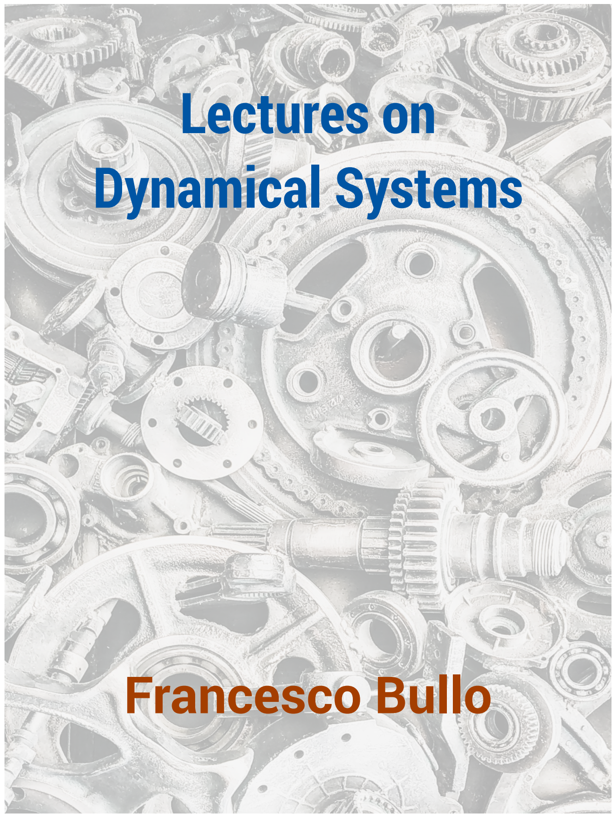
UC Santa Barbara, Department of Mechanical Engineering

ME103 Dynamical Systems. Slides by Chapter.

Francesco Bullo
http://motion.me.ucsb.edu/ME103-Fall2024/syllabus.html



Contents

Chapter 3

Thermal and Fluids Dynamics Systems

3.1 Heat flow

The field of thermodynamics includes the study of dynamical models for heat flow. The three primary mechanisms for heat flow are conduction, convection, and radiation. In these notes we focus on *conduction*, i.e., the transfer of heat energy through a conducting substance, from a region of higher temperature to a region of lower temperature.

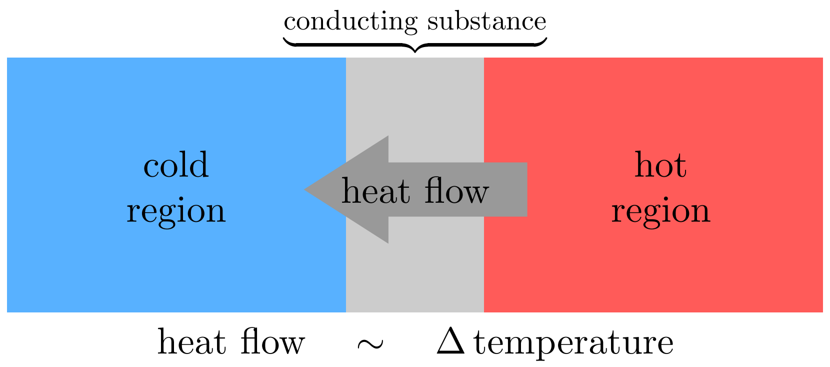


Figure 3.1: In short, Fourier's law states that heat flow is proportional to the temperature differential.

Remark 3.1. *Fourier's law of heat conduction is also known as the resistive heat flow law. This law is analogous to (i) Ohm's law in electrical circuits, where the current is proportional to the voltage difference, and (ii) Fick's first law of diffusion in chemistry, where the diffusion flux is proportional to the concentration gradient.*

Fourier's law of heat conduction from region 1 through a substance to region 2 (e.g., from room 1 through a wall to room 2) states

$$\boxed{q_{1\to 2} = \frac{1}{r}(T_1 - T_2)} \tag{3.1}$$

where

- $q_{1\to 2}$ is the *heat flow rate* from region 1 to region 2, measured in watts W, i.e., joules per second.
- T_i is the *temperature* in region $i = 1, 2$, measured in kelvins (or, Celsius or Fahrenheit),[1]
- r is the *thermal resistance*,[2] measured in kelvin per watt.
- we allow the heat flow $q_{1\to 2}$ to be both positive (when $T_1 > T_2$) or negative (when $T_2 > T_1$). Other references always assume that $T_1 > T_2$ and talk about only positive heat flow.

Equivalently, one could measure the heat flow in the opposite direction, that is, from room 2 to room 1. Clearly, the heat flow in the opposite direction is equal in magnitude and opposite in sign, that is,

$$q_{2\to 1} = -q_{1\to 2} \qquad \text{so that} \qquad q_{2\to 1} = \frac{1}{r}(T_2 - T_1). \tag{3.2}$$

[1]While the Fahrenheit scale is still the most commonly used scale in the United States, the majority of the world uses Celsius. Yet, these notes and scientists prefer the Kelvin, due to its direct link to the absolute zero temperature.

[2]The thermal resistance is typically proportional to a thermal conductivity of the material and to the cross-sectional area of the substance (perpendicular to the heat flow).

Next, from *thermal energy balance*, we know that the heat flow from region 1 into region 2 will change the temperature of both regions according to:

$$c_1\dot{T}_1 = q_{2\to1} \qquad \text{and} \qquad c_2\dot{T}_2 = q_{1\to2} \tag{3.3}$$

where c_i is the *heat capacity* of region i, where $i = 1, 2$, measured in joules per kelvin. In other words, these equations describe the conservation of energy in the context of heat transfer between two regions.

3.1.1 A thermometer

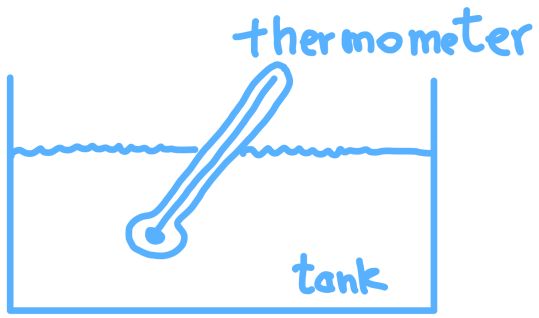


Figure 3.2: We let $T(t)$ denote the thermometer temperature. We let T_{tank} denote the tank temperature; we assume T_{tank} is constant.

Fourier's law for the heat flow gives us

$$q_{\text{tank}\to\text{thermometer}} = \frac{1}{r}\big(T_{\text{tank}} - T(t)\big).$$

Additionally, the temperature of the thermometer is governed by the equation

$$c\,\dot{T}(t) = q_{\text{tank}\to\text{thermometer}}(t).$$

Combining these two equations gives us the *thermometer dynamics*:

$$\dot{T}(t) \;=\; \frac{1}{cr}\big(T_{\text{tank}} - T(t)\big). \tag{3.4}$$

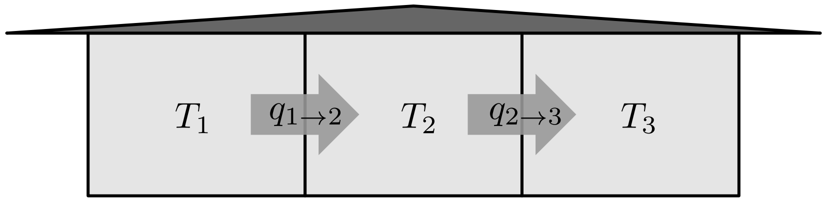
Here $T(t)$ is variable, c and r are two parameters.

Note: The thermometer dynamics (3.4) is a first order system, similar to the linear growth/decay model (1.1) in Chapter 1 and the car velocity system (2.4) in Chapter 2. The thermometer time constant is $\tau = cr$. Also, these dynamics (3.4) are sometimes referred to as *Newton's law of cooling or heating*.

In class assignment

To measure the tank's temperature we immerse in it a thermometer.
Are we measuring the correct temperature?

3.1.2 A building thermal system



Consider a building with three adjacent rooms, numbered $1, 2, 3$ (and assume that the air in each room has uniform temperature at all times). Heat can flow between rooms 1 and 2, and between rooms 2 and 3. Considering room 2, we note the additive effect:

$$c_2\dot{T}_2 = q_{1\to 2} + q_{3\to 2} \qquad \text{where, for example,} \quad q_{1\to 2} = \frac{1}{r_{12}}(T_1 - T_2).$$

Similar equations hold for the other rooms and the other heat flow rates.

Overall we model the heat flow in a building via the Fourier law (3.1); the *building thermal dynamics* are:

$$\begin{aligned} c_1\dot{T}_1 &= \frac{1}{r_{12}}(T_2 - T_1) \\ c_2\dot{T}_2 &= \frac{1}{r_{12}}(T_1 - T_2) + \frac{1}{r_{23}}(T_3 - T_2) \\ c_3\dot{T}_3 &= \frac{1}{r_{23}}(T_2 - T_3) \end{aligned} \tag{3.5}$$

where

- T_i is the temperature in room i, for $i = 1, 2, 3$,
- c_i is the thermal capacity of room i, for $i = 1, 2, 3$, and
- $r_{ij} = r_{ji}$ is the thermal resistance between room i and j, for $i, j = 1, 2, 3$ and $i \neq j$.

Numerical simulation of building thermal system: convergence to uniform temperature

```python
import numpy as np; from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

def heat_flow_dynamics(t, state, c1, c2, c3, r12, r23):
    T1, T2, T3 = state
    T1_dot = (T2 - T1) / (r12 * c1)
    T2_dot = (T1 - T2) / (r12 * c2) + (T3 - T2) / (r23 * c2)
    T3_dot = (T2 - T3) / (r23 * c3)
    return [T1_dot, T2_dot, T3_dot]

# Parameters for the heat flow system
c1 = 1000  # Heat capacity of room 1 (J/K)
c2 = 1000  # Heat capacity of room 2 (J/K)
c3 = 1000  # Heat capacity of room 3 (J/K)
r12 = 0.2  # Thermal resistance between rooms 1 and 2 (K/W)
r23 = 0.9  # Thermal resistance between rooms 2 and 3 (K/W)

# Time array and initial conditions: [T1, T2, T3]
t = np.linspace(0, 1800, 12000)
initial_conditions = [20.0, 28.0, 35.0]
sol = solve_ivp(heat_flow_dynamics, [t[0], t[-1]], initial_conditions, ...
    t_eval=t, args=(c1, c2, c3, r12, r23), method='LSODA')

# Plotting
plt.figure(figsize=(12, 7))
blues = ['#002447', '#003c76', '#0055A4', '#006CD4', '#0085ff', ...
    '#239cff', '#58b1ff']; oranges = ['#471b00', '#752d00', '#a43e00', ...
    '#d35000', '#ff6100', '#ff7f1a', 'ff9b56']
plt.plot(sol.t, sol.y[0], label='Temperature of Room 1 (Celsius)', ...
    color=blues[1])
plt.plot(sol.t, sol.y[1], label='Temperature of Room 2 (Celsius)', ...
    color=blues[3])
plt.plot(sol.t, sol.y[2], label='Temperature of Room 3 (Celsius)', ...
    color=blues[5])

# Add LaTeX labels to the plot
plt.text(300, 24.5, r'$T_1(t)$', fontsize=14, ha='center', va='center', ...
    bbox=dict(facecolor='white', edgecolor='none', pad=5.0))
plt.text(600, 26.6, r'$T_2(t)$', fontsize=14, ha='center', va='center', ...
    bbox=dict(facecolor='white', edgecolor='none', pad=5.0))
plt.text(900, 29.5, r'$T_3(t)$', fontsize=14, ha='center', va='center', ...
    bbox=dict(facecolor='white', edgecolor='none', pad=5.0))

plt.grid(True)
plt.xlabel('Time (s)')
plt.ylabel('Temperature (Celsius)')
plt.xticks(np.arange(0, 1801, 300), ['0', '5 min', '10 min', '15 min', ...
    '20 min', '25 min', '30 min'])
plt.legend()
plt.xlim(0, 1800)
plt.title('Temperatures of the rooms')
plt.tight_layout()
plt.savefig("building.pdf", bbox_inches='tight')
```

