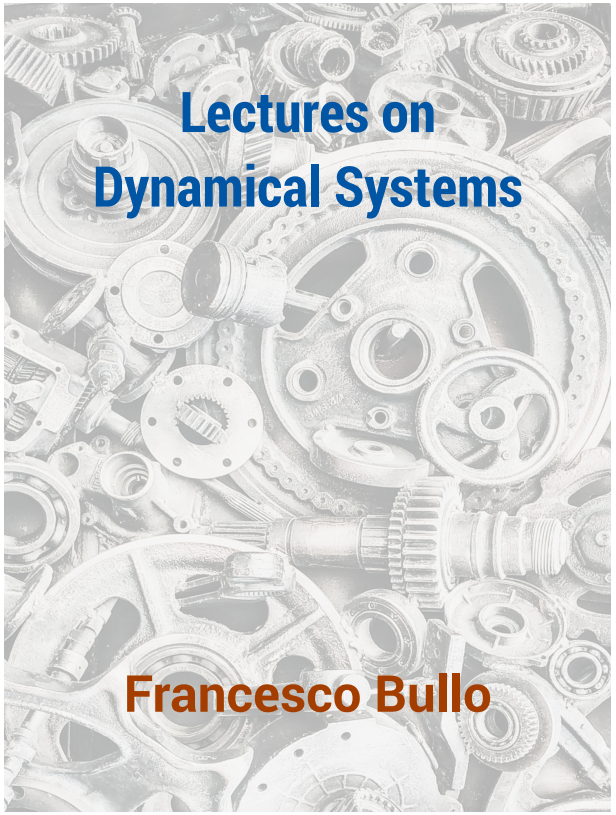
UC Santa Barbara, Department of Mechanical Engineering

ME103 Dynamical Systems. Slides by Chapter.

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



Contents

COURSE EVALUATIONS

To access the evaluation, scan this QR code with your mobile phone or click on this link:

https://go.blueja.io/CQwaIYTFpk6bi0HjAsPc4Q

1

[1] For instructor to monitor the completion rate: https://ucsb.bluera.com/ucsb/svml.aspx?regl=en-us

Chapter 7

Control Systems: Basic Definitions and Concepts

In this chapter we cover basic definitions by studying the cruise control system. We introduce block diagrams (in the time domain) and consider open-loop feedforward systems and closed-loop feedback systems. We define two basic control actions: proportional and integral action (we will study the derivative control action in the next chapter). We evaluate the advantages of closed-loop feedback control and discuss basic ideas on how to tune control gains.

7.1 From cruise control to static and first-order models

7.1.1 Static and first-order models for cruise control

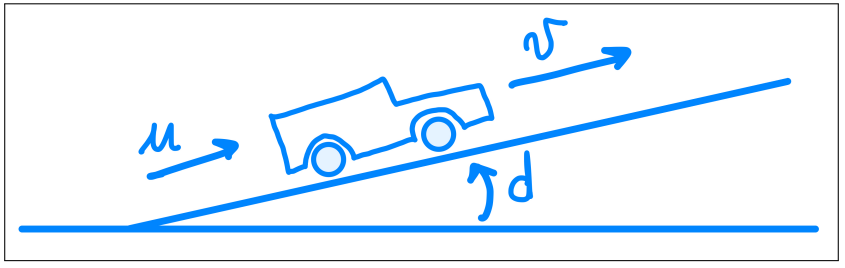


Figure 7.1: Car velocity dynamics on an inclined road. The signal u is the *throttle angle* to the car engine and the signal d is the *road inclination*.

We start by reviewing the car velocity dynamics, which is a first-order system:

$$m\dot{v} = -bv + f.$$

We now assume that the resultant force f on the car is generated by two forces: the engine force regulated by a throttle angle signal u and a road inclination d. In other words, the system is described by three signals:

- v is the car velocity, that is, the *system state*,
- u is the throttle angle, which is under our control. Hence, u is the *control input*.
- d is the road inclination, which is not under our control and it is a *disturbance*.

We assume the force on the car depends linearly upon the throttle angle u and on the road inclination d and write:

$$f = f_u u + f_d d \qquad \text{(where } f_u \text{ and } f_d \text{ are proportionality constants)}$$

Plugging in and dividing by b, we obtain a first-order system subject to control and disturbance:

$$\frac{m}{b}\dot{v}(t) \;=\; -v(t) \;+\; \frac{f_u}{b}u(t) \;+\; \frac{f_d}{b}d(t).$$

We rewrite in canonical form the *first-order system with control and disturbance* as

$$\tau \dot{v} + v \;=\; k_{\text{sys}} u + k_{\text{dist}} d \tag{7.1}$$

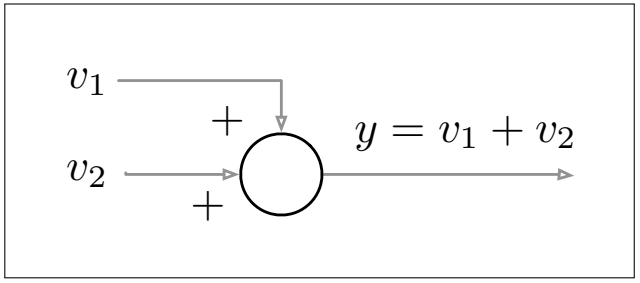
- τ is the system's time constant,
- k_{sys} is the gain from control to state, and
- k_{dist} is the gain from disturbance to state.

Assume now that the signals u and d are constant (or change very slowly, much more slowly than the time constant τ). Then, at equilibrium (or at steady state), we have a *static system with control and disturbance*:

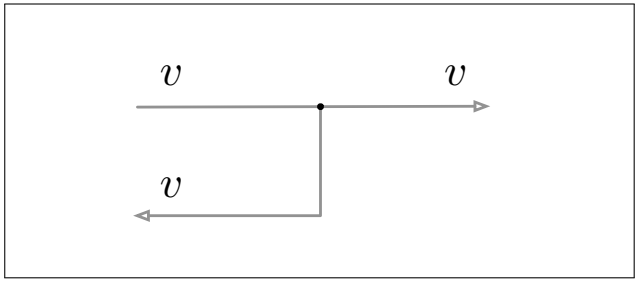
$$v \;=\; k_{\text{sys}} u + k_{\text{dist}} d. \tag{7.2}$$

7.1.2 Block diagrams in the time domain

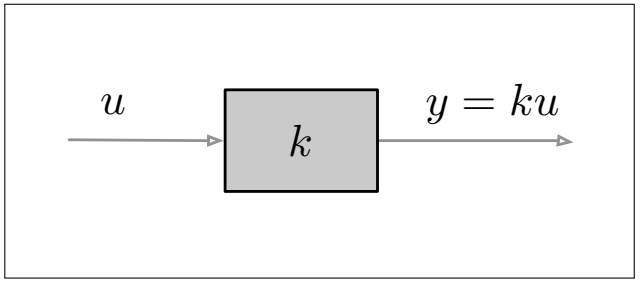
We wish to represent visually dynamical systems with multiple inputs and interconnections. A *block diagram* consists of the interconnection of four basic types of elements.



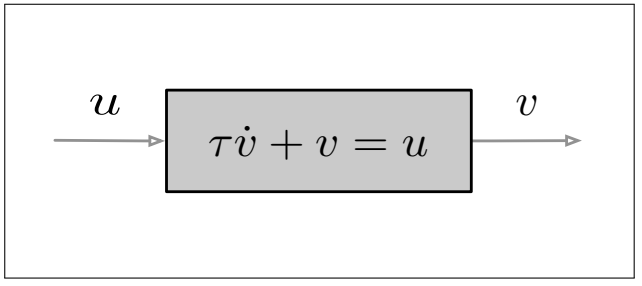
(a) Directed lines representing unidirectional signal flow and a *summing point*. Note that the summing point may includes positive of negative signs to indicate how to compute the algebraic sum.



(b) A *takeoff point*: the signal v is transmitted to multiple destinations.



(c) A block describing a *static input/output relationship*, meaning just a product.

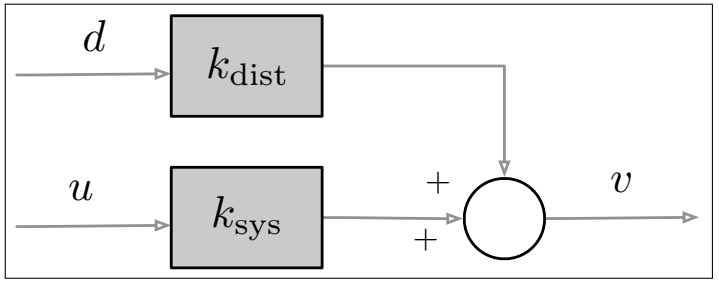


(d) A block describing a *dynamic input/output relationship*. Given a signal $u(t)$ as input, the output is the solution $v(t)$ of the control system.

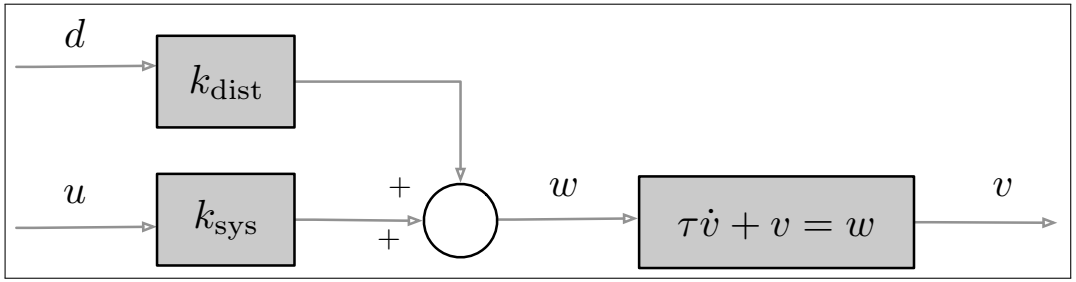
Figure 7.2: Illustrating four elements constituting a block diagram. We will show example of how to combine these blocks in the next slides.

7.1.3 Block diagrams for open and closed-loop control

We visualize static and dynamic models via *block diagrams*.



(a) Static model in equation (7.2)



(b) Dynamic model in equation (7.1)

Figure 7.3: Block diagrams for a static and dynamic model for the car velocity system

7.2 Open-loop and closed-loop control for the static model

In this section we

(i) design an open-loop (proportional) controller for the static model,

(ii) design a closed-loop (proportional) controller for the static model, and

(iii) compare them and draw some lessons on the benefits of closed-loop control.

