Can you balance a cup on top of an inverted pendulum?

Laboratory Courseware
Department of Electrical Engineering,
Chalmers University of Technology

August, 2018

modified August 2019

Figure 1: Experimental lab steup, Quanser 2DOF Inverted Pendulum, [3]

1 Forewords

The inverted pendulum is a special, up-side-down positioned (regular) pendulum, see Fig. 1. In most of the case a rigid rod is mounted in a small cart/manipulator via a pivot. Since, the inverted pendulum is an unstable physical system, the task with such system is to balance (move the manipulator) in order to keep the rod in its upright position. The laboratory device is a 2-degree of freedom pendulum where the rod can turn in 2 orthogonal directions. Instead of a cart, this system has two robot arm manipulators, to move the pivot position and keep the rod in its upright position.

This courseware and lab experiment contributes to the understanding of stabilizing systems under uncertainty and implement robust and optimal control theory. Did you know rocket launching and landing or segways are inverted pendulum like problems, see Fig. 2?

Figure 2: Rocket and segway. Source: Wikipedia

This syllabus is supporting students who are taking the course *Robust and nonlinear control (EEN050)* at Chalmers University of Technology, Gothenburg, Sweden and are selecting the 2-DOF Inverted Pendulum as an option for their lab session. The information contained within this paper has two main parts; (*i*) (mandatory a-priori) **preparation**, (*ii*) real time **experiment**. Students are supposed to read this paper carefully as well as perform all the exercises before attending the session.

The layout of the labware is the following; on the basis of first principal formalism, a linear time invariant LTI model is derived and analyzed. The nominal model will be topped up by uncertainties that originates from balancing a cup. This LTI model will be used to develop robust and optimal output feedback $\mathcal{H}_\infty$ controller (1 and 2 degree of freedom robust controller structures).

Special thanks to former MPSYS students Yalcin Kalafat and Peixi Gong to create the manual under the supervision of Balazs Kulcsar (Automatic Control Group, Division of SYSCON, Department E2, Chalmers University of Technology).

Aim:

1. Answer the title question.
2. Practice weight selection. Turn theory to practice
3. Implement real-time linear $\mathcal{H}_\infty$ controller design to control unstable and uncertain system.

2 Preparation

The equipment used is a 2-DOF Inverted Pendulum (Quanser) with four-links and two robot manipulators. Part of this labmanual is taken from [3].

As you can see from Figure 3, the servo x (left hand side servo) controls the pendulum in direction x while the servo y (right hand side servo) controls the pendulum in direction y.

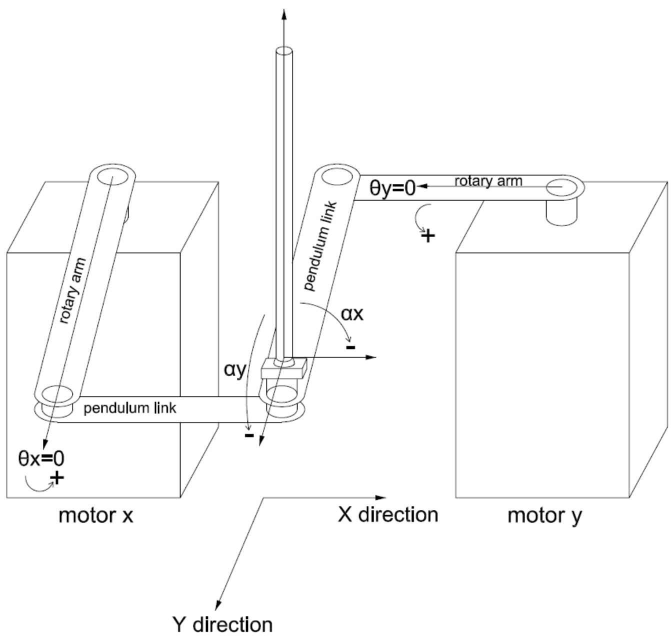

Figure 3: 2-DOF Inverted Pendulum [3].

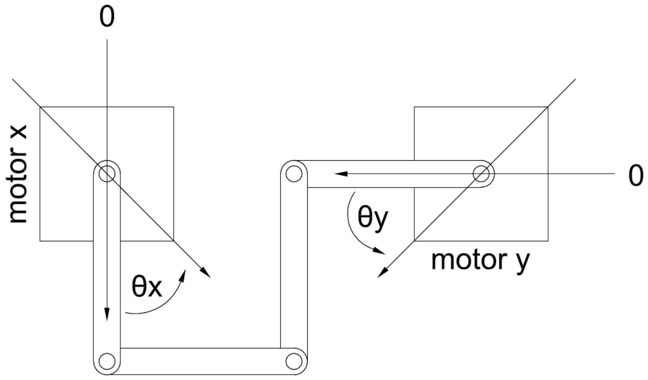

Figure 4: 2-DOF Robot HOME Position [3].

