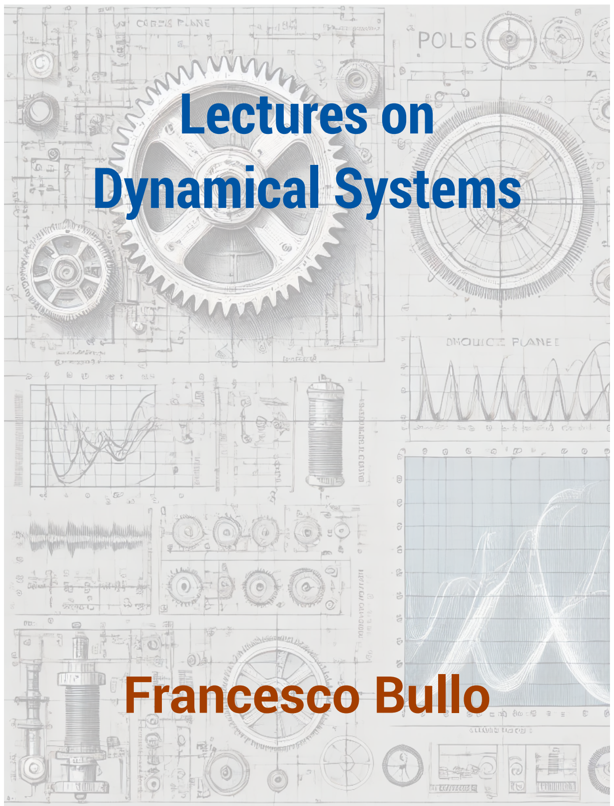
UC Santa Barbara, Department of Mechanical Engineering
ME103 Dynamical Systems. Chapter Slides.

Francesco Bullo
http://motion.me.ucsb.edu/ME103-Fall2025/syllabus.html



Contents

Chapter 7

Control Systems: Basic Definitions and Concepts

In this chapter, we examine the application of control theory to dynamical systems.

We begin by exploring the dynamics of a car on an incline, where the system's velocity is influenced by linear drag, engine propulsion, and gravitational force. Here, the velocity represents the *system state*, while the throttle angle and road inclination act as the *control input* and *disturbance*, respectively. The primary control objectives are *reference tracking* and *disturbance rejection*, which are visually represented through block diagrams. Feedback control, particularly *negative feedback*, is introduced as a crucial mechanism that enables the system to adjust its actions based on the difference between actual and desired outcomes, underscoring its significance in control engineering.

We then design closed-loop control strategies for the car on an incline, emphasizing *proportional (P)* and *proportional-integral (PI)* controllers. The proportional controller minimizes the *error signal* by adjusting the control input, transforming the system into a first-order model. To eliminate steady-state error, the PI controller integrates the error over time, achieving exact reference tracking and disturbance rejection for constant signals. This transforms the system into a second-order mass-spring-damper model, with strategies for tuning control gains to optimize performance.

Additionally, the chapter discusses the advantages of closed-loop control for static systems, highlighting how *negative feedback* enhances robustness to parameter variations and widens the linearity regime. This is contrasted with *open-loop control*, which requires precise system knowledge and is less effective against disturbances.

7.1 Basic control problems and block diagrams

7.1.1 Car dynamics on an incline and the cruise control problem

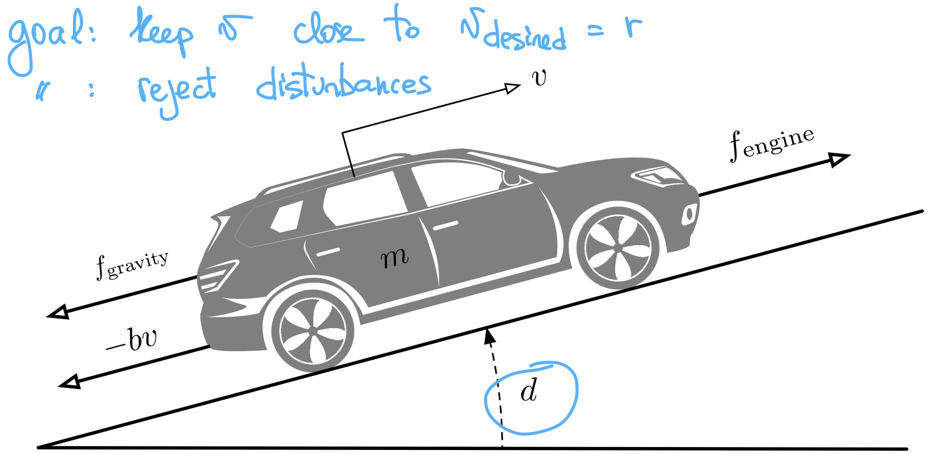


Figure 7.1: A car moving on an inclined road, subject to three forces: a linear drag force $-bv$, a propulsion force f_{engine}, and a gravitational force induced by the road inclination d.

We begin by describing the dynamics of the car's velocity, which form a first-order system:

$$m\dot{v}(t) = -bv(t) + f_{\text{engine}}(t) + f_{\text{gravity}}(t).$$

The engine force f_{engine} is assumed to be regulated by the throttle angle u, and we model it as proportional to u. Similarly, the gravitational force f_{gravity} is modeled as proportional to the road inclination d. Thus,

$$f_{\text{engine}}(t) = f_u u(t), \qquad f_{\text{gravity}}(t) = f_d d(t),$$

where f_u and f_d are positive proportionality constants.

In summary, the system can be described in terms of three variables:

- $v(t)$: the car's velocity, which is the *system state*,
- $u(t)$: the throttle angle, which is the *control input*, and
- $d(t)$: the road inclination, which acts as a *disturbance*.

Substituting the expressions for engine and gravitational forces into the dynamics, and dividing through by b, we obtain a first-order system with both control and disturbance inputs:

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

We can rewrite the car dynamics in canonical form as a *first-order system with control and disturbance*:

$$\tau\dot{v}(t) + v(t) = k_{\text{sys}}u(t) + k_{\text{dist}}d(t), \tag{7.1}$$

where

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

Control objectives: We want to design a control input u such that

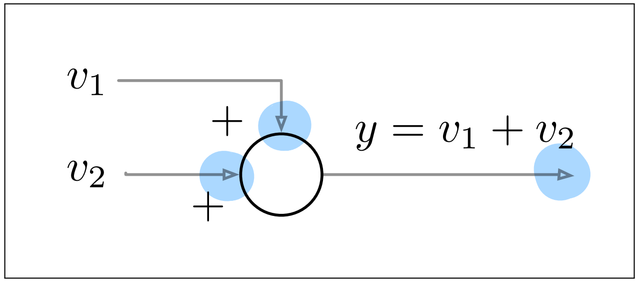
(i) the car maintains a desired reference velocity r (*reference tracking*), and

(ii) the car's speed is unaffected by the disturbance d (*disturbance rejection*).

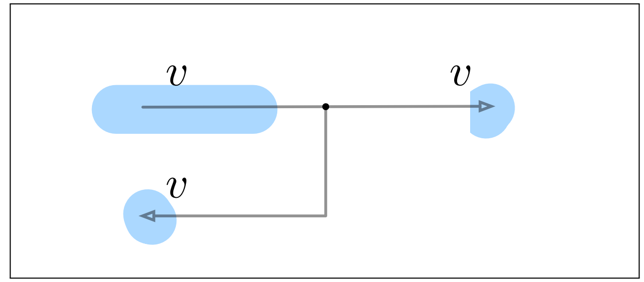
r = reference velocity (desired)
" value for the state

7.1.2 Block diagrams in the time domain

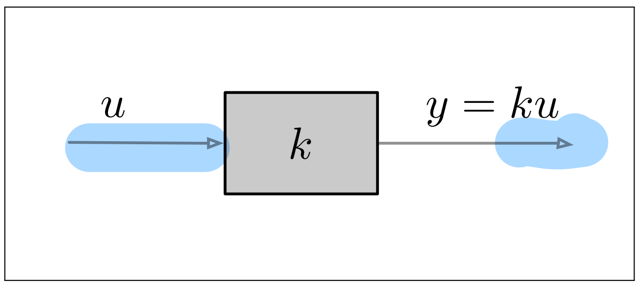
We wish to represent visually dynamical systems and control systems with multiple inputs and interconnections. A *block diagram (in the time domain)* consists of the interconnection of four basic types of elements.



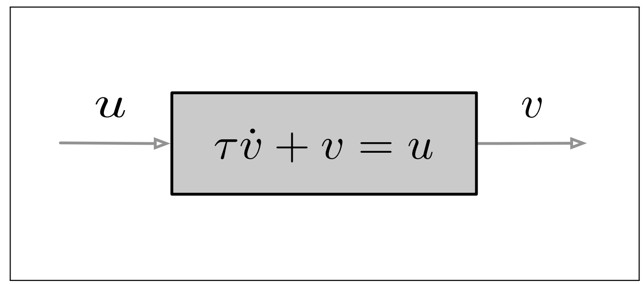
(a) Directed lines representing unidirectional signal flow and a *summing point*. Note that the summing point may includes positive of negative signs to indicate how to compute the algebraic sum.



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



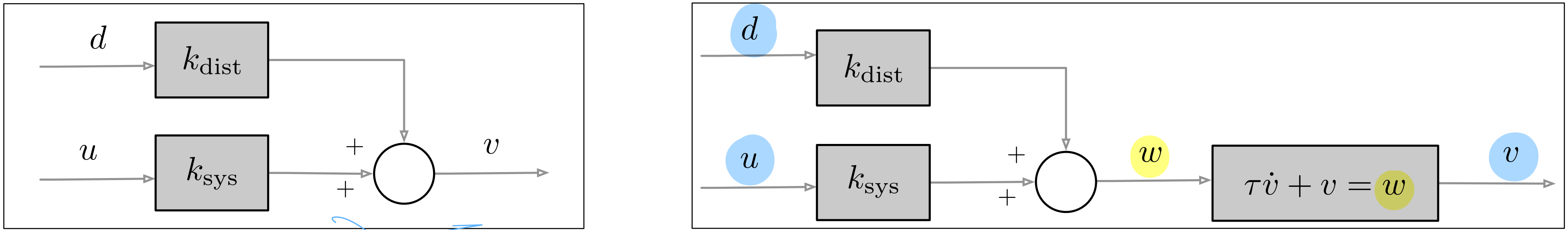
(c) A block describing a *static input/output relationship*, meaning just a product.



(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.

Using block diagrams, it is possible to visualize static and dynamic models.