Control objectives: Apply an appropriate control signal u to ensure that
(i) the car moves at a reference velocity r (*reference tracking*), and
(ii) the car speed is independent of the disturbance d (*disturbance rejection*).

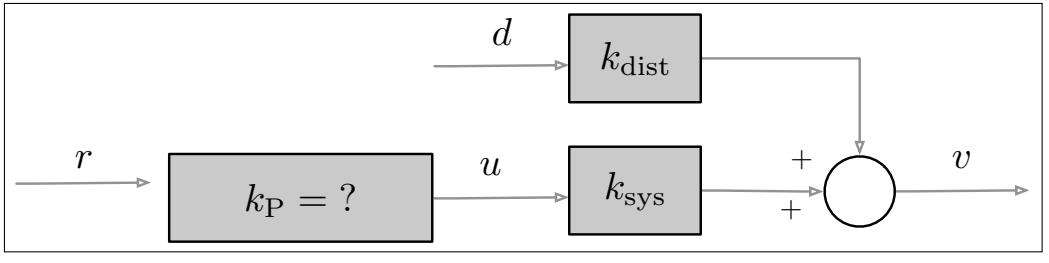


Figure 7.4: Static model with state variable v, input variable u, disturbance variable d, and reference variable r:
$v = k_{\text{sys}}u + k_{\text{dist}}d$
with an input
$u = k_{\text{P}}r$
where the gain k_{P} is not yet specified.

The input action $u = k_{\text{P}}r$ is called *proportional*.

In class assignment

Given a static system with control and disturbance in equation (7.2) and a proportional controller block with an unspecified gain as in Figure 7.4, what gain k_{P} would you choose to achieve reference tracking (i.e., $v = r$)?

7.2.1 Open-loop cruise control (static model) for regulation to reference speed

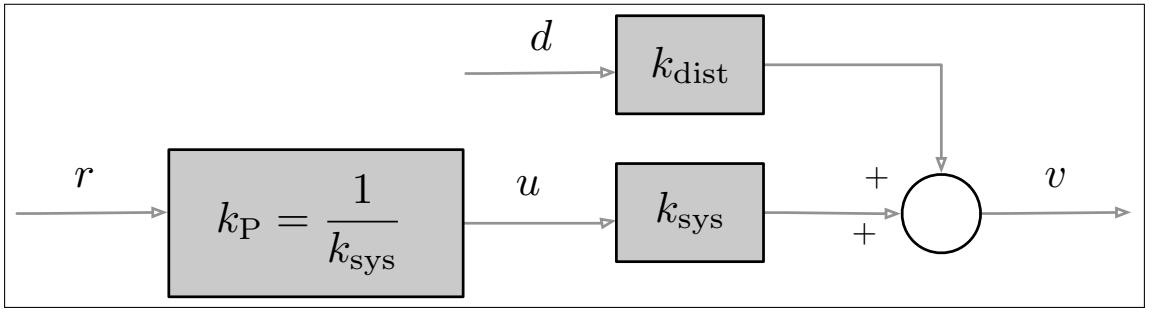


Figure 7.5: Open-loop cruise control of the static car velocity model, illustrated via a block diagram with a control block. The controller block has a gain equal to $1/k_{\text{sys}}$, the exact inverse of the effect of the control on the system.

- *Open-loop control design:* When the car dynamics is such that the gain from control to state is precisely k_{sys} (and there are no disturbances), then a good strategy is to adopt a *proportional control action*

$$u = k_{\text{P}} r, \qquad \text{where } k_{\text{P}} \text{ is a } \textit{control gain}. \quad \text{We select } k_{\text{P}} = \frac{1}{k_{\text{sys}}} \tag{7.3}$$

  Let us compute the response under an open-loop proportional control:

$$v_{\text{open-loop}} = k_{\text{sys}} u + k_{\text{dist}} d \Big|_{u = k_{\text{P}} r,\, k_{\text{P}} = 1/k_{\text{sys}}} = k_{\text{P}} k_{\text{sys}} r + k_{\text{dist}} d \Big|_{k_{\text{P}} = 1/k_{\text{sys}}} = r + k_{\text{dist}} d \tag{7.4}$$

- Lessons:

  (i) This strategy achieves reference tracking at zero disturbance.

  (ii) There are two drawback to this control strategy: we need to know k_{sys} exactly and we are unable to compensate for the disturbance d.

7.2.2 Closed-loop cruise control (static model) via negative feedback

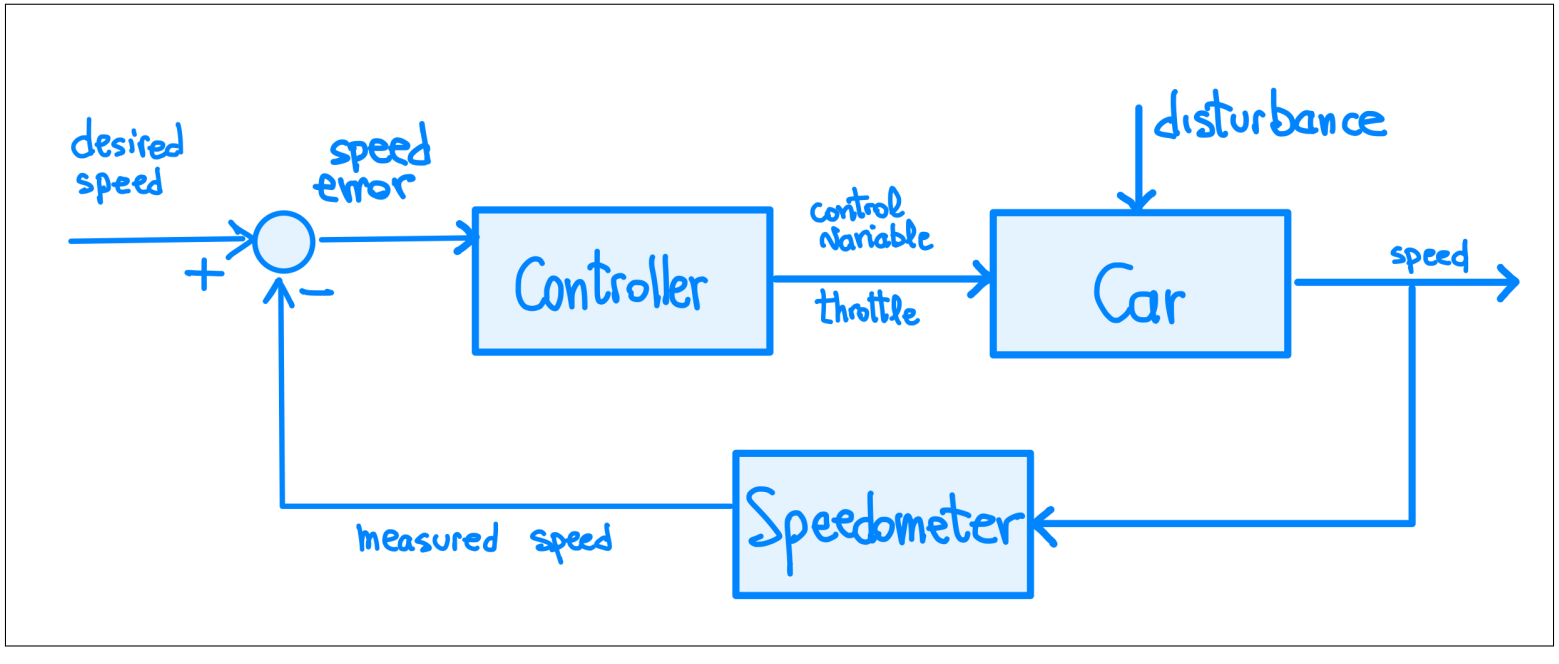


Figure 7.6: Block diagram of a more realistic closed-loop control architecture for cruise control

Key idea of feedback control and closed-loop systems in two steps:

(i) Compare the actual result with the desired result.

(ii) Take actions based on the difference.

This seemingly simple idea is very powerful and it is at the heart of the control engineering.

Closed-loop cruise control (static model) via negative feedback

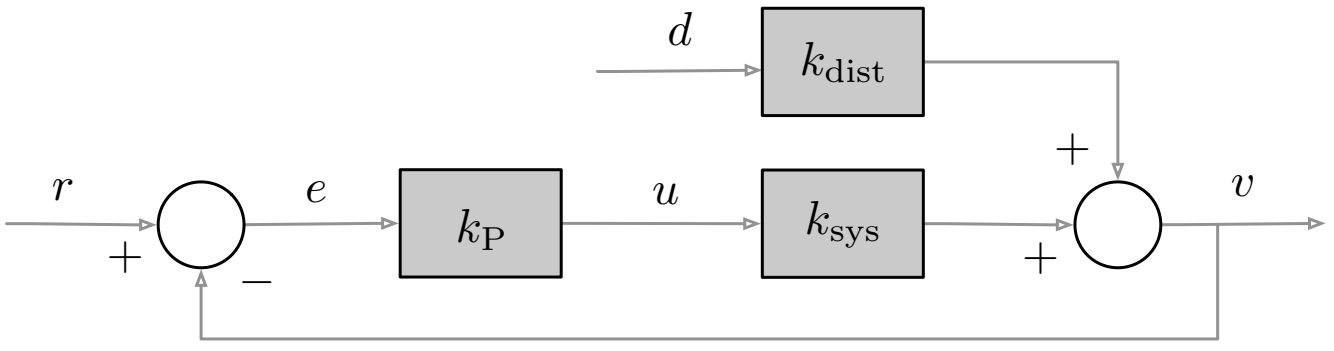


Figure 7.7: Closed-loop feedback control of the static car velocity model.
Illustration via a block diagram with control block and a feedback loop.
Note that the light blue boxes describes, respectively, the *system* (to be controlled and subject to a disturbance) and the *controller*.
In the closed-loop case, it is advantageous to set controller gain to be large.

- The key concept is to *compare* the reference signal with the actual signal and use this information to compute the control signal.
- The *error signal* is:

$$e = r - v_{\text{closed-loop}} \tag{7.5}$$

- The diagram contains a *feedback loop* with negative sign. Hence, this strategy is called *negative feedback*. In a feedback loop, the control gain multiplies the error signal. Hence, potentially, we can select it to be large. Certainly it is not calibrated to be the inverse of the system gain.
- As first control strategy, we use a *proportional controller* and we let k_{P} denote the *control gain*.

