INTERACTIVE CONTROL ENGINEERING
Principles and Applications

Volume 20 Motion Control, Robotics, and Autonomous Systems

CHAPTER 1

# Introduction: Controlling Machines That Move

Pages 1–23

Edited by John D. Hedengren

**How to cite this chapter.** "Introduction: Controlling Machines That Move," in *Motion Control, Robotics, and Autonomous Systems*, John D. Hedengren, Ed., vol. 20 of *Interactive Control Engineering: Principles and Applications*. Advanced Process Solutions, LLC, 2026, ch. 1, pp. 1–23.

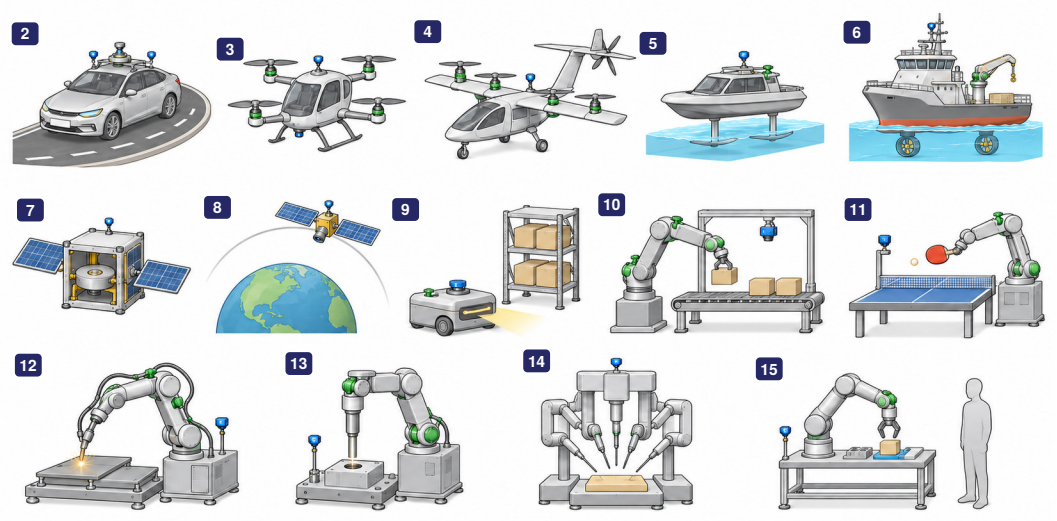
The page numbers in this file are those of the printed volume.

**The volume online**: contents, other chapters and citation records
https://apmonitor.com/apps/books/v20-motion-robotics/

Copyright © 2026 Advanced Process Solutions, LLC. All rights reserved. Produced by Advanced Process Solutions, LLC.

CHAPTER 1

# Introduction: Controlling Machines That Move




**The chapters** 2 lane keeping 3 drone taxi 4 lift-plus-cruise transition 5 foiling boat 6 dynamic positioning 7 CubeSat attitude 8 orbit maintenance 9 warehouse robot 10 conveyor picking 11 table tennis robot 12 arc welding 13 peg-in-hole assembly 14 surgical robot 15 collaborative robot

*Notice that the top row moves through road, air and sea, the middle row through orbit and a warehouse to the first work cells, and the bottom row brings the arm into contact with parts, tissue and a person. Illustration, not to scale.*

**IN PLAIN TERMS** **Guiding Machines That Move Through the World**

**The main idea.** A car keeps to its lane, a ship holds still over one spot of the seabed, and a robot arm picks a box off a moving belt. Each machine is pushed by motors, propellers or wheels, and it must know where it is to know how hard to push.

**The challenge.** Its position comes from sensors that drift, report late or go blind, and its load changes from one trip to the next. It must also be able to stop in time, even when what it believes about itself is wrong.

**Why it matters.** Good motion control lets machines carry goods, fly, sail and work safely beside people.

**THREE WAYS TO READ THIS CHAPTER**

| | | |
|---|---|---|
| **Explore** | 7 min | This page, Section 1.2 with its timeline and two History boxes, Figure 1.2, the Did you know? boxes in Sections 1.4.6, 1.4.7 and 1.4.9, and Key points in Section 1.8. No equations. |
| **Study** | 45 min | The whole chapter in order, working each number of the example C-1, then Exercises 1 to 10. |
| **Practice** | 5 min | This page, Section 1.4.10 with Table 1.2, Section 1.5 with Figures 1.7 and 1.10, and Section 1.8. |

## 1.1 The Question This Volume Answers

A machine that moves is controlled through geometry and inertia. A car keeps its lane by steering, a multirotor moves sideways only by tilting its thrust, a vessel holds station over a wellhead on thrusters alone, and a satellite turns its camera with wheels that keep every torque they absorb. A robot arm meets a product on a moving belt, presses a peg into a hole or works beside a person. In each case the quantity that matters is a distance to a path, a position over a pad, a force at a contact or a separation from a person. It is computed from estimates rather than read from one instrument, and the actuators

reach it through one or two integrations and a mass. This volume answers one question: how does a moving machine complete its task when its position, its load and its surroundings are known only approximately, and how is it kept from doing harm when they are wrong?

Eight control problems define the field, and every chapter meets several of them.

- *The error is geometric, and speed is a gain.* A turning command reaches a lateral position through two integrations whose coefficient is the speed, so a loop tuned at one speed meets a different plant at another.
- *Inertia changes with the load.* A payload changes the mass, the rotational inertia and the braking, and it changes at every stop.
- *Loops are nested.* A position loop commands a velocity or an attitude, which commands a force or a torque, and each inner loop must be faster than the loop it serves.
- *Motion is planned inside limits.* A reference must respect speed, acceleration, jerk and torque, and an interception must arrive at a place, at a time and at a speed.
- *The state is estimated, and often late.* Position comes from fusing encoders, inertial sensors, beacons and cameras. Reports arrive old, and part of a measured motion can be real and still not an error.
- *Actuators are shared, limited or directional.* There are more effectors than demands, effectors whose authority changes with speed, and actuators that act only across a field.
- *Contact turns position into force.* Once the tool touches the part, its stiffness converts every position error into a force.
- *A machine must be able to stop in time.* Its speed is bounded by what it believes about its braking and the space it can see, and a monitor must hold that bound against people who move.

Fourteen chapters answer the question on fourteen machines, each with an app that the reader runs in a browser. Chapters 2 to 8 move vehicles through a medium: a car in its lane, a passenger multirotor between rooftops, a lift-plus-cruise aircraft handing its weight from rotors to wing, a foiling boat, a vessel on dynamic positioning, a small satellite pointing a camera and a satellite holding its orbit among debris. Chapter 9 drives a pallet robot among people. Chapters 10 to 13 put an arm in a work cell, picking from a belt, returning a table tennis ball, welding a seam and inserting a peg. Chapters 14 and 15 close the volume on robots that work on or beside a person.

A reader who has worked through the volume can do the following on a moving machine that is new to the reader.

- Write the kinematics and the rigid-body balances of a vehicle or an arm and say what a payload changes.
- Design a cascade of position, velocity and force loops, and feed it the acceleration and the forces it can be told about.
- Generate a trajectory inside speed, acceleration and jerk limits, and say what smoothness costs in time.
- Make a path-following law independent of speed, and find the offset it leaves on a curve.
- Fuse a drifting sensor with a noisy one, remove a sensor bias and correct a report for its age.
- Allocate demands among limited actuators by priority or by weights.
- Set an impedance, predict a landing force and find the delay margin of a force loop.
- Turn a stopping distance into a speed limit near an obstacle or a person.

This chapter develops each of these on one small example and names the chapter that uses it. PID forms and tuning are taken as known from Volume 2, feedforward and cascade from Volume 3, allocation and overrides from Volume 4, estimation and Kalman filtering from Volume 5, unstable plants and state-space design from Volume 7, trajectories and switching from Volume 8 and predictive control from Volume 9. Standard texts treat the robotics [1], the marine craft [2] and the spacecraft [3] at length.

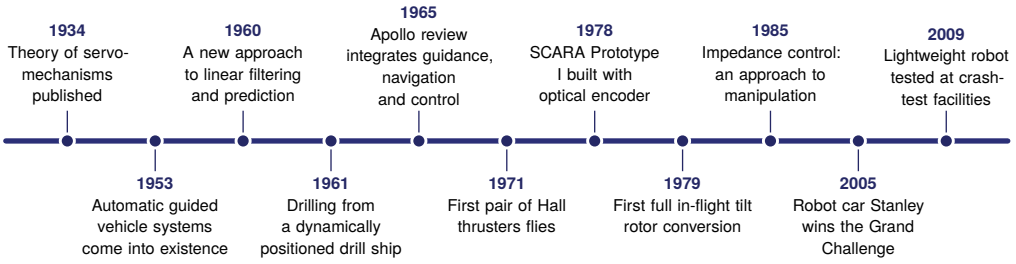



**Figure 1.1** Milestones of motion control, robotics and autonomous systems. The dates are from [4] (1934), [5] (1953), [6] (1960), [7] (1961), [8] (1965), [9] (1971), [10] (1978), [11] (1979), [12] (1985), [13] (2005) and [14] (2009). Notice that the vehicles come first and that the arms which touch parts and people come late. Events are evenly spaced, not to a time scale.

## 1.2 A Short History

The milestones of Figure 1.1 follow the machines of this volume, from the theory of the servo to arms tested for the harm they could do to a person.

A theory of servo-mechanisms was published in 1934 [4], and a servo that makes a shaft or a body follow its command is the motion loop of Section 1.4.3. Automatic guided vehicle systems have existed since 1953 [5], the forerunners of the pallet robot of Chapter 9. In 1960 a new approach to linear filtering and prediction [6] and a guidance system for the rendezvous of satellites [15] gave navigation two tools that Section 1.4.6 and Chapter 8 still use. In March and April of 1961 a drill ship held itself over deep water by dynamic positioning, with radar reflectors and acoustic transducers on buoys as its position reference [7], the problem of Chapter 6.

> ⧖ HISTORY **1961 to 1965: Gyros Aligned by Stars**
>
> The plan of late 1961 was to fly the lunar spacecraft within two to three years, which left no time for a new inertial system, so its design was based on an existing guidance system. Three integrating gyros held a stable member still in space while the spacecraft turned around it, and three accelerometers on that member measured the changes of velocity that the computer added to the state. The gyros drift, so in flight the member was realigned by sighting stars, and the most critical parameter was the gyro bias drift. A body-mounted design was examined and set aside, partly because no such system had yet left the laboratory. A review of 1965 integrated the guidance, navigation and control functions so that an alternate mode remained after a failure [8]. The star tracker and the gyro bias of Chapter 7 are the same pair.

