Lab report

Advanced Control Engineering I (MECH-M-1-RTV-RTV-LB)

Ball in Tube

Master program - Mechatronics

4^{th} semester

Lecturer: Kronthaler Phillip

Group: MA-MECH-24-VZ-2A

Authors: Gabriel Brugger, Jim Buso

February 6, 2025

Contents

1 Learning Target

In this Lab, the primary goal is on developing a deep understanding of position controlling a floating ping-pong ball in a plexiglass tube with an upward flow. The system should be able to bring the ball to a desired position, hold its position and withstand disturbances. A trajectory should be used to insure a smooth enough change in height. The controller design should be implemented in Matlab/Simulink and auto-code generated to a Beckhoff PLC. [1]

2 Preparation Work

2.1 PARAMETERS

All used parameters in this lab report are introduced in the table ??

Name	Symbol	Value	Unit
Area ball	A_B	13	cm^2
Mass of ball	m	2.7	g
Area tube	A_T	15	mm^2
Drag coefficient	k_L	2×10^{-5}	m^3
Air flow coefficient	k_V	1.27×10^{-4}	$\mathrm{kg\ m}^{-1}$
Fan time constant	τ_ω	1.6	s
Fan gain	k_ω	28	$\mathrm{s}^{-1}\mathrm{V}^{-1}$
Density of air	ρ_L	1.225	$\mathrm{kg\ m}^{-3}$
Gravity constant	g	9.81	$\mathrm{m\ s}^{-2}$
Length of tube	h_{max}	0.4	m
max fan Voltage	u_{max}	5	V

Table 2.1: Parameter Table

[1]

2.2 SIMPLIFY THE MATHEMATICAL MODEL

Given from the system we get a mathematical model as the following non-linear state-space model. First the model is simplifed:

$$\ddot{z}(t) = (\alpha_1 \cdot \omega(t) - \alpha_2 \dot{z}(t))^2 - \alpha_3 \tag{1}$$

$$\tau_\omega \dot{\omega}(t) + \omega(t) = k_\omega u(t) \tag{2}$$

[1]

Therefoe the coefficients α_1, α_2, α_3, τ_ω and k_ω are used.

$$\alpha_1 = \sqrt{\frac{k_l}{m}} \frac{k_v}{A_{Gap}} \tag{3}$$

$$\alpha_2 = \sqrt{\frac{k_l}{m}} \frac{A_B}{A_{Gap}} \tag{4}$$

$$\alpha_3 = g \tag{5}$$

$$\tau_\omega = -\frac{JR}{bR + k_m^2} \tag{6}$$

$$k_\omega = -\frac{k_m}{bR + k_m^2} \tag{7}$$

2.3 LINEARIZE THE NON-LINEAR STATE-SPACE MODEL

For a more straightforward PID controller design the non-linear state space model is linearized at an operating Point $\overline{z}$.

$$
\begin{aligned}
&x_1 = z, \quad x_2 = \dot{z}, \quad x_3 = \ddot{z} = \omega && (8)\\
&\dot{x}_1 = \dot{z} = x_2 && (9)\\
&\dot{x}_2 = \ddot{z} = \alpha_1^2 x_3^2 - 2\alpha_1 x_3 \alpha_2 x_2 + \alpha_2^2 x_2^2 - \alpha_3 && (10)\\
&\dot{x}_3 = \dot{\omega} = \frac{k_\omega}{\tau_\omega} u(t) - \frac{1}{\tau_\omega} x_3 && (11)\\
& && (12)
\end{aligned}
$$

I. We determine a stationary point:

$$\overline{x_1} = \overline{z} = 15\text{cm} \quad \Rightarrow \quad \overline{\dot{x}_1} = \overline{x_2} = \overline{\dot{z}} = 0\text{cm} \quad \Rightarrow \overline{\dot{x}_2} = \overline{\ddot{z}} = 0, \quad \Rightarrow \overline{\dot{x}_3} = \overline{\dot{\omega}} = 0 \tag{13}$$

II. First order taylor series expansion of f:

$$f(x_2, x_3) \approx f(\overline{x_2}, \overline{x_3}) + \frac{\partial f(\overline{x_2}, \overline{x_3})}{\partial x_2}(x_2 - \overline{x_2}) + \frac{\partial f(\overline{x_2}, \overline{x_3})}{\partial x_3}(x_3 - \overline{x_3}) \tag{14}$$