The rod can tilt in direction x and y too. Links that are directly connected to the servos are called rotary arms. The *home position* of the 2-DOF Inverted Pendulum is represented by the schematic draw illustrated in Figure 4. This position is defined as $\theta_x = \theta_y = 0$, and rod angle $\theta_x = \theta_y = 0$. Angles increase positively in counter-clockwise(CCW), the servos (and thus the arm) turn CCW when the servo control voltage is positive [3]. The inverted pendulum angle α_x and α_y is zero when it is perfectly in the upright position and increases positively CCW [3]. The home position is selected and used as an operating point for the rest of the studies.

3 First principal pendulum model

If the rod does not deviate much from its home position, the coupling in the dynamics of two directions is negligible. Hence, the 2-DOF Inverted Pendulum can be modelled as the combination of two identical, but 1-DOF Inverted Pendulum (block diagonal). One pendulum moves in x direction while the other in y direction. In the sequel we will use this simplification.

First principal pendulum model can be obtained by finding the equations of motion for robot manipulators with multiple joints and rod displacement.

3.1 Nonlinear Equations of Motion and nominal LTI model

The nonlinear equations of motion for the 1-DOF Rotary Pendulum are given by

$$\begin{aligned}\ddot{\theta} = f_3(\theta, \alpha, \dot{\theta}, \dot{\alpha}, V_m) = &\frac{-\frac{1}{2}((4M_pL_p^2\alpha\dot{\theta}\dot{\alpha} - 8C_oV_m + 8D_r\dot{\theta})J_p + M_p^2L_p^4\alpha\dot{\theta}\dot{\alpha})}{((4J_r + 4M_pL_r^2)J_p + M_pL_p^2J_r)} \\ &- \frac{-\frac{1}{2}((M_p^2L_p^3L_r\dot{\theta}^2 + 2M_p^2L_p^2L_rg)\alpha - 2M_pL_p^2D_r\dot{\theta} + 2M_pL_p^2C_oV_m)}{((4J_r + 4M_pL_r^2)J_p + M_pL_p^2J_r)}\end{aligned} \tag{1}$$

$$\begin{aligned}\ddot{\alpha} = f_4(\theta, \alpha, \dot{\theta}, \dot{\alpha}, V_m) = &\frac{((M_p^2L_p^2L_r^2 + M_pL_p^2J_r)\dot{\theta}^2 + 2J_rM_pL_pg + 2M_p^2L_r^2L_pg)\alpha}{((4J_r + 4M_pL_r^2)J_p + M_pL_p^2J_r)} \\ &+ \frac{2M_pL_rL_pC_oV_m - 2M_pL_rL_pD_r\dot{\theta} - M_p^2L_p^3L_r\alpha\dot{\theta}\dot{\alpha}}{((4J_r + 4M_pL_r^2)J_p + M_pL_p^2J_r)}\end{aligned} \tag{2}$$

where α and θ are angles (see Figure 3 and 4) and V_m the voltage applied to the servomotor. All variables besides $\alpha, \dot{\alpha}, \theta, \dot{\theta}$ and V_m are constant and described in Table 1. In the next section we associate a linear time-invariant state space model with (1) and (2).

Parameters	Description	Value
$\mathbf{M_p}$	Pendulum mass with T-fitting	0.1270
L_r	Length of rotary arm	0.1270
$\mathbf{L_p}$	Length of the pendulum(w/T-fitting)	0.3111
J_r	Equivalent inertia with the 4-bar linkage	0.0083
$\mathbf{J_p}$	Pendulum inertia around CoG	0.0012
D_r	Arm viscous damping coefficient	0.0690
$\mathbf{C_o}$	Voltage convert coefficient	0.1285
g	Gravitational constant	9.81

Table 1: Nominal parameter values. Boldfaced parameters are subjected to change and will be considered uncertain.