A pair of Hall thrusters first flew in 1971, and more than 300 more had flown by 2018 [9], the kind of thruster that Chapter 8 fires. The first prototype of an assembly arm that yields sideways and is stiff vertically was built in October 1978, with a direct-current motor and an optical encoder as its servo elements [10]. A tilt rotor research aircraft made its first full conversion from helicopter to airplane in flight on 24 July 1979 [11], the transition of Chapter 4. Robot table tennis goes back to a challenge of 1983 whose simplified rules allowed for the technology of the time [16]. Impedance control was put forward in 1985 as an approach to manipulation [12], and a robot with improved positioning accuracy for brain surgery guided by computed tomography was reported in 1988 [17]. In 2009 a report described a lightweight arm impact-tested in automobile crash-test facilities to quantify the risk of injury it posed [14], the question behind Chapter 15.

> ⧖ HISTORY **2005: A Desert Robot Built in Layers**
>
> The robot car that won the desert race of October 2005 ran about 30 software modules in parallel, in six layers: sensor interface, perception, control, vehicle interface, user interface and global services. The sensor layer received each laser at 75 Hz, the camera at about 12 Hz, the satellite receiver and its compass at 10 Hz and the inertial unit and the vehicle bus at 100 Hz. No master process kept the modules in step. Every datum was time-stamped, and the stamps were used whenever sources were combined. A filter estimated the position, orientation and velocities, a planner set the trajectory, and two closed-loop controllers followed it, one for the steering and one for the brake and throttle [13]. Figure 1.2 has the same layers.

## 1.3 Notation and Conventions

Positions are in meters, angles in radians or degrees as each chapter gives them, forces in newtons and torques in newton meters. Speed is $v$, heading $\psi$, yaw rate $\omega$ and path curvature $\kappa$. A reference is marked with a subscript $r$ and an estimate with a hat. The framework's $h$ is its integration step and its $a$ the controller action, and neither is used for anything else here. This chapter therefore writes an acceleration limit as $A$, a braking capability as $A_b$ and a jerk limit as $j_{\max}$. Most chapters count time in seconds. Chapter 11, Chapter 12 and Chapter 13 quote their fast events in milliseconds, Chapter 7 counts orbits and minutes, and Chapter 8 counts hours and days.

Loop action follows Section 1.6.2: $a = +1$ is reverse acting, for a process whose measurement rises with the manipulated variable, and $a = -1$ is direct acting. A joint servo whose torque rises when the angle falls short of its reference is therefore reverse acting, as are a wheel speed loop, a force loop that moves its reference deeper when the force reads low and the altitude loop of a multirotor. The position loop of the multirotor in Chapter 3 is direct acting, because a positive tilt accelerates the vehicle toward negative $x$, and so is the pitch loop of the foiling boat in Chapter 5.

The chapters keep their own symbols, and several collide. $\kappa$ is the path curvature in Chapter 2 and the tilt stiffness $\ell K_{c,\theta}$ in Chapter 3. $\lambda$ is the rate of arc self-regulation in Chapter 12, the chamfer force factor in Chapter 13, a damping factor in Chapter 11 and a forgetting factor in Chapter 14. $\mu$ is a friction coefficient in Chapter 9 and Chapter 13, the reduced mass in Chapter 15, the thrust margin in Chapter 3 and the lift margin in Chapter 4. $J$ is the inertia matrix in Chapter 7 and the Jacobian in Chapter 10 and Chapter 11. $K_p$ is a pointing gain in Chapter 7 and a joint stiffness in Chapter 10, never the process gain $K_p$. $a$ is an acceleration in Chapter 2, the semimajor axis in Chapter 8, the braking deceleration in Chapter 15 (written $a_b$ for the braking capability in Chapter 9) and the lift-curve slope in Chapter 4, where $a_z$ and $a_x$ are demanded accelerations, none of them the action. $h$ is the ride height in Chapter 5, the restart hysteresis in Chapter 15, the servo period in Chapter 10 and the height above the surface in Chapter 4. $\theta$ is a tilt or a pitch in several chapters, the heading in Chapter 9 and the crossing angle of a debris pass in Chapter 8. $\varphi$ is the wand angle in Chapter 5, a wheel angle in Chapter 9 and the hole tilt in Chapter 13, while Chapter 7 writes the rotation vector $\phi$. $n$ is the orbital mean motion in the two space chapters. $K$ is an observer bandwidth in Chapter 3, an observer gain in Chapter 6 and a Kalman gain in Chapter 9.

## 1.4 Motion Under Uncertainty

### 1.4.1 The Flow Scheme

A moving machine has no pipes, but its signals follow one path in every chapter, shown in Figure 1.2. A mission layer or a supervisor chooses what to do next and when to switch modes. A planner turns that into a path or a trajectory in time. Motion loops, usually nested, compute the forces and torques that follow it. An allocation shares those demands among the actuators, which act on a rigid body that may be in contact with something. Sensors report on the body, and a navigation layer fuses their reports into the estimates that every other layer uses. A safety monitor watches the estimates and can stop the machine. The figure marks the layer where each chapter's main lesson lies and, in the bottom row, the medium in which each chapter's machine moves.

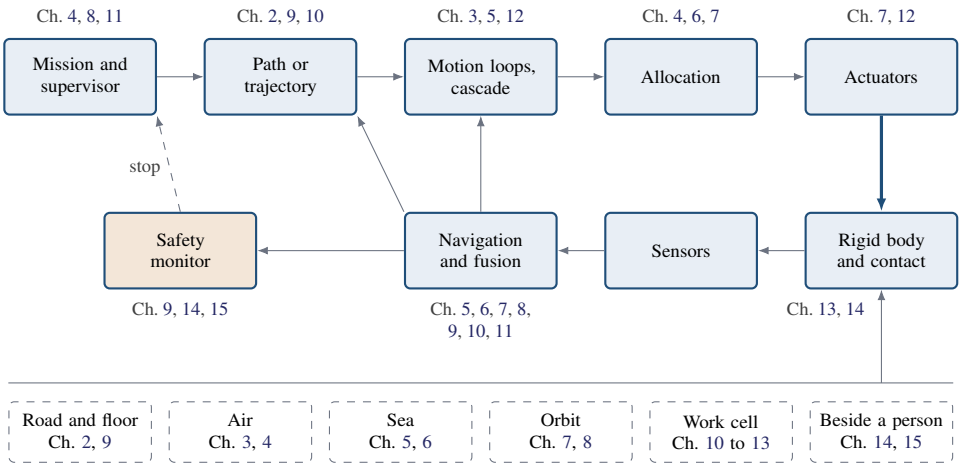



**Figure 1.2** The Flow Scheme of a Moving Machine. Solid Blocks Are the Layers Every Chapter Has, With the Chapters Whose Main Lesson Lies in Each Layer Next to It. The Safety Monitor Reads the Estimates and Can Stop the Machine. Dashed Blocks in the Bottom Row Place Each Chapter's Machine in the Medium It Moves In.

The techniques that the chapters share are developed below on one running example, a small transport cart called C-1. It has two drive wheels on a track of $b = 0.5\,\mathrm{m}$, each limited to 50 N of force at the floor, and casters elsewhere. It weighs 50 kg empty and 100 kg with its payload, so its largest acceleration is $2\,\mathrm{m/s^2}$ empty and $1\,\mathrm{m/s^2}$ loaded. It cruises at 1 m/s, and its controller executes every 10 ms. Every number of C-1 is declared here or where it is used and is computed by the script kept with this chapter's record. The numbers are chosen for clear arithmetic and are not taken from any one chapter.

### 1.4.2 Kinematics and Rigid-Body Dynamics

A wheel that rolls without slipping sideways constrains the body to move along its heading [18]. With the midpoint of the drive axle at $(x, y)$, the planar motion of C-1 is Equation (1.1).

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v\cos\psi, \qquad \frac{\mathrm{d}y}{\mathrm{d}t} = v\sin\psi, \qquad \frac{\mathrm{d}\psi}{\mathrm{d}t} = \omega, \qquad m\frac{\mathrm{d}v}{\mathrm{d}t} = F_R + F_L, \qquad I\frac{\mathrm{d}\omega}{\mathrm{d}t} = \frac{b}{2}(F_R - F_L) \quad (1.1)$$

where $F_R$ and $F_L$ are the wheel forces, $m$ the mass and $I$ the yaw inertia. The first three relations are the kinematics. They have no mass in them, and they make the lateral position the second integral of the turning command with the speed as a coefficient. The last two are the rigid-body balances. The payload doubles $m$ and raises $I$ from 3.00 to 4.33 $\mathrm{kg\,m^2}$, so the same wheel forces give half the acceleration and 69 % of the yaw acceleration. Chapter 9 drives a pallet robot on these equations with wheel slip added. Chapter 2 replaces the yaw-rate command by a steering angle, $\dot{\psi} = (v/L)\tan\delta$, which puts the speed into the heading equation as well.

An arm obeys the same balances in joint coordinates, as shown in Equation (1.2).

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + F(\dot{q}) = \tau, \qquad M \leftarrow M + m_p J^{\top} J \quad (1.2)$$

where $q$ holds the joint angles, $M$ is the inertia matrix, $C\dot{q}$ the velocity terms, $F$ the friction, $\tau$ the joint torques and $J$ the tip Jacobian [1]. A held payload $m_p$ adds $m_p J^{\top} J$ to the inertia, so it changes every entry of $M$ and the coupling between the joints. Chapter 10 checks that term to $10^{-12}$ and shows the torque budget a payload left out of the controller's model breaks. Chapter 15 drives a planar two-link arm of the same form, and Chapter 11 inverts the kinematics of a six-axis arm by damped least squares.

A body that turns in three dimensions needs one more idea. Any change of attitude is a rotation by one angle about one axis, the eigenaxis, and its rotation vector $\phi$ points along that axis with the angle as its length. A pointing law that commands a rate $-k\phi$ turns the body about the eigenaxis, the shortest path [19]. A rate limit keeps it there only if it limits the length of the command and not its parts. For an error of 1.0 and 0.25 rad about two axes, whose eigenaxis lies 14.0° from the first, clipping each component at 0.05 rad/s turns the body about an axis 45.0° away. Limiting the length keeps 14.0°. Chapter 7 flies a 90.45° slew through 90.62° with the length limited and through 105.28° with three separate axes. Its reaction wheels add a momentum budget. A wheel exchanges angular momentum with the body and cannot shed it, so a drag torque that never stops fills one wheel by 37.85 % of its capacity every orbit.

