Jonathan J. Huerta y Munive

Models and Tools for Cyber-Physical Systems
Class sheet 1

Exercise 1: Introduction to Feedback Control

A *controller* aims to regulate a physical quantity—the *process variable* (PV)—so that it reaches or maintains a desired value called the *set-point* (*SP*). Other common names for the set-point include *reference value* and *target value*.
A generic feedback control system consists of four main components: a **plant**, an **actuator**, a **sensor**, and a **controller**.

(a) Match each component name with its description.

Description	Component
(A) Measures the process variable and reports it back to the control loop	1. Plant
(B) The physical system or process being controlled	2. Actuator
(C) Computes the control signal based on the difference between the set-point and the measured process variable	3. Sensor
(D) Applies the control action to the plant (e.g. a motor driver, a valve)	4. Controller

(b) A control system is called *open-loop* if the control action is determined without using any feedback from the plant's output. It is called *closed-loop* if the controller receives measurements from a sensor and adjusts its output accordingly.

Classify each of the following systems as open-loop or closed-loop:

(i) A toaster that heats bread for a fixed amount of time, regardless of how toasted the bread is.

(ii) A thermostat that turns a heater on when the room temperature drops below a threshold and off when it exceeds it.

(iii) A washing machine that runs a pre-programmed cycle of fixed duration.

(iv) Cruise control in a car that measures the current speed and adjusts the throttle to maintain a target speed.

Exercise 2: Controlling a Robot's Position

A small robot moves along a straight line. Its position at time t is $x(t)$ and it starts at $x(0) = 0$. We want the robot to reach a target position SP by controlling its velocity $v(t)$. A proportional controller sets the velocity in proportion to how far the robot still is from the target.

(a) Formalise: "The robot's velocity equals the rate of change of its position."

(b) Formalise: "The controller sets the velocity proportionally to the error between the set-point and the current position, with proportionality constant K_p."

(c) Combine your answers from (a) and (b) to write the ordinary differential equation (ODE) that governs the robot's position under proportional control. State the initial condition.

(d) Solve this ODE. *Hint:* substitute $e(t) = SP - x(t)$.

(e) Find the equilibria of the system. Is the equilibrium asymptotically stable? Justify your answer.

Exercise 3: Controlling an Aircraft's Yaw Rate

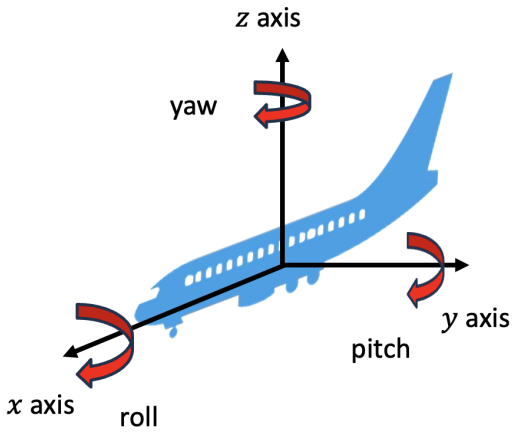

Figure 1: The three rotation axes of an aircraft: *yaw* (rotation around z—turning left or right), *pitch* (rotation around y—nose up or down), and *roll* (rotation around x—banking/tilting).

An aircraft can rotate around three axes (see Figure 1). The (simplified) objective of this exercise is to make the aircraft turn at a desired angular velocity r by applying a proportional controller.

(a) Which of the three angles (yaw, pitch, roll) is relevant when controlling left/right turning? Let us denote it by $\psi(t)$ and its angular velocity as $s(t) = \dot{\psi}(t)$.

The aircraft has moment of inertia $J > 0$ about the vertical axis. It is currently turning with angular velocity $s_0 \neq r$, and a proportional controller applies a corrective torque to bring the angular velocity to the set-point $SP = r$. Formalise each of the following statements.

(b) "The rate of change of the angular velocity equals the applied torque divided by the moment of inertia." *Hint:* this is the rotational analogue of Newton's second law.

(c) "The controller applies a torque proportional to the error between the desired angular velocity and the current angular velocity, with proportionality constant K_p."

(d) "The desired angular velocity is r (the set-point) and the aircraft initially turns with angular velocity s_0."

(e) Combine your answers from (b), (c), and (d) to write the ODE governing the angular velocity $s(t)$ under proportional control, together with the initial condition and the value of SP.

(f) Complete the block diagram in Figure 2 by writing the corresponding equations in the **controller** and **plant** boxes.

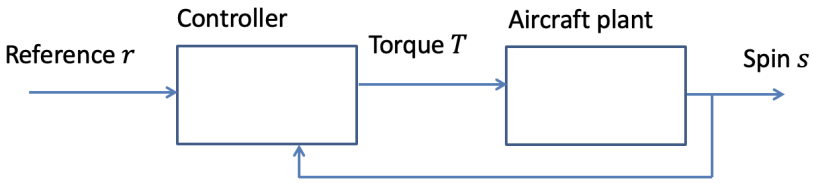

Figure 2: Block diagram of the proportional controller for the aircraft's yaw rate. Fill in the controller and plant equations.

