5 Flatness based controller design for a ball on a plate

M.Sc. Laboratory Advanced Control (WS 20/21)

Prof. Dr.–Ing. habil. Thomas Meurer, Chair of Automatic Control

The objective of this lab exercise is the trajectory tracking of a ball on plate system shown in Fig. 5.1. To achieve this, a flatness-based control is investigated. First, a mathematical model is derived using the Euler-Lagrange approach. After the system dynamics have been reduced to a ball and beam setup different trajectory tracking approaches are applied. The feasibility of an input-output or input-state linearization is investigated and a controller based on a approximative approach is implemented. Finally, flatness theory is applied to determine a feed forward input trajectory to follow a predefined output-trajectory.

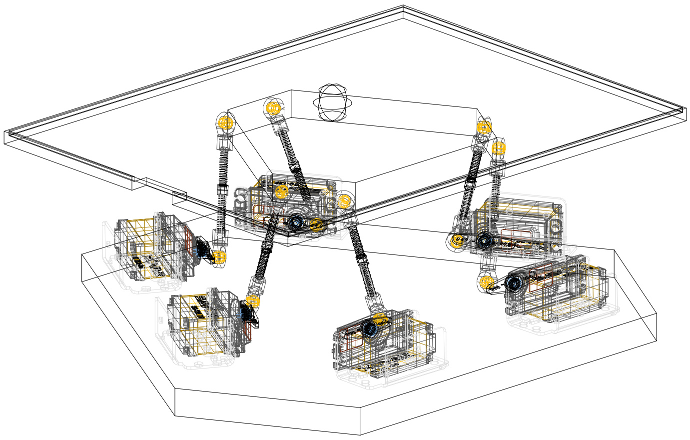

Figure 5.1: Illustration of the ball and plate setup.

This document is part of the zip-archive bap_ex.zip with the following content:

- bap_A.pdf
- par.mat
- BAP_ex1_reference.slx
- ball_and_beam.p
- ball_and_beam_reduced_discrete.p

If you have any questions or suggestions regarding this experiment, please contact:

- Henry Baumann (heba@tf.uni-kiel.de)

5.1 Mathematical model

The steward platform shown in Fig. 5.1, which is considered in this experiment. A ball placed on this platform can be moved by the inclination of the plate, which can be adjusted by the six servo motors that are attached to the base. To measure the current position of the ball on the plate a resistive touch panel is integrated on the surface of the plate.

For modeling purposes, the connection of the servo motors to the momentum or angle of the plate is not examined here. Thus only the energy of the ball and the plate is considered. For the ball and plate setup the kinetic energy of the ball consists of a translational

$$W_{kin,trans,b} = \frac{1}{2} m_b (\dot{x}_x^2 + \dot{x}_y^2) \tag{5.1}$$

and a rotatory

$$W_{kin,rot,b} = \frac{1}{2} J_b (\omega_x^2 + \omega_y^2) \tag{5.2}$$

component, which describes th [illegible] nslation and rotation along its x- and y-axis, respectively. Using the relations

$$\dot{x}_x = r_x \omega_x \qquad \text{and} \qquad \dot{x}_y = r_y \omega_y \tag{5.3}$$

equation (5.2) can be reformulated and a [illegible] .1) to obtain the total kinetic energy of the ball as

$$W_{kin,b} = W_{kin,trans,b} + W_{kin,rot,b} = \frac{1}{2}\left(m_b + \frac{J_b}{r_b^2}\right)(\dot{x}_x^2 + \dot{x}_y^2). \tag{5.4}$$

Considering that a solid metal ball is used in order to be detectable by the resistive touch panel, the balls moment of inertia can be described by $J_b = \frac{2}{5} m_b r_b^2$. Thus the total kinetic energy of the ball can be further simplified to

$$\boxed{W_{kin,b} = W_{kin,trans,b} + W_{kin,rot,b} = \frac{7}{10} m_b (\dot{x}_x^2 + \dot{x}_y^2).} \tag{5.5}$$

Additionally the kinetic energy of the plate is given by its rotational energy

$$W_{kin,p} = \frac{1}{2}(J_b + J_p)(\dot{\varphi}_x^2 + \dot{\varphi}_y^2) + \frac{1}{2} m_b (x_x \dot{\varphi}_x \ [illegible] \tag{5.6}$$

with respect to th [illegible] nter of mass. There [illegible] l is considered as a point mass at position (x_x, x_y) and J_p de [illegible] moment of inertia of the plate.

