Automatic Control Laboratory, ETH Zürich
Prof J. Lygeros

Manual prepared by: R. Nguyen, S. Richter
Revision from: September 8, 2021

IfA Fachpraktikum - Experiment 2.7 :

Air Ball

- In this experiment the height of a ball suspended in an air tube will be controlled. A fan at the bottom of the tube causes upward airflow that pushes the ball up to counteract the downward force of gravity. The fan speed can be controlled to change the air stream velocity, causing a change in ball height. A PID controller will be designed to follow reference trajectories of the ball height and reject disturbances.
- You will learn the basics of PID control and understand the effects of changing the controller gains.

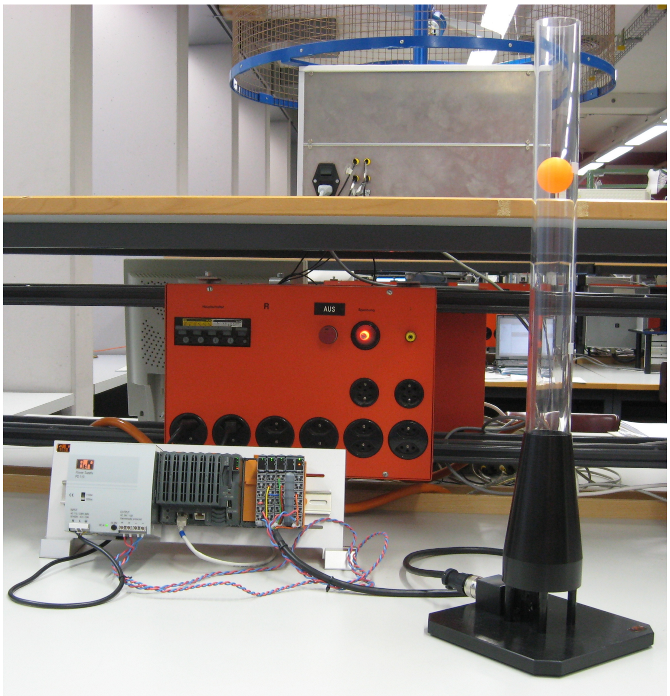

Figure 1: Experimental setup for air ball

Contents

Chapter 1

Problem Setup and Notation

Apart from the mechanical components of the experiment its setup incorporates

DC motor and fan Mounted at the bottom of the airball

Fan speed measurement unit Consisting of an optical sensor reacting on reflections from a black/white paper applied directly on the fan (*have a look at it when you are in the lab*)

Ultrasonic distance sensor Measuring the height of the ball

The control unit is an industrial standard B&R X20CP1484 equipped with a 266MHz Intel Celeron processor with 32 MB RAM. Its interface modules are

Digital Input Module (X20DI2377) This module counts the time between a rising edge and a falling edge of signals coming from the optical speed sensor. By doing so, the time for half of a rotation of the fan is measured and the fan speed deduced from it.

Analog Input Module (X20AI2622) The analog module converts the output voltage of the ultrasonic sensor into a digital number using an analog-to-digital converter.

Motor Control Module (X20MM2436) The motor module drives the DC motor by a pulse-width modulated (PWM) voltage signal. Its duty cycle can be changed manually or automatically by the controller.

For the actual control task the following configuration is used:

- The sampling time is 1 ms, i.e. sensor measurements and control law evaluations take place at a rate of 1 kHz.
- Fan speed measurements turn out to be error prone, meaning, that (rare) spikes degrade the quality of the measurements. Thus, a second order filter is implemented internally to smooth out the measurement signal.