(a) A static model: $v = d + u$. (b) Dynamic model in equation (7.1)

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

$v = k_{dist}\, d + k_{sys}\, u$

v satisfies:

$\tau \dot{v} + v = k_{dist}\, d + k_{sys}\, u$

7.1.3 Closed-loop cruise control via feedback

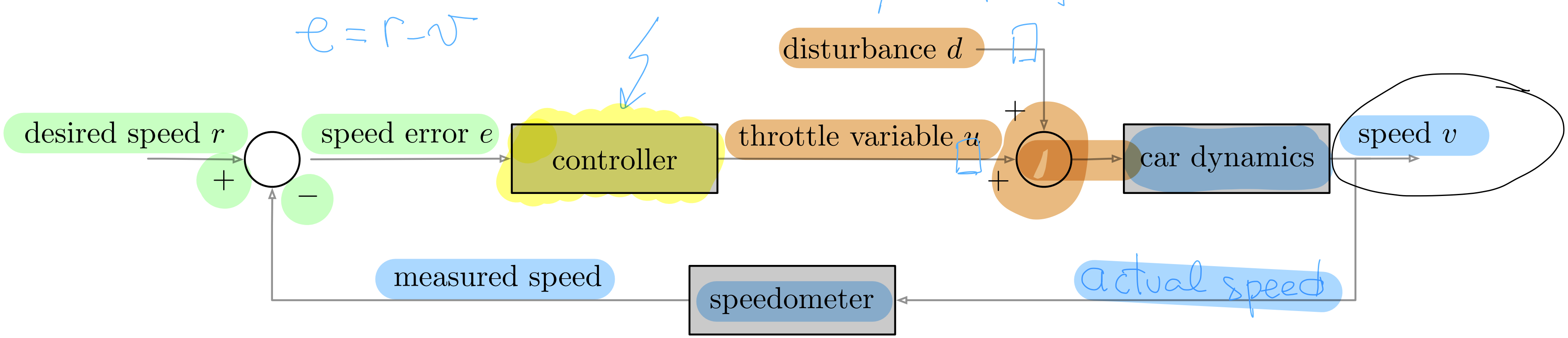


Figure 7.4: Block diagram of a closed-loop feedback control architecture for cruise control

We introduce the key concept of feedback control and closed-loop systems in two steps:

(i) compare the actual outcome with the desired outcome, and

(ii) adjust the system's actions based on the difference.

In practice, this comparison involves feeding the measured output back into the controller with a negative sign — this process is known as *negative feedback*.

Feedback control is a fundamental principle found throughout both nature and engineering. Despite its apparent simplicity, negative feedback is a remarkably powerful idea and lies at the very core of control engineering.

7.2 Proportional control

In this section we design a closed-loop proportional controller.

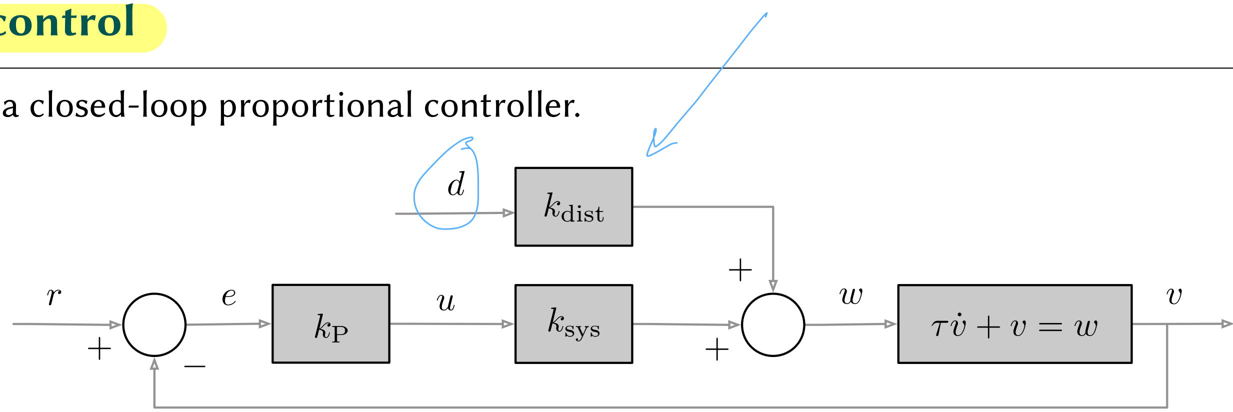


Figure 7.5: Closed-loop P control of the dynamic car velocity model.
Illustration via a block diagram with control block and a feedback loop.
Note that the light blue boxes describes, respectively, the *system* (to be controlled and subject to a disturbance) and the *controller*.

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*)

$$u = k_{\text{P}}e = k_{\text{P}}(r - v). \tag{7.2}$$

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

$$\begin{cases} \tau\dot{v} = -v + k_{\text{sys}}u + k_{\text{dist}}d \\ u = k_{\text{P}}(r - v) \end{cases} \tag{7.3}$$

Next, we simulate the system (7.3) 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?

