CACSD Practical Session
Inverted Pendulum

Oscar Mauricio Agudelo, Bart De Moor
ESAT-STADIUS
Kasteelpark Arenberg 10, 3001 Heverlee
e-mail: mauricio.agudelo@esat.kuleuven.be
tel:(016)32 86 51

The original manual was written by Peter De Gersem and Jeroen Buijs.
This manual has been updated by Oscar Mauricio Agudelo.

2016

1 Aim of this Session

The subject of this session is the *Inverted Pendulum*. This setup consists of a rod (the pendulum), mounted on a cart which slides on a shaft (fig. 1). The cart is equipped with a motor, which lets the cart move on the shaft (by means of a gear and a toothed rack). As you can readily imagine, this system has an unstable equilibrium point (the rod staying straight up). You'll have to design a stabilizing controller for this Inverted Pendulum, i.e. a controller which drives the motor such that the rod stays vertical and the cart goes to and stays at a desired position. The measured signals are the position of the cart on the shaft and the angle of the rod (both obtained with potentiometers), the output is the voltage to the motor. The controller is implemented in software on a PC, equipped with a data-acquisition card.

It's important that you read this manual completely before starting. And please read section 3 again before doing experiments on the real setup.

2 Control Strategy

First of all, the CACSD course is concerned with *model-based* control system design. So the first thing to do is to find the *state-space* model of the system. Next, you can apply the model-based methods described in the course notes to design the controller. You will need a computer program to do so (*Matlab*), and in the meantime you can do some simulations and calculations to predict the response of the *closed-loop* system[1] in *Simulink*. When you're convinced

[1]The closed-loop system is the system consisting of the plant and the controller.

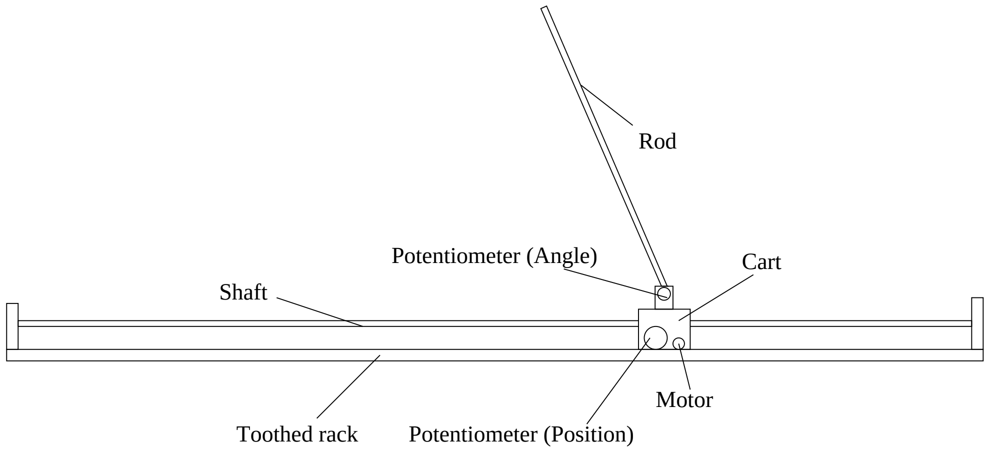

Figure 1: The inverted pendulum setup.

that your controller has the right properties, it's time to try it out on the real system.

2.1 The State-space Model

There are different possibilities to obtain the state-space model of a physical system. One of them is identification. In this case however, physical modeling, i.e. by using some mechanical and electrical laws which you've probably already forgotten for a long time, is simple enough, so that's the way we will do it. The derivation of the model is given, but make sure that you understand what is happening, not only because we might ask you some questions about it, but especially since some insight in how the model is derived may lead to a better understanding of the results you obtain later on. It's logical that good knowledge of the system you're going to control and of the model you use to represent this system, is the first step towards a well functioning controller.

First, we will model the mechanical part of the system. In a second step the electrical part will be modeled and the two models will be concatenated.