The potential energy of the system is given by

$$W_{pot} = m_b g x_x \sin(\varphi_x) + m_b g x_y \sin(\varphi_y) = m_b g (x_x \sin(\varphi_x) + x_y \sin(\varphi_y)), \tag{5.7}$$

The nonlinear dynamics given by the Euler-Langrange equations reads

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = Q_j, \quad j = 1, \dots, n, \tag{5.8}$$

where $\boldsymbol{q} = [x_x, x_y, \varphi_x, \varphi_y]$ marks the generalized coordinates, while Q_j denotes the generalized forces. In addition $\dot{\boldsymbol{q}} = [\dot{x}_x, \dot{x}_y, \dot{\varphi}_x, \dot{\varphi}_y]$ denotes the generalized velocities, while L is the Lagrange-function, given by $L = W_{kin} - W_{pot}$.

Exercise 5.1.

(i) Determine the ball on plate's equations of motion by applying the Euler-Lagrangian method and the `Matlab` *symbolic toolbox.*

(ii) When one axis of motion ist dropped, the system reduces to a ball on beam scenario. Determine the equations of a ball and beam system from (i).

(iii) Find a suitable input transformation to express the angular velocity in terms of the transformed input and implement the system into the Simulink model ballandbeam_model.slx with reduced the state vector $\boldsymbol{x} = [x_x, \dot{x}_x, \varphi_x, \dot{\varphi}_x]$. Compare the output trajectory of your solution with the reference model. The required parameters are listed in Table 5.1 and are stored in the param.mat struct.

(iv) Determine the relative degree of the ball and beam system from (iii) with $y = x_x$. Is the relative degree well defined?

Solution 5.1.

(i) $\boldsymbol{q} = \begin{bmatrix} x_x & x_y & \varphi_x & \varphi_y \end{bmatrix}$

$$\begin{aligned} L =& W_{kin,b} + W_{kin,p} - W_{pot} \\ =& \frac{7}{10} m_b(\dot{x}_x^2 + \dot{x}_y^2) + \frac{1}{2}(J_b + J_p)(\dot{\varphi}_x^2 + \dot{\varphi}_y^2) + \frac{1}{2} m_b (x_x\dot{\varphi}_x + x_y\dot{\varphi}_y)^2 \\ &- m_b g (x_x \sin(\varphi_x) + x_y \sin(\varphi_y)) \end{aligned} \tag{5.9}$$

$$\frac{dL}{d\boldsymbol{q}} = \begin{bmatrix} m_b(x_x\dot{\varphi}_x^2 + x_y\dot{\varphi}_x\dot{\varphi}_y) - m_b g \sin(\varphi_x) \\ m_b(x_y\dot{\varphi}_y^2 + x_x\dot{\varphi}_x\dot{\varphi}_y) - m_b g \sin(\varphi_y) \\ -m_b g x_x \cos(\varphi_x) \\ -m_b g x_y \cos(\varphi_y) \end{bmatrix} \tag{5.10}$$

$$\frac{dL}{d\dot{\boldsymbol{q}}} = \begin{bmatrix} \frac{7}{5} m_b \dot{x}_x \\ \frac{7}{5} m_b \dot{x}_y \\ (J_b + J_p)\dot{\varphi}_x + m_b x_x^2 \dot{\varphi}_x + m_b x_x x_y \dot{\varphi}_y \\ (J_b + J_p)\dot{\varphi}_y + m_b x_y^2 \dot{\varphi}_y + m_b x_y x_x \dot{\varphi}_x \end{bmatrix} \tag{5.11}$$

$$\frac{d}{dt}\frac{dL}{d\dot{\boldsymbol{q}}} = \begin{bmatrix} \frac{7}{5} m_b \ddot{x}_x \\ \frac{7}{5} m_b \ddot{x}_y \\ (J_b + J_p)\ddot{\varphi}_x + m_b x_x^2 \ddot{\varphi}_x + 2 m_b \dot{x}_x x_x \dot{\varphi}_x + m_b(\dot{x}_x x_y \dot{\varphi}_y + x_x \dot{x}_y \dot{\varphi}_y + x_x x_y \ddot{\varphi}_y) \\ (J_b + J_p)\ddot{\varphi}_y + m_b x_y^2 \ddot{\varphi}_y + 2 m_b \dot{x}_y x_y \dot{\varphi}_y + m_b(\dot{x}_y x_x \dot{\varphi}_x + x_y \dot{x}_x \dot{\varphi}_x + x_y x_x \ddot{\varphi}_x) \end{bmatrix} \tag{5.12}$$

$$\boldsymbol{Q} = \begin{bmatrix} 0 \\ 0 \\ \tau_x \\ \tau_y \end{bmatrix} \tag{5.13}$$