Analysis of the closed-loop velocity system (static model): We describe the closed-loop in the block diagram via two equations:

$$\begin{cases} v_{\text{closed-loop}} &= k_{\text{sys}} u + k_{\text{dist}} d \\ u &= k_{\text{P}} e = k_{\text{P}}(r - v_{\text{closed-loop}}) \end{cases} \tag{7.6}$$

Hence

$$v_{\text{closed-loop}} = k_{\text{sys}} k_{\text{P}} r - k_{\text{sys}} k_{\text{P}} v_{\text{closed-loop}} + k_{\text{dist}} d. \tag{7.7}$$

In summary,

$$v_{\text{closed-loop}} = \frac{k_{\text{sys}} k_{\text{P}}}{1 + k_{\text{sys}} k_{\text{P}}} r + \frac{k_{\text{dist}}}{1 + k_{\text{sys}} k_{\text{P}}} d \overset{k_{\text{P}} k_{\text{sys}} \text{ large}}{\approx} r \tag{7.8}$$

where the last approximate equality holds when we select the control gain to be large and satisfy $k_{\text{P}} k_{\text{sys}} \gg 1$ and $k_{\text{P}} k_{\text{sys}} \gg k_{\text{dist}}$.

For convenience of comparison, recall equation (7.4) for the open-loop case:

$$v_{\text{open-loop}} = r + k_{\text{dist}} d \tag{7.9}$$

Remark 7.1. *(i) Compared with the open-loop strategy, the closed-loop strategy (with a large gain k_{P}) has the potential to achieve both: approximate reference tracking $v \approx r$ without exact knowledge of k_{sys} and approximate disturbance rejection.*

(ii) We will soon show how to eliminate the tracking error (so that v will converge exactly to r) using so-called integral control.

(iii) The disturbance attenuation is due to the large "open-loop gain" $k_{\text{P}} k_{\text{sys}}$ from e to the output v. However, it is not always possible to simply increase the gain to reduce the effects of the disturbance d. For example, the magnitude of the control input may become so large to be outside the capabilities (and the linear functionality regime) of the car engine. We will study this tradeoff later.

•

7.3 Closed-loop control for the dynamic model

In this section we study whether closed-loop control works also for dynamic models.

(i) design a closed-loop (proportional) controller for the dynamic model

(ii) design a closed-loop (proportional+integral) controller for the dynamics model.

7.3.1 Proportional feedback control of a first-order system

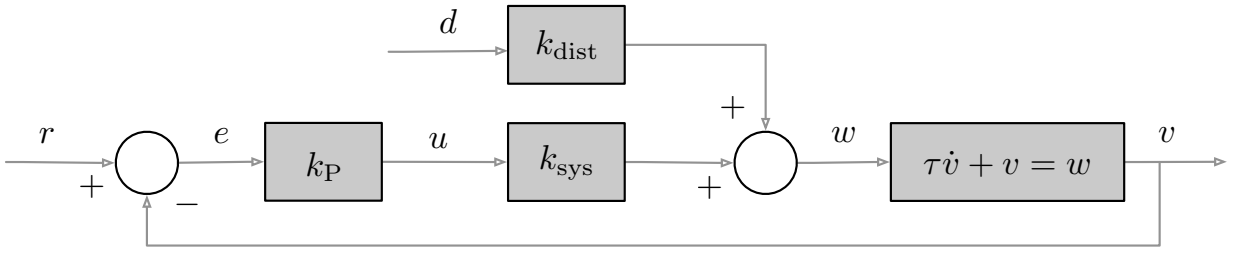


Figure 7.8: Closed-loop P control of the dynamic car velocity model.
Illustration via a block diagram with control block and a feedback loop.
Note that the light blue boxes describes, respectively, the *system* (to be controlled and subject to a disturbance) and the *controller*.

As before, given a reference signal r, we define the *error signal* by $e = r - v$ and design a *proportional controller* (also called *P control*)

$$u = k_{\text{P}} e = k_{\text{P}}(r - v). \tag{7.10}$$

In summary, the closed-loop first-order system (modeling a cruise control system) with the proportional controller is described by

$$\begin{cases} \tau\dot{v} = -v + k_{\text{sys}} u + k_{\text{dist}} d \\ u = k_{\text{P}}(r - v) \end{cases} \tag{7.11}$$

We first simulate this system for various values of the controller gain k_{P}.

In class assignment

Is the closed-loop system always stable?
Is the closed-loop system still first-order? If so, what is the closed-loop time constant? (where does the pole move to?)
Does the velocity converge exactly to the reference value?

Simulation of proportional feedback control for cruise control system: varying k_P

```python
import numpy as np; import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
plt.rcParams.update({"text.usetex": True, "font.family": "serif", ...
    "font.serif": ["Computer Modern Roman"] })

# Constants
ksys = 3                        # system gain, we let kdist=ksys
tau = 5                         # system time constant (slow system)
d = 0                           # disturbance
kp = [0.1, 1, 10, 100]          # control gain (multiple values)

# Define the ODE for the cruise control system
def cruise_control_ode(t, y, K, tau, reference_speed, d):
    speed = y[0]
    control_input = K * (reference_speed - speed)
    acceleration = (-speed + ksys * (control_input + d)) / tau
    dydt = [acceleration]
    return dydt

# Initial conditions: 50 mph.  Time span for simulation
initial_speed = 50;             # initial speed (mph)
reference_speed = 60            # desired speed (mph)
init_cond = [initial_speed]; t_span = (0, 6)

# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8), sharex=True)
ax1.set_title('Cruise control for car dynamics with disturbance: ...
    proportional controller')
colors = ['#752d00', '#a43e00', '#d35000', '#ff6100']

# Solve the ODE and plot the results for each value of K
for i, K in enumerate(kp):
    solution = solve_ivp(cruise_control_ode, t_span, init_cond, ...
        args=(K, tau, reference_speed, d), t_eval=np.arange(0, 6, ...
        0.01), method='LSODA')
    time = solution.t; speed = solution.y[0]; control_input = [K * ...
        (reference_speed - speed[j]) for j in range(len(time))]

    # Plot the speed in the first subplot and control input in the ...
        second subplot
    ax1.plot(time, speed, label=f'$k_{{\\mathrm{{p}}}} = {K}$', ...
        color=colors[i])
    ax1.set_ylabel('Speed (mph)'); ax1.set_xlim(0, 6);  ...
        ax1.set_ylim(35, 65); ax1.grid(True); ax1.legend()
    ax2.plot(time, control_input, label=f'$k_{{\\mathrm{{P}}}} = ...
        {K}$', color=colors[i])
    ax2.set_xlabel('Time (s)');  ax2.set_ylabel('Control Input'); ...
        ax2.set_xlim(0, 6);  ax2.set_ylim(0, 30);  ax2.grid(True); ...
        ax2.legend()

# Save the plot to a PDF file
plt.savefig('cruise-control-proportional.pdf', bbox_inches='tight')
```

Listing 7.1: Python script generating Figure 7.9. Available at cruise-control-proportional.py

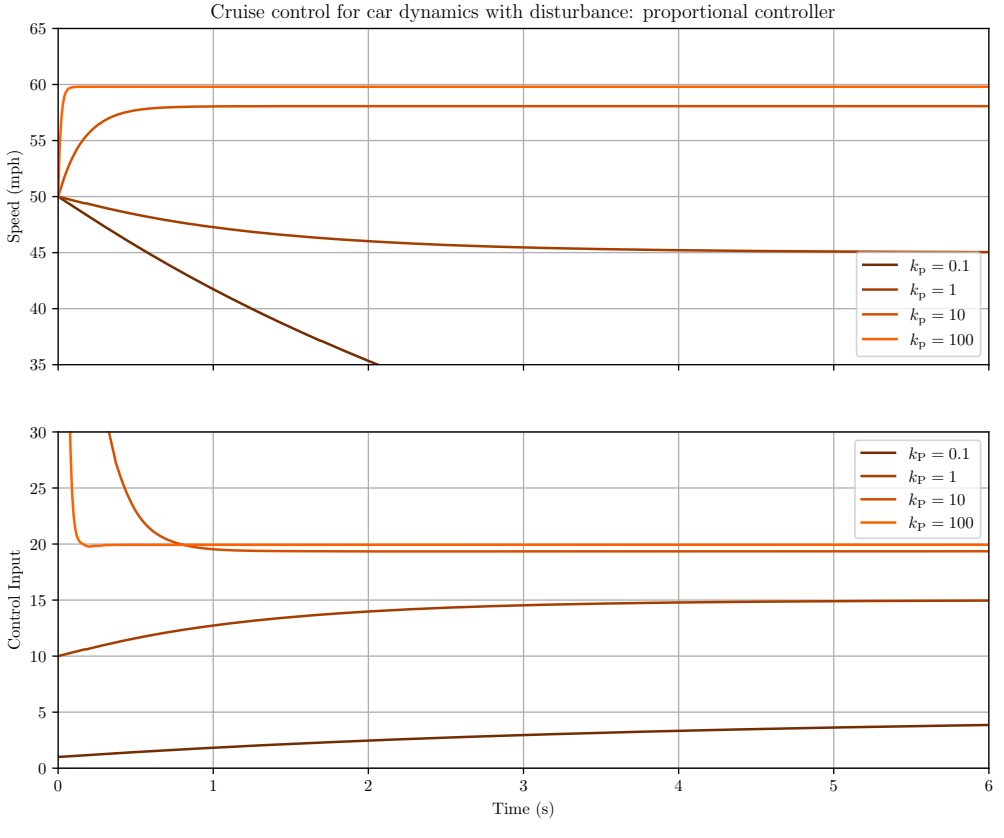


Figure 7.9: Solutions of the cruise control dynamics (7.11): $v(t)$ in the first plot, $u(t)$ in the second plot. The initial velocity is $v(0) = 50$ and the reference velocity is 60.
The closed-loop system is first order; different values of k_P lead to different final values. A larger control gain k_P decreases the time constant and diminishes the steady state error (but no value of k_P achieves perfect regulation with zero steady state error), at the cost of large control signals (see the second plot).
Bottom line: none of these solution is satisfactory (even without disturbance $d = 0$)

Analysis of the closed-loop dynamical system We now analyze the system of equations (7.11), which we report here for convenience:

$$\begin{cases} \tau \dot{v} = -v + k_{\text{sys}} u + k_{\text{dist}} d \\ u = k_{\text{P}}(r - v) \end{cases} \tag{7.12}$$