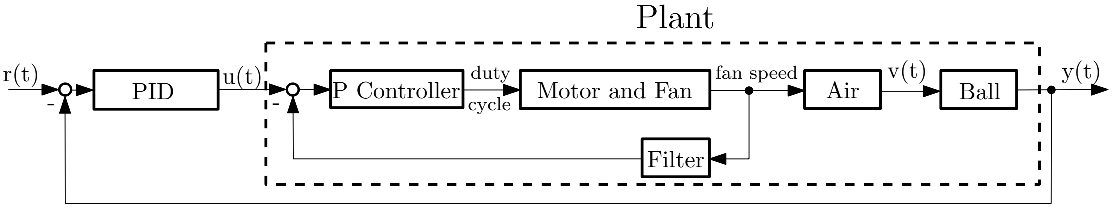

Figure 1.1: Schematic control structure of the air ball

In this experiment, you will mainly consider the control of the ball's height by means of a PID controller. According to Fig. 1.1 which depicts the block diagram of the *overall* air ball's control structure this constitutes only a part of the implemented control.

As shown in the figure, there is another nested control loop (*inner* loop) that controls the speed of the fan using a static proportional controller. Also shown is the second order spike filter. The 'Plant' to control for the PID controller in the *outer* loop then consists of the inner loop in series with the block labeled as 'Air' (stands for the (nonlinear and in this experiment assumed unknown) function relating the fan's speed to air speed) and the block labeled as 'Ball'. The 'Ball' block relates air speed v to the ball's height y and its mathematical model will be derived as an exercise in Chapter 2. From the mismatch between the ball's height and the given reference height r, the PID then computes a control input u (having the meaning of a reference speed for the fan) to make this mismatch vanish. Note that the overall control structure is a so-called *cascaded control structure* which is often used in practice.

Chapter 2

Preparation@Home

You will derive a simple dynamic model of the 'Ball' block in Fig. 1.1. For this, we will consider the ball within an air-stream, neglect the comparatively small buoyancy force, and assume high Reynold's number (> 1000). The Reynold's number, Re, is a dimensionless number that gives a measure of the ratio of the inertial forces to viscous forces. It quantifies the relative importance of these two types of forces. The notation used is illustrated in Figure 2.1. The two forces acting on the ball are gravity, F_g, and the aerodynamic drag force, F_a, which for high Reynold's number is calculated as

$$F_a = \frac{1}{2}(v - \dot{y})^2 \rho_g C_d A, \tag{2.1}$$

where

v	is the velocity of the air stream
$\dot{y}$	is the velocity of the ball
ρ_g	is the density of air (1.0 kg/m^3)
C_d	is the drag coefficient
A	is the cross section of the ball.

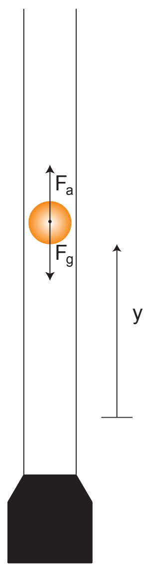

Figure 2.1: Notation used for modeling

2.1 Modeling

Assume the ball's mass is 2.7 g and its radius is 0.020 m and that the drag coefficient is 0.50.

Task 1: Force balance

Write down the force balance equation for the simplified ball model discussed before.

Task 2: Steady state

Find the steady state air velocity (for which $\dot{y}$ and $\ddot{y}$ are zero). Note that it is independent of the ball's height under the simplified assumptions.

Task 3: Verification of high Reynold's number

Verify the assumption of high Reynold's number at the steady state air velocity. Reynold's number, Re, is calculated as

$$\mathrm{Re} = \frac{\rho_g V L}{\mu},$$

where

V	is the mean velocity
L	is a characteristic length, such as the diameter of the ball
μ	is the dynamic viscosity of the fluid, which is 1.78×10^{-5} kg m^{-1} s^{-1} for air.

Task 4: Linearization at Steady State

Notice that the model you have derived so far is nonlinear. Since we control the ball's height around the steady-state (y, v_0), it makes sense to derive a linear model around this state. So, linearize the model for small deviations in v and y around steady state and formulate the transfer function.

Task 5: Stability

Find the roots of the transfer function and comment on the stability of the system.

As seen from Fig. 1.1, we are not able to control the air's speed v directly. Instead, you will be changing the speed of the fan at the bottom of the tube through the inner control loop, by letting the PID controller determine a reference fan speed $u(t)$. Furthermore, the PID controller already takes into account the steady state air velocity v_0 by adding a feedforward term to the input $u(t)$ (this is the $u_0(t)$ term in Figure 2.2).

