Automatic Control Laboratory, ETH Zürich
Profs. M. Morari, J. Lygeros

Manual prepared by: Claudia Fischer, Aldo Zgraggen
Revision from: September 22, 2014

IfA Fachpraktikum - Experiment 2.2:

Self-Erecting Inverted Pendulum

In this experiment, a pendulum is mounted on a cart. The pendulum shall be controlled to stay at its unstable equilibrium, i.e. the upright position. You will design a linear quadratic regulator (LQR) to achieve this goal.
Additionally, you will implement a destabilizing controller that will make the pendulum swing up from its stable downward position. Finally, the two controllers will be combined to yield a self-erecting pendulum.

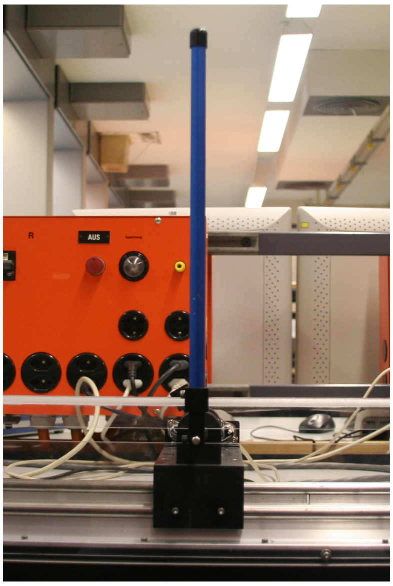

Figure 1: The inverted pendulum in the IfA lab.

For the preparation of this experiment at home you need to download the following files from the IfA Fachpraktikum website `http://control.ee.ethz.ch/~ifa-fp/` .

`stabilizing_control_structure.m`	m-file structure for inverted pendulum
`swingup_control_structure.m`	m-file structure for self-erecting pendulum

During the lab session you will need additional files that are provided locally on your machine.

Contents

Chapter 1

Problem Setup and Notation

The inverted pendulum is a classic problem in dynamics and control theory. It is widely used for testing control algorithms. As opposed to the ordinary pendulum, the inverted pendulum has its mass above its pivot point. Whereas the normal pendulum is stable when pointing downwards, the inverted pendulum is inherently unstable. Therefore it must be actively balanced in