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

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

```
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=(8, 6.4), 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 $v(t)$'); 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 $t$');  ax2.set_ylabel('control input ...
        $u(t)$'); 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 7.6. Available at cruise-control-proportional.py

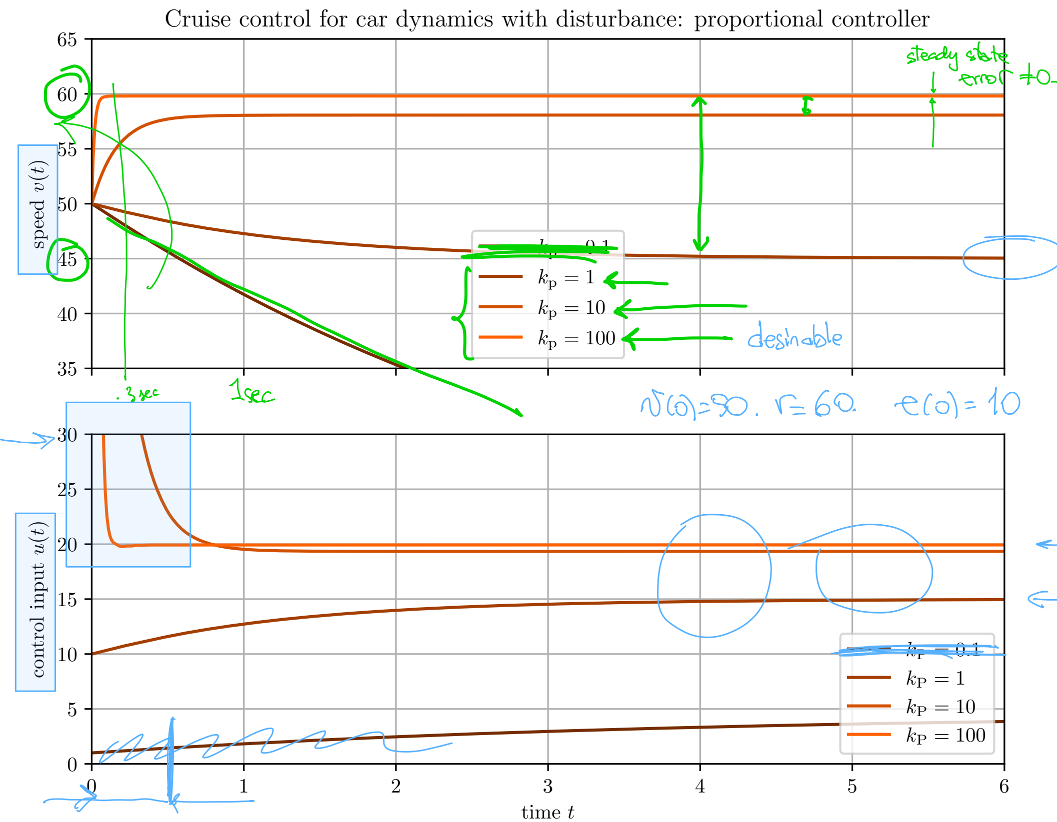


Figure 7.6: Solutions of the cruise control dynamics (7.3): $v(t)$ in the first plot, $u(t)$ in the second plot. The initial velocity is $v(0) = 50$ and the reference velocity is 60. The closed-loop system is first order; different values of k_p lead to different final values. A larger control gain k_p 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 $d = 0$)

We now analyze the closed-loop system arising from applying proportional control to a first-order system. Some calculations (reported in Exercise E7.2) lead to the following result.

The closed-loop system given by the system of equations (7.3) is again a first-order system of the form

$$\tau_{\text{closed-loop}}\dot{v} = -v + \frac{k_{\text{sys}}k_{\text{P}}}{1+k_{\text{sys}}k_{\text{P}}}r + \frac{k_{\text{dist}}}{1+k_{\text{sys}}k_{\text{P}}}d \tag{7.4}$$

with:

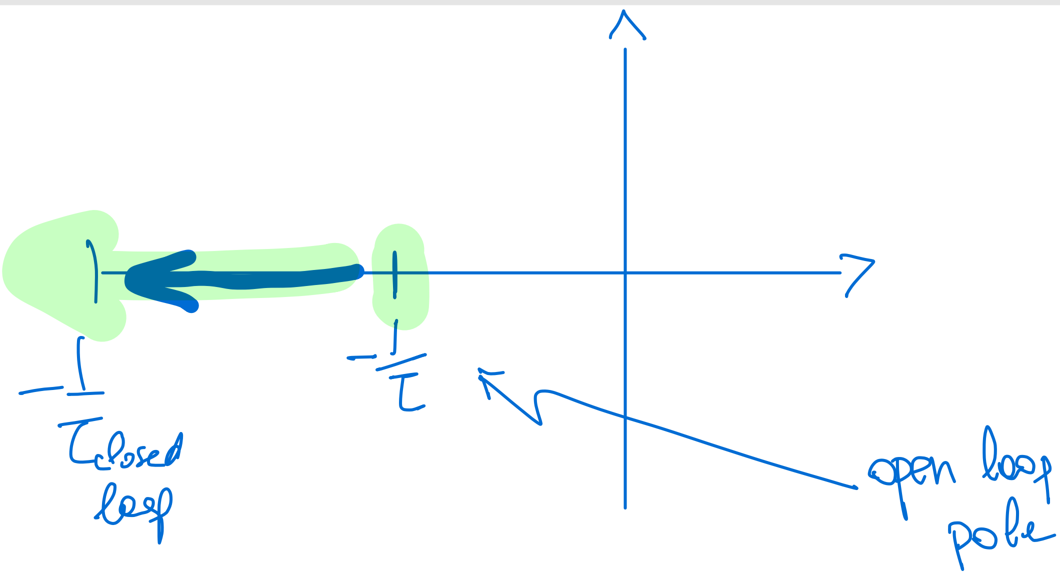
- closed-loop time constant $\tau_{\text{closed-loop}} = \frac{\tau}{1+k_{\text{sys}}k_{\text{P}}}$, corresponding to a closed-loop pole $s_{\text{cl}} = -\frac{1}{\tau_{\text{closed-loop}}} = -\frac{1+k_{\text{sys}}k_{\text{P}}}{\tau}$,
- closed-loop system gain from reference to output equal to $\frac{k_{\text{sys}}k_{\text{P}}}{1+k_{\text{sys}}k_{\text{P}}}$,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

closed loop system is FASTER than

$\tau_{\text{closeloop}} < \tau_{\text{open loop}}$

$\frac{1}{\tau_{\text{closeloop}}} > \frac{1}{\tau}$

$-\frac{1}{\tau_{\text{closeloop}}} < -\frac{1}{\tau}$



Assuming that the reference r and disturbance d are constant signals, it is useful to compute the *steady-state value* of the dynamical system (7.26), i.e., the value when $\dot{v} = 0$ after the transient. Plugging in $\dot{v} = 0$, we obtain:

$$v_{\text{steady-state}} = \underbrace{\frac{k_{\text{sys}} k_{\text{P}}}{1 + k_{\text{sys}} k_{\text{P}}}}_{\text{reference gain}} r + \underbrace{\frac{k_{\text{dist}}}{1 + k_{\text{sys}} k_{\text{P}}}}_{\text{disturbance gain}} d. \tag{7.5}$$

It is also useful to write the *steady-state error*, i.e., the difference between reference speed and steady state speed:

$$e_{\text{steady-state}} = r - v_{\text{steady-state}} = \left(1 - \frac{k_{\text{sys}} k_{\text{P}}}{1 + k_{\text{sys}} k_{\text{P}}}\right) r - \frac{k_{\text{dist}}}{1 + k_{\text{sys}} k_{\text{P}}} d = \frac{1}{1 + k_{\text{sys}} k_{\text{P}}} r - \frac{k_{\text{dist}}}{1 + k_{\text{sys}} k_{\text{P}}} d \tag{7.6}$$

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

(i) the time constant $\tau_{\text{closed-loop}} = \dfrac{\tau}{1 + k_{\text{sys}} k_{\text{P}}}$ 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.

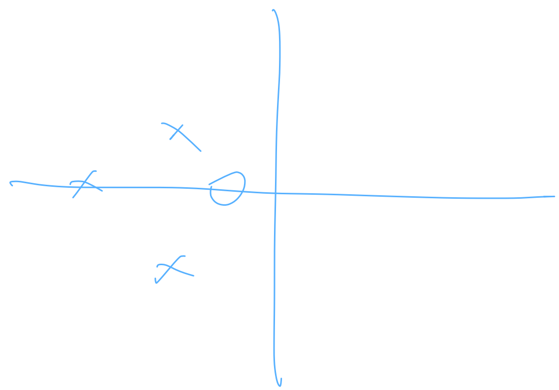
Recall that large control gains k_{P} lead to large control signals in the simulation in Figure 7.6 which may not be physically realizable by the actuator (e.g., the engine). Additionally, when the control gain is large, noise may lead to excessive control chattering.

$$\frac{k_{sys} k_p}{1 + k_{sys} k_p} < 1$$

reference tracking = FAILURE

DISTURBANCE REJECTION = FAILURE

$$\frac{k_{dist}}{1 + k_{sys} k_p} \neq 0$$



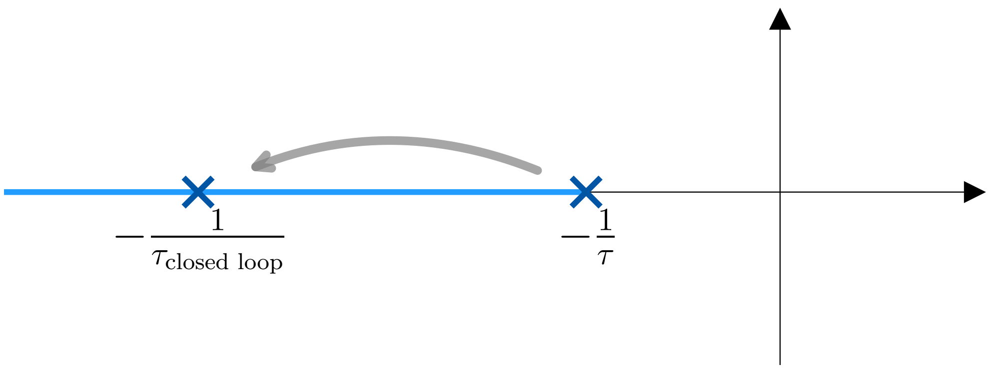


Figure 7.7: Proportional feedback control of a first-order system: the open-loop pole $-1/\tau$ moves to $-1/\tau_{\text{closed-loop}}$, i.e., moves to the left. Recall $\tau_{\text{closed-loop}} = \frac{\tau}{1+k_{\text{sys}}k_{\text{P}}}$. Hence, the closed-loop system is always stable and it is an increasingly faster system as k_{P} increases.

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?

7.3 Proportional integral control

(PI control)

In this section we design a better controller to eliminate the steady-state error. Specifically, 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*

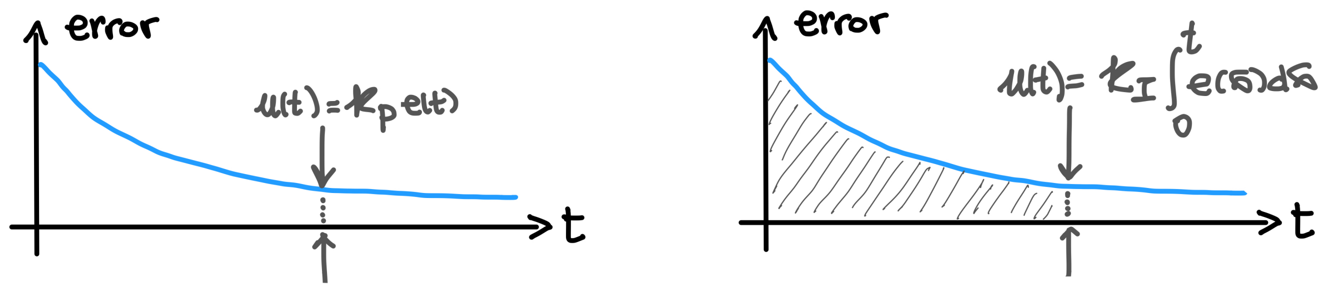


Figure 7.8: Proportional and integral control.

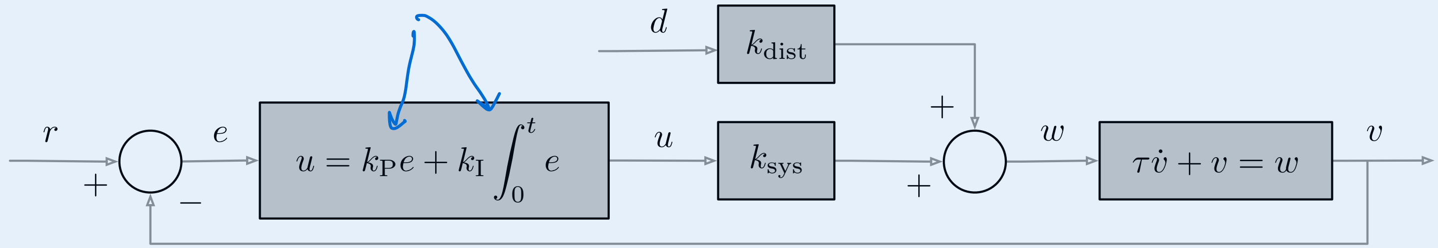


Figure 7.9: Closed-loop PI control of the dynamic car velocity model.

$k_P, k_I = ?$

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 of the variables with respect to time):

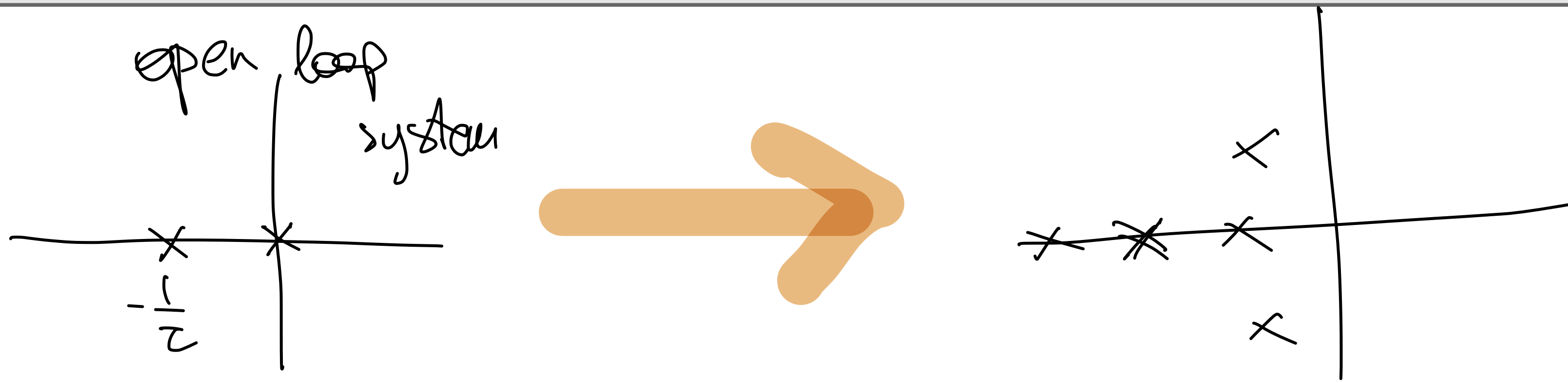
$$\begin{cases} \tau\dot{v}(t) + v(t) = k_{\text{sys}}u(t) + k_{\text{dist}}d(t) \\ e(t) = r - v(t) \\ u(t) = \underbrace{k_P e(t)}_{\text{proportional action}} + \underbrace{k_I \int_0^t e(\sigma)d\sigma}_{\text{integral action}} \end{cases} \tag{7.7}$$

After some calculations (reported in Exercise E7.3), the closed-loop system arising from applying proportional and integral control to a first-order system is as follows

$$\tau\ddot{v}(t) + (1 + \underbrace{k_{\text{sys}}k_P}_{\text{proportional action}})\dot{v}(t) + \underbrace{k_{\text{sys}}k_I}_{\text{integral action}} v(t) = \underbrace{k_{\text{sys}}k_I r}_{\text{effect of reference}} + \underbrace{k_{\text{dist}}\dot{d}(t)}_{\text{effect of disturbance}} \tag{7.8}$$

In class assignment

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



Lessons

(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

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

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

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

$\dot{r} = 0$ $\quad$ $\dot{d} = 0$

$$v_{\mathsf{ss}} \;=\; r.$$

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}.

PI control on 1st order system
achieves reference tracking
disturbance rejection

$\dot{d} = 0$
d is arbitrary but constant