### 1.4.3 Cascaded Motion Loops

A motion loop is nested because the actuator sets a force and the task sets a position. On C-1 an inner loop sets the wheel force from the speed error, $F = K_v(v_r - v)$, and an outer loop sets the speed reference from the position error, $v_r = K_x(x_r - x)$. With $m\dot{v} = F$ the inner loop follows its reference with the time constant $\tau_v = m/K_v$, and the position loop has the characteristic equation (1.3).

$$\tau_v s^2 + s + K_x = 0, \qquad \zeta = \frac{1}{2\sqrt{\tau_v K_x}}, \qquad \tau_v = \frac{m}{K_v} \tag{1.3}$$

With $K_v = 250\,\mathrm{N\,s/m}$ and $K_x = 1.25\,\mathrm{s^{-1}}$ the empty cart has $\tau_v = 0.2\,\mathrm{s}$ and a double root at $-2.5\,\mathrm{s^{-1}}$, critically damped. The inner pole at $5\,\mathrm{s^{-1}}$ is four times the outer gain, which is the separation a cascade needs. Loaded, $\tau_v = 0.4\,\mathrm{s}$, the damping ratio falls to 0.707 and a position step overshoots by 4.3 %. Doubling the mass divides the damping ratio by $\sqrt{2}$. Scheduling $K_v$ in proportion to the believed mass, 500 N s/m loaded, restores the empty cart's roots.

A force the loops are not told about is opposed only through an error. At rest the two proportional loops balance a steady force $F_d$ with the position error of Equation (1.4).

$$e = \frac{F_d}{K_v K_x} \tag{1.4}$$

A 2 % grade pulls the loaded cart back with 19.6 N and leaves 63 mm. An integral term in the inner loop removes it over its integral time. A feedforward of the force removes it at once, if the force is known. A disturbance observer estimates an unknown force from the momentum balance and feeds it forward, so the error decays on the observer's time scale instead [20].

Chapter 3 is this subsection on a multirotor. Its position loop writes a tilt setpoint for a tilt loop that writes a differential thrust, with the two oscillatory modes 5.5 apart [21]. Its altitude gain is scheduled on the believed mass. It compares each loop's error with the proportional-only error of Equation (1.4), and the tilt loop, which has no integral, matches it to 0.01°. Correcting the weight feedforward alone brings a heavy climb from 1.78 m behind to 0.05 m, and an observer with a 0.67 s time constant cuts the drift after a wind step from 1.19 to 0.36 m. Chapter 5 closes its height loop on a heave-rate inner loop, and Chapter 6 feeds its estimated environmental force forward. Chapter 12 has a loop that no controller closes: a constant-voltage arc regrows most of a lost length by itself, with a time constant of 45.5 ms, settling at 3.82 mm against 4.29 mm before the step.

### 1.4.4 Trajectory Generation

A reference that steps asks the loops for an infinite acceleration and lets the actuator limits decide the path. A trajectory generator gives the loops a reference they can follow. For a move of length $D$ inside a speed limit $v_{\max}$ and an acceleration limit $A$, the trapezoid, the jerk-limited trapezoid and the quintic polynomial take the durations of Equation (1.5), for a move long enough to reach the speed limit.

$$T_{\mathrm{trap}} = \frac{D}{v_{\max}} + \frac{v_{\max}}{A}, \qquad T_{\mathrm{jerk}} = T_{\mathrm{trap}} + \frac{A}{j_{\max}}, \qquad T_{\mathrm{quin}} = \max\left(\frac{1.875\,D}{v_{\max}}, \sqrt{\frac{5.774\,D}{A}}\right) \tag{1.5}$$

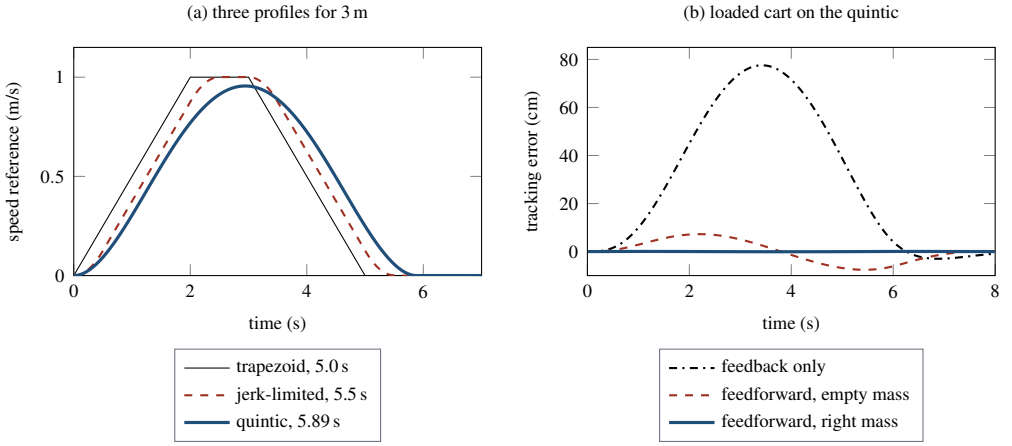



**Figure 1.3** A 3 m Move of C-1 at 1 m/s and 0.5 m/s$^2$. (a) The Trapezoid (Thin), the Jerk-Limited Trapezoid (Dashed) and the Quintic (Thick). (b) Tracking Error of the Loaded Cart on the Quintic Under the Cascade of Section 1.4.3: Feedback Alone (Dash-Dotted) Lags by Up to 78 cm, Feedforward With the Empty Mass (Dashed) Leaves 7.6 cm, and Feedforward With the Right Mass (Thick) Leaves 0.07 cm.

The quintic, $x_r = D(10s^3 - 15s^4 + 6s^5)$ with $s = t/T$, starts and ends at rest with zero acceleration, and it is the minimum-jerk path between two rests [22]. Its peak speed is $1.875\,D/T$, its peak acceleration $5.774\,D/T^2$ and its peak jerk $60\,D/T^3$. The jerk-limited form holds while $v_{\max} \geq A^2/j_{\max}$.

C-1 plans its moves at 1 m/s and 0.5 m/s$^2$, half its loaded capability, so that feedback keeps the other half. A move of 3 m then takes 5.0 s on a trapezoid, 5.5 s with a jerk limit of 1 m/s$^3$ and 5.89 s on a quintic, which is limited by its acceleration. Smoothness costs 18 % of the time against the trapezoid, and the quintic's peak jerk is 0.88 m/s$^3$. Figure 1.3(a) shows the three speed profiles.

A trajectory also carries the feedforward of Section 1.4.3. The outer loop adds the reference speed, $v_r = \dot{x}_r + K_x(x_r - x)$, and the inner loop adds the force the reference acceleration needs, $\hat{m}\ddot{x}_r$. Figure 1.3(b) shows the loaded cart on the quintic. Feedback alone lags by up to 0.78 m, close to the peak speed over $K_x$. Feedforward with the empty mass leaves 76 mm, because the missing 25 N at the peak acceleration acts like the force of Equation (1.4), which predicts 80 mm. Feedforward with the right mass leaves 0.7 mm.

Chapter 10 plans every move as one quintic per joint and feeds the inverse dynamics of Equation (1.2) forward, and its interception is a fixed point, because the goal depends on the duration and the duration on the goal. Chapter 11 shows the price of smoothness on a stroke: a minimum-jerk move of 0.3 m in 0.25 s needs 27.7 m/s$^2$ at its peak against 19.2 m/s$^2$ for a bang-bang move. Chapter 3 generates a jerk-limited reference online, which re-plans from the present state whenever the target moves [23], and Chapter 15 uses its trajectory generator to command a protective stop.

### 1.4.5 Path Following

When the task is a path rather than a schedule, the error is the lateral distance $y$ to the path and the heading error $\psi_e$ relative to it. On a path of curvature $\kappa$, Equation (1.1) gives Equation (1.6).

$$\frac{\mathrm{d}y}{\mathrm{d}t} = v \sin \psi_e, \qquad \frac{\mathrm{d}\psi_e}{\mathrm{d}t} = \omega - v\kappa \tag{1.6}$$

A fixed law $\omega = -k_h \psi_e - k_y y$ gives the linear characteristic polynomial $s^2 + k_h s + k_h k_y v$, whose roots move with speed. The law of Equation (1.7) divides the lateral term by the speed and adds the turn

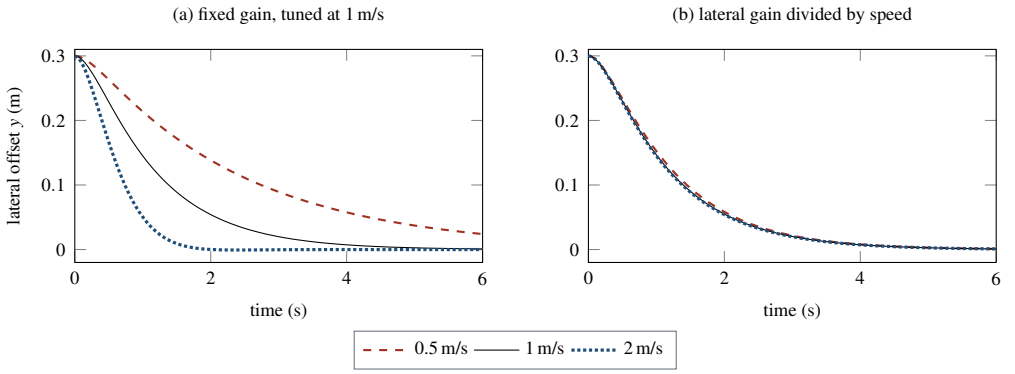



**Figure 1.4** C-1 Recovering From a 0.3 m Lateral Offset at 0.5 m/s (Dashed), 1 m/s (Thin) and 2 m/s (Dotted). (a) With Fixed Gains the Response Is Slow at Low Speed and Fast at High Speed. (b) With the Lateral Gain Divided by the Speed the Three Curves Nearly Coincide. The Small Lag at 0.5 m/s Is the Arctangent Limiting a Large Correction.

rate of the path [13, 24].

$$\omega = -k_h \left[ \psi_e + \arctan \frac{k_e y}{v} \right] + v\kappa \qquad \Longrightarrow \qquad s^2 + k_h s + k_h k_e = 0 \tag{1.7}$$

The speed in the lateral gain cancels the speed in the kinematics, so the poles stay put. For C-1 with $k_h = 5\,\mathrm{s}^{-1}$ and $k_y = k_e = 0.8$ (in $\mathrm{m}^{-1}\,\mathrm{s}^{-1}$ and $\mathrm{s}^{-1}$), both laws have poles at $-1$ and $-4\,\mathrm{s}^{-1}$ at 1 m/s. Figure 1.4 shows the recovery from a 0.3 m offset. The fixed law takes 2.52 s to 37 % of the offset at 0.5 m/s and 0.70 s at 2 m/s, and its damping ratio falls to 0.625 at 4 m/s. The speed-divided law takes 1.29 to 1.35 s at all three speeds, and covers twice the distance at twice the speed.