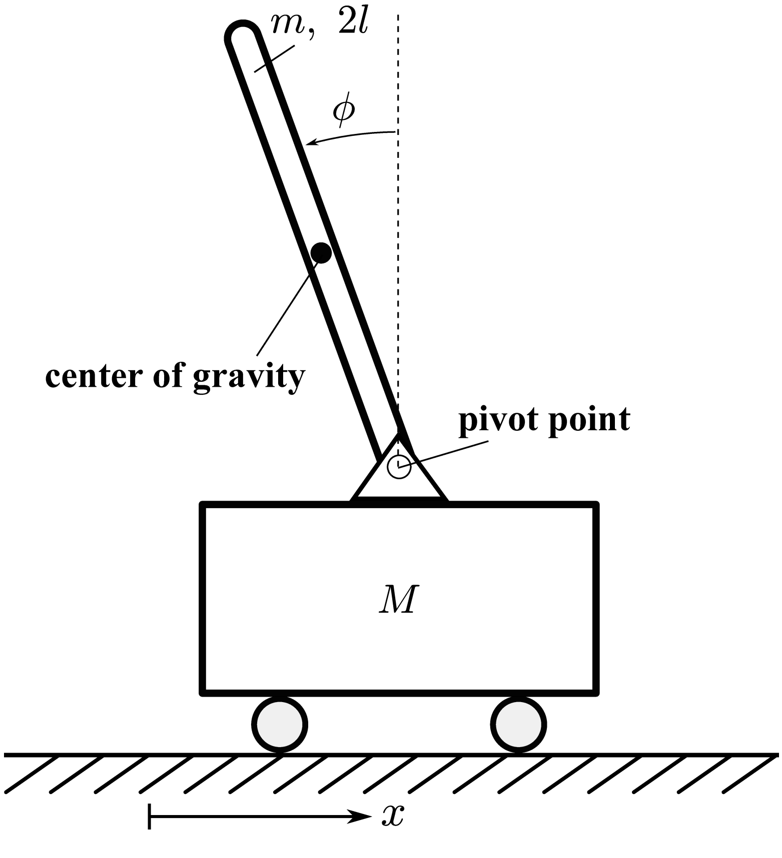

Figure 1.1: A pendulum is mounted on a cart. It can be stabilized in the upright position by moving the cart accordingly.

order to remain upright. To that end, the inverted pendulum is mounted on a cart that can move horizontally (Figure 1.1). As the stabilization of the pendulum is a nonlinear problem, the system must be linearized in order to be able to apply standard linear control methods.

Figure 1.2 shows a picture of the experimental setup you will find in the lab. The plant consists

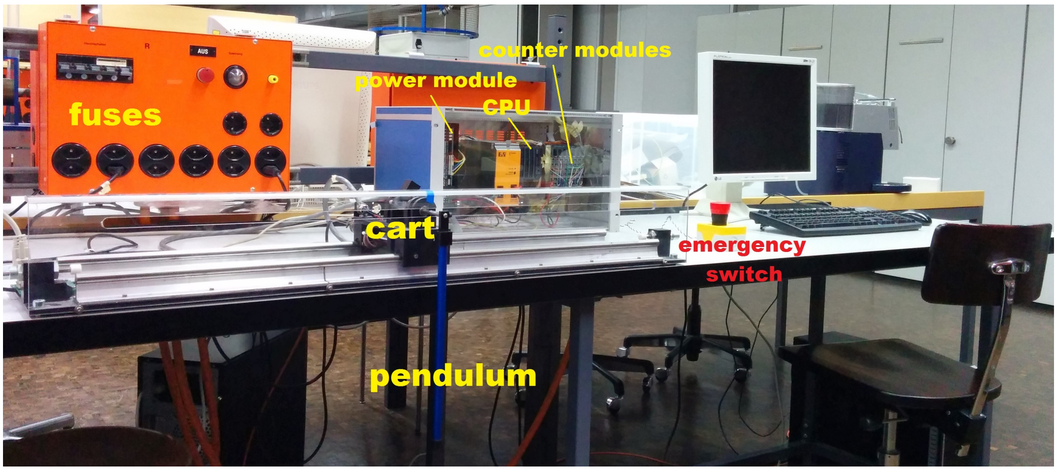

Figure 1.2: Hardware setup of the inverted pendulum in the lab.

of three parts: the pendulum, the cart, and the track, which limits the movement of the cart. The orange box contains five fuses and plugs. All fuses must be set for the plugs to be live. The blue box contains the black power module that provides the power for the motor of the cart, and the B&R industrial control unit comprising a CPU and counter modules, that processes the measurements and computes the control inputs. All code and simulations are implemented using the lab PC on the right.

Chapter 2

Preparation@Home

2.1 Modeling

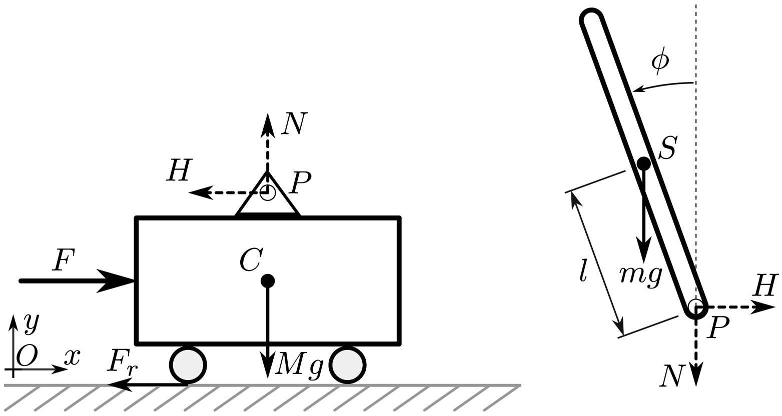

Figure 2.1: Forces acting on the two rigid bodies of the system.