2.1.1 The Mechanical Part

You can simplify the mechanical modeling by neglecting friction, and by considering the mass of the rod to be concentrated in the middle of the rod. Fig. 2 shows a drawing of this simplified model.

The input of the mechanical system is the force F (coming from the motor), the outputs are the position of the cart x and the angle of the rod α. As you might discover, the state-space model of this system contains four states. The most logical choice seems $\{x, \alpha, \dot{x}, \dot{\alpha}\}$, so we'll stick to these.

We will use the Lagrange equations here to derive the non-linear model.

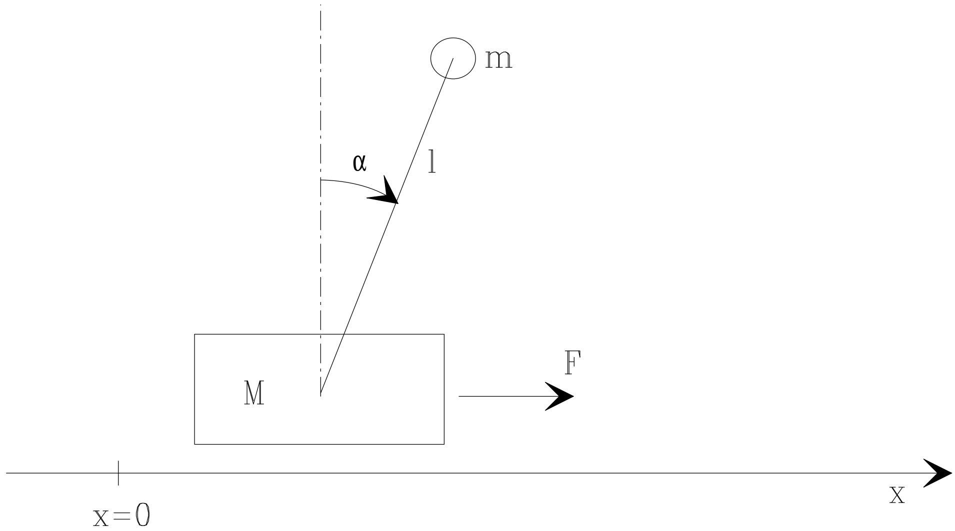

Figure 2: The simplified model of the inverted pendulum setup (with l is half of the real length of the rod).

They are given by

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\alpha}}\right) - \frac{\partial L}{\partial \alpha} &= 0\\ \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} &= F\end{aligned}$$

where the Lagrangian $L = E_{kin,tot} - E_{pot,tot}$. We introduce a coördinate system (X, Y) with its origin being the point on the rack where $x = 0$ and its Y-axis pointing upwards in the vertical direction. In this coördinate system the position of the mass representing the rod can be written as

$$(X_r, Y_r) = (x + l\sin\alpha, l\cos\alpha)$$

and thus its velocity is given by

$$(v_X, v_Y) = (\dot{x} + l\dot{\alpha}\cos\alpha, -l\dot{\alpha}\sin\alpha).$$

Knowing that the square of the absolute velocity $v_a^2 = v_X^2 + v_Y^2$, we can calculate the kinetic energy contained in the rod. The velocity of the cart being $\dot{x}$, the total kinetic energy contained in the inverted pendulum system can now be written as

$$E_{kin,tot} = E_{kin,cart} + E_{kin,rod} = \frac{M\dot{x}^2}{2} + \frac{m(\dot{x}^2 + l^2\dot{\alpha}^2 + 2l\dot{x}\dot{\alpha}\cos\alpha)}{2}$$

while the total potential energy is given by $E_{pot,tot} = mgl\cos\alpha$. Therefore the Lagrange equations lead to the following non-linear model equations,

$$\begin{aligned}ml^2\ddot{\alpha} + ml\ddot{x}\cos\alpha - mgl\sin\alpha &= 0 \quad (1)\\ (M+m)\ddot{x} + ml\ddot{\alpha}\cos\alpha - ml\dot{\alpha}^2\sin\alpha &= F \quad (2)\end{aligned}$$