(ii) $\boldsymbol{x} = \begin{bmatrix} x_x & \dot{x}_x & \varphi_x & \dot{\varphi}_x \end{bmatrix}$

$$\boldsymbol{f}(\boldsymbol{x}, \tau_x) = \begin{bmatrix} \dot{x}_x \\ \frac{5}{7}\left(x_x \dot{\varphi}_x^2 - g\sin(\varphi_x)\right) \\ \dot{\varphi}_x \\ \frac{\tau_x - 2m_b x_x \dot{x}_x \dot{\varphi}_x - m_b g x_x \cos(\varphi_x)}{m_b x_x^2 + J_b + J_p} \end{bmatrix} \tag{5.14}$$

(iii)

$$u = \frac{\tau_x - 2m_b x_x \dot{x}_x \dot{\varphi}_x - m_b g x_x \cos(\varphi_x)}{m_b x_x^2 + J_b + J_p} \tag{5.15}$$

(iv)

$$\begin{aligned} y &= x_1 \\ \dot{y} &= \dot{x}_1 = x_2 \\ \ddot{y} &= \dot{x}_2 = \frac{5}{7}\left(x_1 x_4^2 - g\sin(x_3)\right) \\ y^{(3)} &= \frac{5}{7}\left(\dot{x}_1 x_4^2 + 2x_1 x_4 \dot{x}_4 - g\dot{x}_3\cos(x_3)\right) = \underbrace{\frac{5}{7}\left(x_2 x_4^2 - g x_4 \cos(x_3)\right)}_{=a(\boldsymbol{x})} + \underbrace{\frac{10}{7} x_1 x_4}_{=b(\boldsymbol{x})} u. \end{aligned}$$

No exact input-output linearization can be found for the system, since the relative degree of $r = 3$ is not well defined at $x_1 = 0$ and $x_4 = 0$.

Parameter	**Symbol**	**Value**	**Unit**
gravitational acceleration	g	9.81	$\mathrm{kgm/s^2}$
mass of the ball	m_b	0.0347	kg
radius of the ball	r_b	0.015	m
balls moment of inertia	J_b	3.12×10^{-6}	$\mathrm{kgm^2}$
plates moment of inertia	J_p	8×10^{-4}	$\mathrm{kgm^2}$

Table 5.1: Parameters of the ball and plate experiment.

5.2 Approximative input-output linearization based trajectory tracking

In case the relative degree is not well defined, no suitable input-to-state linearization can be obtained in those critical points. Since the ball and beam system is not suitable for exact input-to-state linearization (its derivation is left to the reader) a different method has to be developed to enable trajectory tracking. One solution to overcome this issue is to find an approximate input-output linearization, which will be addressed subsequently. The general idea is to slightly modify the drift or input vector field to bypass a relative degree which is not well defined. Finally, the control law can be formulated based on the approximated dynamics. However it is crucial, that these approximated vector fields must be close to the original system [1].

Exercise 5.2.

(i) Modify the input vector field: Drop the term $L_{\boldsymbol{g}} L_{\boldsymbol{f}}^{r-1} y$ from the calculations made in task 5.1 and proceed with the calculation of the relative degree. Is it well defined?

(ii) *Modify the drift vector field: Drop the critical term from $L_f^2 y$ in the calculations made in task 5.1, that causes the issue and proceed with the calculation of the relative degree. Is it well defined?*

(iii) *Linearize the system (around $\mathbf{0}$) whenever a nonlinear term comes up. Calculate the relative degree. Is it well defined?*

(iv) *Implement a feedback controller to track a target trajectory using all three approaches. To simplify the implementation all states from the governing system can be accessed.*

Solution 5.2. *(i)*

$$
\begin{aligned}
y &= x_1 \\
\dot{y} &= \dot{x}_1 = x_2 \\
\ddot{y} &= \dot{x}_2 = \frac{5}{7}\left(x_1 x_4^2 - g\sin(x_3)\right) \\
y^{(3)} &= \frac{5}{7}\left(\dot{x}_1 x_4^2 + 2x_1 x_4 \dot{x}_4 - g\dot{x}_3\cos(x_3)\right) \Rightarrow \frac{5}{7}\left(x_2 x_4^2 - g x_4 \cos(x_3)\right) + \cancel{\frac{10}{7} x_1 x_4} u \\
y^{(4)} &= \frac{5}{7}\left(\dot{x}_2 x_4^2 + 2x_2 x_4 \dot{x}_4 - g\dot{x}_4\cos(x_3) + g x_4 \dot{x}_3 \sin(x_3)\right) \\
&= \frac{5}{7}\left(\frac{5}{7}\left(x_1 x_4^2 - g\sin(x_3)\right)x_4^2 + 2x_2 x_4 u - g u\cos(x_3) + g x_4^2 \sin(x_3)\right) \\
&= \underbrace{\frac{5}{7}x_4^2\left(\frac{5}{7}x_1 x_4^2 + \frac{2}{7}g\sin(x_3)\right)}_{=a(\boldsymbol{x})} + \underbrace{\frac{5}{7}\left(2x_2 x_4 - g\cos(x_3)\right)}_{=b(\boldsymbol{x})} u
\end{aligned}
$$

