Institut für Intermodale
Transport- und
Logistiksysteme

Technische
Universität
Braunschweig

Control Engineering 2
(Regelungstechnik 2)

Lecture Notes

Jürgen Pannek

June 28, 2026

Jürgen Pannek
Institute for Intermodal Transport and Logistic Systems
Hermann-Blenck-Str. 42
38519 Braunschweig

FOREWORD

During summer term 2026 I give the lecture to the module *Control Engineering 2 (Regelungstechnik 2)* at the Technical University of Braunschweig. The notes utilized within the lecture are an updated and combined version of Control Engineering 2 and Control Engineering 3, which have been realigned to form a common string of contents. As the lecture notes are a living document, I will integrate remarks and corrections throughout the summer term.
The aim of the module is to provide participating students with knowledge of terms of system theory and control engineering. Moreover, students shall be enabled to understand complex control structures, apply control schemes and analyze control systems. After successfully completing the module, students shall additionally be able to apply the discussed methods within real life applications and be able to assess results.
To this end, the module will tackle the subject areas

- System theory and Modeling,
- Methods and Algorithms, and
- Stability and Control Design

for complex and networked linear as well as nonlinear systems. In particular, we discuss

- Frequency domain
  - Modeling of complex control loops, cascade and disturbance control
  - Bang-bang and double-setpoint control
  - Multi-input multi-output systems
- Time domain
  - Nonlinear control systems, Comparison and Lyapunov functions, controllability, observability

- Optimal control and filtering, LQ control, H_2 control, H_∞ control, Kalman filter

within the lecture and support understanding and application within the tutorial and laboratory classes. The module itself is accredited with 5 credits with an add-on of 2 credits if the requirements of the laboratory classes are met.
During the preparation of the lecture, I utilized the books of Jan Lunze [6–8] as well as AI tools for improving readability.

Literature for further reading

- Linear part
  - FÖLLINGER, O.: *Regelungstechnik*. 13. überarbeitete Auflage. VDE-Verlag, 2022
  - LUNZE, J.: *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*. 11. überarbeitete und ergänzte Auflage. Springer, 2016
  - LUNZE, J.: *Automatisierungstechnik*. 5. Auflage. DeGruyter, 2020
  - UNBEHAUEN, H.: *Regelungstechnik I*. Vieweg / Teubner, 2007
- Nonlinear part
  - Stability and observability
    * SONTAG, E.D.: *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer, 1998
    * HINRICHSEN, D. ; PRITCHARD, A.J.: *Mathematical system theory I: Modeling, state space analysis and robustness*. Springer, 2010
    * ISIDORI, A.: *Nonlinear Control Systems*. 3rd edition. Springer, 1995
  - LQR, H_2 and H_∞ control
    * ANDERSON, B.D.O. ; MOORE, J.B.: *Optimal control: linear quadratic methods*. Courier Corporation, 2007
    * SKOGESTAD, S. ; POSTLETHWAITE, I.: *Multivariable feedback control: analysis and design*. John Wiley & Sons, 2005
    * KALMAN, R.E.: Contributions to the theory of optimal control. In: *Boletin de la Sociedad Matematica Mexicana* 5 (1960), No. 2, pp. 102–119

Contents

List of Tables

List of Figures

List of Definitions and Theorems

LIST OF ALGORITHMS

CHAPTER 1

COMPLEX CONTROL LOOPS

Single-loop feedback control is the basic control structure and we studied it in Control Engineering 1 extensively. We compare one output with one reference, and the resulting error is fed into one controller. Many technical systems, however, provide more information than a single output signal. An electric drive, for example, does not only have a position. It also has a speed, a torque, and an armature current. A production line does not only have a final product quality, it also has intermediate temperatures, conveyor speeds, forces, and material tensions. A vehicle function does not only have a desired trajectory, it also has yaw rate, lateral acceleration, wheel speeds, and estimated disturbances such as road grade or crosswind.
Within this chapter, our aim is to to explain how such additional signals can be used in a structured way. The central idea is not to replace feedback but to complement it by modifying the loop. In line with standard notation in control engineering, we utilize the following notation:

Notation 1.1 (Signals)
The reference signal is denoted by w, the control input by u, the measured output by y, and the disturbance by d. Moreover, t denotes time and s the Laplace variable frequency. Additionally, the Laplace transform of a signal v is denoted by $\hat{v}$.

To distinguish between particular transfer functions G, we utilize subscript indicators.

Notation 1.2 (Transfer functions)
We denote plant transfer function by $G_S(s)$, feedback-controller transfer function are respresented by $G_R(s)$, a feedforward-controller transfer function is given by $G_F(s)$, a prefilter by $G_P(s)$, and $G_Z(s)$ represents a disturbance controller.

Moreover, we use the following standing assumption unless stated otherwise:

Assumption 1.3
All transfer functions are scalar, proper and real-rational.

To illustrate the approaches, we will use three recurring application threads. For a complete consideration, a direct-current motor with a measurable or estimable load torque will serve as the main running example. To allow for additional insights with regards to interpretation in other areas of applications, we consider two secondary examples of a production line with internal variables such as web tension, temperature, and conveyor speed, and an AD/ADAS vehicle function in which low-level actuator dynamics, vehicle motion, and environmental disturbances interact.

Task 1.4 (Running example: DC motor with load torque)
Consider an armature-controlled DC motor described by

$$L_{\mathrm{a}}\dot{i}_{\mathrm{a}}(t) + R_{\mathrm{a}}i_{\mathrm{a}}(t) + K_{\mathrm{e}}\omega(t) = u_{\mathrm{a}}(t), \tag{1.1}$$

$$J_{\mathrm{m}}\dot{\omega}(t) + b_{\mathrm{m}}\omega(t) = K_{\mathrm{t}}i_{\mathrm{a}}(t) - d_M(t), \tag{1.2}$$

$$\dot{\varphi}(t) = \omega(t). \tag{1.3}$$

Assume that armature voltage u_{a}, armature current i_{a}, angular velocity ω, angular position φ, and possibly the load torque d_M are available as signals. Identify the physical meaning of these signals, rank the corresponding variables according to their expected dynamic speed, and state through which equation the load torque first affects the motor motion.

Solution to Task 1.4: The armature voltage u_{a} is the manipulated electrical input. The armature current i_{a} is proportional to electromagnetic torque through $K_{\mathrm{t}}i_{\mathrm{a}}$. Its dominant time scale is electrical and is approximately $T_{\mathrm{e}} = L_{\mathrm{a}}/R_{\mathrm{a}}$ if the speed-induced voltage is treated as a coupling term. The angular velocity ω is a mechanical state with dominant time scale $T_{\mathrm{m}} = J_{\mathrm{m}}/b_{\mathrm{m}}$. The angular position φ is the integral of ω, so it reacts more slowly to changes in the electrical input. Thus, the expected order from fast to slow is

$$i_{\mathrm{a}} \quad \text{before} \quad \omega \quad \text{before} \quad \varphi.$$

The load torque d_M enters the mechanical equation (1.2). Therefore, it first changes angular acceleration and speed before its effect accumulates in the position.

From Task 1.4 we observe that physical signal interpretation may be separated from controller design. This example allows us to decide which loop structure is appropriate given its properties, i.e. what is measured, how fast may each variable react, and where do disturbances enter. In particular, we will use inner measurements for cascade control in Section 1.1, shape desired trajectories by feedforward and prefilters in Section 1.2, and compensate measurable disturbances in Section 1.3.

1.1 Cascade Control

The first more complex loop we consider is the so called *cascade*. Cascade control is useful when the actuator affects several measurable variables on different time scales. Instead of closing only one loop around the final output, the control designer closes an inner loop around a fast variable and then uses the closed inner loop as part of the plant seen by the next outer loop. In the DC motor, current control shapes the torque actuator, speed control shapes the motion, and position control uses the shaped speed loop for accurate tracking.
Figure 1.1 shows the basic idea of cascaded loops. The signal y_{j-1} is the inner controlled variable, while y_j is the outer controlled variable.

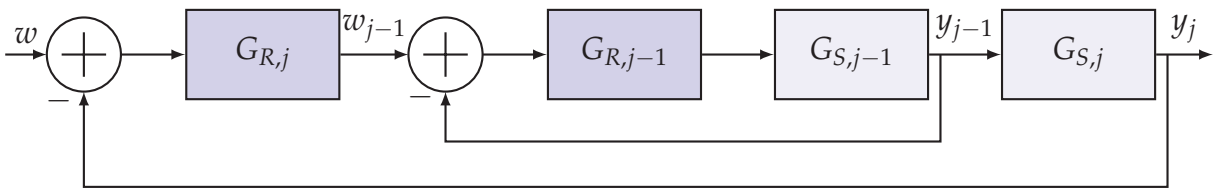


Figure 1.1: Block diagram of a two-loop cascade control.

Remark 1.5 (Time-scale separation)
The inner loop should usually be faster than the outer loop. A common engineering rule is that the closed-loop bandwidth of the inner loop should be at least three times larger than the desired bandwidth of the outer loop. This is not a theorembut a design heuristic that reduces the dynamic interaction between the two loops.

Definition 1.6 (Transfer function of a cascade-controlled system).
Consider a plant with one control input u and $j_{\max} \geq 1$ measured outputs y_j ordered from the inner loop to the outer loop. For each $j \in \{1, \ldots, j_{\max}\}$, let $G_{R,j}(s)$ denote the controller of loop j, and let $G_{S,j}(s)$ denote the plant block between the output of loop $j-1$ and the output of

loop j. Define

$$G_{\text{cascade},0}(s) := 1, \tag{1.4}$$

$$G_{\text{cascade},j}(s) := \frac{G_{R,j}(s)G_{\text{cascade},j-1}(s)G_{S,j}(s)}{1 + G_{R,j}(s)G_{\text{cascade},j-1}(s)G_{S,j}(s)}. \tag{1.5}$$

The transfer function $G_{\text{cascade},j_{\max}}$ is called the reference-to-output *transfer function of the cascade-controlled system*.

The previous definition states the recursive formula but does not yet justify it. The following result gives the block-diagram argument behind the recursion. This is important because every outer-loop design uses the already closed inner loop as its plant model.

Theorem 1.7 (Cascade recursion).
Given the assumptions of Definition 1.6, assume that each loop is connected by unity negative feedback as in Figure 1.1 and that no denominator in (1.5) *is identically zero. Then* $G_{\text{cascade},j}(s)$ *is the transfer function from the reference of loop* j *to the output* y_j *for every* $j \in \{1, \dots, j_{\max}\}$.

Proof. The proof is by induction over the loop index. For $j = 1$, the formula is the standard unity-feedback expression $L/(1+L)$ with the locus $L = G_{R,1}G_{S,1}$. If the formula holds for loop $j-1$, then the inner closed-loop acts as the plant $G_{\text{cascade},j-1}$ seen by loop j. Therefore, the open-loop transfer function of loop j is $G_{R,j}G_{\text{cascade},j-1}G_{S,j}$, and unity negative feedback gives (1.5). □

Note that the cascade design is sequential rather than independent. Using the DC motor example from Task 1.4, once the current loop is closed, the speed controller should be designed for the closed current-loop dynamics, not for the original electrical equation alone. If in a later development stage the current-loop bandwidth is changed, the speed-loop and position-loop design models change as well.
Similarly, in a coating or packaging line in a production setting, a fast web-tension loop may be closed first. The slower line-speed or register-position loop then sees the closed tension loop as part of its plant. If the tension loop is retuned, the outer register controller may need to be retuned as well.
Considering the automotive example of an AD/ADAS path-following function, a low-level torque or steering-actuator loop is often closed below a vehicle-motion loop. The path controller should not be tuned against the bare actuator dynamics, but against the closed actuator response delivered by the low-level controller.

The design process follows immediately from the theorem, but does not provide an explicit design order for implementation. Here, we follow the design from inside to outside. This way we

acknowledge the dependency of the outer loops on the closed-loop behavior produced inside. The following Algorithm 1 turns this idea into a practical workflow.

Algorithm 1 Design of a cascade controller

Input: Cascade structure of Definition 1.6

procedure DESIGN CASCADE CONTROL($G_{S,1}(s), ..., G_{S,j_{\max}}(s)$)
  $G_{\text{cascade},0}(s) \leftarrow 1$
  for $j = 1, \ldots, j_{\max}$ **do**
    Design $G_{R,j}$ for the open-loop transfer function $G_{R,j}(s)G_{\text{cascade},j-1}(s)G_{S,j}(s)$.
    Compute the closed-loop transfer function $G_{\text{cascade},j}$ from (1.5).
  end for
end procedure

Output: Cascade controllers $G_{R,j}$ for $j \in \{1, j_{\max}\}$

The algorithm is not a one-shot formula but instead an iterative engineering procedure. Design an inner loop, replace it by its closed-loop model, and then continue outward. In the DC motor, Task 1.4 explains why current, speed, and position have different natural time scales. Algorithm 1 may be used to construct the control hierarchy in that exact ordering.

Task 1.8 (DC motor cascade transfer functions)
Building on the signal and time-scale analysis in Task 1.4, assume that the back electromotive-force term $K_e\omega$ in (1.1) *is treated as a disturbance for the current loop. Derive the plant blocks for a current loop, a speed loop, and a position loop.*

Solution to Task 1.8: Neglecting the coupling term $K_e\omega$ for the current-loop plant gives

$$G_{S,1}(s) = \frac{\hat{i}_a(s)}{\hat{u}_a(s)} = \frac{1}{L_a s + R_a}. \tag{1.6}$$

From (1.2), the transfer function from current to angular velocity for zero load torque is

$$G_{S,2}(s) = \frac{\hat{\omega}(s)}{\hat{i}_a(s)} = \frac{K_t}{J_m s + b_m}. \tag{1.7}$$

Finally, (1.3) gives

$$G_{S,3}(s) = \frac{\hat{\varphi}(s)}{\hat{\omega}(s)} = \frac{1}{s}. \tag{1.8}$$

A three-loop cascade can therefore be designed in the order current, speed, position.

In the latter task, we turned the qualitative time-scale ordering from Task 1.4 into transfer-function blocks. These blocks are the modeling interface between physical insight and the recursive cascade formula in Theorem 1.7.

Remark 1.9 (Choice of controllers in inner loops)
The inner loops are usually not designed for zero steady-state error with respect to their own references. Their main purpose is to shape the plant seen by the next outer loop. Therefore, fast P or PDT$_1$ controllers are often preferred for inner loops, while integral action is placed in the outermost loop where steady-state tracking accuracy matters.

Figure 1.2 shows how the blocks from the task are used in a complete drive-control structure. The diagram is intentionally more detailed than Figure 1.1 because it also shows the load torque entering the mechanical part of the drive.

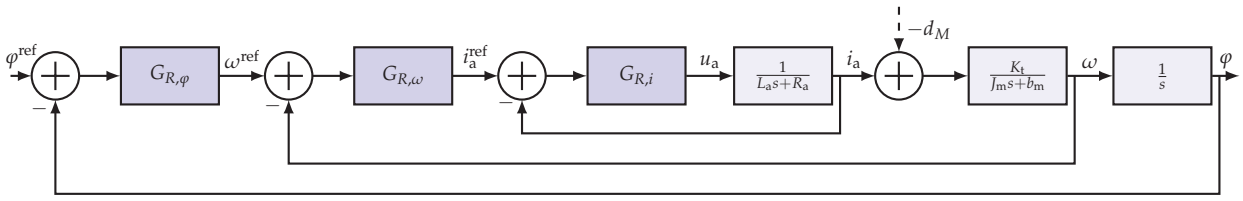


Figure 1.2: Cascade control structure for a DC motor with load torque.

Task 1.10 (Cascade control for an integral outer plant)
Using the interpretation of position as the integral of speed from Task 1.4, consider a two-loop cascade in which the outer plant block is an integrator, $G_{S,2}(s) = 1/s$. Explain why measuring and controlling the inner variable can be advantageous compared with a single-loop controller that only uses the outer output.

Solution to Task 1.10: If the outer plant is an integrator, then the inner output is proportional to the derivative of the outer output. Measuring this inner output therefore provides derivative information without implementing a differentiator on the outer measured signal. This is advantageous because differentiators amplify measurement noise and are often difficult to realize as proper transfer functions. A cascade structure can therefore provide derivative-like damping for the outer loop while avoiding direct differentiation of the outer output.

The task addresses a big issue raised in standard SISO control where the D part of a controller may be sensitive to measurement noise. The tasks explains why an internal measurement can

replace a numerically fragile derivative. For the DC motor, the measured speed is more useful than differentiating the measured position.

Task 1.11 (Outer-loop tuning for a DC motor)
Building on Tasks 1.4 and 1.8, assume that the inner speed loop of a DC motor has already been tuned and can be approximated by

$$G_{\text{cascade},\omega}(s) = \frac{1}{1 + T_\omega s}, \qquad T_\omega = 0.016\,\text{s}.$$

The outer plant is the position integrator $1/s$*. Design a PI position controller*

$$G_{R,\varphi}(s) = K_{P,\varphi}\frac{1 + T_{I,\varphi}s}{T_{I,\varphi}s}$$

by canceling the speed-loop time constant. If the remaining actuator and sensor gain is $K_\varphi = 0.2$ *and the dominant sensor lag is* $T_\varphi = 0.009\,\text{s}$*, use the magnitude-flatness approximation to compute* $K_{P,\varphi}$*.*

Solution to Task 1.11: Cancel the speed-loop time constant of $G_{\text{cascade},\omega}(s)$ by choosing

$$T_{I,\varphi} = T_\omega = 0.016\,\text{s},$$

which reveals the controller

$$G_{R,\varphi}(s) = K_{P,\varphi}\frac{1 + T_{I,\varphi}s}{T_{I,\varphi}s}.$$

The outer open-loop transfer function becomes approximately

$$G_0(s) = G_{R,\varphi}(s) \cdot G_{\text{cascade},\omega}(s) = K_{P,\varphi}\frac{K_\varphi}{T_{I,\varphi}s(1 + T_\varphi s)}.$$

Thus the closed-loop transfer function is

$$G_{\text{cascade},\varphi}(s) = \frac{G_{0,\varphi}(s)}{1 + G_{0,\varphi}(s)} = \frac{K_{P,\varphi}K_\varphi}{K_{P,\varphi}K_\varphi + T_{I,\varphi}s(1 + T_\varphi s)}.$$

For $s = i\omega$, we obtain

$$G_{\text{cascade},\varphi}(i\omega) = \frac{K_{P,\varphi}K_\varphi}{K_{P,\varphi}K_\varphi + T_{I,\varphi}i\omega(1 + T_\varphi i\omega)} = \frac{K_{P,\varphi}K_\varphi}{K_{P,\varphi}K_\varphi - \omega^2 T_{I,\varphi}T_\varphi + i\omega T_{I,\varphi}}.$$

Hence, the amplitude response reads

$$\begin{aligned}\left|G_{\text{cascade},\varphi}(i\omega)\right|^2 &= \frac{K_{P,\varphi}^2 K_\varphi^2}{\left(K_{P,\varphi}K_\varphi - \omega^2 T_{I,\varphi}T_\varphi\right)^2 + \left(\omega T_{I,\varphi}\right)^2} \\ &= \frac{K_{P,\varphi}^2 K_\varphi^2}{K_{P,\varphi}^2 K_\varphi^2 + \omega^2\left(T_{I,\varphi}^2 - 2K_{P,\varphi}K_\varphi T_{I,\varphi}T_\varphi\right) + \omega^4 T_{I,\varphi}^2 T_\varphi^2}.\end{aligned}$$

and the squared magnitude denominator contains the term

$$-\omega^2\left(2K_{P,\varphi}K_\varphi T_{I,\varphi}T_\varphi - T_{I,\varphi}^2\right)$$

which shall vanish, i.e setting the magnitude-flatness approximation to zero. Hence, we need to set

$$K_{P,\varphi} \approx \frac{T_{I,\varphi}}{2K_\varphi T_\varphi} = \frac{0.016}{2 \cdot 0.2 \cdot 0.009} \approx 4.44$$

and obtain the controller

$$T_{I,\varphi} = 0.016\,\text{s}, \qquad K_{P,\varphi} \approx 4.44.$$

The task shows how the closed speed-loop model from the cascade design becomes the plant model for the position controller. From the workflow perspective, the most valuable takeaway is to cancel only dynamics that are sufficiently reliable, and then tune the remaining dominant behavior.

Summing up, cascade control is attractive because every loop can be designed with familiar SISO tools. Its main limitation stems from the workflow as it renders the design not to be modular in the strict sense. Any change of an inner loop changes the plant model for all outer loops.

Table 1.1: Advantages and limitations of cascade control

Advantage	Limitation
✓ Standard single-loop design methods can be applied loop by loop.	✗ Additional controllers and sensors are required.
✓ Fast internal disturbances can be rejected before affecting outer output.	✗ Poorly separated bandwidths can create undesirable loop interactions.
✓ Local actuator and sensor bounds can be considered.	✗ Integral action in inner loops can produce windup under local saturation.
✓ Outer plant can be simplified by shaping the inner closed-loop dynamics.	✗ A later change of an inner loop changes all outer design models.

1.2 Precontrol and Prefiltering

Feedback reacts after an error is visible at the measured output. Feedforward acts before an error is created. Its ideal form is inverse dynamics: the feedforward controller should compensate the plant dynamics between the reference and the output. This idea is powerful but also dangerous, because exact inverse dynamics may require noncausal prediction, differentiation, or unstable pole-zero cancellation.

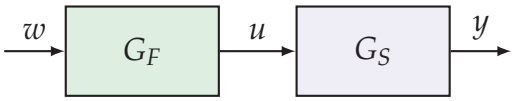


Figure 1.3: Simple feedforward control.

An ideal feedforward controller satisfies

$$G_F(s)G_S(s) = 1. \tag{1.9}$$

Thus, formally, $G_F = 1/G_S$. This inverse may fail to be realizable for three fundamental reasons:

- It may be improper if the plant is strictly proper.
- It may be noncausal if the plant contains a time delay.
- It may be unstable if the plant contains unstable zeros.

Next to these reasons, it is not advisable to cancel zeros, in particular unstable ones. The practical reason for the latter is that exact cancellation is never exact in implementation. A small mismatch turns the canceled unstable mode into an internal instability or a strongly amplified oscillation.

To avoid the fundamental reasons mentioned above, approximations of the inverse may be used.

Task 1.12 (Realizable approximate inverse for the DC motor speed plant)
Building on the speed block derived in Task 1.8, consider the DC motor speed plant from current to angular velocity,

$$G_{S,\omega}(s) = \frac{K_\mathrm{t}}{J_\mathrm{m}s + b_\mathrm{m}}.$$

Construct an approximate proper feedforward controller from a desired speed ω^{ref} to a current reference $i_\mathrm{a}^{\mathrm{ref}}$.

Solution to Task 1.12: The exact inverse is

$$G_{F,\omega}(s) = \frac{J_\mathrm{m}s + b_\mathrm{m}}{K_\mathrm{t}},$$

which is improper. A proper approximation is obtained by adding a small filter time constant $T_f > 0$:

$$G_{F,\omega}(s) = \frac{J_\mathrm{m}s + b_\mathrm{m}}{K_\mathrm{t}(1 + T_f s)}.$$

The smaller T_f is, the closer the approximation is to the inverse, but the more strongly high-frequency reference components are amplified.

The task shows why inverse dynamics must be filtered in practice. The filter time constant is a design knob between tracking accuracy and robustness against noise or abrupt references.

Building up on the background of such an approximation, our aim now is to get unwanted behavior out of the system as much as possible by an feedforward and take care about the rest by a feedback. There are exactly two combinations of feedforward and feedback revealing the precontrol and prefilter structure. In a precontrol structure shown in Figure 1.4, the feedforward signal is added at the plant input. In a prefilter structure given in Figure 1.5, the reference is filtered before it enters the feedback loop.
For these loop, we obtain the following:

Definition 1.13 (Transfer function of a precontrolled system).
Consider the SISO structure in Figure 1.4 with reference input w. If the closed-loop denominator

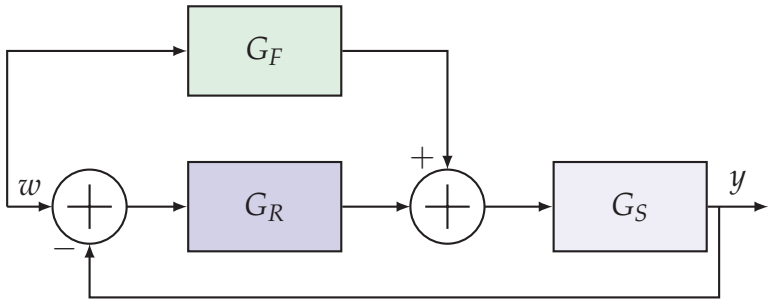


Figure 1.4: Precontrol structure with feedback and feedforward at the plant input.

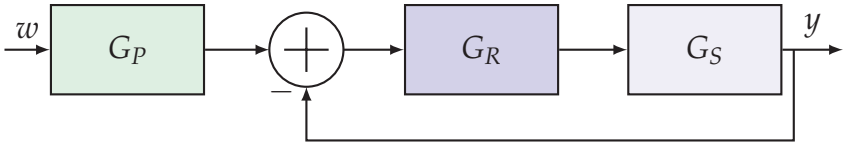


Figure 1.5: Prefilter structure with feedback and reference shaping.

is not identically zero, then

$$\hat{y}(s) = \frac{G_R(s)G_S(s) + G_F(s)G_S(s)}{1 + G_R(s)G_S(s)}\hat{w}(s) \tag{1.10}$$

is called the transfer function of the *precontrolled system* from w to y.

Definition 1.14 (Transfer function of a prefiltered system).
Consider the SISO structure in Figure 1.5 with reference input w. If the closed-loop denominator is not identically zero, then

$$\hat{y}(s) = \frac{G_P(s)G_R(s)G_S(s)}{1 + G_R(s)G_S(s)}\hat{w}(s) \tag{1.11}$$

is called the transfer function of the *prefiltered system* from w to y.

The two structures in Figures 1.4 and 1.5 look different. From the reference-to-output point of view, however, they can be made equivalent. This matters in design as sometimes the precontrol structure is physically more transparent, while the prefilter structure is easier to implement in a digital controller.

Theorem 1.15 (Equivalence of precontrol and prefilter).
Consider the transfer functions in Definitions 1.13 and 1.14. Assume that G_R is not identically

zero. If

$$G_P(s) = 1 + \frac{G_F(s)}{G_R(s)}, \tag{1.12}$$

then the precontrol and prefilter structures have identical reference-to-output transfer functions wherever the quotient is defined.

Proof. The precontrol transfer function has numerator $(G_R(s) + G_F(s))G_S(s)$. The prefilter transfer function has numerator $G_P(s)G_R(s)G_S(s)$. Substituting (1.12) gives $G_P(s)G_R(s) = G_R(s) + G_F(s)$. The denominators are identical. □

The theorem is an algebraic equivalence, not a statement about internal implementation. For the DC motor, a speed-feedforward term can either be added to the current reference or absorbed into a reference prefilter. The external reference response may be the same, but the internal current command and saturation behavior may differ.

Revisiting our production example, a planned speed increase can be used as a feedforward signal for the tension actuator. Alternatively, the same effect can be embedded in a reference prefilter. The output behavior may match, but the actuator usage and saturation risk should still be checked. Similarly in automotive engineering, curvature feedforward in a lane-centering function may be added directly to the steering command, or the path reference may be shaped before entering the feedback controller. Both can lead to the same nominal lateral response, while actuator limits and comfort constraints may favor one implementation.

The theorem also explains why feedforward and prefiltering are often tuned after the feedback loop is already stable. The feedback controller determines the closed-loop denominator, while precontrol and prefiltering mainly shape the reference numerator and the internal command signals. The next algorithm starts with the feedback loop for this reason: feedforward should improve a stable design, not hide an unstable one.

Algorithm 2 Design of a precontrol

Input: Precontrol structure of Definition 1.13

1: **procedure** DESIGN PRECONTROL($G_S(s)$)
2: Design the feed forward $G_F(s)$ to be (approximately) the inverse of the system $G_S(s)$.
3: Design the feedback $G_R(s)$ such that the nominal closed-loop is stable and sufficiently robust.
4: **end procedure**

Output: Controllers $G_R(s)$, $G_F(s)$

Precontrol is appropriate when an additional input-side command is physically meaningful. In the DC motor, this may be a current reference computed from a desired acceleration or speed.

The design must still be checked at the actuator, because inverse dynamics can produce large commands even when the external output response looks good.

A prefilter is motivated by a different implementation question. Sometimes the feedback loop should remain untouched, for example because it was already validated. In that case, the desired reference behavior can be shaped before the reference enters the loop.

Algorithm 3 Design of a prefilter

Input: Prefilter structure of Definition 1.14

1: **procedure** DESIGN PREFILTER($G_S(s)$)
2: Design the feedback controller $G_R(s)$ such that the nominal closed-loop is stable and sufficiently robust.
3: Design the feed forward $G_P(s)$ to shape the reference response.
4: **end procedure**

Output: Controllers $G_R(s)$, $G_P(s)$

Hence, prefiltering is a reference-shaping method. It does not require a second actuator input, but it also cannot directly compensate an external disturbance. Its main use is to make the commanded trajectory compatible with the closed-loop dynamics and actuator limits.

Remark 1.16 (Stability under feedforward and prefiltering)
Feedforward and prefiltering do not change the feedback denominator $1 + G_R(s)G_S(s)$. *Therefore, they do not move the closed-loop poles of the nominal feedback loop. They can, however, change internal signals and may cause actuator saturation or excite unmodeled dynamics.*

Task 1.17 (Prefilter for a PD-controlled plant)
Consider the plant

$$G_S(s) = \frac{5}{s(s+5)^2}$$

and the PD feedback controller

$$G_R(s) = K_P(s+1).$$

Design a prefilter that removes the closed-loop zero at $s = -1$ *and has unit static gain.*

Solution to Task 1.17: The closed-loop transfer function without prefilter is

$$G(s) = \frac{G_R(s)G_S(s)}{1 + G_R(s)G_S(s)} = \frac{5K_P(s+1)}{s(s+5)^2 + 5K_P(s+1)}.$$

The zero $s = -1$ is caused by the controller and may lead to an undesirable reference response. A prefilter with unit static gain that cancels this zero is

$$G_P(s) = \frac{1}{s+1}.$$

Then $G_P(0) = 1$, and the numerator zero of the reference transfer function is removed.