III. Introducing Δ:

$$
\begin{aligned}
\Delta x_2 &= x_2 - \overline{x_2} = \dot{z} - \overline{\dot{z}} = x_2 && (15)\\
\Delta x_3 &= x_3 - \overline{x_3} && (16)\\
\Delta \dot{x} = \dot{x} - \overline{\dot{x}} &= \frac{\partial f(\overline{x_2}, \overline{x_3})}{\partial x_2}(x_2 - \overline{x_2}) + \frac{\partial f(\overline{x_2}, \overline{x_3})}{\partial x_3}(x_3 - \overline{x_3}) && (17)\\
&= A\Delta x + B\Delta u \quad \Rightarrow \text{ for } \Delta x,\ \Delta x_2 \text{ and } \Delta x_3 \text{ is used.} && (18)
\end{aligned}
$$

IV. Calculating the partial differential:

$$
\begin{aligned}
\frac{\partial f}{\partial x_2} &= -2\alpha_1\alpha_2 x_3 + 2\alpha_2^2 x_2 = -2\alpha_1\alpha_2 \frac{\sqrt{\alpha_3}}{\alpha_1} = -2\alpha_2\sqrt{\alpha_3} && (19)\\
\frac{\partial f}{\partial x_3} &= 2\alpha_1^2 x_3 - 2\alpha_1\alpha_2 x_2 = 2\alpha_1^2 \frac{\sqrt{\alpha_3}}{\alpha_1} = 2\alpha_1\sqrt{\alpha_3} && (20)\\
& && (21)
\end{aligned}
$$

V. The non-linear state space:

$$
\begin{aligned}
\vec{x} &= \begin{bmatrix} \Delta z \\ \Delta \dot{z} \\ \Delta \omega \end{bmatrix}, \quad \dot{\vec{x}} = \begin{bmatrix} \Delta \dot{z} \\ \Delta \ddot{z} \\ \Delta \dot{\omega} \end{bmatrix} && (22)\\
\dot{\vec{x}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2\alpha_2\sqrt{\alpha_3} & 2\alpha_1\sqrt{\alpha_3} \\ 0 & 0 & -\frac{1}{\tau\omega} \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 0 \\ \frac{k_\omega}{\tau_\omega} \end{bmatrix} u(t) && (23)\\
y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \vec{x} && (24)
\end{aligned}
$$

2.4 PID CONTROL DESIGN

The PID controller values are chosen using the linearized state space. With the matlab PIDTuner function, the following values are found: Kp = 1,5; Ki = 0,5; Kd = 1,118.

2.5 SIMULATE CLOSED LOOP SYSTEM

The closed loop system can be seen in the figure 2.1

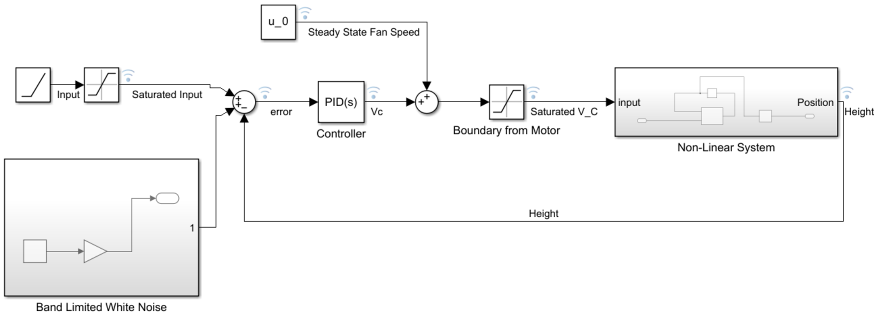


Figure 2.1: Image of the closed loop system in simulink

To counteract the integral term of the PID controller and minimize excessive overshoot, a ramp input instead of a step input is used. The steady-state fan speed is added, due to the linearization, and by that it can be ensured that the fan is spinning with the steady state fan speed ω_0. Furthermore, the initial states are chosen to be a staring position of $0\,\mathrm{m}$ and start velocity of $0\,\mathrm{m\,sec^{-1}}$. This results in an initial states of the integrator as $[0; 0; \omega_0]$.