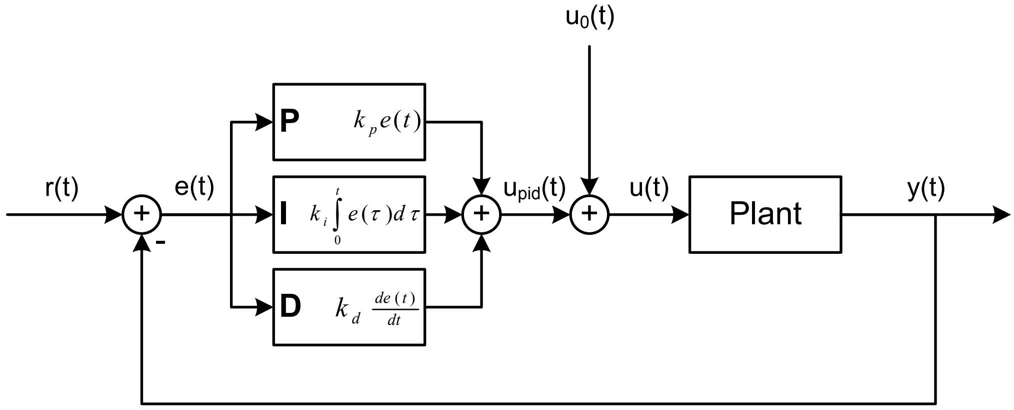

Figure 2.2: Outer PID loop control block diagram. Recall that $u_0(t)$ is the feedforward term required to keep the ball in steady state according to the steady state air speed v_0.

2.2 PID Control

A proportional-integral-derivative (PID) controller is a common feedback loop component in industrial control systems. The controller compares a measured value from a process with a reference setpoint value. The mismatch is then used to calculate a new value for a manipulable input to the process that drives the process' measured value towards the desired setpoint.

PID is named after its three correcting calculations (refer to Equation (2.2) and Figure 2.2), which are summed to obtain the control input:

- Proportional - To handle the present, the error is multiplied by a constant k_p. This term has the tendency to bring the output towards the setpoint (reference value). Note that when the error is zero, the proportional component is also zero. However, be warned that in general, proportional control alone will not cause the system to settle at the reference value resulting in a steady state error.
- Integral - To handle the past, the error is integrated (added up) over a period of time, and then multiplied by a constant k_i, and added to the controlled quantity. This term is proportional to both the magnitude and duration of the error, and eliminates the residual steady-state error that can occur with only proportional control.
- Derivative - To handle the future, the first derivative (the slope of the error) over time is calculated, and multiplied by another constant k_d. The derivative term can decrease the rate of change of the controller output, reducing the magnitude of the overshoot. However, this term is highly sensitive to measurement noise!

To be more precise, we first define the input to the plant as $u(t)$, output of the plant as $y(t)$, and the desired reference value as $r(t)$. We also define the error as $e(t) = r(t) - y(t)$ (see Figure 2.2). Then, the PID controller has the following, ideal parallel form:

$$u_{pid}(t) = \underbrace{k_p e(t)}_{\text{proportional}} + \underbrace{k_i \int_0^t e(\tau)\, d\tau}_{\text{integral}} + \underbrace{k_d \frac{de(t)}{dt}}_{\text{derivative}}, \tag{2.2}$$

where k_p, k_i, and k_d are constants that are used to tune the PID control loop.

As a general guideline, the effects of changing the gains are summarized below.

- k_p, proportional gain: Larger values typically mean faster response since the larger the error, the larger the feedback to compensate.
- k_i, integral gain: Larger values eliminate steady state error quicker. The trade-off is larger overshoot and settling time because any positive error integrated during a transient response must be integrated away by negative error before steady state is reached.
- k_d, derivative gain: Larger values decrease overshoot, but, due to high sensitivity to noise in the error term, can cause a process to become unstable. To prevent such behaviour, sometimes a "Dirty D" is implemented instead, which does not amplify high frequency noise components. The transfer function of a "Dirty D" is

$$k_{dd}\frac{Ts}{Ts+1}, \tag{2.3}$$

where k_{dd} and T are design parameters.