3.2 Linear Time-invariant State-space Model

Let the state vector $x(t) \in \mathbb{R}^4$ and control signal $u(t) \in \mathbb{R}$ be defined by

$$x(t) = \begin{bmatrix}\theta(t) & \alpha(t) & \dot{\theta}(t) & \dot{\alpha}(t)\end{bmatrix}^T \text{ and } u(t) = V_m(t). \tag{3}$$

Moreover, let $f_1(x) = x_1$ and $f_2(x) = x_2$, then the state equation associated with (1) and (2) is given by

$$\dot{x}(t) = f(x(t), u(t)), \tag{4}$$

where $f = \begin{bmatrix}f_1 & f_2 & f_3 & f_4\end{bmatrix}^T$. The angles $\theta(t)$ and $\alpha(t)$ are available for measurement, so the measurement equation is given by

$$y(t) = \begin{bmatrix}x_1(t) & x_2(t)\end{bmatrix}^T = \begin{bmatrix}\theta(t) & \alpha(t)\end{bmatrix}^T. \tag{5}$$

In this lab we will stabilize (4) about the upright position, which by convention is the origin. Using standard procedures we linearize (4) about the operating point $(x_0, u_0) = 0 \in \mathbb{R}^5$ to get the model approximation

$$\begin{aligned}\dot{\delta x}(t) &= A_\delta \delta x(t) + B_\delta \delta u \\ y(t) &= C_\delta \delta x(t) + D_\delta u(t)\end{aligned} \tag{6}$$

where $\delta x(t) \approx x(t) - x_0$, $\delta u(t) = u(t) - u_0$, and

$$C_\delta = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}, \quad D_\delta = \begin{bmatrix}0\\0\end{bmatrix}. \tag{7}$$

The system matrices A_δ and B_δ are addressed in Question 1.

Question 1 Open up the script **EEN050_lab_preparation.m**. Here we have partially defined the system matrices A_δ=Adelta and B_δ=Bdelta. Using the command "ureal" define (M_p, L_p, J_p, C_o) with the uncertainties $(\pm 50\%, \pm 50\%, \pm 50\%, \pm 10\%)$ respectively. To obtain the nominal value of "Adelta" you can call the command "Adelta.NominalValue". What are the eigenvalues of this matrix?

Answer

3.3 Uncertain pendulum models

We recommend to use Matlab/Robust Control Toolbox and the following commands may especially be helpful for the solution: *ureal,ufrd,udyn,ucover,usample.* Check the numerical examples in the Toolbox for more help.

Create a block diagonal nominal LTI system model as

$$A = \begin{bmatrix} A_\delta & 0 \\ 0 & A_\delta \end{bmatrix},\ B = \begin{bmatrix} B_\delta & 0 \\ 0 & B_\delta \end{bmatrix},\ C = \begin{bmatrix} C_\delta & 0 \\ 0 & C_\delta \end{bmatrix},\ D = \begin{bmatrix} D_\delta & 0 \\ 0 & D_\delta \end{bmatrix}.$$

The inverted pendulum in Figure 1 will be modified with a small platform (cup-holder) and a cup will be added on top too. Therefore, the nominal system model $P_{nom} = (A, B, C, D)$ will be changed

Question 2 Create an equivalent unstructured uncertainty representation to bound the effect of real parametric uncertainties. We aim at finding a 4th order equivalent input multiplicative uncertainty and related overbound weighting W_{iM} (hint: find a SISO transfer function overbound and use it as w_{iM}, $W_{iM} = 10 \cdot \begin{bmatrix} w_{iM} & 0 \\ 0 & w_{iM} \end{bmatrix}$). Draw the block diagram of the equivalent unstructured uncertainty representation.

Answer

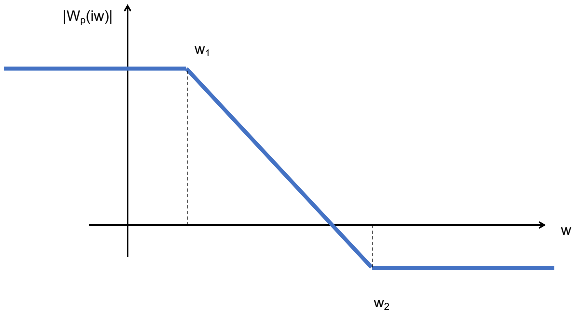

Figure 5: Shape for tracking performance weight W_p in frequency domain.

3.4 Robust controller design

In Matlab/Robust Control Toolbox and the following commands may especially be helpful for the solution: *sysic,makeweight,hynfsyn,h2syn,connect.* Check the numerical examples in the Toolbox for more help.