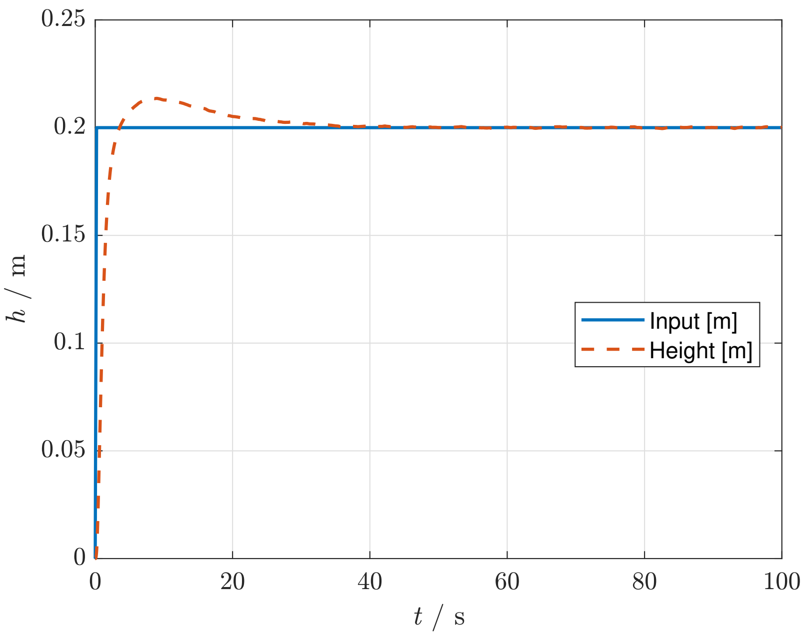


Figure 2.2: Plot of input and height over time of the closed loop System

As seen in Figure 2.2, our system wants to bring the ball from $0\mathrm{m}$ to $0.2\mathrm{m}$. The system has a small overshoot and it takes $40\mathrm{sec}$ to reach the desired height. This can surely be improved in further steps.

2.6 FLATNESS BASED FEED FORWARD

Using the non-linear state-space from equations 1 and 2. Using a constant operation point for the height $z(t)$. All states and inputs can be represented by z, so z is a flat output.

$$u = \frac{1}{k_\omega}(\tau\dot{\omega} + \omega) \tag{25}$$

$$\text{with: } \omega = \frac{\sqrt{\ddot{z} + \alpha_3} + \alpha_2 \dot{z}}{\alpha_1} \tag{26}$$

$$\text{and: } \dot{\omega} = \frac{0.5 \frac{1}{\sqrt{\ddot{z}+\alpha_3}} z^{(3)} + \alpha_2 \ddot{z}}{\alpha_1} \tag{27}$$

With $\omega = \ddot{z}$ and $\dot{\omega} = z^{(3)}$, $u(\dot{z}, \ddot{z}, z^{(3)})$. A trajectory must be continuously differentiable at least as many times as the highest derivative of the flat output that appears in the system dynamics. The input u is explained with the third derivative of z, so the trajectory must be three times continuously differentiable.
The reference trajectory and the feed-forward signal are both implemented in the matlab script and in the simulink simulation 2.3.

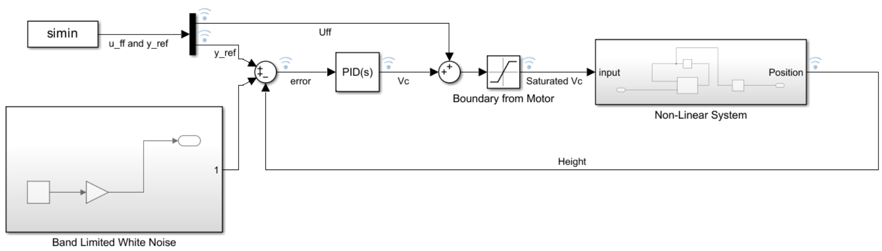
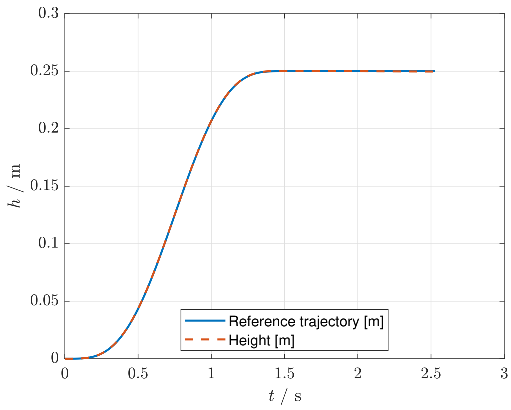
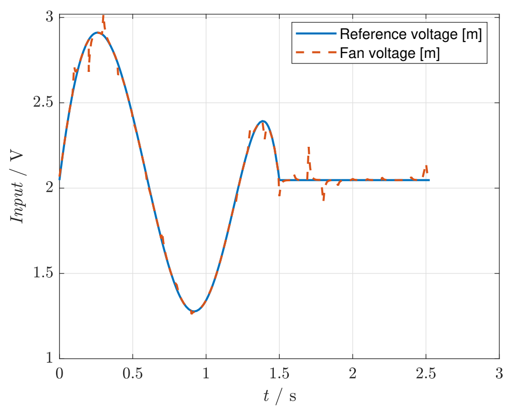