(ii)

$$
\begin{aligned}
y &= x_1 \\
\dot{y} &= \dot{x}_1 = x_2 \\
\ddot{y} &= \dot{x}_2 = \frac{5}{7}\left(x_1 x_4^2 - g\sin(x_3)\right) \Rightarrow -\frac{5}{7}g\sin(x_3) + \cancel{\frac{5}{7}\left(x_1 x_4^2\right)} \\
y^{(3)} &= -\frac{5}{7}g\dot{x}_3\cos(x_3) = -\frac{5}{7}g x_4\cos(x_3) \\
y^{(4)} &= \frac{5}{7}g x_4 \dot{x}_3 \sin(x_3) - \frac{5}{7}g\dot{x}_4\cos(x_3) = \underbrace{\frac{5}{7}g x_4^2 \sin(x_3)}_{=a(\boldsymbol{x})} + \underbrace{\left(-\frac{5}{7}g\cos(x_3)\right)}_{=b(\boldsymbol{x})} u
\end{aligned}
$$

(iii)

$$
\begin{aligned}
y &= x_1 \\
\dot{y} &= \dot{x}_1 = x_2 \\
\ddot{y} &= \dot{x}_2 = \frac{5}{7}\left(x_1 x_4^2 - g\sin(x_3)\right) \Rightarrow -\frac{5}{7}g x_3 \\
y^{(3)} &= -\frac{5}{7}g\dot{x}_3 = -\frac{5}{7}g x_4 \\
y^{(4)} &= -\frac{5}{7}g\dot{x}_4 = \underbrace{\left(-\frac{5}{7}g\right)}_{=b(\boldsymbol{x})} u
\end{aligned}
$$

(iv) See matlab_solutions

Remark 5.1

Use

$$x_x(t) = 0.04\cos\left(\frac{2\pi}{4}t\right) \tag{5.16}$$

as the target trajectory.

5.3 Flatness based trajectory tracking

Since some dynamic parameters (like the moments of inertia and the moment created by the servo motors) are unknown, the system description will be reduced. Therefore small angular velocities $\dot{\varphi}_x \ll 1$ and $\dot{\varphi}_y \ll 1$ are assumed, which results in

$$\begin{aligned} \ddot{x}_x &= -\frac{5}{7} g \sin(\varphi_x) \\ \ddot{x}_y &= -\frac{5}{7} g \sin(\varphi_y). \end{aligned} \tag{5.17}$$

The reduced system (5.17) basically describes the movement of a ball on an inclined plane in x and y direction. The angles along the x- and y-axes can be directly set by the six servo motors, so that the system input is given by $\boldsymbol{u} = [\varphi_x, \varphi_y]$. It can be observed that the axes are decoupled, so that it is sufficient to consider two separate ball on beam systems.

Remark 5.2

Due to the decoupling, only the motion along the x-axis is considered in the following tasks. The methods are directly applicable to the y-axis. However note, that both directions are necessary for the real experiment.

Exercise 5.3. *Observer and linear control design* ***along one axis***

(i) Transfer the system to state state space and linearize the reduced system around a set point $(\boldsymbol{x}_R, u_R)$. The system descriptions

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u) \qquad \text{and} \qquad \dot{\boldsymbol{x}} = A\boldsymbol{x} + Bu, \tag{5.18}$$

respectively. Please note that ***only one axis*** *should be taken into account in all tasks of 5.3*

(ii) Show that the linearized system is observable and controllable with $y = x_x$ and $u = \varphi_x$.

(iii) Discretize the nonlinear system dynamics using explicit euler scheme and implement it in Simulink. Design a discrete-time extended Kalman Filter for the reduced system to estimate the full state. Evaluate the performance of the observer by comparing the estimated states with the actual states of the reduced nonlinear system. Choose the initial condition of the observer such that $\boldsymbol{x}_0 \neq \hat{\boldsymbol{x}}_0$.

(iv) Design a linear state feedback controller to stabilize the ball around an arbitrary but fixed point. The feedback control gain can be calculated by the MATLAB Command `place`. Check if the controller (applied to the reduced nonlinear system) provides suitable stabilization around the selected position.