2.3 Controller tuning

2.3.1 Manual tuning method

One way to tune the gains of a PID controller is to use knowledge of the effects of changing the gains. These effects are summarized in Table 2.1 and illustrated in Figures 2.3 to 2.5.

Table 2.1: Effect of *increasing* gains (see Section 2.4 for definition of performance measures)

Parameter	Rise Time	Overshoot	Settling Time	Steady State Error
k_p	decrease	increase	small change	decrease
k_i	decrease	increase	increase	eliminate
k_d	small change	decrease	decrease	none

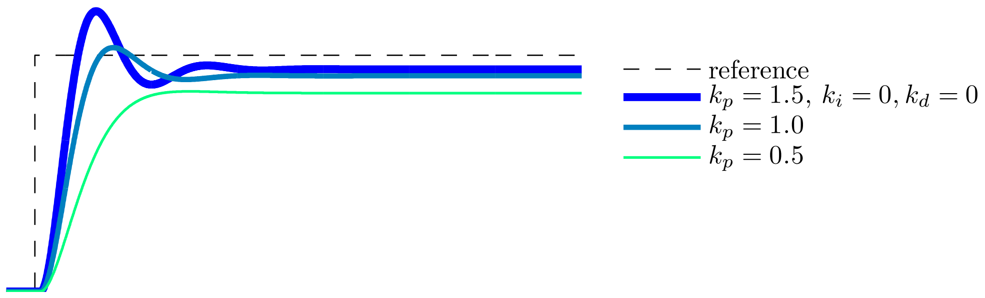

Figure 2.3: Closed loop step responses with different PID controllers, showing the effect of changing k_p.

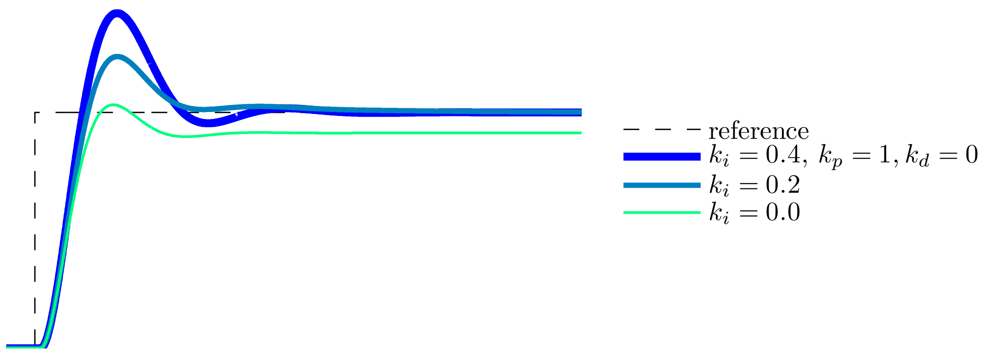

Figure 2.4: Closed loop step responses with different PID controllers, showing the effect of changing k_i.

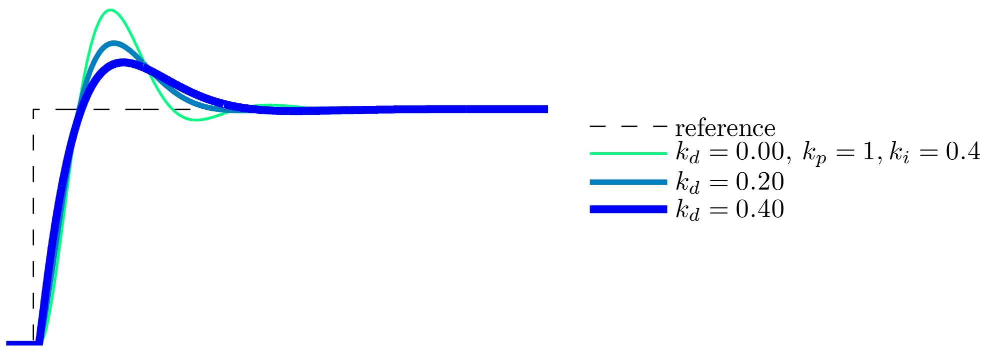

Figure 2.5: Closed loop step responses with different PID controllers, showing the effect of changing k_d.