Plugging the value for the control u into the differential equation we obtain:

$$\tau \dot{v} = -v + k_{\text{sys}}\big(k_{\text{P}}(r - v)\big) + k_{\text{dist}} d = -v + k_{\text{sys}} k_{\text{P}} r - k_{\text{sys}} k_{\text{P}} v + k_{\text{dist}} d \tag{7.13}$$

$$= -(1 + k_{\text{sys}} k_{\text{P}}) v + k_{\text{sys}} k_{\text{P}} r + k_{\text{dist}} d. \tag{7.14}$$

Dividing by $(1 + k_{\text{sys}} k_{\text{P}})$, we write the closed-loop system in canonical form:

$$\frac{\tau}{1 + k_{\text{sys}} k_{\text{P}}} \dot{v} \;=\; -v + \frac{k_{\text{sys}} k_{\text{P}}}{1 + k_{\text{sys}} k_{\text{P}}} r + \frac{k_{\text{dist}}}{1 + k_{\text{sys}} k_{\text{P}}} d. \tag{7.15}$$

In other words, the closed-loop system is again a first-order system of the form

$$\tau_{\text{closed-loop}} \dot{v} = -v + \frac{k_{\text{sys}} k_{\text{P}}}{1 + k_{\text{sys}} k_{\text{P}}} r + \frac{k_{\text{dist}}}{1 + k_{\text{sys}} k_{\text{P}}} d \tag{7.16}$$

with:

- time constant $\tau_{\text{closed-loop}} = \dfrac{\tau}{1 + k_{\text{sys}} k_{\text{P}}}$,
- system gain from reference to output equal to $\dfrac{k_{\text{sys}} k_{\text{P}}}{1 + k_{\text{sys}} k_{\text{P}}}$, and
- system gain from disturbance to output equal to $\dfrac{k_{\text{dist}}}{1 + k_{\text{sys}} k_{\text{P}}}$.

To check that we calculated the dynamic closed-loop system correctly, it is useful to compare the input-output dynamic model with the input-output static model. Recall that equation (7.8) for the static model predicts a steady-state value

$$v_{\text{steady-state}} = \frac{k_{\text{sys}}k_{\text{P}}}{1 + k_{\text{sys}}k_{\text{P}}}r + \frac{k_{\text{dist}}}{1 + k_{\text{sys}}k_{\text{P}}}d \tag{7.17}$$

which is the same as the dynamic model (7.16) when $\dot{v} = 0$. It is also useful to write the *steady-state error*

$$e_{\text{steady-state}} = r - v_{\text{steady-state}} = \left(1 - \frac{k_{\text{sys}}k_{\text{P}}}{1 + k_{\text{sys}}k_{\text{P}}}\right)r - \frac{k_{\text{dist}}}{1 + k_{\text{sys}}k_{\text{P}}}d \quad = \quad \frac{1}{1 + k_{\text{sys}}k_{\text{P}}}r - \frac{k_{\text{dist}}}{1 + k_{\text{sys}}k_{\text{P}}}d \tag{7.18}$$

This analysis confirms that, as the proportional control gain k_{P} grows:

(i) the time constant $\dfrac{\tau}{1 + k_{\text{sys}}k_{\text{P}}}$ decreases,

(ii) the steady-state error $e_{\text{steady-state}}$ due to the reference signal r decreases, and

(iii) the steady-state error $e_{\text{steady-state}}$ due to the disturbance d decreases.

(iv) However, recall that we observed excessively large control signals in the simulation in Figure 7.9 above.

In class assignment

Given an error signal $e(t)$ we have designed a proportional controller.
But there remains a steady state error, since large gains and control actions are undesirable.
What other control action could you take to remove the steady state error?

(BTW: Large contraction actions may not be physically realizable by the actuator (e.g., the engine) and because noise may lead to excessive control chattering.)

7.3.2 From proportional control to proportional+integral control

In order to eliminate the steady-state error, we introduce a second control action based upon the following idea:

> *integrate the error over time and apply a control action proportional to this integral*

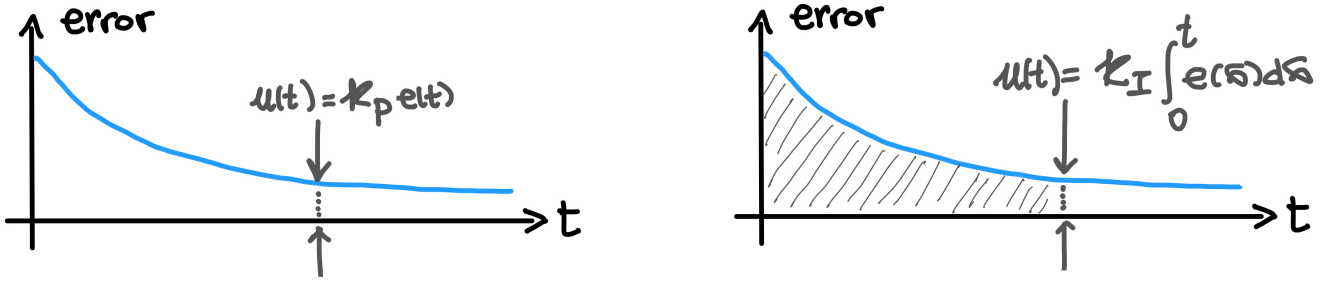


Figure 7.10: Proportional and integral control.

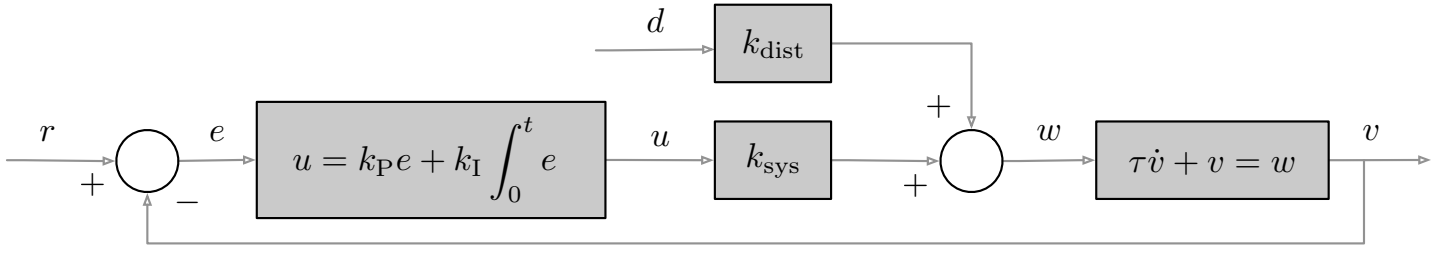


Figure 7.11: Closed-loop PI control of the dynamic car velocity model.

For the first-order system studied so far, we consider a *proportional+integral control* with two positive control gains k_{P} and k_{I} (it is useful to pay attention to the dependency with respect to time):

$$\begin{cases} \tau \dot{v}(t) + v(t) = k_{\mathsf{sys}} u(t) + k_{\mathsf{dist}} d(t) \\ e(t) = r - v(t) \\ u(t) = \underbrace{k_{\mathrm{P}} e(t)}_{\text{proportional action}} + \underbrace{k_{\mathrm{I}} \int_0^t e(\sigma) d\sigma}_{\text{integral action}} \end{cases} \tag{7.19}$$

There are two approaches to understand the behavior of this system of equations. First, here we use a time domain approach. The next chapter takes an easier and more general approach based upon Laplace transforms.

We rewrite the system of equations (7.19) by *differentiating with respect to time* each equation:

$$\begin{cases} \tau\ddot{v}(t) + \dot{v}(t) = k_{\text{sys}}\dot{u}(t) + k_{\text{dist}}\dot{d}(t) \\ e(t) = r - v(t), \qquad \dot{e}(t) = -\dot{v}(t) \\ \dot{u}(t) = \underbrace{k_{\text{P}}\dot{e}(t)}_{\text{proportional action}} + \underbrace{k_{\text{I}}e(t)}_{\text{integral action}} = k_{\text{P}}(-\dot{v}(t)) + k_{\text{I}}(-v(t)) + k_{\text{I}}r \end{cases} \tag{7.20}$$

Note that we used knowledge of the fact that the reference signal r is constant. These are linear equations and we are now ready to the eliminate undesirable variables. In summary we obtain:

$$\tau\ddot{v}(t) + \dot{v}(t) = -\underbrace{k_{\text{sys}}k_{\text{P}}\dot{v}(t)}_{\text{proportional action}} - \underbrace{k_{\text{sys}}k_{\text{I}}v(t)}_{\text{integral action}} + k_{\text{sys}}k_{\text{I}}r + k_{\text{dist}}\dot{d}(t) \tag{7.21}$$

Finally, reorganizing terms we obtain the closed-loop system

$$\tau\ddot{v}(t) + \Big(1 + \underbrace{k_{\text{sys}}k_{\text{P}}}_{\text{proportional action}}\Big)\dot{v}(t) + \underbrace{k_{\text{sys}}k_{\text{I}}}_{\text{integral action}} v(t) = \underbrace{k_{\text{sys}}k_{\text{I}}r}_{\text{effect of reference}} + \underbrace{k_{\text{dist}}\dot{d}(t)}_{\text{effect of disturbance}} \tag{7.22}$$

In class assignment

Classify the system: What order is it? What will it behave like? what is the final value?

(i) In summary, the car velocity system, modeled as a first-order system, subject to a proportional + integral control, reference signal, and disturbance signal obeys the dynamics

$$\underbrace{\tau}_{\bar{m}}\ddot{v}(t) + \underbrace{\left(1 + k_{\text{sys}}k_{\text{P}}\right)}_{\bar{b}}\dot{v}(t) + \underbrace{k_{\text{sys}}k_{\text{I}}}_{\bar{k}}v(t) \;=\; \underbrace{k_{\text{sys}}k_{\text{I}}r + k_{\text{dist}}\dot{d}(t)}_{\bar{f}(t)},$$

so that the system has the same dynamic behavior as a *forced mass-spring-damper system* with fictional mass $\bar{m} = \tau$, damping coefficient $\bar{b} = 1 + k_{\text{sys}}k_{\text{P}}$, spring stiffness $\bar{k} = k_{\text{sys}}k_{\text{I}}$, and force $\bar{f}(t) = k_{\text{sys}}k_{\text{I}}r + k_{\text{dist}}\dot{d}(t)$;

(ii) if that reference r and disturbance d are constant, then the steady-state velocity v_{ss} satisfies $k_{\text{sys}}k_{\text{I}}v_{\text{ss}} = k_{\text{sys}}k_{\text{I}}r$ so that

$$v_{\text{ss}} = r.$$

Proportional+integral control for a first-order system achieves exact reference tracking and exact disturbance rejection (assuming that the reference r and disturbance d are constant).