Solution 5.3.

(i)

$$\boldsymbol{f}(\boldsymbol{x}, u) = \begin{bmatrix} \dot{x}_x \\ -\frac{5}{7} g \sin(u) \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ -5/7g \end{bmatrix}$$

(ii)

$$\mathscr{C} = \begin{bmatrix} 0 & -5/7g \\ -5/7g & 0 \end{bmatrix} \qquad \mathscr{O} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Since both matrices have full rank the system is controllable and observable.

(iii) *See matlab_solutions.*

(iv) *See matlab_solutions.*

Exercise 5.4. *Design a flatness-based trajectory tracking controller*

$$u = a^{-1}(\boldsymbol{\xi}, v) \tag{5.19}$$

to stabilize the following process of the predefined output trajectory. In this case, u denotes the system input with the new input

$$v = x_x^{*(r)} - \sum_{i=0}^{r-1} p_i \left(x_x^{(i)} - x_x^{*(i)} \right), \tag{5.20}$$

where p_i, $i = 0, \dots, n-1$ are coefficients of a Hurwitz polynomial. The term a refers to the compensation of the nonlinearity and can be determined by [2, (4.46)] with $a = L_f^n y$.

Implement the flatness-based controller together with the target trajectory from (5.16) *in Simulink. Simulate the closed-loop system consisting of the nonlinear reduced ball on beam from exercise 5.3, the extended Kalman Filter from task (iii) and the flatness-based controller.*

Remark 5.3

Choose the eigenvalues imposed by the Hurwitz polynomial so that the controller acts as fast as possible but also keeps the input bounded by $-10° \le u \le 10°$.

Solution 5.4. *The system inversion reads $u = -\frac{7}{5g} \arcsin\left(\ddot{\xi}\right)$ with the Hurwitz polynomial $p = [20, 8]$. For the implementation see matlab_solutions.*

References

[1] John Hauser, Shankar Sastry, and Petar Kokotović. „Nonlinear control via approximate input-output linearization: The ball and beam example“. In: *Automatic Control, IEEE Transactions on* 37 (Apr. 1992), pp. 392–398.

[2] T. Meurer. „Nonlinear Control Systems“. In: `https://www.control.tf.uni-kiel.de/en/teaching/summer-term/nonlinear-control-systems` (2020).

5.4 Additional tasks

If the steward platform is not perfectly aligned (orthogonal to the gravitational axis), it will lead to incorrect tracking behavior. Since the flatness-based controller has been setup for the x-direction of (5.17) it can be clearly seen, that a constant model error $\Delta\varphi_x$ of the plates surface angle will lead to poorer performance. This issue, the generation of reference trajectories as well as the implementation on the real system will be covered in these additional tasks.

The constant angular error, which leads to poorer trajectory tracking behavior, can be addressed in different ways. Either the angle error could be estimated or the observation that a constant angle error results in a constant positioning offset $y = x_x + \Delta x(\Delta\varphi_x)$ can be used. In this laboratory, we will focus on the second approach.

Exercise 5.5.

(i) *Add an artificial constant angle error to your reduced system dynamics in order to be able to check the accuracy of the designed controller. Evaluate the tracking behavior of the linear state feedback and the flatness-based controller form task 5.3 and 5.4, respectively.*

(ii) *Implement a PI-controller to stabilize the ball around an arbitrary set point. Evaluate the performance, when the PI controller is applied in the uncertain environment*

(iii) *Extend the idea of integrating the tracking error to the flatness-based controller from task 5.4. Implement an extended flatness-based trajectory tracking controller and evaluate its performance applied in the uncertain environment.*

Exercise 5.6.

(i) *Implement a user defined matlab function to create 2D Lissajous-figures, depending on x- and y-oriented amplitudes and frequencies. Display the results in a XY-Graph window.*

(ii) *Add the y-direction to regain the ball and plate setup and connect the flatness-based controller from task 5.5 to track the created 2D trajectory with and without error angle.*

Exercise 5.7. *Evaluate the flatness-based tracking controler controller at the real ball and plate setup. Let your supervisor explain you the hardware and software and how to operate the setup.*

> **Remark 5.4**
> Export your Simulink model to MATLAB version 2017b.

References

[1] John Hauser, Shankar Sastry, and Petar Kokotović. „Nonlinear control via approximate input-output linearization: The ball and beam example“. In: *Automatic Control, IEEE Transactions on* 37 (Apr. 1992), pp. 392–398.

[2] T. Meurer. „Nonlinear Control Systems“. In: `https://www.control.tf.uni-kiel.de/en/teaching/summer-term/nonlinear-control-systems` (2020).