Listing 3.1: Python script generating Figure 3.3. Available at building.py

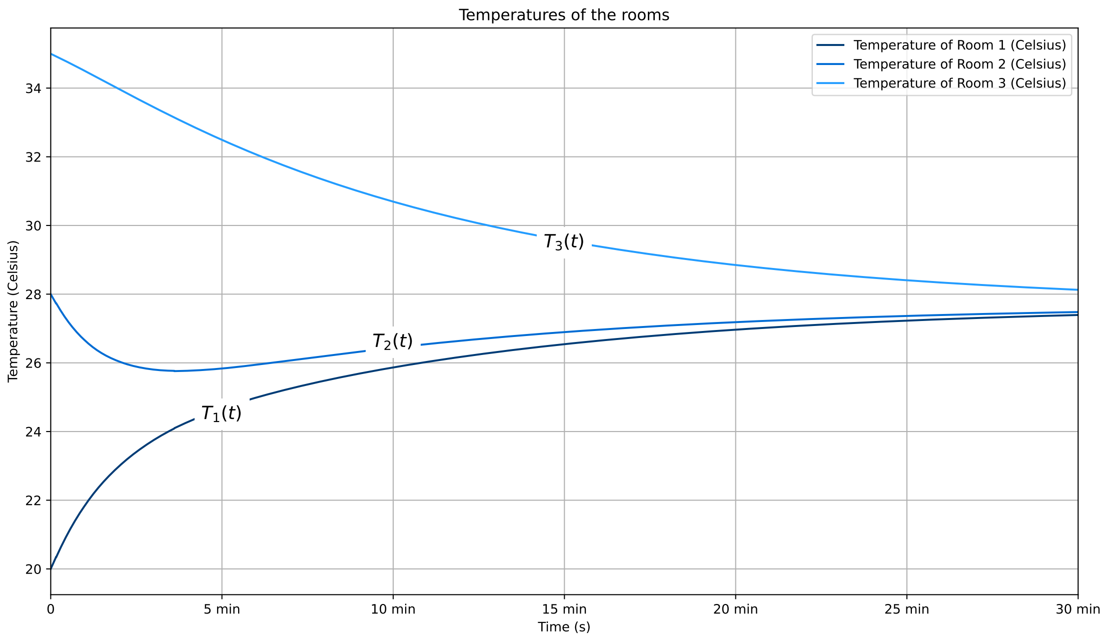
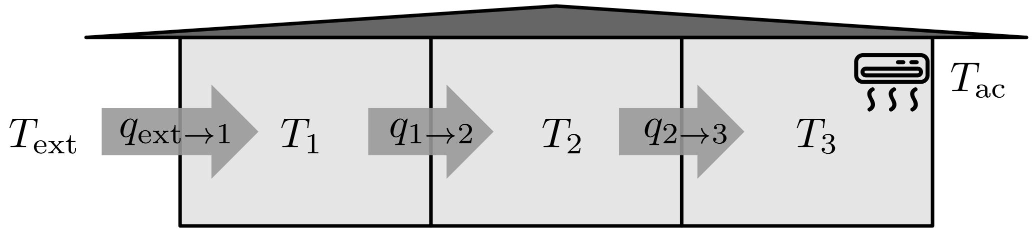


Figure 3.3: Solutions of the building system (3.5).
Since the thermal resistance between rooms 2 and 3 is selected much larger than that between rooms 1 and 2, rooms 1 and 2 quickly reach a temperature similar to each other. Ultimately, however, all temperatures will become equal.

3.1.3 A controlled building thermal system



Next, we include two additional effects and design a *temperature controller*. We assume that:

(i) room 1 is in direct contact with an *external environment* with a constant temperature T_{ext} through a thermal resistance $r_{1\text{ext}}$. We assume T_{ext} is high and the external environment is heating up the building;

(ii) room 3 contains an air conditioning system that supplies air at a constant temperature T_{ac} where $T_{\text{ac}} < T_{\text{ext}}$. The quantity of air and, therefore, the effect of the air conditioner is modulated by a *control signal* $u(t)$, which is *binary*:

$$u(t) = \begin{cases} 1 & \text{sets the air conditioner ON, or} \\ 0 & \text{sets the air conditioner OFF,} \end{cases} \tag{3.6}$$

(iii) as first *control design*, we choose a desirable temperature, cooler than the warm environment T_{ext} and yet warmer than the cold air of the air-conditioner T_{ac}. For example, with $T_{\text{ext}} = 30°\text{C}$ and $T_{\text{ac}} = 20°\text{C}$, we turn ON the air conditioner every time $T_3 > 23°\text{C}$, that is,

$$u(T_3) = \begin{cases} 1 & \text{if } T_3 > 23°\text{C}, \\ 0 & \text{if } T_3 \leq 23°\text{C}. \end{cases} \tag{3.7}$$

This control law is called *on-off control*.

In summary, the governing differential equations for this system are:

$$\begin{aligned} c_1\dot{T}_1 &= \frac{1}{r_{12}}(T_2 - T_1) + \frac{1}{r_{1\text{ext}}}(T_{\text{ext}} - T_1), \\ c_2\dot{T}_2 &= \frac{1}{r_{12}}(T_1 - T_2) + \frac{1}{r_{23}}(T_3 - T_2), \\ c_3\dot{T}_3 &= \frac{1}{r_{23}}(T_2 - T_3) \ + \ ku(T_3)(T_{\text{ac}} - T_3). \end{aligned} \tag{3.8}$$

where $k > 0$ is a proportionality constant determining the effectiveness of the air conditioner and related to the amount of air flowing into room 3.

Numerical simulation: no control, only the external environment

```python
import numpy as np; import matplotlib.pyplot as plt; import matplotlib

def heat_flow_dynamics(T1, T2, T3, c1, c2, c3, r12, r23, k, Tac, Text):
    u = 0 # no control = no air conditioning
    T1_dot = (T2 - T1) / (r12 * c1) + (Text - T1) / (r1ext * c1)
    T2_dot = (T1 - T2) / (r12 * c2) + (T3 - T2) / (r23 * c2)
    T3_dot = (T2 - T3) / (r23 * c3) + k * u * (Tac - T3) / c3
    return T1_dot, T2_dot, T3_dot, u

# Parameters
c1, c2, c3, r12, r23, r1ext, Text, Tac, k = 1000, 1000, 1000, 0.2, 0.9, .5, 30, ...
    20, 3.0
u_prev = [0]; t_end = 5400; dt = t_end / 24000; times = np.arange(0, t_end + dt, dt)

# Initialize values
T1_vals, T2_vals, T3_vals, u_vals = [18.0], [18.0], [27.0], [0]
T1, T2, T3 = 15.0, 18.0, 27.0
for t in times[1:]:
    T1_dot, T2_dot, T3_dot, u = heat_flow_dynamics(T1, T2, T3, c1, c2, c3, r12, ...
        r23, k, Tac, Text)
    T1 += dt * T1_dot; T2 += dt * T2_dot; T3 += dt * T3_dot
    T1_vals.append(T1); T2_vals.append(T2); T3_vals.append(T3); u_vals.append(u)

# Create the figure and the gridspec
fig = plt.figure(figsize=(12, 8)); gs = matplotlib.gridspec.GridSpec(2, 1, ...
    height_ratios=[3, 1])
blues = ['#002447', '#003c76', '#0055A4', '#006CD4', '#0085ff', '#239cff', '#58b1ff']
orngs = ['#471b00', '#752d00', '#a43e00', '#d35000', '#ff6100', '#ff7f1a', '#ff9b56']

# First subplot (Temperatures)
ax1 = plt.subplot(gs[0]); ax1.set_ylim(15, 28)
ax1.plot(times, T1_vals, label='Temperature of Room 1 (C)', color=blues[1])
ax1.plot(times, T2_vals, label='Temperature of Room 2 (C)', color=blues[3])
ax1.plot(times, T3_vals, label='Temperature of Room 3 (C)', color=blues[5])
ax1.grid(True); ax1.legend(); ax1.set_ylabel('Temperature (C)')
ax1.set_title('Temperatures of the rooms and AC control signal over 1.5 hours')
ax1.set_yticks(np.arange(15, max(max(T1_vals), max(T2_vals), max(T3_vals)) + 1, 1))
ax1.set_xlim(0, t_end); ax1.set_xticks([0, 900, 1800, 2700, 3600, 4500, 5400])
ax1.set_xticklabels(['0', '15 min', '30 min', '45 min', '60 min', '75 min', '90 min'])
ax1.text(900, 25.5, r'$T_1(t)$', fontsize=14, ha='center', va='center', ...
    bbox=dict(facecolor='white', edgecolor='none', pad=5.0))
ax1.text(1800, 27, r'$T_2(t)$', fontsize=14, ha='center', va='center', ...
    bbox=dict(facecolor='white', edgecolor='none', pad=5.0))
ax1.text(2700, 26.75, r'$T_3(t)$', fontsize=14, ha='center', va='center', ...
    bbox=dict(facecolor='white', edgecolor='none', pad=5.0))

# Second subplot (Control signal)
ax2 = plt.subplot(gs[1]); ax2.set_xlim(0, t_end); ax2.grid(True)
ax2.plot(times, u_vals, label='Control signal (u)', color='black')
ax2.set_xlabel('Time (s)'); ax2.set_ylabel('Control Signal')
ax2.set_xticks([0, 900, 1800, 2700, 3600, 4500, 5400])
ax2.set_xticklabels(['0', '15 min', '30 min', '45 min', '60 min', '75 min', '90 min'])
ax2.set_yticks([0, 1]); ax2.set_yticklabels(['OFF', 'ON']); ax2.set_ylim(-0.1, 1.1)

# Save the figure
plt.tight_layout(); plt.savefig("building-Text.pdf", bbox_inches='tight')
```

Listing 3.2: Python script generating Figure 3.4. Available at building-Text.py

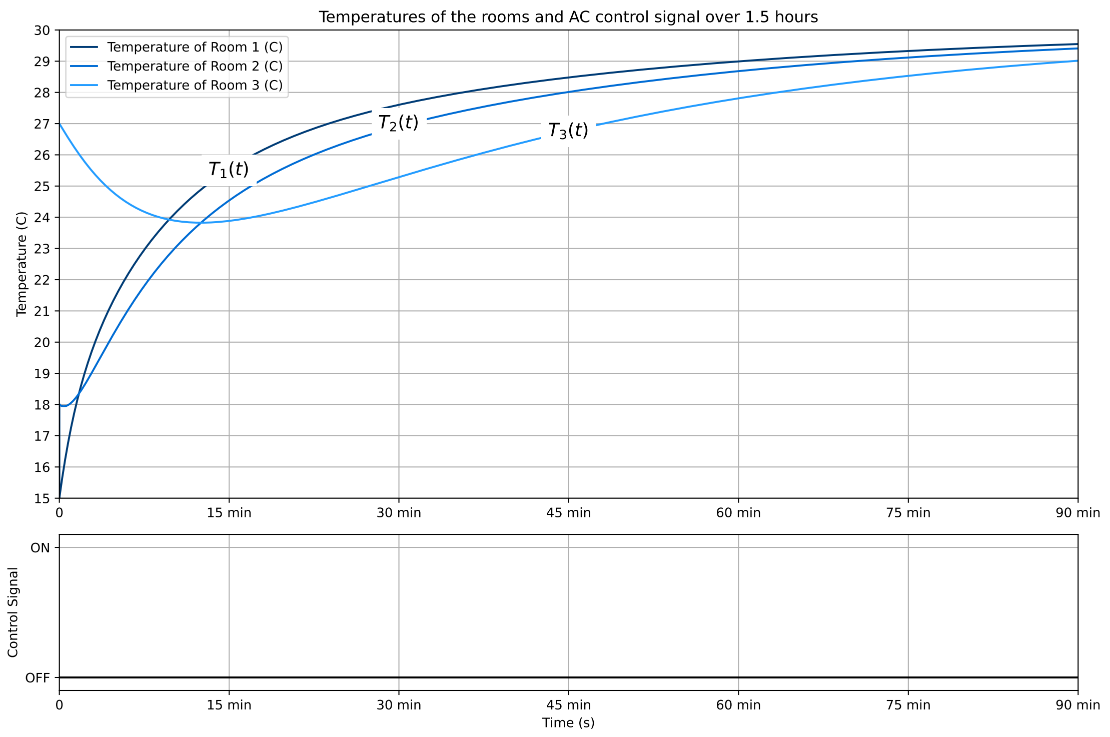


Figure 3.4: Solutions of the building system with heat exchange with the external environment (3.8) and zero control (the air conditioning is off).
The external environment is at $T_{\text{ext}} = 30°\text{C}$ and so the temperature in each room, slowly but surely, converges to $30°\text{C}$.