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=(8,6.4), 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 $v(t)$')
    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 $t$');    ax2.set_ylabel('control input ...
        $u(t)$');    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 7.10. Available at cruise-control-proportional-integral.py

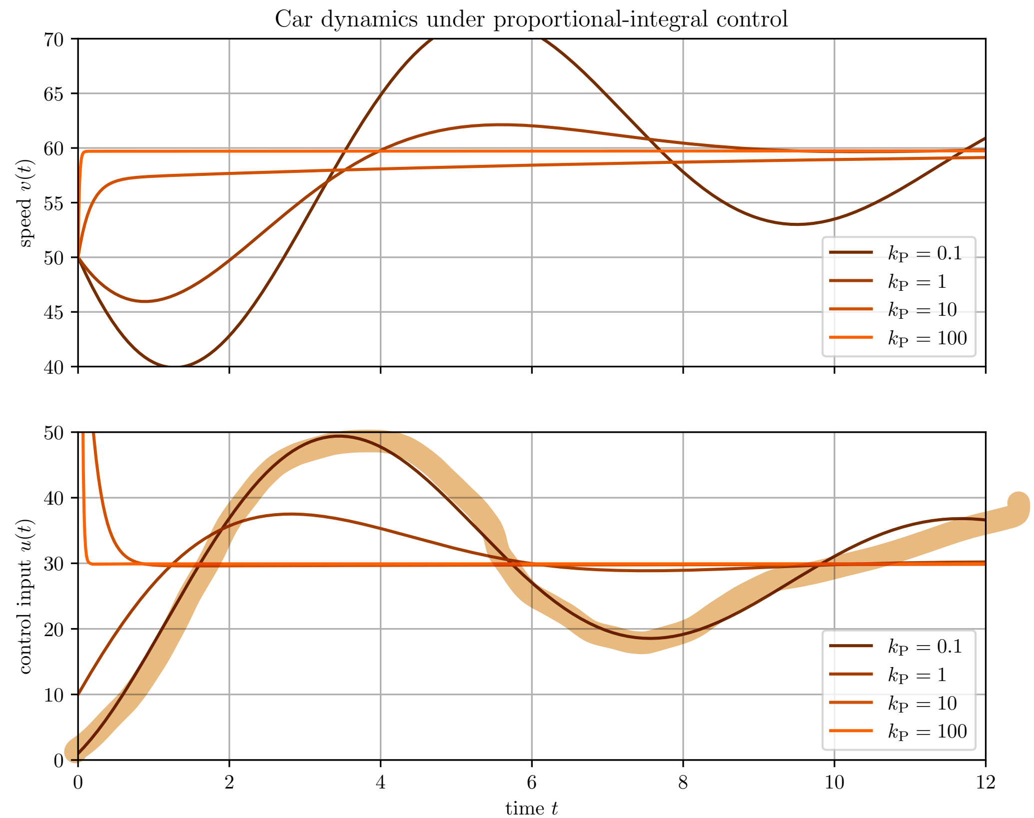


Figure 7.10: Solutions of the cruise-control dynamics with proportional and integral control (7.7): $v(t)$ in the first plot, $u(t)$ in the second plot. The initial velocity is $v(0) = 50$ and the reference velocity is 60. (The response is from non-zero initial conditions and in response to a step input 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: $k_\mathrm{P} = 1$ 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 $t$');     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 7.11. Available at cruise-control-proportional-integral-2.py

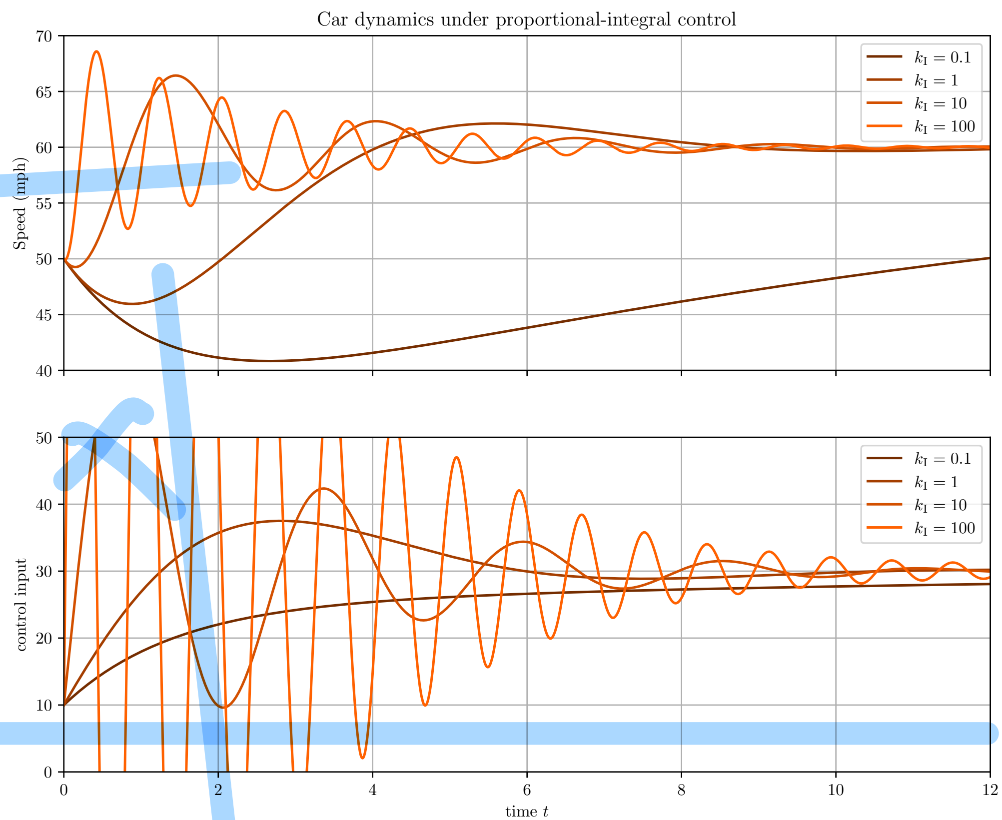


Figure 7.11: Solutions of the cruise-control dynamics with proportional and integral control (7.7): $v(t)$ in the first plot, $u(t)$ in the second plot. The initial velocity is $v(0) = 50$ and the reference velocity is 60. (The response is from non-zero initial conditions and in response to a step input in r and d.)
Multiple values of the integral control gain k_I and fixed value of the proportional $k_\mathrm{P} = 1$. 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.

7.4 Block diagrams in the Laplace domain

We report here Figure 7.9 and the system of equations (7.7). Note that all quantities and transformations are for signals in the time domain.

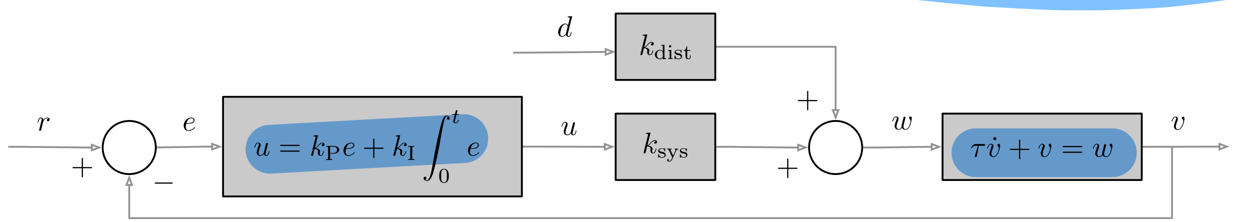


Figure 7.12: Closed-loop PI control of the dynamic car velocity model:

$$\tau \dot{v}(t) + v(t) = k_{\text{sys}} u(t) + k_{\text{dist}} d(t),$$
$$e(t) = r - v(t),$$
$$u(t) = k_{\text{P}} e(t) + k_{\text{I}} \int_0^t e(\sigma) d\sigma.$$

To simplify the analysis of block diagrams and interconnected systems, we now transform these time-domain equations into the Laplace domain, where differentiation and integration become simple algebraic multiplications and divisions in s. In other words, we now argue that it is convenient to represent feedback control systems via block diagrams in the Laplace domain.

Recall our convention that lowercase letters (e.g., $y(t)$) denote time-domain signals, while uppercase letters (e.g., $Y(s)$) denote their Laplace transforms. In block diagrams, we may use either notation, since the algebra is consistent in both domains.

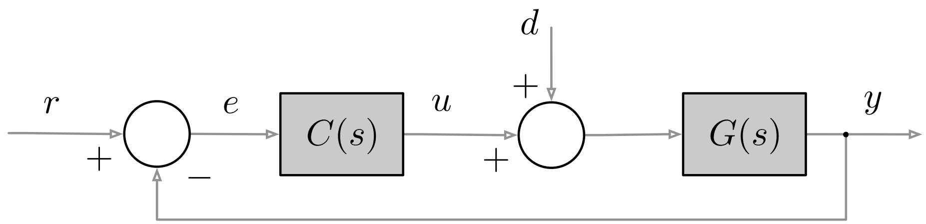


Figure 7.13: A feedback diagram. This block diagram is equivalent to the following equations:

$$Y(s) = G(s)(U(s) + D(s)),$$
$$E(s) = R(s) - Y(s),$$
$$U(s) = C(s)E(s),$$

where, as usual, we let $R(s)$ be a reference signal and $Y(s)$ be the system response.

The three equations in the caption of Figure 7.13 are essentially the same as the three equations in the caption of Figure 7.12. But it is substantially easier to manipulate multiplication and division by s, rather than differentiation and integration with respect to time. For instance, solving the three equations in Figure 7.13 is relatively easy and yields the closed-loop transfer function:

closed loop TF =
$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)} \overset{\text{if } C(s)G(s) = \frac{\text{Num}(s)}{\text{Den}(s)}}{=} \frac{\text{Num}(s)}{\text{Num}(s) + \text{Den}(s)}. \tag{7.9}$$

open loop TF = $C(s)G(s) = \frac{num(s)}{den(s)}$

$Y = G(U+D)$, $E = R - Y$, $U = CE$ } $Y = G(U+D)$, $U = C(R-Y)$ } $Y = G(C(R-Y)+D)$

$D = 0$. $Y = GC(R-Y) \Rightarrow (1 + G(s)C(s))Y(s) = G(s)C(s) \cdot R(s)$

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)}$$

7.4.1 Block diagram algebra for interconnected transfer functions

Block diagrams in the Laplace domain allow for easy manipulation, as illustrated in Figure 7.14.

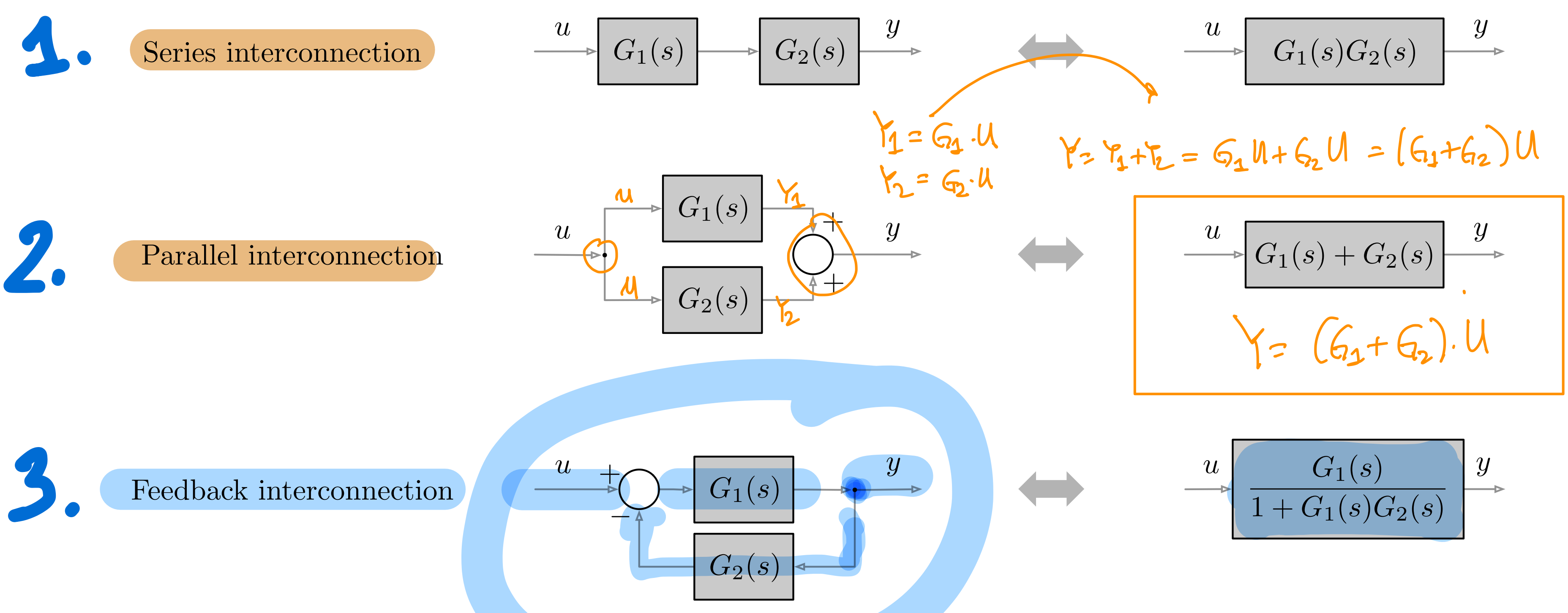


Figure 7.14: The three basic block diagram interconnections in the Laplace domain: series, parallel, and feedback. Each interconnection has an equivalent representation as a single block with equivalent transfer function. These equivalences (and others, e.g., see Exercise E7.8) allow straightforward manipulation of complex block diagrams of transfer functions.

The first two results in Figure 7.14 are easy to see. To verify the formula for the feedback interconnection, we compute

$$Y(s) = G_1(s)\big(U(s) - G_2(s)Y(s)\big) \quad \Longrightarrow \quad \big(1 + G_1(s)G_2(s)\big)Y(s) = G_1(s)U(s) \quad \Longrightarrow \quad Y(s) = \frac{G_1(s)}{1 + G_1(s)G_2(s)}U(s).$$

This calculation also confirms the correctness of equation (7.9).

7.4.2 The superposition property of linear systems

superposition

Consider a block diagram where two input signals u_1 and u_2 are summed into an input signal u and then fed into a multiplicative block with constant k. Then the output y satisfies:

$$y = ku = k(u_1 + u_2) = ku_1 + ku_2 \tag{7.10}$$

If we were to feed into the block the two inputs separately, we would obtain

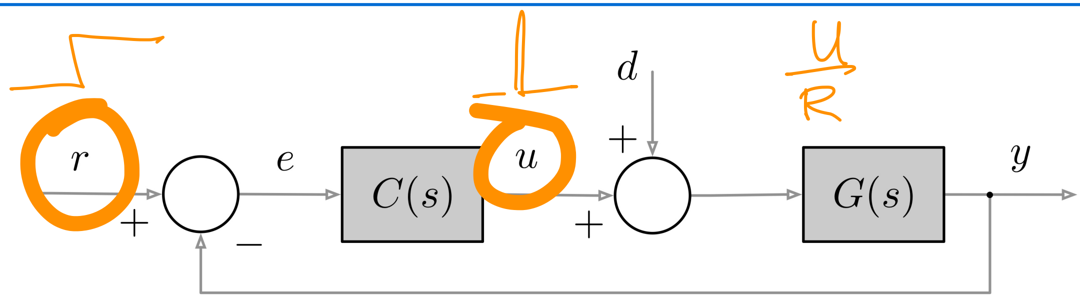
$$y_1 = ku_1 \quad \text{and} \quad y_2 = ku_2 \tag{7.11}$$

The multiplicative block is linear and therefore it satisfies the *superposition property*, namely:

$$y = y_1 + y_2 \tag{7.12}$$

The interpretation is as follows: the effect due to the sum of two causes is the sum of the two individual isolated effects.

For linear dynamical systems, the superposition property is that the response of a linear system to a sum of inputs is equal to the sum of the individual responses to each input.



two inputs and one output

r, d → y

$U = CE = C(R-Y)$, $Y = G(U+D) = GD + GC(R-Y) = GD + GCR - GCY$

$(1+GC)Y = GD + GC \cdot R \Rightarrow$

$$Y(s) = \frac{GC}{1+GC} R + \frac{G}{1+GC} D$$

7.5 Historical notes and further resources