The task emphasizes that a prefilter shapes the reference response without retuning the feedback poles. This is useful when a derivative term is needed for stability or damping but creates an undesirable zero in the reference transfer function.

Table 1.2: Advantages and limitations of precontrol and prefilter structures

Advantage	Limitation
✓ Reference tracking can be improved without changing the feedback loop stability.	✗ Model errors in the feedforward or prefilter path directly affect the reference response.
✓ Precontrol can be designed from the plant model and therefore remains independent of later feedback retuning.	✗ Precontrol may generate large actuator commands if the approximate inverse is too aggressive.
✓ Prefilter can shape the closed-loop reference response and reduce overshoot due to unfavorable closed-loop zeros.	✗ Prefilter depends on the closed-loop transfer function and usually has to be redesigned after feedback retuning.
✓ Both structures can be made equivalent by choosing the prefilter according to the precontrol-feedback relation.	✗ Equivalence requires a realizable and well-defined ratio between feedforward and feedback.
✓ Fast reference changes can be smoothed before they excite poorly damped plant or actuator dynamics.	✗ Excessive smoothing improves robustness but reduces command-following speed.

Continued on next page

Table 1.2 – continued from previous page

Advantage	Limitation
✓ The feedback controller can still be designed mainly for disturbance rejection and robustness.	✗ Feedforward and prefilter paths cannot stabilize an unstable plant by themselves.
✓ Stable and minimum-phase plant dynamics can be approximately inverted to improve command tracking.	✗ Delays, unstable zeros, and improper inverses cannot be canceled exactly by a causal realizable element.
✓ In software implementation, a prefilter is often easy to integrate because it only modifies the reference signal.	✗ A prefilter cannot compensate disturbances that enter behind the reference-shaping path.
✓ In actuator-oriented implementations, precontrol gives an intuitive anticipatory input command.	✗ Input saturation, rate limits, and noise amplification must be handled explicitly in the precontrol path.

The precontrol and prefilter structures introduced above use additional model knowledge to improve the response to the reference signal. In both cases, the additional element is driven by the reference. Hence, these structures are well suited if the desired command trajectory is known and if the main objective is to shape how the plant follows this command.
However, many performance losses in closed-loop systems are not caused by the reference but by disturbances entering the plant at a different point. For the DC motor considered in Task 1.4, a typical example is a load torque acting on the shaft. In a production system, such a disturbance may be caused by a sudden change in material tension, tool contact, or friction. In a vehicle function, it may correspond to road slope, wind gusts, or tire-road interaction. These effects are not reference commands. They enter the physical system directly and therefore cannot be fully handled by shaping the reference alone. In the following section, we address the case where the disturbance injection is known.

1.3 Disturbance Control

Typically, a feedback controller is designed to suppress disturbances after they have affected the measured output. This is robust, but it is also reactive as the output must first deviate before the controller can respond.
If, however, the disturbance is measurable or can be estimated as shown in Figure 1.6, this delay

can be reduced. The basic idea is then to use the measured disturbance as an additional input signal and to compensate its expected effect before it propagates through the plant. This leads to disturbance control. In contrast to precontrol and prefiltering, disturbance control is not driven by the reference signal but by the disturbance signal itself. The design question therefore changes from reference shaping to disturbance compensation. In particular, instead of asking how the reference should be modified, we ask which additional actuator command cancels the measured disturbance path as far as causality, stability, and realizability allow.

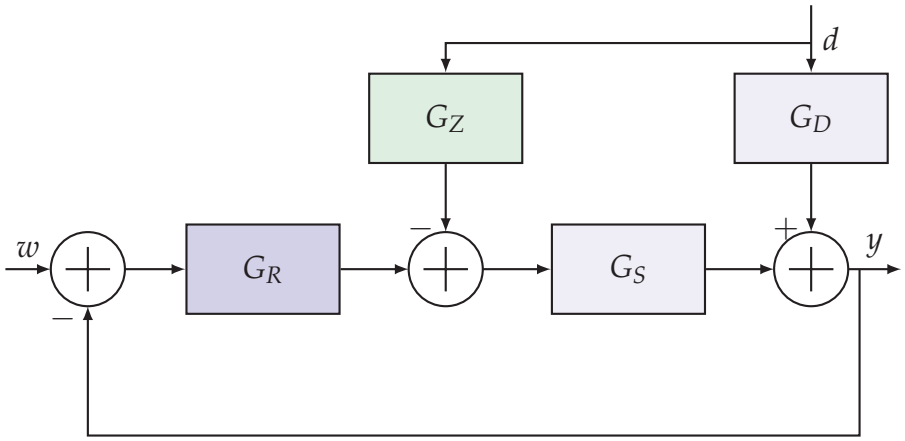


Figure 1.6: Disturbance control with measurable disturbance.

For such a system, we obtain the following:

Definition 1.18 (Transfer function of a disturbance-controlled system).
Consider the SISO disturbance-control structure in Figure 1.6. The plant transfer function is G_S, the disturbance-path transfer function is G_D, the feedback controller is G_R, and the disturbance controller is G_Z. If the closed-loop denominator is not identically zero, then

$$\hat{y}(s) = \frac{G_R(s)G_S(s)}{1 + G_R(s)G_S(s)}\hat{w}(s) + \frac{G_D(s) - G_S(s)G_Z(s)}{1 + G_R(s)G_S(s)}\hat{d}(s) \tag{1.13}$$

is called the transfer function representation of the *disturbance-controlled system*.

The formula contains the complete design idea in one term. The disturbance contribution disappears if the numerator $G_D - G_S G_Z$ disappears. The following theorem makes this cancellation condition explicit and separates it from the stabilizing role of feedback.

Theorem 1.19 (Ideal measurable-disturbance cancellation).
Consider the disturbance-controlled system in Definition 1.18. Assume that G_S is not identically

zero and that the closed-loop denominator is not identically zero. If

$$G_Z(s) = \frac{G_D(s)}{G_S(s)}, \tag{1.14}$$

then the transfer function from d to y is zero wherever the quotient in (1.14) *is defined. Moreover, the reference-to-output denominator remains* $1 + G_R G_S$.

Proof. The disturbance-to-output numerator in (1.13) is $G_D - G_S G_Z$. Substituting (1.14) makes this numerator zero. The feedback denominator does not contain G_Z, so the nominal characteristic equation is unchanged. □

Task 1.20 (Load-torque compensation for the DC motor)
Building on Tasks 1.4 and 1.8, assume that the current loop of the DC motor is ideal, so that the commanded current is applied directly. For the mechanical equation

$$J_\mathrm{m}\dot{\omega}(t) + b_\mathrm{m}\omega(t) = K_\mathrm{t} i_\mathrm{a}(t) - d_M(t),$$

compute the ideal disturbance controller from measured load torque d_M to the current command. Use the sign convention of Figure 1.6.

Solution to Task 1.20: The plant path from current to speed is $G_S = K_\mathrm{t}/(J_\mathrm{m}s + b_\mathrm{m})$. The disturbance path from load torque to speed is $G_D = -1/(J_\mathrm{m}s + b_\mathrm{m})$ if the load torque is added with its physical negative sign. In Figure 1.6, the disturbance controller is subtracted at the plant input. Therefore, one may equivalently model $G_D = 1/(J_\mathrm{m}s + b_\mathrm{m})$ for the magnitude of the disturbing load and obtain

$$G_Z(s) = \frac{G_D(s)}{G_S(s)} = \frac{1}{K_\mathrm{t}}.$$

Thus, the measured load torque is compensated by adding the current equivalent d_M/K_t with the sign that counteracts the load.

The task shows the special value of a disturbance measurement: the load torque can be compensated before it appears as a speed
Abstracting from the example, the result has a different impact than for prefilter and precontrol as it addresses the deviation before it can occur, i.e. before the feedback G_R must act on it. Note

that this does not replace the speed or position feedback loop, but instead it reduces how much error the feedback loop must correct.
In the production line, a measurable change in material thickness or roll diameter can be fed forward to the tension actuator. The ordinary tension feedback remains necessary, but it no longer has to compensate the full disturbance after the tension error has appeared.
In an AD/ADAS longitudinal-control function, an estimated road grade can be fed forward to the drive-torque command. The speed feedback still stabilizes the vehicle speed, while the grade feedforward reduces speed droop on uphill sections.

The ideal formula is useful as a target, but very often shows a several drawback. This drawback occurs if the quotient in (1.14) cannot be realized, i.e. if the quotient violates properness, causality, or stability.

Corollary 1.21 (Realizability limitation).
If G_D / G_S is improper, noncausal, or unstable, then exact disturbance cancellation cannot be implemented by a stable causal proper controller. In that case, G_Z must be replaced by a realizable approximation.

Proof. Considering the properties of G_D / G_S, we obtain

- If this quotient is improper, then it cannot be realized as a proper finite-dimensional causal transfer function.
- If it contains a negative time delay, then it depends on future values of the disturbance and is noncausal.
- If it has unstable poles, then it is not internally stable as a feedforward system.

Therefore only approximate cancellation remains possible. □

Task 1.22 (Realizable disturbance controller with delay)
Using the realizability restrictions from Corollary 1.21, consider

$$G_S(s) = \frac{K}{(1+T_1 s)(1+T_2 s)} \exp(-2s).$$

Compute the ideal and one realizable approximate disturbance controller for the cases $G_D = 1$ and $G_D = \exp(-3s)/(1+T_3 s)$.

Solution to Task 1.22: For $G_D = 1$, the ideal quotient is

$$G_Z(s) = \frac{(1+T_1 s)(1+T_2 s)}{K} \exp(2s),$$

which contains a negative delay and is not causal. Ignoring the noncausal delay and adding a denominator to make the controller proper gives, for example,

$$G_Z(s) = \frac{(1+T_1 s)(1+T_2 s)}{K(1+Ts)^2}.$$

For $G_D = \exp(-3s)/(1+T_3 s)$, the ideal quotient is

$$G_Z(s) = \frac{(1+T_1 s)(1+T_2 s)}{K(1+T_3 s)} \exp(-s).$$

The delay is now causal. A proper approximation is

$$G_Z(s) = \frac{(1+T_1 s)(1+T_2 s)}{K(1+T_3 s)(1+Ts)} \exp(-s).$$

The task separates the ideal cancellation formula from an implementable controller. Delays and relative degrees decide whether the ideal quotient can be used directly or must be replaced by a filtered approximation.

In general, the mathematical ideal in Corollary 1.21 should be read as a benchmark. In the DC motor, exact load-torque compensation is simple if the disturbance enters through the same mechanical channel as the motor torque. If the disturbance path contains additional delay or filtering, only approximate compensation may be realizable.

In our production setting, a roll-diameter disturbance may be measured before it affects web tension. If the measurement is delayed or strongly filtered, the feedforward action must be slowed down as well or otherwise, it may amplify noise or produce actuator peaks.

Within the automotive setting, road-grade compensation is useful only if the grade estimate is available early enough and is not too noisy. If the estimate is delayed, exact cancellation would require a predictor, and the practical solution is a filtered approximation.

Compared with precontrol and prefiltering, disturbance control uses measured disturbance information. This can significantly improve disturbance suppression, but it also requires additional sensing or estimation and may introduce derivative-like behavior in the actuator command.

The chapter therefore leads to a practical design principle. Use feedback to guarantee stability and robustness. Use cascade loops when meaningful internal variables are available. Use precon-

Table 1.3: Advantages and limitations of disturbance control compared with precontrol and prefiltering.

Advantage	Limitation
✓ Measurable disturbances can be compensated before they create a large output error.	✗ Additional sensors or estimators are required.
✓ Feedback stability is retained because the feedback denominator is unchanged.	✗ The disturbance controller may be improper, noncausal, or unstable.
✓ Disturbance dynamics can be included explicitly.	✗ Approximate inverse dynamics can increase noise sensitivity or actuator peaks.
✓ The feedback controller can be designed mainly for robustness and remaining uncertainty.	✗ Model errors in the disturbance path reduce compensation quality.

trol or prefilters when reference behavior can be improved by model knowledge. Use disturbance control when the disturbance is measurable early enough and its effect on the plant is sufficiently well known. In all cases, the additional structure should be checked against the same three realizability properties properness, causality and stability.

CHAPTER 2

COMPLEX CONTROL ELEMENTS

In Chapter 1, we modified the feedback loop by adding inner loops, feedforward paths, prefilters, and disturbance controllers. This chapter changes the point of view. We now look at control elements whose input-output behavior is not described by a linear transfer function. The most important examples are switching elements and static nonlinear maps. These occur if an actuator does not accept an arbitrary continuous command but only a fixed number of operating modes.
Within this chapter, our aim is to describe the properties of and induced by such elements. In particular, we discuss how these elements may be utilized to replace continuous actuator commands by switching or map-based components.
The running example remains the DC motor with load torque. In Chapter 1, the motor was mainly used to motivate internal variables such as current, torque, speed, and position. Here, the same motor is viewed from the actuator side. A power electronic stage often cannot apply every voltage continuously. Instead, it switches between voltage levels and uses pulse-width modulation to approximate a desired average voltage.
The same principle appears also in production where, e.g., heating elements, pneumatic valves, conveyors, clutches, and simple dosing units often switch between operating modes. They are inexpensive and robust, but the switching behavior may introduce oscillations, noise, or wear.
In vehicle functions, switching elements appear in gear selection, brake-pressure valves, actuator enable states, and discrete driving modes. A continuous desired behavior is therefore often implemented through actuators with internal switching or saturation limits.
The following Section 2.1 introduces bang-bang and Section 2.2 double-setpoint control, explains their switching behavior, and studies how hysteresis, filtering, and dead zones affect oscillations and convergence. Section 2.3 then generalizes this view to characteristic maps and separates nonlinear static behavior from linear dynamic behavior using Hammerstein and Wiener structures.

2.1 Bang-bang control

In contrast to our previous control elements, switching elements do not exhibit a continuous command but instead a finite set of admissible values. This makes them attractive for actuators with physically discrete modes and, as they are cheap, for low-cost hardware. At the same time, these elements render the closed-loop to be nonlinear. Hence, linear superposition, transfer-function multiplication, and pole-zero cancellation can no longer be applied to the switching element itself.

Task 2.1 (Running example: switched actuation of a DC motor)
Consider the DC motor from Chapter 1. Instead of applying an arbitrary armature voltage $U(t)$, suppose that the power electronics can apply only the two voltage levels 0 *and* $U_{\max}$. *Explain how this changes the interpretation of the actuator, and identify the signal that is still approximately continuous from the point of view of the motor.*

Solution to Task 2.1: If the switching frequency is high compared with the electrical time constant of the motor, the armature current does not follow every individual pulse. Instead, the motor reacts mainly to the average voltage over one switching period. Thus, the instantaneous voltage is discrete, but the averaged voltage can be approximately continuous.

From this basic example we get that switching does not automatically contradict continuous modeling. If the plant filters high-frequency switching components, the averaged behavior can still be described by an approximate continuous signal. We will utilize this observation later to mimic continuous behavior via pulse-width modulation.
Switching devices exist in a variety of realizations, cf. Table 2.1. The table is not meant as a complete catalog. Its purpose is to show that the same mathematical idea occurs in several technical domains.

Table 2.1: Technical possibilities of continuous and switching actuators

	Continuous actuator	**Switching actuator**
Valve	Proportional valve, servo valve, nozzle, slit	Shift valve
Electrical drive	Linear amplifier, controlled voltage source	Relay, transistor, inverter, thyristor

Continued on next page

Table 2.1 – continued from previous page

	Continuous actuator	**Switching actuator**
Clutch	Friction clutch, converter	Coupling clutch
Thermal process	Modulated power stage	On/off heater

Practitioners use such devices because they are cheap, robust, compact, and efficient. However, these elements show some severe drawbacks as switching induces oscillations, increases component wear, may excite resonances, and complicates analysis. The following definition make this behavior precise.

Definition 2.2 (Bang-bang control).
Consider a static element $f : \mathcal{U} \to \mathcal{Y}$ with $\mathcal{Y} := \{y_{\min}, y_{\max}\} \subset \mathbb{R}$ and $y_{\min} < y_{\max}$. Let $\theta \in \mathcal{U}$ be a threshold. The element is called a *bang-bang control* if

$$f(u) = \begin{cases} y_{\min}, & \text{if } u \leq \theta \\ y_{\max}, & \text{if } u > \theta \end{cases}. \tag{2.1}$$

(a) Bang bang control

(b) Bang bang control with hysteresis

Figure 2.1: Block diagram of a bang-bang control

Remark 2.3 (Nonlinearity)
A bang-bang element is not linear. In particular, it does not satisfy superposition. For this reason, the switching element itself cannot be represented by a transfer function. Transfer functions may still be used for the linear plant connected to the switching element.

A controller without hysteresis switches exactly at one threshold. This may cause chattering if measurement noise or small oscillations make the signal cross the threshold repeatedly. Hysteresis is a countermeasure to reduce chattering by introducing two thresholds. The output changes only after the input has moved sufficiently far away from the previous switching point.

Definition 2.4 (Bang-bang control with hysteresis).
Consider a system $f : \mathcal{U} \to \mathcal{Y}$ with $\mathcal{Y} := [y_{\min}, y_{\max}] \subset \mathbb{R}$ such that $\mathcal{Y} \neq \emptyset$. Furthermore, consider a threshold $\theta \in \mathcal{U}$ and a hysteresis width $\Delta > 0$ to be given. A *bang-bang element with hysteresis* is a dynamic switching rule satisfying

$$f(u) = \begin{cases} y_{\min}, & \text{if } u \leq \theta - \Delta \\ y_{\max}, & \text{if } u \geq \theta + \Delta \end{cases}. \tag{2.2}$$

This definition shows why hysteresis is not a memoryless function. Inside the hysteresis band, the current output depends on the previous output. This memory is small, but it is essential for avoiding arbitrarily fast switching.

Remark 2.5
Note that practical bang-bang elements are not ideal, i.e. there always exists a short delay after which a switch is actually applied.

Now the idea to render an implementation cheaper is to replace a continuous transfer function by means of a bang-bang element (with or without hysteresis). Looking at the output of a bang-bang element as given in Figure 2.1(left), we observe that it takes the form of a so called *pulse modulation function*. This class of functions include:

- Pulse-width modulation (PWM): The amplitude of output is fixed. The length of the impulse depends on the amplitude of the input signal.
- Pulse-frequency modulation (PFM): The amplitude of output is fixed. The frequency of the pulse depends on the amplitude of the input signal.
- Pulse-amplitude modulation (PAM): The amplitude of output depends on amplitude of input. the frequency of the pulse is fixed.

While all three options are applied, PWM is the most common one due to its similarity to electronics. Here, we focus on PWM only.
To replace a continuous signal by means of bang-bang elements, the signal to be mimiced is required as an input and is compared to a so called generator function, cf. Figure 2.2 for the general setting.
The easiest way is to generate a PWM signal is to apply a triangle function and compare it to a reference signal, cf. Figure 2.3. Note that the reference $w(t)$ in Figures 2.2, 2.3 is the input

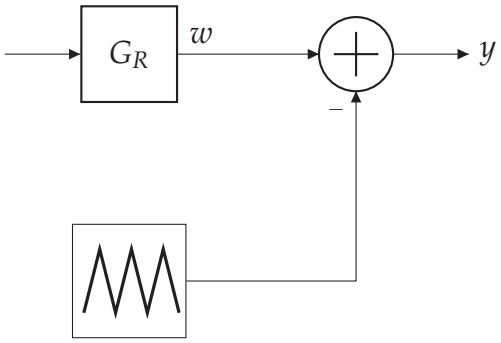


Figure 2.2: Mimicry of a continuous input function

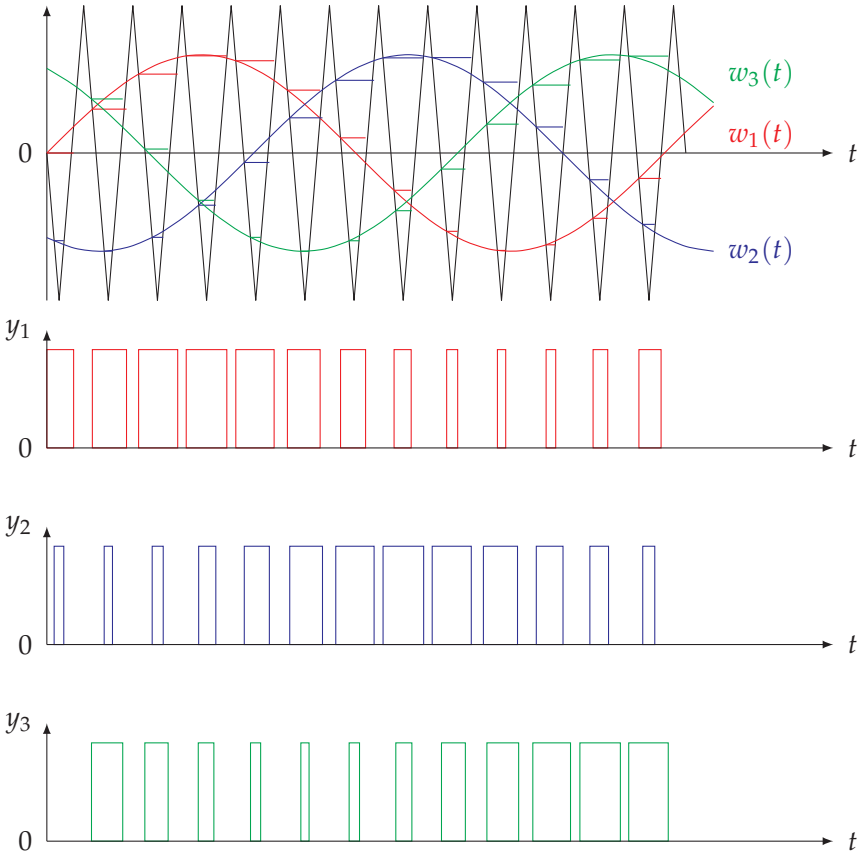


Figure 2.3: Sketch of a pulse width modulation using triangle functions

for our controller. Whenever the triangle function is less than the reference, the lower bound is applied. If it is higher, then the upper bound is used.

Remark 2.6
Note that other funtions like a sawtooth or delta function with respect to limits can be applied.

On an algebraic level, pulse-width modulation causes the actuator to change on-time within a fixed switching period.

Definition 2.7 (Pulse-width modulation).
Let $T_{\text{pwm}} > 0$ be a carrier period and let $d(t) \in [0,1]$ be a duty cycle. A two-level *pulse-width modulated signal* with levels $u_{\min}$ and $u_{\max}$ is a signal $u_{\text{pwm}}(\cdot)$, whose average over one carrier period T_{pwm} satisfies

$$\frac{1}{T_{\text{pwm}}} \int_t^{t+T_{\text{pwm}}} u_{\text{pwm}}(\tau)\, d\tau = u_{\min} + d(t)\,(u_{\max} - u_{\min})\,. \tag{2.3}$$

Task 2.8 (PWM voltage for the DC motor)
Use the actuator from Task 2.1. The available voltage levels are 0 *and* $U_{\max}$. *Compute the duty cycle that realizes the desired average voltage* $\bar{U} \in [0, U_{\max}]$. *Then repeat the calculation for a bipolar inverter with voltage levels* $-U_{\max}$ *and* $U_{\max}$.

Solution to Task 2.8: For the unipolar actuator, Definition 2.7 gives

$$\bar{U} = d(t) U_{\max} \quad \Longrightarrow \quad d(t) = \frac{\bar{U}}{U_{\max}}.$$

For the bipolar actuator, the averaged voltage is

$$\bar{U} = -U_{\max} + d(t)\,(2U_{\max}) \quad \Longrightarrow \quad d(t) = \frac{1}{2}\left(1 + \frac{\bar{U}}{U_{\max}}\right).$$

Hence, the duty cycle is the interface between the continuous controller and the switching actuator. The controller may compute an average voltage, while the power electronics realizes this value by switching.

The next result explains why bang-bang control leads to oscillations even for a very simple plant shown in Figure 2.4.

Theorem 2.9 (Oscillation for a scalar integrator).
Consider the closed-loop system given in Figure 2.4 and let $w \in \mathbb{R}$ *be constant* $u \in \{u_{\min}, u_{\max}\}$

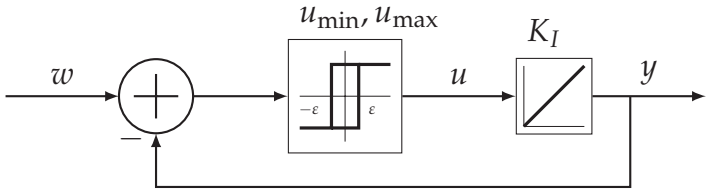


Figure 2.4: Bang-bang feedback for a scalar integrator.

and hysteresis width $\Delta > 0$. Then the solution enters a periodic motion between $w - \Delta$ and $w + \Delta$. Its amplitude with respect to w is Δ, and its period is

$$T_{\text{osc}} = 2\Delta \left(\frac{1}{u_{\max}} + \frac{1}{-u_{\min}} \right). \tag{2.4}$$

Proof. Starting at $w - \Delta$, the control $u_{\max}$ drives the integrator linearly to $w + \Delta$. The distance is 2Δ, so the travel time is $2\Delta / u_{\max}$. Then the switch to $u_{\min}$ occurs and the return time is $2\Delta/(-u_{\min})$. Adding both times gives (2.4). □

The interpretation is direct. Hysteresis removes arbitrarily fast switching, but it does not remove oscillation. Larger hysteresis produces larger oscillation and lower switching frequency. Larger control bounds reduce the period but increase actuator stress.

Task 2.10 (Estimating bang-bang oscillation)
A simple on/off heater is approximated by $\dot{y} = u$, with $u_{\max} = 2$, $u_{\min} = -1$, and hysteresis half-width $\Delta = 0.5$. Compute the bang-bang element oscillation period predicted by Theorem 2.9.

Solution to Task 2.10: Using (2.4), we obtain

$$T_{\text{osc}} = 2 \cdot 0.5 \left(\frac{1}{2} + \frac{1}{1} \right) = 1.5.$$

Thus, one complete oscillation takes 1.5 time units in this idealized model.

From the latter we observe that the asymmetry of the actuator matters. Heating up with slope 2 is faster than cooling down with slope -1, so the cooling phase dominates the period.

We can extend the result to the general linear case shown in Figure 2.5:

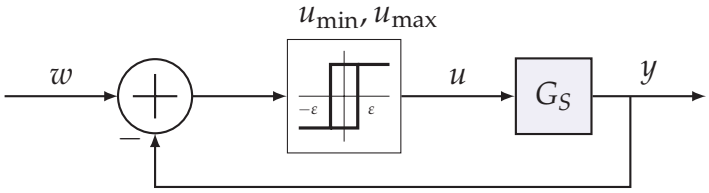


Figure 2.5: Bang-bang feedback for a linear system

Corollary 2.11 (Oscillation for bang-bang control w/o hysteresis).
Consider a closed-loop system as given in Figure 2.5. Suppose a controllable operating point $x^\star$ to exist for system G_S and let $u^\star$ to satisfy $u^\star \in \{u_{\min}, u_{\max}\}$. Then the solution enters a periodic motion. Its amplitude is directly proportional to Δ whereas the frequency satisfies

$$T_{\text{osc}} \sim 2\Delta \left(\frac{1}{u_{\max}} + \frac{1}{-u_{\min}} \right). \tag{2.5}$$

Proof. Follows by linearity using the arguments of the proof of Theorem 2.9. □

Interpreting the result in terms of production engineering, a thermostat in a heating tank with a large hysteresis band switches less often but produces larger temperature variation. A narrow band improves product temperature accuracy but increases wearoff.
Similarly, for an actuator with discrete brake-pressure valves in a car, a narrow switching band can improve tracking of a requested pressure, but it may increase valve activity, noise, and wear.

In order to reduce the oscillation amplitude, a natural engineering response is to amplify small errors before the bang-bang element. To avoid measurement noise to affect the loop, a filter to reduce high-frequency measurement noise may be added. This results in the following engineering approximation given in Figure 2.6.

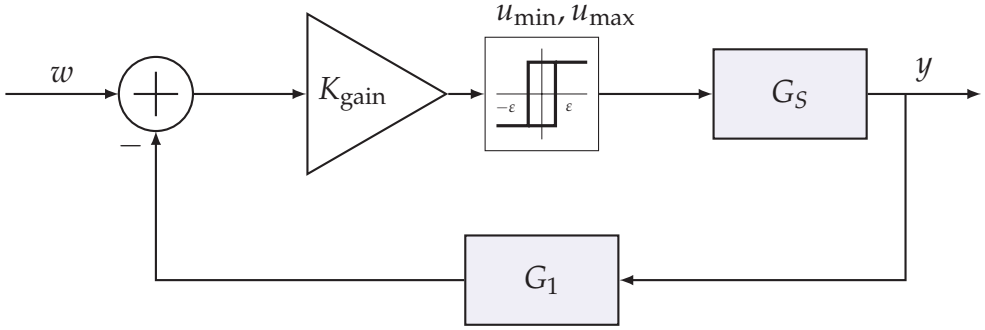


Figure 2.6: Bang-bang feedback with gain and low-pass measurement path.

Note that while the loop is exact for a static sensor gain, it becomes approximate when a dynamic low-pass filter is used. Low pass with static gain κ

Theorem 2.12 (Effective hysteresis under gain scaling).
Consider the system shown in Figure 2.6 with gain $K_{\text{gain}} > 0$ *and low pass static gain* $\kappa > 0$ *of* G_1*. Then the solution enters a quasi-static approximation of a periodic motion which is slow compared with the filter bandwidth. Its amplitude is directly proportional to* $\Delta/(K_{\text{gain}} \cdot \kappa)$ *whereas the frequency satisfies*

$$T_{\text{osc}} \sim 2\frac{\Delta}{K_{\text{gain}} \cdot \kappa}\left(\frac{1}{u_{\max}} + \frac{1}{-u_{\min}}\right). \tag{2.6}$$

Proof. The bang-bang element switches at $e_{\text{a}} = \pm\Delta$. Substituting gain $K_{\text{gain}} > 0$ and low pass static gain $\kappa > 0$ shows that the corresponding output thresholds are separated from the nominal value by $\Delta/(K_{\text{gain}}\kappa)$. As a low-pass filter behaves like its static gain for sufficiently slow oscillations, the assertion follows by Corollary 2.11. □

If we interpret the latter result physically, we see that an amplifier makes the bang-bang element more sensitive to small output errors, while the low-pass filter prevents measurement noise from dominating the switching decision. The price is additional phase lag. If the filter is too slow, it may enlarge oscillations rather than reduce them.
In a production setting, we may consider a conveyor tension controller, where filtering the tension sensor reduces switching caused by measurement noise. Too much filtering delays the reaction to real material changes.
Similarly in an ADAS brake actuator, filtering estimated slip or pressure signals avoids valve chatter. Excessive filtering may delay intervention and degrade safety margins.