Note: we still need to decide how to *tune the proportional and integral gains* k_{P} and k_{I}.

Simulation of proportional+integral feedback control for cruise control system: $k_\mathrm{I} = 1$ and varying k_P

```python
import numpy as np; import matplotlib.pyplot as plt;
from scipy.integrate import solve_ivp
plt.rcParams.update({"text.usetex": True, "font.family": "serif", ...
    "font.serif": ["Computer Modern Roman"] })

# Constants
ksys = 3                        # system gain, we let kdist=ksys
tau = 5                         # Plant time constant (slow system)
d = -10                         # Disturbance
kp = [0.1, 1, 10, 100]          # Proportional control gain (multiple values)
ki = 1                          # Integral control gain

# Define the ODE for the cruise control system
def cruise_control_ode(t, y, KP, ki, tau, reference_speed, d):
    speed, integral = y
    error = reference_speed - speed
    control_input = KP * error + ki * integral
    acceleration = (-speed + ksys*(control_input + d)) / tau
    dydt = [acceleration, error]
    return dydt

# Initial conditions and time span
reference_speed = 60          # Desired speed (mph)
initial_conditions = [50, 0] # initial speed and initial error
t_span = (0, 12)

# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8), sharex=True)
ax1.set_title('Car dynamics under proportional-integral control')
colors = ['#752d00', '#a43e00', '#d35000', '#ff6100']

# Solve the ODE and plot the results for each value of K
for i, Kp in enumerate(kp):
    solution = solve_ivp(cruise_control_ode, t_span, initial_conditions,
args=(Kp, ki, tau, reference_speed, d), t_eval=np.arange(0, 12, 0.01),
                         method='LSODA')
    time = solution.t;  speed = solution.y[0]
    control_input = [Kp * (reference_speed - speed[j]) + ki * ...
        solution.y[1][j] for j in range(len(time))]

    # Plot the speed in the first subplot
    ax1.plot(time, speed, label=f'$k_{{\\mathrm{{P}}}} = {Kp}$', ...
        color=colors[i])
    ax1.set_ylabel('Speed (mph)')
    ax1.set_xlim(0, 12); ax1.set_ylim(40, 70); ax1.grid(True); ax1.legend()

    # Plot the control input in the second subplot
    ax2.plot(time, control_input, label=f'$k_{{\\mathrm{{P}}}} = {Kp}$', ...
        color=colors[i]);
    ax2.set_xlabel('Time (s)');     ax2.set_ylabel('Control Input');    ...
        ax2.set_xlim(0, 12);     ax2.set_ylim(0, 50);    ax2.grid(True);   ...
        ax2.legend()

# Save the plot to a PDF file
plt.savefig('cruise-control-proportional-integral.pdf', bbox_inches='tight')
```

Listing 7.2: Python script generating Figure 7.12. Available at cruise-control-proportional-integral.py

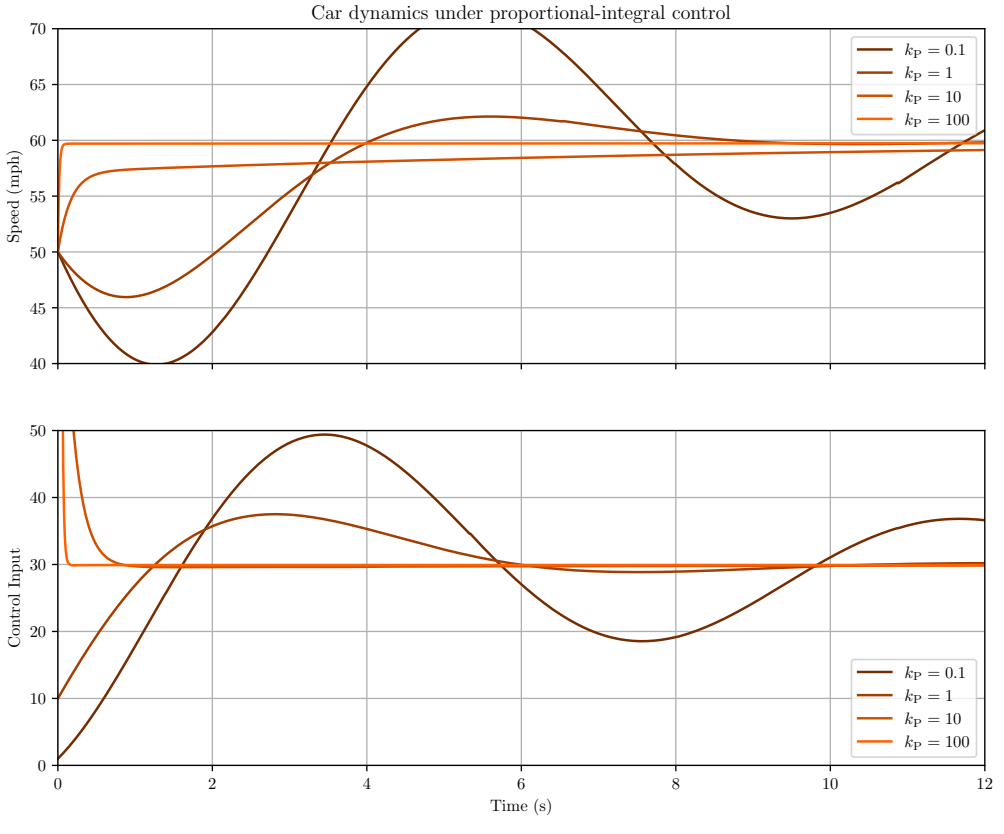


Figure 7.12: Solutions of the cruise-control dynamics with proportional and integral control (7.19): $v(t)$ in the first plot, $u(t)$ in the second plot. The initial velocity is $v(0) = 50$ and the reference velocity is 60. (The response is from non-zero initial conditions and in response to a step input in r and d.)
Multiple values of the proportional control gain k_P and fixed integral gain $k_\mathrm{I} = 1$.

The behavior of the closed-loop system is that of a mass-spring-damper system. The steady state speed is equal to r, hence the disturbance d is rejected. Large k_P leads to overly large control signals. Too small values of k_P lead to excessive overshoot.

Simulation of proportional+integral feedback control for cruise control system: $k_P = 1$ and varying k_I

```python
import numpy as np; import matplotlib.pyplot as plt;
from scipy.integrate import solve_ivp
plt.rcParams.update({"text.usetex": True, "font.family": "serif", ...
    "font.serif": ["Computer Modern Roman"] })

# Constants
ksys = 3                        # system gain, we let kdist=ksys
tau = 5                         # Plant time constant (slow system)
d = -10                         # Disturbance
kp = 1                          # Proportional control gain
ki = [0.1, 1, 10, 100]          # Integral control gain (multiple values)

# Define the ODE for the cruise control system
def cruise_control_ode(t, y, kp, KI, tau, reference_speed, d):
    speed, integral = y
    error = reference_speed - speed
    control_input = kp * error + KI * integral
    acceleration = (-speed + ksys*(control_input + d)) / tau
    dydt = [acceleration, error]
    return dydt

# Initial conditions and time span
reference_speed = 60          # Desired speed (mph)
initial_conditions = [50, 0] # initial speed and initial error
t_span = (0, 12)

# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8), sharex=True)
ax1.set_title('Car dynamics under proportional-integral control')
colors = ['#752d00', '#a43e00', '#d35000', '#ff6100']

# Solve the ODE and plot the results for each value of K
for i, Ki in enumerate(ki):
    solution = solve_ivp(cruise_control_ode, t_span, initial_conditions,
args=(kp, Ki, tau, reference_speed, d), t_eval=np.arange(0, 12, 0.01),
                         method='LSODA')
    time = solution.t;  speed = solution.y[0]
    control_input = [kp * (reference_speed - speed[j]) + Ki * ...
        solution.y[1][j] for j in range(len(time))]

    # Plot the speed in the first subplot
    ax1.plot(time, speed, label=f'$k_{{\\mathrm{{I}}}} = {Ki}$', ...
        color=colors[i])
    ax1.set_ylabel('Speed (mph)')
    ax1.set_xlim(0, 12); ax1.set_ylim(40, 70); ax1.grid(True); ax1.legend()

    # Plot the control input in the second subplot
    ax2.plot(time, control_input, label=f'$k_{{\\mathrm{{I}}}} = {Ki}$', ...
        color=colors[i]);
    ax2.set_xlabel('Time (s)');     ax2.set_ylabel('Control Input');    ...
        ax2.set_xlim(0, 12);     ax2.set_ylim(0, 50);    ax2.grid(True);   ...
        ax2.legend()

# Save the plot to a PDF file
plt.savefig('cruise-control-proportional-integral-2.pdf', bbox_inches='tight')
```

Listing 7.3: Python script generating Figure 7.13. Available at cruise-control-proportional-integral-2.py

