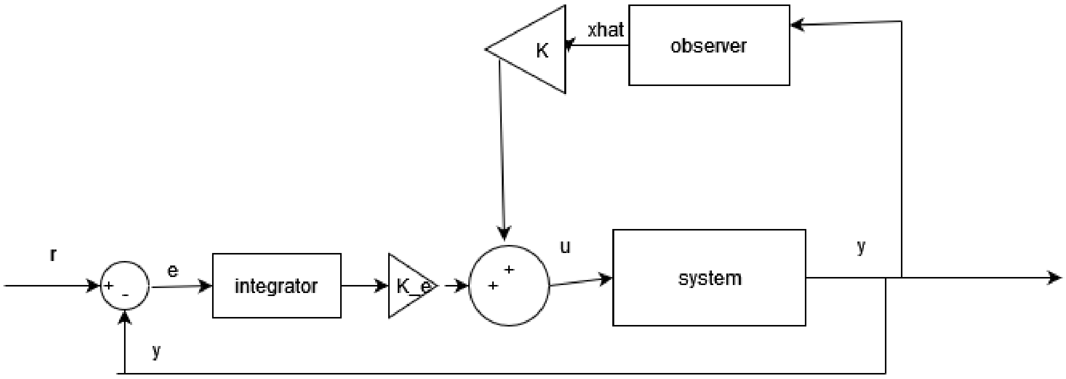


Figure 12: Diagram for the LQI control

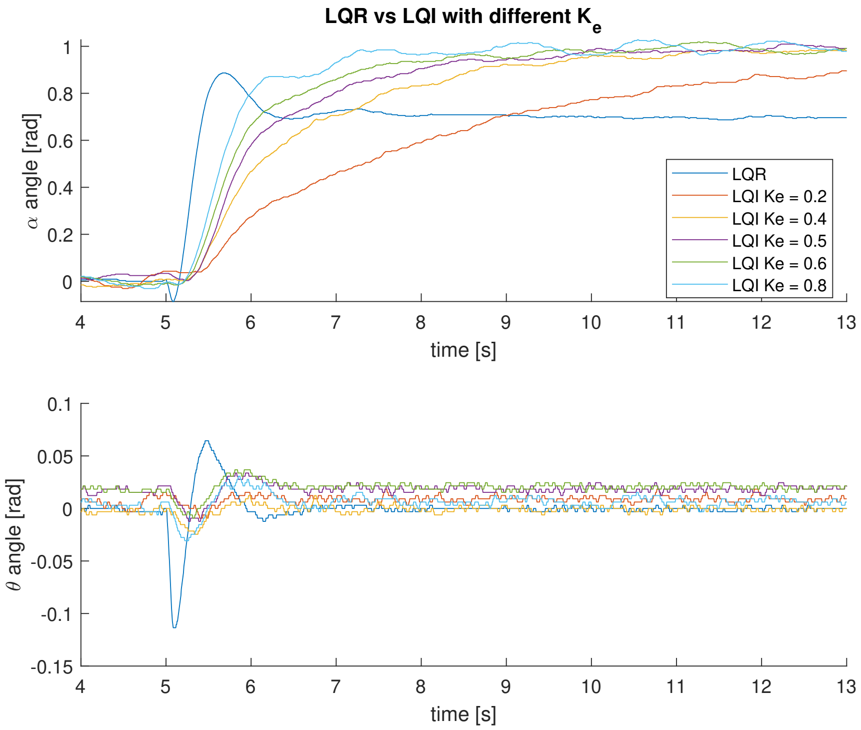


Figure 13: Comparison of the LQR controller with LQI controller with different K_e values

7.3 Tick disturbance rejection

Since a tick cannot be formulated in the system via disturbance observer, it has been decided to let the feed-back gain to fully reject the tick disturbance. For that it has been tested that the faster the sampling rate, the faster the system can react to the tick disturbance and thus rejecting it. The sampling time h is then reduced to 0.005 seconds to be more robust against the tick disturbance.