Question 3

Design an 1 Degree of Freedom $\mathcal{H}_\infty$ controller that can balance the extra load (glass). Draw the augmented block diagram first.

Use the following outputs:

- Input multiplicative weight W_{iM} and the weighted uncertainty output $y_\Delta = W_{iM}u$
- Penalize the control input u by W_u as a performance output
- Angle tracking errors $(y - W_r r)$ performance weighted by W_p (for both θ and α)
- Plant output y

and the inputs as:

- Uncertainty inputs (identity weightings), u_Δ
- $W_{r\theta}$ as the weight to shape the reference signal r
- W_n weighted sensor noise input n
- the control input u

The augmented system model has two control inputs (u) and 4 noisy measured angles as outputs $(y + W_n n)$ to the $\mathcal{H}_\infty$ controller.

Answer

Question 4

Find the weightings for the above terms with the following conditions.

- The input signal is penalized with a constant and channel-wise penalty of 0.05.
- W_r is a block diagonal matrix with first order lags where the time constant and the DC gain are both 1.
- In order to weight tracking error $(y - W_r r)$, a block-diagonal weighting is proposed (see the Bode diagram of the weight in Figure 5) such as

$$W_p = w_p(\frac{\frac{s}{7}+1}{\frac{s}{8E-3}+1} + 1).$$

Find 3 possible values for w_p (positive real scalar) so that the expected steady state tracking error is

Option 1: 0.0025 radian

Option 2: 0.0251 radian

Option 3: 0.001 radian.

- Account for 0.3 degree equivalent and additive noise in all of the measurement channels and calculate W_n accordingly.

Answer

Question 5

Create the system interconnection for the robust controller design in Matlab. Design a $\mathcal{H}_\infty$ controller. What is the γ value found to be minimal? Is the robust performance condition fulfilled? Check robust stability condition, is that kept? Check the time-domain closed loop simulations with the help of the Simulink file *ExperimentRobustControl.* Is it stabilizing (nominal stability)? What is the settling time?

Answer

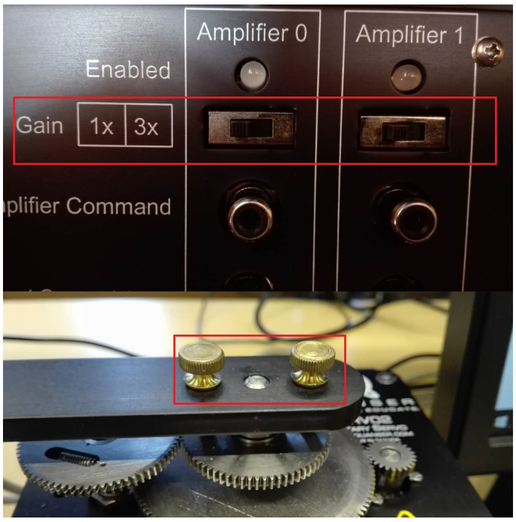

Figure 6: Voltage Amplifier Gain.

Save the controllers state-space matrices as variables ah, bh, ch, dh. **PLEASE BRING ALONG YOUR MATLAB SCRIPT AND SIMULINK FILE TO THE LABSESSION.**

4 Experiment

DOS and DONTS

- Be aware, the inverted pendulum device is quite sensitive, work gently with it.
- Put on and take off cup from the top of the pendulum gently.
- Before you run your code, ask permission to execute it from the teaching assistant!
- Before turning on the voltage amplifier device, ensure the voltage gains of both amplifier ports are set to 1×(Figure 6), or the pendulum device will be damaged!
- If there comes error about '*quarc_comm*' when running the experiment with Simulink, go to *SRV02+2DIP-E→ HIL-Initialize*, set *Board type* to *q8_usb*. If the problem still persists, select QUARC|Set default options.
- After long time running, the 6 thumb screws(Figure 6) on the pendulum might be loose and have some bad impact, ensure the thumb screws are properly adjusted to be tense.

The **ExperimentRobustControl.mdl** in Simulink as shown in Figure 7 is used to balance the 2-DOF Inverted Pendulum to keep the rod in the upright position. The 2-DOF Inverted Pendulum contains **QUARC®** blocks that interface with the DC motor and the angular sensors of the 2-DOF Inverted Pendulum system.

As you can see from Figure 7, the direction x and y are separately controlled by two identical LQG controllers, due to the fact of link coupling around the home position.

Question 6
First we do an experiment with an LQG controller that has been designed in class SSY285.