Classic textbooks on control include (DiStefano et al., 1997; Ogata, 2003; Dorf and Bishop, 2011; Nise, 2019; Franklin et al., 2015). A modern approach is taken by Åström and Murray (2021).

Here is a recommended award-winning video explaining sensing, actuation and control. It is entitled Automation (2m 42sec).

The first Bode lecture "Respect the unstable," delivered by Dr. 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* in control design. (1h 11m).

7.6 Appendix: Open-loop and closed-loop control for a static model

In this appendix we

(i) define a static control model,

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

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

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

7.6.1 A static control problem

Recall from (7.1), the *first-order system with control and disturbance* is

$$\tau \dot{v} + v \;=\; k_{\mathsf{sys}} u + k_{\mathsf{dist}} d \tag{7.13}$$

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*:

$$v \;=\; k_{\mathsf{sys}} u + k_{\mathsf{dist}} d. \tag{7.14}$$

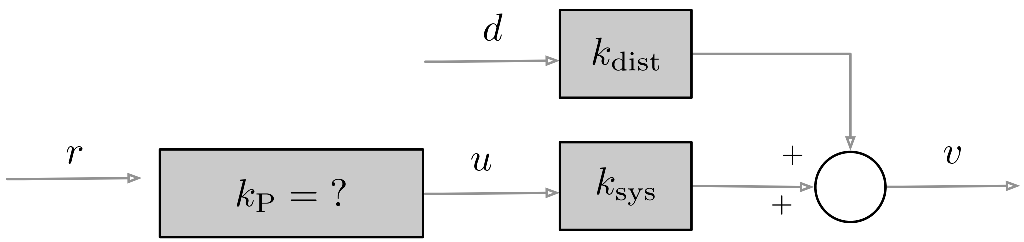


Figure 7.15: Static model with state variable v, input variable u, disturbance variable d, and reference variable r:
$v \;=\; k_{\mathsf{sys}} u + k_{\mathsf{dist}} d$
with an input
$u \;=\; k_{\mathsf{P}} r$
where the gain k_{P} is not yet specified.
The input action $u = k_{\mathsf{P}} r$ is called *proportional*.

In class assignment

Given a static system with control and disturbance in equation (7.14) and a proportional controller block with an unspecified gain as in Figure 7.15, what gain k_{P} would you choose to achieve reference tracking (i.e., $v = r$)?

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

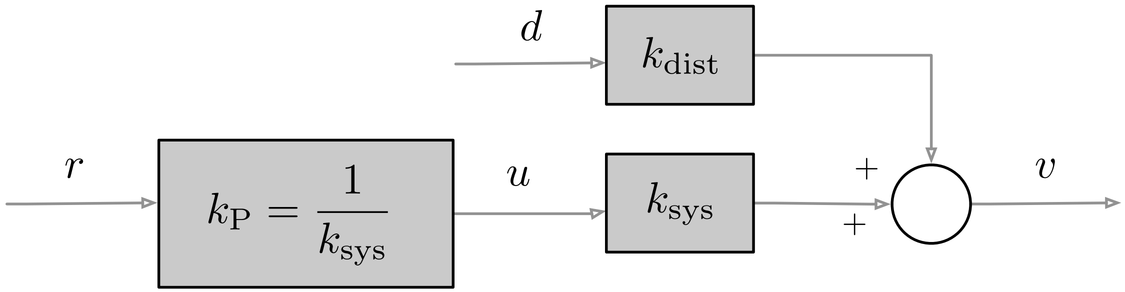


Figure 7.16: Open-loop cruise control of the static car velocity model, illustrated via a block diagram with a control block. The controller block has a gain equal to $1/k_{sys}$, the exact inverse of the effect of the control on the system.

- *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*

$$u = k_P r, \qquad \text{where } k_P \text{ is a } \textit{control gain}. \quad \text{We select } k_P = \frac{1}{k_{sys}} \tag{7.15}$$

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

$$v_{\text{open-loop}} = k_{sys}u + k_{dist}d\Big|_{u=k_P r, k_P=1/k_{sys}} = k_P k_{sys} r + k_{dist}d\Big|_{k_P=1/k_{sys}} = r + k_{dist}d \tag{7.16}$$

- 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.

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

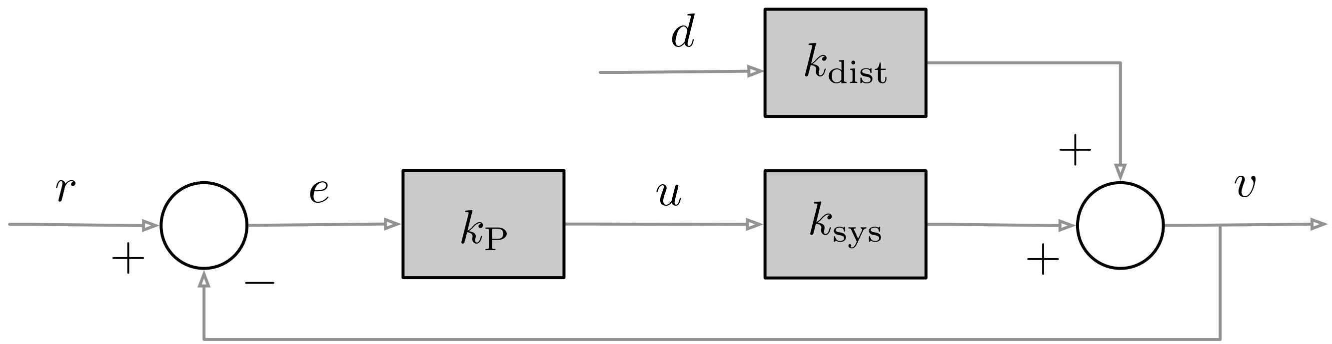


Figure 7.17: Closed-loop feedback control of the static car velocity model.
Illustration via a block diagram with control block and a feedback loop.
Note that the light blue boxes describes, respectively, the *system* (to be controlled and subject to a disturbance) and the *controller*.
In the closed-loop 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:

$$e = r - v_{\text{closed-loop}} \tag{7.17}$$

- 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*.

We describe the closed-loop in the block diagram via two equations:

$$\begin{cases} v_{\text{closed-loop}} & = k_{\text{sys}} u + k_{\text{dist}} d \\ u & = k_{\text{P}} e = k_{\text{P}}(r - v_{\text{closed-loop}}) \end{cases} \tag{7.18}$$

Hence,

$$v_{\text{closed-loop}} = k_{\text{sys}} k_{\text{P}} r - k_{\text{sys}} k_{\text{P}} v_{\text{closed-loop}} + k_{\text{dist}} d. \tag{7.19}$$

In summary,

$$v_{\text{closed-loop}} = \frac{k_{\text{sys}} k_{\text{P}}}{1 + k_{\text{sys}} k_{\text{P}}} r + \frac{k_{\text{dist}}}{1 + k_{\text{sys}} k_{\text{P}}} d \overset{k_{\text{P}} k_{\text{sys}} \text{ large}}{\approx} r \tag{7.20}$$

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 (7.16) for the open-loop case:

$$v_{\text{open-loop}} = r + k_{\text{dist}} d \tag{7.21}$$

Remark 7.1 (Comparison between open and closed loop strategies). *(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) 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.

•

7.7 Exercises

Section 7.1: Basic control problems and block diagrams

E7.1 **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:

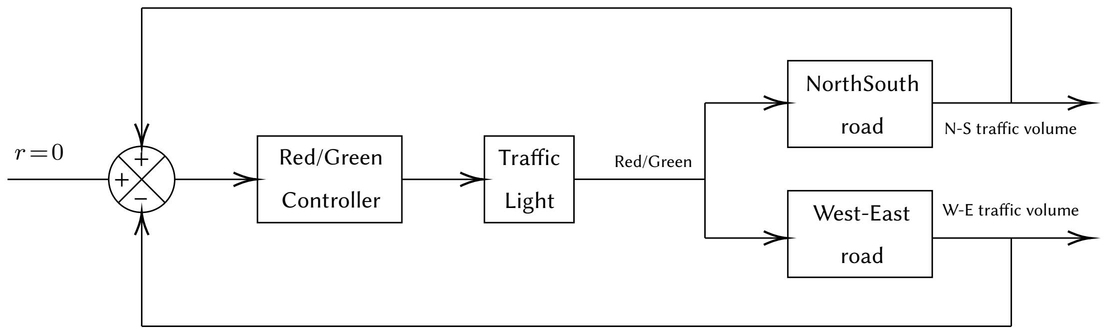
(i) The control action is the timing of the green/red light intervals. (When red is shown to North-South traffic, green is shown to West-East traffic, naturally.)

(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:



▼

Section 7.2: Proportional control

E7.2 **Closed-form of first-order system with proportional control.** Consider the system of equations (7.3), which we report here for convenience:

$$\begin{cases} \tau \dot{v} = -v + k_{\mathsf{sys}} u + k_{\mathsf{dist}} d \\ u = k_{\mathsf{P}}(r - v) \end{cases} \tag{7.22}$$

Show that this closed-loop system is again a first-order system of the form

$$\tau_{\mathsf{closed\text{-}loop}} \dot{v} = -v + \frac{k_{\mathsf{sys}} k_{\mathsf{P}}}{1 + k_{\mathsf{sys}} k_{\mathsf{P}}} r + \frac{k_{\mathsf{dist}}}{1 + k_{\mathsf{sys}} k_{\mathsf{P}}} d \tag{7.23}$$

with:

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

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

$$\begin{aligned} \tau\dot{v} &= -v + k_{\text{sys}}\big(k_{\text{P}}(r-v)\big) + k_{\text{dist}}d = -v + k_{\text{sys}}k_{\text{P}}r - k_{\text{sys}}k_{\text{P}}v + k_{\text{dist}}d && (7.24)\\ &= -(1+k_{\text{sys}}k_{\text{P}})v + k_{\text{sys}}k_{\text{P}}r + k_{\text{dist}}d. && (7.25) \end{aligned}$$

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

$$\frac{\tau}{1+k_{\text{sys}}k_{\text{P}}}\dot{v} \;=\; -v + \frac{k_{\text{sys}}k_{\text{P}}}{1+k_{\text{sys}}k_{\text{P}}}r + \frac{k_{\text{dist}}}{1+k_{\text{sys}}k_{\text{P}}}d. \qquad (7.26)$$

▼

Section 7.3: Proportional integral control

E7.3 **Closed-form of first-order system with proportional and integral control.** Consider the system of equations (7.7), which we report here for convenience:

$$\begin{cases} \tau\dot{v}(t) + v(t) = k_{\mathsf{sys}}u(t) + k_{\mathsf{dist}}d(t) \\ e(t) = r - v(t) \\ u(t) = \underbrace{k_{\mathsf{P}}e(t)}_{\text{proportional action}} + \underbrace{k_{\mathsf{I}}\int_0^t e(\sigma)d\sigma}_{\text{integral action}} \end{cases} \tag{7.27}$$

Show that this closed-loop system is a second-order system of the form

$$\tau\ddot{v}(t) + (1 + \underbrace{k_{\mathsf{sys}}k_{\mathsf{P}}}_{\text{proportional action}})\dot{v}(t) + \underbrace{k_{\mathsf{sys}}k_{\mathsf{I}}}_{\text{integral action}} v(t) = \underbrace{k_{\mathsf{sys}}k_{\mathsf{I}}r}_{\text{effect of reference}} + \underbrace{k_{\mathsf{dist}}\dot{d}(t)}_{\text{effect of disturbance}} \tag{7.28}$$

$$\tau\dot{v} + v = k_{sys}\left(k_P(r-v) + k_I\int_0^t (r - v(\sigma))d\sigma\right) + k_{dist}\, d$$

$\frac{d}{dt}$ of both LHS and RHS

$$\tau\ddot{v} + \dot{v} = k_{sys}k_P(-\dot{v}) + k_{sys}k_I(r - v(t)) + k_{dist}\dot{d}$$

Assume
$r(t)$ constant
$r(t) = r$

E7.4 **Properties of integral control.** In this exercise we study two properties of integral control.

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

$$u(t) = k_{\mathsf{P}} e(t) + k_{\mathsf{I}} \int_0^t e(\tau) d\tau \tag{7.29}$$

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.*

E7.5 **Integral control of first-order systems.** Consider a first-order system with time constant τ and system gain k_{sys} subject to integral control $u(t) = k_{\mathsf{I}} \int_0^t x(\sigma) d\sigma$ (only integral control, no proportional control). Given a positive number α, define the integral control gain to be $k_{\mathsf{I}} = \alpha \frac{1}{k_{\mathsf{sys}}\tau}$. For the resulting closed-loop system, answer the following questions.

(i) Compute the natural frequency ω_{n} and damping ratio ζ.

(ii) Can you choose both ω_{n} and ζ arbitrarily?

(iii) Determine for what values of α the system is underdamped, critically damped, and overdamped.

(iv) Compute the damped natural frequency ω_{d} and the two complex conjugate poles, when the system is underdamped.

(v) Compute 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.*

E7.6 **Tuning PI gains for a first-order cruise-control system.** Consider the closed-loop cruise-control dynamics under PI control

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

obtained in Section 7.3 from a first-order plant with proportional–integral feedback.

(i) *Mass–spring–damper analogy.* By identifying $\bar{m} = \tau$, $\bar{b} = 1 + k_{\mathsf{sys}}k_{\mathsf{P}}$, $\bar{k} = k_{\mathsf{sys}}k_{\mathsf{I}}$, derive the closed-loop natural frequency ω_{n} and damping ratio ζ in terms of $(\tau, k_{\mathsf{sys}}, k_{\mathsf{P}}, k_{\mathsf{I}})$.

(ii) *Minimum achievable damping.* Show that for fixed $(\tau, k_{\mathsf{sys}}, k_{\mathsf{I}})$, the smallest achievable damping ratio (over all $k_{\mathsf{P}} \geq 0$) occurs at $k_{\mathsf{P}} = 0$ and equals $\zeta_{\min} = \dfrac{1}{2\sqrt{\tau\, k_{\mathsf{sys}}\, k_{\mathsf{I}}}}$. Interpret this limitation physically.

(iii) *Inverse design.* Given desired closed-loop specifications $(\omega_{\mathsf{n}}, \zeta)$ with $\zeta \geq \zeta_{\min}$, solve for PI gains $(k_{\mathsf{P}}, k_{\mathsf{I}})$ in terms of $(\tau, k_{\mathsf{sys}}, \omega_{\mathsf{n}}, \zeta)$.

(iv) *Feasibility condition.* Rewrite the condition $\zeta \geq \zeta_{\min}$ as a constraint coupling $(\omega_{\mathsf{n}}, \zeta)$, independent of k_{I}, and explain what it implies for attempting very small ζ (highly underdamped designs).

(v) *Numerical design and interpretation.* For $\tau = 5$ and $k_{\mathsf{sys}} = 3$, choose $\omega_{\mathsf{n}} = 1$ and $\zeta = 0.6$:

   (i) compute $(k_{\mathsf{P}}, k_{\mathsf{I}})$;
   (ii) predict qualitatively the step response of $v(t)$ to a unit step in r;
   (iii) discuss how increasing ω_{n} while keeping ζ fixed affects the control effort $u(t)$ (magnitude/peaks) and actuator saturation risk.

Hint: Compare the characteristic polynomial $\tau s^2 + (1 + k_{\mathsf{sys}}k_{\mathsf{P}})s + k_{\mathsf{sys}}k_{\mathsf{I}}$ with the standard $s^2 + 2\zeta\omega_{\mathsf{n}}s + \omega_{\mathsf{n}}^2$.

Answer:

(i) Matching coefficients:

$$\omega_\mathsf{n} = \sqrt{\frac{k_\mathsf{sys} k_\mathsf{I}}{\tau}} \qquad \text{and} \qquad \zeta = \frac{1 + k_\mathsf{sys} k_\mathsf{P}}{2\sqrt{\tau k_\mathsf{sys} k_\mathsf{I}}} \tag{7.30}$$

(ii) For fixed $(\tau, k_\mathsf{sys}, k_\mathsf{I})$, ζ increases monotonically with k_P, so ζ_min occurs at $k_\mathsf{P} = 0$: $\zeta_\mathrm{min} = \frac{1}{2\sqrt{\tau\, k_\mathsf{sys}\, k_\mathsf{I}}}$. Physically, even with $k_\mathsf{P} = 0$ the plant has intrinsic damping (1 in the $\dot{v}$ coefficient), so the closed loop cannot be less damped than that.

(iii) Solve $\omega_\mathsf{n}^2 = \frac{k_\mathsf{sys} k_\mathsf{I}}{\tau}$ to get

$$k_\mathsf{I} = \frac{\tau\, \omega_\mathsf{n}^2}{k_\mathsf{sys}}, \tag{7.31}$$

and substitute into ζ to get

$$k_\mathsf{P} = \frac{2\tau\, \omega_\mathsf{n}\, \zeta - 1}{k_\mathsf{sys}}. \tag{7.32}$$

(iv) Using (iii) with $k_\mathsf{P} \geq 0$ gives $2\tau\omega_\mathsf{n}\zeta \geq 1$ or $\zeta \geq \frac{1}{2\tau\omega_\mathsf{n}}$. Thus very small ζ at a fixed ω_n is infeasible; achieving very small ζ demands either reducing ω_n or altering the system parameters.

(v) We fix $\tau = 5$, $k_\mathsf{sys} = 3$, $\omega_\mathsf{n} = 1$, $\zeta = 0.6$.

(a) The control gains are:

$$k_\mathsf{I} = \frac{5 \cdot 1^2}{3} = \frac{5}{3}, \qquad k_\mathsf{P} = \frac{2 \cdot 5 \cdot 1 \cdot 0.6 - 1}{3} = \frac{6-1}{3} = \frac{5}{3}.$$

(b) Regarding the step response: underdamped ($\zeta = 0.6$) with modest overshoot and finite settling time $\approx 4/(\zeta\omega_\mathsf{n}) \approx 6.7\,\mathrm{s}$ (rule of thumb).

(c) Increasing ω_n (fixed ζ) scales $k_\mathsf{I} \propto \omega_\mathsf{n}^2$ and $k_\mathsf{P} \propto \omega_\mathsf{n}$, yielding faster dynamics but larger percent overshoot and higher saturation/noise sensitivity risks.

▼

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