2.3.2 Ziegler-Nichols closed-loop tuning method

A well-known systematic tuning method is the Ziegler-Nichols method, introduced by John G. Ziegler and Nathaniel B. Nichols in 1942. The objective is to obtain a controller that gives a closed loop response that is “quarter decay”, which means the ratio between the second peak and first peak is a quarter. Furthermore, although the method assumes that the plant transfer function has the following form:

$$\frac{K}{s+a}e^{-sT},$$

in practice the method works well with a variety of plants.

In this experiment, a closed-loop Ziegler-Nichols tuning method will be implemented. Note however, that there also exists an open-loop method that tests the open-loop response of a system to a step input. The latter method is appropriate for systems that cannot be reliably tuned with the former method (e.g., due to infeasibility of inducing periodic output behaviour).

To implement the closed-loop Ziegler-Nichols tuning method, start by setting the k_i and k_d gains to zero and gradually increasing the k_p gain until the system just starts to oscillate continually (this is the point where the system begins to become unstable). The corresponding gain is K_u (“ultimate gain”) and the corresponding period is P_u (“ultimate period”). From this, reasonable settings can be achieved by the rules in Table 2.2. The rules are based on the following, *standard form* of the PID controller:

$$u_{pid}(t) = K_c\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right). \tag{2.4}$$

This form is equivalent to the ideal parallel form of Equation (2.2) as you can learn from the next task.

Table 2.2: Tuning gains

Type of Controller	Optimum Gain
P	$K_c = (1/2)K_u$
PI	$K_c = (1/2.2)K_u$ $T_i = (1/1.2)P_u$
PID	$K_c = (1/1.7)K_u$ $T_i = (1/2)P_u$ $T_d = (1/8)P_u$

Task 6: Equivalence of PID forms

Write the parameters of the ideal parallel form (Equation (2.2)) in terms of the parameters of the standard form (Equation (2.4)).

2.4 Time domain controller performance

Two key requirements of a good controller are reference tracking and disturbance rejection. These can be quantified in the time domain using the following measures (see Figure 2.6) that describe the response of the overall system to a step change in reference value or disturbance.

- *Overshoot*: the difference between the most extreme value and the final value, expressed as a percentage of the final value.
- *Steady-state offset*: the difference between the actual final value and the desired final value.
- *Rise time*: the time required for the output to first reach 90% of the final value.
- *Settling time*: the time required for the output to remain within ±5% of the final value.

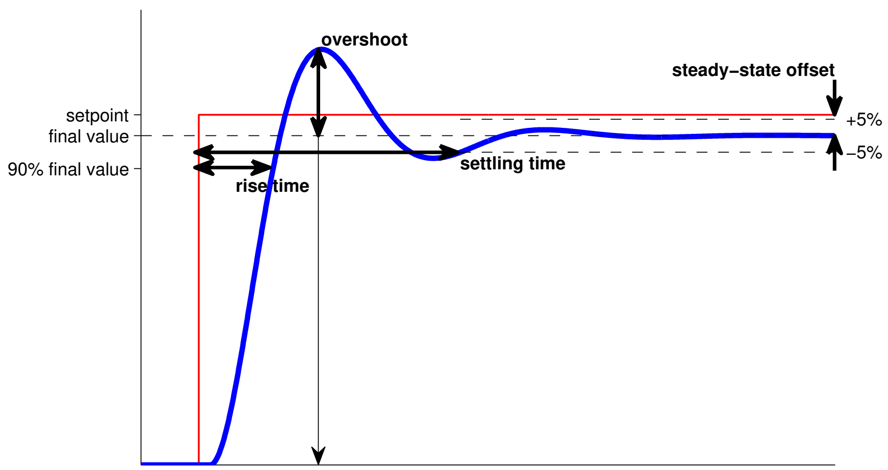

Figure 2.6: Time domain controller performance measures are shown on the system response to a step change in setpoint.