Figure 2.3: Image of the flatness based feed forward system in simulink

Figure 4(a) shows a plot of the reference trajectory and the position of the ball over time for the flatness-based feed forward approach. The transit duration T is chosen to be $1.5\,\mathrm{sec}$, start position $y_0 = 0\,\mathrm{m}$ and end position $y_{end} = 0.25\,\mathrm{m}$.
Figure 4(b) shows a plot of U_{ff} and V_s over time for the flatness-based feed forward approach.
As seen, both the motor voltage and the height are perfectly following our calculated values.



((a)) Plot of the reference trajectory and the position of the ball over time for the flatness-based feedforward approach.



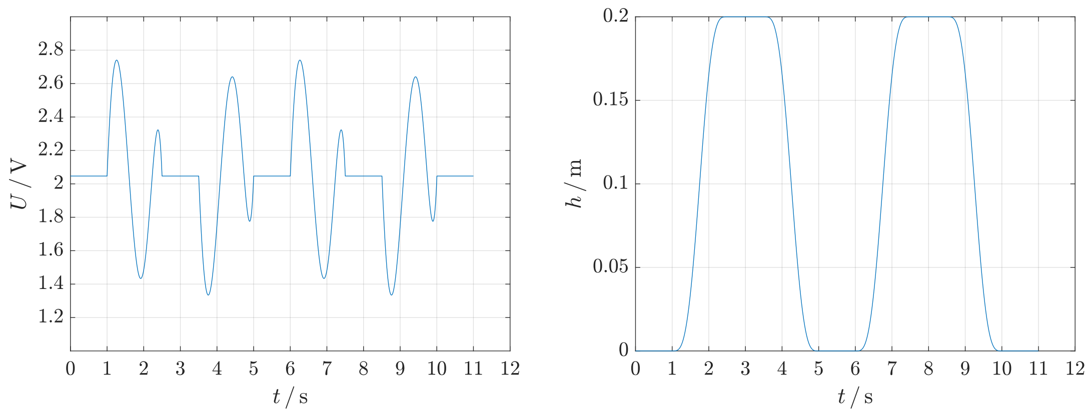
((b)) Plot of U_{ff} and V_s over time for the flatness-based feedforward approach.

3 Laboratory Task

As the laboratory setup unfortunately did not work the task of the lab was set to toggle the ball between two positions in the simulation. This is done by MATLAB and Simulink.

3.1 SIMULATION

To toggle the ball between the two different heights $y_0 = 0\,\mathrm{m}$ and $y_{end} = 0.2\,\mathrm{m}$, the reference trajectory is modified in such a way that the ball moves up and down with the transition time $T = 1.5\,\mathrm{sec}$ and breaks of $1\,\mathrm{sec}$. To get the reference trajectory, the prototype function is calculated for both rising and falling individually and respectively also the reference height y_{ref} and the reference voltage u_{ref}, which can be seen in the following figures 1(a) and 1(b).



((a)) reference voltage ((b)) reference height

Figure 3.1: reference trajectories

The SIMULINK model is still the same as shown in 2.3 as the system itself does not change at all. Just the input, which are the reference height y_{ref} and the reference voltage u_{ref} are changed.

3.2 RESULT AND INTERPRETATION

With the SIMULINK model the input of the toggling trajectory is simulated.