Table 2.2: Advantages and limitations of bang-bang control

Advantage	Limitation
✓ Simple and robust hardware.	✗ Oscillation is unavoidable in many loops.
✓ Full actuator authority is used.	✗ Switching increases wear and noise.
✓ Hysteresis reduces chatter.	✗ Larger hysteresis increases output ripple.
✓ PWM can mimic average control.	✗ Averaging requires bandwidth separation.
✓ Easy implementation in software.	✗ Linear transfer functions do not apply.

To address the issue of oscillation, a modification of the bang-bang element may be utilized. Its aim is to come to a stand still using (at least) a third value.

2.2 Double setpoint control

A complete and general avoidance is also not possible, yet in particular cases a resolution can be found. These cases refer to systems which exhibit an I like behavior close to the reference. The reason why a resolution is possible is that for I like behavior close to the reference the control input satisfies $u \equiv u^\star$. A double-setpoint element is the basic version of a switching element that adds a third output value to a bang-bang element as captured in Figure 2.7.

(a) Without hysteresis (b) With hysteresis

Figure 2.7: Block diagram of a double-setpoint control

This middle value is used in a dead zone around the reference. Its purpose is to generate an actuator value that can hold the operating point approximately constant. For the DC motor, this may correspond to a voltage or torque value that balances friction and load at a desired operating point. More formally, we define:

Definition 2.13 (Double-setpoint control with hysteresis).
Consider a system $f : \mathcal{U} \to \mathcal{Y}$ with $\mathcal{Y} := [y_{\min}, y_{\max}] \subset \mathbb{R}$ such that $\mathcal{Y} \neq \emptyset$. Furthermore, consider a control value $y^\star \in \mathcal{Y}$, a threshold $\theta \in \mathcal{U}$ and two hysteresis widths $\Delta_2 > \Delta_1 > 0$ to be given. Then we call f to be *double-setpoint with hysteresis* iff

$$f(u) = \begin{cases} y^\star, & \text{if } u \leq \theta - \Delta_1 \\ y^\star, & \text{if } u \geq \theta + \Delta_1 \\ y_{\min}, & \text{if } u \leq \theta - \Delta_2 \\ y_{\max}, & \text{if } u \geq \theta + \Delta_2 \end{cases} \tag{2.7}$$

is the setting command of the system.

Before stating a stability result, one must be precise about what is stabilized. A double-setpoint element with a finite dead zone cannot generally force convergence to one point. It can, however, force convergence into a small target set and then stop switching. The next theorem makes this exact for an ideal integrator shown in Figure 2.8.

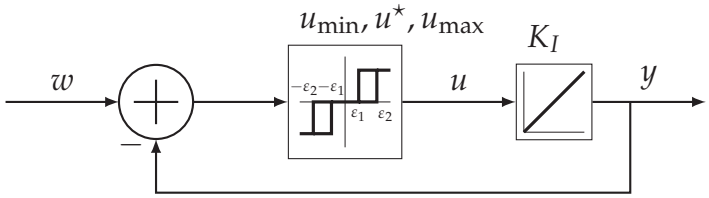


Figure 2.8: Double-setpoint control of a scalar integrator.

Theorem 2.14 (Dead-zone convergence for double-setpoint control).
Consider the closed-loop system given in Figure 2.8 and let $w \in \mathbb{R}$ *be constant,* $u \in \{u_{\min}, u_{\max}\}$ *and hysteresis widths* $0 < \Delta_1 < \Delta_2$. *Moreover, set* $\mathbf{u}^{p,\star} = 0$. *Then every solution reaches the dead zone*

$$\mathcal{D}_w := [w - \Delta_1, w + \Delta_1] \tag{2.8}$$

in finite time and remains there. Thus, the set $\mathcal{D}_w$ *is finite-time attractive and invariant.*

Proof. If the output is above the upper threshold, the selected input is negative and the integrator moves downward. If the output is below the lower threshold, the selected input is positive and the integrator moves upward. In both cases, the motion is monotone and reaches the dead zone in finite time. Inside the dead zone the input is zero, so the output remains constant. □

From the result we obtain that double-setpoint control removes steady oscillation only because it accepts a dead zone. It stabilizes a small set, not necessarily one exact point. Exact asymptotic convergence requires additional continuous action or a dead zone that shrinks to zero, which would again increase switching activity.
In production, this may be seen as a packaging machine, which may accept a small position tolerance. A double-setpoint actuator can move forward, stop inside the tolerance band, and avoid unnecessary corrections.
For an ADAS comfort function, we may use a dead band around a desired acceleration to avoid frequent small torque or brake requests that passengers would feel as jerk.

Remark 2.15
Comparing the explicit cases of Theorems 2.9 and 2.14, we observe that bang-bang control is usually a ripple-management problem, while double-setpoint control is a tolerance-management problem. Bang-bang control keeps moving while double-setpoint control tries to stop inside an acceptable band.

We can extend the result to the general linear case shown in Figure 2.9. Here, we directly integrate the amplifier and low pass we already discussed for the bang-bang element in Theorem 2.12 and enforce the I-like behavior by adding an integrator.

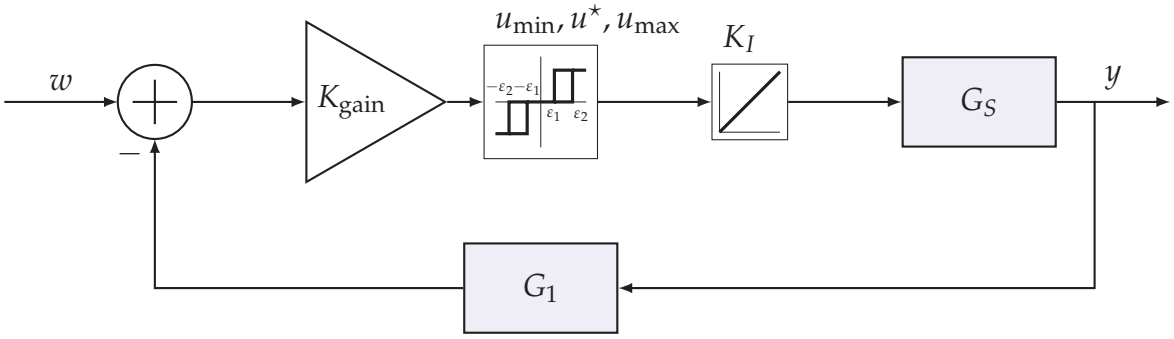


Figure 2.9: Double-setpoint feedback with gain, integral and low-pass measurement path.

For this setting, we can show the following:

Theorem 2.16 (Stability for double-setpoint control with hysteresis).
Consider a closed-loop system as given in Figure 2.9 and suppose a controllable operating point $(x^\star, u^\star)$ *to exist for system* G_S *with* $u^\star \in \{u_{\min}, u_{\max}\}$. *Then every solution reaches the dead zone*

$$\mathcal{D}_w \sim 2\frac{\Delta}{K_{\text{gain}} \cdot \kappa}\,[w - \Delta_1, w + \Delta_1] \tag{2.9}$$

in finite time and remains there.

Proof. Follows directly from Theorem 2.12 together with Theorem 2.14. □

The previous theorem also explains why double-setpoint control can reduce wear. Once the dead zone has been reached, no additional switching is needed in the ideal model. One can even go one step further and minimize the number of switches that are necessary to reach the reference value. To this end, a latency can be introduced to decelerate the speed of the control, cf. Figure 2.10. For such a structure, we can show an extension of Theorem 2.16 revealing:

Theorem 2.17 (Minimal asymptotic stability for double-setpoint control with hysteresis).
Given the setting from Figure 2.10 and suppose a controllable operating point $(x^\star, u^\star)$ *to exist for system* G_S *with* $u^\star \in \{u_{\min}, u_{\max}\}$. *Then the assertion of Theorem 2.16 holds and only a finite number of switches occur.*

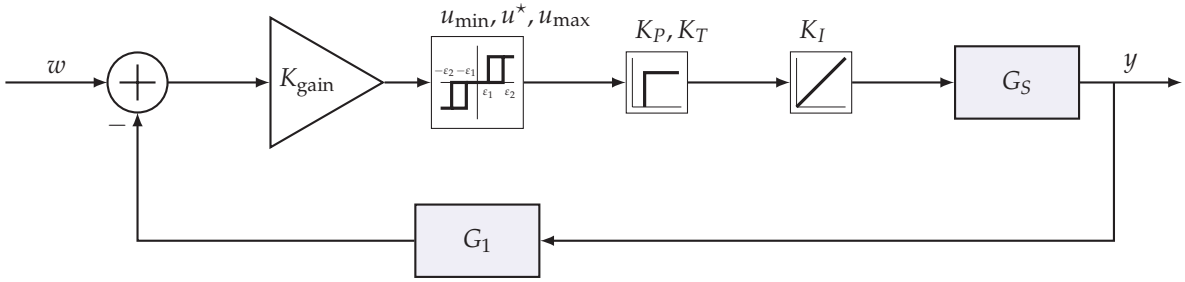


Figure 2.10: Double-setpoint feedback with gain, integral, latency and low-pass measurement path.

Proof. By Theorem 2.16 we obtain that the closed-loop enters the dead zone in finite time. Due to the latency, only a finite number of switches are possible in that time showing the assertion. □

Including the latency, however, comes at the price of reduced convergence speed of the closed loop. If the actuator reacts late, the output can overshoot the dead zone before the neutral command becomes effective. Hence, the dead zone depends on the latency leading to the problem of balancing the low pass and the latency parameters. In practice, the switching thresholds are therefore shifted in advance or combined with a dwell-time rule. This reduces switching but also slows the response.

The same switching hardware can also mimic a continuous controller. The following theorem gives the averaging idea behind this approximation.

Theorem 2.18 (Averaged approximation of continuous control).
Let u_{c} be a bounded continuous control signal with values in $[u_{\min}, u_{\max}]$. Let u_{pwm} be a PWM signal with duty cycle $d(t) \in [0,1]$ chosen such that its period average equals u_{c}. If the connected plant G_S is stable and its relevant bandwidth is small compared with $1/T_{\text{pwm}}$, then output of the closed-loop generated by u_{pwm} approximates the output of the closed-loop generated by u_{c}. In this averaged sense, a switching element can mimic a continuous P, PI, PD, or PID controller.

Proof sketch. By construction, the PWM signal has the same period average as the continuous command. The remaining signal is a high-frequency ripple. A stable plant with low bandwidth attenuates this ripple. Therefore, the low-frequency plant response is close to the response generated by the continuous command. □

The interpretation is that PWM does not make the switching element linear. Instead, we only cause the plant to see an averaged input if the switching is sufficiently fast. This is why PWM is useful for motor current control but may be unsuitable for mechanical components that cannot tolerate high switching rates.

Task 2.19 (Mimicking a continuous voltage command)
A speed controller for the DC motor computes the continuous voltage command $U_c = 6\,\mathrm{V}$. *The inverter can switch between* $0\,\mathrm{V}$ *and* $24\,\mathrm{V}$. *Compute the duty cycle and explain why the motor current can still be smooth.*

Solution to Task 2.19: The duty cycle is

$$d = \frac{U_c}{U_{max}} = \frac{6}{24} = 0.25.$$

Thus, the high voltage is applied for 25% of each PWM period. The current can remain smooth if the PWM frequency is high relative to the electrical time constant L/R, because the motor inductance filters the switching ripple.

In a production setting, this averaging can be seen for a heater, which can approximate fractional power by fast on/off cycling if the thermal dynamics are slow. The product temperature reacts mainly to average power.
Similarly, an electric steering or brake actuator may use high-frequency power electronics internally. The ADAS controller commands a continuous torque or pressure, while the actuator realizes it through switching.

Table 2.3: Advantages and limitations of double-setpoint control

Advantage	Limitation
✓ Switching can stop inside a dead zone.	✗ Exact point convergence is not given.
✓ Component wear can be reduced.	✗ A holding value must exist.
✓ Tolerances are handled explicitly.	✗ Dead-zone size limits accuracy.
✓ Finite switching is possible.	✗ Delay may cause overshoot.
✓ Useful for mode-based actuators.	✗ Parameter depend on application.

At this point, we like to come back to our Remark 2.3 stating that bang-bang and therefore also double-setpoint are nonlinear components. The ideas used in the results shown before follow one idea only: Additional components are included in the closed loop to simplify, transform and compensate nonlinearities and map the system to a linear one. To this end, adding low pass

and amplifier starting in Figure 2.6 and onward is equivalent to reducing the dead zone where a linearization is applied and at the same time making the linear reaction to dominate the nonlinear parts. Adding the integrator and latency in Figure 2.10 basically increases the order of the system by integration, i.e. the control is applied as to a derivative of the system, and compensating for tardiness of the system wrt. the control.
In general, even more complex connections can be drawn as we will see in the following section.

2.3 Characteristic map

Bang-bang and double-setpoint elements are special static nonlinearities. In practice, static nonlinearities are often more complex than two or three switching values. Engine torque, valve flow, battery limits, tire forces, and actuator efficiencies are typically stored as characteristic maps, which can also be multidimensional.

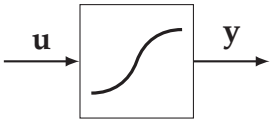


Figure 2.11: Nonlinear static system.

A characteristic map is a model that combines measured data and interpolation. It is not necessarily derived from first principles, hence also not necessarily given by a function. It is often the practical interface between experiments, calibration, and control design and may be implemented, e.g., by a (possibly multidimensional) data set. More formally, we define the following:

Definition 2.20 (Characteristic map).
Consider two sets $\mathcal{U}$ and $\mathcal{Y}$ and a set $\mathfrak{U} \times \mathfrak{Y} \subsetneq \mathcal{U} \times \mathcal{Y}$. Moreover, suppose an interpolation or approximation rule p to be given. Then we call $f : \mathcal{U} \to \mathcal{Y}$ satisfying

$$f(\mathbf{u}) = \begin{cases} \mathbf{y}, & \forall\ (\mathbf{u},\mathbf{y}) \in \mathfrak{U} \times \mathfrak{Y} \\ p(\mathbf{u}, \mathfrak{U} \times \mathfrak{Y}), & \forall\ (\mathbf{u},\mathbf{y}) \notin \mathfrak{U} \times \mathfrak{Y} \end{cases} \tag{2.10}$$

characteristic map.

These maps are typically identified offline using optimization methods such as regression or MINLP, or online via filter techniques such as Kalman. Note that the required storage for the input/output set rises exponentially with the dimension of the data sets, which make interpolation/approximation necessary in practice. Typical interpolation and approximation choices include polynomials, splines, radial basis functions, fuzzy rules, neural networks, and lookup tables

with multilinear interpolation. The choice is not only numerical. It determines smoothness, memory demand, extrapolation behavior, and whether derivatives are available for control design.

Task 2.21 (Characteristic torque map)
A simplified motor calibration table stores the torque M as a function of speed ω and averaged voltage $\bar{U}$. Explain why this is a characteristic map. Then name one reason why a differentiable interpolation may be useful for control design.

Solution to Task 2.21: The table contains sampled input-output data with input $\mathbf{u} = (\omega, \bar{U})$ and output $\mathbf{y} = M$. Values between measured grid points must be obtained by interpolation. Hence, it is a characteristic map in the sense of Definition 2.20. A differentiable interpolation is useful because local gains such as $\partial M/\partial \bar{U}$ and $\partial M/\partial \omega$ can be used for linearization, gain scheduling, or sensitivity analysis.

From the latter, we observe that a characteristic map is more than a table. It is a modeling assumption about how the system behaves between measured points.
The next result motivates why characteristic maps are still compatible with linear control methods near an operating point. Note that the compatibility is local, not global.

Theorem 2.22 (Local linearization of a characteristic map).
Consider a system $f : \mathcal{U} \to \mathcal{Y}$ be continuously differentiable in a neighborhood of $\mathbf{u}^\star \in \mathcal{U}$. Then, for $\mathbf{u}$ close to $\mathbf{u}^\star$, we have

$$f(\mathbf{u}) = f(\mathbf{u}^\star) + Df(\mathbf{u}^\star)(\mathbf{u} - \mathbf{u}^\star) + r(\mathbf{u}), \tag{2.11}$$

where

$$\lim_{\mathbf{u}\to\mathbf{u}^\star} \frac{\|r(\mathbf{u})\|}{\|\mathbf{u} - \mathbf{u}^\star\|} = 0. \tag{2.12}$$

Proof sketch. The statement is the first-order Taylor expansion of a continuously differentiable map. The derivative $Df(\mathbf{u}^\star)$ is the Jacobian, and the differentiability definition gives the remainder property (2.12). □

Based on the latter, a characteristic map can be replaced locally by a static gain matrix, e.g. a linearization or the interpolation of a data set. This supports linear controller design around one operating point. Note that it does not justify using the same linear approximation for large signal changes.

In an extrusion process in production, a local gain from screw speed to material flow may be valid near one material and temperature point but fail after a material change.
Similarly, in an automotive setting, a local tire-force slope may be useful near small slip angles but fail near the friction limit.

Static nonlinearities can be connected to linear dynamics in two common ways. If the nonlinearity is before the linear dynamics, the structure is called a Hammerstein structure. If the nonlinearity is after the linear dynamics, the structure is called a Wiener structure.

(a) Hammerstein structure (b) Wiener structure

Figure 2.12: Separation of static nonlinear maps and linear dynamics.

From Figure 2.12, we can subsummize the following:

Corollary 2.23 (Hammerstein structure cases).
The bang-bang and double-setpoint elements are special cases of the Hammerstein structure.

Proof. Follows by construction of bang-bang and double-setpoint elements. □

Since we are now dealing with nonlinear elements, we like to stress that Hammerstein and Wiener structures are generally not equivalent. Linear blocks commute with each other under multiplication of transfer functions, but nonlinear static maps do not commute with dynamic systems in general. Therefore, the position of the nonlinear element is part of the model.

This observation closes the chapter. Switching controllers and characteristic maps are practically useful because they describe actuators and calibrated components more honestly than purely linear models. At the same time, their nonlinearity limits the direct use of frequency-domain tools. In the next chapters, the analysis therefore shifts toward structures and state-space methods that can handle multivariable and nonlinear effects more systematically.

CHAPTER 3

COMPLEX CONTROL STRUCTURES

In Chapters 1 and 2, we extended the standard single-loop control setting in two different directions. Chapter 1 added structure by using several nested or parallel signal paths, and Chapter 2 showed that even a single control loop may become substantially more complex if the controller or actuator is nonlinear, switching, or map-based. In both chapters, however, the main viewpoint was to use one manipulated variable to influence one controlled variable, and limitations of manipulated and controlled variables were not treated as a central design issue.

Many technical systems do not fit this setting. A production machine may control temperature, tension, speed, and position at the same time. A vehicle function may simultaneously influence steering, braking, and drive torque while observing lateral position, yaw rate, velocity, and acceleration. Even the DC motor example from the previous chapters becomes multivariable as soon as several axes are operated together or as soon as torque, speed, position, current, and load interactions are considered in one common control architecture. In such cases, the central question is no longer only how one loop should be designed, but also how several loops interact with each other and how their inputs and outputs remain within admissible limits.

This chapter therefore introduces multi-input multi-output systems. The main difficulty is coupling between variables, i.e. changing one manipulated variable may affect more than one controlled variable as introduced in Section 3.1. Hence, changing one controlled variable may require more than one actuator. If these couplings are weak, they can be considered as disturbance and several single-loop controllers may still be sufficient. If they are strong, the controller must explicitly account for the so called MIMO structure as detailed in Section 3.2.

After introducing the latter, the chapter turns to implementation-oriented extensions that address limitations and nonidealities of real control loops. In particular, Section 3.3 treats actuator saturation and anti-windup methods to address the mismatch between controller output and actuator input under saturation. Thereafter, we discuss bumpless transfer to prevent undesirable jumps

when switching between controllers or modes in Section 3.4. Last, in Section 3.5 we discuss the Smith predictor to compensate known time delays by using a model-based prediction structure.

Since we are now dealing with multiple inputs and multiple outputs, we adapt the standing notation from Notation 1.2 and the standing assumptions from Assumption 1.3 to the multivariable setting.

Notation 3.1 (Multivariable transfer matrices)
For a linear time-invariant system with several inputs and outputs, we write

$$\underbrace{\begin{bmatrix}\hat{\mathbf{y}}_1(s)\\ \hat{\mathbf{y}}_2(s)\end{bmatrix}}_{\hat{\mathbf{y}}(s)} = \underbrace{\begin{bmatrix}G_{11}(s) & G_{12}(s)\\ G_{21}(s) & G_{22}(s)\end{bmatrix}}_{G(s)} \underbrace{\begin{bmatrix}\hat{\mathbf{u}}_1(s)\\ \hat{\mathbf{u}}_2(s)\end{bmatrix}}_{\hat{\mathbf{u}}(s)}. \tag{3.1}$$

The diagonal entries describe the desired input-output paths. The off-diagonal entries describe coupling paths.

Assumption 3.2
Unless stated otherwise, all transfer matrices in this chapter are real-rational and proper. Whenever inverse transfer functions are used, realizability, stability, and causality have to be checked explicitly.

Similarly, we extend our running example utilized throughout the chapter.

Task 3.3
Consider two identical DC motors from Task 1.4 driving a common mechanical load, for example the left and right wheels of a small automated vehicle or two synchronized conveyor axes. Let u_1 and u_2 denote the two motor commands, and let y_1 and y_2 denote two measured speeds. Explain why the system may not be treated as two independent SISO systems if the mechanical load couples both axes. Give one physical mechanism that can produce the coupling.

Solution to Task 3.3: The system is not independent if a command applied to motor 1 changes not only y_1 but also y_2, or if a command applied to motor 2 changes y_1. In transfer-matrix form, this means that at least one off-diagonal entry G_{12} or G_{21} is relevant.

A physical mechanism is a rigid mechanical coupling through the common load. For a vehicle, a steering or traction action on one side can affect yaw motion and therefore the measured speed or slip on the other side. For a conveyor, belt tension can transmit one drive action to another axis.

As we see from this example, multivariable control is not defined by the number of inputs alone. It is defined by the interaction between manipulated variables and controlled variables. If off-diagonal paths are relevant for desired properties such as stability or performance, the design has to consider them.

3.1 MIMO Systems and Coupling

A multi-input multi-output system is needed when several inputs and outputs cannot be assigned independently. Weak coupling can often be treated as a disturbance, using the ideas from Chapter 1. Strong coupling must be included in the controller design. Practically speaking, we decide whether separate SISO loops are acceptable or whether a multivariable structure is needed.

Definition 3.4 (MIMO system and coupling).
Consider transfer matrix $G : \mathbb{C}^{n_u} \to \mathbb{C}^{n_y}$. Then the system is called a *multi-input multi-output* system if $n_u > 1$ and $n_y > 1$. For any $j \neq k$ we call $G_{jk}(s)$ *coupling*. For a given tolerance $\theta > 0$, the coupling is called *weak on* Ω if

$$\left|G_{jk}(i\omega)\right| \leq \theta \qquad \forall \omega \in \Omega \text{ and all } j \neq k. \tag{3.2}$$

Otherwise, the coupling is called *strong on* Ω.

The definition separates two aspects. First, it describes connections. The diagonal entries describe the intended main input-output paths. The off-diagonal entries describe cross-couplings. And secondly, the strength of a path is not an absolute property. It is evaluated on a frequency range Ω and relative to a tolerance θ. If all off-diagonal transfer paths remain smaller than θ on Ω, then the coupling is called weak on this frequency range and may be disregarded. If at least one off-diagonal path exceeds the tolerance, then the coupling is strong and interaction between the control loops must be considered explicitly.

Remark 3.5
The term MIMO refers to the number of external input and output signals, not to the number

of internal states. Hence, a system with many states can still be a SISO system if it has one manipulated variable and one controlled variable only. In state-space form, this means that

$$\mathbf{x}(t) \in \mathbb{R}^{n_x}, \qquad \mathbf{u}(t) \in \mathbb{R}, \qquad \mathbf{y}(t) \in \mathbb{R}$$

with $n_x > 1$ is still a SISO system. Its transfer representation is a scalar transfer function.
In contrast, a MIMO system has several manipulated variables and several controlled variables

$$\mathbf{x}(t) \in \mathbb{R}^{n_x}, \qquad \mathbf{u}(t) \in \mathbb{R}^{n_u}, \qquad \mathbf{y}(t) \in \mathbb{R}^{n_y}, \qquad n_u > 1, \qquad n_y > 1.$$

Its frequency-domain representation is therefore a transfer matrix. The entries of this matrix describe the input-output paths, while the state dimension describes the internal memory and dynamics of the system.

For a production system, weak coupling means that changing the speed of one drive only has a small effect on another tension or position loop. Strong coupling means that one actuator significantly changes several process variables. For a vehicle function, weak coupling may allow separate steering and longitudinal controllers, whereas strong coupling appears when steering, braking, and drive torque jointly affect lateral motion, yaw rate, and vehicle speed.
The next definitions introduce two canonical structures illustrated in Figure 3.1. The P-canonical structure is close to the transfer matrix and convenient for controller design. The V-canonical structure is closer to modeling because coupling is described as feedback between variables.

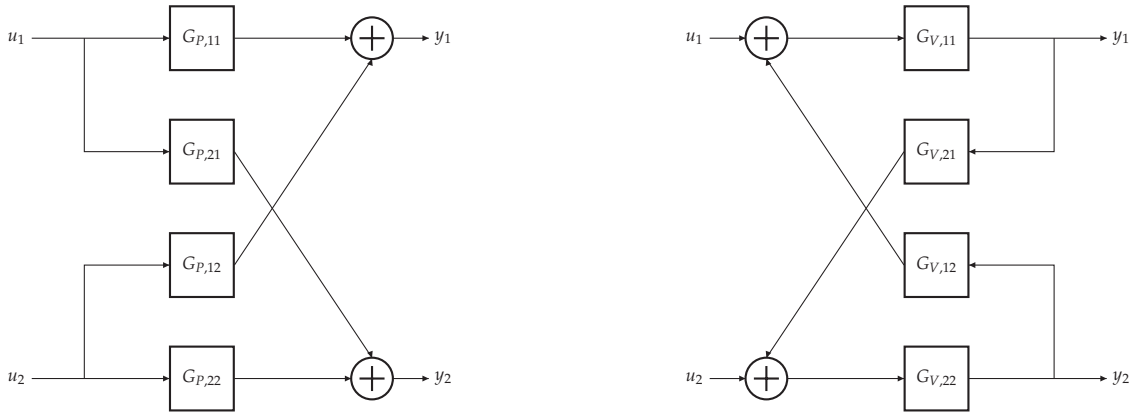
Definition 3.6 (P-canonical structure).
A MIMO system is said to be in P-canonical structure if its input-output behavior is given by

$$\begin{bmatrix} \hat{y}_1(s) \\ \hat{y}_2(s) \end{bmatrix} = \begin{bmatrix} G_{P,11}(s) & G_{P,12}(s) \\ G_{P,21}(s) & G_{P,22}(s) \end{bmatrix} \begin{bmatrix} \hat{u}_1(s) \\ \hat{u}_2(s) \end{bmatrix}. \tag{3.3}$$

Definition 3.7 (V-canonical structure).
A MIMO system is said to be in V-canonical structure if its input-output behavior is given by

$$\begin{pmatrix} \hat{y}_1(s) \\ \hat{y}_2(s) \end{pmatrix} = \begin{bmatrix} G_{V,11}(s) & 0 \\ 0 & G_{V,22}(s) \end{bmatrix} \cdot \left(\begin{pmatrix} \hat{u}_1(s) \\ \hat{u}_2(s) \end{pmatrix} + \begin{bmatrix} 0 & G_{V,12}(s) \\ G_{V,21}(s) & 0 \end{bmatrix} \cdot \begin{pmatrix} \hat{y}_1(s) \\ \hat{y}_2(s) \end{pmatrix} \right) \tag{3.4}$$

The V-canonical structure is useful for modeling, but P-canonical transfer matrices are easier to use in controller formulas. The following theorem gives the conversion.



(a) P canonical structure (b) V canonical structure

Figure 3.1: Canonical structures of MIMO systems with two inputs and two outputs

Theorem 3.8 (Conversion from V-canonical to P-canonical structure).
Consider two systems with two inputs and two outputs to be given. Suppose one system is in P-canonical structure and one in V-canonical structure. If

$$\det\left(\begin{bmatrix} 1 & -G_{V,11}(s)\cdot G_{V,12}(s) \\ -G_{V,22}(s)\cdot G_{V,21}(s) & 1 \end{bmatrix}\right) \not\equiv 0 \tag{3.5}$$

holds, then the equivalent P-canonical transfer matrix is given by

$$\begin{bmatrix} G_{P,11}(s) & G_{P,12}(s) \\ G_{P,21}(s) & G_{P,22}(s) \end{bmatrix} = \begin{bmatrix} 1 & -G_{V,11}(s)\cdot G_{V,12}(s) \\ -G_{V,22}(s)\cdot G_{V,21}(s) & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} G_{V,11}(s) & 0 \\ 0 & G_{V,22}(s) \end{bmatrix} \tag{3.6}$$

and the transfer matrices of both systems are equivalent.