Numerical simulation: chattering in on-off control

```python
import numpy as np; from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt; import matplotlib.gridspec as gridspec

def heat_flow_dynamics(t, state, c1, c2, c3, r12, r23, k, Tac, Text):
    T1, T2, T3 = state
    u = 1.0 if T3 > 23.0 else 0.0     ## on off control
    air_conditioning = k * u * (Tac - T3)
    T1_dot = (T2 - T1) / (r12 * c1) + (Text - T1) / (r1ext * c1)
    T2_dot = (T1 - T2) / (r12 * c2) + (T3 - T2) / (r23 * c2)
    T3_dot = (T2 - T3) / (r23 * c3) + air_conditioning / c3
    return [T1_dot, T2_dot, T3_dot, u]

# Parameters for the heat flow system
c1, c2, c3, r12, r23, r1ext, k = 1000, 1000, 1000, 0.2, 0.9, .5, 3.0
Text, Tac = 30, 20; initial_conditions = [17.0, 18.0, 27.0]
t = np.linspace(0, 1800, 18000)

def wrap_dynamics(t, y):
    return heat_flow_dynamics(t, y, c1, c2, c3, r12, r23, k, Tac, Text)[:3]

sol = solve_ivp(wrap_dynamics, [t[0], t[-1]], initial_conditions, t_eval=t, ...
     method='LSODA')
u_values = [heat_flow_dynamics(ti, y, c1, c2, c3, r12, r23, k, Tac, Text)[3] for ...
     ti, y in zip(sol.t, sol.y.T)]

# Create the figure and the gridspec
fig = plt.figure(figsize=(12, 8)); gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1])
blues = ['#002447', '#003c76', '#0055A4', '#006CD4', '#0085ff', '#239cff', '#58b1ff']
oranges = ['#471b00', '#752d00', '#a43e00', '#d35000', '#ff6100', '#ff7f1a', 'ff9b56']

# First subplot (Temperatures)
ax1 = plt.subplot(gs[0])
ax1.plot(sol.t, sol.y[0], label='Temperature of Room 1 (C)', color=blues[1])
ax1.plot(sol.t, sol.y[1], label='Temperature of Room 2 (C)', color=blues[3])
ax1.plot(sol.t, sol.y[2], label='Temperature of Room 3 (C)', color=blues[5])
ax1.axhline(y=23, color=oranges[4], linestyle='--', linewidth=1.5,
            label='Target Temperature (23C)')
ax1.grid(True); ax1.set_ylabel('Temperature (C)'); ax1.legend()
ax1.set_title('Temperatures of the rooms and AC control signal over 30 minutes')

# Add LaTeX labels to the plot
ax1.text(150, 20.5, r'$T_1(t)$', fontsize=14, ha='center', va='center', ...
     bbox=dict(facecolor='white', edgecolor='none', pad=5.0))
ax1.text(300, 20.5, r'$T_2(t)$', fontsize=14, ha='center', va='center', ...
     bbox=dict(facecolor='white', edgecolor='none', pad=5.0))
ax1.text(800, 22.5, r'$T_3(t)$', fontsize=14, ha='center', va='center', ...
     bbox=dict(facecolor='white', edgecolor='none', pad=5.0))

ax1.set_yticks(np.arange(17, max(max(sol.y[0]), max(sol.y[1]), max(sol.y[2])) + ...
     1, 1))
ax1.set_xlim(0, 1800); ax1.set_xticks([0, 600, 1200, 1800])
ax1.set_xticklabels(['0', '10 min', '20 min', '30 min'])

# Second subplot (Control signal)
ax2 = plt.subplot(gs[1])
ax2.plot(sol.t, u_values, label='Control signal (u)', color='black')
ax2.grid(True); ax2.set_xlabel('Time (s)'); ax2.set_ylabel('Control Signal')
ax2.set_xticks([0, 600, 1200, 1800]); ax2.set_xlim(0, 1800)
ax2.set_xticklabels(['0', '10 min', '20 min', '30 min'])
ax2.set_yticks([0, 1]); ax2.set_yticklabels(['OFF', 'ON']); ax2.set_ylim(-0.1, 1.1)
plt.tight_layout(); plt.savefig("building-onoff.pdf", bbox_inches='tight')
```

Listing 3.3: Python script generating Figure 3.5. Available at building-onoff.py

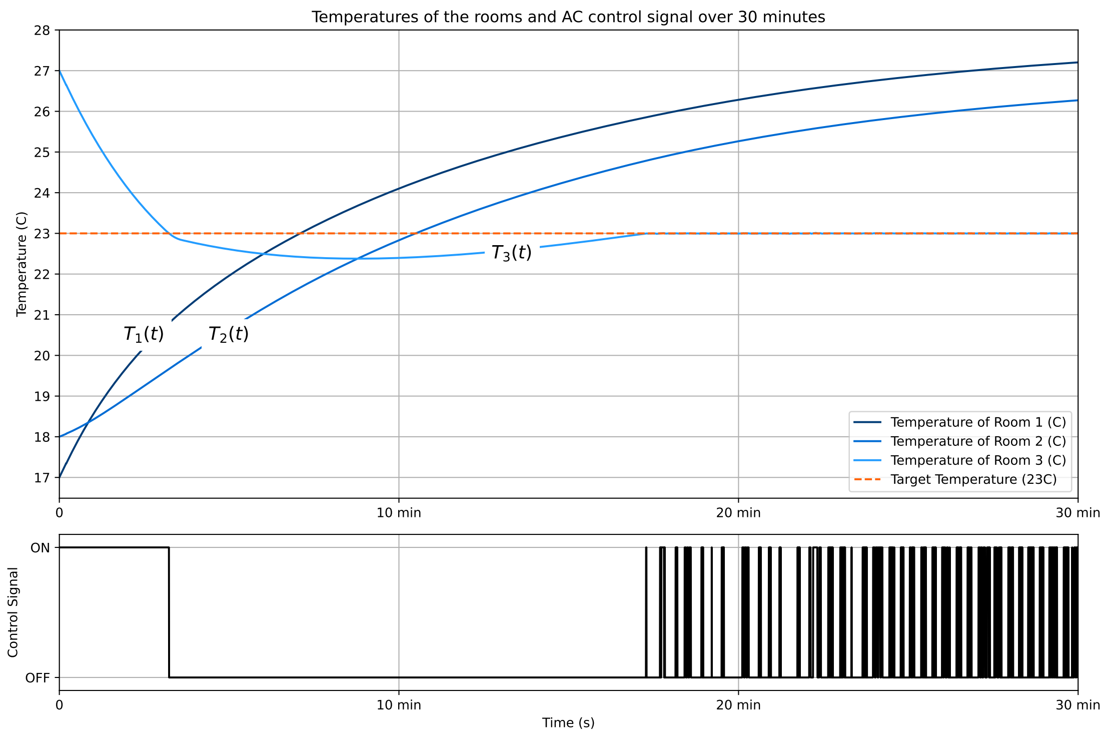
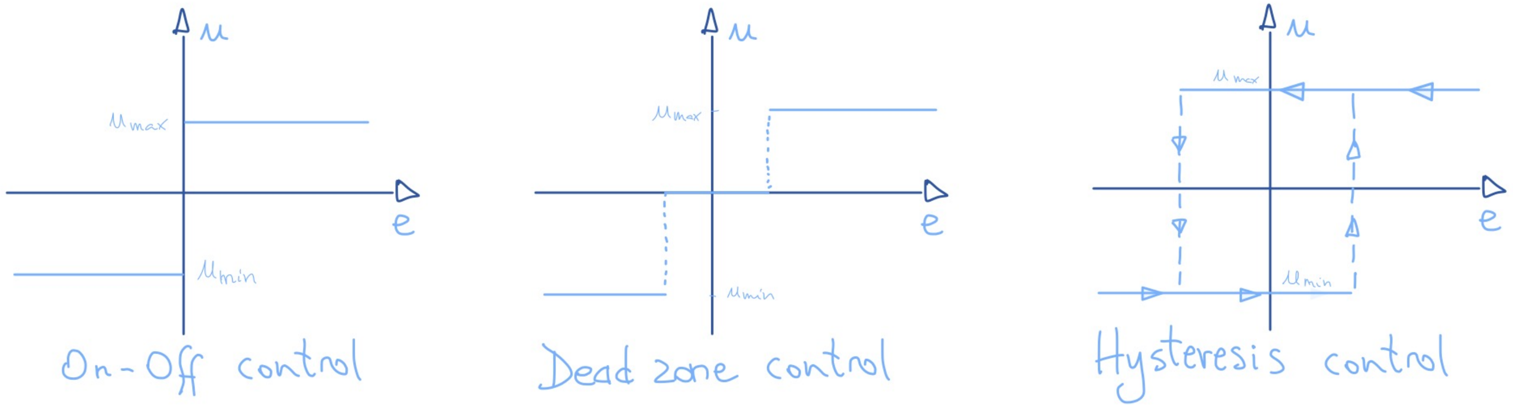


Figure 3.5: Solutions of the building system with air conditioning and heat exchange with the external environment (3.8).
Room 1 starts at a cold $10°$C, but it is directly exposed to a warm external environment at $30°$C, it ends up as the warmest room.
Room 3 starts at $27°$C and therefore the air conditioning is ON for small times.
So long as $T_2 < T_3$, room 3 is cooled by room 2. Once room 2 becomes hot, room 1 starts to warm up and then the air conditioning exhibit a *chattering behavior* about the desired temperature of $23°$C. Fast switching happens too frequently, can damage the device, and is undesirable.

The *on-off control* is possibly the simplest control strategy one could design: when the error is positive, the control is on, when the error is negative, the control of off.

In the following figure, we define two variations: *dead-zone control* and *hysteresis control*.