2.1.2 The Electrical Part

Since our control input is not the mechanical force F, but the electrical voltage V applied to the motor that generates this force, we have to eliminate F by introducing the electrical equations of the motor. The equations governing a DC-motor are

$$V = IR_m + K_b\omega_m$$
$$T_m = K_m I\,,$$

in which R_m is the *Armature resistance*, K_b the *Back EMF constant*, K_m is the *Motor torque constant*, T_m is the torque produced by the motor at its shaft. The meaning of V, I and ω_m is clear, I hope. The motor drives a gearbox of ratio K_g (i.e. $\omega_m = K_g\omega_g = K_g\frac{\dot{x}}{r}$ and $T_g = K_g T_m$, in which ω_g and T_g are the rotation speed and the torque at the output of the gear). Finally, the force F is calculated from $F = \frac{T_g}{r}$, where r is the radius of the output gear. Concluding, the relation between the mechanical force F and the applied voltage V is given by

$$F = \frac{K_m K_g}{R_m r}V - \frac{K_m K_b K_g^2}{R_m r^2}\dot{x} \quad (3)$$

Eq. (3) can be used to eliminate F from (2).

2.1.3 The Global Model

Using eq. (1) to eliminate $\ddot{\alpha}$ from eq. (2), we obtain non-linear model equations for $\ddot{x}$ and $\ddot{\alpha}$. Since the methods we will use to control this system are based on linear models, we must linearize the obtained model around the non-stable equilibrium point. Doing so (think a second about how this is done), we obtain a standard linear state equation of the form $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$,

$$\begin{bmatrix} \dot{x} \\ \dot{\alpha} \\ \ddot{x} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mg}{M} & \frac{-K_g^2 K_m K_b}{M R_m r^2} & 0 \\ 0 & \frac{(M+m)g}{Ml} & \frac{K_g^2 K_m K_b}{M R_m r^2 l} & 0 \end{bmatrix} \begin{bmatrix} x \\ \alpha \\ \dot{x} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_m K_g}{M R_m r} \\ \frac{-K_m K_g}{r R_m M l} \end{bmatrix} V$$

Adding the output equation, you obtain a global state space model. The values of the different constants for the inverted pendulum are given at the end of this manual. Make sure to convert them to the right units. Now the model is known, make a short open loop analysis of the system and the model, and try to formulate the control problem.

2.2 Building the controller

The controller you're going to design is of LQR-type. The controller will be **designed in continuous time** and it will be implemented in a digital computer. If the sampling time in the computer is small enough, we can consider that the controller would be operating in continuous time (of course, this is an approximation). This is the approach that would be followed in this practical session. Because the states x and α are measured, and $\dot{x}$ and $\dot{\alpha}$ can be easily calculated from these measurements, no state-observer is needed.

The LQR-controller determines an optimal state-feedback gain K, such that the closed-loop system $A - BK$ (i.e. $\mathbf{u} = -K\mathbf{x}$) minimizes the quadratic performance index

$$J = \int (\mathbf{x}^t Q \mathbf{x} + \mathbf{u}^t R \mathbf{u}) dt \,.$$

The weight-matrices Q and R determine the relative importance of minimizing the states and the input-signal. Let's take

$$Q = \begin{pmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$R = 0.003$$

as initial values (this assumes your states are $\mathbf{x} = \begin{bmatrix} x & \alpha & \dot{x} & \dot{\alpha} \end{bmatrix}^t$, and that your state-space matrices are expressed in SI units). Use the Matlab-command[2] `lqr` to calculate K.

It might not be clear to you how to apply an input to the closed loop system, and how to simulate that in *Simulink* (e.g. for the step- or impulse-response): to position the cart at a certain place, we have to tell the LQR-controller that the x-state has to go to x_{desired} instead of zero. To obtain that behavior, we calculate the state-feedback not with $\mathbf{u} = -K\mathbf{x}$, but with $\mathbf{u} = -K(\mathbf{x} - \mathbf{x_d})$, where $\mathbf{x_d}$ equals $\begin{bmatrix} x_{\text{desired}} & 0 & 0 & 0 \end{bmatrix}$. So the input to your controller block K in *Simulink* has to be the difference between x and x_d[3].