Task 1: Nonlinear Differential Equations

The pendulum is approximated as a point mass in its center of gravity. The free body diagram is given in Figure 2.1. The pivot joint of the pendulum on the cart is assumed to be frictionless. The rolling resistance of the cart is given by $F_r = b\dot{x}$. The forces H and N are the forces in the pivot joint of the pendulum. The cart can be moved with the external force $F = nK_t/ri(t)$ applied by the motor. To establish the differential equations the principle of linear and angular momentums are used.

Linear and angular momentum: Recall the definitions of the linear and angular momentum and their principles for rigid bodies:

- The **linear momentum** is the product of the mass and the velocity of the center of gravity of a rigid body:

$$\vec{p} := m \cdot \dot{\vec{r}}_{OS}$$

where r_{OS} is the vector from the origin O of the coordinate frame to the center of gravity S of the rigid body.

- The **angular momentum** is the sum of the linear momentum of a stationary point of the rigid body and the spin of the body:

$$\vec{L}_p := m \cdot \vec{r}_{OS} \times \dot{\vec{r}}_{OS} + \Theta_S \dot{\phi}$$

  where r_{PS} is the vector from the reference point P to the center of gravity S.

- The **principle of linear momentum** states that the derivative of the linear momentum is equal to the sum of the forces acting on the system:

$$\dot{\vec{p}} = \sum \vec{F}_i$$

- The **principle of angular momentum** states that the derivative of the angular momentum is equal to the sum of the induced moments acting on the body:

$$\dot{\vec{L}}_p = \sum \vec{r}_{0B_i} \times \vec{F}_{B_i} + \sum M_i$$

Proceed as follows:

1. Determine the linear momentum $\vec{p}_c$ of the cart in x direction.
2. Take the first derivative of the linear momentum $\vec{p}_c$ of the cart and set it equal to all forces acting on the cart in x direction.
3. Determine the linear momentum $\vec{p}_p$ of the pendulum in x and y direction.
4. Take the first derivative of the linear momentum $\vec{p}_p$ of the pendulum in both directions and set it equal to all forces acting on the pendulum in its respective direction.
5. Determine the angular momentum $\vec{L}_p$ of the pendulum.
6. Take the first derivative of the angular momentum $\vec{L}_p$ of the pendulum and set it equal to all moments acting on the pendulum.
7. Bring the system to the form $\ddot{x} = f_1(x, \dot{x}, \phi, \dot{\phi}, F)$; $\ddot{\phi} = f_2(x, \dot{x}, \phi, \dot{\phi}, F)$

Task 2: Linearization of the system

In order to be able to apply linear control methods, this nonlinear system must be linearized. Use the method presented in the control systems lecture to linearize the system around its unstable equilibrium $\phi = 0$, $\dot{x} = 0$, $\dot{\phi} = 0$, $F = 0$, i.e. $\tilde{x}_{ss} = [0, 0, 0, 0]$, $u_{ss} = 0$.

Task 3: State space representation

Write the linearized system in state space representation.

Task 4: Numerical state space representation

Include the state space representation in the Matlab script *stabilizing_control_structure.m*. Save it as *stabilizing_control.m*. The numerical parameters are:

pendulum mass	m	$0.104kg$
cart mass	M	$1.79kg$
pendulum length from pivot to center of gravity	l	$0.1524m$
gravitational acceleration	g	$9.81\frac{m}{s^2}$
friction coefficient	b	$0.001\frac{Ns}{m}$
motor efficiency	n	1
motor torque constant	K_t	$0.07\frac{Nm}{A}$
motor pinion radius	r	$6.35e-3m$
force acting on the cart	F	$\frac{n \cdot K_t}{r} \cdot i(t)$