Numerical simulation: hysteresis control

```python
import numpy as np; import matplotlib.pyplot as plt; import matplotlib

def heat_flow_dynamics(T1, T2, T3, c1, c2, c3, r12, r23, k, Tac, Text):
    u = 1.0 if T3 > 24.0 else (0.0 if T3 < 22.0 else u_prev[0])
    u_prev[0] = u           ## hysteresis control
    T1_dot = (T2 - T1) / (r12 * c1) + (Text - T1) / (r1ext * c1)
    T2_dot = (T1 - T2) / (r12 * c2) + (T3 - T2) / (r23 * c2)
    T3_dot = (T2 - T3) / (r23 * c3) + k * u * (Tac - T3) / c3
    return T1_dot, T2_dot, T3_dot, u

# Parameters
c1, c2, c3, r12, r23, r1ext, Text, Tac, k = 1000, 1000, 1000, 0.2, 0.9, .5, 30, ...
    20, 3.0
u_prev = [0]; t_end = 5400; dt = t_end / 24000; times = np.arange(0, t_end + dt, dt)

# Initialize values
T1_vals, T2_vals, T3_vals, u_vals = [18.0], [18.0], [27.0], [0]
T1, T2, T3 = 15.0, 18.0, 27.0
for t in times[1:]:
    T1_dot, T2_dot, T3_dot, u = heat_flow_dynamics(T1, T2, T3, c1, c2, c3, r12, ...
        r23, k, Tac, Text)
    T1 += dt * T1_dot; T2 += dt * T2_dot; T3 += dt * T3_dot
    T1_vals.append(T1); T2_vals.append(T2); T3_vals.append(T3); u_vals.append(u)

# Create the figure and the gridspec
fig = plt.figure(figsize=(12, 8)); gs = matplotlib.gridspec.GridSpec(2, 1, ...
    height_ratios=[3, 1])
blues = ['#002447', '#003c76', '#0055A4', '#006CD4', '#0085ff', '#239cff', '#58b1ff']
orngs = ['#471b00', '#752d00', '#a43e00', '#d35000', '#ff6100', '#ff7f1a', '#ff9b56']

# First subplot (Temperatures)
ax1 = plt.subplot(gs[0]); ax1.set_ylim(15, 28)
ax1.plot(times, T1_vals, label='Temperature of Room 1 (C)', color=blues[1])
ax1.plot(times, T2_vals, label='Temperature of Room 2 (C)', color=blues[3])
ax1.plot(times, T3_vals, label='Temperature of Room 3 (C)', color=blues[5])
ax1.axhline(y=22, color=orngs[4], linestyle='--', linewidth=1.5, label='Lower ...
    Bound (22C)')
ax1.axhline(y=24, color=orngs[4], linestyle='--', linewidth=1.5, label='Upper ...
    Bound (24C)')
ax1.grid(True); ax1.legend(); ax1.set_ylabel('Temperature (C)')
ax1.set_title('Temperatures of the rooms and AC control signal over 1.5 hours')
ax1.set_yticks(np.arange(15, max(max(T1_vals), max(T2_vals), max(T3_vals)) + 1, 1))
ax1.set_xlim(0, t_end); ax1.set_xticks([0, 900, 1800, 2700, 3600, 4500, 5400])
ax1.set_xticklabels(['0', '15 min', '30 min', '45 min', '60 min', '75 min', '90 min'])
ax1.text(900, 25, r'$T_1(t)$', fontsize=14, ha='center', va='center', ...
    bbox=dict(facecolor='white', edgecolor='none', pad=5.0))
ax1.text(1800, 26.25, r'$T_2(t)$', fontsize=14, ha='center', va='center', ...
    bbox=dict(facecolor='white', edgecolor='none', pad=5.0))
ax1.text(2700, 23, r'$T_3(t)$', fontsize=14, ha='center', va='center', ...
    bbox=dict(facecolor='white', edgecolor='none', pad=5.0))

# Second subplot (Control signal)
ax2 = plt.subplot(gs[1]); ax2.set_xlim(0, t_end); ax2.grid(True)
ax2.plot(times, u_vals, label='Control signal (u)', color='black')
ax2.set_xlabel('Time (s)'); ax2.set_ylabel('Control Signal')
ax2.set_xticks([0, 900, 1800, 2700, 3600, 4500, 5400])
ax2.set_xticklabels(['0', '15 min', '30 min', '45 min', '60 min', '75 min', '90 min'])
ax2.set_yticks([0, 1]); ax2.set_yticklabels(['OFF', 'ON']); ax2.set_ylim(-0.1, 1.1)

# Save the figure
plt.tight_layout(); plt.savefig("building-hysteresis.pdf", bbox_inches='tight')
```

Listing 3.4: Python script generating Figure 3.6. Available at building-hysteresis.py

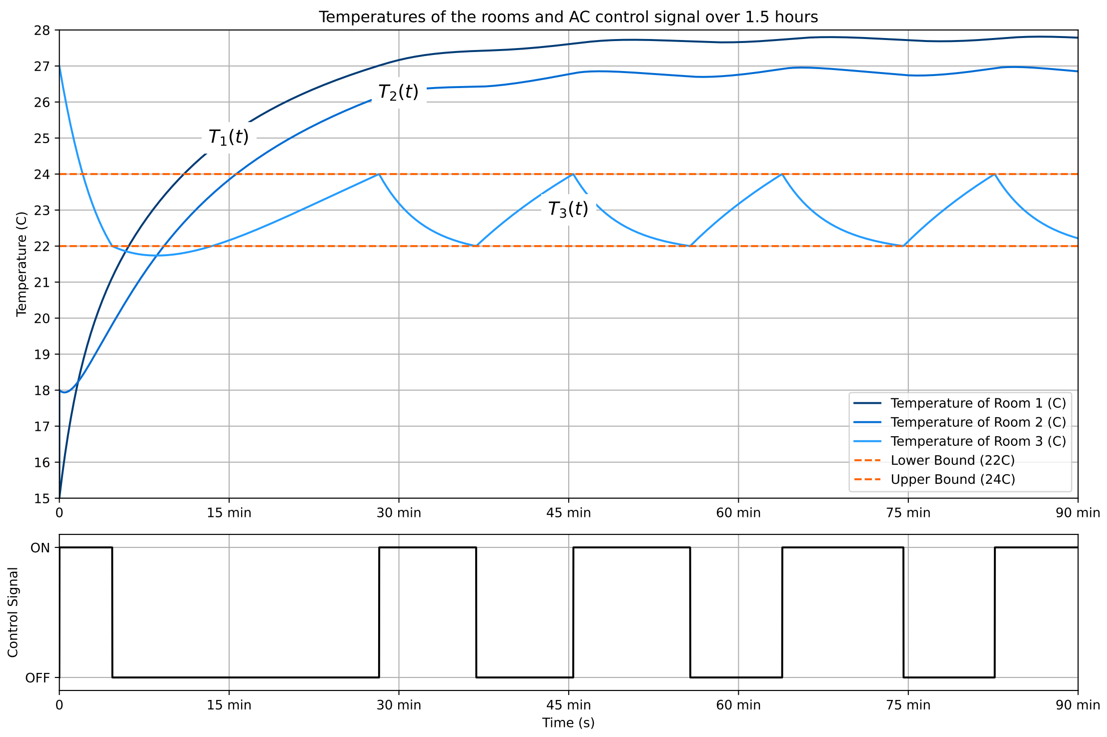


Figure 3.6: Solutions of the building system with air conditioning and heat exchange with the external environment (3.8).

We now design a controller with a *hysteresis*, that is, the air conditioner will:
turn ON when the temperature exceeds 24°C.
turn OFF when the temperature drops below 22°C.
Do nothing if the temperature is between 22°Cand 24°C.

3.2 Incompressible fluid flow

Fluid flows are widely studied and of major importance in mechanical engineering systems. The simplest physical law governing fluid flow is *continuity* or *conservation of mass*.

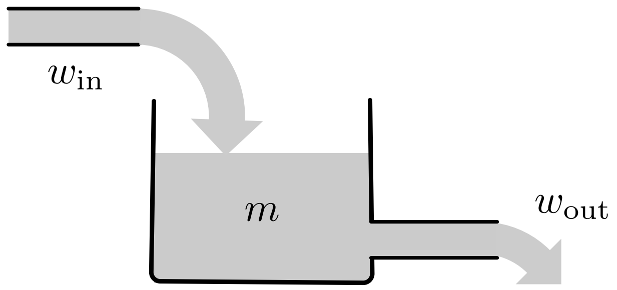


Figure 3.7: Water tank example

For the water tank example in Figure 3.7, conservation of mass implies

$$\dot{m} = w_{\text{in}} - w_{\text{out}} \tag{3.9}$$

where

- m is the *fluid mass* inside the tank, measured in kg,
- w_{in} is the *incoming mass flow rate* into the tank, measured in kg/sec, and
- w_{out} is the *outgoing mass flow rate* out of the tank.

A second physical law regulating incompressible fluid flow is *force equilibrium*.

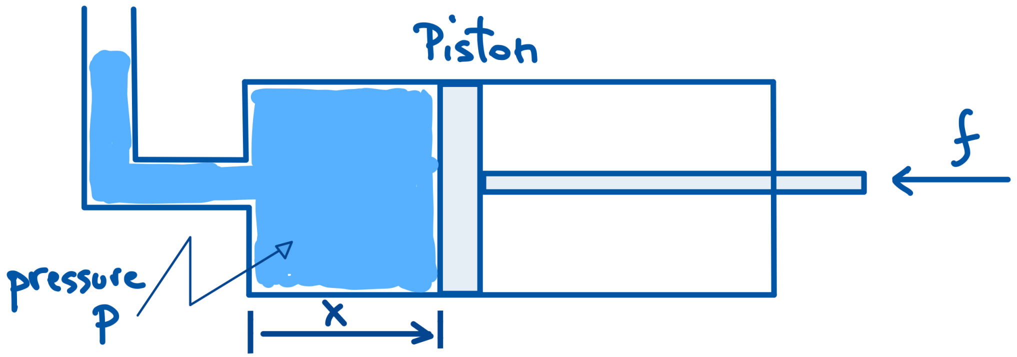


Figure 3.8: Piston example, with incompressible fluid in chamber

For the piston example in Figure 3.8, the fluid flow system is counteracting a force acting on the piston. Newton's law (force equal mass times acceleration) implies

$$M\ddot{x} = Ap - f \tag{3.10}$$

where

- M is the mass of the piston,
- x is the position of the piston,
- A is the cross-sectional area of the piston,
- p is the *pressure* in the piston chamber, and
- f is the force applied to the piston.

A third and final physical law governing fluid flow is the law of *nonlinear resistance*.

(a) Laminar flow (b) Turbulent flow

Figure 3.9: Fluid flow in a pipe: laminar versus turbulent.

Consider a fluid flow through a pipe from a location with high pressure p_1 to a location with low pressure p_2, that is, $p_1 > p_2$. The effect of resistance is approximately modeled by

$$w = \frac{1}{r}(p_1 - p_2)^{1/\alpha} \tag{3.11}$$

where

- w is the *mass flow rate*,
- $p_1 - p_2 > 0$ is the *pressure difference*,
- r is a *flow resistance*, determined by the pipe's roughness, diameter, length, and the fluid's viscosity, and
- $1 < \alpha < 2$ is a *flow behavior parameter*, which (unfortunately) does not directly correlate to laminar or turbulent flow as in Figure 3.9. Roughly speaking,

$\alpha \approx 2$	is an appropriate value for high flow rates through pipes or nozzles,
$\alpha \approx 1$	is an appropriate value for very slow flows through long pipes,
$1 < \alpha < 2$	corresponds to intermediate regimes.

(A complete analysis of flow behavior is outside the scope of these notes.)

An example water tank: height dynamics with pressure-dependent outflow

Starting from the water tank model $\dot{m} = w_{\text{in}} - w_{\text{out}}$, we recall $m(t) = A\rho h(t)$, where $h(t) \geq 0$ is the *water height*, A is the cross-sectional area[3] of the tank, and ρ is the density of water. We write the dynamics of the water height as

$$\dot{h} = \frac{1}{A\rho}\big(w_{\text{in}} - w_{\text{out}}\big). \tag{3.12}$$

We consider a constant incoming mass flow rate w_{in}. For the outgoing mass flow rate, using the flow resistance model (3.11) with $\alpha = 1/2$, we assume instead that

$$w_{\text{out}}(h) = \frac{1}{r}\big(p_{\text{hydrostatic}}(h) - p_{\text{ambient}}\big)^{1/2} \tag{3.13}$$

where p_{ambient} is the ambient pressure and $p_{\text{hydrostatic}}(h)$ is the *hydrostatic pressure*, that is, the pressure exerted by a fluid at equilibrium due to the force of gravity.