While designing a controller, you need to evaluate the performance of your design, to be able to compare it with other designs. Some *evaluation criteria* are

- the **closed-loop eigenvalues (or pole-zero plot)**. They give information about the stability-margin, the damping and the eigenfrequency of the closed-loop system. The closed-loop system should at least be stable.

- the **step-response**. This gives information about the settling-time, rise-time, overshoot, amount of input-signal...

- the **impulse-response**. This gives information about the damping, the eigenfrequencies of the closed-loop system,etc.

With LQR, the choice of the weight-matrices is the most difficult part of the design. The strategy proposed here is simply *trial-and-error*: start with

---

[2]To figure out how a Matlab-command works, type `help` *command* at the Matlab-prompt. Also if you don't know how to do something in *Matlab*, you should try `help`. If you're really stuck, you can always mail us for suggestions.

[3]Another approach would be to calculate one closed-loop state space system from (A, B, C, D) and K with input x_d. But never mind.

the initial values, evaluate the prestations, change one parameter and evaluate, change another parameter, ... The ideal controller is stable, robust (can recover from a large disturbance), and is not (too) nervous or noise-sensitive.

It's up to you to find good weight-matrices with trial-and-error. Keep track of your different trials (make plots of the evaluation criteria - see also section 'Report'), and argument why you are satisfied with your final design. Put the representative information in your report (see section 4). Don't spend *too* much time on this: **it's important that you have a satisfying controller, and that you understand what the effect is of changing parameters in Q and R.** It's not the aim of this practical session to try as much different values for Q and R as possible.

The result of your iterative design is a state-feedback gain K.

2.3 Implementation Details

The controller is implemented in *Simulink* on a digital computer with a data acquisition board. Normally *Simulink* is used to carry out simulations, but using the *Real-Time Windows Target* and the *Simulink Coder* of *Matlab*, *Simulink* can be used to control processes in real-time through a Data acquisition system. This approach is used in the practical sessions.

The inputs of the system (cart position and angle of the rob) are measured by means of a biased potentiometer[4] and a data-acquisition card in the computer: the voltage at the wiper-pin of the potentiometer is a measure for the position of the cart and the angle of the rod. These voltages are converted to a digital values by an A/D-converter (voltage range: -10V to 10V, resolution: 16 bits). These digital values are read by the controller (real-time Simulink diagram containing the control algorithm), and are converted from voltage to engineering units (meters for the cart position and radians for the angle of the rob). We have a linear relationship: -4.41V to $+4.41$V corresponds with -0.456 m to 0.456 m, and -6.328V to $+6.328$V corresponds with -90 to $+90$ degrees.

The controller samples these voltages at a frequency of 200 Hz (that's an arbitrary choice: fast enough compared to the dynamics of the system, not too fast for the computer to handle). The states $\dot{x}$ and $\dot{\alpha}$ are calculated using the backward difference equation:

$$\dot{y} = \frac{y^f(t) - y^f(t - T_s)}{T_s},$$

where T_s is 1/200 s (5 ms). To use this technique, it's necessary to low-pass filter the raw measurements before calculating the difference, because these are very fluctuation- and noise-sensitive (that's why there's y^f instead of y). In the controller-program, you can try different values for the cut-off frequency ω_c of this low-pass filter to study the effects. A typical value is 2 Hz ($\omega_c = 2 \cdot 2\pi$

[4]The two ends of the potentiometer are connected to the $+12$V and -12V terminals of the Power Supply, and the wiper-pin of the potentiometer is connected to the gear of the cart resp. the rod.

rad/s). The low-pass filter obeys the following law:

$$y_n^f = \frac{\omega_c T_s}{1+\omega_c T_s} y_n + \frac{1}{1+\omega_c T_s} y_{n-1}^f \,.$$

Make sure you understand what this filter is doing so that you understand where exactly the outputs should be filtered, since you will have to include this filter in your simulation schemes. Hint: Typically 50% of the students is doing this wrong.