Proof. Considering the V-canonical form, we obtain

$$\begin{pmatrix} \hat{y}_1(s) \\ \hat{y}_2(s) \end{pmatrix} = \begin{bmatrix} G_{V,11}(s) & 0 \\ 0 & G_{V,22}(s) \end{bmatrix} \cdot \begin{pmatrix} \hat{u}_1(s) \\ \hat{u}_2(s) \end{pmatrix} + \begin{bmatrix} 0 & G_{V,11}(s)\cdot G_{V,12}(s) \\ G_{V,22}(s)\cdot G_{V,21}(s) & 0 \end{bmatrix} \cdot \begin{pmatrix} \hat{y}_1(s) \\ \hat{y}_2(s) \end{pmatrix}.$$

Utilizing invertability from (3.5) reveals

$$\begin{bmatrix} 1 & -G_{V,11}(s)\cdot G_{V,12}(s) \\ -G_{V,22}(s)\cdot G_{V,21}(s) & 1 \end{bmatrix} \cdot \begin{pmatrix} \hat{y}_1(s) \\ \hat{y}_2(s) \end{pmatrix} = \begin{bmatrix} G_{V,11}(s) & 0 \\ 0 & G_{V,22}(s) \end{bmatrix} \cdot \begin{pmatrix} \hat{u}_1(s) \\ \hat{u}_2(s) \end{pmatrix}$$

$$\iff \quad \begin{pmatrix} \hat{y}_1(s) \\ \hat{y}_2(s) \end{pmatrix} = \begin{bmatrix} 1 & -G_{V,11}(s)\cdot G_{V,12}(s) \\ -G_{V,22}(s)\cdot G_{V,21}(s) & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} G_{V,11}(s) & 0 \\ 0 & G_{V,22}(s) \end{bmatrix} \cdot \begin{pmatrix} \hat{u}_1(s) \\ \hat{u}_2(s) \end{pmatrix}$$

providing the P-canonical form. □

This equivalence is useful upon usage. For the two-axis drive from Task 3.3, a V-canonical model may be natural if each wheel speed influences the vehicle body motion, which then feeds back into the other wheel speed. For controller design, the P-canonical matrix directly shows the four input-output paths. Hence, each of the descriptions has its advantages as summarized in Table 3.1.

Table 3.1: Advantages and limitations of P- and V-canonical structures

Advantage	Limitation
✓ P-canonical form directly gives the transfer matrix.	✗ P-canonical entries may be far from the physics.
✓ V-canonical form is often obtained from modeling.	✗ V-canonical form must be converted for many design formulas.
✓ P-canonical form simplifies decoupling calculations.	✗ Conversion may introduce nonrealizable transfer functions.
✓ V-canonical couplings can be easier to interpret.	✗ Feedback-like couplings can hide closed-loop singularities.

Our next aim is to make use of that structure to control the MIMO system appropriately.

3.2 Decoupling Control

There are three basic responses to coupling in MIMO systems.

- Decentralized control: For each input/output pair we design exactly one control. For each pair, the input of the other pair is considered to be a disturbance.
- Decoupling control: For each input/output pair we design one main control and for all couplings one decoupling control. The task of the latter is to reduce or eliminate the input of other pairs such that the pairs can be treated separately.
- Multivariable control: The control exhibits as many inputs and outputs as the system does.

This chapter focuses on decoupling because it is the closest extension of the SISO methods used in Chapters 1 and 2. We will consider the multivariable case in the later Chapter 5.
Considering the most simple case of MIMO-systems with only two inputs and two outputs together with a decentralized control, we can define its setting as follows:

Definition 3.9 (Decentralized control).
Consider system $G_S : \mathbb{C}^2 \to \mathbb{C}^2$. Then we call a controller exhibiting the diagonal structure

$$G_R(s) = \begin{bmatrix} G_{R,11}(s) & 0 \\ 0 & G_{R,22}(s) \end{bmatrix} \tag{3.7}$$

decentralized controller.

The latter case is illustrated in Figure 3.2. Here, the inputs act as a disturbance of the respective other outputs via the system and remain uncompensated. Note that in that case the effective plant is not diagonal in general but only if $G_{S,12}(s) = G_{S,21}(s) \equiv 0$.

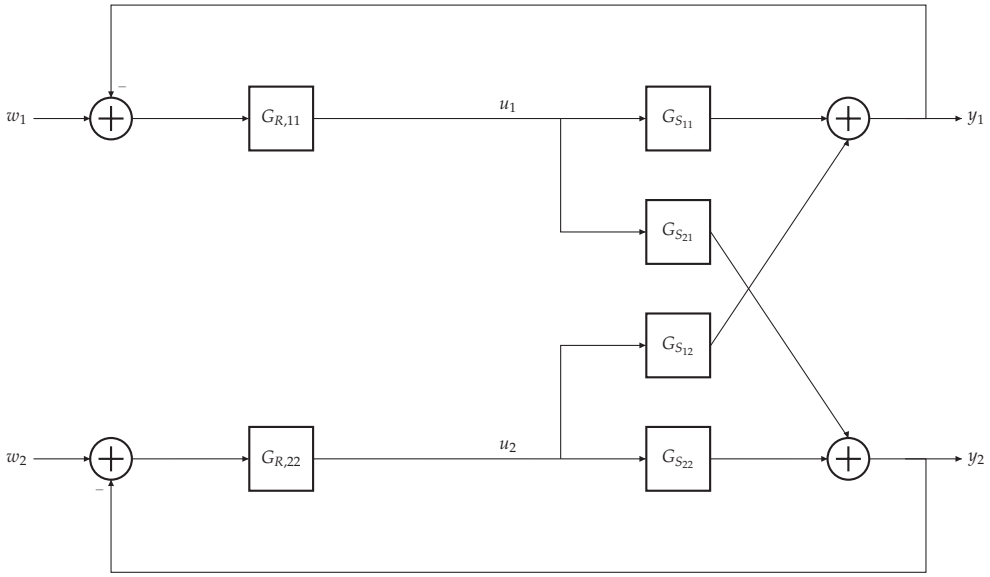


Figure 3.2: Decentralized control structure of MIMO system with P canonical structure

In case of decoupling, our aim is to achieve exactly the latter, i.e. to obtain that the effective plant $G_R(s) \cdot G_S(s)$ exhibits diagonal structure. To this end, we impose the structure shown in Figure 3.3.
Formally, we then define the following:

Definition 3.10 (Decoupling control).
Given a transfer function $G_S : \mathbb{C}^2 \to \mathbb{C}^2$, we call a controller $G_R : \mathbb{C}^2 \to \mathbb{C}^2$ a *decoupling controller* if the effective plant seen by the main controllers $G_{R,11}(s)$, $G_{R,22}(s)$ is diagonal or approximately diagonal.

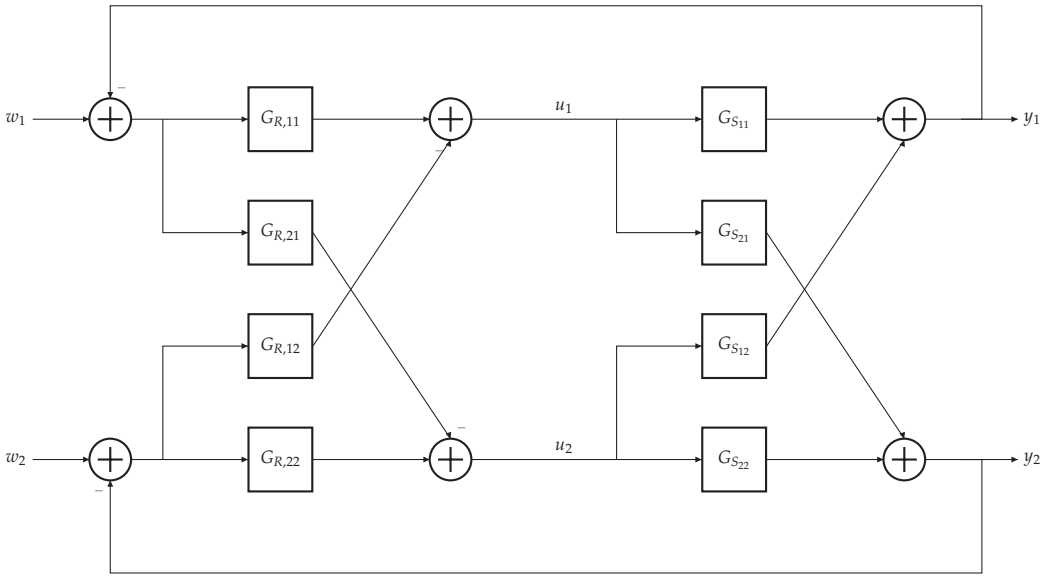


Figure 3.3: Decoupling structure of MIMO system with P canonical structure

The idea of decoupling control is a special case of disturbance rejection, i.e. we eliminate or at least reduce the impact of the systems on one another, which allows us to apply standard methods for the decoupled circuits. We now focus on eliminating the impact of the second system on the first, cf. Figure 3.4.

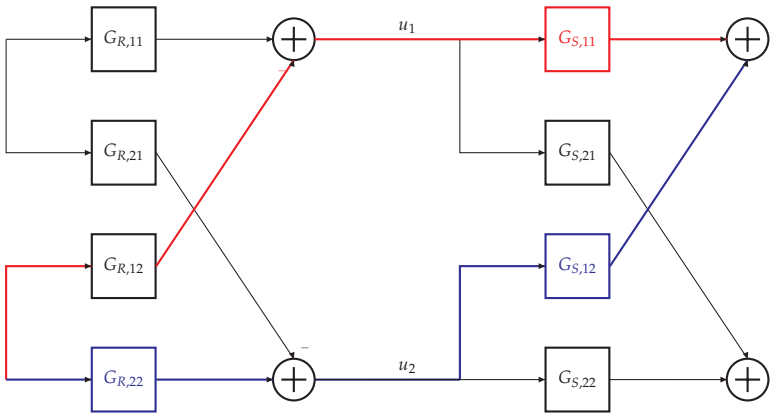


Figure 3.4: Elimination of coupling

In order to eliminate one another, the blue and red paths in Figure 3.4 need to be identical. Hence, we directly obtain

Corollary 3.11 (Decoupling condition).
Given a system $G_S : \mathbb{C}^2 \to \mathbb{C}^2$ *and a controller* $G_R : \mathbb{C}^2 \to \mathbb{C}^2$ *both in P-canonical structure*

given in Figure 3.3. If the conditions

$$G_{R,12}(s) = G_{R,22}(s) \cdot \frac{G_{S,12}(s)}{G_{S,11}(s)} \tag{3.8}$$

$$G_{R,21}(s) = G_{R,11}(s) \cdot \frac{G_{S,21}(s)}{G_{S,22}(s)} \tag{3.9}$$

hold, then the system is decoupled.

Proof. Results from a pathwise comparison of closed-loops including $R_{12}(s)$ and $R_{12}(s)$. □

Corollary 3.11 does show some implications. Since we have

$$\text{decoupling control} = \text{main control} \cdot \frac{\text{coupling system}}{\text{main system}}$$

and the main control must satisfy degree of nominator is equal to degree of denominator, then the decoupling control is realizable if and only if the number of poles of the coupling system is larger than number of poles of the main system. Additionally, the known limitations for realizability apply, cf. Section 1.3. Additionally, if a perfect decoupling is not possible, then its impact should be reduced.

It is of particular importance that even in the case of ideal decoupling, each system depends on both the main and the decoupling control. As a consequence, if we want to adapt the main controller in a later stage of the development, the decoupling controller needs to be adapted as well. One way to circumvent this problem is to consider a slight modification of the decoupling circuitry, cf. Figure 3.5.

Now, we consider a coupling path as an internally generated disturbance and aim to cancel this disturbance before it reaches the affected output. The following theorem gives one algebraic form of exact input decoupling.

Theorem 3.12 (Static input decoupling condition).
Consider a system $G_S : \mathbb{C}^2 \to \mathbb{C}^2$ *with* $G_{S,11} \not\equiv 0$ *and* $G_{S,22} \not\equiv 0$. *Define the input decoupler*

$$G_D(s) = \begin{bmatrix} 1 & -\dfrac{G_{S,12}(s)}{G_{S,11}(s)} \\ -\dfrac{G_{S,21}(s)}{G_{S,22}(s)} & 1 \end{bmatrix}. \tag{3.10}$$

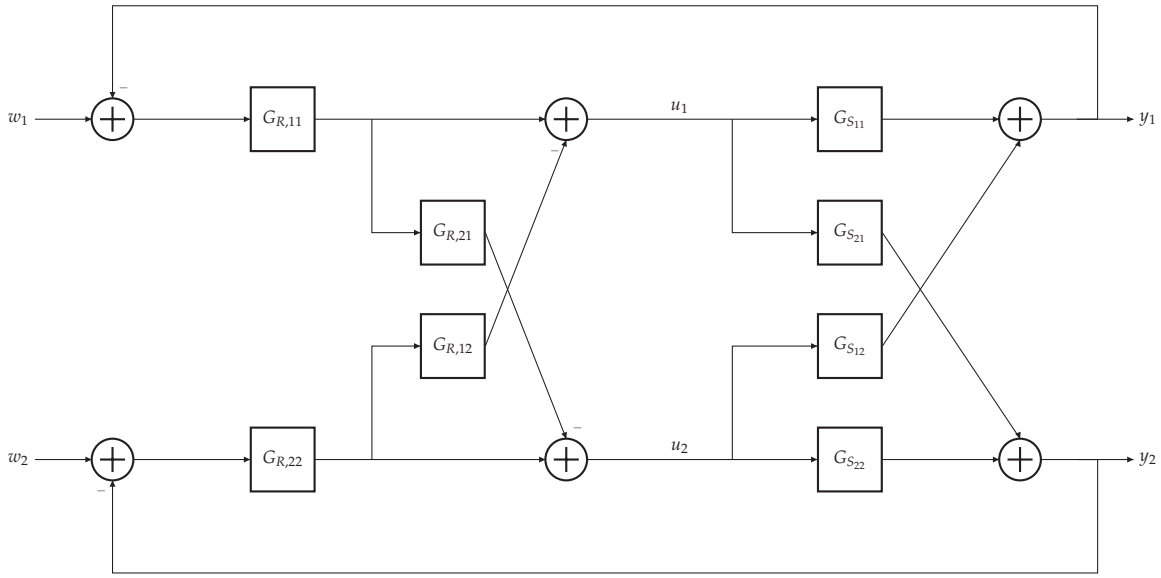


Figure 3.5: Adaptable decoupling structure of MIMO system with P canonical structure

If all entries of $G_D(s)$ are realizable, then the effective plant is diagonal with

$$G_S(s) \cdot G_D(s) = \begin{bmatrix} G_{S,11}(s) - \frac{G_{S,12}(s)G_{S,21}(s)}{G_{S,22}(s)} & 0 \\ 0 & G_{S,22}(s) - \frac{G_{S,21}(s)G_{S,12}(s)}{G_{S,11}(s)} \end{bmatrix}. \tag{3.11}$$

Proof. Multiplying the plant by the decoupler reveals

$$\begin{aligned} G_S(s) \cdot G_D(s) &= \begin{bmatrix} G_{S,11}(s) - \dfrac{G_{S,12}(s)G_{S,21}(s)}{G_{S,22}(s)} & -G_{S,11}(s)\dfrac{G_{S,12}(s)}{G_{S,11}(s)} + G_{S,12}(s) \\ G_{S,21}(s) - G_{S,22}(s)\dfrac{G_{S,21}(s)}{G_{S,22}(s)} & G_{S,22}(s) - \dfrac{G_{S,21}(s)G_{S,12}(s)}{G_{S,11}(s)} \end{bmatrix} \\ &= \begin{bmatrix} G_{S,11}(s) - \dfrac{G_{S,12}(s)G_{S,21}(s)}{G_{S,22}(s)} & 0 \\ 0 & G_{S,22}(s) - \dfrac{G_{S,21}(s)G_{S,12}(s)}{G_{S,11}(s)} \end{bmatrix}. \end{aligned}$$

□

The interpretation is important. Decoupling is not free. It cancels cross paths by changing what each main controller sees as its plant. Therefore, the SISO main controllers should be designed for the corrected diagonal dynamics, not blindly for G_{11} and G_{22}.

Task 3.13 (Decoupling a two-axis drive)
Building on Task 3.3, assume the approximate transfer matrix

$$G(s) = \begin{bmatrix} \frac{1}{1+s} & \frac{0.2}{1+2s} \\ \frac{0.1}{1+2s} & \frac{1}{1+s} \end{bmatrix}.$$

Compute the off-diagonal entries of the input decoupler from Theorem 3.12.

Solution to Task 3.13: The decoupler is

$$G_D(s) = \begin{bmatrix} 1 & -\frac{G_{S,12}(s)}{G_{S,11}(s)} \\ -\frac{G_{S,21}(s)}{G_{S,22}(s)} & 1 \end{bmatrix} = \begin{bmatrix} 1 & -\frac{0.2}{1+2s}(1+s) \\ -\frac{0.1}{1+2s}(1+s) & 1 \end{bmatrix}.$$

Both entries are proper and stable if the time constants are positive. Thus, this approximate decoupler is realizable in the usual transfer-function sense.

Note that the decoupling formulas are simple, but their implementation is limited by realizability. Whenever a quotient produces differentiating behavior, delays with the wrong sign, or unstable pole-zero cancellations, exact decoupling must be replaced by approximate decoupling.
For our production setting, we consider a coating line where changing web speed may also change web tension. A decoupler can reduce this interaction, but the remaining speed path after decoupling is not necessarily the original speed path.
Regarding aa vehicle stability function, differential braking may influence yaw rate and longitudinal speed. A decoupler can reduce yaw-speed interaction, but tire saturation may invalidate the exact cancellation.

Last, we want to emphasize a matrix inversion viewpoint, a result, which may be especially useful for larger multivariable systems, but it also makes the limitation of exact inversion visible.

Corollary 3.14 (Exact inverse decoupling).
Let $G : \mathbb{C}^{n_u} \to \mathbb{C}^{n_y}$ be square and invertible. For any diagonal desired transfer matrix G_d, the precompensator

$$G_D(s) = G^{-1}(s) \cdot G_\mathrm{d}(s) \tag{3.12}$$

gives the effective plant $G(s) \cdot G_D(s) = G_{\mathrm{d}}(s)$, provided $G_D(s)$ is realizable and internally stable.

Proof. Since $G_D(s)$ is assumed to be realizabe and internal stable, we can substitute (3.12) to obtain

$$G(s) \cdot G_D(s) = G(s) \cdot G^{-1}(s) \cdot G_{\mathrm{d}}(s) = G_{\mathrm{d}}(s).$$

Since G_{d} is diagonal, the effective plant is decoupled. □

The interpretation is the same as for feedforward and disturbance control in Chapter 1: exact inversion is mathematically attractive but physically limited. Nonminimum-phase zeros, delays and relative degree constraints may prevent exact implementation.
The following design procedure summarizes the engineering workflow.

Algorithm 4 Design of a decoupling controller

Input: System $G_S : \mathbb{C}^2 \to \mathbb{C}^2$
1: **procedure** DESIGN DECOUPLING CONTROL($G_S(s)$)
2: Apply Definition 3.4 to estimate whether coupling is weak or strong.
3: **if** Coupling is weak **then**
4: Start with a decentralized controller and verify robustness.
5: **else**
6: Choose and compute decoupler from Corollary 3.11 or Theorem 3.12.
7: **end if**
8: Check properness, stability, causality, delays, and actuator limits of the decoupler.
9: Design the main SISO controllers for the effective diagonal plant after decoupling.
10: Validate the complete multivariable loop by simulation and experiment.
11: **end procedure**
Output: Decoupling controllers $G_R(s)$

The motivation for Algorithm 4 is to avoid an all-or-nothing view of coupling. Weak coupling can be tolerated. Strong coupling should be compensated. Exact compensation is only acceptable if the required transfer functions are implementable.
The interpretation is that decoupling is a model-based extension of SISO design. It keeps familiar loop-shaping tools available, but it also introduces model dependence. If the model changes, the decoupler may have to change as well.

Table 3.2: Advantages and limitations of decoupling control

Advantage	Limitation
✓ Coupling is reduced systematically.	✗ Exact decoupling depends on model.
✓ SISO design can be reused.	✗ The effective diagonal may change.
✓ Matrix formulas scale better than block manipulation.	✗ Inversion may be improper or non-causal.
✓ Cross effects become visible.	✗ Saturation can destroy decoupling.

3.3 Anti-Windup Methods

The previous sections assumed that the computed input can be applied to the plant. This is rarely true. Motors have voltage and current limits, valves have opening limits, and vehicle actuators have comfort and safety bounds. In particular, it may happend that a controller generates a saturation, e.g. via an integrator, which causes a so called *windup*. In such a case, the desired input continues to grow internally although the real actuator is already at its bound.

Definition 3.15 (Saturation and windup).
Consider a desired controller output u_c as well as input bounds $u_{\min} < u_{\max}$ to be given. Then we called

$$u = \mathrm{sat}(u_c) = \begin{cases} u_{\min}, & \text{if } u_c < u_{\min}, \\ u_c, & \text{if } u_{\min} \leq u_c \leq u_{\max}, \\ u_{\max}, & \text{if } u_c > u_{\max}. \end{cases} \tag{3.13}$$

the actually *applied input*. Moreover, we denote the growth of an internal controller state caused by integrating an error while $u \neq u_c$ as *windup*.

Remark 3.16
Windup is most commonly associated with integral action, because the integral state continues to accumulate the control error while the actuator is saturated. However, the underlying phenomenon is more general. Any dynamic controller with internal states may exhibit windup if these states evolve according to the unconstrained controller output although the plant receives only

the saturated input. A purely static proportional controller can saturate, but it does not exhibit windup in this strict sense because it has no internal state that can accumulate the mismatch.

In blocks, we may utilize the so called saturation or limit block sketched in Figure 3.6.

(a) Saturation (b) Limit

Figure 3.6: Block diagram of a saturation of controllers

While the occuring error by not applying the intended input u_c is one issue, a secondary one arises when the error changes sign, which results in the controller requiring time to unwind its stored integral action. To illustrate this point, we consider the following example:

Task 3.17 (Windup in the DC motor)
Refer to the DC motor from Task 1.4. A speed PI controller requests an armature voltage u_c, but the inverter can apply only $u \in [-U_{\max}, U_{\max}]$. Explain why a large speed step can create windup.

Solution to Task 3.17: For a large speed step, the speed error is initially large. The PI controller increases its desired voltage u_c, but the inverter saturates at $U_{\max}$. While the real voltage cannot increase further, the integrator may continue to integrate the positive speed error. Thus, u_c moves far beyond $U_{\max}$. When the speed approaches the reference and the error becomes smaller or negative, the integrator must first unwind before the desired voltage returns to the admissible range. This causes overshoot and slow recovery.

Here, we observed that input saturation is not only an actuator problem, but also affects the controller state. To compensate for this wind up, an we need to modify the controller dynamics and not only clip the final input to create a so called *anti-windup* strategy.
The simplest anti-windup mechanism is back-calculation. It feeds the difference between real and desired input back to the integrator.

Definition 3.18 (Back-calculation anti-windup).
Consider a PI controller with error e, desired input u_c, applied input $u = \mathrm{sat}(u_c)$. By introducing

an integrator state z such that

$$u_c(t) = K_P e(t) + z(t), \tag{3.14}$$
$$\dot{z}(t) = K_I e(t) + K_\mathrm{a}\left(u(t) - u_c(t)\right), \tag{3.15}$$

we define the *anti-windup gain* $K_\mathrm{a} > 0$.

The idea of the back-calculation is sketched in Figure 3.7.

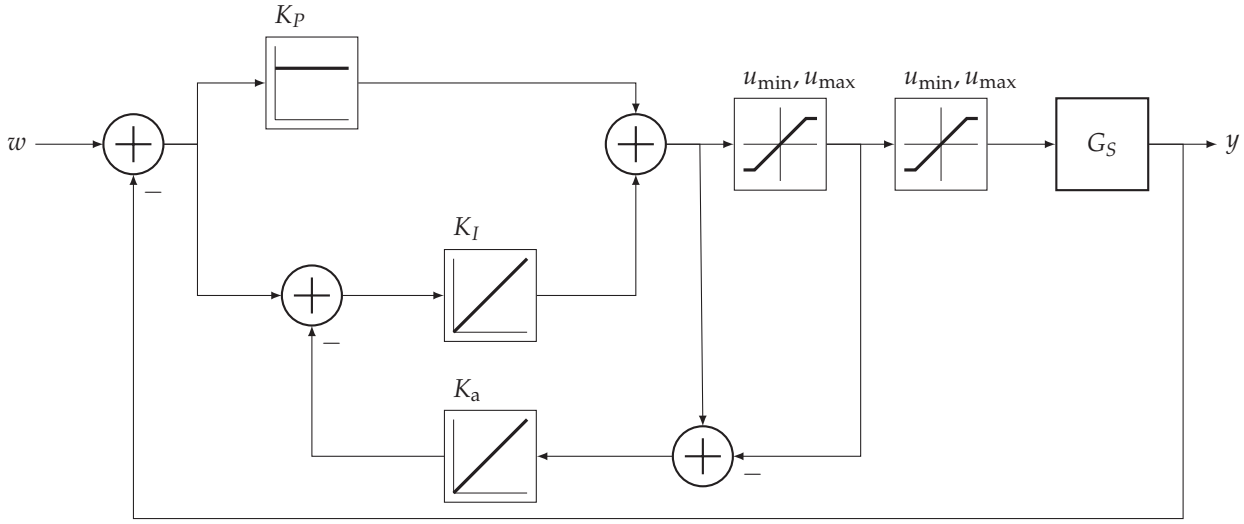


Figure 3.7: Block diagram back-calculation anti-windup

Based on anti-windup, we can derive the following local result. Note that global stability of an arbitrary saturated feedback loop is not shown here and would in this generality be too broad.

Theorem 3.19 (Integrator unwinding under back-calculation).
Assume the back-calculation controller from Definition 3.18 to be given. Suppose a zero error case $e(t_0) = 0$ and that the actuator is held at a constant saturated value $\bar{u}$. Then the desired input u_c converges exponentially to $\bar{u}$ with rate K_a.

Proof. For $e(t_0) = 0$, we have that (3.14) gives $u_c = z(t_0)$. With $u = \bar{u}$, we obtain from (3.15)

$$\dot{z}(t) = K_\mathrm{a}(\bar{u} - z(t)).$$

Thus, $z(t) - \bar{u}$ satisfies $\dot{z}(t) - 0 = -K_\mathrm{a}(z(t) - \bar{u})$. Hence,

$$z(t) - \bar{u} = \exp\left(-K_\mathrm{a}(t - t_0)\right)\left(z(t_0) - \bar{u}\right).$$

Since $u_c = z(t_0)$, the desired input converges exponentially to the saturated input. □

Hence, we obtain that anti-windup creates a recovery path for the controller state. The larger K_a is, the faster the stored integral contribution is reduced. Very large values, however, may introduce additional sensitivity and should be tuned with the actuator and measurement dynamics in mind.
In production practice, a heater may be fully on for a long time after a large setpoint step. Anti-windup prevents the integral part from demanding impossible heat power far beyond the hardware limit.
Similarly for an automotive application, propulsion or braking torque is limited by tire-road friction and comfort constraints. Anti-windup helps the controller recover when the requested acceleration cannot be delivered.

From a practical side, Algorithm 5 provides insights into procedures to design an anti-windup tuning.

Algorithm 5 Back-calculation anti-windup tuning

Input: PI or PID controller $G_R : \mathbb{C} \to \mathbb{C}$ with actuator saturation
1: **procedure** DESIGN ANTI-WINDUP CONTROL($G_R(s)$)
2: Implement the saturation block explicitly and distinguish u_c from u.
3: Generate feedback on the difference $u - u_c$.
4: Start with a moderate anti-windup gain (e.g. order of the inverse integral time).
5: Increase the gain if overshoot is dominated by slow unwinding.
6: Reduce the gain if noise, delay, or actuator dynamics produce oscillatory recovery.
7: **end procedure**
Output: Anti-windup control $G_R(s)$

The motivation for Algorithm 5 is that anti-windup is not a replacement for actuator-aware design. It is a safeguard for the controller state. The algorithm therefore starts from the physical saturation block and only then modifies the integrator.
Hence, anti-windup is tuned as part of the implementation. A purely linear design can look good on paper but fail in experiments if the actuator saturates regularly. Table 3.3 summarizes these last ideas.

Table 3.3: Advantages and limitations of anti-windup methods

Advantage	Limitation
✓ Integrator recovery is improved after saturation.	✗ Global stability is not guaranteed automatically.
Continued on next page	

Table 3.3 – continued from previous page

Advantage	Limitation
✓ Existing PI or PID controllers can be extended.	✗ The anti-windup gain must be tuned.
✓ Overshoot caused by stored integral action is reduced.	✗ Sensor delay and actuator dynamics still matter.
✓ The implementation uses the real actuator signal.	✗ Persistent saturation indicates an undersized actuator or an infeasiblity.

3.4 Bumpless Transfer

In practical applications, controllers may be switched during operation, e.g. between two types of controllers or manual/automatic mode. These switches are not the exception but instead occur in general. Examples of manual/automatic modes switches can be found in plants, which are typically run in manual mode during commissioning, but are switched to automatic mode in operations. Switching between two types of controllers are found, e.g., in vehicle functions switching between a comfort controller and a safety fallback.
The arising difficulty of such a switch is that different controllers most likely produce different inputs at the switching time. Hence, a sudden input jump occurs in the input signal. The latter is a so called *bump*. To avoid such a case, we define the following:

Definition 3.20 (Bumpless transfer).
Let t_s be a switching time and $u^-(t_s)$, $u^+(t_s)$ be the inputs requested immediately before and after switching. The transfer is called bumpless if

$$u^+(t_s) = u^-(t_s). \tag{3.16}$$

The main idea is to initialize the internal controller state such that the new controller starts from the currently applied input. To this end, an I-like behavior is most appropriate as we can see by the following:

Theorem 3.21 (Bumpless initialization of a PI controller).
Consider a PI controller

$$u(t) = K_P e(t) + z(t), \tag{3.17}$$
$$\dot{z}(t) = K_I e(t). \tag{3.18}$$

Suppose the input applied before switching is $u^-(t_s)$, and the error at the switching time is $e(t_s)$. If the integrator state is initialized as

$$z(t_s) = u^-(t_s) - K_P e(t_s), \tag{3.19}$$

then the transfer to the PI controller is bumpless.

Proof. At t_s, the new controller output is

$$u^+(t_s) = K_P e(t_s) + z(t_s).$$

Substituting (3.19) gives

$$u^+(t_s) = K_P e(t_s) + u^-(t_s) - K_P e(t_s) = u^-(t_s).$$

showing the assertion. □

Within the latter result, we utilized an internal state in order to match old/new control input. To illustrate the latter, we reconsider our running example.

Task 3.22 (Bumpless speed-control activation)
For the DC motor speed controller in Task 3.17, assume manual operation applies $u^-(t_s) = 8\,\mathrm{V}$. At the switching time, the speed error is $e(t_s) = 0.4$, and the proportional gain is $K_P = 5$. Compute the required integrator state for bumpless transfer.