Now, (from Pascal's principle) recall that the hydrostatic pressure at the bottom of the tank is the sum of the ambient pressure and the pressure due to the "water column," that is

$$p_{\text{hydrostatic}}(h) = p_{\text{ambient}} + \rho g h. \tag{3.14}$$

In summary, the tank height dynamics are

$$\dot{h} = \frac{1}{A\rho}\Big(w_{\text{in}} - \frac{1}{r}\big(\rho g h\big)^{1/2}\Big) = -\frac{\sqrt{\rho g}}{rA\rho}\sqrt{h} + \frac{1}{A\rho}w_{\text{in}}. \tag{3.15}$$

[3]The cross-sectional area of a tank is the area of the shape formed by intersecting the tank with a plane perpendicular to its axis at a given level.

For simplicity, we introduce the parameter $a = \frac{\sqrt{\rho g}}{rA\rho} > 0$ and $b = \frac{1}{A\rho} > 0$ so that we can rewrite the water tank equations as

$$\dot{h} = -a\sqrt{h} + bw_{\text{in}} \tag{3.16}$$

Next, we compute the equilibria of the water tank system by setting $\dot{h} = 0$. Note that there is only one equilibrium:

$$a\sqrt{h^*} = bw_{\text{in}} \qquad \Longleftrightarrow \qquad h^* = (b/a)^2 w_{\text{in}}^2. \tag{3.17}$$

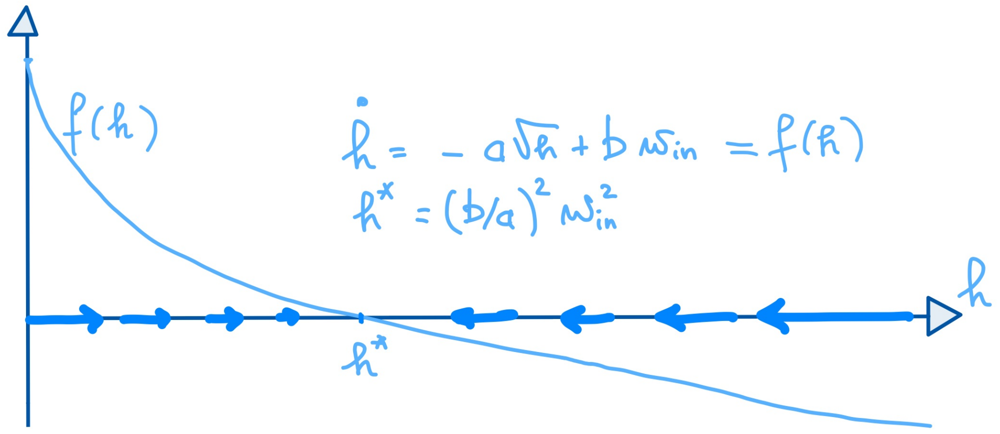


Figure 3.10: Phase portrait on the line for the water tank system with input mass flow rate w_{in}.
We observe that there exists only 1 equilibrium point and it is stable.

3.3 Linearization of nonlinear systems for small signals

A *dynamical system with a control input* or a *control system* with n state variables x and m state variables u is a system of the form

$$\dot{x} = f(x, u) \tag{3.18}$$

For example, first order systems include:

- the car velocity system (2.4) $\dot{v} = -(b/m)v + f/m$ has $n = 1$ state v and $m = 1$ input f;
- the water tank system (3.16) $\dot{h} = -a\sqrt{h} + bw_{\text{in}}$ has 1 state h and 1 input w_{in};

and second order systems include:

- the forced mass-spring-damper system (2.12) $m\ddot{x} + b\dot{x} + kx = f$ has 2 states $(x, \dot{x})$ and 1 input f;
- the air-conditioned building system (3.8) has three states (T_1, T_2, T_3) and 2 inputs: the air conditioning control u and the external temperature T_{ext}.

3.3.1 Mathematical analysis: Equilibrium pair for a control system

For a control system (3.18), an *equilibrium* is a pair (x^*, u^*) such that

$$f(x^*, u^*) = 0 \tag{3.19}$$

so that the constant trajectory $\big(x(t), u(t)\big) = \big(x^*, u^*\big)$ is a solution for all time. For example

- for the water tank system $\big(h^*, w_{\text{in}}^*\big) = \big((b/a)^2(w^*)^2, w^*\big)$ is an equilibrium,
- for the forced mass-spring-damper system $(x, \dot{x}, f) = (f_0/k, 0, f_0)$ is an equilibrium.

3.3.2 Mathematical analysis: Linearization of nonlinear systems near an equilibrium

Next, we discuss the *linearization process* that is very useful to analyze nonlinear dynamical systems. Specifically, to understand the properties of nonlinear systems (e.g., the stability of an equilibrium point), we will proceed in two steps: (1) via *small signal linearization*, we understand how nonlinear systems behave linearly locally near the equilibrium, and (2) we will then study the stability problem for general linear systems.

In what follows, we will use Taylor expansions, see the review in Appendix 3.5.

Loosely speaking, as we did for the car velocity system, we now consider a *change of coordinate*. Near the equilibrium pair (x^*, u^*), we consider small variations and write

$$\begin{aligned} x(t) &= x^* + \delta x(t), \\ u(t) &= u^* + \delta u(t), \end{aligned}$$

and we plug into both sides of $\dot{x} = f(x, u)$ to obtain

$$\begin{aligned} \dot{x}(t) = \frac{d}{dt}\big(x^* + \delta x(t)\big) = \frac{d}{dt}\delta x(t) &= f\big(x^* + \delta x(t), u^* + \delta u(t)\big) \\ &\approx f(x^*, u^*) \; + \; \underbrace{\frac{\partial f}{\partial x}(x^*, u^*)}_{=F}\,\delta x(t) \; + \; \underbrace{\frac{\partial f}{\partial u}(x^*, u^*)}_{=G}\,\delta u(t), \end{aligned}$$

where we have used the *Taylor expansion about the equilibrium point* (x^*, u^*).

One-dimensional scalar systems For one-dimensional systems with one input (that is, $x(t) \in \mathbb{R}$ and inputs $u(t) \in \mathbb{R}$), the *small signal linearization* of the control system (3.18) about the equilibrium point (x^*, u^*) is

$$\frac{d}{dt}\delta x = F\delta x + G\delta u, \tag{3.20}$$

or equivalently, emphasizing the dimensions of vectors and matrices:

$$\underset{(1\times1)}{\boxed{\dot{\delta x}}} \approx \underset{(1\times1)}{\boxed{F}}\;\underset{(1\times1)}{\boxed{\delta x}} + \underset{(1\times1)}{\boxed{G}}\underset{(1\times1)}{\boxed{\delta u}}. \tag{3.21}$$

The classic example is the so-called *small-angle approximation*. The nonlinear first-order system

$$\dot{\theta} = -\sin\theta$$

is approximated, for small angles θ, by the linear first-order system

$$\dot{\delta\theta} = -\sin\delta\theta$$

where the variable is $\delta\theta = \theta - 0$.

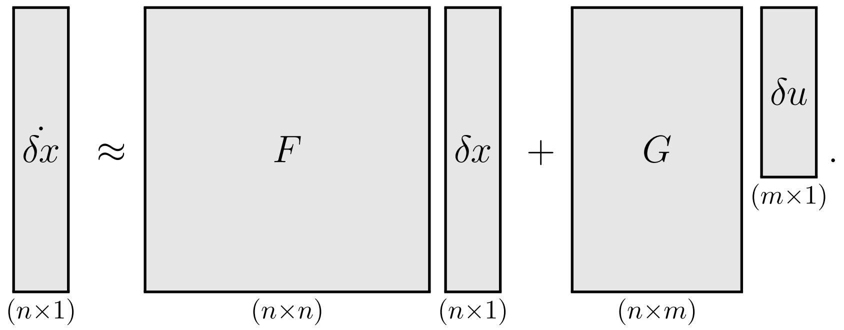
Two and larger dimensional systems For control systems with multiple states and inputs (that is, $x(t) \in \mathbb{R}^n$ and inputs $u(t) \in \mathbb{R}^m$), we recall that the matrix of partial derivatives is called the *Jacobian matrix*. Since f is a function of x and u, it has two Jacobian matrices:

$$F = \frac{\partial f}{\partial x}(x^*, u^*) \qquad \text{and} \qquad G = \frac{\partial f}{\partial u}(x^*, u^*). \tag{3.22}$$

In summary, the *small signal linearization* (or *Jacobian linearization*) of the control system (3.18) about the equilibrium point (x^*, u^*) is

$$\frac{d}{dt}\delta x = F\delta x + G\delta u, \tag{3.23}$$

or equivalently, emphasizing the dimensions of vectors and matrices:



(3.24)

Recall: $n = \#$ states and $m = \#$ inputs.

3.3.3 Linearizing the pendulum equations

As in Section 2.3.1, we consider a pendulum of length ℓ with mass m concentrated at its end, subject to gravity with constant g and to friction (due to air or due to the mechanical rotation at the pivot point) described by a damping coefficient b.

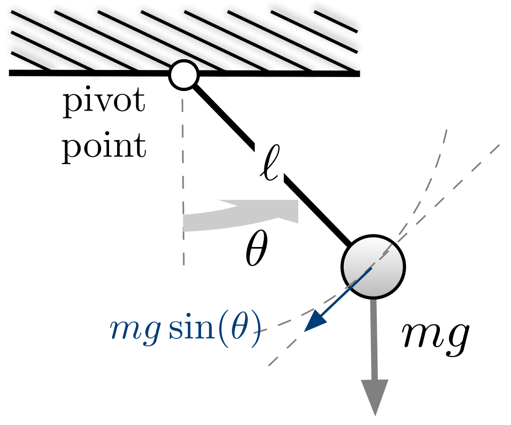


Figure 3.11: A pendulum subject to gravity, connected to a pivot point.
The variable is the angle θ, measured counterclockwise from the zero value when the pendulum is in its vertical rest state.
The moment of inertia of the pendulum about the pivot point is $I = m\ell^2$.
The pendulum is subject to the gravity force of magnitude mg, which translates into a restoring torque of magnitude $m\ell g\sin(\theta)$.

As in equation (2.27), the equations of motion are:

$$\begin{bmatrix}\dot\theta\\ \dot\omega\end{bmatrix} = \begin{bmatrix}\omega\\ -\frac{b}{m\ell^2}\omega - \frac{g}{\ell}\sin(\theta)\end{bmatrix} \tag{3.25}$$

with the two equilibria:

- the equilibrium point $\begin{bmatrix}\theta^*_{\text{down}}\\ \omega^*\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}$, corresponding to the pendulum in its *down position*,
- the equilibrium point $\begin{bmatrix}\theta^*_{\text{up}}\\ \omega^*\end{bmatrix} = \begin{bmatrix}\pi\\0\end{bmatrix}$, corresponding to the pendulum in its *up position*.