Then the output voltage is calculated as $V = -K\mathbf{x}$ or $V = -K_1(x - x_{\text{desired}}) - K_2\alpha - K_3\dot{x} - K_4\dot{\alpha}$. Of course, the output voltage is limited. The maximum voltage for the motor is ±6V, but the controller limits this to ±5V. This voltage is then converted to an analog value with the D/A-converter, and a power-amplifier applies this voltage to the motor of the cart.

The *Real-Time Windows Target* includes a real-time engine that runs in Windows kernel mode. This real-time engine loads I/O device drivers and establishes a connection with *Simulink*. For more details you can consult the documentation of this toolbox.

3 The Proof of the Pudding

Now it's time to implement your controller. A Simulink template has been created for you (see figure 3). This template has been configured appropriately for running in real-time and it contains two blocks: the "Analog inputs" block and the "Analog Output" block. The "Analog Inputs" block gives you the measurements of the sensors in engineering units each sampling time (5 ms). The "Analog Output" Block allows you to send the control action (volts) to the plant. So, what you have to do is to complete the Simulink diagram by means of the necessary blocks in order to implement your control strategy.

You should plan your experiments before you come to the setup, because the time is limited. Try to spend your time at the setup as efficiently as possible.

The actual setup resides in room 02.58. On a table you find the rack with the cart and the rod, a power module, and a PC. Before you do anything, read these instructions thoroughly, because the setup is not that robust. **To use the set-up, you'll have to make a reservation first**. To this end, use the online CACSD practical setup reservation system[5]. Normally the best thing to do is to make a reservation for two up to four hours, just in case something goes wrong. More practical information about the actual testing can be found in the webpage of the CACSD course in Toledo. Remember to regularly check this webpage for frequently asked questions and other advice. Initially, don't hesitate to exchange information between groups and help each other out. Eventually, don't hesitate to contact the CACSD assistants if unlisted problems arise.

3.1 Description of the Simulink template

The Simulink template is shown in Figure 3. The "Analog Inputs" block gives the measurements of the sensors (they are updated each 5 ms) in meters and

[5]Check the webpage of the CACSD course in Toledo. Section: Practical sessions.

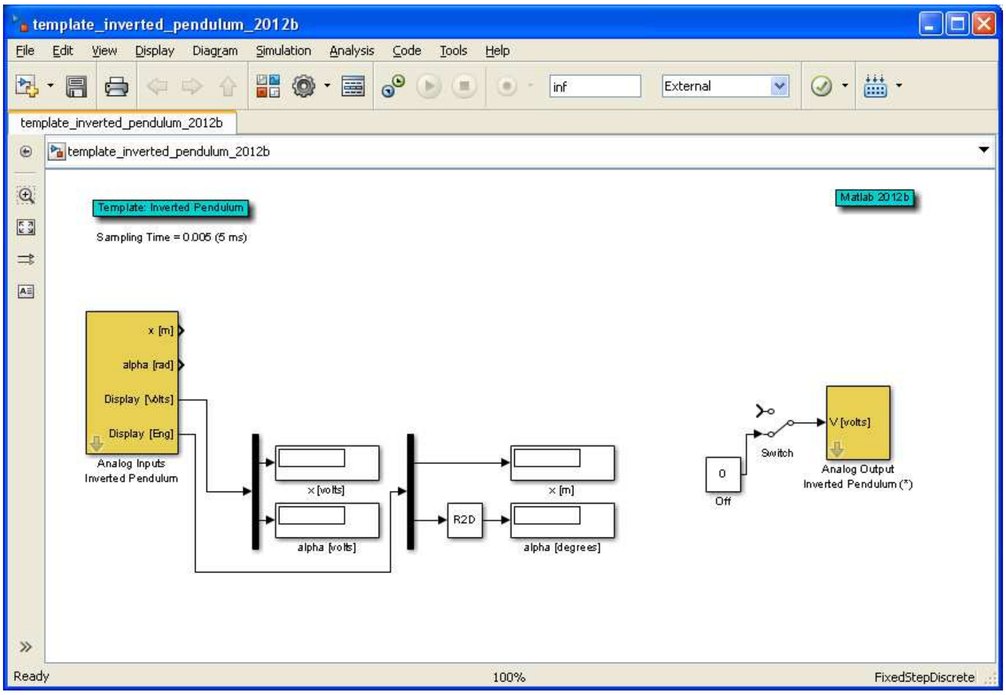

Figure 3: Template of the real-time Simulink diagram.