2.2 Optimal State Feedback Control - LQR

To keep the pendulum in the upright position, a stabilizing state feedback controller shall be implemented. The corresponding block diagram is given in Figure 2.2.
$(x_{ref}, \phi_{ref}, \dot{x}_{ref}, \dot{\phi}_{ref})$ is the state reference. The reference angle ϕ_{ref} must be equal to the unstable equilibrium π. The pendulum's and cart's velocities $\dot{\phi}_{ref}$ and $\dot{x}_{ref}$ must be zero. The cart position does not influence the equilibrium, its reference x_{ref} can therefore be chosen arbitrarily inside its allowed range.
The task here is to find an optimal feedback controller, i.e. a linear quadratic regulator (LQR) that stabilizes the system.

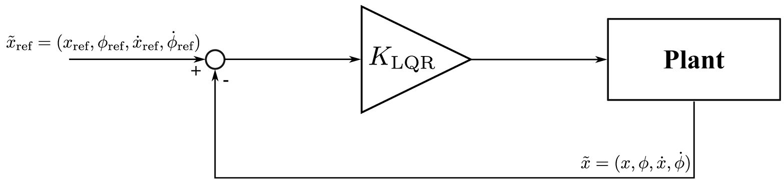

Figure 2.2: Block diagram of the state feedback system.

The idea behind LQR is to find a feedback gain K_{LQR}, such that the control law

$$u(t) = -K_{\mathrm{LQR}}\tilde{x}$$

minimises the cost function

$$J(u(t)) = \int_0^\infty \tilde{x}(t)^T Q \tilde{x}(t) + u(t)^T R u(t) \mathrm{d}t$$

and stabilises the closed control loop ($||\tilde{x}(t)|| \to 0$, and $||u(t)|| \to 0$ as $t \to \infty$) while keeping the control energy as small as possible or within certain bounds.

The optimal feedback gain K_{LQR} can be obtained using the Matlab command *lqr*. For further mathematical details of LQR, please refer to the control systems lecture notes.

Task 5: Theoretical considerations for LQR control

1. What are the dimensions of the Q-matrix?
2. Which matrix properties are necessary for Q? Why?
3. What are the dimensions of the R-matrix?
4. Which matrix properties are necessary for R? Why?
5. How is the control behaviour affected if you choose much larger values for the entries of Q than for R? What problems may occur?
6. What happens in the opposite case?
7. What happens if you increase/decrease the entries of Q and R at the same rate?
8. What is the impact of a deviation of the position of the cart from the equilibrium state, what is the impact of a deviation in the pendulum angle? What does this mean for the tuning of Q?
9. What are the units of the states and what is the expected maximum deviation? What does this mean for the tuning of Q?

Hint: Helpful information can be found in the RS1 lecture notes chapter 13

Task 6: LQR controller gain

The next step is to define the matrices Q and R and compute the gain K. For the sake of comparison to the master solution, keep $R = 1$ and only tune Q. The controller should be rather fast, but not too aggressive in order not to violate the current saturation constraints. You can compute K in *stabilizing_control.m* using the Matlab command 'lqr'!

2.3 Self-Erecting Pendulum Control

The final goal of this experiment is to obtain a self-erecting pendulum. To that end, a destabilizing controller for the angle (swing-up controller) must be implemented.

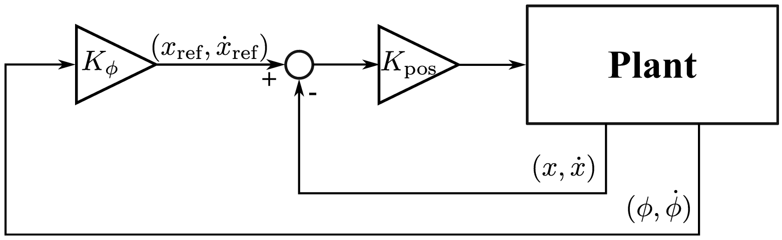

Figure 2.3: Schematics of the cascaded control of the swing-up motion.