(g) Solve this ODE.

(h) What happens if $K_p < 0$? Does the angular velocity converge to or diverge from the set-point?

(i) What happens if $K_p > 0$?

(j) What should one do to ensure faster convergence to the desired angular velocity?

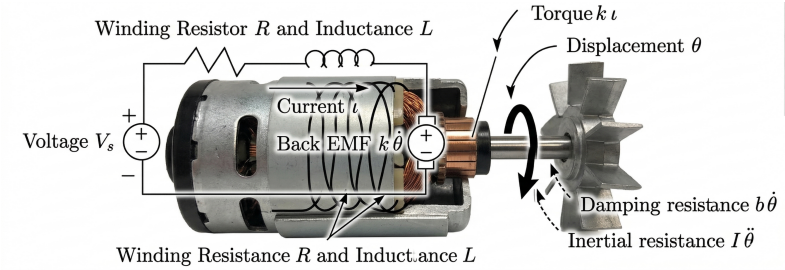

Figure 3: Electromechanical model of a DC motor. The electrical circuit represents the internal winding resistance (R) and inductance (L), while the mechanical side illustrates the conversion of current into torque to overcome rotor inertia and viscous damping.

Exercise 4: Modelling a DC Motor

We now study a more realistic control problem: regulating the rotational speed of a DC motor (for instance, one that drives a drone propeller). The goal is to derive the governing equations from first principles (see figure 3).

Electrical subsystem. Formalise each of the following physical statements as a mathematical equation.

(a) "The voltage across a resistor equals the product of the resistance R and the electrical current I flowing through the circuit."

(b) "The voltage across an inductor equals the product of the inductance L and the rate of change of the electrical current flowing through the circuit."

(c) "The electromotive-force (EMF) voltage generated by the rotating motor equals k times its angular velocity $\omega = \dot{\theta}$."

(d) "The input voltage V is a non-negative constant."

(e) "Kirchhoff's voltage law: the sum of voltages around a closed loop in an electrical circuit equals zero." Apply this to the motor circuit consisting of V, V_R, V_L, and V_M.

Mechanical subsystem. Formalise each of the following statements.

(f) "The resulting inertial torque equals the moment of inertia J times the angular acceleration." *Hint:* think of Newton's second law for rotation, where J plays the role of mass and θ the role of position.

(g) "The damping torque is proportional to the angular velocity, with proportionality constant b."

(h) "The torque produced by the motor is directly proportional to the motor's current via the EMF constant k."

(i) "The net torque acting on the motor shaft equals the motor torque minus the damping torque."

State-space representation.

(j) What is the quantity we would like to control in this system?

(k) Using the equations you derived, write the continuous-time state-space model of the DC motor with input voltage V and states ω (angular velocity) and I (current). A template is shown in Figure 4.

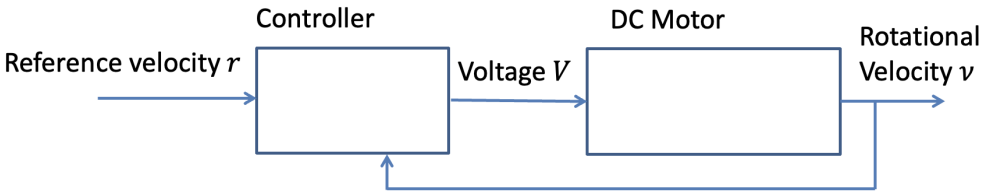

Figure 4: Fill this diagram with the appropriate equations.

Exercise 5: PID Control of the DC Motor

We continue with the DC motor model from Exercise 4. The input voltage V is now determined by a controller, and the set-point is $SP = 1$ (rad/s). Use the default parameters $J = 0.01$, $b = 0.1$, $k = 0.01$, $R = 1$, $L = 0.5$.

Proportional control.

(a) A proportional controller sets $V = K_p \cdot (SP - \omega)$. Based on your experience from Exercises 2 and 3, do you expect the proportional controller to drive ω to the set-point? *Hint:* think about what the controller does once the plant *reaches* the desired speed, and what happens to the plant afterwards.

(b) Compute the equilibria of the closed-loop system. That is, set $\dot{I} = 0$ and $\dot{\omega} = 0$ and solve for I and ω in terms of K_p and the motor parameters.

(c) Evaluate the equilibrium angular velocity ω^* for $R = 1$, $b = 0.1$, $k = 0.01$. Is ω^* equal to SP?

(d) What happens if we increase K_p? Can we eliminate the steady-state error entirely by choosing K_p large enough?

(e) In physical terms, what is the key difference between this DC motor example and the simpler systems of Exercises 2 and 3 that produces this steady-state error?

Adding integral action.