The term $v\kappa$ is a feedforward from the path itself. Without it the steady turn must come from a lateral error, and the balance of Equation (1.7) at $\psi_e = 0$ gives the offset of Equation (1.8).

$$y_\infty = -\frac{v^2 \kappa}{k_h k_e} \tag{1.8}$$

On a 4 m radius C-1 runs 0.063 m outside the path at 1 m/s and 0.25 m at 2 m/s, as the simulation confirms, and the feedforward removes it exactly. What the division by speed does not remove is everything the law leaves out, such as an actuator lag that costs a fixed time and therefore a distance that grows with speed.

Chapter 2 is this subsection on a car, with the Stanley law on a kinematic bicycle. Its cross-track time constant moves only from 0.917 to 0.861 s between 10 and 30 m/s. Its offset without the curvature feedforward is 0.314 m in a 200 m corner at 25 m/s, and a heading error referenced to the wrong point counts the feedforward twice. Its steering rack leaves every cross-track gain stable below $v = L/\tau_\delta = 29\,\mathrm{m/s}$ and sets a finite critical gain above it. Chapter 9 follows A* routes by pure pursuit, closing a heading loop on the bearing to a point ahead on the route [25]. Chapter 12 follows a seam read by a laser ahead of the torch, where a reading belongs to a place along the path and not to a moment.

### 1.4.6 Navigation and Sensor Fusion

No instrument reports where C-1 is. Its wheel encoders report the distance traveled, accurately over a second and with an error that grows with distance. A beacon reports the position every 0.1 s with a noise of 0.05 m that does not grow. Dead reckoning integrates the encoders alone, and a 2 % error in the wheel radius leaves 0.60 m after 30 m. A complementary filter uses each sensor where it is good,

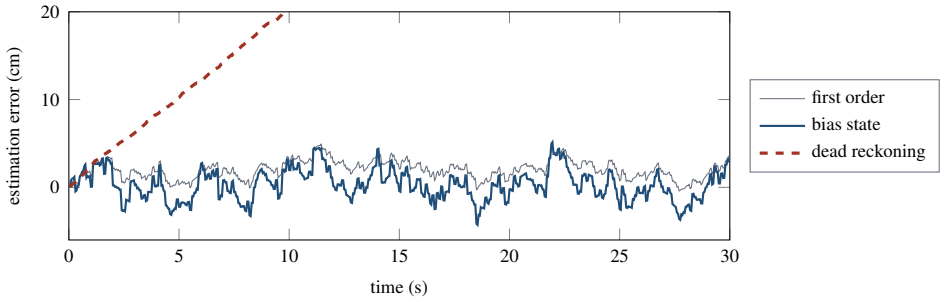



**Figure 1.5** Position Estimates of C-1 at 1 m/s With Encoders That Read 2 % High and a Beacon Fix Every 0.1 s With 0.05 m of Noise, Seed 11. Dead Reckoning (Dashed) Leaves the Axis at 10 s and Reaches 60 cm at 30 s. The First-Order Filter (Thin) Settles 1.9 cm Ahead. The Filter With a Bias State (Thick) Has No Steady Error.

as shown in Equation (1.9).

$$\frac{\mathrm{d}\hat{x}}{\mathrm{d}t} = v_m + \omega_c\,(y_m - \hat{x}) \qquad \Longrightarrow \qquad \hat{X} = \frac{\omega_c}{s+\omega_c}\,Y_m + \frac{s}{s+\omega_c}\,\frac{V_m}{s} \tag{1.9}$$

where $v_m$ is the encoder speed, $y_m$ the beacon fix and $\omega_c$ the crossover. Below $\omega_c$ the estimate follows the beacon and above it the encoders, and the two weights add to one at every frequency. A speed error $\varepsilon v$ that never stops leaves the steady estimation error $\varepsilon v/\omega_c$. A second state $\hat{b}$ that integrates the innovation estimates that error and removes it, as shown in Equation (1.10).

$$\frac{\mathrm{d}\hat{x}}{\mathrm{d}t} = v_m - \hat{b} + 2\omega_c\,(y_m - \hat{x}), \qquad \frac{\mathrm{d}\hat{b}}{\mathrm{d}t} = -\omega_c^2\,(y_m - \hat{x}) \tag{1.10}$$

Figure 1.5 shows the three estimates of C-1 at 1 m/s with $\omega_c = 1\,\mathrm{rad/s}$. After the first 15 s the first-order filter sits 0.019 m ahead, against $\varepsilon v/\omega_c = 0.020\,\mathrm{m}$, with an rms error of 0.022 m. The filter with the bias state has a mean error under 0.001 m and an rms error of 0.017 m, a third of the beacon's 0.050 m. A Kalman filter has the same structure and chooses the gains from the noise variances instead of a crossover [6, 26].

A report also has an age. A fix taken $L$ seconds ago describes where the machine was, and treating it as current leaves the error of Equation (1.11) at a speed $v$.

$$e_L = vL \tag{1.11}$$

With fixes 0.2 s late, the first-order filter of C-1 settles 0.181 m behind, which is $\varepsilon v/\omega_c - vL = -0.180\,\mathrm{m}$. Comparing each fix with the estimate stored for its capture time removes the $vL$ part and leaves the 0.019 m of the speed error. A moving target needs the same correction. A product on a belt has moved on by the belt speed times the report's age and the time still to come.

> ✦ DID YOU KNOW? **Two Minutes Without Satellites**
>
> The desert robot of 2005 kept an accurate pose through losses of its satellite fixes lasting up to 2 min, with gyros for its orientation and wheel speeds for its position. The error it gathered was usually of the order of centimeters, and its health monitor cut the top speed to 10 miles an hour while the fixes were lost [13].

The chapters build on these elements. Chapter 5 fuses a height sensor with an accelerometer through three poles at the crossover, with a bias state, and Chapter 9 fuses encoders and a gyro with reflector sightings in an extended Kalman filter that forms each innovation against the pose stored

**Table 1.1** Three Ways for C-1 to Cut a Demand of 80 N and 10 N m That Asks One Wheel for More Than Its 50 N

| Rule | $F_R$ (N) | $F_L$ (N) | Force $F$ | Moment $M$ |
|---|---|---|---|---|
| Clip each wheel | 50 | 20 | 70 N, 88 % | 7.5 N m, 75 % |
| Moment first | 50 | 10 | 60 N, 75 % | 10 N m, 100 % |
| Force first | 50 | 30 | 80 N, 100 % | 5 N m, 50 % |

for the moment of measurement, a reflector delay before the scan arrives. Chapter 7 estimates the gyro bias in a multiplicative extended Kalman filter, and without the bias state its camera drifts 2.35° while the Sun blinds the star tracker. Chapter 10 moves each camera report on by the belt travel since its capture, and an arm aimed at the last report closes behind the product by $64.4 + 148.2L$ in millimeters. Chapter 11 predicts a flight with drag and spin from frames it must stop watching before it is sure, and Chapter 8 flies on mean elements, averages that no instrument reports. It sizes its avoidance burns on linear relative motion, in which a burn moves the satellite along its track by about three times the velocity change for every second that follows [15].

Part of a measured motion can be real and still not an error. A vessel riding over a 4 m sea moves 2.4 m at an eleven-second period without leaving its station, and a foiling boat can follow long waves or let short ones pass under it. The crossover then decides what the loops are shown. Chapter 6 carries a wave resonator in its observer, and the same sea and tuning cost 327.5 MN of thruster command travel with the raw measurement shown to the loops and 72.0 MN with the full observer [2]. Chapter 5 chooses between platforming and contouring by its observer's crossover.

### 1.4.7 Allocation

The motion loops ask for a total force and a moment, and the actuators must supply them. For C-1 the two wheels give Equation (1.12).

$$F = F_R + F_L, \quad M = \frac{b}{2}\left(F_R - F_L\right) \qquad \Longrightarrow \qquad F_R = \frac{F}{2} + \frac{M}{b}, \quad F_L = \frac{F}{2} - \frac{M}{b} \tag{1.12}$$

A demand of 80 N with 10 N m asks 60 N of the right wheel, which can give 50 N. Table 1.1 compares three ways to cut the demand. Clipping each wheel loses some of both. Serving the moment first keeps the turn and gives up speed, and serving the force first keeps the speed and gives up half the turn. A path follower that loses its turn leaves its path, so a moving machine usually serves its attitude or its turn first.

With more actuators than demands the split is a choice. The weighted least-norm solution of Equation (1.13) spreads a demand $\tau$ over the actuator commands $\mathbf{u}$ through the configuration matrix $B$ [27].

$$\mathbf{u} = W^{-1}B^{\top}\left(BW^{-1}B^{\top}\right)^{-1}\tau \tag{1.13}$$

A large weight in $W$ makes an actuator expensive, and a failed one is given a weight so large that the others take its share with no rule written for the case.

Chapter 6 solves Equation (1.13) for four thrusters, six forces and three demands, weighting each machine by the inverse square of its rating. Chapter 3 serves the tilt before the total thrust, so a rotor group at its limit costs altitude and never attitude. Chapter 4 asks the wing first and gives the rotors the rest, and the wing's authority grows with the square of the airspeed. Chapter 7 has an actuator that acts in only two directions: a magnetic coil makes a torque across the Earth's field and none along it, so over an orbit an excess of wheel momentum along the orbit normal decays with a time constant of 15.24 min and one in the orbit plane with 29.5 to 29.6 min.

> ✦ DID YOU KNOW? **Thirty-Eight Millinewtons From Five Hundred Watts**
>
> Electric thrusters generally give less than 75 mN of thrust per kilowatt, so a small spacecraft that can feed one 500 W may get no more than 38 mN. Such a thruster may fire for hundreds or thousands of hours where a chemical one fires for seconds or minutes [9]. Chapter 8 flies one.

### 1.4.8 Contact and Impedance

Once the tool touches the part, the part fixes some of its coordinates. The controller can then only choose how hard to push along them, and its stiffness turns every position error into force. Impedance control makes that stiffness a design choice, as shown in Equation (1.14) [12].

$$F = K\,(x_r - x) + D\,(\dot{x}_r - \dot{x}), \qquad D = 2\zeta\sqrt{Km} \tag{1.14}$$

Suppose C-1 docks on a charger misplaced by 5 mm. A stiff position servo of 20 kN/m pushes the misplacement with 100 N, the whole drive force. A spring of 500 N/m pushes with 2.5 N, and with $\zeta = 0.7$ it needs 221 N s/m of damping empty. The desired contact force is set by programming the reference past the contact, 40 mm for 20 N.

No controller acts during the landing itself. A mass $m$ meeting a stiff surface $k_s$ at speed $v$ is stopped within a quarter period $(\pi/2)\sqrt{m/k_s}$, and its peak force is Equation (1.15).

$$F_{\text{peak}} \approx v\sqrt{k_s\,m} \tag{1.15}$$

Against a 20 kN/m bumper the empty cart lands with 1000 N per m/s, 20 N at 0.02 m/s and 100 N at 0.1 m/s, and the impedance spring, 40 times softer, adds little in the 79 ms it lasts. The approach speed is the only defense.