```
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, 10)

# 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, 10, 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, ...
        10); 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 $t$'); ...
        ax2.set_ylabel('control input'); ax2.set_xlim(0, 10); ...
        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 7.18. Available at cruise-control-proportional-integral-design.py

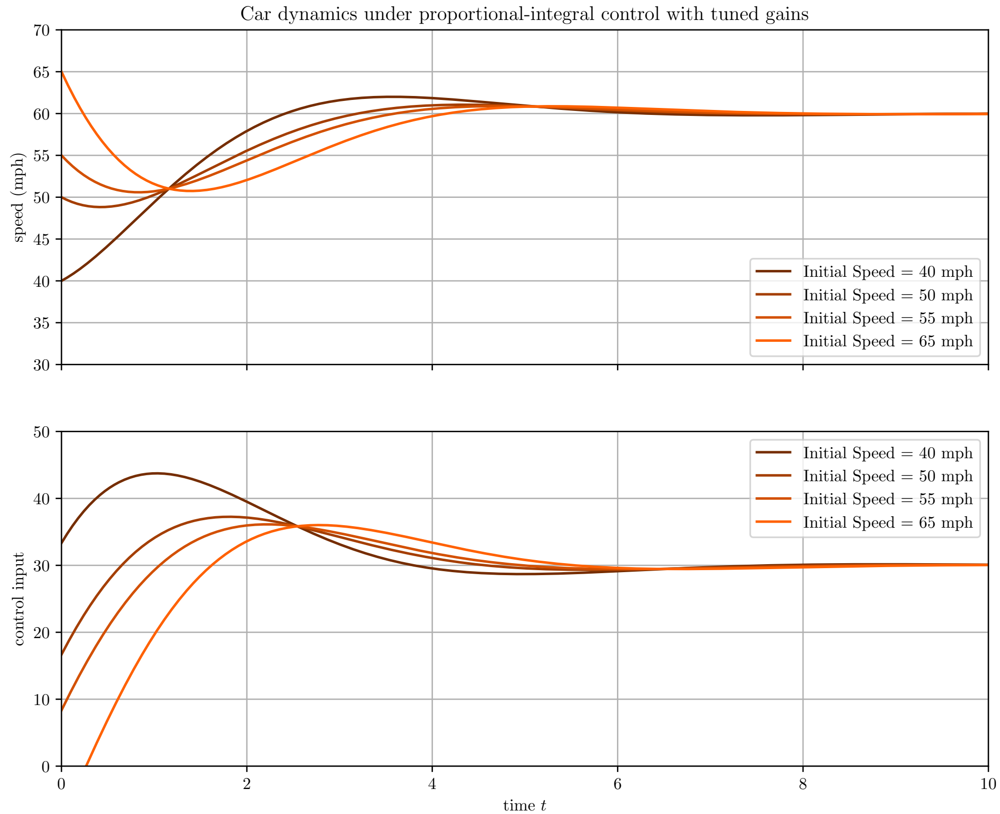
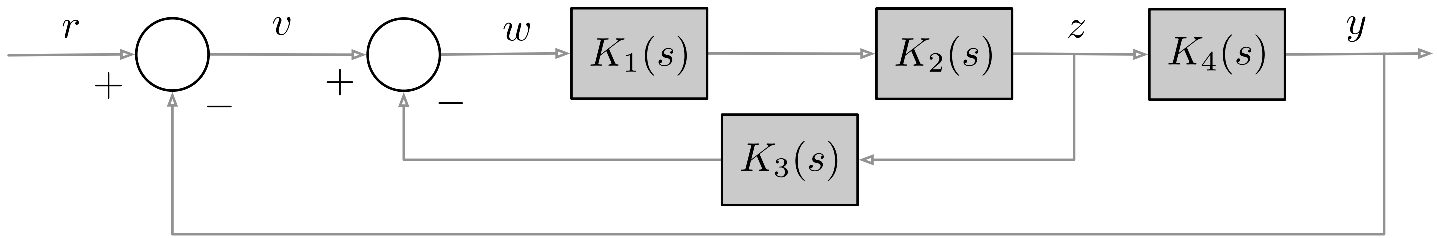


Figure 7.18: Solutions of the cruise-control dynamics with proportional and integral control (7.7): $v(t)$ in the first plot, $u(t)$ in the second plot. (The response is from multiple non-zero initial conditions and in response to a step input in r and d.)
Proportional and integral gains computed from (7.31)-(7.32) 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 inevitably lead to very large control signals (since $k_P \propto \omega_n$ and $k_I \propto \omega_n^2$).

Section 7.4: Block diagrams in the Laplace domain

E7.7 **An example block diagram.** Compute the transfer function $Y(s)/R(s)$ for the block diagram in figure.



E7.8 **Block diagram algebra (Franklin et al., 2015).** Show the three equivalences depicted in Figure 7.19.

