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 (5.10) 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 the additional tasks we will address both approaches.

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.

Joint state and parameter estimation

Exercise 5.6.

(i) Read Chapter 4.3 of the identification and estimation script from our website

(ii) Extend the reduced model to use a parameter $\Delta\varphi_x$.

(iii) Extend the Kalman Filter from task 5.3 for joint state and parameter estimation. Evaluate the approach together with both the state feedback controller from the preparation task and your controller from task 5.5.

Exercise 5.7.

(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 5.6 to track the created 2D trajectory with and without error angle.

Exercise 5.8.

(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 5.6 to track the created 2D trajectory with and without error angle.*

Exercise 5.9. *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).