Linearization of the pendulum equations at the down and up positions

We compute the Jacobian matrix of (3.25) at an arbitrary point (θ, ω):

$$J(\theta,\omega) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{\ell}\cos\theta & -\frac{b}{m\ell^2} \end{bmatrix} \tag{3.26}$$

and evaluate it at the two equilibria $\theta^*_{\text{down}} = 0$ and $\theta^*_{\text{up}} = \pi$ (where $\omega^*_{\text{down}} = \omega^*_{\text{up}} = 0$):

$$J_{\text{down}}(\theta^*_{\text{down}} = 0, \omega^*_{\text{down}} = 0) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{\ell} & -\frac{b}{m\ell^2} \end{bmatrix} \quad \text{and} \quad J_{\text{up}}(\theta^*_{\text{up}} = \pi, \omega^*_{\text{up}} = 0) = \begin{bmatrix} 0 & 1 \\ \frac{g}{\ell} & -\frac{b}{m\ell^2} \end{bmatrix} \tag{3.27}$$

Therefore, the two linearizations of the pendulum dynamics are:

$$\begin{bmatrix} \dot{\delta\theta} \\ \dot{\delta\omega} \end{bmatrix} = J_{\text{down}} \begin{bmatrix} \delta\theta \\ \delta\omega \end{bmatrix} \tag{3.28}$$

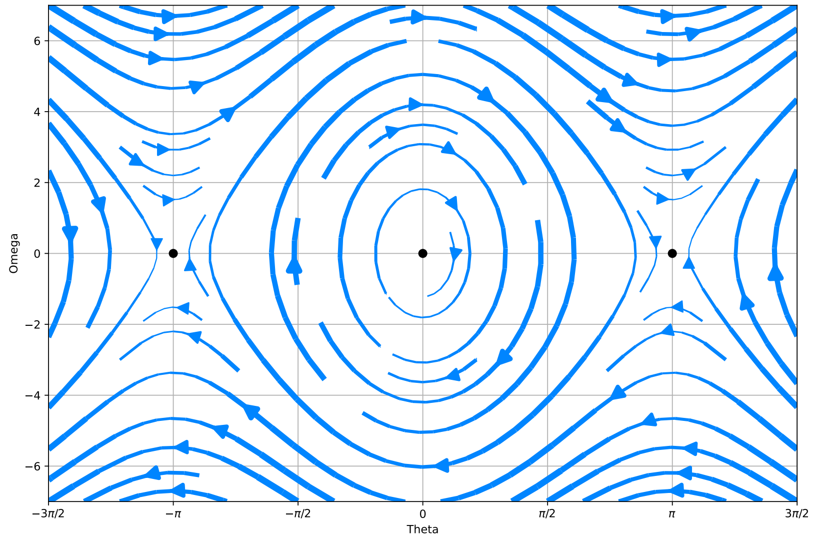
where $\delta\theta = \theta - \theta^*_{\text{down}} = \theta$ and $\delta\omega = \omega - \omega^*_{\text{down}} = \omega$, and

$$\begin{bmatrix} \dot{\delta\theta} \\ \dot{\delta\omega} \end{bmatrix} = J_{\text{up}} \begin{bmatrix} \delta\theta \\ \delta\omega \end{bmatrix} \tag{3.29}$$

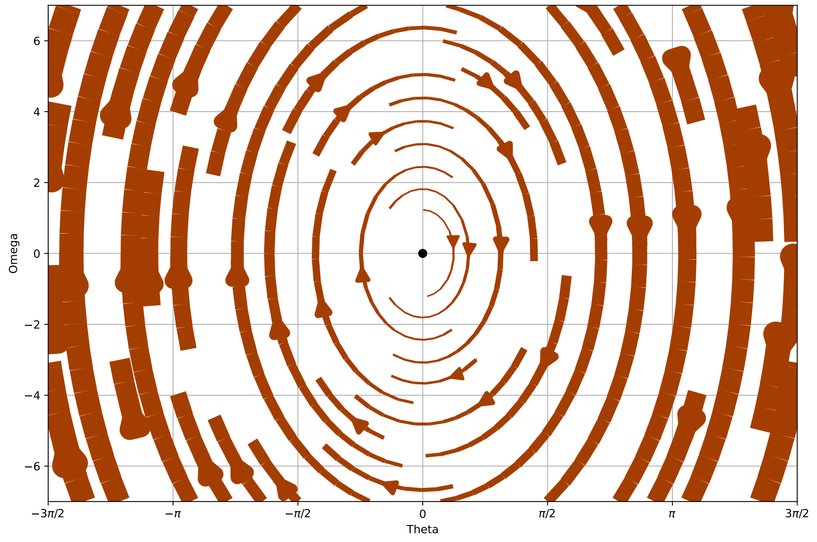
where $\delta\theta = \theta - \theta^*_{\text{up}} = \theta - \pi$ and $\delta\omega = \omega - \omega^*_{\text{up}} = \omega$, Intuitively, the pendulum down case corresponds to approximating $\sin(\theta)$ by θ. This approximation is reasonable for small θ and because the point $(\theta, \dot\theta) = (0, 0)$ is a stable equilibrium of the pendulum dynamics. We note that this approximated model is identical to a linear mass-spring-damper system:

$$\ddot\theta + \frac{b}{m\ell^2}\dot\theta + \frac{g}{\ell}\theta = 0. \tag{3.30}$$

Let us now verify empirically the impact of this approximation. We compare the phase portrait for the exact dynamics (2.25) with $\sin\theta$ and for the approximated dynamics, where $\sin\theta$ is replaced by θ.



(a) Exact pendulum dynamics



(b) Pendulum dynamics linearized at the pendulum down configuration

Figure 3.12: Phase portrait for the undamped pendulum dynamics and its linearization about the "pendulum down" configuration. Note that the phase portraits are reasonably similar for small θ.

3.4 Historical notes, further reading, and online resources

Additional example systems can be found in Ogata (2003).

For additional information about heat flow models, including for example the differential form of Fourier's law, we refer to wikipedia: Thermal conduction and wikipedia: Thermal conductivity.

It is instructive to consider how a binary ON/OFF controller is implemented in a circuit. A *relay circuit* is an electrical control device that typically uses an electromagnet to mechanically switch an electrical load on or off. It consists of a coil, a set of contacts, and a spring-loaded mechanism. When an electrical current is applied to the coil, a magnetic field attracts or repels the contacts, causing them to make or break an electrical connection. Relays are commonly used in automotive systems, industrial automation, and electronics. They are especially useful in high-voltage or high-current devices for electrical isolation and protection.

3.5 Appendix: Taylor expansions

Taylor Expansion for a Scalar Function of a Scalar Variable (n=1) For a function $f(x)$ that is differentiable at a point x^*, its *Taylor expansion up to the first order* about x^* is:

$$f(x) \approx f(x^*) + \frac{df}{dx}(x^*) \cdot (x - x^*)$$

where:

- $f(x^*)$ is the function at x^*,
- $\frac{df}{dx}(x^*)$ is the derivative of the function at x^*, and
- $x - x^*$ is the difference between the point of interest x and the expansion point x^*.

Equivalently, emphasizing the dimensions of vectors and matrices:

$$\underset{(1\times1)}{\boxed{f(x)}} \approx \underset{(1\times1)}{\boxed{f(x^*)}} + \underset{(1\times1)}{\boxed{\frac{df}{dx}(x^*)}}\underset{(1\times1)}{\boxed{x-x^*}} \tag{3.31}$$

Taylor Expansion for a Vector Function of a Vector Variable (n=2) Now we consider a differentiable function $f\colon \mathbb{R}^2 \to \mathbb{R}^2$ where

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$$

and

$$f(x) = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} \in \mathbb{R}^2.$$

The *first-order Taylor expansion of f about a point* $x^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} \in \mathbb{R}^2$ is:

$$f(x) \approx f(x^*) + J(x^*) \cdot (x - x^*)$$

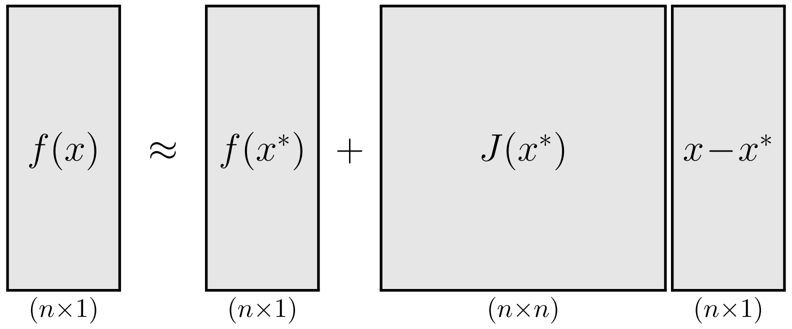
where:

- $f(x^*)$ is the vector function evaluated at x^*,
- $J(x^*)$ is the *Jacobian matrix* of f evaluated at x^*, defined by

$$J(x^*) = \frac{\partial f}{\partial x}(x^*) = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} \end{bmatrix}_{x=x^*}$$

- $x - x^*$ is the difference vector between the point of interest x and the expansion point x^*.

Equivalently, emphasizing the dimensions of vectors and matrices:



(3.32)

3.6 Appendix: Basic models in convective and radiative heat transfer

In this appendix, we present simple low-dimensional examples of convective and radiative heat transfer.

3.6.1 Convective heat transfer

This model describes how an object cools or heats over time when placed in an environment where heat is exchanged between the object's surface and the surrounding fluid. This model is commonly used to describe transient heat conduction in simple geometries with convective boundary conditions. The *convective heat transfer dynamics* is:

$$\frac{dT}{dt} = -\frac{hA}{mc}(T - T_\infty), \tag{3.33}$$

where:

- $T(t)$is the temperature of the object at time t,
- T_∞ is the ambient temperature (assumed constant),
- h is the convective heat transfer coefficient,
- A is the surface area of the object,
- m is the mass of the object, and
- c is the specific heat capacity.

This first-order linear ODE provides insights into cooling rates in real-world applications like engines, heat exchangers, or electronic devices.

3.6.2 Radiative cooling of a blackbody

The *Stefan-Boltzmann law* describes radiative heat transfer from an object to its surroundings. For a body emitting thermal radiation, the *radiative cooling dynamics of a blackbody* is

$$\frac{dT}{dt} = -\frac{\sigma\varepsilon A}{mc}\left(T^4 - T_\infty^4\right), \tag{3.34}$$

where:

- $T(t)$is the temperature of the object at time t,
- T_∞ is the ambient temperature (assumed constant),
- σ is the Stefan-Boltzmann constant,
- ε is the emissivity of the object,
- A is the surface area,
- m is the mass, and
- c is the specific heat capacity.

This nonlinear ODE models is especially relevant for high-temperature systems like engines, furnaces, or even spacecraft, where radiation is a dominant heat transfer mechanism. A key feature of this model is that radiation depends on T^4 and so the model is nonlinear.

3.7 Exercises

E3.1 **Positive and negative fluid flow in a pipe with positive and negative pressure difference.** Given a flow resistance r and a flow behavior parameter α, recall that equation (3.11) assumes $p_1 > p_2$ and provides a mass flow rate w that is always positive. When the pressure difference $p_1 - p_2$ can be both positive and negative, the resulting mass flow rate from 1 to 2 can be both positive and negative.

Let $w_{1\to 2}$ denote the signed flow ("signed" means positive or negative) from point 1 to point 2.

(i) Write an equation for $w_{1\to 2}$ as function of $p_1 - p_2$, when $p_1 - p_2$ can be both positive and negative.

Hint: In other words, given two arbitrary numbers p_1 and p_2 (without assuming necessarily that $p_1 > p_2$), how would you compute the mass flow rate?

(ii) Verify that your proposed equation reduces to equation (3.11), when $p_1 > p_2$.

(iii) Write an equation for $w_{2\to 1}$ and explain how it relates to $w_{1\to 2}$.

Answer:

(i) We propose

$$w_{1\to 2} = \operatorname{sign}(p_1 - p_2)\frac{1}{r}(|p_1 - p_2|)^{1/\alpha} = \begin{cases} \frac{1}{r}(p_1 - p_2)^{1/\alpha} & \text{if } p_1 \geq p_2 \\ -\frac{1}{r}(p_2 - p_1)^{1/\alpha} & \text{if } p_2 > p_1 \end{cases} \tag{E3.1}$$

(ii) When $p_1 > p_2$, we compute

$$w_{1\to 2} = \operatorname{sign}(p_1 - p_2)\frac{1}{r}(|p_1 - p_2|)^{1/\alpha} = \frac{1}{r}(p_1 - p_2)^{1/\alpha} \tag{E3.2}$$

(iii) $w_{2\to 1} = -w_{1\to 2} = \operatorname{sign}(p_2 - p_1)\frac{1}{r}(|p_1 - p_2|)^{1/\alpha}$

■

E3.2 **Water heater model.** Consider a water tank above a coil heater as in Figure E3.1.