In Figure 14, 2 ticks has been introduced while tracking the references $\alpha = \theta = 0$ using the LQI controller and still maintaining the pendulum upright. Note that wobbliness of α is because of the integrator.

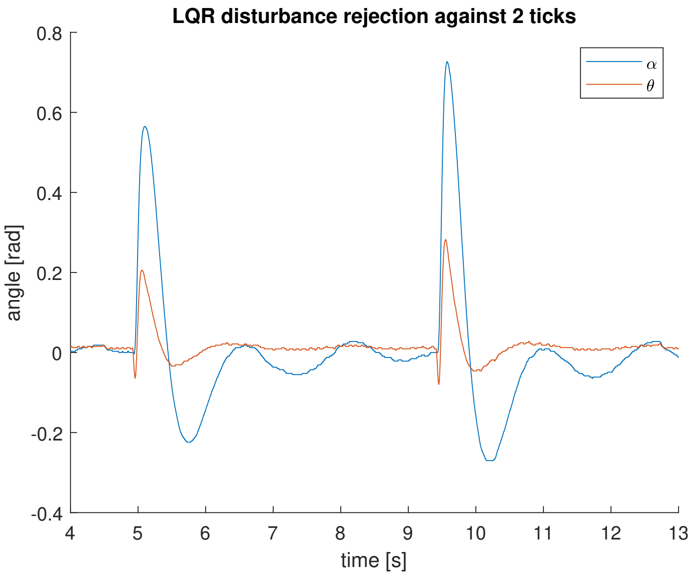


Figure 14: Tracking $\alpha = \theta = 0$ with 2 tick disturbances

8 MPC controller

In this section the MPC controller will be designed and coupled with the Luenberger observer designed in Section 7.1.

8.1 Designing the MPC controller

The MPC controller will be designed with the *mpcDesigner* in Matlab.
First the sampling time h is kept at 0.005 seconds, since the LQR can reject the tick disturbance well with the sampling time.

For the prediction horizon horizon, it should be based on the slowest response time of the closed loop system which is α. Using the matlab *settlingtime*() on the final LQR weight in Figure 10, the Risetime and Settlingtime of α is 0.45 seconds and 0.92 seconds respectively. The prediction horizon N is then selected to be 0.7 seconds in the future which results a horizon of $N = 140$. To ease the computation of the MPC optimizer, the control horizon is set to 20 horizons.

For the constraints, the input Voltage is constrained between $V = [-1, 1]$. The angle $\alpha = [-\frac{3\pi}{4}, \frac{3\pi}{4}]$, since α cannot physically move beyond those limits. And lastly the angle $\theta = [-\frac{10\pi}{180}, -\frac{10\pi}{180}]$, since going beyond those angles the system cannot be approximated by the linearized system.

For the weights, the weight on the states is the same as Q_{final} in Section 7.2 and for the input weight $R = 1$.

In Figure 15, the comparison of the MPC controller with the different models and the real setup. It can be seen that in the real setup the reference tracking performance is worse compared to the other model, with a lot of oscillations around the references.
Different horizons and weight has been chosen, but all of them still exhibit oscillations around the references similar to in Figure 15. We suspect that this is due to the integrator in the MPC controller for α, which is causing the oscillating behaviour.