radians for the cart position and the angle of the rob respectively. Additionally this block has two special outputs, "Display [volts]" and "Display [Eng]", which are used for visualization purposes. These two outputs are connected to a set of displays in order to provide a local visual measure of the process variables. Each display is updated each 0.25 seconds. **Don't use "Display [volts]" or "Display [Eng]" in your control strategy because their update time is larger than 5 ms**. The "Analog Inputs" block has the following configuration parameters:

- **F_x:** This is the Conversion factor from volts to meters for the Cart Position. It has been set for you.
- **Off_x:** This is a voltage value that is added to or subtracted from the voltage read by the Data acquisition system. It is used to set the zero position of the Cart.
- **F_alpha:** This is the Conversion factor from volts to radians for the Angle of the Rob. It has been set for you.
- **Off_alpha:** This is a voltage value that is added to or subtracted from the voltage read by the Data acquisition system. It is used to set the zero position of the rod.

The "Analog Output" block allows the controller to send a voltage signal to the motor of the Cart. At the input of this block there is an on-off switch. This switch allows you to switch on or switch off the motor of the cart. If you want to change its state, you only have to click it. Internally the "Analog Output" block includes a "Saturator" block which limits the output voltage from -5V to 5V.

In order to run the real-time Simulink diagram you have to follow these steps:

- In the Simulink window, and from the "Code" menu, point to "C/C++ Code", and then click "Build Model". During the Build Process the Simulink diagram is converted into C code and afterwards it is compiled in order to generate a special binary file, the "real-time application".
- From the "Simulation" menu click "Connect To Target". This action loads the "real-time application" into memory. Also, you can connect to the target from the toolbar by clicking  .
- Finally, from the "Simulation" menu click "Run" for starting the execution of the real-time application. You can also start the execution from the toolbar by clicking  .

For stopping the real-time application, you have to click "Stop" in the "Simulation" menu (or by clicking  in the toolbar). This action automatically unloads the "real-time application" from memory.

The template was created using *Matlab* R2012b and *Simulink* 8.0. It is important to open such template using the mentioned *Matlab* version in order to avoid compatibility problems. The Simulink template is available in the webpage of the CACSD course in Toledo.

3.2 Try it out

If the rack with the cart is on the seesaw, lift it off and put it straight on the table. If the load cart with the spring is still connected to the motor cart, you have to contact the CACSD assistants to fix the setup. Please don't do that yourself.

Before testing your controller it is really important to calibrate the zero position for the angle of the rob and for the cart. Once you have loaded the Simulink template, ensure that the position of the switch (it is connected to the "Analog Output" block) in the Simulink template is pointing to "Off". Then start the execution of the real-time application (the template) as it was explained in the previous section. Now, turn on the Power Module PA0103 by pressing the Red On/Off button. Once the real-time application is running you will observe the measurements of the sensors in the displays. Now, hold the rod in the equilibrium position and take a look to the "alpha[volts]" display. Its reading must be close to zero. If it is not, follow these steps:

- Double click the "Analog Inputs" block in order to see its configuration parameters.

- Write the voltage read (in the “alpha[volts]” display) with opposite sign in the “Off_alpha” text box.

- Click Ok.

After the previous actions, you will observe how the reading in the “alpha[volts]” display will be close to zero.