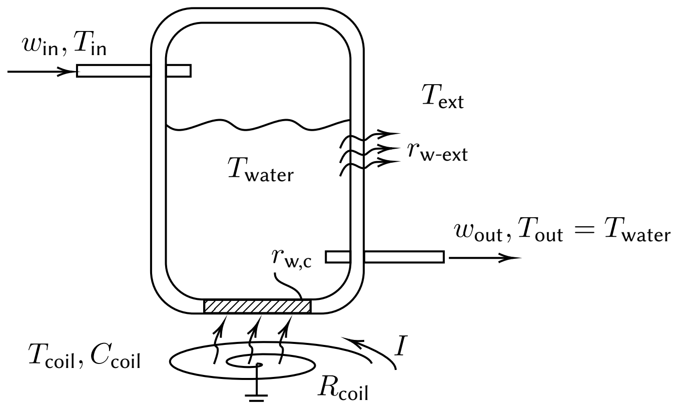

$w_{\text{in}}, T_{\text{in}}$
T_{ext}
$r_{\text{w-ext}}$
T_{water}
$w_{\text{out}}, T_{\text{out}} = T_{\text{water}}$
$r_{\text{w,c}}$
$T_{\text{coil}}, C_{\text{coil}}$
I
R_{coil}

Figure E3.1: A water tank model.

The water in the tank is thermally connected to (i) a coil heater through a thermal resistance $r_{\text{w,c}}$ and (ii) the external environment through a thermal resistance $r_{\text{w-ext}}$.

Let T_{water}, T_{coil} and T_{ext} denote the temperatures of water, coil heater, and external environment, respectively.

(i) Using Fourier's law, write an equation for the heat flow $q_{\text{coil}\to\text{water}}$ from coil to tank, and $q_{\text{water}\to\text{ext}}$ from tank to external environment.

Now, assume that the coil heater warms up due to Joule resistive heating with its electrical resistance R_{coil} when a current I is applied. This means that the energy produced per second, that is, power generated is:

$$P = R_{\text{coil}} I^2 \qquad \text{(measured in watts = Joule per sec)}$$

Let C_{coil} be the total heat capacity of the coil heater. Let m denote the water mass in the tank, let c_{water} denote the *specific heat capacity*, that is, the amount of heat required per unit mass to raise the temperature of water by one-degree Celsius. Therefore, mc_{water} is the total heat capacity of the water in the heater. For now, assume no water flows into and out of the tank (both valves are closed).

(ii) Write an equation for the evolution of the temperature of the tank water and of the coil.

Finally, open the both valves and assume $w_{\text{in}} > 0$ and $w_{\text{out}} > 0$ are incoming and outgoing mass flow rates, as in the mass balance equation (3.9). Let T_{in} denote the temperature of the incoming water. Note that

- the *total thermal energy* of the tank water is $c_{\text{water}} m T_{\text{water}}$,
- due to the incoming water, energy is added to the water tank at a rate $q_{\text{in}} = w_{\text{in}} c_{\text{water}} T_{\text{in}}$ and subtracted at a rate $q_{\text{out}} = w_{\text{out}} c_{\text{water}} T_{\text{water}}$.

(iii) Write the balance equation for the time-derivative of the total thermal energy.

(iv) Using the formula from (iii) and the mass balance equation $\dot{m} = w_{\text{in}} - w_{\text{out}}$, obtain a single equation for $\dot{T}_{\text{water}}$.

(v) Collect the various results and write the resulting control system with three variables (m, T_{coil}, and T_{water}) with two inputs (w_{in}, I^2) and a disturbance (w_{out}).

Answer:

(i) According to Fourier's law, the heat flow from coil to water and from water to external environment are

$$\boxed{q_{\text{coil}\to\text{water}} = \frac{1}{r_{\text{w,c}}}(T_{\text{coil}} - T_{\text{water}})} \qquad \text{and} \qquad \boxed{q_{\text{water}\to\text{ext}} = \frac{1}{r_{\text{w-ext}}}(T_{\text{water}} - T_{\text{ext}})}$$

(ii) The temperature change of the coil heater is:

$$C_{\text{coil}}\dot{T}_{\text{coil}} = -q_{\text{coil}\to\text{water}} + P \qquad \Longrightarrow \qquad \boxed{C_{\text{coil}}\dot{T}_{\text{coil}} = \frac{1}{r_{\text{w,c}}}(T_{\text{water}} - T_{\text{coil}}) + I^2 R_{\text{coil}}}$$

The temperature change of the tank water is:

$$mc_{\text{water}}\dot{T}_{\text{water}} = -q_{\text{water}\to\text{coil}} + q_{\text{water}\to\text{ext}} \qquad \Longrightarrow \qquad \boxed{mc_{\text{water}}\dot{T}_{\text{water}} = \frac{1}{r_{\text{w,c}}}(T_{\text{coil}} - T_{\text{water}}) + \frac{1}{r_{\text{w-ext}}}(T_{\text{ext}} - T_{\text{water}})}$$

(iii) Now, both valves are opened and cause a change of temperature and water mass. The total thermal energy changes according to

$$\frac{d(mc_{\text{water}}T_{\text{water}})}{dt} = q_{\text{coil}\to\text{water}} - q_{\text{water}\to\text{ext}} + q_{\text{in}} - q_{\text{out}}$$

$$\Longrightarrow \quad \boxed{\frac{dm}{dt}c_{\text{water}}T_{\text{water}} + mc_{\text{water}}\frac{dT_{\text{water}}}{dt} = \frac{1}{r_{\text{w,c}}}(T_{\text{coil}} - T_{\text{water}}) + \frac{1}{r_{\text{w-ext}}}(T_{\text{ext}} - T_{\text{water}}) + w_{\text{in}}c_{\text{water}}T_{\text{in}} - w_{\text{out}}c_{\text{water}}T_{\text{water}}} \tag{E3.3}$$

(iv) Plugging the mass balance equation $\dot{m} = w_{\text{in}} - w_{\text{out}}$ into (E3.3) we obtain

$$mc_{\text{water}}\frac{dT_{\text{water}}}{dt} = \frac{1}{r_{\text{w,c}}}(T_{\text{coil}} - T_{\text{water}}) + \frac{1}{r_{\text{w-ext}}}(T_{\text{ext}} - T_{\text{water}}) + w_{\text{in}}c_{\text{water}}T_{\text{in}} - w_{\text{out}}c_{\text{water}}T_{\text{water}} - (w_{\text{in}} - w_{\text{out}})c_{\text{water}}T_{\text{water}} \tag{E3.4}$$

so that, after a simplification,

$$\boxed{m\frac{dT_{\text{water}}}{dt} = \frac{1}{r_{\text{w,c}}c_{\text{water}}}(T_{\text{coil}} - T_{\text{water}}) + \frac{1}{r_{\text{w-ext}}c_{\text{water}}}(T_{\text{ext}} - T_{\text{water}}) + w_{\text{in}}(T_{\text{in}} - T_{\text{water}})}\,. \tag{E3.5}$$

(v) In summary, we have a control system with three variables (m, T_{coil}, and T_{water}) with two inputs (w_{in}, I^2) and a disturbance (w_{out}):

$$\boxed{\dot{m} = w_{\text{in}} - w_{\text{out}}}\,,$$

$$\boxed{C_{\text{coil}}\dot{T}_{\text{coil}} = \frac{1}{r_{\text{w,c}}}(T_{\text{water}} - T_{\text{coil}}) + I^2 R_{\text{coil}}}\,,$$

$$\boxed{m\frac{dT_{\text{water}}}{dt} = \frac{1}{r_{\text{w,c}}c_{\text{water}}}(T_{\text{coil}} - T_{\text{water}}) + \frac{1}{r_{\text{w-ext}}c_{\text{water}}}(T_{\text{ext}} - T_{\text{water}}) + w_{\text{in}}(T_{\text{in}} - T_{\text{water}})}\,.$$

■

E3.3 **Epidemic model and linearization.** In this exercise we explore the dynamics of epidemics and the occurrence of *epidemic outbreaks*. Given a population and a pathogen, we assume that each individual is in one of three possible states:

susceptible: the individual is not infected, but is vulnerable to being infected,

infected: the individual is infected and capable of transmitting the disease to susceptible individuals, and

recovered: the individual has been infected and has now recovered, gaining immunity to the pathogen.

Mathematically, we let s, x, r denote the fractions of *susceptible*, *infected*, and *recovered individuals*, respectively. Note that $s(t) + x(t) + r(t) = 1$ at all times t. We call this simplified model the SIR model and we allow only two types of transitions:[4]

susceptible $\rightarrow$ infected: this transition is due to an interaction between one susceptible individual and one infected individual. Therefore, the transition rate depends upon the likelihood of contact between individuals in the two states. Mathematically, the transition rate equals βsx, where the *infection rate* $\beta > 0$ is describes how transmissible is the pathogen;

infected $\rightarrow$ recovered: this transition is spontaneous, independent of interactions, and proportional to the fraction of infected individuals. The transition rate is γx, where the *recovery rate* $\gamma > 0$ describes the decay rate of the infection.

We illustrate the transitions and their rates in Figure E3.2.

$$\begin{tikzcd} \boxed{\begin{array}{c} s = \text{fraction} \\ \text{susceptible} \end{array}} \arrow[r, "\beta sx"] & \boxed{\begin{array}{c} x = \text{fraction} \\ \text{infected} \end{array}} \arrow[r, "\gamma x"] & \boxed{\begin{array}{c} r = \text{fraction} \\ \text{recovered} \end{array}} \end{tikzcd}$$

Figure E3.2: The three states, the two possible transitions, and the transition rates for the two possible transitions.

In summary, our proposed *SIR epidemic model* is

$$\dot{s} = -\beta sx \tag{E3.6}$$
$$\dot{x} = \beta sx - \gamma x \tag{E3.7}$$
$$\dot{r} = \gamma x. \tag{E3.8}$$

(i) How many variables and parameters are in this epidemic system? (Here we mean variables dependent on time and we do not count time)

(ii) Identify all equilibria of the system.

(iii) Linearize the model around the equilibrium $(1, 0, 0)$.

(iv) Show that, for the linearized dynamics, the time derivative of the infected individuals δx is independent of the values of δs and δr.

(v) An *epidemic outbreak* is said to occur when: (i) all individuals are susceptible except for a small positive fraction of infected individuals, and (ii) the number of infected individuals starts to grow exponentially fast. Use the linearized model from questions (iii) and (iv) to find a condition on the ratio β/γ that leads to an epidemic outbreak.

***Note:** Some additional analysis explains that the ratio β/γ equals the R_0 value in the literature. The R_0 value equals the average number of individuals that an infected individual will infect. This value was widely publicized during the COVID-19 pandemic.*

[4] More realistic models include more states and more possible transitions.

Answer:

(i) 3 variables and 2 parameters

(ii) We can see that it is both necessary and sufficient that $x = 0$ for the system to be in equilibrium.

Thus, all points of the form $(a, 0, b)$ are equilibria.

(Since $s + x + r$ is a conserved quantity, we can equivalently write $(a, 0, 1 - a)$, assuming that the system starts from $(1, 0, 0)$.)