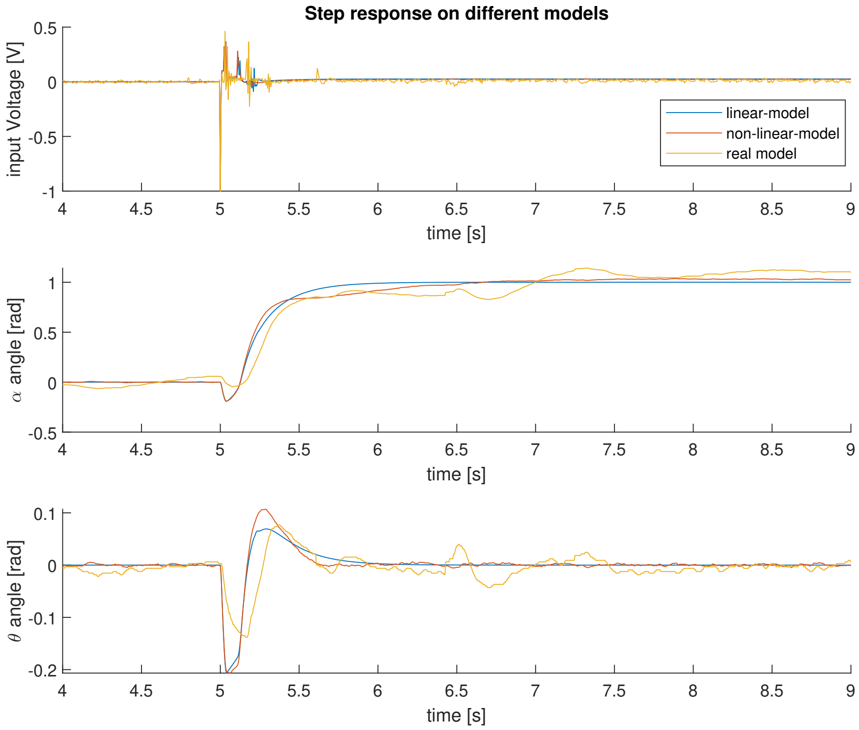


Figure 15: Step response on different models

8.2 Tick disturbance rejection

Using the same MPC controller made in previous section, 2 ticks has been introduced to the system while tracking $\alpha = \theta = 0$. In Figure 16 the responses to the 2 tick disturbances is shown. Although the reference is twitching, the controller can still reject the tick disturbances.

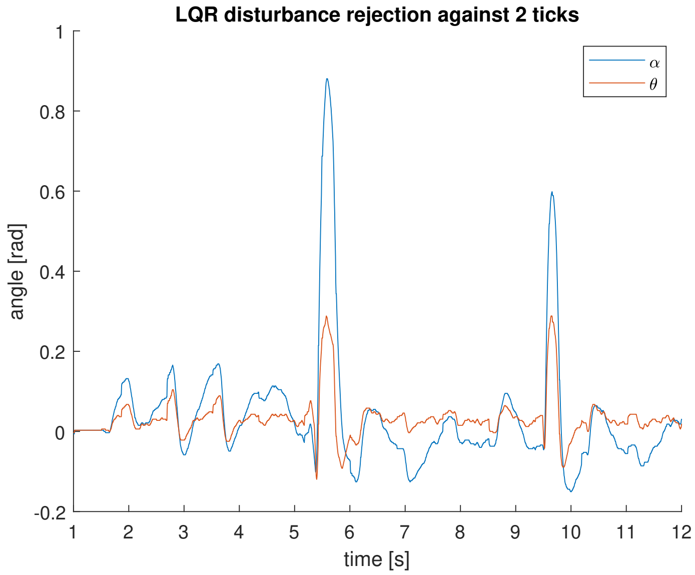


Figure 16: tracking $\alpha = \theta = 0$ with 2 tick disturbances

9 Swingup

A third controller is designed to get the pendulum from downward position into upward position.

9.1 Nonlinear model

The original non-linear state space is no longer good enough since it doesn't contain any centrifugal and gyroscopic effects when the pendulum has a high velocity at a 90° angle.

This can be resolved by treating the pendulum as a set of particles and integrating the kinetic energy of each particle over its length.

$$v_p = \begin{bmatrix} L_r\dot{\alpha} + r\cos(\theta)\dot{\theta} \\ r\sin(\theta)\dot{\theta} \\ r\sin(\theta)\dot{\alpha} \end{bmatrix} \tag{20}$$

$$T_p = \int_0^{L_p} \frac{1}{2}\frac{m_p}{L_p}(v_p^T v_p)dr \tag{21}$$

The equations for potential energy and kinetic energy of the arm stay the same. The steps in section modeling are repeated to find a new state space that includes the full pendulum dynamics.