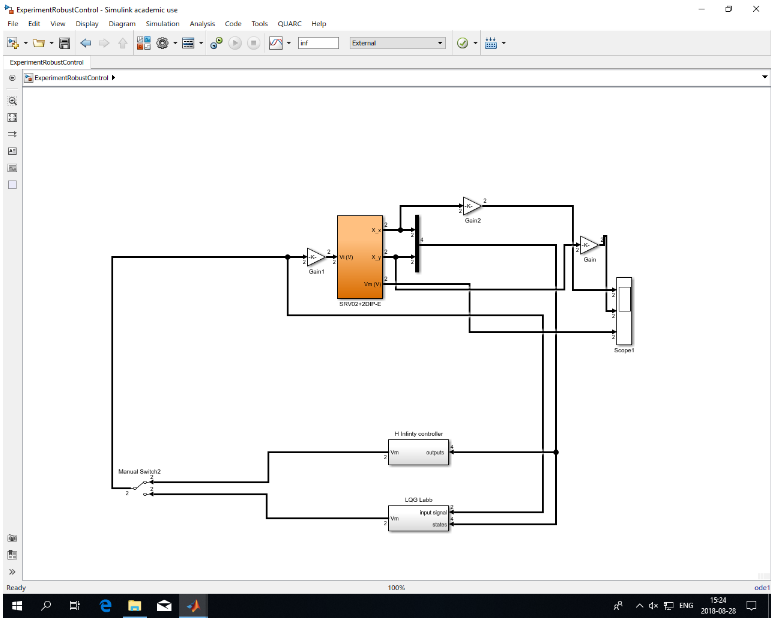

Figure 7: Simulink model used with **QUARC®** to run controller on the 2-DOF Inverted Pendulum.

1. Open the script **EEN050_lab_experiment_setup.m**. First turn the manual switch that selects the controller to LQG.
2. Load in your $\mathcal{H}_\infty$ controller parameter matrices to the appropriate part in the code (load).
3. Run the script.
4. Open Simulink file **ExperimentRobustControl.mdl**.
5. Open the scope.
6. In the Simulink diagram, run QUARC|Build.
7. Manually bring the pendulum to the upright vertical position and hold it by the uppermost tip (home position). Make sure that the pendulum is centered along both axes and is motionless.
8. To start the controller, click on the *Connect to target* button and then the *Start* on the Simulink toolbar (or select QUARC|Start from the menu). Gently release the pendulum once you feel the servo begin to stabilize the pendulum.
9. Inspect the movement of the pendulum, and check the scopes.

Try to gently apply a disturbance on the pendulum. Observe how it reacts. Take a glass located next to the pendulum and try to place on the platform located on the top of the pendulum. Be prepared to catch it in case the LQG controller fails to balance with.

Question 7

1. Open the script **EEN050_lab_experiment_setup.m**. First turn the manual switch that selects the controller to $\mathcal{H}_\infty$ controller.

2. Make sure the state space matrices for tour $\mathcal{H}_\infty$ are in the work space.
3. Comment our "clear all" and run the script.
4. Open the scope.
5. In the Simulink diagram, run QUARC|Build.
6. Manually bring the pendulum to the upright vertical position and hold it by the uppermost tip (home position). Make sure that the pendulum is centered along both axes and is motionless.
7. To start the controller, click on the *Connect to target* button and then the *Start* on the Simulink toolbar (or select QUARC|Start from the menu). Gently release the pendulum once you feel the servo begin to stabilize the pendulum.
8. Inspect the movement of the pendulum, and check the scopes.

Try to gently apply a disturbance on the pendulum. Observe how it reacts. Take a glass located next to the pendulum and try to place on the platform located on the top of the pendulum. Be prepared to catch it in case of power loss.

Distrubance reaction properties and tracking compenstation of the $\mathcal{H}_\infty$ controller differs (α, θ).

References

[1] O Wigström, *3-DOF Helicopter Laboratory Session*, laboratory courseware and manual, Chalmers University of Technology, Göteborg, Sweden, 2014.

[2] T Wik, *LQG control of a double-tank*, laboratory courseware and manual, Chalmers University of Technology, Göteborg, Sweden, 2004.

[3] J Apkarian, H Lacheray, M Lévis, *LABORATORY GUIDE: 2 DOF Inverted Pendulum Experiment for MATLAB®Simulink®Users*, Quanser, 2012.

[4] B Kulcsar, *Lecture notes for Linear Control System Design, SSY285*, Chalmers University of Technology, Göteborg, 2013-16.