The control of the position of the cart needs to be stable to keep the cart on the track. The idea is to have a cascaded control with an inner loop that stabilizes the cart's position, and an outer control loop that destabilizes the pendulum's down-equilibrium by calculating a corresponding cart position reference (Figure 2.3).

To obtain a self-erecting pendulum, switching between the swing-up and the stabilizing controller is necessary. This is denoted "mode selection" in Figure 2.4. Once the pendulum position reaches an angle from which it can be stabilized, the control is switched from swing-up to stabilizing.

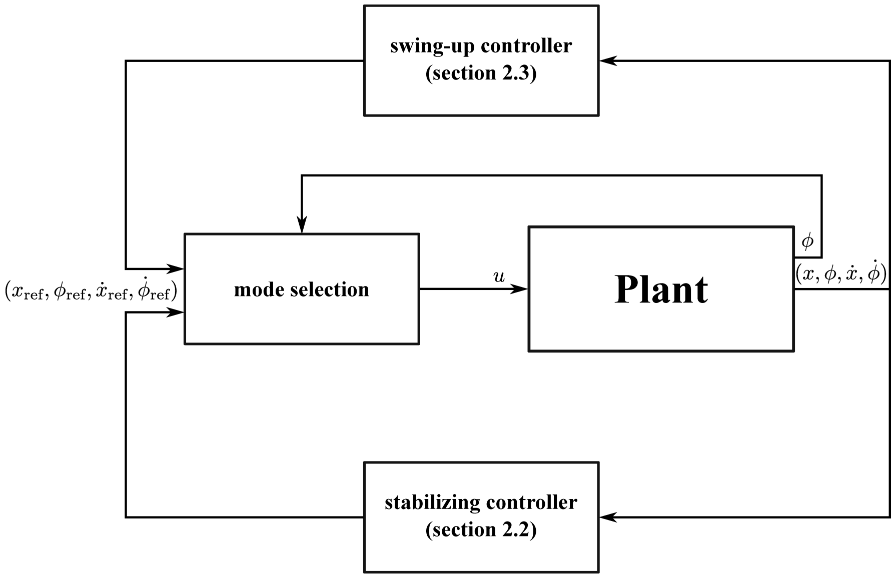

Figure 2.4: Schematics of the self-erecting pendulum control.

For the following tasks, use the file *swingup_control_structure.m*. Save the edited code as *swingup_control.m*.

Task 7: State space representation

For the **stabilizing position control**, calculate the model of the reduced system $(x, \dot{x})$. Neglect the dependence on ϕ.

Task 8: Stabilizing position control parameters

The poles shall be placed at -7 and -15, respectively. Determine the corresponding gain K_{pos} using Ackerman's pole placement formula! *Hint:* Use the Matlab command 'place'.

Task 9: Destabilizing pendulum control parameters

For the **position reference**, consider $x = P \cdot \phi + D \cdot \dot{\phi}$, where $[P, D] = K_\phi$, and $P \in \mathbb{R}$, $D \in \mathbb{R}$. How should the cart respond to the pendulum angle in order to swing up?
Remark: $\phi = 0$ is the stable equilibrium (pendulum pointing downwards), $\phi = \pi$ is the upright position.

Task 10: Switch to stabilizing control

To stabilize the pendulum in the upright position, one must switch to the stabilizing controller once the pendulum has reached an angle close enough to the upright position ($\phi = \pi$). What angle would you choose?

Chapter 3

Lab Session Tasks

The lab session consists of two parts: In a first step, the homework results will be tested in a Simulink model. In a second step, the feedback control will be employed in the actual experimental setup.

3.1 Hardware Setup

Before you start, make sure the hardware is set up correctly. First, mount the medium (i.e. 30cm long) pendulum rod to the cart front pendulum shaft, as illustrated in Figure 3.1.