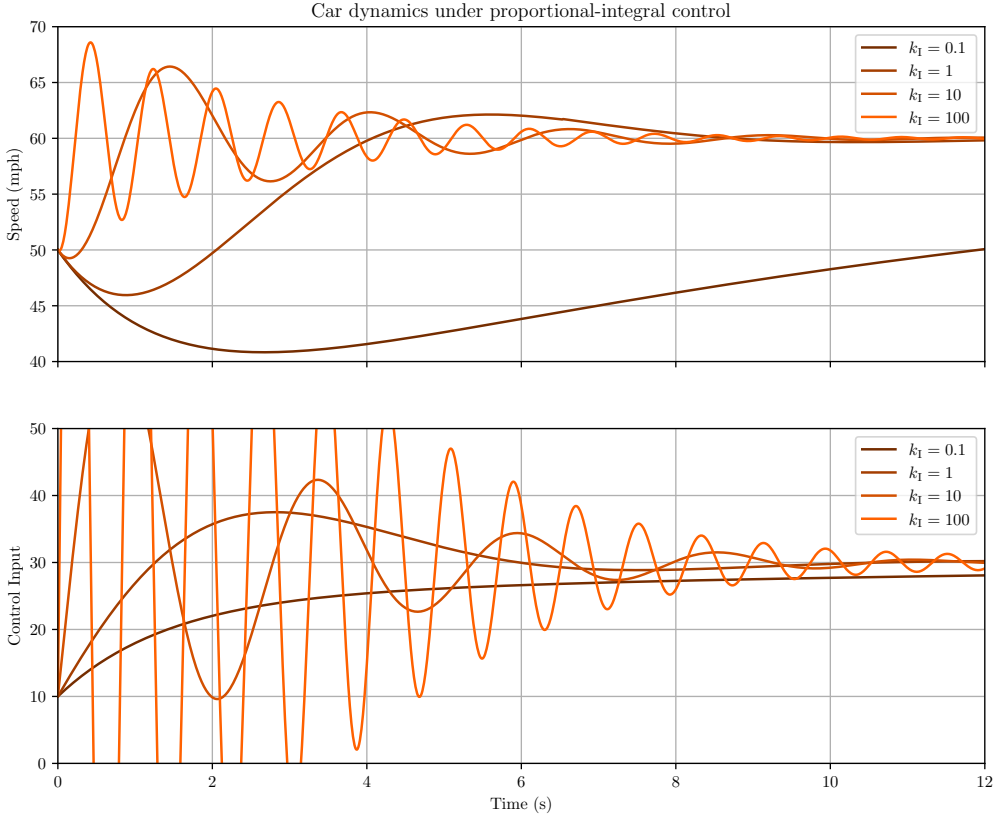


Figure 7.13: Solutions of the cruise-control dynamics with proportional and integral control (7.19): $v(t)$ in the first plot, $u(t)$ in the second plot. The initial velocity is $v(0) = 50$ and the reference velocity is 60. (The response is from non-zero initial conditions and in response to a step input in r and d.)
Multiple values of the integral control gain k_I and fixed value of the proportional $k_P = 1$. The behavior of the closed-loop system is that of a mass-spring-damper system. The steady state speed is equal to r, hence the disturbance d is rejected. However, large k_I leads to excessive overshoot.

7.3.3 Tuning the control gains

Consider again the closed-loop system written in second-order form:

$$\tau\ddot{v}(t) + \big(1 + k_{\text{sys}}k_{\text{P}}\big)\dot{v}(t) + k_{\text{sys}}k_{\text{I}}v(t) \;=\; k_{\text{sys}}k_{\text{I}}r + k_{\text{dist}}\dot{d}(t).$$

Regarding it as a forced mass-spring-damper system with fictional mass $\bar{m} = \tau$, damping coefficient $\bar{b} = 1 + k_{\text{sys}}k_{\text{P}}$, spring stiffness $\bar{k} = k_{\text{sys}}k_{\text{I}}$, and force $\bar{f}(t) = k_{\text{sys}}k_{\text{I}}r + k_{\text{dist}}\dot{d}(t)$, the formulas for the natural frequency and damping ratio are:

$$\omega_{\text{n}} \;=\; \sqrt{\frac{\bar{k}}{\bar{m}}} \;=\; \sqrt{\frac{k_{\text{sys}}k_{\text{I}}}{\tau}} \qquad \text{and} \qquad \zeta \;=\; \frac{\bar{b}}{2\sqrt{\bar{m}\bar{k}}} = \frac{1 + k_{\text{sys}}k_{\text{P}}}{2\sqrt{\tau k_{\text{sys}}k_{\text{I}}}} \tag{7.23}$$

Key design idea: choose the control gains $(k_\mathsf{P}, k_\mathsf{I})$ to obtain a desirable natural frequency and damping ratio in the closed loop. There is only one caveat: ζ cannot be completely arbitrary small, since the original system has some natural damping. In other words, even at $k_\mathsf{P} = 0$, the smallest achievable value of ζ is: $\zeta_\mathsf{min} = \dfrac{1}{2\sqrt{\tau k_\mathsf{sys} k_\mathsf{I}}}$.

Gain selection for proportional+integral control of first-order systems

Consider a given first-order system with time constant τ and system gain k_sys.
Given a desirable closed-loop natural frequency ω_n, we set:

$$k_\mathsf{I} := \frac{\tau \omega_\mathsf{n}^2}{k_\mathsf{sys}}, \tag{7.24}$$

and given a desirable closed-loop damping ratio $\zeta \geq \zeta_\mathsf{min} = \dfrac{1}{2\tau\omega_\mathsf{n}}$, we set

$$k_\mathsf{P} := \frac{2\sqrt{\tau k_\mathsf{sys} k_\mathsf{I}}\zeta - 1}{k_\mathsf{sys}} = \frac{2\tau\omega_\mathsf{n}\zeta - 1}{k_\mathsf{sys}}. \tag{7.25}$$

Simulation of proportional+integral feedback control for cruise control system: $(\omega_n)_{\text{desired}} = 1$ and $\zeta_{\text{desired}} = .6$

```python
import numpy as np; import matplotlib.pyplot as plt; from scipy.integrate ...
    import solve_ivp; plt.rcParams.update({"text.usetex": True, ...
    "font.family": "serif", "font.serif": ["Computer Modern Roman"]})

# Constants
ksys = 3                        # system gain, we let kdist=ksys
tau = 5                         # Plant time constant (slow system)
d = -10                         # Disturbance
# desired natural frequency and damping ratio:
omegan_desired = 1
zeta_desired = .6
# note: zeta_desired > 1/(2 tau omegan_desired)  = 1/(2 x 5) = 1/10.
ki = tau * omegan_desired**2 / ksys
kp = (2 * tau * omegan_desired * zeta_desired - 1) / ksys

# Define the ODE for the cruise control system
def cruise_control_ode(t, y, kp, ki, tau, reference_speed, d):
    speed, integral = y
    error = reference_speed - speed
    control_input = kp * error + ki * integral
    acceleration = (-speed + ksys * (control_input + d)) / tau
    dydt = [acceleration, error]
    return dydt

# Initial conditions and time span
reference_speed = 60                # Desired speed (mph)
initial_speeds = [40, 50, 55, 65]  # Various initial speeds
t_span = (0, 12)

# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8), sharex=True)
ax1.set_title('Car dynamics under proportional-integral control with tuned ...
    gains')
colors = ['#752d00', '#a43e00', '#d35000', '#ff6100']

# Solve the ODE and plot the results for each initial speed
for i, initial_speed in enumerate(initial_speeds):
    initial_conditions = [initial_speed, 0]  # initial speed and initial ...
        integral error
    solution = solve_ivp(cruise_control_ode, t_span, initial_conditions,
                         args=(kp, ki, tau, reference_speed, d),
                         t_eval=np.arange(0, 12, 0.01), method='LSODA')
    time = solution.t; speed = solution.y[0]
    control_input = [kp * (reference_speed - speed[j]) + ki * ...
        solution.y[1][j] for j in range(len(time))]

    # Plot the speed in the first subplot and control signal in the second
    ax1.plot(time, speed, label=f'Initial Speed = {initial_speed} mph', ...
        color=colors[i]); ax1.set_ylabel('Speed (mph)'); ax1.set_xlim(0, ...
        12); ax1.set_ylim(30, 70); ax1.grid(True); ax1.legend()
    ax2.plot(time, control_input, label=f'Initial Speed = {initial_speed} ...
        mph', color=colors[i]); ax2.set_xlabel('Time (s)'); ...
        ax2.set_ylabel('Control Input'); ax2.set_xlim(0, 12); ...
        ax2.set_ylim(0, 50); ax2.grid(True); ax2.legend()
# Save the plot to a PDF file
plt.savefig('cruise-control-proportional-integral-design.pdf', ...
    bbox_inches='tight')
```

Listing 7.4: Python script generating Figure 7.14. Available at cruise-control-proportional-integral-design.py

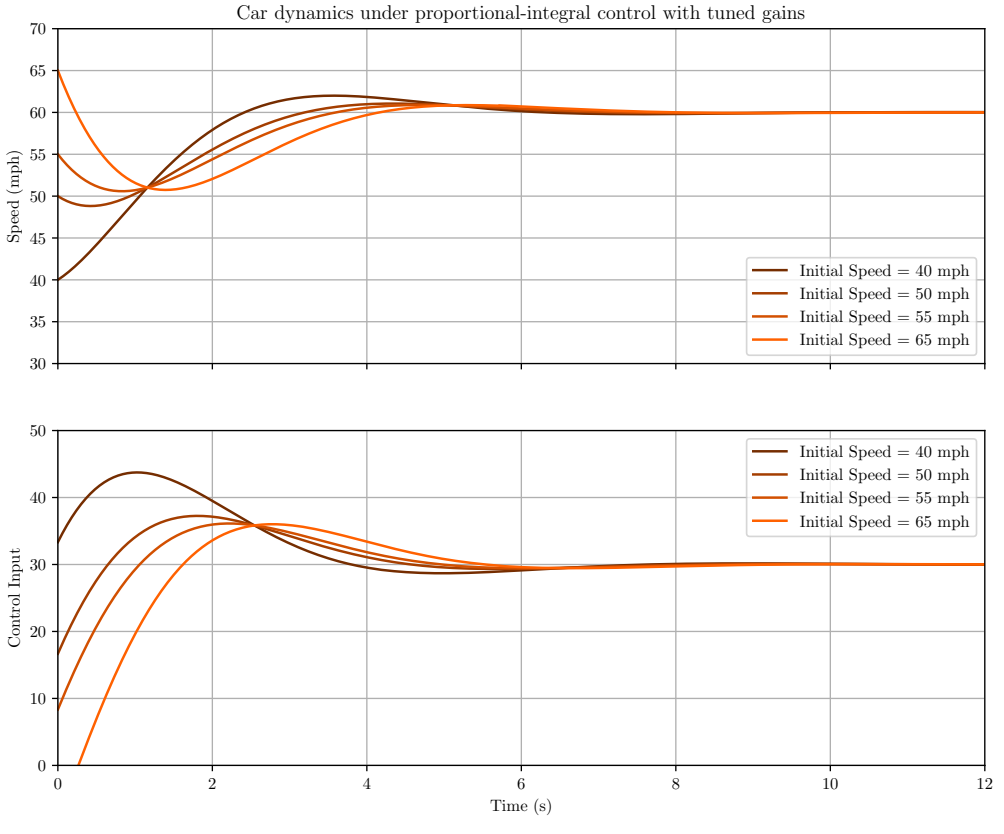


Figure 7.14: Solutions of the cruise-control dynamics with proportional and integral control (7.19): $v(t)$ in the first plot, $u(t)$ in the second plot. The initial velocity is $v(0) = 50$ and the reference velocity is 60. (The response is from multiple non-zero initial conditions and in response to a step input in r and d.)
Proportional and integral gains computed from (7.24)-(7.25) as function of desired values of natural frequency and damping ratio: for this first-order system (with $\tau = 5$ and $k_{\text{sys}} = 3$), $\omega_n = 1$ and $\zeta = .6$ imply $k_P = k_I = 1.667$.
Note: asking for too high ω_n will inevitable lead to very large control signals (since $k_P \propto \omega_n$ and $k_I \propto \omega_n^2$).