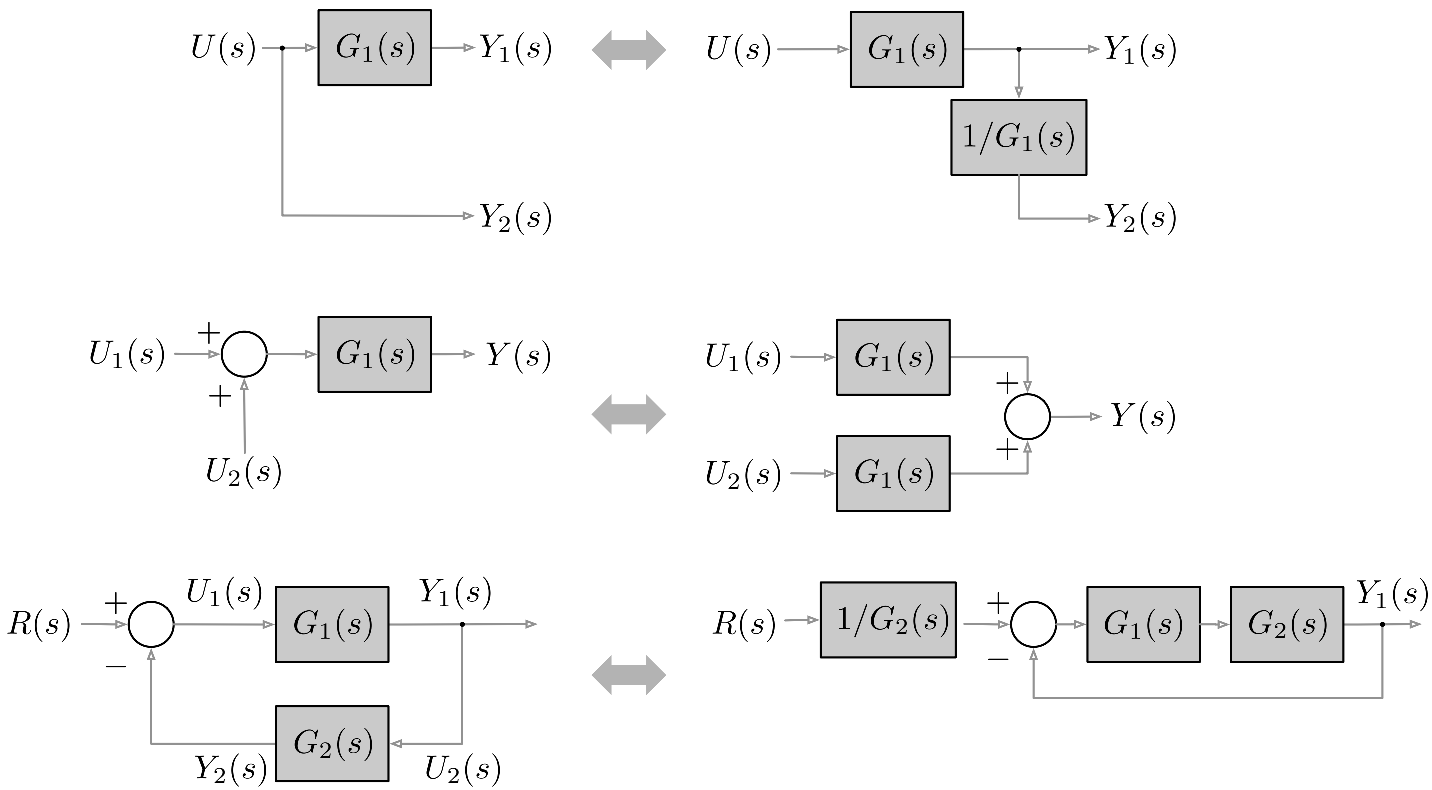


Figure 7.19: Equivalences in block diagram algebra

Appendix 7.6: Open-loop and closed-loop control for a static model

E7.9 **Advantages of closed-loop control for static systems (Åström and Murray, 2021).** In this exercise we study two advantages of negative feedback in simple static systems: reduced sensitivity to parameter variations and a widened linearity regime. We start by studying how *feedback enhances robustness to parameter variations.*

(i) Given an open-loop gain $k > 0$, consider the static input–output map $y = kr$, as illustrated in the left image in Figure 7.22. Show that, under open-loop control (i.e., for $y = kr$), the relative variation in the output y due to a variation in the gain k is

$$\frac{dy}{y} = \frac{dk}{k}.$$

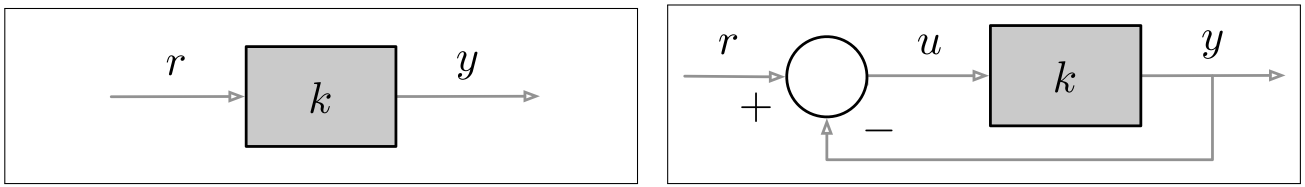


Figure 7.20: Left image: open-loop static map $y = kr$. Right image: closed-loop diagram with feedback signal $u = r - y$. The open-loop gain is k for both open-loop and closed-loop block diagrams.

(ii) Consider now the closed-loop system depicted in the right image in Figure 7.22. Show that the input–output relation is

$$y = \frac{k}{1+k} r.$$

Derive the relative variation in the output y with respect to k and show that

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

(iii) Compare the two cases for $k \approx 1$ and $k \approx 100$. Quantify how a 10% change in k affects the output y in both cases. What does this illustrate about the robustness of closed-loop control?

Next, we study how *feedback widens the linearity regime.* Consider the nonlinear static system with saturation nonlinearity as depicted in Figure 7.21.

(iv) Consider the open-loop system $y = \mathrm{sat}(kr)$ depicted in the left image in Figure 7.22. Show that the input–output map is linear with gain k for $|r| < 1/k$.

(v) Consider the closed-loop system depicted in the right image in Figure 7.22, with equations

$$y = \mathrm{sat}(u), \qquad u = k(r - y).$$

Derive the closed-loop input–output relation

$$y = \mathrm{sat}\Big(\frac{k}{k+1} r\Big).$$

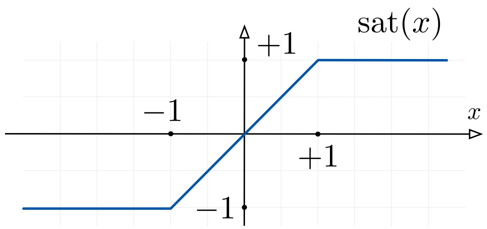


Figure 7.21: The *saturation nonlinearity* is often found in control problems. It is formally defined by:

$$\operatorname{sat}(x) = \begin{cases} +1, & x > 1 \\ x, & -1 \le x \le 1 \\ -1, & x < -1 \end{cases}.$$

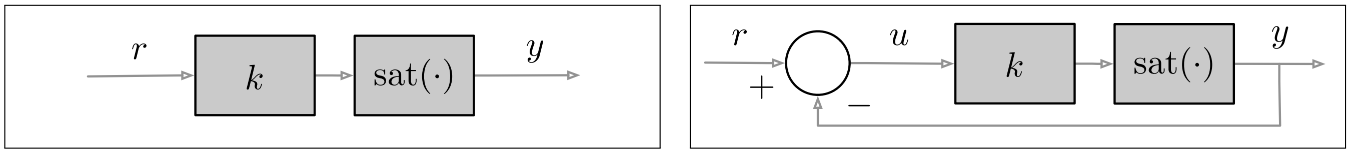


Figure 7.22: Left image: open-loop static map $y = \operatorname{sat}(kr)$. Right image: closed-loop diagram with feedback signal $u = r - y$.

(vi) Show that the closed-loop system is linear with gain $\dfrac{k}{k+1}$ for $|r| < 1 + \dfrac{1}{k}$.

(vii) Compare the ranges of linearity in the open-loop and closed-loop cases. By what factor is the linear range widened in the closed loop?

Note: *In summary, we see that negative feedback reduces sensitivity to parameter variations and enlarges the linear operating range of systems subject to saturation nonlinearities.*

Answer: Regarding *robustness to parameter variations*, let $y = kr$.

(i) In the open-loop case, the relation between input and output is simply $y = kr$. If we consider the effect of changing the gain k while keeping r fixed, the derivative of y with respect to k is

$$\frac{dy}{dk} = r.$$

Since $r = \frac{y}{k}$, this derivative can also be written as $\frac{dy}{dk} = \frac{y}{k}$. Dividing both sides by y gives

$$\frac{1}{y}\frac{dy}{dk} = \frac{1}{k},$$

which leads to the expression

$$\frac{dy}{y} = \frac{dk}{k}. \tag{7.33}$$

This shows that the relative variation in y is exactly equal to the relative variation in k. Therefore, in open loop, the system output is fully sensitive to parameter changes: a $p\%$ variation in k leads to a $p\%$ variation in y.

(ii) In the closed-loop case, the system equations are $y = ku$ and $u = r - y$. Substituting gives $y = k(r - y)$, which rearranges to

$$(1 + k)y = kr \qquad \Longrightarrow \qquad y = \frac{k}{1+k}r.$$

To analyze sensitivity, we compute the derivative with respect to k:

$$\frac{dy}{dk} = \frac{r}{(1+k)^2}.$$

Next, we eliminate r using the relation $r = \frac{1+k}{k}y$, which yields

$$\frac{dy}{dk} = \frac{1}{k(1+k)}y.$$

Dividing both sides by y, we obtain

$$\frac{dy}{y} = \frac{1}{1+k}\frac{dk}{k}. \tag{7.34}$$

This shows that in closed loop, the relative change in y is reduced by the factor $1/(1 + k)$ compared to the open-loop case.

(iii) Let us now compare numerically. When $k \approx 1$, a 10% variation in k causes about a 10% change in y in open loop, but only about a 5% change in closed loop. When $k \approx 100$, a 10% change in k still causes a 10% change in y in open loop, but only about a 0.1% change in closed loop.

In summary, negative feedback reduces sensitivity to parameter variations by the factor $1/(1 + k)$. This is one of the key advantages of closed-loop systems: their outputs are much less affected by uncertainty or drift in system parameters.

Next, we answer the questions regarding the *widened linearity regime.*

(iv) In the open-loop case, the input–output relation is $y = \mathrm{sat}(kr)$. The system behaves linearly as long as the argument of the saturation lies in the interval $(-1, 1)$. This condition is $|kr| < 1$, which is equivalent to

$$|r| < \frac{1}{k}. \tag{7.35}$$

(v) In the closed-loop case, the equations are $y = \mathrm{sat}(u)$ and $u = k(r - y)$. If the signal u remains within the linear region of the saturation (that is, $|u| < 1$), then $y = u$. Substituting this into the relation for u gives $y = k(r - y)$, or equivalently

$$(1 + k)y = kr \quad \Longrightarrow \quad y = \frac{k}{1+k} r. \tag{7.36}$$

Thus, in the absence of saturation, the closed-loop input–output map is linear with effective gain $k/(1 + k)$. When saturation is taken into account, the correct closed-loop relation is

$$y = \mathrm{sat}\left(\frac{k}{k+1} r\right). \tag{7.37}$$

(vi) For the closed-loop system to remain linear, the argument of the saturation must again lie between -1 and $+1$. This requires

$$\left|\frac{k}{1+k} r\right| < 1, \tag{7.38}$$

which simplifies to

$$|r| < 1 + \frac{1}{k}. \tag{7.39}$$

(vii) Finally, we compare the ranges of linearity. In open loop, the system is linear for $|r| < 1/k$, whereas in closed loop it is linear for $|r| < 1 + 1/k$. The factor of improvement is

$$\frac{1 + 1/k}{1/k} = k + 1. \tag{7.40}$$

Thus, negative feedback widens the linear operating range of the system by a factor of $k + 1$.

▼

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.

R. H. Dorf and R. C. Bishop. *Modern Control Systems*. Prentice Hall, 2011. ISBN 0136024580.

G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall, 4 edition, 2015.

N. S. Nise. *Control Systems Engineering*. John Wiley & Sons, 2019. ISBN 1119590132.

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