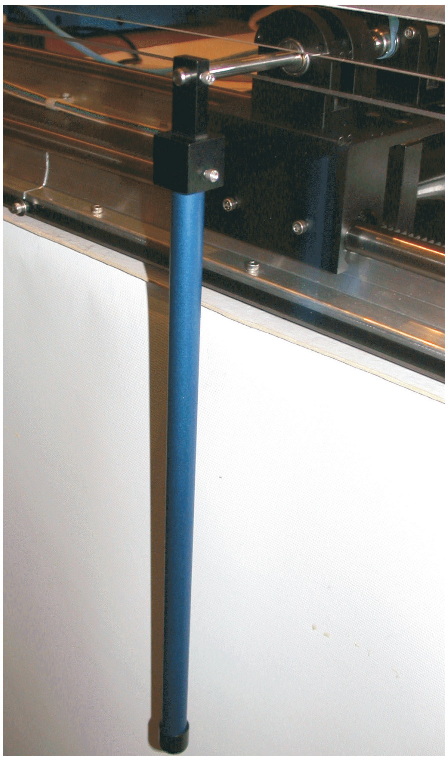

Figure 3.1: The pendulum is mounted on the cart.

3.1.1 Safety precautions

- For safety reasons ensure that there is about $1m$ **of clearance** from either end of the track at any time to avoid collisions.
- Before you start experiments, put the **cart to the middle of the track**.
- Verify that the suspended **pendulum is not moving** before starting the system.
- If you change any parameters, e.g. K, **always** verify the new controller with the Simulink model first. Before implementing the controller on the actual system, you must ensure the specifications are met without saturating the actuator.
- Ensure you are always able to **safely** reach the emergency stop button.

3.2 Inverted Pendulum Experiment

To start the session in the lab, right-click the icon **ifa 2.2** on the desktop of the lab computer and select 'run with power shell'. This will download all necessary files and start Matlab. To run Matlab scripts, just type their name in the Matlab command window.

3.2.1 Simulation

The next step is to test the controller designed in the homework in a model. First, run **ifa2_2a** if not done automatically. This will:

- change the Matlab directory to the correct folder, and
- open the necessary Simulink model *InvertedPendulum.mdl*.

Copy the Matlab script file *stabilizing_control.m* prepared in the homework to the current Matlab folder and run it. Run the Simulink model.
Investigate the model behaviour for different tuning parameters in Q! Check if the constraints on the cart postion ($\pm 500mm$) and the controller input ($\pm 15A$) are respected! Save the improved version of stabilizing_control.m to the current folder.

3.2.2 Controller Real-Time Implementation

The next step is to implement the previously designed controller on the hardware. Run **ifa2_2b**. This will:

- change the Matlab directory to the correct folder, and
- open the Simulink model *InvertedPendulum_Real.mdl*.

Copy the Matlab script file *stabilizing_control.m* with the improved parameters found in the simulation (section 3.2.1) to the current Matlab folder.

Preparation and implementation

- Before you start, make sure that **save operation is guaranteed** and the parameters are set accordingly (check with simulation!).
- Run **'START_InvertedPendulum.m'** to initialize the amplifiers and setup the hardware.
- Ensure the real-time code is run safely by **manually moving the cart to the middle of the track** so it is free to move to both sides. The LED on the black power module should turn green.
- In order to observe the system's real-time responses, use the scope in the Simulink model.

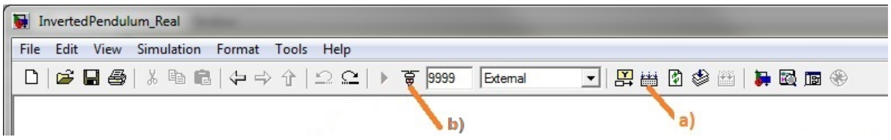

Figure 3.2: Press first a) to build the model, and then b) to connect to the target.