A force loop closes on the measured force by moving the reference. With an integral law $\dot{x}_r = k_i(F_{SP} - F_m)$ against a surface much stiffer than the spring, the force follows $F \approx K(x_r - x_c)$, and the loop is an integrator of gain $g = Kk_i$. A measurement delay $\tau_d$ takes $g\tau_d$ radians of phase at the crossover $g$, so the loop holds while Equation (1.16) does.

$$\tau_d < \frac{\pi}{2\,K\,k_i} \tag{1.16}$$

With $k_i = 0.01\,\text{m}/(\text{N s})$ C-1 has $g = 5\,\text{s}^{-1}$, a time constant of 0.2 s and a delay margin of 0.314 s, which a simulation of the delayed loop reproduces.

Chapter 13 is this subsection on a peg. Its stiff servo damages the part beyond about 0.22 mm of fixture offset, and at 0.4 mm a 3 N/mm spring inserts the peg at 13.7 N. Its landings cost 1.35 N per mm/s, against 1.44 N from Equation (1.15), and its force loop, whose gain $K_{c,F}K_z$ is 1.0, loses its phase margin at 18.7 ms of delay. Chapter 14 holds a force on tissue that relaxes and creeps, through an estimate biased by the friction of the port.

### 1.4.9 Safety Monitoring

A machine that may meet an obstacle must be able to stop before it. After a reaction time $t_r$ and with a braking capability $A_b$, the stopping distance and the fastest speed that stops within a clear distance $d$ are Equation (1.17).

$$d_s = v\,t_r + \frac{v^2}{2A_b}, \qquad V(d) = A_b\left[\sqrt{t_r^2 + \frac{2d}{A_b}} - t_r\right] \tag{1.17}$$

With $t_r = 0.2\,\text{s}$ the loaded cart needs 0.70 m to stop from 1 m/s and the empty one 0.45 m. The limit is only as good as $A_b$. A cart that believes its empty braking while carrying the payload is allowed 1.07 m/s with an obstacle 0.5 m ahead. It needs 0.79 m to stop and strikes at 0.76 m/s. Believing the loaded braking allows 0.82 m/s, which stops in 0.50 m. Figure 1.6 shows both limits.

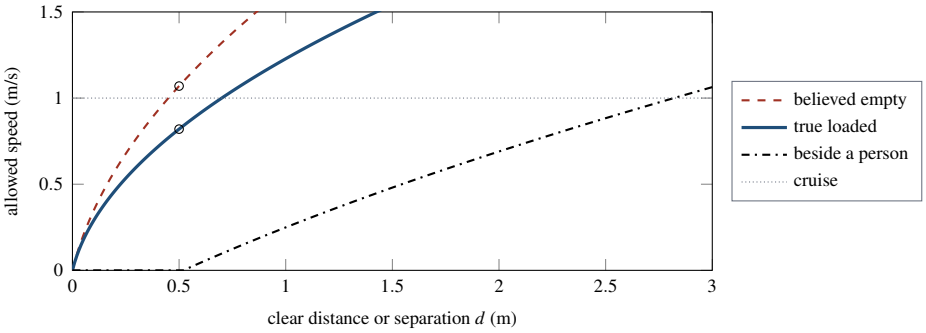



**Figure 1.6** Speed Limits of C-1 From Equation (1.17) With the Braking It Believes Empty (Dashed) and Has Loaded (Thick), and From Equation (1.18) for a Person Walking In at 1.6 m/s (Dash-Dotted). The Circles Mark the Two Limits for an Obstacle 0.5 m Ahead. The Dotted Line Is the Cruise Speed.

A person does not stand still. The protective separation distance adds the distance a person walking at $v_h$ covers while the machine reacts and brakes, with the braking time $t_s = v/A_b$ and an intrusion allowance $C$, as shown in Equation (1.18) [28].

$$S_p = v_h\,(t_r + t_s) + v\,t_r + \frac{v^2}{2A_b} + C \tag{1.18}$$

At 1 m/s, with $v_h = 1.6\,\mathrm{m/s}$ and $C = 0.2\,\mathrm{m}$, the loaded cart needs 2.82 m, and 0.52 m at rest. The braking time enters twice, once as the cart's braking distance and again as time in which the person keeps walking. Leaving out the person's walk during the stop gives 1.22 m, short by the 1.6 m a person covers in that second. Solving Equation (1.18) for $v$ gives the speed law plotted in Figure 1.6, 0.48 m/s at 1.5 m.

A monitor acts through samples. A scanner that reports every 40 ms sees a person at 1.6 m/s come 64 mm closer between reports, so the speed law needs a margin, and a restart needs a hysteresis or the machine stops and starts on every scan. A monitor can also be checked: replaying each decision on the true state and the true braking says whether a person could have been reached, which a monitor that sees only its own estimates cannot say.

> ✦ DID YOU KNOW? **Twenty-Two Meters of Sight**
>
> The desert robot of 2005 found obstacles with its lasers out to about 22 m, enough to avoid them reliably at up to 25 miles an hour. When its camera found no drivable road within 40 m, it slowed to that speed [13]. Its speed was set by the space it could see.

Chapter 9 sets its speed from Equation (1.17) on visible space and believed braking. With a 1000 kg pallet planned on the empty robot's $2.26\,\mathrm{m/s^2}$ it hits a parked cart at 0.74 m/s, while payload-aware braking stops 0.25 m short. Chapter 15 turns Equation (1.18) into a speed law with a margin and a hysteresis that cuts 17 protective stops in a 90 s case to 2 at the cost of one cycle, and its truth audit replays every decision on a copy of the true arm. At 1 m/s with a 3 kg part it needs 1448 mm. Chapter 14 keeps the instrument out of forbidden regions with virtual fixtures, and Chapter 4 earns each change of mode on measured airspeed and lift margin rather than on a clock.

### 1.4.10 Completing a Task Under Uncertainty

Every subsection above ends at the same point: the task succeeds only if what the controller believes is close enough to what is true. There are four ways to close the gap. A controller can be told beforehand, through a load sheet, a recipe mass or a map, and feed the value forward. It can estimate

**Table 1.2** For Each Chapter, the Task, What Is Uncertain and How the Controller Learns of It or Guards Against It

| Ch. | Task | Uncertain | Learned or guarded by |
|---|---|---|---|
| 2 | Hold the lane at any speed | Speed, curvature, linkage misalignment, rack lag | Speed-divided law, curvature feedforward, MPC |
| 3 | Fly a route with stops | Load, balance, wind among buildings | Load-sheet feedforward, scheduled gains, disturbance observer, dispatch margin |
| 4 | Hand the weight from rotors to wing | Airspeed, lift margin, a failed motor | Wing-first allocation, gates on measured airspeed and lift margin |
| 5 | Fly between slam and ventilation | Waves, sensor loss | Complementary observers, platform or contour, heave-rate cascade, fallback |
| 6 | Hold station on thrusters | Wave motion, environmental force, a failed thruster | Observer with bias and wave states, weighted allocation, heading supervisor |
| 7 | Point the camera through a pass | Gyro bias, stored momentum, a failed wheel | Filter with a bias state, magnetic unloading, eigenaxis limit |
| 8 | Keep altitude and avoid debris | Drag, debris covariance, thruster output | Mean elements, predictive planner, collision probability, burn verification |
| 9 | Deliver pallets among people | Position, braking with a load, unseen space | Fused estimate, payload-aware envelope, speed for unseen space |
| 10 | Grasp products on a belt | Report age, belt speed, payload | Time-stamped prediction, interception at belt speed, payload in the arm model |
| 11 | Return the ball | Drag, spin, time left | Filter on a flight model, commit rule, minimum-jerk swing |
| 12 | Weld a drifting seam | Torch height, seam ahead of the arc | Arc self-regulation, height through the current, distance-keyed preview |
| 13 | Insert a peg | Hole position, friction, late force signal | Impedance, contact detection and search, force loops |
| 14 | Work through a port on moving tissue | Tremor, delay, organ motion, tissue creep | Scaling and filtering, virtual fixtures, harmonic predictor, force loop |
| 15 | Work beside a person | The person's motion, latency, braking | Separation budget, speed law with margin and hysteresis, truth audit |

the unknown online, with an observer, a filter or an adaptive predictor. It can bound the unknown and keep a margin, as a speed law, a transition gate or a probability threshold does. Or it can commit to a prediction before the outcome is known, because the deadline will not wait. Each choice spends something: error while an estimate converges, time when a margin slows the machine, or risk when a prediction is wrong. The chapters measure these costs, often as an error budget that splits a result among its causes.

Table 1.2 lists, for each chapter, the task, what is uncertain and how the controller learns of it or guards against it. Chapter 16 adds the measured results and compares how each system completed its task under uncertainty.

## 1.5 Instruments and Final Elements in This Volume

Figure 1.7 lists what the controllers of each chapter read and what they move. Nine chapters close their loops on the estimate of a filter or an observer, more than read any single kind of sensor, and a gyro or an accelerometer is the commonest instrument, in six. An electric drive on a wheel, a joint or a rotor is the final element of ten chapters.

An optical encoder chops a light beam with a slotted wheel and sends a train of pulses [29]. With two trains a quarter period apart, the one that leads tells which way the wheel turns, so a counter can add or subtract [29]. Encoders read the joints of four arms in this volume and the wheels of Chapter 9. They count distance, which is why dead reckoning drifts in Section 1.4.6.

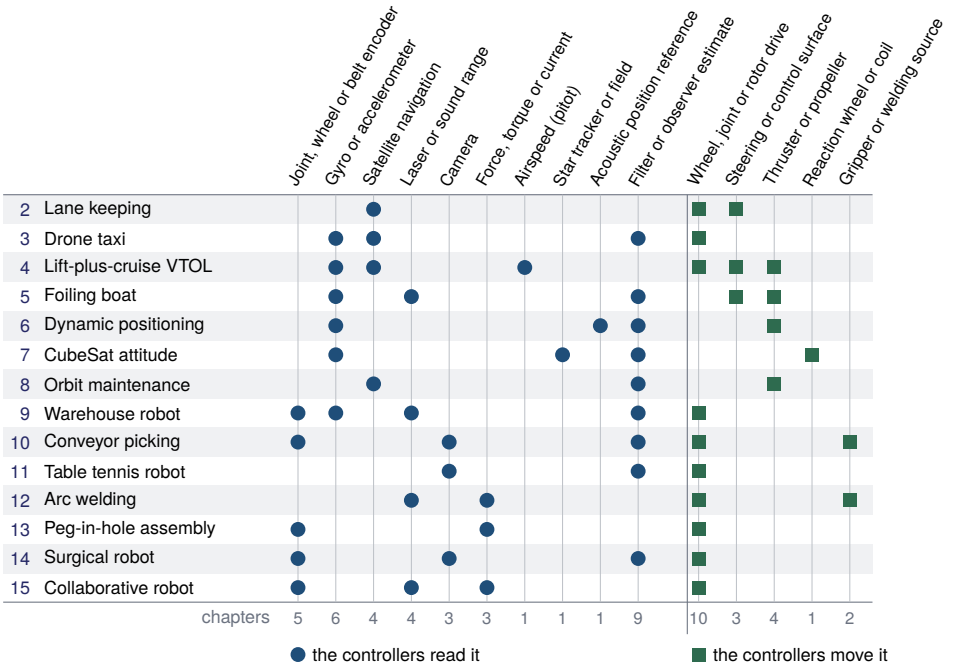



**Figure 1.7** Sensors and final elements by chapter: a round mark where the chapter's controllers read the device and a square mark where they move it, as its variables table and its technology section name them. A value computed from other readings is not marked, and the last column of reads marks a loop closed on the estimate of a filter or an observer. Notice that the estimate outnumbers every sensor.