Solution to Task 3.22: Using Theorem 3.21, we obtain

$$z(t_s) = u^-(t_s) - K_P e(t_s) = 8 - 5 \cdot 0.4 = 6.$$

Thus, the PI integrator should be initialized with $z(t_s) = 6\,\mathrm{V}$. The new controller then produces $5 \cdot 0.4 + 6 = 8\,\mathrm{V}$ at the switching instant.

As we have seen, bumpless transfer is an algebraic consistency condition at the switching instant. After the switch, the controller dynamics continue from this consistent state.
Upon implementation in a production setting, an operator may move a valve manually during startup. When automatic control is activated, the controller should start from the current valve position, not from an unrelated internal integral value.
Similarly in an ADAS setting, switching from lane-centering to a fallback steering controller should not create a sudden steering torque jump. The fallback controller state must be initialized consistently.

Within practice, a respective workflow to generate a bunpless transfer may proceed as in Algorithm 6.

Algorithm 6 Design of bumpless transfer

Input: Two inputs $G_{R,1}, G_{R,2} : \mathbb{C} \to \mathbb{C}$
1: **procedure** GENERATE BUMPLESS SWITCHING($G_{R,1}(s), G_{R,2}(s)$)
2: Measure or reconstruct the input $u^-(t_s)$ applied just before switching.
3: Evaluate all static terms of the new controller at the switching time.
4: Choose the internal controller state $z(t_s)$ so that the new output equals the old input.
5: Activate the new controller and continue its state dynamics from the initialized state.
6: Check rate limits if equality of input values is not sufficient for comfort or safety.
7: **end procedure**
Output: Continuity in internal controller state $z(t)$

The motivation for Algorithm 6 is that practical controllers usually contain states. Safe switching therefore requires state initialization, not only changing a software flag. Hence, the workflow addresses the input value at the switching instant, yet it does not guarantee identical future trajectories. If the new controller has very different dynamics, the input can still change rapidly after the switch.

Table 3.4: Advantages and limitations of bumpless transfer

Advantage	Limitation
✓ Jumps at switching can be avoided.	✗ Future input changes may still be fast.
✓ Manual-to-automatic transfer becomes safer.	✗ Internal states must be available or reconstructable.
✓ Controller switching can be organized systematically.	✗ Rate limits may require additional smoothing.

Continued on next page

Table 3.4 – continued from previous page

Advantage	Limitation
✓ The method is algebraically simple for PI control.	✗ Higher-order controllers require more state matching.

3.5 Smith Predictor

Yet another difficulty for feedback generation occurs in the presence of time delays, which cause the measured output to react late on inputs. A controller that sees only delayed error information tends to be conservative. For one, this is systematic as dead times reduce the phase in the frequency response (while the magnitude remains unchanged), which contributes to destabilization and thus limits the maximum possible loop gain and bandwidth. On the other hand, delays may be due to the fact that many control methods are based on fractional-rational transfer functions and cannot directly be applied to systems with dead time. While the first issue cannot be overcome, the second may be circumvented by the so called *Smith predictor*.
The Smith predictor uses a model of the delay-free plant to predict the output that would be available without delay. In this way, the controller can be designed mainly for the delay-free dynamics, while the physical delay remains in the reference-to-output response.

Definition 3.23 (Delayed plant).
A plant is called a *delayed plant* if it can be decomposed as

$$G_S(s) = G_0(s)\exp(-K_T s), \tag{3.20}$$

where $K_T > 0$ is the dead time and G_0 is the delay-free plant part.

Following the definition, the idea of the Smith predictor is to modify to loop as schematically sketched in Figure 3.8 to Figure 3.9, i.e. to develop the control for the delay-free plant.

Theorem 3.24 (Nominal Smith-predictor closed-loop transfer function).
Consider the delayed plant from Definition 3.23. Let $\tilde{G}_R : \mathbb{C} \to \mathbb{C}$ *be a controller designed for the delay-free plant* $G_0 : \mathbb{C} \to \mathbb{C}$. *The equivalent Smith-predictor controller in a standard*

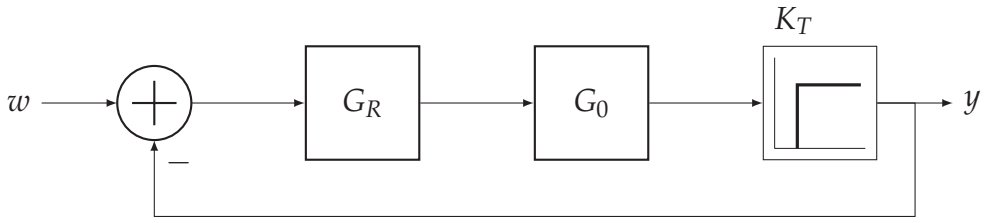


Figure 3.8: Block diagram delay plant decomposition

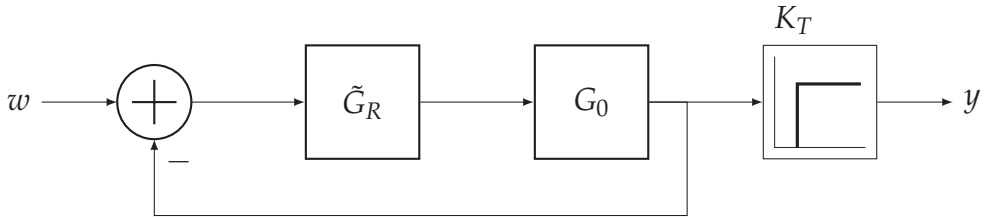


Figure 3.9: Block diagram delay plant decomposition for Smith predictor

feedback loop is

$$G_{\text{Smith}}(s) = \frac{\tilde{G}_R(s)}{1 + \tilde{G}_R(s)G_0(s)\,(1 - \exp(-K_T s))}. \tag{3.21}$$

With a perfect model, the reference-to-output transfer function is

$$G(s) = \frac{G_R(s)G_0(s)}{1 + G_R(s)G_0(s)} \exp(-K_T s). \tag{3.22}$$

Proof. As for the Smith-predictor, the signal is fed back before entering the latency, we obtain two different closed loop descriptions. Suppose both control loops to be identical, then we obtain

$$\frac{G_R(s)G_0(s)\exp\left(-K_T s\right)}{1 + G_R(s)G_0(s)\exp\left(-K_T s\right)} = \frac{\tilde{G}_R(s)G_0(s)\exp\left(-K_T s\right)}{1 + \tilde{G}_R(s)G_0(s)}.$$

Hence, by solving for $G_{\text{Smith}}(s) = G_R(s)$ we obtain the assertion. □

The interpretation is that the Smith predictor does not remove the physical delay from the output. It removes the delay from the nominal characteristic equation used for controller design as shown. Model errors in the delay or in G_0 can therefore degrade performance strongly, which is also viewable from the new control structure illustrated in Figure 3.10

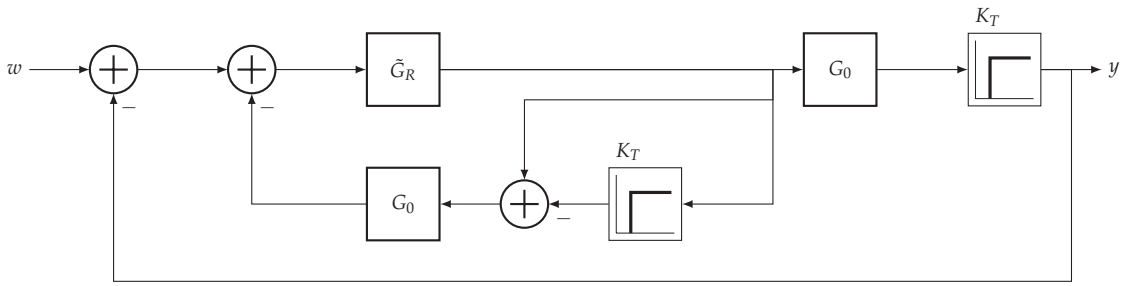


Figure 3.10: Block diagram delay plant decomposition for explicit Smith predictor

Task 3.25 (Smith predictor for a delayed first-order plant)
Consider

$$G_S(s) = \frac{1}{1+s}\exp(-3s),$$

and a delay-free controller $G_R(s) = K_P$. Compute the equivalent Smith-predictor controller.

Solution to Task 3.25: Here, $G_0(s) = 1/(1+s)$ and $K_T = 3$. Thus,

$$G_{\text{Smith}}(s) = \frac{K_P}{1 + K_P \dfrac{1}{1+s}\left(1-\exp(-3s)\right)} = \frac{K_P(1+s)}{1+s+K_P\left(1-\exp(-3s)\right)}.$$

Hence, Smith-predictor controller may be more complicated than the delay-free one. The advantage, however, is not simplicity but improved nominal design freedom for delayed plants.

Remark 3.26
The Smith-predictor is a very special type of a so called internal model controller. For the latter, we utilize a model of the entire plant $G_S(s)$ instead of the latency component. If the model was perfect, then the feedback transforms into a feedforward. Hence, the model quality is essential for such a approach. Here, we do not go into details on internal model controllers as our next step is the even more complex version of internal model in state space.

Table 3.5: Advantages and limitations of Smith predictor

Advantage	Limitation
✓ Smith prediction improves nominal delay handling.	✗ Delay-model errors can strongly reduce performance.
✓ The delay-free controller can be tuned more aggressively.	✗ The physical output delay remains.

The chapter shows that complex control structures do not replace the basic feedback loop. They organize additional information around it. MIMO control organizes several inputs and outputs. Decoupling uses model knowledge to recover SISO design paths. Anti-windup and bumpless transfer make controllers compatible with real actuators and operating modes. Last, Smith prediction uses plant model to handle delay which could be extended to internal model control to separate nominal behavior from correction. Across all methods, the same warning applies: additional structure improves performance only if its assumptions remain compatible with realizability, actuator limits, and model uncertainty.

CHAPTER 4

STABILITY AND OBSERVABILITY

In the previous chapters we focused on how feedback structures, nonlinear elements, and multivariable couplings are represented and implemented. Now, we change our perspective and ask whether the resulting dynamic system has the structural properties. Based on these properties, we will derive whether feedback and estimation are possible. In particular, these properties can be checked without solving the dynamics of the system, which allows us to build control and observer elements on a structural basis.

The two central properties are stability and observability. Stability describes whether trajectories remain controlled and converge to the desired operating point. Observability describes whether the internal state can be reconstructed from measured input-output data. These properties are essential for the later design chapters for optimal stabilizing controllers and optimal observers.

To this end, Section 4.1 we shortly recap the state-space viewpoint used throughout the remaining chapters, which was outlined in Control Engineering 1. Instead of describing a system only by input-output transfer functions, we represent its *internal memory* by the state $\mathbf{x}(\cdot)$. This allows us to describe multi-state dynamics, initial conditions, trajectories, and sampled or continuous-time evolution in one common notation. Furthermore, we derive the solution formulas for linear time-invariant systems. These formulas are the basis for all later results because they show explicitly how the initial state $\mathbf{x}_0$ and the input $\mathbf{u}(\cdot)$ affect the future state.

In Section 4.2 we study whether the state $\mathbf{x}(\cdot)$ will stay close to an *operating point* $\mathbf{x}^\star$ and whether the system behavior can be modified accordingly by suitable inputs $\mathbf{u}(\cdot)$. The first concept is called *stability* and describes the natural behavior of the autonomous system. The second one is termed *controllability* and describes the authority of the input over the state. Thereafter, we connect these ideas to *stabilizability* and *state feedback*. In particular, these concepts explain when unstable modes can be influenced by the input and how feedback can be used to assign closed-loop poles. In the following chapters, we will use this structural foundation to derive

optimal stabilizing controllers.
Last, in Section 4.3 we develop the dual question. Instead of asking whether inputs can influence the state, we ask whether outputs contain enough information to reconstruct it. We start from *distinguishability* of initial states and then derive *observability criteria* for linear systems. Duality shows that observability and controllability have the same algebraic structure. Similar to controllability and stabilizability, detectability weakens observability as unobservable modes are allowed if they are already stable. Lateron, we will apply the latter in observers.
Throughout this chapter, we will utilize our running DC-motor example from Task 1.4. In the previous chapters, the motor was mainly used from an input-output perspective. We asked how the armature voltage influences torque, speed, or position, and how disturbances such as a load torque can be rejected by suitable control-loop structures. As we will see, the state-space viewpoint refines this description. This example will be useful to us because it illustrates the main ideas of the chapter in a physically transparent way. Stability asks whether the motor state remains close to an operating point, for example constant speed or constant position. Controllability asks whether the voltage input can move the electrical and mechanical states as desired. Observability asks whether the available measurements contain enough information to reconstruct the unmeasured states. Detectability asks whether unmeasured state components are harmless because their effects decay naturally.
Similar to the previous chapter, we will utilize the following standing assumptions unless stated otherwise.

Assumption 4.1
All finite-dimensional state spaces are real vector spaces $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ with the Euclidean norm $\|\cdot\|_2$. Continuous-time inputs $\mathbf{u}(t)$ are assumed to be piecewise continuous on compact intervals with time indicated by t, and discrete-time inputs $\mathbf{u}(k)$ are arbitrary sequences of compatible dimension with time index k. For stability statements, equilibria are shifted to the origin whenever this simplifies notation.

4.1 Recap on State-Space Systems and Solutions

Frequency-domain transfer functions describe the input-output behavior of linear systems. They are well suited for the loop structures. However, stability and observability require access to the internal memory of the system. The term *system*, however, is typically not defined clearly. In certain areas, a system stands for a connected graph, a dynamically evolving entity or even a simulation or an optimization. While the intention of the latter are quite distinct, they all can be boiled down to the following:

A system is the connection of different interacting components to realize given tasks.

As in the input–output setting illustrated in Figure 4.1, we formally define the following:

Definition 4.2 (System).
Consider two sets $\mathcal{U}$ and $\mathcal{Y}$. Then a map $\Sigma : \mathcal{U} \to \mathcal{Y}$ is called a system.

Still, we need to extend our common understanding of inputs and outputs. An element from the input set $\mathbf{u} \in \mathcal{U}$ is called an input, acts from the environment to the system and is not dependent on the system itself or its properties. As we have seen in Section 1.3, there are two types of inputs. First those, which are used to specifically manipulate (or control) the system, and secondly the ones, which are not manipulated on purpose. We call the first ones *control or manipulation inputs*, and refer to the second ones as *disturbance inputs*.
Similarly, an element from the output set $\mathbf{y} \in \mathcal{Y}$ is called an output. In contrast to an input, the output is generated by the system and influences the environment. We distinguish output variables whether we measure them or not. We call the measured ones *measurement outputs*.

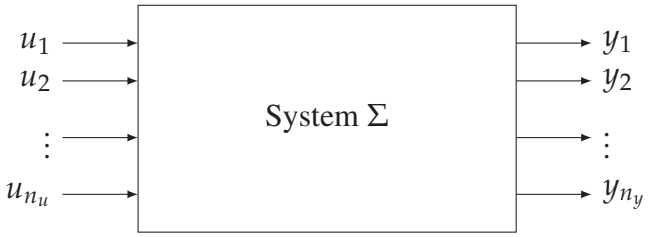


Figure 4.1: Term of a system

In the literature, certain classes of systems are considered:

- A system is called *linear* if it is linear in inputs and outputs, and *nonlinear* if it is not linear in either the inputs or outputs.
- A system is *time invariant* if all parameters are constants, and *time varying* if at least one parameter is time-dependent.
- Systems can be classified as *static* or *dynamic* depending on whether their outputs depend solely on the input at the same time instant or also on its history.
- *Causal* systems depend only on the history of the inputs, while *acausal* systems include future values.

- If inputs are mapped directly to outputs, then the map is called *input output system*. If the input triggers changes of an internal variable and the output depends on the latter, then the map is called *state space system*.
- If time is measured continuously, the system is said to be in *continuous time*. If time is sampled, it is referred to as *discrete time system*.

To assess systems, we require a formal notation of time:

Definition 4.3 (Time).
A *time set* $\mathcal{T}$ is a subgroup of $(\mathbb{R}, +)$.

Within the lecture, we focus on state space systems, which are time invariant, dynamic and causal. To introduce such systems, we first need to define what we referred to as internal variable:

Definition 4.4 (State).
Consider a system $\Sigma : \mathcal{U} \to \mathcal{Y}$. If the output $\mathbf{y}(t)$ uniquely depends on the history of inputs $\mathbf{u}(\tau)$ for $t_0 \leq \tau \leq t$ and some $\mathbf{x}(t_0)$, then the variable $\mathbf{x}(t)$ is called state of the system and the corresponding set $\mathcal{X}$ is called state set.

Within Definition 4.4, input, output and state refer to tuples

$$\mathbf{u} = [u_1 \; u_2 \; \dots \; u_{n_u}]^\top \tag{4.1a}$$

$$\mathbf{y} = \left[y_1 \; y_2 \; \dots \; y_{n_y}\right]^\top \tag{4.1b}$$

$$\mathbf{x} = [x_1 \; x_2 \; \dots \; x_{n_x}]^\top . \tag{4.1c}$$

where u_j is an element within the subset j of the input set $\mathcal{U}$, y_j is an element within the subset j of the output set $\mathcal{Y}$ and x_j is an element within the subset j of the state set $\mathcal{X}$.
This definition is intentionally broad. It covers electrical, mechanical, thermal, and software-controlled systems. In engineering models, the next step is to describe how the state evolves.

Definition 4.5 (State space – continuous time system).
Consider a system $\Sigma : \mathcal{U} \to \mathcal{Y}$ in continuous time $\mathcal{T} = \mathbb{R}$ satisfying the property from Definition 4.4. If $\mathcal{X}$ is a vector space, then we call it *state space* and refer to

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{4.2a}$$

$$\mathbf{y}(t) = h(\mathbf{x}(t), \mathbf{u}(t), t). \tag{4.2b}$$

as *continuous time system*. Moreover, $\mathbf{u}$, $\mathbf{y}$ and $\mathbf{x}$ are called *input*, *output* and *state* of the system.

The state of a system at time instant t can then be depicted as a point in the n_x–dimensional state space. The curve of points for variable time t in the state space is called *trajectory* and is denoted by $\mathbf{x}(\cdot)$.

Remark 4.6
Systems with infinite dimensional states are called distributed parametric systems and are described, e.g., via partial differential equations. Examples of such systems are beams, boards, membranes, electromagnetic fields, heat etc..

Similarly, in discrete time $\mathcal{T} = \mathbb{Z}$ we define the following:

Definition 4.7 (State space – discrete time system).
Consider a system $\Sigma : \mathcal{U} \to \mathcal{Y}$ in discrete time $\mathcal{T} = \mathbb{Z}$ satisfying the property from Definition 4.4. If $\mathcal{X}$ is a vector space, then we refer to

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k), k), \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{4.3a}$$
$$\mathbf{y}(k) = h(\mathbf{x}(k), \mathbf{u}(k), k). \tag{4.3b}$$

as *discrete time system*. Again, $\mathbf{u}$, $\mathbf{y}$ and $\mathbf{x}$ are called *input*, *output* and *state* of the system.

While we have $t \in \mathbb{R}$ in continuous time, for discrete time systems the matter of time refers to an index $k \in \mathbb{Z}$. Therefore, trajectories are no longer represented as curves but as sequences of points within their respective set. Digitalization typically results in discrete time systems, which are obtained by sampling continuous time systems using an A/D and D/A converter. The outcome of this process is a time grid. The simplest case is equidistant sampling with a fixed sampling time T, which produces

$$\mathcal{T} := \{t_k \mid t_k := t_0 + kT\} \subset \mathbb{R}. \tag{4.4}$$

where t_0 is some fixed initial time stamp. Apart from equidistant sampling, other types such as event based or sequence based are possible.
Note that in both discrete and continuous time, the map shows a *flow* within the state space. A trajectory is obtained by specifying an initial value and an input sequence. Figure 4.2 illustrates the concept of flow and trajectory. In this case, the flow is colored to indicate its intensity whereas the

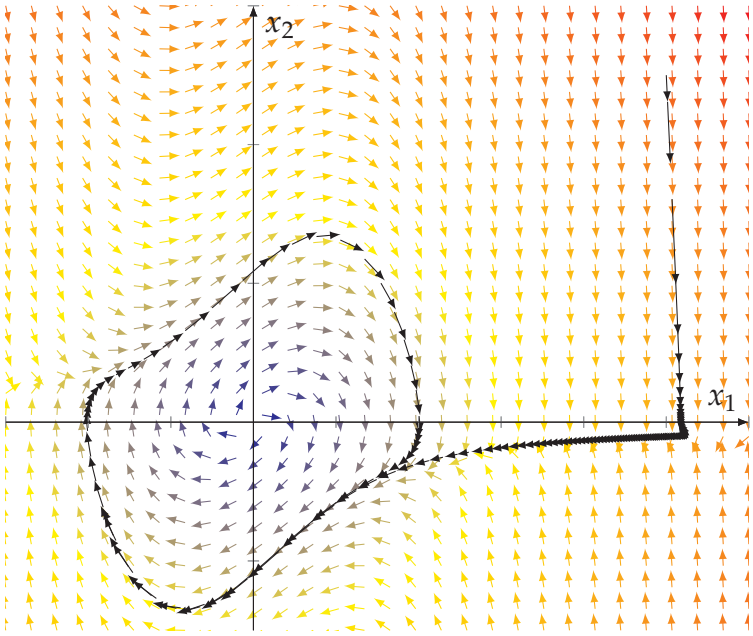


Figure 4.2: Sketch of a dynamic flow and a trajectory

arrows indicate its direction. The trajectory is evaluated for a specific initial value and „follows“ the flow accordingly.

As stated in the introduction, stability refers to the ability to control a system to achieve a specific goal, such as boundedness or convergence. In order to achieve this, the input must have an impact on the states, either directly or indirectly. Observability, on the other hand, refers to the ability to identify the status of a system, that is, to be able to measure states directly or indirectly. This context is illustrated in Figure 4.3. The figure demonstrates that not all states can be manipulated, even indirectly, and not all states can be observed. However, we will see that even in this case methods can be applied to ensure stability and observability.

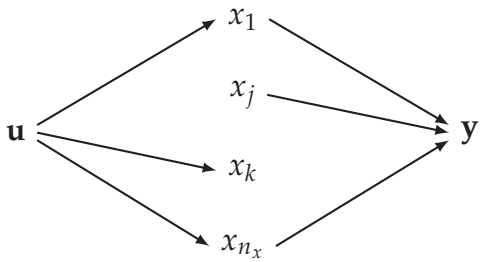


Figure 4.3: Flow of information for controllability and observability

In order to discuss the terms stability and observability in detail, we focus on the special class of linear control systems:

Definition 4.8 (Linear control system).
For matrices $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$, $D \in \mathbb{R}^{n_y \times n_u}$, we call the system

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0 \tag{4.5a}$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) \tag{4.5b}$$

linear time invariant control system in continuous time with initial value $\mathbf{x}_0 \in \mathbb{R}^{n_x}$. The time discrete equivalent reads

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k), \qquad \mathbf{x}(0) = \mathbf{x}_0 \tag{4.6a}$$
$$\mathbf{y}(k+1) = C\mathbf{x}(k) + D\mathbf{u}(k). \tag{4.6b}$$

Note that the number of states is not the same as the number of inputs or outputs. A system may have many states and still be SISO if $n_u = 1$ and $n_y = 1$. Conversely, a MIMO system from Definition 3.4 may have a small state dimension but several manipulated and controlled variables. In short, state dimension describes internal memory whereas MIMO dimension describes external signal interfaces.

Task 4.9 (Running example: State-space model of the DC motor)
Consider the DC motor from Task 1.4. Use the state vector

$$\mathbf{x}(t) = \begin{bmatrix} i_\mathrm{a}(t) & \omega(t) & \varphi(t) \end{bmatrix}^\top$$

and the armature voltage $\mathbf{u} = u_\mathrm{a}$ *as input. Write the system in the form* (4.5) *if the load torque is treated as an additional disturbance input.*

Solution to Task 4.9: From the electrical and mechanical equations, we obtain

$$\begin{aligned}
\dot{i}_\mathrm{a}(t) &= -\frac{R_\mathrm{a}}{L_\mathrm{a}} i_\mathrm{a}(t) - \frac{K_\mathrm{e}}{L_\mathrm{a}} \omega(t) + \frac{1}{L_\mathrm{a}} u_\mathrm{a}(t), \\
\dot{\omega}(t) &= \frac{K_\mathrm{t}}{J_\mathrm{m}} i_\mathrm{a}(t) - \frac{b_\mathrm{m}}{J_\mathrm{m}} \omega(t) - \frac{1}{J_\mathrm{m}} d_M(t), \\
\dot{\varphi}(t) &= \omega(t).
\end{aligned}$$

Thus, we have

$$\dot{\mathbf{x}}(t) = \underbrace{\begin{bmatrix} -\frac{R_\mathrm{a}}{L_\mathrm{a}} & -\frac{K_\mathrm{e}}{L_\mathrm{a}} & 0 \\ \frac{K_\mathrm{t}}{J_\mathrm{m}} & -\frac{b_\mathrm{m}}{J_\mathrm{m}} & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{A} \mathbf{x} + \underbrace{\begin{bmatrix} \frac{1}{L_\mathrm{a}} \\ 0 \\ 0 \end{bmatrix}}_{B} \mathbf{u}(t) + \underbrace{\begin{bmatrix} 0 \\ -\frac{1}{J_\mathrm{m}} \\ 0 \end{bmatrix}}_{B_d} d_M(t).$$

If the output is the position, then $C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$. If the output is the speed, then $C = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$.

The class of LTI systems is of particular interest as we can directly give its solution, which allows us to use it as basic tool for all following structural properties. It shows how the initial condition and the input contribute separately to the state trajectory.

Theorem 4.10 (Solution of a linear time-invariant system).
Consider the continuous-time LTI system (4.5). *For every initial value* $\mathbf{x}(t_0) = \mathbf{x}_0$ *and every piecewise continuous input* $\mathbf{u}(\cdot)$*, the unique solution is*

$$\mathbf{x}(t; t_0, \mathbf{x}_0, \mathbf{u}) = \exp\left(A(t-t_0)\right)\mathbf{x}_0 + \int_{t_0}^{t} \exp\left(A(t-\tau)\right) B\mathbf{u}(\tau)\, d\tau. \tag{4.7}$$

For the discrete-time LTI system (4.6)*, the solution is*

$$\mathbf{x}(k) = A^k \mathbf{x}_0 + \sum_{j=0}^{k-1} A^{k-1-j} B\mathbf{u}(j). \tag{4.8}$$

Proof. For continuous time, multiply the differential equation by the inverse transition matrix $\exp(-A(t-t_0))$, integrate, and multiply back by $\exp(A(t-t_0))$. For discrete time, repeatedly substitute the recursion into itself. Both arguments give the displayed formulas and uniqueness follows from the linear dynamics. □

The interpretation of Theorem 4.10 is fundamental: The first term in (4.7) is the free response due to the initial state $\mathbf{x}_0$, and the integral term is the forced response due to the input $\mathbf{u}(\cdot)$. In the upcoming section, we utilize the latter and will see that stability mainly concerns the free response whereas controllability concerns what the forced response can achieve.
In a production setting, the free response may describe how an initial temperature or tension deviation decays without further actuation. The forced response describes how heaters, motors, or valves can move the process.

Similarly in an AD function, the free response describes the natural vehicle motion after an initial yaw-rate deviation. The forced response describes how steering, braking, or propulsion can change the motion.

The solution formula also explains why linear systems are easier to analyze than nonlinear systems. In particular, we observet that the effects of initial condition and of the input appear additively. This gives the following superposition and time shifting property:

Corollary 4.11 (Superposition and time shift).
For the continuous-time LTI system (4.5)*, the state trajectory satisfies*

$$\mathbf{x}(t;t_0,\mathbf{x}_0,\mathbf{u}) = \mathbf{x}(t;t_0,\mathbf{x}_0,0) + \mathbf{x}(t;t_0,0,\mathbf{u}). \tag{4.9}$$

Moreover, for every $\tau \in [t_0,t]$,

$$\mathbf{x}(t;t_0,\mathbf{x}_0,\mathbf{u}) = \mathbf{x}(t;\tau,\mathbf{x}(\tau;t_0,\mathbf{x}_0,\mathbf{u}),\mathbf{u}). \tag{4.10}$$

Proof. Superposition (4.9) is obtained by separating the two terms in (4.7). The time-shift property (4.10) follows from the semigroup property of the matrix exponential and by restarting the solution at the intermediate state $\mathbf{x}(\tau)$. □

In particular, the superposition principle allows us to separate the uncontrolled solution ($\mathbf{u} = 0$) and the unforced solution ($\mathbf{x}_0 = 0$) and study initial-state effects and input effects separately. This separation is used repeatedly in the controllability and observability arguments below.

4.2 Stability and Controllability

A controller is useful only if it can shape the relevant system behavior. Stability describes the behavior without corrective action or after a feedback loop has been closed. Controllability describes whether inputs can move the state through the relevant state-space directions. Stabilizability is the practically important bridge since not every state direction must be controllable if all uncontrollable directions are already stable.
Starting with stability, this property is linked to specific points in the state space known as *operating points*. At these points, the system's dynamics should come to a rest. In other words, the input (as a control) must be selected appropriately to ensure that the system remains stable.

Definition 4.12 (Operating point).
For the continuous-time system (4.2), a pair $(\mathbf{x}^\star, \mathbf{u}^\star)$ is called an operating point if

$$f(\mathbf{x}^\star, \mathbf{u}^\star) = 0. \tag{4.11}$$

For the discrete-time system (4.3), it is called an operating point if

$$f(\mathbf{x}^\star, \mathbf{u}^\star) = \mathbf{x}^\star. \tag{4.12}$$

If (4.11) or (4.12) hold true respectively for any $\mathbf{u}^\star \in \mathcal{U}$, then the operating point is called *strong* or *robust* operating point.