Execution

1. As depicted in Fig. 3.2 press a) to build the Simulink model.

2. You will be asked twice to enter the path to B&R Automation Studio in the Matlab command line. The path is C:\Program Files\BrAutomation\AS30090

3. In the Simulink model, press b) to connect to the target.

4. To activate the controller, set the 'manual allow signal' in the Simulink file to 1. Then, slowly rotate the pendulum **counterclockwise** to the upright position (the cart will begin to move). Release the pendulum as soon as the controller is activated.

5. The cart should now track the position reference and regulate the pendulum angle to 180^o.

6. Also, try introducing a disturbance by **gently** poking the tip of the rod.

7. Set the 'manual allow signal' in the Simulink file back to 0 and disconnect from the target (button b)) if you want to stop the real-time execution or change the gain K. You do not need to rebuild the project after changing K, it is sufficient to restart from 3.

3.3 Self-Erecting Pendulum

In a second experiment, we want to implement a self-erecting pendulum. As before, in a first step simulations must be run to check the homework results. Run **ifa2_2c**. This will

- change the Matlab directory to the correct folder, and
- open the necessary Simulink model *SelfErectingPendulum.mdl*.

Copy the Matlab script file *swingup_control.m* from your homework to the current folder and run it.

3.3.1 Simulation

The simulink model consists of two parts: the swing-up control and the stabilizing inverted pendulum control. Corresponding to the pendulum angle, the respective controller is active. The model also includes a small angular excitation.

- Set the switching angle to $3rad$.
- Run *swingup_control.m* and *stabilizing_control.m*. Run *SelfErectingPendulum.mdl*. Are the simulation results satisfactory?
- Tune K_{pos}, P and D and compare the results!
- Run *SelfErectingPendulum.mdl* with different switching angles, e.g. $2rad$, $2.5rad$, $3rad$. Which ones yield satisfactory results?
- Is this control save in terms of physical limitations (e.g. cart position)?
- Save the improved version of *swingup_control.m* to the current Matlab directory.

3.3.2 Controller Real-Time Implementation

For the self-erecting pendulum control, run **ifa2_2d**. This will

- change the Matlab directory to the correct folder, and
- open the Simulink model *SelfErectingPendulum_Real.mdl*.

Copy your improved version of *swingup_control.m* to the current Matlab directory.

Preparation and implementation

- Before you start, make sure that **save operation is guaranteed** and the parameters are set accordingly (see simulation!).
- Run **START_SelfErectingPendulum.m** to initialize the amplifiers and setup the hardware.
- Ensure the real-time code is run safely by **manually moving the cart to the middle of the track** so it is free to move to both sides. The LED on the black power module should turn green.
- In order to observe the system's real-time responses, use the scope in the Simulink model.

Execution

- To start the experiment, proceed as in the first point described in section 3.2.2 and Fig. 3.2.
- To activate the controller, poke the pendulum **VERY gently**. The pendulum will start to swing up.
- Set the 'manual allow signal' in the Simulink file back to 0 and then press button b) (Fig. 3.2) again if you want to stop the real-time execution.

Chapter 4

Lessons Learnt

An inverted pendulum can be stabilized using a state feedback controller based on a linearized model of the pendulum. The cart on which the pendulum is mounted is able to follow a given trajectory while keeping the pendulum in the upright position. Furthermore, it is possible to implement a self-erecting pendulum using feedback that destabilizes the stable pendulum position.

Lesson Learnt 1: Model validity

> In the simulation, for which range of starting angles does the stabilizing controller work? Why?

Lesson Learnt 2: Tuning

> What happens if you tune the destabilizing parameters for the self-erecting pendulum too aggressively?

Lesson Learnt 3: Switching

> What happens if you switch at an inadequate angle?

Lesson Learnt 4: Completion of Experiment

> Please, fill out the online feedback form on the registration page under `MyExperiments`. Each student/participant has to fill out its own feedback form. This will help us to improve the experiment. Thank you for your help.