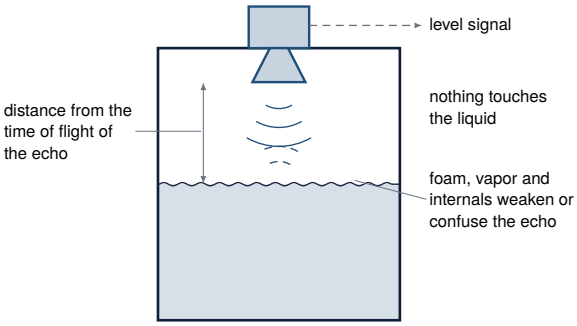



**Figure 1.8** Principle of ranging by time of flight, drawn as a radar level gauge. Notice that the distance is half the round trip of the wave, so a timing error counts half, the rule by which the laser scanners of Chapter 9 and Chapter 15 measure the space around them.

A satellite receiver is now the main means of finding the orbit of a spacecraft in low orbit, and it needs at least four navigation satellites in view [9]. On the ground it is fused with gyros, accelerometers and wheel speeds, as on the desert robot of Section 1.2 [13].

A laser scanner finds range from the time of flight of light. A clock time of 1 ns is 150 mm of range, and the accuracy depends on the reflectivity of the target and the angle at which the beam strikes it [30]. Figure 1.8 shows the principle on a radar gauge, which times a radio wave the same way.

A wrist force sensor measures all six components of force and torque in one compliant structure. Strain gauges give it high accuracy but need careful wiring, pick up electrical interference and often

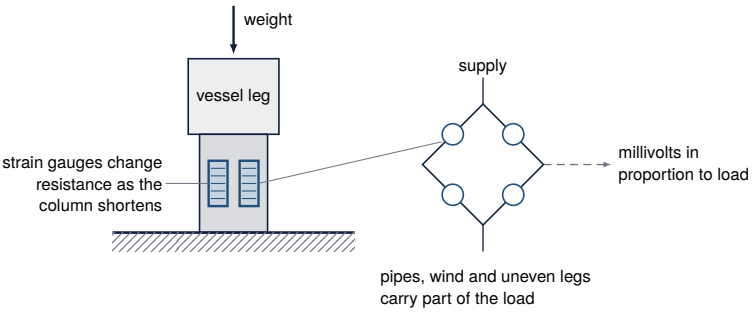



**Figure 1.9** Principle of a strain-gauge load cell. Notice that a load that reaches the support by another path is not measured. A wrist sensor puts such a compliant body at the end of the arm, between the arm and its tool (Chapter 13).

| Layer | What it does in this volume | Where it runs | How often |
|---|---|---|---|
| **People who decide** | a driver holds the wheel, a surgeon moves the console, operators screen an orbit | **By hand**, at the wheel, at the console or in operations | as needed; an orbit at least daily |
| **Mission and supervisor** | chooses modes and rotor states, plans burns, sets the heading | **Flight software** or the robot's program | 10 ms to 6 h in the apps |
| **Safety monitor** | stopping envelopes, speed and separation, transition gates | safety-rated sensing, or a computer that sends slow and stop | with each scan; 20 ms to 0.1 s in the apps |
| **Planning and navigation** | plans paths and trajectories, fuses the sensors into estimates | **Onboard computer** of the vehicle or the robot | 5 ms to 10 s in the apps |
| **Motion loops and allocation** | cascades of position, speed, attitude and force, shared among actuators | **Embedded controller** or the robot controller's servos | milliseconds; 0.3 ms to 10 s in the apps |
| **Sensors and drives** | encoders, inertial units, receivers, scanners, cameras, motors and thrusters | on wires or the vehicle bus | reports from every 10 s to 200 Hz in the apps |

**Figure 1.10** The control layers of this volume, the platform that hosts each and how often each acts. Loop periods are from [32] and the app periods from the chapters' technology sections. Notice that the safety monitor sits above the loops it can stop, and a person above it.

couple one axis into another [31]. A strain gauge is bonded to a metal element that stretches and compresses with the force like a very stiff spring [29] (Figure 1.9).

Figure 1.10 places the volume's loops on their platforms. Supervisory and lower loops in industry run with cycle times from milliseconds to minutes [32]. The apps span a wider range, from the 0.3 ms force loops of Chapter 13 to the 6 h planner of Chapter 8, and a person sits above several of them.

## 1.6 The Simulator Framework

Every chapter of this book has an app that runs in a web browser, collected with the others at https://apmonitor.com/apps/. About 240 apps exist across the series. All but a few are built on one shared framework, so the engine, the PID algorithm, the trends and the data export behave the same way in every chapter. The exceptions are standalone studios, such as the cascade studio and the two Temperature Control Lab apps, which follow the same ideas in their own code. This section describes the framework once and states where the standalone apps differ. Table 1.3 lists what a reader meets.

**Table 1.3** What Every App Provides. Standalone Apps Are Described in Their Own Chapters.

| Element | What it does |
|---|---|
| Run, Pause, Reset | Start, hold and restart the run. Reset keeps the reader's settings. |
| Speed | Simulated time per second of real time. It changes the pace and never the result. |
| Manual and Auto | Hand each actuator to the reader or to the controller, without a bump. |
| Scenario list | Numbered cases, each with a briefing, a completion rule and a reload button. |
| Control panel | Setpoints, manual outputs, tuning, options and disturbances, at a Beginner or an Advanced level. |
| Trends | SP and PV above the MV for each loop, with limits, event marks and a cursor. |
| Scorecard | IAE, overshoot, settling time, MV travel and time at a limit for each loop. |
| Learn card | Predict, Try It, two hints and Explain, opened by the reader in that order. |
| Show Model | Equations, live variable values, editable parameters and assumptions. |
| Export CSV | Every logged sample of the run as a text file with a header row. |
| Instructions | A separate page with background, an operating guide and the model's limitations. |

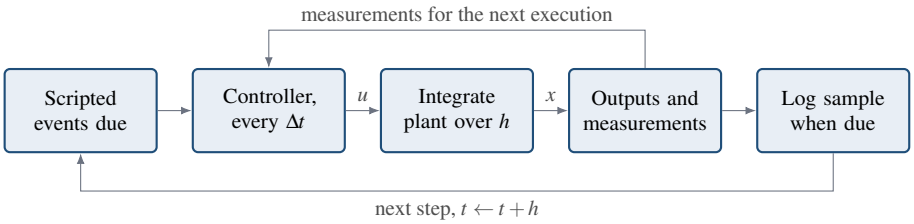



**Figure 1.11** Work of the Engine in One Integration Step

### 1.6.1 The Simulation Engine

Each app carries a plant model written as ordinary differential equations $\frac{dx}{dt} = f(x,u,d,p)$ in the states $x$, the inputs $u$, the disturbances $d$ and the parameters $p$. The engine integrates the model with the classical fourth-order Runge–Kutta method on a fixed step $h$ set for the fastest dynamics of the app. Some apps supply their own step, such as explicit Euler for a discontinuous right-hand side or an exact update for a model driven by events.

Three clocks run at separate rates. The plant advances every step $h$. The controller executes every control period $\Delta t$, equal to $h$ unless the app declares a slower controller, and holds its outputs between executions. The log takes one sample at a fixed interval of simulated time for the trends, the scorecard and the exported file. The order of work in one step is shown in Figure 1.11. The controller also executes once when a scenario is loaded, so every run starts from an initialized controller.

Drawing the screen is kept apart from these clocks. A speed of 5 means five units of the app's time base per second of real time. Speed, frame rate and pauses change how fast a run is shown and never what it computes, so a phone and a workstation produce the same trajectory and the same file.

Noise comes from a deterministic pseudorandom generator. Its seed is declared by the app, or by a scenario that needs its own noise record, and it is restored whenever a scenario is loaded or Reset is pressed. The framework gives the reader no seed setting, so *default seed* in this book means the seed the app ships with. A run repeats bit for bit when the same scenario is loaded and the same settings are made before Run. A change made by hand during a run lands on a different step each time, which is one reason numeric answers carry a tolerance. The standalone simulators draw noise from the browser's own generator, so a noisy run there repeats in its statistics and not sample for sample.

Run and Pause start and hold the clock. Reset restarts the run at $t = 0$, paused, and keeps the reader's settings: every setpoint, tuning value, option, model parameter, disturbance slider and mode the reader changed is put back before the plant and the controller are initialized. Scripted events replay, and a one-shot disturbance button pressed during the run is cleared. The reload button beside

the scenario list restores the scenario's own settings instead.

### 1.6.2 The Velocity-Form PID Algorithm

One PID implementation serves the PID loops of every framework app. It is the velocity (incremental) form: each execution adds the change in the proportional and derivative terms to a running sum, and the integral is a separate term, as shown in Equations (1.19) to (1.22) for execution $k$ and control period $\Delta t$.

$$e_k = a\,(\mathrm{SP}_k - \mathrm{PV}_k), \qquad P_k = K_c\,e_k, \qquad D_k = -a\,K_c\,\tau_D\,\frac{\mathrm{PV}_k^f - \mathrm{PV}_{k-1}^f}{\Delta t} \tag{1.19}$$

$$S_k = S_{k-1} + (P_k - P_{k-1}) + (D_k - D_{k-1}) \tag{1.20}$$

$$v_k = u_b + S_k + I_{k-1}, \qquad u_k = \mathrm{sat}(v_k) \tag{1.21}$$

$$I_k = I_{k-1} + \left[\frac{K_c}{\tau_I}\,e_k + \frac{u_k - v_k}{T_t}\right]\Delta t \tag{1.22}$$

where $a = \pm 1$ is the controller action, $\mathrm{PV}^f$ is the measurement after an optional first-order filter, $u_b$ is the output bias, $S$ is the running sum of proportional and derivative increments, $v$ is the output before limiting, $u$ is the output sent to the actuator, $I$ is the integral term and $T_t$ is the tracking time constant, equal to $\tau_I$ unless a loop sets another value. The function sat clamps $v_k$ to the actuator range and then applies a rate limit if the loop declares one.