(iii) Linearizing the model about an arbitrary point (s, x, r) yields the following system:

$$\frac{d}{dt}\begin{bmatrix}\delta s\\ \delta x\\ \delta r\end{bmatrix} = \begin{bmatrix}-\beta x & -\beta s & 0\\ \beta x & \beta s - \gamma & 0\\ 0 & \gamma & 0\end{bmatrix}\begin{bmatrix}\delta s\\ \delta x\\ \delta r\end{bmatrix}. \tag{E3.9}$$

Evaluating the linearized system at the point $(1, 0, 0)$ yields

$$\frac{d}{dt}\begin{bmatrix}\delta s\\ \delta x\\ \delta r\end{bmatrix} = \begin{bmatrix}0 & -\beta & 0\\ 0 & \beta - \gamma & 0\\ 0 & \gamma & 0\end{bmatrix}\begin{bmatrix}\delta s\\ \delta x\\ \delta r\end{bmatrix} \tag{E3.10}$$

(iv) The equation $\dot{\delta x} = (\beta - \gamma)\delta x$ is decoupled from the evolution of the susceptible and recovered fractions.

(v) We can see that the linear system $\dot{\delta x} = (\beta - \gamma)\delta x$ is a linear growth model (studied in Chapter 1) and it is an unstable system if $\beta - \gamma > 0$. Manipulating this expression, we obtain the equivalent condition:

if $\beta/\gamma > 1$, then the linearized system is unstable

When this linearized system is unstable, $\delta x(t)$ grows exponentially fast (for small times, before we leave the neighborhood of the equilibrium point $(1, 0, 0)$) and so we have an epidemic outbreak.

This answer to question (v) shows that, when $R_0 > 1$, there is an epidemic outbreak.

■

E3.4 **Linearization of the water tank dynamics with input.** Recall that, given two positive coefficients a, b, the water tank dynamics is

$$\dot{h}(t) = -a\sqrt{h(t)} + bw(t) \tag{E3.11}$$

where $h(t)$ is the water height and $w(t) \geq 0$ is the input signal.

(i) Compute each possible equilibrium point of the system.
Hint: Recall that an equilibrium point is a pair (h^*, w^*).

(ii) Compute the small signal linearization of the system at an equilibrium point such that $w^* > 0$.

(iii) Is it possible to compute the small-signal linearization at the equilibrium point such that $w^* = 0$? It not, explain why not. Otherwise compute it.
Note: *As discussed after the flow resistance model (3.11), when h is small and the flow is slow, then the assumption $\alpha = 2$ is inappropriate.*

E3.5 **Linearization of a mass-spring-damper system with a nonlinear spring .** Consider a nonlinear spring with zero rest length and restoring force equal, in magnitude, to

$$f_{\text{nonlinear-spring}}(x) = k_1 x + k_2 x^3 \tag{E3.12}$$

where x is the displacement. Consider a mass-spring-damper system with this nonlinear spring:

$$m\ddot{x} + b\dot{x} + k_1 x + k_2 x^3 = f \tag{E3.13}$$

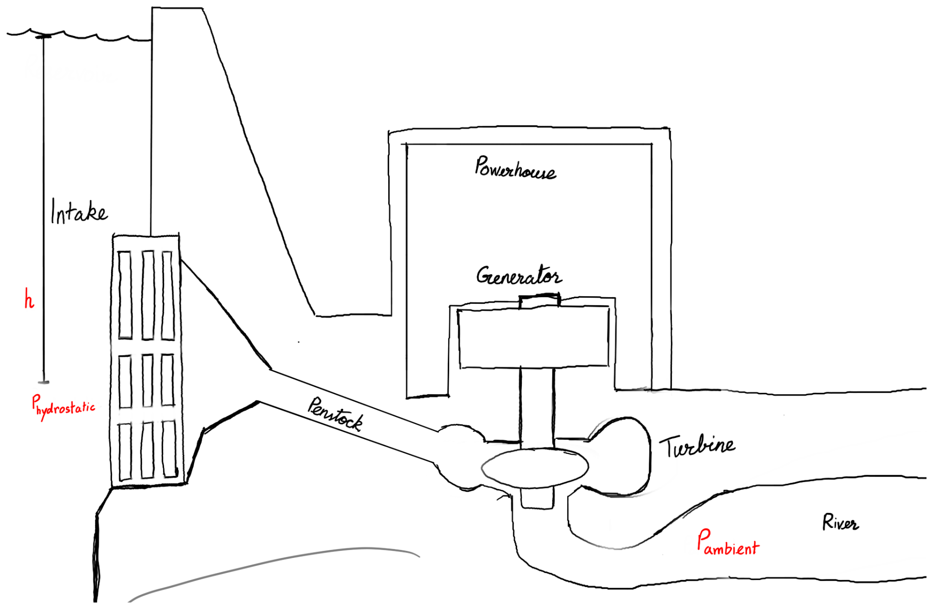
(i) Assume $k_1 = 0$. For each value of f (positive and negative) and $k_2 > 0$, compute each possible equilibrium point of the mass-spring-damper system.
Hint: As a minor point, recall that an equilibrium point for this dynamical system is not just a position x^*.

(ii) If $k_1 > 0$, $k_2 > 0$, and $f = k_1 + k_2$, what is the equilibrium? Let $(x^*, 0)$ denote this equilibrium point.

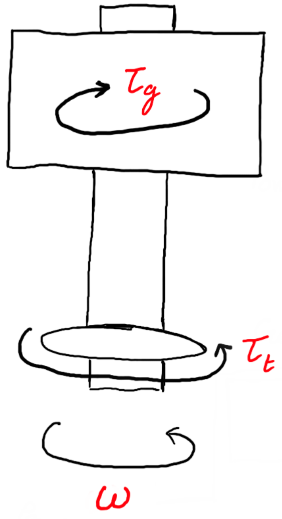
(iii) Compute the small signal linearization of the mass-spring-damper system at the equilibrium point $(x^*, 0)$ when $f = k_1 + k_2$.

E3.6 **Hydroelectric dam.** Consider the hydroelectric dam shown in Figure E3.3a. When the intake is open, water flows from the reservoir through the penstock and out the outlet. The turbine and generator are coupled with a shaft so that energy is extracted from the flowing water and converted into electricity.

- The state variable for the hydroelectric dam control system is the rotation speed of the turbine-generator shaft ω.
- The hydraulic head h is the input signal to the hydroelectric dam control system.
- We assume that the pressure at the penstock intake is $p_{\text{hydrostatic}}$, and the pressure at the outlet is p_{ambient}. The combined flow resistance of the penstock and turbine is r, and the flow behavior parameter is $\alpha = 2$ (recall the flow resistance law in equation (3.11)). The density of the water is ρ, and the acceleration due to gravity is g. As shown in Figure E3.3b, the shaft has moment of inertia I and experiences a torque τ_{t} due to the water flowing through the turbine which is counteracted by a torque τ_{g} due to the generator.



(a) Schematic of a hydroelectric dam



(b) Free body diagram of turbine-generator shaft

Figure E3.3: Schematic and free body diagram of a hydroelectric dam.

(i) Derive the equation of motion for the rotation speed ω of the turbine-generator shaft in terms of the moment of inertia I and the torques $\tau_{\text{t}}, \tau_{\text{g}}$.

(ii) The turbine torque $\tau_{\text{t}} = (c_t P_t)/\omega$ where c_t is an efficiency constant (unitless) and P_t is the mechanical power of the turbine (in watts). We have the following equation for P_t:

$$P_t = q(p_{\text{hydrostatic}} - p_{\text{ambient}})$$

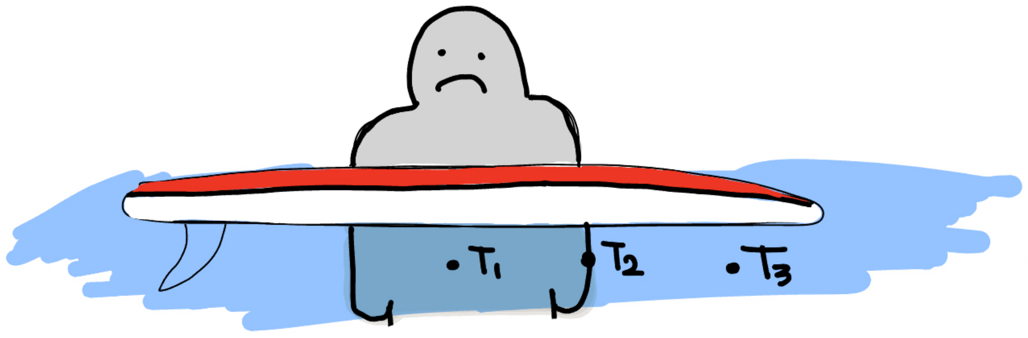
where q is the volumetric flowrate[5] through the penstock (in cubic meters per second). Use these relations along with others from the chapter to write an expression for the turbine torque τ_{t} in terms of the state ω, the input h, and other defined quantities.

(iii) Substitute the expression found in part (ii) into the equation of motion found in part (i). Identify the equilibrium of the resulting system. Sketch a phase portrait and classify the stability of the equilibrium.

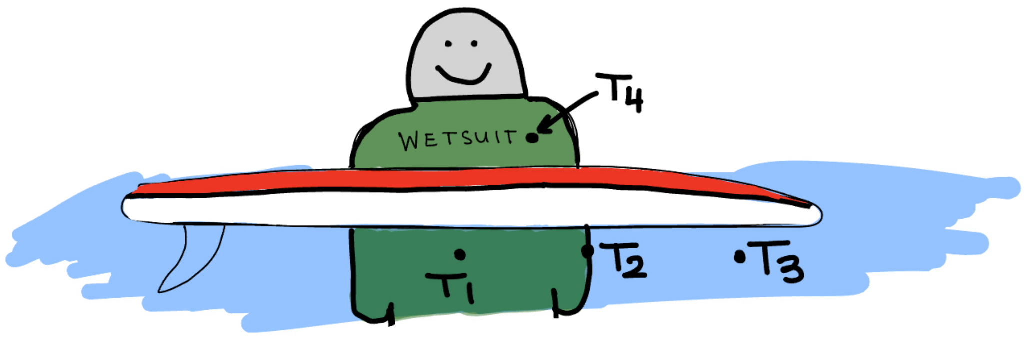
(iv) Linearize the system about the identified equilibrium. Use the notation $\delta\omega, \delta h$ for the transformed coordinates. Use the linearized model to verify the stability or instability of the equilibrium.

[5] For the purpose of this exercise, assume the volumetric flow rate $q = w/\rho$, where w is the flow rate and ρ the water density.

E3.7 **Surf's up, the thermal dynamics of surfing.** In this problem, we will investigate how the presence or absence of a wetsuit affects the body temperature of a person while surfing in cold water. Assume that all heat transfer occurs through conduction. The system can be analyzed in two scenarios: (a) without a wetsuit and (b) with a wetsuit. These configurations are illustrated in Figure E3.4.



(a) System with no wetsuit



(b) System with a wetsuit

Figure E3.4: Person in the water (a) without a wetsuit and (b) with a wetsuit.

No Wetsuit Denote the core body temperature by T_1 and thermal capacity by c_1, and the surface temperature of the body by T_2 and its thermal capacity by c_2. Assume that the body generates heat at a rate q_{body}. Assume the water temperature T_3 is constant.

(i) Write down the dynamics for the system in Figure E3.4a.

(ii) Derive an expression for the equilibrium temperature of the body. Express the result in terms of T_1, q_{body}, and the relevant thermal resistances.

With Wetsuit In the second scenario, a wetsuit is introduced into the system. Let T_4 represent the temperature at the surface of the wetsuit. The rest of the setup remains the same.

(iii) Write down the new dynamics for the system in Figure E3.4b.

(iv) Derive a new expression for the equilibrium temperature of the body. Express the result in terms of T_1, T_3 and the appropriate thermal resistances.

(v) Assume the thermal resistances are related by $r_{12} = r_{34} = r_{23} = \frac{1}{10} r_{24}$. How much greater is the difference $T_1 - T_3$ with a wetsuit than without?

Hint: Recall that the equilibrium point is found where $\dot{T}_i = 0$ for $i \in 1, 2, 3, 4$.

Bibliography

K. Ogata. *Dynamical Systems*. Pearson, 4 edition, 2003. ISBN 0131424629.