9.2 Nonlinear Observer

Since the observer no longer only has to operate in the linear portion of the state space a new non-linear observer is required. To achieve this the prediction portion of the linear observer is replaced by a nonlinear predictor of the setup. After this the gains are copied over from the linear observer to create the non-linear observer.

9.3 Hybrid controller

There are already two optimized linear controllers for reference tracking around the upright equilibrium. The new combined controller will reuse the LQR controller to perform reference tracking while in upward position, and then switch into the swingup controller whenever the pendulum is not near the upward equilibrium.

The criteria to switch to the swingup controller is chosen to be when the pendulum is more than 20deg away from its equilibrium.

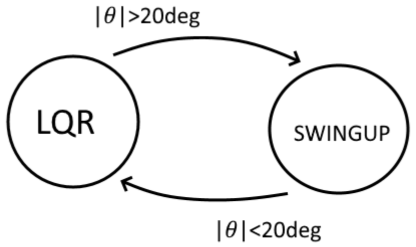


Figure 17: Controller states in a hybrid automaton

9.4 Nonlinear MPC swingup controller

The swingup controller is chosen to be a non-linear MPC controller. Non-linear MPC was chosen because the relatively tight constraints on α are very hard to deal with using proportional controllers or other hand-crafted algorithms. However, Non-linear MPC brings several problems of its own because MPC usually makes strong use of linearity to solve the optimization problem. Some of these issues are slightly less problematic because this controller is not designed to perform reference tracking or stabilization, but its objective is to perform a swingup instead.

There are four controller objectives:

- Increase pendulum energy until is has the right amount of energy to stand up right.
- Minimize input use.
- Minimize use of high frequency inputs.
- Keep the arm angle near the center.

There are also two constraints:

- The arm angle can not go out of bounds.
- The input u is bounded.

From these objectives and constraints, the following cost function for the swing up control is used:

$$\begin{aligned} \min_{\mathbf{u}\in U} \quad & \sum_{k=1}^{n} \left[C_e|E(x_k) - E_r| + \alpha_k^2 Q + u_k^2 R\right] + R_\Delta(u_0 - u_{last}) \\ s.t. \quad & x_{k+1} = f(x_k, u_k) \\ & |\alpha_i| \leq \alpha_{max} \end{aligned} \tag{22}$$

The energy function $E(x)$ is the sum of potential energy and all kinetic energy that can be converted into potential energy when system moves. Not all kinetic energy in the pendulum can be converted into potential energy. Imagine a situation where the arm spins with constant velocity with the pendulum in downward position, the pendulum has kinetic energy but will never swing upward on its own. Therefore an attempt is made to only count kinetic energy that is part of the free swinging mode of the system where the pendulum and arm oscillate in opposite direction. This energy is approximated by calculating the kinetic energy from the perspective of the moving reference frame that follows the shared center of mass of the arm and pendulum, the angular velocity of this frame is approximated by:

$$\dot{\phi} = \frac{\dot{\alpha} J_r + (\dot{\alpha} + \dot{\theta}\cos(\theta)\frac{1}{2}L_p) m_p L_r^2}{m_p L_r^2 + J_r} \tag{23}$$

The energy is now approximated by:

$$\begin{aligned}\dot{\alpha}_\phi &= \dot{\alpha} - \dot{\phi} \\ E(\mathbf{x}) &= \frac{1}{2}J_r\dot{\alpha}_\phi^2 \\ &+ \frac{1}{2}m_p(\dot{\alpha}_\phi L_r + \cos(\theta)\dot{\theta}\frac{1}{2}L_p)^2 \\ &+ \frac{1}{2}m_p(\sin(\theta)\dot{\theta}\frac{1}{1}L_p)^2 \\ &+ \frac{1}{2}J_p\dot{\theta}^2 \\ &+ m_p\frac{1}{2}L_p g(1-\cos(\theta))\end{aligned} \tag{24}$$

After tuning the following tuning parameters are found:

Description	Symbol	Value
Cost on input	R	0.004
Cost on change of input	R_Δ	1
Cost to keep the arm in the middle	Q	2000
Cost on mismatch of pendulum energy	C_e	10
Target pendulum energy	E_{ref}	$1.05 \cdot E(\begin{bmatrix}0 & \pi & 0 & 0\end{bmatrix}^T)$
Boundary on α	α_{max}	1.1 rad

Table 5: Tuning parameters on swingup

9.5 Input selection

In non-linear MPC the controller can no longer exploit linearity to solve the optimization problem, every sequence of inputs has to be evaluated separately and scored separately. At the same time, the optimization problem is no longer convex, so gradient descent based optimization will not yield a global optimum. These problems are solved by only looking at a relatively small set of inputs sequences and finding the optimal input by simulating each. The selected inputs are shown in Figure 18.

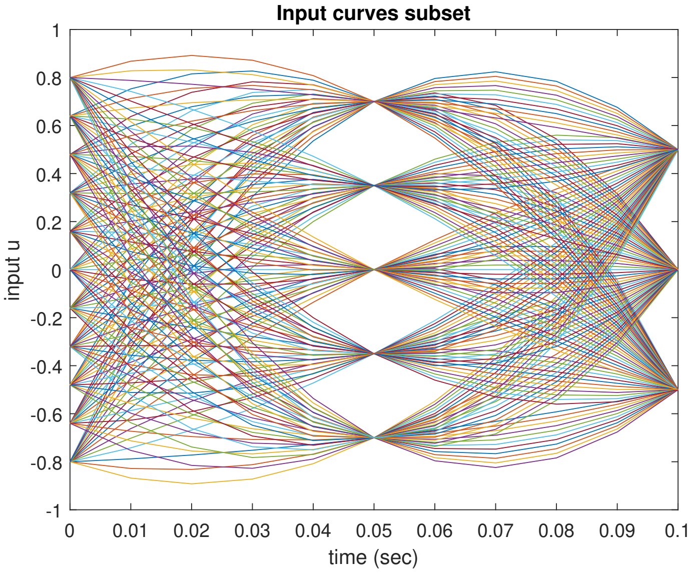
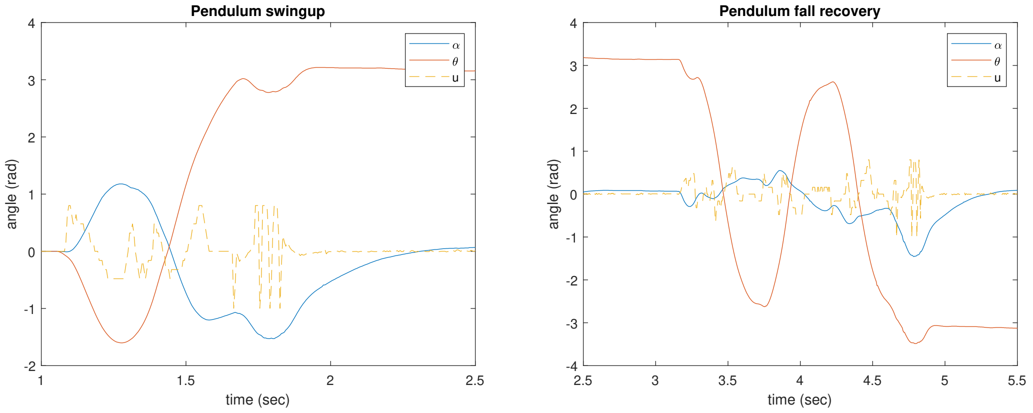


Figure 18: Swingup input trajectories (U)

These inputs curves are chosen because they correspond to the input curves often seen while tuning the other controllers.

9.6 Result

The swingup controller is very effective at moving the pendulum into upward position from any possible begin state. Two real world results are shown in Figure 19.



(a) Swingup from downward position with no velocity.

(b) Pendulum recovery after being pushed out of its upright equilibrium

Figure 19: Swingup and fall recovery.