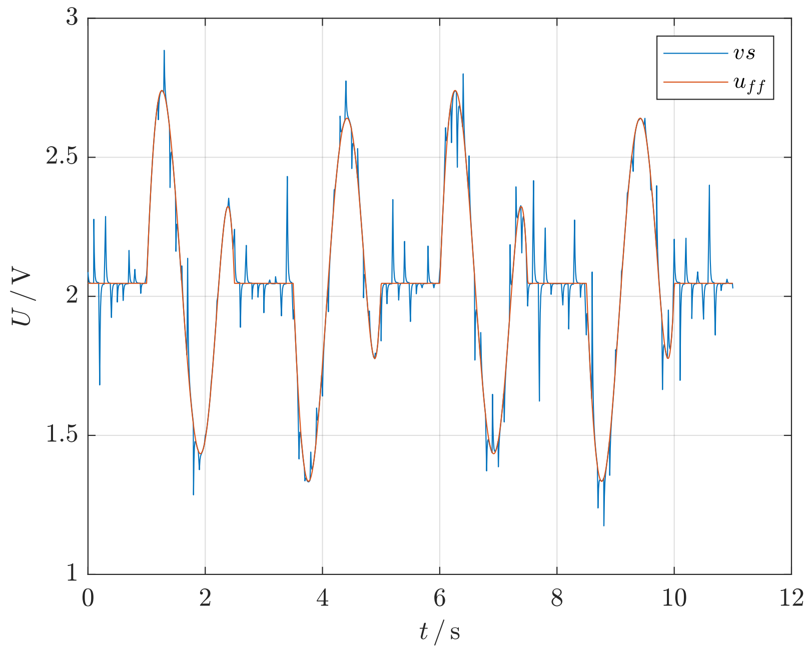


Figure 3.2: comparison vs and u_{ff}

As shown in figure 3.2, the reference voltage of the feed forward u_{ff} meets the voltage vs, which is the voltage input to the system so including the PID, quite accurately. There are just some small deviations, which are due to the added noise.

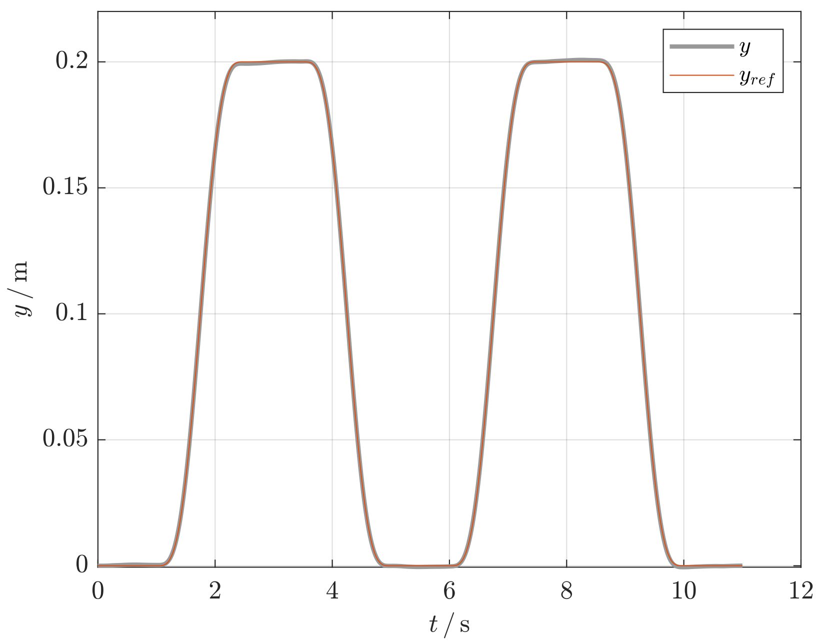


Figure 3.3: comparison y and y_{ref}

Figure 3.3 shows that the output y and the reference y_{ref} are also very close to each other. The maximum error is $0.0031\,\mathrm{m}$, which is coming again from the added noise.

4 Conclusion

Summarized can be said that unfortunately it was not possible to test the system in the laboratory setup, but the simulation already showed that the implementation of a flatness based feed forward controller is a very good approach to control a system. The flatness based feed forward itself is already a good solution but in combination with a PID it is very good as seen in the results.

5 Instruction to run files

To run the simulation in MATLAB either run the file „Ball_in_Tube.m“ for the ball just to fly up to a certain position or the file „Ball_in_Tube_toggling.m“ for the ball to toggle between two positions. Both files work then with the simulation „BiT_Simulation.slx“.

List of Figures

References

[1] K. Phillip, “Advanced Control Engineering I: Ball in Tube,” MCI, 2024.