Based on this definition, the property of stability can be characterized by boundedness and convergence of solutions:

Definition 4.13 (Stability and Controllability).
For a system (4.2) we call $\mathbf{x}^\star$

- *strongly* or *robustly stable* operating point if, for each $\varepsilon > 0$, there exists a real number $\delta = \delta(\varepsilon) > 0$ such that for all $\mathbf{u}$ we have

$$\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq \delta \implies \|\mathbf{x}(t) - \mathbf{x}^\star\| \leq \varepsilon \qquad \forall t \geq 0 \tag{4.13}$$

- *strongly* or *robustly asymptotically stable* operating point if it is stable and there exists a positive real constant r such that for all $\mathbf{u}$

$$\lim_{t\to\infty} \|\mathbf{x}(t) - \mathbf{x}^\star\| = 0 \tag{4.14}$$

  holds for all $\mathbf{x}_0$ satisfying $\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq r$. If additionally r can be chosen arbitrary large, then $\mathbf{x}^\star$ is called *globally strongly* or *robustly asymptotically stable*.

- *weakly stable* or *controllable* operating point if, for each $\varepsilon > 0$, there exists a real number $\delta = \delta(\varepsilon) > 0$ such that for each $\mathbf{x}_0$ there exists a control $\mathbf{u}$ guaranteeing

$$\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq \delta \implies \|\mathbf{x}(t) - \mathbf{x}^\star\| \leq \varepsilon \qquad \forall t \geq 0. \tag{4.15}$$

- *weakly asymptotically stable* or *asymptotically controllable* operating point if there exists a control $\mathbf{u}$ depending on $\mathbf{x}_0$ such that (4.15) holds and there exists a positive constant r such

that

$$\lim_{t\to\infty} \|\mathbf{x}(t) - \mathbf{x}^\star\| = 0 \qquad \forall \|\mathbf{x}_0 - \mathbf{x}^\star\| \leq r. \tag{4.16}$$

If additionally r can be chosen arbitrary large, then $\mathbf{x}^\star$ is called *globally asymptotically stable*.

The first computable stability result uses the eigenvalues of the system matrix. It applies to the autonomous linear system and is the state-space counterpart of looking at poles in the frequency domain. In the linear case, we can derive sufficient properties for the system to be stable using the Eigenvalue criterion.

Theorem 4.14 (Eigenvalue criterion).
Consider the continuous-time autonomous LTI system

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t).$$

The equilibrium $\mathbf{x}^\star = 0$ is stable if and only if every eigenvalue $\lambda \in \mathbb{C}$ of A satisfies $Re(\lambda) \leq 0$ and every eigenvalue with $Re(\lambda) = 0$ is semisimple. The equilibrium is asymptotically stable if and only if every eigenvalue λ of A satisfies $Re(\lambda) < 0$.

Proof. Transform A to Jordan form. Each Jordan block contributes a factor $\exp(\lambda t)$ and, for nontrivial blocks, polynomial terms in t. Negative real parts force decay. Positive real parts force growth. Eigenvalues on the imaginary axis are bounded only if their Jordan blocks have no polynomial growth. Hence stability and asymptotic stability follow from the stated eigenvalue conditions. □

The interpretation is that stability is a property of modes. Each eigenvalue represents a state-space mode. A mode with negative real part decays, a mode with positive real part grows, and a nonsemisimple mode on the imaginary axis grows polynomially.
In a production system, such an unstable thermal mode may describe a temperature deviation that grows without sufficient cooling or feedback. A stable mode describes a deviation that naturally decays.
Similarly in an driving function, an unstable mode may correspond to a yaw or sideslip behavior that grows unless corrected by steering or braking. A stable mode decays after a perturbation.
An important class of systems occurs if all Eigenvalues of a matrix A exhibit negative real part.

Definition 4.15 (Hurwitz matrix).
A matrix A is called Hurwitz if every eigenvalue λ of A satisfies $\mathrm{Re}(\lambda) < 0$.

The eigenvalue criterion tells us how to check stability. Feedback design asks whether we can change the eigenvalues by choosing a state feedback law. Hence, given the Eigenvalue criterion, it is straightforward to derive an input, which induces the stability property.

Theorem 4.16 (Linear state feedback).
Consider the continuous-time LTI system (4.5) *with state feedback*

$$\mathbf{u}(t) = F\mathbf{x}(t).$$

Then the closed-loop equilibrium $\mathbf{x}^\star = 0$ *is asymptotically stable if and only if* $A + BF$ *is Hurwitz.*

Proof. Substitution of $\mathbf{u} = F\mathbf{x}$ gives the autonomous closed-loop system $\dot{\mathbf{x}} = (A + BF)\mathbf{x}$. Applying Theorem 4.14 proves the statement. □

The theorem directly shows that a the feedback matrix F should be chosen that the Hurwitz property of the closed-loop emerges. Yet technically, we don't know

1. whether or not it is actually possible that a feedback F can be constructed such that the conditions of Theorem 4.16 hold, or
2. how such a feedback can be constructed.

These questions lead to controllability. Since the dimension of the set reachable by the dynamics only cannot grow larger after $n_x - 1$ iterations, we introduce the so called *reachable subspace*.

Definition 4.17 (Reachability, controllability, and stabilizability).
For the continuous-time LTI system (4.5), the *reachable subspace* is

$$\mathcal{R}(A, B) := \mathrm{Im}\begin{bmatrix} B & AB & \cdots & A^{n_x-1}B \end{bmatrix}. \tag{4.17}$$

The pair (A, B) is called *controllable* if $\mathcal{R}(A, B) = \mathbb{R}^{n_x}$. It is called *stabilizable* if there exists a feedback matrix F such that $A + BF$ is Hurwitz.

To answer the first question, Kalman formulated a rank condition which turns controllability into a finite-dimensional linear-algebra test. The condition states that the input directions and their images under repeated multiplication by A must span the full state space.

Theorem 4.18 (Kalman controllability criterion).
The pair (A, B) is controllable if and only if

$$\operatorname{rk}\begin{bmatrix} B & AB & \cdots & A^{n_x-1}B \end{bmatrix} = n_x. \tag{4.18}$$

Proof. The reachable subspace is the smallest A-invariant subspace that contains the columns of B. By the Cayley–Hamilton theorem, this subspace is already generated by $B, AB, \ldots, A^{n_x-1}B$. Hence, full-state reachability is equivalent to the rank condition. □

Hence, we observe that inputs need not act on every state directly, they may also act indirectly through the dynamics. For the DC motor in Task 4.9, the voltage acts directly on the current, the current acts on speed, and speed acts on position. Therefore, the physical chain can make position controllable even though voltage does not appear directly in the position equation.

Remark 4.19
The reachable set is typically defined as the set of points, which can be reached from $\mathbf{x}_0 = 0$ within a certain time $t \geq 0$ via

$$\mathcal{R}(t) := \{\mathbf{x}(t, 0, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}\}.$$

Similarly, the controllable set refers to those points $\mathbf{x}_0$, for which a control $\mathbf{u}$ can be found to drive the solution to the origin, i.e.

$$\mathcal{C}(t) := \{\mathbf{x}_0 \mid \exists \mathbf{u} \in \mathcal{U} : \ \mathbf{x}(t, \mathbf{x}_0, \mathbf{u}) = 0\}.$$

Unfortunately, Kalman assumed that the control needs to affect all dimensions of the state space for the system to be controllable. However, if a part of the system is already controllable without the control affecting it, then only the controllability of the remaining part needs to be ensured. Therefore, Hautus introduced separability in the state space:

Theorem 4.20 (Separability).
For any system (4.5)*, which is not controllable, there exists a linear transformation T such that*

$$\tilde{A} := T^{-1}AT = \begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix}, \qquad \tilde{B} := T^{-1}B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix} \tag{4.19}$$

where (A_1, B_1) is controllable.

Proof. Follows by definition of controllability. □

Now, the idea is to simply apply the Kalman criterion to the separated part of the dynamics/state space:

Theorem 4.21 (Hautus controllability and stabilizability criteria).
The pair (A, B) is controllable if and only if

$$\operatorname{rk}\begin{bmatrix}\lambda Id - A & B\end{bmatrix} = n_x \qquad \text{for all } \lambda \in \mathbb{C}. \tag{4.20}$$

It is stabilizable if and only if (4.20) *holds for every eigenvalue λ of A with $Re(\lambda) \geq 0$.*

Proof. The rank condition fails exactly when a nonzero left eigenvector $q^\top A = \lambda q^\top$ satisfies $q^\top B = 0$. Such a mode cannot be influenced by the input. Controllability excludes this for all eigenvalues. Stabilizability only needs to exclude uncontrollable modes with nonnegative real part, because stable uncontrollable modes may remain unchanged. □

The interpretation is modal. Kalman's criterion asks whether the input-generated subspace is full. Hautus' criterion asks whether each eigenmode can be reached by the input. Stabilizability accepts modes that cannot be reached if they already decay.

After addressing whether feedback can be constructed, we will now shift our focus to computing such feedback. We will achieve this by applying basic linear algebra, which will provide us with the controllable canonical form.

Theorem 4.22 (Controllable canonical form).
Consider a system (4.5). *Then (A, B) is controllable iff there exists a linear transformation T with*

$$\tilde{A} = T^{-1}AT = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_{n_x} \end{pmatrix} \qquad \tilde{B} = T^{-1}B = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{4.21}$$

with coefficients α_j of the assigned polynomial $\Xi_A = z^{n_x} - \alpha_{n_x} z^{n_x-1} - \cdots - \alpha_2 z - \alpha_1$.

Proof sketch. The statement follows from a change of basis generated by the controllability matrix. If the pair (A, B) is controllable, then the vectors

$$B,\ AB,\ \ldots,\ A^{n_x-1}B$$

span the whole state space. Hence, they can be used to construct an invertible transformation T. In the transformed coordinates, the input vector becomes the last unit vector and the action of A shifts the basis vectors. The last row is determined by the characteristic polynomial of A, using the Cayley–Hamilton theorem. This gives the controllable canonical form in (4.21).
Conversely, if such a transformation exists, then the transformed controllability matrix has full rank because the canonical form generates all unit directions by repeated multiplication with $\tilde{A}$. Since controllability is invariant under invertible coordinate transformations, the original pair (A, B) is controllable. □

Based on the latter, we directly obtain controllability if we can assign any polynomial.

Theorem 4.23 (Assignable polynomial).
Consider a system (4.5)*. Then the pair* (A, B) *is controllable iff every polynomial of degree* n_x *is assignable.*

Proof sketch. If every polynomial of degree n_x is assignable, then in particular a polynomial with arbitrary roots can be imposed by a suitable feedback. This is only possible if no mode of A is unreachable from the input. Otherwise, an uncontrollable mode would remain unchanged under every state feedback and its corresponding factor in the closed-loop characteristic polynomial could not be assigned freely.
Conversely, assume that (A, B) is controllable. By Theorem 4.22, there exists a coordinate transformation that brings the system into controllable canonical form. In this form, the feedback entries act directly on the coefficients of the characteristic polynomial. Hence, for any desired polynomial of degree n_x, the feedback can be chosen such that the closed-loop characteristic polynomial equals this polynomial. Since polynomial assignment is invariant under invertible coordinate transformations, the same conclusion holds in the original coordinates. □

To enforce the stability property, we require that the roots of an assignable polynomial are in the negative complex half-plain. Hence, if any polynomial is assignable, the existence of stabilizing feedback can now be stated as a design consequence for both the Kalman and the Hautus case combined.

Corollary 4.24 (Existence of stabilizing state feedback).
There exists a feedback matrix F *such that* $A + BF$ *is Hurwitz if and only if* (A, B) *is stabilizable. If* (A, B) *is controllable, then the eigenvalues of* $A + BF$ *can be assigned arbitrarily by a suitable feedback* F*.*

Proof. Using separability, we obtain controllable and uncontrollable parts. Now we can apply the control canonical form to define the assignable polynomial. Then, the feedback assigns the controllable eigenvalues but leaves uncontrollable eigenvalues fixed. Therefore, a Hurwitz closed loop exists exactly when all fixed uncontrollable eigenvalues are already stable. If no uncontrollable part exists, arbitrary pole placement is possible. □

Practically speaking, not every internal state needs to be controllable, but every unstable internal state must be controllable. This distinction is essential in high-order models, where some stable parasitic dynamics may be left untouched.
Combining these lines of argumentation, Figure 4.4 provides an overview of the results.

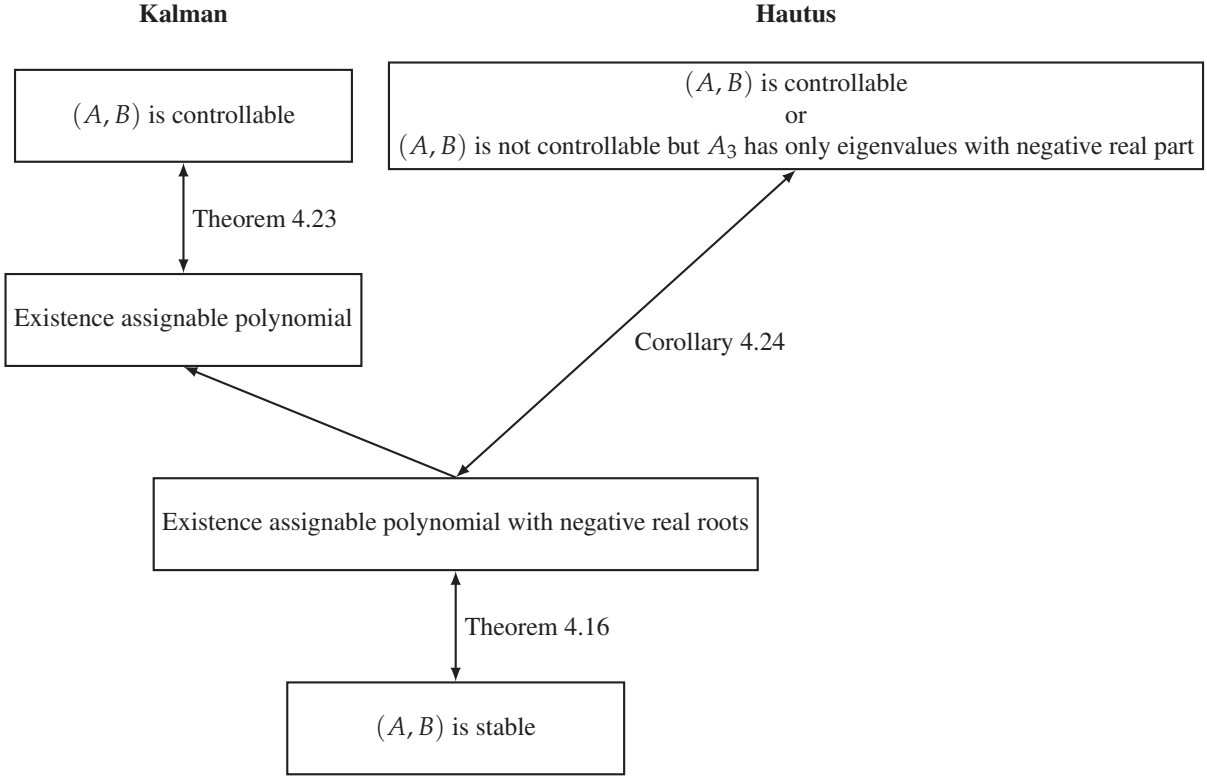


Figure 4.4: Connection of controllability and stability

Remark 4.25
Theorem 4.23 and Corollary 4.24 are often called pole shifting theorem as the roots of the characteristic polynomial are equivalent to the poles of the transfer matrix of the system.

Task 4.26 (Controllability chain in the DC motor)
Use the state-space structure from Task 4.9. Explain why the armature voltage can influence the position although it enters only the current equation.

Solution to Task 4.26: The voltage directly changes the current. The current changes the electromagnetic torque and therefore the angular velocity. The angular velocity integrates to the angular position. In matrix terms, the column B points into the current direction, AB contains the induced speed direction, and A^2B contains the induced position direction under typical nondegenerate parameters. Hence, the repeated products in the Kalman matrix express the physical actuation chain.

From this example, we observe that the controllability matrix is not an abstract construction but its columns represent how the input direction is propagated through the physical dynamics.

Table 4.1: Advantages and limitations of stability and controllability criteria

Advantage	Limitation
✓ Eigenvalues give a direct stability test.	✗ The test is exact only for linear autonomous systems.
✓ State feedback changes closed-loop modes.	✗ Uncontrollable modes cannot be moved.
✓ Kalman's criterion is easy to compute.	✗ Poor scaling can make rank tests numerically fragile.
✓ Hautus' criterion identifies uncontrollable modes.	✗ It still requires an accurate system matrix.
✓ Stabilizability allows stable hidden dynamics.	✗ Unstable hidden dynamics must be controllable.

4.3 Observability and Detectability

In the following, we want to change our focus. In practical applications, measurements are typically not available for all system states, yet feedback laws such as $\mathbf{u} = F\mathbf{x}$ require state information. To address this discrepancy, observability asks whether the missing state information

can be reconstructed from the input-output history. Similar to controllability and stability, also in this realm there exists a weaker property called *detectability*, which may be used if unobservable modes do exist but can be disregarded as the respective states are stable. More formally, we define the following:

Definition 4.27 (Distinguishability and observability).
Consider the continuous-time system (4.2). Two initial states $\mathbf{x}_1$ and $\mathbf{x}_2$ are called *distinguishable* if there exist an input $\mathbf{u}(\cdot)$ and a time $t \geq 0$ such that the corresponding outputs satisfy

$$\mathbf{y}(t; \mathbf{x}_1, \mathbf{u}) \neq \mathbf{y}(t; \mathbf{x}_2, \mathbf{u}). \tag{4.22}$$

The system is called *observable* if every pair of distinct initial states is distinguishable.

For linear systems, distinguishability does not depend on clever input choices but is a systemic property. The difference of two trajectories follows the homogeneous dynamics, and the output difference reveals whether this difference can be seen.

Lemma 4.28 (Distinguishability for LTI systems).
For the continuous-time LTI system (4.5)*, two initial states* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are distinguishable if and only if there exists* $t \geq 0$ *such that*

$$C \exp(At)(\mathbf{x}_1 - \mathbf{x}_2) \neq 0. \tag{4.23}$$

Proof. Define $z(t) = \mathbf{x}_1(t) - \mathbf{x}_2(t)$ and subtract the two state equations generated by the same input. The input terms cancel, so the difference evolves as $\dot{z}(t) = Az(t)$ with $z(0) = \mathbf{x}_1(0) - \mathbf{x}_2(0)$. The output difference is $Cz(t)$, which gives (4.23). □

Hence, observability is a property of the autonomous state dynamics together with the output map. In the DC motor, a speed sensor can reveal current effects through acceleration only if the model captures the current-speed coupling. A position sensor can reveal speed through the time history of position.
So again as in Section 4.2, we

1. need to identify conditions to ensure that a system is observable, and

2. have to construct an *observer*.

The following criterion is the observability counterpart of Kalman's controllability criterion. It checks whether the output and its successive dynamic derivatives contain enough independent information about the state.

Theorem 4.29 (Kalman observability criterion).
For a given LTI system (4.5)*, the pair* (A, C) *is observable if and only if*

$$\operatorname{rk} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n_x-1} \end{bmatrix} = n_x. \tag{4.24}$$

Proof. By Lemma 4.28, a nonzero state difference $z(t) = \mathbf{x}_1(t) - \mathbf{x}_2(t)$ is unobservable if $C\exp(At)z = 0$ for all $t \geq 0$. Differentiating at $t = 0$ gives $CA^j z = 0$ for all j. By the Cayley–Hamilton theorem, it is sufficient to check $j = 0, \ldots, n_x - 1$. Hence, observability is equivalent to the displayed matrix having trivial kernel. □

The interpretation mirrors controllability. The rows $C, CA, \ldots$ describe how the measured output and its dynamic evolution reveal state directions. If a state direction is invisible to all these rows, it cannot be reconstructed from output data.
In a production context, if only the final product temperature is measured, an internal heater state may still be observable if its effect appears in the temperature trajectory. If it leaves no trace in measured data, no estimator can recover it.
Similarly, if lateral position and yaw rate are measured in a driving funciton, lateral velocity may still be observable through the vehicle dynamics. If a motion state does not affect any measured signal, it cannot be estimated reliably.

From Theorems 4.18 and 4.29 we observe a certain similarity of the formulas for controllability and observability. The reason for this similarity lies in the so called *duality*. For every propagating system, there exists a so called *adjoint system* propagating in the exact opposite direction, e.g. forward/backward in time.

Definition 4.30 (Dual system).
Consider the system (4.5) defined by (A, B, C). Then we define the *dual system* as given by $(A^\top, C^\top, B^\top)$.

Generically speaking, while the system propagates states such as energy, mass or motion forward,

the adjoint system propagates duals such as sensitivities, constraints or information backwards through the dynamics. For further details, we refer to [6].
In short, observability of (A, C) is controllability of the transposed pair $(A^\top, C^\top)$.

Theorem 4.31 (Duality of controllability and observability).
For an LTI system (4.5) *the pair* (A, B) *is controllable if and only if the pair* $(A^\top, B^\top)$ *is observable. The pair* (A, C) *is observable if and only if the pair* $(A^\top, C^\top)$ *is controllable.*

Proof. Transpose the Kalman controllability matrix. The result is the Kalman observability matrix of the transposed pair. Since rank is invariant under transposition, the controllability and observability rank conditions are equivalent under duality. □

Duality is not merely a mathematical curiosity but a powerful tool for understanding and designing dynamical systems. It provides a convenient mnemonic: many statements and algorithms for controllability can be transformed into corresponding statements for observability simply by exchanging inputs and outputs and replacing matrices by their transposes.

Remark 4.32
From a graph-theoretic perspective, controllability asks whether information injected by the actuators can propagate to every state variable, whereas observability asks whether information originating from every state variable can eventually reach the sensors. The two problems are therefore related by reversing the direction of information flow.

The interpretation is useful for memory and design. Every controllability result has an observability counterpart. Every state-feedback pole-placement result has an observer-gain counterpart. We will use this connection in Chapter 6 when the Kalman filter is derived by a Riccati equation dual to the LQR equation.
Similar to the difference between the Kalman and the Hautus criterion in Theorems 4.18 and 4.21, we can use the idea of separability (Theorem 4.20) also in the context of observability. Similar to the reachability set from Definition 4.17, we can define the set of nonobservable states.

Definition 4.33 (Nonobservable set).
Given a continuous-time LTI system (4.5), we call

$$\mathcal{N}(t) := \{\mathbf{x}_0 \mid C\mathbf{x}(t, \mathbf{x}_0, 0) = 0 \ \forall t \geq 0\}$$

the set of *non-observable* states.

Now, we can directly derive the following:

Theorem 4.34.
Consider a continuous-time LTI system (4.5)*. Then the reachable set of the dual system is identical to the observable set*

$$\left(\bigcup_{t\geq 0}\mathcal{R}(t)\right)\top =: \mathcal{R}^{\top} = \mathcal{N}^{\perp} := \left(\bigcap_{t\geq 0}\mathcal{N}(t)\right)^{\perp}.$$

and vice versa.

Proof. Follows directly from duality. □

In particular, if a system is not observable, estimation is still possible if all invisible modes are stable. This is the estimator counterpart of stabilizability.

Definition 4.35 (Detectability).
Consider an LTI system (4.5). The pair (A, C) is called *detectable* if every unobservable mode of A is asymptotically stable. Equivalently, there exists a matrix L such that $A + LC$ is Hurwitz.

The Hautus criterion again gives the modal test. In observability form, it identifies eigenmodes that do not appear in the output.

Theorem 4.36 (Hautus observability and detectability criteria).
Given an LTI system (4.5)*, the pair* (A, C) *is observable if and only if*

$$\operatorname{rk}\begin{bmatrix}\lambda Id - A^{\top} & C^{\top}\end{bmatrix} = n_x \qquad \forall \lambda \in \mathbb{C}. \tag{4.25}$$

It is detectable if and only if (4.25) *holds for every eigenvalue* $\lambda \in \mathbb{C}$ *of* A *with* $\operatorname{Re}(\lambda) \geq 0$.

Proof. By applying duality, observability of (A, C) is controllability of $(A^{\top}, C^{\top})$. Hence detectability is stabilizability of the same dual pair. Applying Theorem 4.21, we direclty obtain the stated rank condition. □

The interpretation is again modal. Observable modes can be reconstructed from data. Unobservable stable modes may remain hidden because their effect decays. Unobservable unstable modes are dangerous because they can grow without being seen.

Task 4.37 (Sensor choice for the DC motor)
Use the DC-motor model from Task 4.9. Compare two output choices: measuring angular position φ only and measuring armature current i_a only. Explain which physical state information is more directly visible in each case and why the observability matrix must be checked for a rigorous answer.

Solution to Task 4.37: If φ is measured, then position is directly visible. Speed is visible through the time derivative of position, and current can become visible through the acceleration relation if the mechanical and electrical coupling parameters are nonzero.
If i_a is measured, then current is directly visible, and speed may be visible through the back-electromotive-force term in the electrical equation. Position, however, may be invisible if only current is measured and no position-dependent term enters the dynamics.
The observability matrix gives the rigorous answer because it checks whether the measured output and its dynamic propagation span all state directions.

From this example, we can conclude that sensor selection is a structural design decision. A sensor that is accurate for one variable is not automatically sufficient for reconstructing all states needed by a feedback controller. Observability depends on the interaction between the measured output and the internal dynamics of the system. Thus, sensor placement should not be judged only by measurement quality, but also by whether the selected outputs contain enough information to estimate the states that are relevant for control.

Combining these lines of argumentation together with the core of stability, Figure 4.5 provides an overview of the results.
We like to point out that the properties controllability and observability are independent from one another and only connected for the respective dual system. Consequently, there exist four classes of systems

1. controllable and observable,
2. controllable and not observable,
3. not controllable and observable, and
4. not controllable and not observable.

These classes can also be seen in Figure 4.3, which served as starting point for these terms.

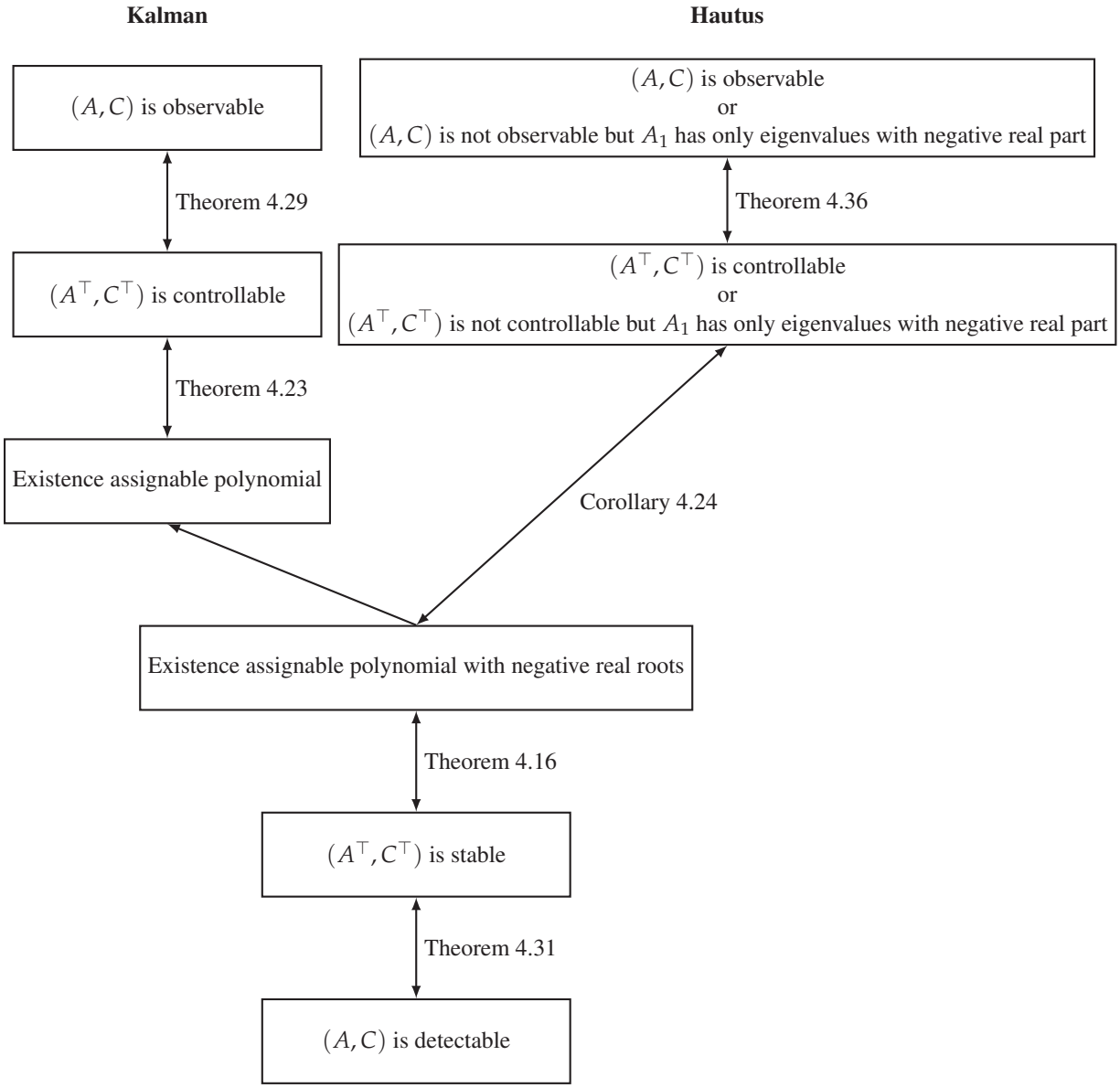


Figure 4.5: Connection of observability and detectability

Table 4.2: Advantages and limitations of observability and detectability criteria

Advantage	Limitation
✓ Observability checks whether states are reconstructable.	✗ It depends on the accuracy of the model and sensors.
✓ Kalman's criterion is a finite rank test.	✗ Rank tests can be ill-conditioned.
Continued on next page	

Table 4.2 – continued from previous page

Advantage	Limitation
✓ Duality allows to transfer controllability to observability results.	✗ Physical interpretation may be less direct after transposition.
✓ Detectability allows stable hidden modes.	✗ Unobservable unstable modes are not acceptable.
✓ Sensor placement can be evaluated structurally.	✗ Structural observability does not guarantee low-noise estimation.

The chapter has established the structural foundation for the optimal design methods that follow. Stability and stabilizability tell us whether feedback can make the state converge, either because the uncontrolled system is already stable or because the unstable modes can be influenced by the input. Observability and detectability tell us whether the state information required for feedback can be recovered from measurements, at least for those modes that are relevant for stability.