10 Discussion of the controllers

10.1 LQR/LQI

The feedforward LQR is very easy to tune to get the closed loop response that we want. But in the real setup, it will always have a steady state error due to the non-linearities and model inaccuracy in the spring.
The steady state errors can be fixed by adding an integrator on the error of the reference and the output to make an LQI controller, but this has the effect of slowing down the closed loop response. The response time can be faster by tuning the integrator gain of the error, but this will start to introduce an oscillating behaviour around the reference.

10.2 MPC

The MPC is simple to design due to it having the same weight matrices as the LQR and easy implementation of the constraints.
The MPC can also track the reference and has a fast rise time comparable to the LQR. But the downside is that there is an oscillating effect on the angle α due to the integrator. The integrator could not be modified, because of the lack of experience using the MPC toolbox in matlab.

10.3 Hybrid swingup

The swingup controller has been shown to be successful in different conditions. The weight function makes it possible to tune the controller further and adapt it for different conditions and constraints. There is also still significant space for improvement, the very low prediction horizon is a weakness that some times leads to bad decisions by the controller. This problem is largely due to constraints on computing cost of the optimization problem. A possible improvement would be to use fewer initial guesses and then perform gradient descent on the best candidates. Another problem is input jitter when the controller switches back and forth between swingup and LQR, a possible way to address this would be to add some hysteresis to the switch.

11 Conclusion

The Qube/Rotary pendulum has been modeled using first principles to obtain the nonlinear EoM. With the EoM, a linearized state space model has been obtained by fitting PRBS a persistently exciting input signal to output with the use of PEM grey box modelling.
The linearized state space model is used to control the real Qube setup with the use of the LQR and MPC controllers. Both the LQR and MPC controller can be used to obtain desirable close-loop responses and has

good disturbance rejection. But the LQR has a steady state error for reference tracking and that was fixed with an LQI controller with the cost of having a higher rise time. The MPC also has an issue where the it is oscillating around the reference due to the integrator.

Finally a nonlinear swing-up control has been developed based on the pendulum energy. The swingup controller is very effective at moving the pendulum into upward position from any state while staying within bounds on the arm angle.

12 Appendix

12.1 Pendulum-only model

12.1.1 Schematic

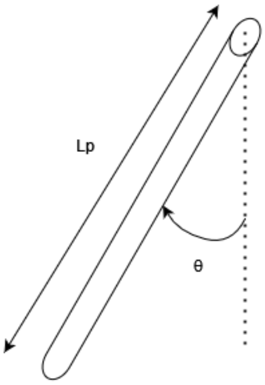


Figure 20: Pendulum-only schematic

12.1.2 EoM

$$\ddot{\theta} = -\frac{m_p L_p g sin(\theta)}{2J_p} - \frac{C_p \dot{\theta}}{J_p}, \quad J_p = \frac{L_p^2 m_p}{3}$$

12.1.3 Linear state space matrices

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ -\frac{mgL_p}{2J_p} & -\frac{C_p}{J_p} & 0 \\ \hline 1 & 0 & 0 \end{array}\right] \quad x^* = \begin{bmatrix} \theta^* \\ \dot{\theta}^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

References

[1] Quanser Inc. *Qube-servo User Manual.* URL: `http://wwwlehre.dhbw-stuttgart.de/~flaemig/MATLAB/Quanser_CubeServo/QUBE-Servo%20User%20Manual.pdf`. (accessed: 02.10.2022).

[2] Abhishek Kathpal and Ashish Singla. "SimMechanics™ based modeling, simulation and real-time control of Rotary Inverted Pendulum". In: *2017 11th International Conference on Intelligent Systems and Control (ISCO).* IEEE. 2017, pp. 166–172.

[3] Luca Laurenti Sander Bregman Dimitris Boskos. *SC52035 Qube Info.* URL: `http://homepage.tudelft.nl/b3k1c/sc52035/setups/qube.html`. (accessed: 12.10.2022).