7.4 Appendix: Advantages of closed-loop control for static systems

Feedback control is widespread in nature and engineering and a key idea in the area of systems and control. The properties of closed-loop negative feedback control are:

(i) reduces the steady-state error (see Section 7.3.2),

(ii) reduces the effects of disturbances (see Section 7.3.2),

(iii) can stabilize an unstable system (to be seen).

(iv) reduces the sensitivity to parameter variations, and

(v) widens the linear regime.

In this appendix we explain properties (iv) and (v), following the treatment in (Åström and Murray, 2021, Chapter 2).

7.4.1 Feedback enhances robustness to parameter variations

Next, let us consider the situation where system and control parameters are uncertain, that is, the system gain k_{sys} and the control gain k_{P} are not exactly known. This situation arises because of many reasons. For example, (1) parameters are never measured fully exactly, but rather with some limited accuracy, (2) parameters drift with time, and (3) ultimately anyway the true system is not linear and nonlinearities are only approximated as linear.

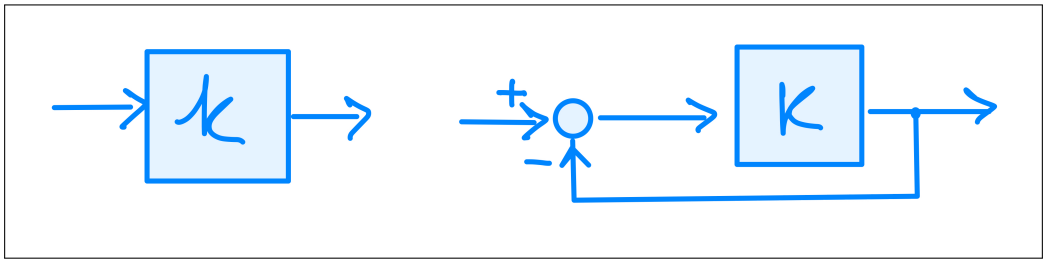


Figure 7.15: The open-loop gain is k for both open-loop and closed-loop block diagrams. We now consider the situation where k is uncertain.

For simplicity, we let k denote the open-loop gain k from r to y and drop the disturbance signal, i.e., we set $w = 0$. We compare:

(i) open-loop control has open-loop gain $k = k_{\text{sys}} k_{\text{P}} \approx 10 \cdot \frac{1}{10} = 1$, and

(ii) closed-loop control has open-loop gain $k = k_{\text{sys}} k_{\text{P}} \approx 100$.

Analysis For open-loop control $y = kr$,

- the partial derivative is $\dfrac{dy}{dk} = r$,
- we eliminate $r = \dfrac{y}{k}$ to obtain $\dfrac{dy}{dk} = \dfrac{y}{k}$, and
- therefore, $\dfrac{dy}{y} = \dfrac{dk}{k}$.

For closed-loop control $y = \dfrac{k}{1+k} r$,

- the partial derivative is $\dfrac{dy}{dk} = \dfrac{r}{1+k} - \dfrac{kr}{(1+k)^2} = \dfrac{r(1+k) - kr}{(1+k)^2} = \dfrac{r}{(1+k)^2}$,
- we eliminate $r = \dfrac{1+k}{k} y$ to obtain $\dfrac{dy}{dk} = \dfrac{1+k}{k} y \dfrac{1}{(1+k)^2} = \dfrac{1}{k(1+k)} y$, and
- therefore, $\dfrac{dy}{y} = \dfrac{1}{k+1} \dfrac{dk}{k}$.

(i) In summary, open-loop control $y = kr$ satisfies

$$\frac{dy}{y} = \frac{dk}{k} \tag{7.26}$$

so that, for $k \approx 1$, a 10% variation of k lead to 10% variation of y.

(ii) Instead, closed-loop control $y = \frac{k}{1+k} r$ satisfies

$$\frac{dy}{y} = \frac{1}{k+1} \frac{dk}{k} \tag{7.27}$$

so that, for $k \approx 100$, a 10% variation of k lead to 0.1% variation of y.

Comparing the two strategies, *robustness to parameter variation* is an advantage of closed-loop control. In other words, the output does depend upon the open-loop gain $k = k_{\text{sys}} k_{\text{P}}$, but it is much less sensitive to variations in k_{sys} than open-loop control.

7.4.2 Second additional advantage: Feedback widens the linearity regime

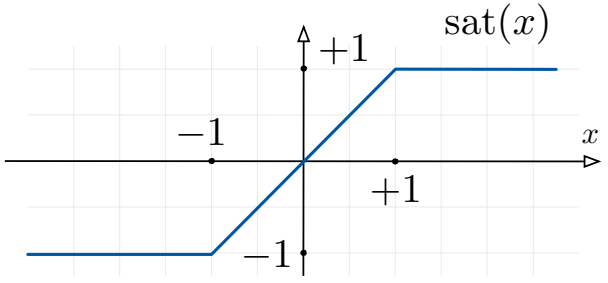


Figure 7.16: Consider the *saturation nonlinearity* $\mathrm{sat}(x) = \begin{cases} +1 & x > 1 \\ x & -1 \le x \le 1. \\ -1 & x < -1 \end{cases}$

We now study the effect of negative feedback on nonlinearities.

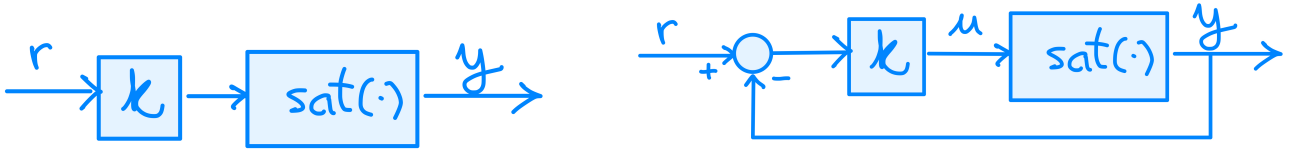


Figure 7.17: The open-loop gain is k for both open-loop and closed-loop block diagrams. We now consider the situation where k is uncertain.

Open-loop system The open-loop map from input r to output y is:

$$y = \mathrm{sat}(kr). \tag{7.28}$$

This nonlinear input-output map is linear for each $|r| < 1/k$, that is, *the open-loop linear gain is k over the range $|r| < \frac{1}{k}$*.

Closed-loop system The closed-loop map from input r to output y can be computed with some algebraic work. For now, we note that

$$y = \mathrm{sat}(u) \qquad \text{and} \qquad u = k(r-y) \qquad \Longrightarrow \qquad y = \mathrm{sat}\big(k(r-y)\big) \tag{7.29}$$

and that, with some work, one solve the implicit equation $y = \mathrm{sat}\big(k(r-y)\big)$ to obtain

$$y = \mathrm{sat}\left(\frac{k}{k+1}r\right) \tag{7.30}$$

This nonlinear input-output function is linear for each $|\frac{k}{k+1}r| < 1$, that is, $|r| < \frac{k+1}{k} = 1 + \frac{1}{k}$.

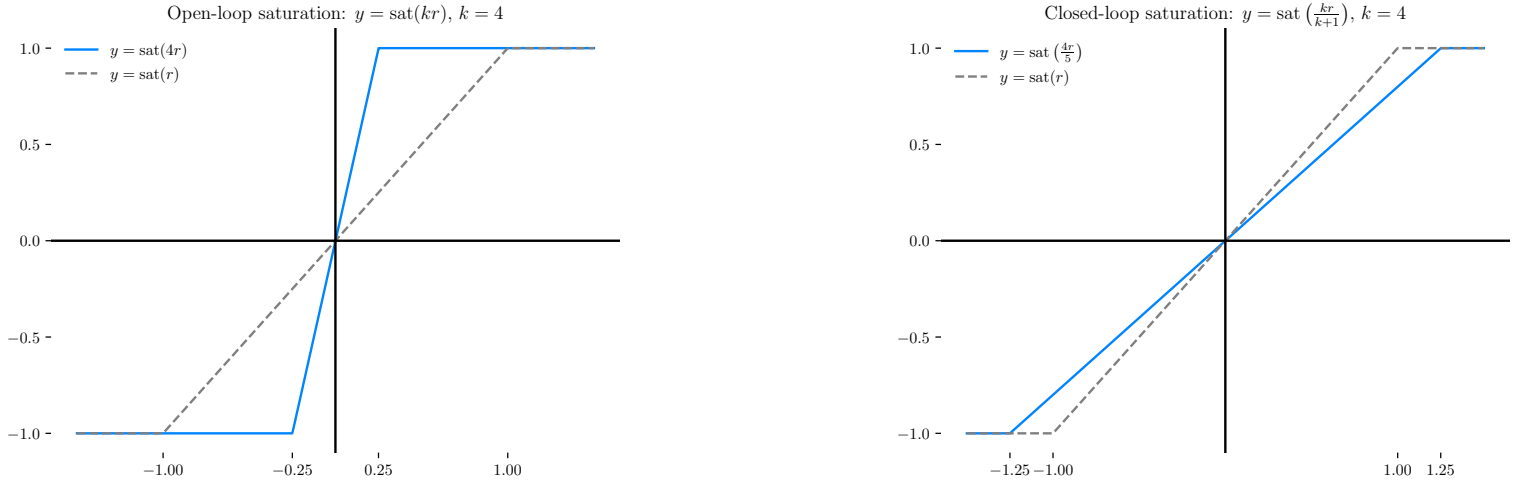


Figure 7.18: We plot the open and closed-loop saturation system for $k = 4$. The open-loop relationship is $y = \mathrm{sat}(kr) = \mathrm{sat}(4r)$ and the closed-loop is $y = \mathrm{sat}(\frac{k}{k+1}r) = \mathrm{sat}(4r/5)$.

In summary, *the closed-loop linear gain is $\frac{k}{k+1}$ over the range $|r| < 1 + \frac{1}{k}$* Comparing the two strategies, for closed-loop control, negative feedback widens the linear range of the system by a factor $k + 1$.