The limit acts on the emitted output $u_k$ only. The sum $S_k$ is never clamped, so it equals $P_k + D_k$ and the output before limiting is the position form $u_b + K_c e_k + D_k + I_{k-1}$. Anti-windup touches the integral alone. While the output is limited, the last term of Equation (1.22) drives $I$ toward the value that reproduces the output applied, which is back-calculation [33]. With no integral action, Equation (1.22) is skipped. Clamping the running sum would instead discard each increment that crosses a limit and keep each one that returns, so symmetric noise would push the output off the limit even in a proportional-only loop.

The derivative acts on the measurement, so a setpoint step gives a proportional kick and no derivative kick. A change of $K_c$ acts on the whole error at the next execution, as in the position form. The action $a = +1$ suits a process whose PV rises with the MV, a reverse-acting controller in the ISA convention, and $a = -1$ gives the direct-acting one.

In Manual the controller keeps executing and sets $I_k = u_{\mathrm{man}} - u_b - S_k$, so the output it would send equals the manual value $u_{\mathrm{man}}$. The first automatic output differs from it by one period's change in $P$ and $D$. In Auto the engine copies each controller output to the manual station, so a switch to Manual is also bumpless. The standalone studios with PID loops use velocity-form code of their own, with the limit on the output and anti-windup on the integral alone, in which a gain change applies only to later increments.

### 1.6.3 The Predictive Control and Estimator Options

More than 100 apps offer model predictive control (MPC) as an option beside the structure they open with, which stays the default. Volume 9 develops the theory. In most of them the controller plans with a small linear model in discrete time as shown in Equation (1.23).

$$x_{k+1} = A\,x_k + B\,u_k + E\,w_k + c, \qquad y_k = C\,x_k \tag{1.23}$$

where $w$ is a measured or scheduled disturbance and $c$ is a constant left by linearization. The model comes from a step test at the design point, from the app's balances linearized there or again at every decision, or from the model the reader fits. It never reads the true states or parameters of the run, so a parameter the reader changes in Show Model is model mismatch, as on a real plant.

At each decision the controller minimizes the objective of Equation (1.24) over a horizon of $N$ steps.

$$J = \sum_{k=1}^{N} (y_k - r_k)^{\top} Q (y_k - r_k) + \sum_{j} \Delta u_j^{\top} R \Delta u_j + \sum_{i} \rho_i s_i^2 \tag{1.24}$$

where $r$ is the setpoint, $Q$ weights the tracking error, $R$ is the move suppression on each input change $\Delta u_j$ and $\rho_i$ is the price of a slack $s_i \geq 0$ that lets an output band be broken when it cannot be held. Actuator limits are hard constraints. The quadratic program is solved in the browser by a dual active-set method [34]. The first move is applied, and the plan is made again at the next decision.

The plan starts from an estimate of the model state. About half of these apps use a steady-state Kalman filter on the model with an added disturbance state, which absorbs what the model did not predict and removes steady offset without an integral term [35]. Others correct the model output with a filtered bias to the same end. About a dozen apps also offer moving horizon estimation (MHE), which solves the estimation problem over a window of recent samples and can hold bounds that a Kalman filter cannot, such as a concentration that is never negative [36].

With the MPC selected, the tuning fields of the loops it replaces leave the Control panel. In most of these apps the Advanced level shows up to three settings, each a multiple of the shipped value: the prediction horizon, the move suppression and the band price. The trend draws the accepted plan ahead of the present time, and a list of constraint prices shows which limits bind. A plan that is not optimal, not finite or outside its limits is rejected, and a fallback, in most apps the shipped controller, holds the actuator until a plan succeeds. The Model Fitting and Tuning Studio builds the same kind of controller from the reader's own data in its tab, MPC from the fit.

### 1.6.4 Scenarios, Progress and Data

An app teaches through scenarios, the numbered cases at the top of the Control panel, nine to thirteen in most apps. Loading one sets the model parameters, the controller settings, the disturbances, the mode, the initial state and any scripted events. The information button beside the list opens a briefing in four parts: what the case is, why it matters, what to do and one detail for the practicing engineer. The Learn card holds the lesson back, with Predict and Try It before the run, hints on request and Explain afterward.

Each scenario has a completion rule that the app judges from the run on the simulated clock: a run length, the scripted events passed, a setting made by the reader, a state held for a stated time or a target met. Choosing a case earns nothing. A completed case is marked in the list, and a badge row shows the count and a level, Explorer after one case, Practitioner at half and Completed at all. Progress is kept in the reader's browser with no account. A certificate of progress or of completion can be issued from it, as a record of self-paced work and not an assessed credential.

One case, usually the last, is the graded challenge. It states a target, such as an error band to hold through a sequence of upsets. Changing the model parameters or the disturbances voids the attempt in most apps, and a record made with Reveal suggested tuning says so.

Show Model opens the equations, live values of the variables, the editable parameters and the assumptions. The trends plot SP and PV above the MV for each loop, with actuator limits and marks at setpoint moves, disturbances, mode changes and scripted events. The scorecard reports for each loop the integral of the absolute error (IAE), overshoot, settling time, MV travel and share of time at a limit, with the app's own measures, and it does not combine them into one score. IAE and travel accumulate from Reset, so two runs are compared at the same clock reading.

Export CSV saves the log as a text file with a header row of column names that carry their units and one row per logged sample. In most apps the time column is in seconds, minutes or hours according to the run length. The log holds the most recent 60 000 samples unless the app sets another size. Apps with the MPC option begin the file with comment lines that start with #, which a script must skip.

An app exercise in this book names the scenario by its number and name as the app shows them, every setting that differs from the scenario's defaults, the run length in simulated time, the seed

(the default seed unless stated), what to read and where (a trend, a scorecard tile or a column of the exported file) and a tolerance on a numeric answer. Settings are made before Run unless the exercise gives a time.

## 1.7 Working With AI Agents While Studying

A reader of this book is likely to have an AI assistant within reach, and the apps make its work easy to test. This section gives the editor's view of how to use one while working through a chapter. In short, the assistant may do the algebra and the scripting, and the reader keeps the checking.

**What to delegate.** Tasks with a clear input and a result that can be tested are good candidates. An assistant can derive the linearization of a chapter's balance at an operating point and the transfer function that follows from it. It can write a short script that reads the exported CSV file and fits a first-order-plus-dead-time model to a step test. It can propose a first tuning from that fit with a named rule, convert units, explain a term in a model or outline a report. The assistant knows nothing about the app until it is told. Give it the equations and parameter values from Show Model, the column names of the exported file and the scenario that produced the data, as described in Section 1.6.4. Ask it to state its assumptions and the units of every result.

**What to check, and how.** An assistant returns a fluent answer whether or not the answer is right, so each result needs a test that it could fail. Four tests cover most of the errors a chapter will produce.

- *Units.* Every term of an equation carries the same units, and a process gain is in PV units per MV unit. A gain computed with the valve as a fraction is 100 times the gain of a loop whose valve is in percent.
- *Signs.* Decide from the physics which way the PV moves when the MV rises. Compare that with the sign of the fitted gain and with the controller action of Section 1.6.2.
- *Steady state.* Set the derivatives to zero. The model must return the operating point that Show Model reports, and a linearization must give zero deviation there.
- *A run on the app.* Load the named scenario at the default seed, enter the proposed numbers and run. The trend and the scorecard decide. A tuning that reads well on paper and holds the valve at a limit for half the run has failed.

A fitted model gets one more test: compare it with data it was not fitted to, such as a second step in the opposite direction. When a check fails, report the failure to the assistant with the evidence and ask for the cause, then check the correction the same way.

**How the exercises train the habit.** One exercise in each chapter is marked *Review the agent*. It presents a tuning, a model or a claim as an AI assistant might propose it, with a flaw, and asks the reader to find the flaw in about five minutes. The flaws are ordinary ones: a unit, a sign, a steady state that does not close, a result measured on the wrong scenario or at a different run length, or a limit that was ignored. Each is found with one of the tests above. The purpose is not distrust. An engineer who can say why an answer is right can hand an assistant much larger tasks and still stand behind the result. That larger picture, in which agents do most of the routine work and the engineer supervises and checks, is the subject of Chapter 16.

## 1.8 Chapter Roadmap

Table 1.4 lists the chapters with the lesson each adds to this one. The level says how much of Section 1.4 a chapter assumes. The time is an estimate for reading the chapter once and running each case of its app once, at about eight minutes a case.

Chapters 3, 9, 13 and 15 form a first course: a cascade told about its load, a speed set by braking, a contact set by stiffness and a stop budgeted against a person. They rest on Sections 1.4.3, 1.4.8

**Table 1.4** Roadmap of the Volume

| Ch. | Plant | The lesson | Level | Time |
|---|---|---|---|---|
| 2 | Car on a kinematic bicycle | Speed enters twice and cancels in the Stanley law. Lag, offset and the rack remain. | intermediate | 3 h |
| 3 | Passenger multirotor | Tell the loops the load and estimate the wind. The feedforward does the work. | introductory | 2.5 h |
| 4 | Lift-plus-cruise aircraft | Stop the rotors on measured lift margin, not on a clock. | intermediate | 2 h |
| 5 | Foiling boat | Choose which waves to reject. A cascade on heave rate contours. | intermediate | 2.5 h |
| 6 | Vessel on thrusters | Part of a measurement is real and not an error. Structure before tuning. | advanced | 3 h |
| 7 | CubeSat attitude | Wheels store every torque, and coils act only across the field. | advanced | 2.5 h |
| 8 | Satellite in low orbit | Flown on predictions and a probability, not on a miss distance. | advanced | 2 h |
| 9 | Pallet robot | Speed is set by believed braking and visible space. | introductory | 2 h |
| 10 | Picking arm at a belt | Move old reports forward in time and meet the product at its speed. | intermediate | 2 h |
| 11 | Table tennis arm | Commit before the prediction is certain, and swing within the time left. | intermediate | 2.5 h |
| 12 | Arc welding cell | A plant that regulates itself and a sensor that reads places. | intermediate | 2 h |
| 13 | Peg in a hole | Stiffness turns position error into force. A force loop has a delay margin. | introductory | 2 h |
| 14 | Surgical instrument | Each layer of the stack has a measurable cost. | advanced | 2.5 h |
| 15 | Collaborative arm | Stopping in time is a budget, checked against the truth. | introductory | 2 h |

and 1.4.9. Chapters 2, 4, 5, 10, 11 and 12 add path following, allocation, estimation and timing, and Sections 1.4.4 to 1.4.7 prepare them. Chapters 6, 7, 8 and 14 combine estimation, allocation and supervision on one machine and are best left until Section 1.4.10 is familiar. A reader interested in vehicles can read Chapters 2 to 9 in order, and a reader interested in arms Chapters 10 to 15.