These properties do not yet choose an optimal feedback or observer. Instead, they answer a more fundamental question: is such a design meaningful in the first place? If an unstable mode cannot be controlled, no feedback law can stabilize it. If an unstable mode cannot be observed, no observer can reliably recover the information needed to compensate for it. Thus, stabilizability and detectability are the structural conditions that make optimal stabilization and optimal observation possible.

Chapter 5 will use these insights to compute optimal stabilizing feedback laws by balancing control effort against state performance. Chapter 6 will then use the dual viewpoint to compute state estimates from noisy measurements. Together, these two developments lead toward observer-based optimal control, where feedback is designed for the estimated state rather than for a directly measured full state vector.

CHAPTER 5

OPTIMAL STABILIZATION

In Chapter 4, we focused on the structural question whether feedback can stabilize a linear system. We introduced the properties controllability and stabilizability, which tell us whether the input can influence the states that matter for stability. If the relevant controllability assumptions are satisfied, then we saw that pole placement provides a constructive design method for the respective closed-loop eigenvalues. However, pole placement alone does not decide which stabilizing feedback should be preferred. In general, many stabilizing feedbacks are possible, and they may differ substantially in their transient response, required actuator effort, robustness properties, and sensitivity to model uncertainties.

In this chapter, we move from the structural question of stabilizability to the design question of optimal stabilization. The previous chapter told us when feedback can be used to stabilize a system. Here, we ask which stabilizing feedback should be chosen. Instead of only assigning closed-loop eigenvalues, we introduce a cost functional that penalizes state deviations and control effort. The feedback is then determined by minimizing this cost. This is important in applications because stabilization alone does not express the relevant engineering trade-offs. For example, a DC motor can often be brought quickly to a desired angular velocity or position by using a large armature voltage. Such a controller may give a fast response, but it may also require high input energy, violate actuator limitations, increase thermal load, or make measurement noise more visible in the closed-loop behavior. A less aggressive controller may be more economical and less sensitive, but the state errors will usually decay more slowly.

To develop this idea, Section 5.1 introduces the linear-quadratic regulator as the basic optimal stabilization problem. We formulate the infinite-horizon cost functional, interpret the value function, and derive the Riccati equation as the central design equation. Under suitable stabilizability and detectability assumptions, the algebraic Riccati equation yields a linear state feedback that is both optimal for the quadratic cost and stabilizing for the closed-loop system. The weighting

matrices therefore become design parameters, which specify how much importance we assign to state regulation compared with input effort.

Section 5.2 then changes the point of view. Instead of starting from an initial state and measuring the cost of regulating it to the origin, we consider the effect of disturbances on a performance output. The H_2 norm measures the output energy generated by impulse-like disturbances. In particular, we thereby connect optimal stabilization with input-output performance. The latter can be interpreted such that the controller reduces the average effect of disturbances on the relevant performance variables.

Last, Section 5.3 introduces the H_∞ viewpoint. In this case, our goal is not to minimize an average energy measure. Instead, we design a controller to bound the worst-case amplification from disturbances to performance outputs. As a result, the controller provides a robust performance certificate for the selected disturbance and performance channels. The H_∞ formulation is therefore particularly useful when uncertain models, external disturbances, or conservative performance guarantees must be taken into account.

The DC-motor example is used throughout the chapter to relate these abstract criteria to a concrete control task. In the LQR setting, the weighting matrices determine how strongly angular position and angular velocity errors are penalized relative to the armature voltage. In the H_2 setting, load-torque variations are interpreted as typical disturbance effects, and the controller is assessed by the resulting output energy. In the H_∞ setting, the focus is on unfavorable disturbances and on guaranteed attenuation from load torque to the chosen performance output. The example therefore shows that the same physical system can lead to different controller designs depending on whether the main concern is regulation effort, average disturbance performance, or worst-case robustness.

Similar to the previous chapters, we will utilize the following standing assumptions unless stated otherwise.

Assumption 5.1

All finite-dimensional state spaces are real vector spaces $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ with the Euclidean norm $\|\cdot\|_2$. Continuous-time inputs $\mathbf{u}(t)$ are assumed to be piecewise continuous on compact intervals with time indicated by t, and discrete-time inputs $\mathbf{u}(k)$ are arbitrary sequences of compatible dimension with time index k. For stability statements, equilibria are shifted to the origin whenever this simplifies notation. The standing system model is the continuous-time LTI system

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0 \tag{5.1a}$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) \tag{5.1b}$$

with stabilizable pair (A, B) and the full state is assumed to be available for feedback.

5.1 Linear Quadratic Regulator — LQR

The starting point for optimal feedback design is the question of how good performance should be measured. Stabilization alone only tells us that the state converges to the desired equilibrium. It does not tell us whether the convergence is fast enough, whether the required input is acceptable, or whether disturbances are attenuated in a satisfactory way. Therefore, an optimal control problem requires a quantification via a so called *key performance criterion*. This criterion translates engineering requirements into a mathematical object that can be minimized or bounded. More formally:

Definition 5.2 (Key performance criterion).
A *key performance criterion* is a function, which measures defined information retrieved from the system against a standard.

Focusing on the state space, we typically speak of cost functions. These combined information on state and input of the system to quantify performance of the control.

Definition 5.3 (Cost function).
We call a key performance criterion given by a function $\ell : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_0^+$ a *cost function*.

There are different ways to formulate such a criterion. In the LQR setting, the state-space representation is used directly. The performance measure is a quadratic cost functional that penalizes deviations of the state from the origin and, at the same time, penalizes the size of the control input. The weighting matrices specify the relative importance of these two effects. For example, in a DC-motor application, one may penalize angular position and angular velocity errors, but also the armature voltage needed to correct them. In this way, the cost functional makes the design trade-off explicit.
For H_2 and H_∞ control, the interpretation is different. Here, the focus is not primarily on the cost generated by a given initial state, but on the input-output behavior of the closed-loop system. Disturbances, reference signals, measurement noise, or model uncertainties are treated as external inputs, while the variables of interest are collected in a performance output. The transfer function from these external inputs to the performance output then describes how strongly the closed loop transmits unwanted effects. The H_2 norm measures this behavior in an energy or average-performance sense, whereas the H_∞ norm gives a worst-case amplification bound.
As we can already see from the H_∞ case, we should be able to consider entire trajectories. The value of a cost function according to Definition 5.3, however, only shows a snapshot, i.e. the evaluation at a single point in time $t \in \mathcal{T}$. To get the performance, we need to evaluate it over the lifetime of the system. Since we are defining a function of a function, this is called a functional.

Definition 5.4 (Running cost and cost functional).
A function $\ell : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$ is called a *running cost*. For an initial state $\mathbf{x}_0$ and an input function $\mathbf{u}(\cdot)$, the infinite-horizon cost functional is

$$J(\mathbf{x}_0, \mathbf{u}) := \int_0^\infty \ell\big(\mathbf{x}(t; 0, \mathbf{x}_0, \mathbf{u}), \mathbf{u}(t)\big)\, dt. \tag{5.2}$$

Remark 5.5
Note that the cost functional is defined such that it may cover the cases of LQR, H_2 and $H\infty$ and is additionally not limited to the linear case, i.e. may also be used in a nonlinear setting.

We can now combine the cost functional with an evaluation of the dynamics of our system. This allows to to not only judge whether a given operating point is desirable, we also measure how the system moves from its current state toward this operating point. In control applications, this distinction is essential. A controller should not only make the equilibrium attractive, but instead should also shape the transient in a useful way. Large overshoots, slow convergence, oscillatory behavior, or excessive input amplitudes may all be unacceptable, even if the closed-loop system is asymptotically stable.
Mathematically, this is expressed by assigning a cost to each admissible input trajectory. For a given initial condition $\mathbf{x}_0$, the input $\mathbf{u}(\cdot)$ generates a state trajectory $\mathbf{x}(\cdot)$ through the system dynamics. The cost functional then evaluates this pair $(\mathbf{x}, \mathbf{u})$ over the infinite time interval. As such, the cost functional typically penalizes state deviations and input effort. From an engineering perspective, this means that settling time, damping behavior, actuator usage, and energy consumption are reflected indirectly through the chosen weights.

Definition 5.6 (Optimal control problem and value function).
For the system (5.1) and the cost functional (5.2), the infinite-horizon optimal control problem is

$$\inf_{\mathbf{u}} J(\mathbf{x}_0, \mathbf{u}) \quad \text{subject to} \quad \dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0. \tag{5.3}$$

The optimal value function is

$$V(\mathbf{x}_0) := \inf_{\mathbf{u}} J(\mathbf{x}_0, \mathbf{u}). \tag{5.4}$$

The value function tells us how much cost is still unavoidable when the system starts in a given state. It therefore measures how difficult this state is to regulate under the best possible input. In

a stabilization problem, this quantity should decrease along the optimally controlled trajectory, because the system moves closer to the origin and the remaining cost becomes smaller. This decrease is exactly the property we also use in Lyapunov theory to prove stability. The value function therefore connects the design goal of minimizing a cost with the stability goal of making the state converge.

Task 5.7 (Running example: optimal speed stabilization of a DC motor)
Consider the armature-controlled DC motor from Chapter 4 with state $\mathbf{x} = \begin{bmatrix} i_\mathrm{a} & \omega & \varphi \end{bmatrix}^\top$ *and input* $\mathbf{u} = u_\mathrm{a}$. *Suppose that the control objective is to bring speed and current deviations back to zero after a disturbance while avoiding unnecessarily large armature voltages. Propose a quadratic running cost and explain qualitatively what happens if the input weight is increased.*

Solution to Task 5.7: A natural choice is

$$\ell(\mathbf{x}, \mathbf{u}) = q_i i_\mathrm{a}^2 + q_\omega \omega^2 + q_\varphi \varphi^2 + r u_\mathrm{a}^2$$

with $q_i, q_\omega, q_\varphi \geq 0$ and $r > 0$. If the main objective is speed stabilization rather than position regulation, then q_ω should be large compared with q_φ. The coefficient r penalizes armature voltage. Increasing r usually leads to smaller voltage commands and a less aggressive feedback. The transient becomes slower, but the controller is less demanding for the actuator. Decreasing r has the opposite effect and may create fast responses at the price of large input amplitudes.

In a production setting, the same LQR idea appears when process deviations and actuator effort must be balanced. The state may contain web-tension errors, temperature deviations, conveyor-speed deviations, register-position errors, or internal actuator states. The input may represent motor torque, heating power, valve motion, or a conveyor command. Large state weights express strict product-quality requirements, while large input weights express energy limitations, actuator wear, or smooth-operation requirements.
Similarly, in an AD/ADAS vehicle function, the state may contain lateral deviation, yaw-rate error, velocity error, acceleration error, or actuator states. The input may represent steering torque, braking torque, or propulsion torque. An LQR design then balances tracking accuracy, comfort, and actuator effort. Increasing the weight on a motion error improves regulation of the vehicle motion, while increasing the input weight avoids aggressive steering, braking, or propulsion commands.

This task illustrates the central idea of optimal stabilization. We do not choose eigenvalues but instead encode the engineering trade-off in the cost. In particular, we are going to apply the following idea:

Definition 5.8 (Null controlling).
An optimal control problem is called *null controlling* if

$$J(\mathbf{x}_0, \mathbf{u}) < \infty \qquad \Longrightarrow \qquad \mathbf{x}(t; 0, \mathbf{x}_0, \mathbf{u}) \to 0 \quad \text{as } t \to \infty. \tag{5.5}$$

This definition formalizes the idea that a finite cost should only be possible if the controller actually stabilizes the system. If the state did not converge to the origin, then a cost functional that penalizes state deviations would keep accumulating cost over time. Thus, finite cost and convergence to the origin are linked. From an engineering point of view, this means that the performance criterion is not allowed to hide an unstable or non-decaying behavior. Null controlling therefore ensures that minimizing the cost is also meaningful as a stabilization objective.
To utilize this relationship, we need to design the key performance criterion to be zero at the desired operating point. Hence, once the operating point is reached, no additional costs will be incurred over the operating period. Therefore, the state of the system will remain at the operating point. Note that by Definition 4.12 for each operating point there exists an input such that the state remains unchanged.
For stabilization problems, this definition excludes inputs with finite cost but nonconvergent state trajectories. In the quadratic setting below this property follows from positive definiteness of the cost in state and input.

Definition 5.9 (Quadratic running cost).
A *quadratic running cost* is given by

$$\ell(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}. \tag{5.6}$$

where $Q \in \mathbb{R}^{n_x \times n_x}$, $S \in \mathbb{R}^{n_x \times n_u}$ and $R \in \mathbb{R}^{n_u \times n_u}$ form a symmetric and positive definite matrix in (5.6).

Remark 5.10
In many engineering designs the mixed term is set to zero. Then

$$\ell(\mathbf{x}, \mathbf{u}) = \mathbf{x}^\top Q \mathbf{x} + \mathbf{u}^\top R \mathbf{u},$$

with $Q \succeq 0$ and $R \succ 0$. This is the form most commonly used in numerical software. The more general form in Definition 5.9 is useful for the proof because it shows where mixed state-input penalties would enter.

We now focus on the important subcase of a so called *linear-quadratic problem*. Here, the dynamics are linear, and the performance criterion is chosen to be quadratic in the state and the input. This choice is mathematically convenient as we will see in the upcoming stability analysis, but it also has a clear engineering interpretation. The state-dependent terms penalize deviations from the desired equilibrium, the input-dependent terms penalize the required actuator effort, and the mixed term allows one to include cross-couplings between state and input.

Definition 5.11 (Linear-quadratic problem).
The infinite-horizon *linear-quadratic problem* consists of the LTI system (5.1) and the quadratic cost functional

$$J(\mathbf{x}_0, \mathbf{u}) = \int_0^\infty \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} dt. \tag{5.7}$$

Now, we want to link back to stability of the closed loop by utilizing the quadratic performance criterion and the null-controlling property.

Theorem 5.12 (Null-controlling property of the LQ problem).
Consider the LQ problem and assume that the quadratic running cost $\ell(\mathbf{x}, \mathbf{u})$ is symmetric positive definite. Then the LQ problem is null controlling.

Proof. Positive definiteness of the quadratic running cost gives a lower bound by the squared state norm and, because the vector field $A\mathbf{x} + B\mathbf{u}$ is linear, also by the squared velocity up to constants. If a finite-cost trajectory did not converge to zero, it would visit a fixed distance from the origin infinitely often. On each such visit, either it remains away from the origin for a short time or it moves a fixed distance in state space. Both alternatives imply a fixed positive amount of accumulated cost. Infinitely many such visits would make the cost infinite. Hence finite cost implies convergence to the origin. □

The theorem explains why the infinite-horizon LQ cost is compatible with stabilization. As state deviations are penalized over the whole time axis, a trajectory can have finite total cost only if these deviations disappear asymptotically. The cost functional therefore does not merely measure performance, it also enforces the qualitative behavior required for stabilization.

The central question now is to compute the solution of the LQ problem. In particular, we are not simply interested in a solution but in a solution which can be evaluated based on the state of the system, i.e. a feedback. To this end, we utilize the idea of the value function and suppose it can be chosen in the ansatz

$$V(\mathbf{x}) = \mathbf{x}^\top P \mathbf{x} \tag{5.8}$$

for $P \in \mathbb{R}^{n_x \times n_x}$. This quadratic ansatz is special to the linear-quadratic structure. It turns the Hamilton–Jacobi–Bellman equation into an algebraic matrix equation.

Theorem 5.13 (LQR feedback from the algebraic Riccati equation).
Consider the LQ problem (5.7) *with* $\ell(\mathbf{x}, \mathbf{u})$ *being symmetric positive definite. Suppose there exists a symmetric positive definite matrix* $P = P^\top \succ 0$ *satisfying the algebraic Riccati equation*

$$PA + A^\top P + Q - (PB + S)R^{-1}(B^\top P + S^\top) = 0. \tag{5.9}$$

Then the optimal value function is

$$V(\mathbf{x}) = \mathbf{x}^\top P \mathbf{x}, \tag{5.10}$$

and the optimal input is the linear state feedback

$$\mathbf{u}^\star(t) = F\mathbf{x}(t), \qquad F = -R^{-1}(B^\top P + S^\top). \tag{5.11}$$

The closed-loop origin of $\dot{\mathbf{x}} = (A + BF)\mathbf{x}$ *is exponentially stable.*

Proof. Set $W(\mathbf{x}) = \mathbf{x}^\top P \mathbf{x}$. The Hamilton–Jacobi–Bellman expression

$$DW(\mathbf{x})(A\mathbf{x} + B\mathbf{u}) + \ell(\mathbf{x}, \mathbf{u})$$

is a strictly convex quadratic function of $\mathbf{u}$. Its unique minimizer is

$$\mathbf{u} = -R^{-1}(B^\top P + S^\top)\mathbf{x}.$$

Substituting this minimizer into the Hamilton–Jacobi–Bellman expression leaves exactly the Riccati expression in (5.9). Hence, the minimum is zero. The sufficient optimality condition for null-controlling problems then gives $V(\mathbf{x}) = W(\mathbf{x})$ and optimality of the feedback. Along the optimal closed loop, $\dot{W} = -\ell(\mathbf{x}, F\mathbf{x})$, and positive definiteness of the running cost makes W a strict Lyapunov function. Since W is quadratic, the origin is exponentially stable. □

For the common case $S = 0$, the Riccati equation and feedback simplify to

$$A^\top P + PA - PBR^{-1}B^\top P + Q = 0, \tag{5.12}$$

$$F = -R^{-1}B^\top P. \tag{5.13}$$

This is the form that is most often meant by the abbreviation LQR.

Theorem 5.14 (Existence of the stabilizing Riccati solution).
For the LQ problem with $\ell(\mathbf{x}, \mathbf{u}) \succ 0$, the following statements are equivalent:

1. *(A, B) is stabilizable.*
2. *The algebraic Riccati equation (5.9) has a unique symmetric positive definite stabilizing solution P.*
3. *The optimal value function is quadratic, $V(\mathbf{x}) = \mathbf{x}^\top P\mathbf{x}$.*
4. *There exists an optimal linear feedback law that stabilizes the system.*

Proof. A Riccati solution gives the value function and the optimal stabilizing feedback by Theorem 5.13. Conversely, if an optimal stabilizing linear feedback exists, then the pair (A, B) must be stabilizable by the structural results of Chapter 4. The nontrivial direction is that stabilizability implies the existence of a stabilizing Riccati solution. This is shown by solving the Riccati differential equation on finite horizons, comparing its finite-horizon values with the finite cost of an arbitrary stabilizing feedback, and passing to the infinite-horizon limit. The limit satisfies the algebraic Riccati equation. Uniqueness follows because two positive definite stabilizing solutions would generate the same optimal value function and hence the same quadratic form. □

When computing of a solution P of (5.9), we have to be careful about the requirements of the solution for the following reason: While the algebraic Riccati equation can have more than one solution, there exists at most one semi positive definite P.

Remark 5.15
Note that Algorithm 7 does not state on how to design the matrices Q, R and S. These need to be prefixed by the designer.

The connections between the latter results are visualized in Figure 5.1.

Algorithm 7 Computation of an LQR feedback

Input: LTI system (5.1), weights Q, R, and optionally S

1: **procedure** GENERATE LQR(A, B, Q, R, S)
2: Check that (A, B) is stabilizable.
3: Solve algebraic Riccati equation (5.9) for its stabilizing symmetric positive definite solution P.
4: Compute F from (5.11) and apply $\mathbf{u} = F\mathbf{x}$.
5: Verify the closed-loop eigenvalues of $A + BF$ and simulate representative transients.
6: **end procedure**

Output: LQR feedback $\mathbf{u} = F\mathbf{x}$

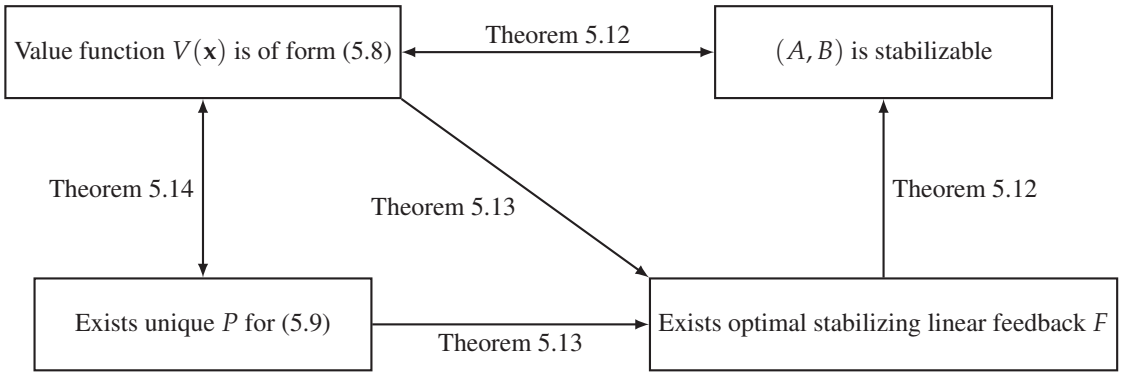


Figure 5.1: Connection of LQR results

Remark 5.16

The state based setting described within this section can be extended to the output based setting. For this case, we utilize the quadratic cost function

$$\ell(\mathbf{y}, \mathbf{u}) = \begin{bmatrix} \mathbf{y}^\top & \mathbf{u}^\top \end{bmatrix} \cdot \begin{pmatrix} \widetilde{Q} & \widetilde{S} \\ \widetilde{S}^\top & R \end{pmatrix} \cdot \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \end{bmatrix} \tag{5.14}$$

with $Q = C^\top \widetilde{Q} C$ and $S = C^\top S$. Given output values $\mathbf{y}$, we obtain that the respective LQ problem is null controlling if the pair (A, C) is observable.

Now, we apply this approach to the most simple case of a scalar LTI system:

Task 5.17 (Scalar LQR calculation)

Consider the scalar system $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ with $B \neq 0$ and cost

$$J(\mathbf{x}_0, \mathbf{u}) = \int_0^\infty (Q\mathbf{x}(t)^2 + R\mathbf{u}(t)^2)\, dt,$$

where $Q > 0$ and $R > 0$. Compute the stabilizing Riccati solution and the LQR feedback.

Solution to Task 5.17: The Riccati equation is

$$2AP - \frac{B^2}{R}P^2 + Q = 0.$$

Solving the quadratic equation gives

$$P = \frac{R}{B^2}\left(A \pm \sqrt{A^2 + \frac{B^2Q}{R}}\right).$$

The stabilizing positive solution is

$$P = \frac{R}{B^2}\left(A + \sqrt{A^2 + \frac{B^2Q}{R}}\right),$$

and the feedback is

$$\mathbf{u} = F\mathbf{x}, \qquad F = -\frac{B}{R}P = -\frac{1}{B}\left(A + \sqrt{A^2 + \frac{B^2Q}{R}}\right).$$

The closed-loop coefficient is $A + BF = -\sqrt{A^2 + B^2Q/R} < 0$, hence the feedback is stabilizing.

The scalar example makes the tuning effect transparent. Increasing Q increases the magnitude of the stabilizing feedback and moves the closed-loop pole further left. Increasing R reduces the magnitude of the feedback and moves the closed-loop pole closer to the origin.

Task 5.18 (Interpreting LQR weights for the DC motor)
For the DC motor model, compare the two diagonal weight choices

$$Q_1 = \mathrm{diag}(1, 100, 1), \qquad R_1 = 1,$$
$$Q_2 = \mathrm{diag}(1, 100, 1), \qquad R_2 = 100.$$

What qualitative difference do you expect between the corresponding LQR controllers?

Solution to Task 5.18: Both choices penalize speed error strongly because the second diagonal entry of Q is large. The second design, however, penalizes armature voltage much more strongly. Therefore, the controller for (Q_2, R_2) will be less aggressive. It uses smaller voltage commands, accepts slower speed-error decay, and produces a less demanding transient. The controller for (Q_1, R_1) is expected to react faster but with larger voltage peaks.

The same qualitative tuning logic applies to the production and vehicle examples. In production automation, increasing the state weight on a temperature deviation, tension deviation, or position error makes the controller react more strongly to this quality-relevant variable. Increasing the input weight on heating power, motor torque, or valve motion makes the controller less aggressive and may reduce energy consumption, actuator wear, and saturation risk.
In an AD/ADAS vehicle function, increasing the state weight on lateral deviation, yaw-rate error, or velocity error makes the controller prioritize tracking and motion regulation. Increasing the input weight reduces steering, braking, or propulsion activity and therefore supports comfort, actuator protection, and smoother interaction with lower-level control loops.

The LQR design gives a clean and powerful answer to the optimal stabilization problem. Once the weights have been chosen, the Riccati equation provides a stabilizing feedback and the corresponding value function. At the same time, the discussion also shows where the method has its limitations. The main strengths and restrictions are summarized in Table 5.1.

Table 5.1: Advantages and limitations of LQR design

	Advantage		**Limitation**
✓	Gives an optimal stabilizing state feedback under structural assumptions.	✗	Requires a state-space model and full state information or an observer.
✓	Weight matrices encode the trade-off between performance and input effort.	✗	The result is only as meaningful as the chosen weights and model.
✓	The Riccati equation can be solved reliably with standard numerical tools.	✗	Hard input and state constraints are not handled directly.
✓	The value function provides a Lyapunov function for the closed loop.	✗	Robustness is indirect; uncertainty is not optimized explicitly.

Thus, LQR should be seen as the basic Riccati-based design method for optimal state feedback. It gives a precise way to balance state regulation and actuator effort. The following sections

keep this optimal-control viewpoint, but shift the interpretation from initial-state regulation to input-output performance and robustness.

5.2 H_2 Control

The LQR design in the previous section measured performance from the state-space point of view. For a given initial condition, the cost functional penalized the resulting state and input trajectories over time. The H_2 approach keeps the idea of energy-based performance, but changes the perspective. Instead of asking how expensive the transient is for one initial state, we ask how strongly a system maps external inputs, such as disturbances or noise, to performance outputs. This leads to an input-output interpretation of optimal stabilization.
To make this precise, we first need a norm that measures the size of a signal over time. For disturbance attenuation, the relevant quantity is not only the instantaneous amplitude of the output, but the total energy contained in the output signal. The L_2-norm provides the means to measure the latter. It assigns a finite number to signals whose squared magnitude is integrable over time and therefore gives the basic notion of signal energy used in the so called H_2 framework.

Remark 5.19
In the literature, the term L_2 space is typically found to be the correct one. Yet, talking about function which are bounded and analytic in the right half plane and exhibit finite L_p norms on the imaginary axis – which are fundamental for stable function – are called Hardy spaces, the term H_2 norm has become dominant.

Definition 5.20 (L_2 / H_2 norm).
For a signal $v : [0, \infty) \to \mathbb{R}^p$, the L_2 or H_2 norm is

$$\|v(t)\|_2 := \left(\int_0^\infty v(t)^\top v(t)\, dt \right)^{1/2}, \tag{5.15}$$

provided that the integral is finite.

The square of the H_2 norm is the accumulated signal energy. For a scalar signal this is simply the area under $v(t)^2$. For a vector-valued signal, the Euclidean norm of the vector is integrated over time. In engineering terms, this gives a natural way to measure how much total output activity a system produces after it has been excited.

Task 5.21 (Running example: disturbance energy in the DC motor)
Consider the armature-controlled DC motor from Chapter 4 with

$$\mathbf{x} = \begin{bmatrix} i_\mathrm{a} & \omega & \varphi \end{bmatrix}^\top, \qquad \mathbf{u} = u_\mathrm{a}. \tag{5.16}$$

Assume that load-torque variations are modeled by an external disturbance input $\mathbf{d}$. *Explain why an* H_2 *formulation is useful if the main performance output is the speed deviation.*

Solution to Task 5.21: In the LQR formulation, the controller is mainly evaluated by the cost generated from an initial state deviation. In the H_2 formulation, the question is different. We ask how much speed-error energy is generated when the motor is excited by load-torque disturbances. If the performance output is chosen as $\mathbf{y} = \omega$ or as a weighted combination of speed error and armature voltage, then the H_2 norm measures how strongly load-torque disturbances are transmitted to this performance output. From an engineering point of view, a small H_2 value means that typical short load variations produce only small accumulated speed deviations.

To turn this idea into a feedback design problem, we explicitly include a disturbance input $\mathbf{d}$ in the plant. This signal represents unwanted excitations such as load variations, process disturbances, or noise entering the dynamics. The performance output $\mathbf{y}$ collects the quantities that should remain small, for example state deviations, velocity errors, or weighted actuator usage. The H_2 state-feedback problem therefore asks for a stabilizing feedback that makes the closed-loop response from disturbance $\mathbf{d}$ to output $\mathbf{y}$ as small as possible in an energy sense.

Definition 5.22 (H_2 state-feedback problem).
Consider the LTI system

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) + E\mathbf{d}(t), \tag{5.17a}$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t). \tag{5.17b}$$

The H_2 state-feedback problem is to choose a stabilizing feedback $\mathbf{u}(t) = F\mathbf{x}(t)$ such that the H_2 norm of the closed-loop transfer from $\mathbf{d}$ to $\mathbf{y}$ is minimized.

Task 5.23 (Choosing the performance output for the DC motor)
For the DC motor, let the disturbance input $\mathbf{d}(t)$ *represent load-torque variations. Suppose*

that the controller should reduce speed deviations but should also avoid unnecessarily large armature voltages. Propose a performance output of the form $\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$.

Solution to Task 5.23: A suitable choice is to collect the weighted speed deviation and the weighted input in the performance output, for example

$$\mathbf{y} = \begin{bmatrix} \sqrt{q_\omega}\,\omega \\ \sqrt{r}\,u_\mathrm{a} \end{bmatrix}, \qquad q_\omega > 0, \quad r > 0.$$

With

$$\mathbf{x} = \begin{bmatrix} i_\mathrm{a} & \omega & \varphi \end{bmatrix}^\top, \qquad \mathbf{u} = u_\mathrm{a},$$

this corresponds to

$$C = \begin{bmatrix} 0 & \sqrt{q_\omega} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \\ \sqrt{r} \end{bmatrix}.$$

Then $\mathbf{y}^\top\mathbf{y} = q_\omega\omega^2 + ru_\mathrm{a}^2$. The parameter q_ω determines how strongly speed deviations caused by load-torque disturbances are penalized. The parameter r determines how expensive armature-voltage usage is. Thus, the performance output directly encodes the engineering compromise between disturbance rejection and actuator effort.