The problem with the proportional controller is that it does not account for the accumulated past error. Below is pseudo-code for a (discretised) proportional controller:

```
LOOP
  PV := sensor_read()
  error := SP − PV
  u := K_p · error
```

(f) What modifications would you make to the pseudo-code above so that the controller keeps track of the *history* of the error? You may *not* use arrays or logging; instead, use a running sum variable. *Hint:* think about adding a term $K_i \cdot$ `error_sum`.

(g) *Using the provided simulation:* does the addition of integral action help the system reach the set-point? Why or why not?

(h) *Using the provided simulation:* what happens if you increase the constant K_i?

Step-response characteristics.

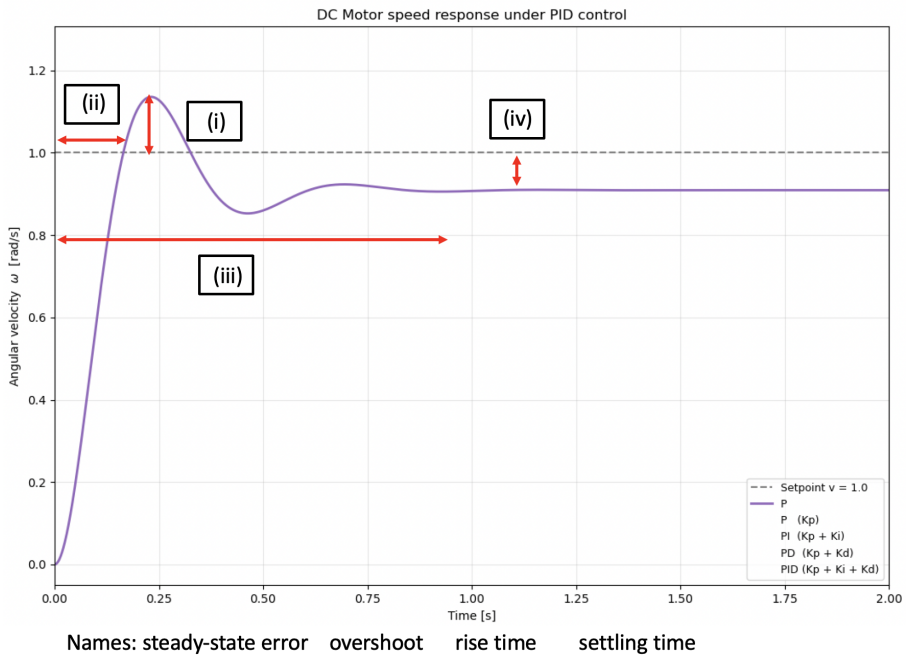

Figure 5: Step response of a controlled system. Match the labels (i)–(iv) to the names.

(i) Given the step-response plot in Figure 5, match the labels (i)–(iv) to the names in the figure.

(j) Match each definition on the left with the correct term on the right.

Definition	Term
(A) Difference between the maximum output value and the reference value	1. Rise time
(B) Time at which the output first crosses the reference value	2. Settling time
(C) Time at which the output reaches and stays within a band around the steady-state value	3. Overshoot
(D) Difference between the steady-state output value and the reference value	4. Steady-state error

(k) *Using the lecturer's projected simulation*, estimate the following for the current PI controller settings:

(i) percentage of overshoot,

(ii) rise time,

(iii) settling time,

(iv) steady-state error percentage.

Adding derivative action.

(l) Suppose the PI controller overshoots excessively. Can we reduce the overshoot by adjusting K_i? If we reduce K_i, what is the consequence for the settling time, and why?

(m) To reduce overshoot while keeping a relatively short settling time, we could look at how fast the error is *changing* over time—in a sense, "look into the future." What modifications would you make to your PI pseudo-code to incorporate this rate of change? (Hint: the continuous version is presented in Figure 6.)

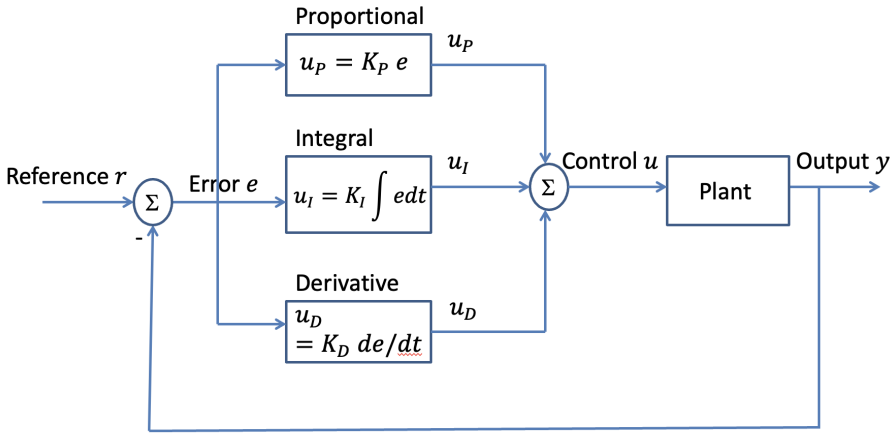

Figure 6: Generic representation of a PID controller.