The zero position of the cart should be located in the middle of the rack. In order to do this, you have to roll the cart on the rack totally to the left until the potentiometer (the largest gear) is at its physical stop. Then, roll it back to the other side to check that the physical stop at the other end is not reached. DO THIS ALWAYS IF YOU’RE NOT SURE THAT THE POTENTIOMETER IS CORRECTLY SET, OTHERWISE YOU WILL DAMAGE IT![6]
Finally, roll the cart to the middle of the rack. Afterwards, you have to check the “x[volts]” display. If the value shown is not close to zero, you have to proceed in a similar way like in the case of the angle of the rob, but having in mind that the display to read is “x[volts]” and the configuration parameter in the “Analog Inputs” block is “Off_x”.

When you are ready for testing the controller, follow this guideline:

- Start your real-time application, keeping the position of the switch (it is connected to the “Analog Output” block) at “Off”.

- Turn the rod straight up (as close as possible to the equilibrium position) with the cart approximately in the middle , hold the track with your other hand so that it can’t slide over the table, and turn the motor on by clicking the switch in the Simulink diagram. If the system goes unstable, turn the motor back off (quick by clicking the switch), grasp the rod before it falls, and reconsider your design. Maybe you used the wrong signs for the gains. Also think about the magnitude of the gains, e.g. what voltage is generated if the position has a small deviation from the equilibrium. . .

  If the rods stays straight up, try disturbing it slightly and see what happens. If that works too, try changing the position (start with 5 or 10 cm) and see what happens. If everything goes well, collect data from the system (cart position and velocity, angle of the rob, the voltage sent to the cart, etc.) in order to generate some representative plots (see section “Report”) in *Matlab*.

[6]The original manual explains it like this: *The potentiometer which measures the cart position **has physical stops**. This means that if the physical stop is reached while the cart is not at an end, the motor will drive against it and eventually the potentiometer will fail! You should therefore ensure that the track potentiometer reaches its limits outside the range of the track. You do this by turning the potentiometer shaft, by hand, completely to the right until it reaches its limit. Then lift the cart off the track and bring it completely to the right. Now turn the potentiometer shaft approximately* 1/4 *turn to the left and place the cart on the track. Slowly push the cart all the way to the left while the gears mesh with the track. The cart should reach the left end before the potentiometer hits the other limit. If not, you are doing something wrong or you have already damaged the linear sensor. Whenever there is an instability repeat this procedure to match the physical range of the potentiometer with the physical range of the track.*

If the system is in equilibrium, it may happen that the position and the angle are not exactly zero. Can you figure out what's wrong, and what you can do about it?

You can also try lifting one end of the track (slowly). Can you describe what happens?

- When you have finished, stop the real-time application (your controller in *Simulink*), turn the power module off, and turn the computer off.

4 Report

It might be a good idea to read this section before you start your calculations, simulations and tests. It might give you an idea of the things you should do and the things you should't waste time on.

The report should be written preferably **in English**. One report per group suffices. This report will be used at the oral "examination" (see further).

Contents of the report:

1. <u>Model and open loop analysis</u>

   (a) Provide the linear state-space model of the system. Define clearly which are the state variables of the model and present the numerical values of the matrices A, B, C and D. The matrices C and D must be determined based on the sensors present in the setup.

   (b) Perform a short open loop analysis. Provide the poles and transmission zeros of the system. Is the system stable? controllable? observeable? stabilizable? detectable? minimal? Justify your answer.

   (c) State very clearly what are the control goals (e.g., stabilization, disturbance rejection, tracking of setpoints, fast response, etc.).

2. <u>Design of the LQR controller and creation of the first closed-loop simulation diagram</u>

   Here you have to create a Simulink diagram mainly containing the linear model (continuous time) of the plant and the LQR controller. The assumption here is that the entire state vector is available and therefore for this particular case the matrix C must be set equal to the identity matrix.

   (a) Provide the Simulink diagram and a very brief explanation of the blocks it contains.

   (b) Starting with the matrices Q and R given before, you should follow a systematic trial and error procedure to find a good pair of matrices Q and R for your controller that satisfy the control goals (to this end you can use some of the evaluation criteria presented before, like for example the step response). Discuss the effect of changing the different diagonal entries of the matrices Q and R on the dynamics of the closed-loop system.