Chapter 3

Lab Session Tasks

3.1 Accessing the graphical interface to the controller

The control unit is directly connected to the inputs and outputs of the air ball device via its interface modules. A VNC server installed on the control unit can be accessed from an external computer via a remote connection. Establish this connection by executing the following:

1. Click on the "VNC Viewer" icon on the desktop.
2. Type "autx20-03" in the textbox under Server and click OK.
3. Leave the Password field blank and click OK.
4. Click on the 'Start' button of the startup screen.

3.2 Manual control

Task 7: Manual control

Click on the button "Manual Mode". Here, it is possible to set manually the duty cycle of the PWM voltage signal which is directly applied to the DC motor (see Fig. 1.1). In this experiment, it is an integer between 0 and 32767 (corresponding to fully off and fully on respectively). You can set the value either directly by first clicking into the field and then using the keyboard or by using the keyboard arrow keys to increase and decrease it in small steps. Using this mode, try to keep the ball steady at 0.0 m, which is marked on the tube. Then try to change the height to 0.2 m, which is also marked on the tube, as best as you can.

3.3 Controller tuning

A PID controller has already been implemented on the control hardware. Using the graphical interface you can manually change the controller gains under the "Setup" tab. The relevant controller gains are labeled k_p, k_i, and k_d (these parameters correspond to those in Equation (2.2)).

Task 8: Manual controller tuning

Spend some time to tune the PID gains manually based on what you know about their effects. To determine the reference tracking performance, induce a change in the reference height from 0.0 m to 0.2 m and use the sine function in the "Run" tab. To determine the disturbance rejection performance, mimic a disturbance by dropping a second ball into the tube. What is the best controller you can obtain in terms of the following performance measures for both good reference tracking and disturbance rejection: overshoot, steady-state offset, rise time and settling time. Record these values and state the gains that you used.

Important: Leave the inner loop gain $k = 1000$ unchanged for the time being.

Task 9: Ziegler-Nichols tuning

Repeat the previous task using the Ziegler-Nichols tuning method. What is the ultimate gain/period? *Hint*: Maximize the VNC client window to allow for better measurements on screen.

Task 10: Tuning of the inner loop

Use the PID controller parameters giving the best performance so far. Now, change the proportional gain k of the inner loop controlling the speed of the fan. Start with values < 1000 and observe its effect on the closed-loop performance for setpoint changes and sine reference tracking. Then increase this value to > 1000. From which value on do you observe oscillatory behaviour? Why does this happen?

3.4 Integral windup

Task 11: Integral windup

Use the best controller you have obtained so far to bring the height of the ball from 0.0 m to 0.2 m. Once settled, cover the top of the air tube. Because no air flow is possible, the ball will quickly drop to the bottom of the tube. Keep the top covered for at least 3 s, then uncover the top of the air tube. Based on your observations, you will answer the questions in Lesson Learned 4.

Chapter 4

Lessons Learned

4.1 Manual control

Lesson Learned 1: Manual control

Comment on your ability to control manually the height of the ball.

4.2 Controller tuning

Lesson Learned 2: Manual controller tuning

Comment on the ease or difficulty of tuning the controller manually.

Lesson Learned 3: Ziegler-Nichols tuning

How do the results obtained for this task compare to the previous task? Propose reasons for any discrepancies.

4.3 Integral windup

Lesson Learned 4: Integral windup

What did you notice when the air tube was uncovered (Task 11)? Why did this happen? Propose a modification that can be made to the controller to prevent this behaviour.

4.4 Completion of Experiment

Lesson Learned 5: Feedback and Testat

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. You can now discuss the lab session with the assistant to get you testat.