Explanation of the implication in equation (7.29): We show that two basic equations (7.29) imply the input-output map (7.30). Clearly we have

$$y = \mathrm{sat}\big(k(r-y)\big) \tag{7.31}$$

(i) When $|k(r-y)| < 1$, we have $y = k(r-y)$, that is, as in previous calculations, $y = \frac{k}{k+1}r$. We now plug this result back into the condition $|k(r-y)| < 1$ to obtain $|k(r - \frac{k}{k+1}r)| < 1$, that is, $|\frac{k}{k+1}r| < 1$. In summary, we have understood that

$$y = \frac{k}{k+1}r = \mathrm{sat}\Big(\frac{k}{k+1}r\Big) \qquad \text{if } \Big|\frac{k}{k+1}r\Big| < 1 \tag{7.32}$$

(ii) Next, we show that $y = +1$ when $\frac{k}{k+1}r \geq 1$. It is easy to see $y = +1$ and $r \geq \frac{k+1}{k} = 1 + \frac{1}{k}$ are a solution to the equation (7.31). Indeed, we compute:

$$k(r-y) = k(r-1) \geq k\frac{1}{k} = 1. \tag{7.33}$$

Since $k(r-y) \geq 1$, we know $\mathrm{sat}\big(k(r-y)\big) = +1$. We skip the proof that this solution is unique.

(iii) Finally, the case $y = -1$ when $\frac{k}{k+1}r \leq 1$ is proved in a similar manner.

7.5 Historical notes, further reading, and online resources

Award winning video: how control theory explains teeth brushing (2m 42sec).

The first Bode lecture "Respect the unstable," by Gunter Stein in 1989. (The Bode lecture is the most prestigious research lecture in the field of control engineering. The video is of limited quality.) The talk focuses on dangerous systems,inherent limitations of control systems, including the 1986 Chernobyl accident, and the *conservation of dirt*. (1h 11m).

7.6 Exercises

E7.1 **Properties of integral control.** In this exercise we study two properties of integral control.

(i) Consider a control system subject to proportional integral control:

$$u(t) = k_{\mathsf{P}} e(t) + k_{\mathsf{I}} \int_0^t e(\tau) d\tau \tag{E7.1}$$

Assume the closed-loop system has an equilibrium point in which $e(t) = e^*$ and $u(t) = u^*$, where e^* and u^* are constant values. Show that it must be true that either $e^* = 0$ or $k_{\mathsf{I}} = 0$.

(ii) Consider an integrator system $\dot{x}(t) = u(t)$ subject to integral action $u(t) = k_{\mathsf{I}} \int_0^t x(\sigma) d\sigma$. Is the closed-loop system stable, marginally stable, or unstable?

Note: *The two lessons here are that (1) integral control ensures zero steady-state tracking error in reference tracking problems (typically where the reference signal r is constant). (2) However, by itself integral control cannot be trusted to stabilize a system.*

Answer:

(i) At the equilibrium we must have

$$u^* = k_{\mathsf{P}} e^* + k_{\mathsf{I}} \int_0^t e^* d\tau = k_{\mathsf{P}} e^* + k_{\mathsf{I}} t e^* \tag{E7.2}$$

Hence, in order for the equality to hold for all $t \geq 0$, since $k_{\mathsf{I}} > 0$, it must be that $e^* = 0$.

(ii) The closed-loop system is $\dot{x}(t) = -k_{\mathsf{I}} \int_0^t x(\sigma) d\sigma$. Taking derivatives with respect to time of both left and right hand side we obtain:

$$\ddot{x}(t) = -k_{\mathsf{I}} x(t) \tag{E7.3}$$

This is an harmonic oscillator. The closed-loop system oscillates with natural frequency $\sqrt{k_{\mathsf{I}}}$ and so it is only marginally stable.

■

E7.2 **Integral control of first-order systems.** Consider a first-order system with time constant τ and system gain k_{sys} subject to integral control $u(t) = k_{\text{I}} \int_0^t x(\sigma)d\sigma$ (zero proportional control). Given a positive number α, define the integral control gain to be $k_{\text{I}} = \alpha \frac{1}{k_{\text{sys}}\tau}$. For the resulting closed-loop system, compute:

(i) the natural frequency ω_{n} and damping ratio ζ,
(ii) can you choose both ω_{n} and ζ arbitrarily?
(iii) for what values of α is the system underdamped, critically damped, and overdamped?
(iv) the damped natural frequency ω_{d} and the two complex conjugate poles, when the system is underdamped, and
(v) the natural frequency ω_{n}, the damping ratio ζ, and the poles when $\alpha = 1/2$ and $\alpha = 2$.

Note: *Compare the damping ratio achieved at $\alpha = 1/2$ and 2 with the range $0.4 - 0.8$ of damping ratio recommended in Chapter 5.*

Answer:

(i) From equation (7.23), we compute

$$\omega_{\text{n}} = \sqrt{\frac{k_{\text{sys}}k_{\text{I}}}{\tau}}\Big|_{k_{\text{I}}=\alpha\frac{1}{k_{\text{sys}}\tau}} = \frac{\sqrt{\alpha}}{\tau} \tag{E7.4}$$

$$\zeta = \frac{1 + k_{\text{sys}}k_{\text{P}}}{2\sqrt{\tau k_{\text{sys}}k_{\text{I}}}}\Big|_{k_{\text{I}}=\alpha\frac{1}{k_{\text{sys}}\tau},\, k_{\text{P}}=0} = \frac{1}{2\sqrt{\alpha}} \tag{E7.5}$$

(ii) no, there is only one degree of freedom α,
(iii) we note

$$\text{underdamped } (\zeta < 1)\text{: } \alpha > \frac{1}{4}, \qquad \text{critically damped } (\zeta = 1)\text{: } \alpha = \frac{1}{4}, \qquad \text{overdamped } (\zeta > 1)\text{: } \alpha < \frac{1}{4} \tag{E7.6}$$

(iv) When the system is underdamped ($\alpha > 1/4$), we compute poles are at:

$$\omega_{\text{d}} = \zeta\omega_{\text{n}} = \frac{1}{2\tau} \qquad \text{poles} = -\omega_{\text{n}}\left(\zeta + \text{i}\sqrt{1-\zeta^2}\right) = -\frac{\sqrt{\alpha}}{\tau}\left(\frac{1}{2\sqrt{\alpha}} \pm \text{i}\sqrt{1 - \frac{1}{4\alpha}}\right) = -\frac{1}{\tau}\left(\frac{1}{2} \pm \text{i}\sqrt{\alpha - \frac{1}{4}}\right), \tag{E7.7}$$

(v) When $\alpha = 1/2$, we compute

$$\omega_{\text{n}} = \frac{1}{\sqrt{2}\tau} \approx \frac{0.71}{\tau}, \quad \zeta = \frac{1}{\sqrt{2}} \approx 0.71, \qquad \text{and the poles} = -\frac{1}{\tau}\left(\frac{1}{2} \pm \text{i}\frac{1}{2}\right) = \frac{1}{\tau}(-0.5 \pm 0.5) \tag{E7.8}$$

When $\alpha = 2$, we compute

$$\omega_{\text{n}} = \frac{\sqrt{2}}{\tau} \approx \frac{1.41}{\tau}, \quad \zeta = \frac{1}{2\sqrt{2}} \approx 0.35, \qquad \text{and the poles} = -\frac{1}{\tau}\left(\frac{1}{2} \pm \text{i}\frac{\sqrt{7}}{2}\right) \approx \frac{1}{\tau}(-0.5 \pm \text{i}1.32) \tag{E7.9}$$

■

E7.3 **Traffic lights: open loop versus closed loop systems (Edited from (DiStefano et al., 1997)).** Consider the operation of a traffic light that regulates traffic at an intersection between North-South and West-East roads.

(i) What is the control action of the traffic light, that is, how does a traffic light regulate traffic?

(ii) Are preset timing mechanisms open or closed-loop strategies?

(iii) How would you control traffic in a more efficient manner, than preset timing strategies?

(iv) What would you require to enable your strategy for more efficient traffic?

(v) Draw a block diagram for a closed-loop strategy for traffic control.

Hint: Measure traffic on both roads, compare it, and regulate accordingly.

Answer:

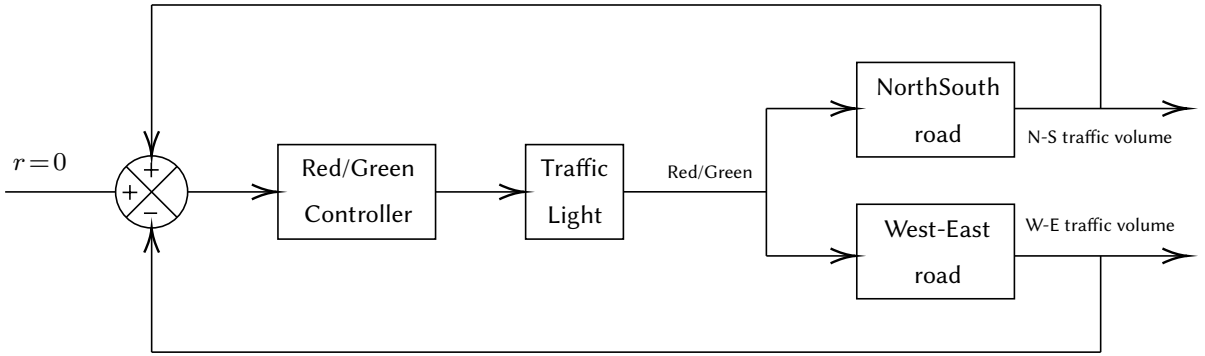
(i) The control action is the timing of the green/red light intervals. (When red is shown to North-South traffic, green is shown to West-East traffic, naturally.)

(ii) Preset timing are open-loop. The timing, including duration, of the traffic light is independent of the amount of traffic on both directions.

(iii) A more efficient design would enable the direction containing greater traffic volume to have longer green durations, than the direction containing lower traffic volume.

(iv) An ideal traffic light controller would (1) measure the volume of traffic on both directions, (2) compare the traffic, and (3) regulate the timing (e.g., duration) of the green/red lights to increase the overall traffic volume.

(v) Here is a simple diagram.



■

Bibliography

K. J. Åström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2 edition, 2021. URL http://www.cds.caltech.edu/~murray/books/AM08/pdf/fbs-public_24Jul2020.pdf.

J. J. DiStefano, , A. R. Stubberu, and I. J. Williams. *Schaum's Outline of Feedback and Control Systems*. McGraw-Hill, 2 edition, 1997.