**KEY POINTS**

- A moving machine reaches its task through geometry and inertia, and what matters is computed from estimates rather than read from one instrument.
- Each inner loop must be faster than the loop it serves, and a force the loops are not told about is opposed only through an error.
- A complementary filter uses each sensor where it is good, and a report taken some time ago describes where the machine was.
- Speed is bounded by the braking a machine believes it has and the space it can see, so the limit is only as good as that belief.

**Where else this applies.** Feedforward and cascade are the subject of Volume 3, allocation and overrides of Volume 4, estimation of Volume 5 and predictive control of Volume 9.

## References

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, ser. Advanced Textbooks in Control and Signal Processing. London: Springer, 2009.

[2] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Chichester, UK: Wiley, 2011.

[3] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. New York: Springer, 2014.

[4] H. L. Hazen, "Theory of servo-mechanisms," *Journal of the Franklin Institute*, vol. 218, no. 3, pp. 279–331, 1934.

[5] R. J. Norcross, R. V. Bostelman, and J. A. Falco, "Automated guided vehicle bumper test method development," National Institute of Standards and Technology, Gaithersburg, MD, NIST Interagency/Internal Report 8029, 2015.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[7] Deep Sea Drilling Project, "Dynamic positioning," Deep Sea Drilling Project, Tech. Rep., 1971.

[8] M. D. Holley, W. L. Swingle, S. L. Bachman, C. J. LeBlanc, H. T. Howard, and H. M. Biggs, "Apollo experience report: Guidance and control systems: Primary guidance, navigation, and control system development," National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, Houston, TX, NASA Technical Note TN D-8227, May 1976. [Online]. Available: https://ntrs.nasa.gov/citations/19760016247

[9] S. V. Weston, C. D. Burkhard, J. M. Stupl, R. L. Ticknor, B. D. Yost, R. A. Austin, P. Galchenko, L. K. Newman, and L. Santos Soto, "State-of-the-art small spacecraft technology," National Aeronautics and Space Administration, Ames Research Center, Moffett Field, CA, Technical Publication NASA/TP-20250000142, Jan. 2025. [Online]. Available: https://ntrs.nasa.gov/citations/20250000142

[10] H. Makino, "Development of the SCARA," *Journal of Robotics and Mechatronics*, vol. 26, no. 1, pp. 5–8, 2014.

[11] M. D. Maisel, D. J. Giulianetti, and D. C. Dugan, *The History of the XV-15 Tilt Rotor Research Aircraft: From Concept to Flight*, ser. Monographs in Aerospace History. Washington, DC: National Aeronautics and Space Administration, NASA History Division, 2000, no. 17, nASA SP-2000-4517, https://ntrs.nasa.gov/citations/20000027499.

[12] N. Hogan, "Impedance control: An approach to manipulation: Part I, theory," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–7, 1985.

[13] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[14] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, 2009.

[15] W. H. Clohessy and R. S. Wiltshire, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.

[16] D. B. D'Ambrosio, S. Abeyruwan, L. Graesser, A. Iscen, H. Ben Amor, A. Bewley, B. J. Reed, K. Reymann, L. Takayama, Y. Tassa *et al.*, "Achieving human level competitive robot table tennis," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 74–82.

[17] Y. S. Kwoh, J. Hou, E. A. Jonckheere, and S. Hayati, "A robot with improved absolute positioning accuracy for CT guided stereotactic brain surgery," *IEEE Transactions on Biomedical Engineering*, vol. 35, no. 2, pp. 153–160, 1988.

[18] G. Campion, G. Bastin, and B. D'Andréa-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, 1996.

[19] B. Wie and P. M. Barba, "Quaternion feedback for spacecraft large angle maneuvers," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 3, pp. 360–365, 1985.

[20] W.-H. Chen, J. Yang, L. Guo, and S. Li, "Disturbance-observer-based control and related methods: An overview," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1083–1095, 2016.

[21] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[22] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *The Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[23] T. Kröger and F. M. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–111, 2010.

[24] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in *2007 American Control Conference*, 2007, pp. 2296–2301.

[25] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[26] W. T. Higgins, "A comparison of complementary and Kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-11, no. 3, pp. 321–325, 1975.

[27] T. A. Johansen and T. I. Fossen, "Control allocation: A survey," *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.

[28] J. A. Marvel and R. Norcross, "Implementing speed and separation monitoring in collaborative robot workcells," *Robotics and Computer-Integrated Manufacturing*, vol. 44, pp. 144–155, 2017.

[29] T. R. Kuphaldt, *Lessons in Industrial Instrumentation*, version 2.33 ed. Open textbook, Creative Commons Attribution 4.0 license, ibiblio.org/kuphaldt/socratic/sinst, 2024.

[30] W. C. Stone, M. Juberts, N. Dagalakis, J. Stone, and J. Gorman, "Performance analysis of next-generation LADAR for manufacturing, construction, and mobility," National Institute of Standards and Technology, Gaithersburg, MD, NIST Interagency/Internal Report 7117, 2004.

[31] S. Y. Lee, J. Y. Sim, D.-Y. Seok, Y. B. Kim, J. Shim, U. Kim, and H. R. Choi, "A redundant-sensing-based six-axis force/torque sensor enabling compactness and high sensitivity," *Sensors*, vol. 26, no. 3, p. 871, 2026.

[32] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, "Guide to operational technology (OT) security," National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-82r3, 2023.

[33] K. J. Åström and L. Rundqwist, "Integrator windup and how to avoid it," in *Proc. 1989 American Control Conference*, 1989, pp. 1693–1698.

[34] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, 1983.

[35] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model-predictive control," *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[36] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE Transactions on Automatic Control*, vol. 48, no. 2, pp. 246–258, 2003.

## 1.9 Exercises

**1.1** Concept check · 2 min

A joint servo forms its error as $e = a\,(\mathrm{SP} - \mathrm{PV})$ with a positive gain, where the PV is the measured joint angle. A positive torque turns the joint in the positive direction, so the torque must rise when the angle falls short of its reference. Which setting and name follow the convention of this book?

(a) $a = -1$, direct acting, because the torque rises with the error

(b) $a = +1$, reverse acting, because the torque falls as the angle rises

(c) $a = +1$, direct acting, because a positive error gives a positive torque

(d) $a = -1$, reverse acting, because the angle lags the reference

**1.2** Theory and calculation · 6 min

A mobile robot of 80 kg has an inner speed loop $F = K_v(v_r - v)$ with $K_v = 400\,\mathrm{N\,s/m}$ and an outer position loop $v_r = K_x(x_r - x)$ with $K_x = 1\,\mathrm{s}^{-1}$.

(a) Find the inner time constant and the damping ratio of the position loop. *Checkpoint:* 0.20 s and 1.12.

(b) The robot picks up 80 kg. Find the new damping ratio, and the speed gain that restores the old one. *Checkpoint:* 0.79, and 800 N s/m.

**1.3** Theory and calculation · 5 min

A gantry axis moves 0.8 m with a speed limit of 0.5 m/s and an acceleration limit of $1\,\mathrm{m/s^2}$. Find the shortest quintic (minimum-jerk) move and the trapezoidal move, and say which limit sets the quintic.

**1.4** Theory and calculation · 6 min

A robot follows a path with the speed-divided law $\omega = -k_h[\psi_e + \arctan(k_e y/v)]$, with $k_h = 4\,\mathrm{s}^{-1}$ and $k_e = 1\,\mathrm{s}^{-1}$, and without the term $v\kappa$.

(a) Find the poles of the linearized lateral loop. *Checkpoint:* a double pole at $-2\,\mathrm{s}^{-1}$.

(b) Find the steady offset on a 10 m radius at 2 m/s and at 4 m/s, and the feedforward that removes it at 2 m/s. *Checkpoint:* 0.10 m, 0.40 m, 0.2 rad/s.

**1.5** Theory and calculation · 5 min

A vehicle reacts in 0.3 s and brakes at 1.5 m/s$^2$. Find the fastest speed at which it can stop within 2 m, check it, and repeat for a braking capability of 0.75 m/s$^2$.

**1.6** Theory and calculation · 6 min

A collaborative arm moves its tool at 0.5 m/s and brakes at 2 m/s$^2$. Its reaction time is 0.25 s and its intrusion allowance 0.1 m, and a person may walk in at 1.6 m/s. Find each term of the protective separation distance and the total, and the total if the person's walk during the stop is left out.

**1.7** Theory and calculation · 5 min

A differential-drive robot with a track of 0.6 m and wheels limited to 40 N each is asked for a total force of 60 N and a moment of 12 N m. Find the wheel forces it asks for, then the force and moment it delivers if it serves the moment first and if it clips each wheel.

**1.8** Concept check · 3 min

A first-order complementary filter fuses wheel odometry that reads 3 % high with a position fix, at a crossover of 0.5 rad/s, on a robot moving at a steady 1.5 m/s. Which statement is right?

(a) The estimate lags 0.045 m, and a higher crossover removes it

(b) The estimate has no steady error, because the fix corrects it

(c) The estimate leads 0.09 m, and only a perfect odometer removes it

(d) The estimate leads 0.09 m, and a bias state removes it

**1.9** Theory and calculation · 8 min

A 2 kg end effector carries a peg.

(a) It lands at 10 mm/s on a part of stiffness 5 N/µm. Estimate the peak force, and the force a 3 N/mm spring and a 200 N/mm servo need to correct a 0.4 mm misalignment. *Checkpoint:* 31.6 N, 1.2 N, 80 N.

(b) An integral force law $\dot{x}_r = k_i(F_{SP} - F_m)$ with $k_i = 0.002\,\mathrm{m/(N\,s)}$ acts through a spring of 2000 N/m on a much stiffer part. Find the loop gain, the time constant and the largest measurement delay it tolerates. *Checkpoint:* 4 s$^{-1}$, 0.25 s, 0.39 s.

**1.10** Concept check · 3 min

A camera reports a product on a belt moving at 0.4 m/s with a latency of 0.12 s, and the gripper takes 0.08 s to close after it arrives. The arm is aimed at the reported position. What happens, and what corrects it?

(a) It closes 80 mm behind, and predicting from the time stamp corrects it

(b) It closes 48 mm behind, and a faster camera alone corrects it

(c) It closes 32 mm behind, and a faster gripper alone corrects it

(d) It closes on the product, because the arm tracks the camera