(c) With the chosen Q and R simulate the closed-loop system after a change in the setpoint of x (this setpoint change must be set according to the physical limits of the setup. So, you must perform a realistic setpoint change!!!). Provide the simulation results and discuss them very briefly. Do not forget to check the magnitude of the control actions.

3. Creation of the second closed-loop simulation diagram

In order to have a more realistic simulation, you have to create a Simulink diagram where the outputs of the linear model are the variables that are measured in the real setup (x and α). Therefore in order to derive the entire state vector, you have to compute the derivatives of the measured outputs by using the backward difference equation and the low pass filter described in Section 2.3. Additionally in this diagram, you have to include the saturation of the actuator, the quantization effects of the A/D converters and the measurement noise.

(a) Provide the Simulink diagram and explain the blocks it contains. Do not forget to mention how you set the saturation limits, the quantization interval and the magnitude of the measurement noise (can you estimate it from real measurements?).

(b) By using the LQR found previously and a cut-off frequency of 2 Hz for the low-pass filters, apply the same step change as in 2(c) and simulate the closed-loop system. Compare the simulation results with those obtained with the first simulation diagram and discuss. If it is necessary you can adjust the values of Q and R to have a better performance. If you have to do this then add the new simulation results.

(c) Investigate the role of cut-off frequency of the low pass filters. What is impact of the cut-off frequency on the dynamics of the closed-loop system? What happen when this parameter is set too small? or too large?

4. Experimental results ("The proof of the pudding")

By using the provided Simulink template, you have to implement and test the control system you designed before with the real setup.

(a) Provide the real-time Simulink diagram and explain it very briefly.

(b) Perform some setpoint tracking tests and make some representative plots (states, control actions). Try different setpoints. Compare the experimental results with those found with the second closed-loop simulation diagram (in this case you have to apply the same setpoint change and try to start the real system around the same initial conditions as in the simulation) and discuss. It is very likely that you have to retune your controller a little bit in order to have better results. So, do it if necessary and comment. Is there steady-state error? if so, what could be the reasons?

(c) Carry out some disturbance rejection tests and make some representative plots (states, control actions). To this end you can apply a small kick against the rod. In addition, you can also try lifting one end of the track (slowly). Discuss the results.

(d) Corroborate your findings regarding the influence of the cut-off frequency of the low pass filters on the controlled system. Present some representative plots and discuss very briefly.

(e) Complementary material - "OPTIONAL". Although it is not mandatory, you are encouraged to make some short videos of the different experiments you have carried out.

5. Conclusions

Every well written report ends with this section, so we will not make an exception here.

Apart from the technical correctness of your practicum, the quality and presentation of your report is also evaluated. So, make sure that everything is clear and well explained. For example, make sure that you label every axis of every plot, that your figures include legends if necessary, that in the text you discuss or address what is presented in every figure of the manuscript, that the text is free of typos and well redacted, etc. If you include a Simulink diagram that contains subsystems, you should also present the contents of these subsystems.

5 Examination

Once the report is finished, you can give us a copy. **Do not forget to send us the Simulink schemes together with the necessary m-files as well.** The oral exam will include some questions about the report and some general questions. Further information (dates of examination, how the points are distributed between exercises and practicum, etc.) can be found in the webpage of the CACSD course in Toledo.

6 Specifications

This section contains some data on the inverted pendulum setup.

Motor armature resistance : $R_m = 2.6\,\Omega$

Motor torque constant : $K_m = 0.00767\ \frac{\mathrm{N\,m}}{\mathrm{A}}$

Motor back EMF constant : $K_b = 0.00767\ \frac{\mathrm{V}}{\mathrm{rad/s}}$

Motor gear ratio : $K_g = 3.7 : 1$ (The output is slower)

Cart mass with motor and parts : $M = 0.455$ kg

Rod length : $l = \frac{l_{tot}}{2} = 0.305$ m

Rod mass : $m = 0.210$ kg

Radius of motor output gear : $r = 0.635 \cdot 10^{-2}$ m