To proceed into this direction, we first emphasize that the same signal energy can be computed in the frequency domain. This is useful because transfer matrices are naturally evaluated on the imaginary axis. The precise relation is Parseval's identity.

Corollary 5.24 (H_2 norm equivalence in time and frequency domain).
Consider a function $v : [0,\infty) \to \mathbb{R}^{n_y}$ *and its Laplace transform* $V(s) := \hat{v}(t)$. *Then*

$$\|v(t)\|_2 = \|V(s)\|_2 \tag{5.18}$$

holds for the H_2 *norms.*

Proof. By definition of the time-domain L_2-norm,

$$\|v(t)\|_2^2 = \int_{-\infty}^{\infty} v(t)^\top v(t)\,dt = \sum_{j=1}^{n_y} \int_{-\infty}^{\infty} |v_j(t)|^2\,dt.$$

Parseval's theorem applied componentwise gives

$$\int_{-\infty}^{\infty} |v_j(t)|^2 \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |V_j(i\omega)|^2 \, d\omega, \qquad j = 1, \ldots, n_y.$$

Hence

$$\|v(t)\|_2^2 = \frac{1}{2\pi} \sum_{j=1}^{n_y} \int_{-\infty}^{\infty} |V_j(i\omega)|^2 \, d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} V(i\omega)^* V(i\omega) \, d\omega.$$

The last expression is the squared L_2-norm of the frequency-domain representation, with the standard normalization used for Parseval's identity. Therefore, we obtain $\|v(t)\|_2^2 = \|V(s)\|_2^2$ and by taking the square root the assertion follows. □

In order to apply this result, we reconsider our dynamics. For multivariable systems, we know that a reformulation via the Laplace transform reveals a transfer matrix connecting inputs to outputs. In particular, for the LTI case

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t) \end{aligned}$$

the frequency domain equivalent is given by

$$G(s) = C \left(s\mathrm{Id} - A\right)^{-1} B + D.$$

For zero initial condition, the output is obtained by convolution with the impulse response. If the direct term is absent, i.e., $D = 0$, this impulse response is the ordinary matrix-valued function. Computing the solution of the LTI system reveals

$$\mathbf{y}(t) = C \exp\left(At\right) \mathbf{x}_0 + \int_0^t H(t-\tau)\mathbf{u}(\tau) d\tau \tag{5.19}$$

where $H(t-\tau)$ is the impulse response

$$H(t) := \begin{cases} C \exp\left(At\right) B + D, & \text{if } t \geq 0 \\ 0, & \text{if } t < 0. \end{cases}$$

Combined, we obtain the Laplace transform of the impulse response, which describes how each input channel affects each performance output over time.

Corollary 5.25 (Laplace-transform impulse response).
Consider an LTI system (5.1)*. Then*

$$G(s) = \int_0^\infty H(t) \exp(-st)\, dt \tag{5.20}$$

represents the transfer matrix of the system.

Proof. Follows by derivation. □

We can now define the H_2 norm of a stable transfer matrix. Note that the direct term D must be zero in this definition. Otherwise, an ideal impulse input produces an instantaneous impulse at the output, and the corresponding output energy is not finite.

Definition 5.26 (H_2 norm of a stable transfer matrix).
Let $G : \mathbb{C} \to \mathbb{C}$ be a stable, strictly proper transfer matrix with impulse response g. Its H_2 norm is

$$\|G(s)\|_2 := \left(\int_0^\infty \operatorname{tr}\big(g(t)^\top g(t)\big)\, dt \right)^{1/2}. \tag{5.21}$$

Equivalently, if $G : \mathbb{C} \to \mathbb{C}$ is stable,

$$\|G(s)\|_2^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \operatorname{tr}\big(G(i\omega)^* G(i\omega)\big)\, d\omega. \tag{5.22}$$

Hence, for a single-input single-output stable system, the squared H_2 norm is the output energy generated by an impulse input. For multiple inputs, the squared H_2 norm is the sum of these impulse-response energies over all input channels. Thus the H_2 norm measures an average energy effect of disturbances on the performance output.

Theorem 5.27 (H_2 norm equivalence for LTI systems).
Consider the stable, strictly proper LTI system (5.1) *with* $D = 0$ *and let* $G(s) = C(sId - A)^{-1}B$.
Then we have

$$\|G(s)\|_2 = \|g(t)\|_2, \tag{5.23}$$

where

$$\|g(t)\|_2^2 = \int_0^\infty \operatorname{tr}\left(g(t)^\top g(t)\right)\, dt. \tag{5.24}$$

Proof. By Corollary 5.25, the transfer matrix is the Laplace transform of the impulse response. Evaluating this transform on the imaginary axis gives the frequency response $G(i\omega)$. Applying Corollary 5.24 componentwise to the entries of g yields

$$\int_0^\infty \operatorname{tr}\left(g(t)^\top g(t)\right)\, dt = \frac{1}{2\pi}\int_{-\infty}^\infty \operatorname{tr}\left(G(i\omega)^* G(i\omega)\right)\, d\omega.$$

The left-hand side is $\|g(t)\|_2^2$, and the right-hand side is $\|G(s)\|_2^2$. Taking square roots proves the assertion. □

For computations, it is often convenient to avoid the frequency integral. If A is Hurwitz, then the impulse response decays exponentially, and the following constant matrices are well-defined:

$$X := \int_0^\infty \exp(At)BB^\top \exp(A^\top t)\, dt, \qquad Y := \int_0^\infty \exp(A^\top t)C^\top C \exp(At)\, dt. \tag{5.25}$$

They are the unique symmetric positive semidefinite solutions of

$$AX + XA^\top + BB^\top = 0, \qquad A^\top Y + YA + C^\top C = 0. \tag{5.26}$$

Theorem 5.28 (H_2 norm from algebraic matrix equations).
Consider the stable, strictly proper LTI system (5.1) *where A is Hurwitz and $D = 0$. Let $X = X^\top \succeq 0$ and $Y = Y^\top \succeq 0$ solve*

$$AX + XA^\top + BB^\top = 0, \qquad A^\top Y + YA + C^\top C = 0. \tag{5.27}$$

Then we have

$$\|G(s)\|_2^2 = \operatorname{tr}\left(CXC^\top\right) = \operatorname{tr}\left(B^\top YB\right), \tag{5.28}$$

where $G(s) = C(sId - A)^{-1}B$.

Proof. Since A is Hurwitz, the impulse response is

$$g(t) = C\exp(At)B, \qquad t \geq 0,$$

and the integrals in (5.25) are finite. Differentiating $\exp(At)BB^\top \exp(A^\top t)$ with respect to t and integrating over $[0,\infty)$ gives $AX + XA^\top + BB^\top = 0$. Thus X solves (5.27). Using Definition 5.26 and cyclic invariance of the trace, we obtain

$$\begin{aligned}\|G(s)\|_2^2 &= \int_0^\infty \operatorname{tr}\left(B^\top \exp(A^\top t)C^\top C \exp(At)B\right)\, dt \\ &= \operatorname{tr}\left(C\left(\int_0^\infty \exp(At)BB^\top \exp(A^\top t)\, dt\right)C^\top\right) = \operatorname{tr}\left(CXC^\top\right).\end{aligned}$$

The second representation follows in the same way from

$$Y = \int_0^\infty \exp(A^\top t)C^\top C \exp(At)\, dt.$$

Indeed, $A^\top Y + YA + C^\top C = 0$ and cyclic invariance of the trace yields $\|G(s)\|_2^2 = \operatorname{tr}\left(B^\top Y B\right)$. □

The result gives two equivalent ways to compute the H_2 norm. The frequency-domain formula measures the size of the transfer matrix along the imaginary axis. The time-domain formula measures the energy of the impulse response. The algebraic matrix equations provide the computational bridge used in controller synthesis, i.e. instead of evaluating an integral over all frequencies, we solve matrix equations in the state-space representation.
Now the clue of generating an H_2 feedback is to choose an appropriate performance objective as an output-energy criterion. The matrices C and D are chosen such that

$$\mathbf{y}(t)^\top \mathbf{y}(t) = (C\mathbf{x}(t) + D\mathbf{u}(t))^\top (C\mathbf{x}(t) + D\mathbf{u}(t)),$$

where $C^\top C$ defines the state penalty and $D^\top D$ represents the input penalty. Note that $C^\top D = 0$ must be zero to avoid instantaneous output. Hence, we have

Theorem 5.29 (H_2 feedback).
Consider the H_2 problem and suppose $\|G(s)\|_2 = \|g(t)\|_2$ to be finite as well as $C^\top D = 0$, $D^\top D \succ 0$. Then the solution to the H_2 problem is asymptotically stable and the feedback $\mathbf{u}(t) = F\mathbf{x}(t)$ is given by

$$F = -\left(D^\top D\right)^{-1} B^\top P \tag{5.29}$$

where P is the unique symmetric positive semidefinite solution of the algebraic Riccati equation

$$A^\top P + PA - PB\left(D^\top D\right)^{-1} B^\top P + C^\top C = 0. \tag{5.30}$$

Proof. By choosing $Q := C^\top C$, $R := D^\top D$, we can apply the proof of Theorem 5.13 to obtain the assertion. □

From an engineering point of view, the performance output defines what is expensive. The matrix C weights the state deviations, while D weights the actuator effort. The Riccati equation is therefore the algebraic condition that balances fast regulation against moderate input usage. Hence, the H_2 builds upon the LQR in the sense that it reduces the energy of the relevant performance output while avoiding unnecessarily large control inputs.

Algorithm 8 Computation of a basic H_2 state feedback

Input: LTI system (5.1)

1: **procedure** GENERATE $H_2(A, B)$
2: Choose C and D so that $\mathbf{y}^\top(t)\mathbf{y}(t)$ represents the desired state and input penalties.
3: Solve the algebraic Riccati equation (5.30) for its stabilizing symmetric positive definite solution P.
4: Compute F from (5.29) and apply $\mathbf{u}(t) = F\mathbf{x}(t)$.
5: Verify that $A + BF$ is Hurwitz and evaluate the closed-loop H_2 norm.
6: **end procedure**

Output: H_2 feedback $\mathbf{u}(t) = F\mathbf{x}(t)$

Referring back to the production system example, the disturbance input $\mathbf{d}(\cdot)$ may represent variations in material properties, unknown process forces, temperature fluctuations, pressure variations, or short load changes at a drive. The performance output $\mathbf{y}(\cdot)$ may collect web-tension errors, register-position errors, temperature deviations, velocity errors, or weighted actuator signals. A small H_2 norm then means that typical short disturbances generate only little accumulated performance error. Thus, the H_2 criterion is well suited if average disturbance rejection and energy-like performance measures are the main design objectives.
Similarly, for an AD/ADAS vehicle function, $\mathbf{d}(\cdot)$ may represent road excitation, wind gusts, road-slope changes, changes in tire-road interaction, or load variations. The performance output $\mathbf{y}(\cdot)$ may contain lateral tracking error, yaw-rate deviation, velocity error, acceleration error, comfort-relevant body motion, or weighted actuator demand. In this interpretation, the H_2 controller does not protect against the single worst disturbance. Instead, it reduces the average energy transferred from typical disturbances to the variables that are relevant for comfort, safety, and tracking quality.

Similar to LQR, we obtain the (dis-)advantages shown in Table 5.2.

Table 5.2: Advantages and limitations of H_2 control

Advantage		Limitation	
✓	Directly measures output energy caused by disturbances.	✗	Requires a meaningful disturbance and performance-output model.
✓	Uses a clear input-output interpretation of performance.	✗	Result depends on the chosen performance output $\mathbf{y}$.
✓	Connects naturally to LQR in full-state quadratic settings.	✗	Minimizes average energy, not worst-case amplification.
✓	Leads to Riccati-based feedback laws.	✗	Does not directly encode constraints.

5.3 H_∞ Control

The H_2 design in Section 5.2 evaluates disturbance effects in an energy sense. This is useful if disturbances are interpreted as typical or average inputs. In many engineering applications, however, this is not sufficient. A drive may be excited by an unfavorable load-torque oscillation, a vehicle function may be affected by a critical road excitation, and a production process may experience a disturbance close to a lightly damped resonance. In such cases, the relevant question is not only how large the accumulated disturbance effect is on average. Instead, we ask for a guaranteed upper bound on the largest possible amplification from disturbance inputs to performance outputs.

This leads to the H_∞ viewpoint. The closed loop is regarded as an input-output system from an external disturbance $\mathbf{d}(\cdot)$ to a performance output $\mathbf{y}(\cdot)$. The design goal is to find a stabilizing controller such that this input-output map has a prescribed worst-case gain. Thus, in the H_∞ case the supremum norm is used to minimize the highest deviation.

Definition 5.30 (L_∞ norm).
Consider a function $v : [0,\infty) \to \mathbb{R}^{n_y}$. Then we call

$$\|v\|_\infty = \sup_t \|v(t)\| \tag{5.31}$$

the L_∞ norm of the function. If

$$V(s) := \mathcal{L}(f(t)) = \int_0^\infty \exp(-st) \cdot f(t) dt, \qquad s = \alpha + i\omega$$

denotes the Laplace transform of v, then we call

$$\|\mathbf{y}(t)\|_\infty = \sup_v \left\{ \frac{\|G(i\omega) \cdot v\|}{\|v\|} \mid v \neq 0,\ v \in \mathbb{C}^{n_y} \right\} \tag{5.32}$$

the L_∞ norm of the transform.

Again, the terms H_∞ refers to the L_∞ response on the imaginary axis. In the case of H_∞, we will not go into deep but only highlight connections to H_2. The first connection is about conservatism of the controllers.

Theorem 5.31 (H_∞ conservatism).
Consider a function $v : [0, \infty] \to \mathbb{R}^{n_y}$ and a transfer function $G : \mathbb{C} \to \mathbb{C}$. Then we have

$$\|G(s)\|_\infty \geq \frac{\|G(s) \cdot v\|_2}{\|v\|_2} \qquad \forall v \neq 0. \tag{5.33}$$

Proof. Since we have

$$\|G(s) \cdot v\|_2 = \left(\int_{-\infty}^{\infty} \|G(i\omega) \cdot v(i\omega)\|^2 d\omega \right)^{\frac{1}{2}} = \left(\int_{-\infty}^{\infty} \|G(i\omega)\|^2 \cdot \|v(i\omega)\|^2 d\omega \right)^{\frac{1}{2}}$$
$$\leq \sup_\omega \left(\sigma\left(G(i\omega)\right)\right) \cdot \left(\int_{-\infty}^{\infty} \|v(i\omega)\|^2 d\omega \right)^{\frac{1}{2}} = \|G(s)\|_\infty \cdot \|v\|_2$$

where $\sigma(\cdot)$ denotes the maximal singular value, we directly obtain the assertion. □

As a consequence, the H_∞ feedback is always more conservative than the H_2 feedback as it aims to hold down the maximal amplification. The result of Theorem 5.31 be interpreted as the concentrated impact of v close to the frequency range of $\|G(s)\|_\infty$. Hence, the H_∞ norm gives the maximum factor by which the system magnifies the H_2 norm of any input. For this reason, $\|G(s)\|_\infty$ is also referred to as gain of the system.
For a SISO system, Theorem 5.31 reduces to the maximum magnitude of the frequency response. For a MIMO system, the largest singular value describes the largest input-output amplification

over all input directions at a fixed frequency. The H_∞ norm therefore asks for the most critical combination of frequency and input direction.

Remark 5.32 (Worst-case gain interpretation)
The H_∞ norm is the induced L_2 gain of a stable LTI system (5.1)*. It is the smallest number γ such that*

$$\|\mathbf{y}\|_2 \leq \gamma \|\mathbf{d}\|_2$$

for all square-integrable disturbance inputs $\mathbf{d}$ and zero initial condition.

The scalar value γ is therefore an engineering performance certificate. If γ is small, no disturbance signal with finite energy can be amplified strongly into the selected performance output. If γ is large, the closed loop may still be stable, but it may react very sensitively to certain disturbance frequencies or input directions.

Remark 5.33 (Difference from H_∞ to LQR and H_2 design)
The difference between LQR, H_2, and H_∞ design is not only a different mathematical norm. It is a different engineering question.

- *LQR starts from an initial state $\mathbf{x}_0$ and asks for a feedback that minimizes a quadratic trade-off between state deviation and input effort.*
- *H_2 control considers a disturbance-to-output map and minimizes an energy-type measure of the resulting performance output.*
- *H_∞ control considers the same type of disturbance-to-output map, but bounds the worst-case amplification over all disturbance signals.*

Thus, LQR is primarily an optimal regulation method, H_2 control is an average-energy performance method, and H_∞ control is a worst-case robust performance method.

To connect this interpretation with our running example, we consider again the DC motor from Task 4.9.

Task 5.34 (Worst-case interpretation for the DC motor)
Reconsider the DC motor example from Task 4.9. For speed control, a natural performance

variable is the speed error. If actuator usage shall also be penalized, the performance output may be chosen as

$$\mathbf{y} = \begin{bmatrix} q_\omega\, \omega \\ r_u\, \mathbf{u} \end{bmatrix}, \tag{5.34}$$

after shifting the desired speed to the origin. The factor q_ω expresses the importance of speed regulation, whereas r_u expresses the cost of using armature voltage. Assume that the disturbance input is the load torque d_M. What does it mean to reduce the H_∞ norm of the closed-loop transfer from load torque to performance output?

Solution to Task 5.34: Via H_∞, the largest possible amplification from load-torque disturbance energy to performance-output energy is treated. In practical terms, the controller is designed so that no load-torque signal with finite energy, and in particular no unfavorable disturbance frequency, can create an excessive speed error or excessive weighted actuator demand. Compared with an H_2 design, the resulting controller is usually more conservative because the design is governed by the worst disturbance direction and the worst frequency, not by an average energy measure.

For analysis, the following bounded real condition gives a useful certificate. It does not yet synthesize a controller, but it verifies that a given stable disturbance-to-output system has an H_∞ norm smaller than a prescribed number γ.

Theorem 5.35 (Bounded real condition).
Consider the stable LTI system (5.1) *with A Hurwitz and $B = D = 0$. If there exists a symmetric matrix $P = P^\top \succ 0$ such that*

$$A^\top P + PA + C^\top C + \frac{1}{\gamma^2} PEE^\top P \prec 0, \tag{5.35}$$

then $\|G(s)\|_\infty < \gamma$ for $G(s) = C(sId - A)^{-1}E$.

Proof. Using $W(\mathbf{x}) = \mathbf{x}^\top P \mathbf{x}$ as a storage function, the matrix inequality implies

$$\dot{W} + \mathbf{y}^\top \mathbf{y} - \gamma^2 \mathbf{d}^\top \mathbf{d} < 0$$

after completing the square in the disturbance input. Integrating this (dissipation) inequality from 0 to T and using $W \geq 0$ gives $\|\mathbf{y}\|_2 \leq \gamma \|\mathbf{d}\|_2$ for zero initial state. Thus the induced L_2 gain, equivalently the H_∞ norm, is below γ. □

Using the H_∞ norm, we define the H_∞ problem in analogy to the H_2 problem. However, the interpretation is different. The H_2 problem measures the accumulated output energy generated by impulse-like disturbance inputs. The H_∞ problem, in contrast, measures the largest possible amplification from an external disturbance to a performance output. Thus, the disturbance is not fixed to an impulse. It ranges over all square-integrable disturbance signals.
For this reason, we distinguish between the control input $\mathbf{u}(\cdot)$ and the disturbance input $\mathbf{d}(\cdot)$ as in the generalized LTI system (5.17)

Definition 5.36 (H_∞ problem).
Consider the generalized LTI system (5.17) and a state feedback

$$\mathbf{u}(t) = F\mathbf{x}(t). \tag{5.36}$$

Assume that the closed-loop system is asymptotically stable and denote by $G_{\mathbf{d}}(s)$ the closed-loop transfer matrix from the disturbance $\mathbf{d}(\cdot)$ to the performance output $\mathbf{y}(\cdot)$. The corresponding H_∞ cost is

$$\|G_{\mathbf{d}}(s)\|_\infty^2 = \sup_{\mathbf{d}\in L_2,\, \mathbf{d}\neq 0} \frac{\|\mathbf{y}\|_2^2}{\|\mathbf{d}\|_2^2} = \sup_{\omega\in\mathbb{R}} \overline{\sigma}^2\left(G_{\mathbf{d}}(i\omega)\right). \tag{5.37}$$

The H_∞ problem is to find a stabilizing feedback $\mathbf{u}(t) = F\mathbf{x}(t)$ such that

$$\|G_{\mathbf{d}}(s)\|_\infty < \gamma \tag{5.38}$$

for a prescribed performance level $\gamma > 0$, or to minimize the smallest achievable value of γ.

The scalar γ is a worst-case performance bound. If (5.38) holds, then every disturbance signal with finite energy satisfies

$$\|\mathbf{y}\|_2 < \gamma\|\mathbf{d}\|_2. \tag{5.39}$$

Hence, no admissible disturbance signal can be amplified by more than the factor γ into the chosen performance output. For the DC motor, this means that the effect of all load-torque disturbance signals on the selected performance variables is uniformly bounded. This is the main difference to the H_2 design, where the average energy effect of impulse-like disturbances is minimized.
The following result gives a Riccati-based state-feedback construction for the standard case in which the state and input weights are separated in the performance output.

Theorem 5.37 (H_∞ feedback).
Consider the generalized LTI system (5.17) *and assume*

$$D^\top D \succ 0, \qquad C^\top D = 0. \tag{5.40}$$

Let $\gamma > 0$ *be given. Suppose that there exists a symmetric positive semidefinite matrix* $P = P^\top \succeq 0$ *solving the algebraic Riccati equation*

$$A^\top P + PA + C^\top C + \gamma^{-2} PEE^\top P - PB\left(D^\top D\right)^{-1} B^\top P = 0 \tag{5.41}$$

such that $A + BF$ *is Hurwitz. Then the feedback*

$$\mathbf{u}^\star(t) = F\mathbf{x}(t) \tag{5.42}$$

with

$$F = -\left(D^\top D\right)^{-1} B^\top P \tag{5.43}$$

asymptotically stabilizes the system and guarantees

$$\|G_{\mathbf{d}}\|_\infty \leq \gamma. \tag{5.44}$$

Proof. With the feedback (5.43), the closed-loop system is

$$\dot{\mathbf{x}}(t) = (A + BF)\,\mathbf{x}(t) + E\mathbf{d}(t), \qquad \mathbf{y}(t) = (C + DF)\,\mathbf{x}(t). \tag{5.45}$$

Define $A_{\mathrm{cl}} := A + BF$, $C_{\mathrm{cl}} := C + DF$. By assumption, A_{cl} is Hurwitz. Using (5.43) and $C^\top D = 0$, we obtain

$$\begin{aligned} A_{\mathrm{cl}}^\top P + PA_{\mathrm{cl}} + C_{\mathrm{cl}}^\top C_{\mathrm{cl}} + \gamma^{-2} PEE^\top P &= \\ &= A^\top P + PA + C^\top C + \gamma^{-2} PEE^\top P + F^\top B^\top P + PBF + F^\top D^\top DF \\ &= A^\top P + PA + C^\top C + \gamma^{-2} PEE^\top P - PB\left(D^\top D\right)^{-1} B^\top P. \end{aligned}$$

The last expression is zero by (5.41). Hence,

$$A_{\mathrm{cl}}^\top P + PA_{\mathrm{cl}} + C_{\mathrm{cl}}^\top C_{\mathrm{cl}} + \gamma^{-2} PEE^\top P = 0. \tag{5.46}$$

Now, the bounded real condition from Theorem 5.35 implies $\|G_{\mathbf{d}}\|_\infty \leq \gamma$ and the assertion

follows. □

Note that the assumption (5.40) mean that actuator usage is weighted in the performance output and a mixed state-input term is avoided. It is the same simplifying structure that also appears in the H_2 feedback formula.

The Riccati equation (5.41) has the same feedback term as the H_2 and LQR equations. This term describes the ability of the actuator to reduce the performance output. The additional term $\gamma^{-2}PEE^\top P$ is the characteristic H_∞ term. It represents the adversarial disturbance channel. The smaller γ is chosen, the larger this term becomes. Thus, demanding a small worst-case gain makes the Riccati equation harder to satisfy and the resulting feedback typically more conservative.

Algorithm 9 Computation of a basic H_∞ state feedback

Input: LTI system (5.17) and performance level $\gamma > 0$

procedure GENERATE $H_\infty(A, B, C, D, E, \gamma)$
  Check that $D^\top D \succ 0$ and $C^\top D = 0$.
  Solve algebraic Riccati equation (5.41) for a symmetric positive semidefinite solution P.
  Compute F via (5.43)
  Verify that $A + BF$ is Hurwitz.
  Form the closed-loop transfer matrix $G_{\mathbf{d}}(s) = (C + DF)\,(s\mathrm{Id} - A - BF)^{-1}\,E$.
  Verify that $\|G_{\mathbf{d}}\|_\infty \leq \gamma$.
  if (5.41) has no stabilizing solution or (5.44) is not satisfied **then**
    Increase γ or modify the performance weights in C and D.
  end if
end procedure

Output: H_∞ feedback $\mathbf{u} = F\mathbf{x}$

In practice, the value of γ is often not known beforehand. One therefore starts with a feasible, sufficiently large value of γ and then decreases it until the Riccati equation no longer has a stabilizing solution. This gives the smallest achievable worst-case gain for the chosen disturbance channel and performance output. For the DC motor, this procedure quantifies how strongly the effect of load-torque disturbances on speed error and actuator usage can be attenuated.

In a production setting, the H_∞ interpretation becomes relevant when the process must remain acceptable under unfavorable disturbance patterns. Examples are periodic material-thickness variations close to a lightly damped tension mode, thermal disturbances close to a slow dominant time constant, pressure oscillations, or load oscillations at a drive. The H_∞ controller is then not tuned for an average disturbance effect. It is tuned to guarantee that the worst modeled amplification from $\mathbf{d}(\cdot)$ to $\mathbf{y}(\cdot)$ remains below the prescribed bound γ.

Similarly, in an AD/ADAS vehicle function, the H_∞ viewpoint is useful when critical disturbance scenarios dominate the design. Crosswind gusts, road irregularities, slope changes, load variations, or uncertain actuator dynamics may excite frequencies at which the vehicle response

is particularly sensitive. A small H_∞ bound means that the selected performance variables, for example lateral deviation, yaw-rate error, acceleration error, or weighted actuator usage, cannot be amplified beyond the certified gain for the modeled disturbance channel.

Remark 5.38 (Difference to the H_2 algorithm)
The H_2 algorithm solves one Riccati equation for a fixed quadratic output-energy criterion. The H_∞ algorithm additionally depends on the performance level γ. Hence, H_∞ synthesis exhibits a feasibility problem that can be addressed by reducing γ.

Table 5.3: Advantages and limitations of H_∞ control

Advantage	Limitation
✓ Directly addresses worst-case disturbance amplification.	✗ Can be conservative because it is governed by the worst frequency.
✓ Closely connected to robustness analysis and gain bounds.	✗ Requires careful choice of performance channels and weights.
✓ Gives certificates through Riccati inequalities.	✗ The resulting design may be harder to interpret than LQR tuning.

This chapter has shown how stabilizing feedback laws can be designed optimally by LQR, H_2, and H_∞ methods. All these designs assume that the state $\mathbf{x}(t)$ is available for feedback. In many engineering systems, however, only selected outputs are measured. Chapter 6 therefore turns to the optimal reconstruction of the state from measurements and introduces observer design and the Kalman filter.

CHAPTER 6

OPTIMAL OBSERVATION

State-feedback laws such as the LQR, H_2, and H_∞ controllers from Chapter 5 require access to the state $\mathbf{x}(t)$. In technical systems, however, only selected outputs are measured, often affected by noise, delays, bias, or limited bandwidth. This chapter therefore studies how state estimates can be reconstructed from input-output data and used in feedback control.

To this end, we discussed the necessary concepts of observability and detectability in Chapter 4. Now, we turn these structural concepts into a design procedure. Sections 6.1 and 6.2 introduces the Luenberger filter as a deterministic state estimator. The observer gain is chosen such that the estimation error dynamics are stable, which gives a direct dual counterpart to state-feedback pole placement from Chapter 4. This part emphasizes when a model-based observer is sufficient, how the innovation enters the estimator, and how the convergence speed of the estimate is shaped by the observer poles.

Section 6.3 then combines state estimation with feedback control. The separation principle shows that, for linear systems under the stated assumptions, the controller dynamics and observer error dynamics can be designed independently. Hence, a stabilizing state feedback from Chapter 5 can be implemented with an estimated state if the observer error converges sufficiently fast. This result explains why full-state optimal controllers remain useful even when not all states are measured.

Finally, Sections 6.4 and 6.5 introduce the Kalman filter as an optimal observer for systems affected by process and measurement noise. Instead of assigning observer poles directly, the observer gain is computed from a Riccati equation that balances model uncertainty against measurement uncertainty. The resulting filter has the same innovation structure as the Luenberger filter, but its gain has a statistical interpretation.

The DC-motor example, the production examples, and the AD/ADAS driving-function examples are used throughout to show how deterministic and stochastic observers differ in tuning effort,

robustness, and engineering interpretation.
As in the previous chapters, we use the following standing assumptions unless stated otherwise.

Assumption 6.1
All finite-dimensional state spaces are real vector spaces $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ with the Euclidean norm $\|\cdot\|_2$. Continuous-time inputs $\mathbf{u}(t)$ are assumed to be piecewise continuous on compact intervals with time indicated by t, and discrete-time inputs $\mathbf{u}(k)$ are arbitrary sequences of compatible dimension with time index k. For stability and convergence statements, equilibria are shifted to the origin whenever this simplifies notation. The standing deterministic system model is the continuous-time LTI system

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0, \tag{6.1a}$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t). \tag{6.1b}$$

The input $\mathbf{u}(\cdot)$ is known to the observer, the output $\mathbf{y}(\cdot)$ is measured, and the state $\mathbf{x}(\cdot)$ is not necessarily measured. The reconstructed state is denoted by $\hat{\mathbf{x}}(\cdot)$. The pair (A, C) is assumed to be detectable. Whenever arbitrary observer pole placement is used, the stronger assumption that (A, C) is observable is imposed. For stochastic filtering results, process and measurement noise are assumed to be zero-mean, mutually independent white-noise processes with positive semidefinite process covariance and positive definite measurement covariance of compatible dimensions.

6.1 Observer Structure and Innovation Feedback

Given the setting of our standing Assumption 6.1, we must reconstruct the state $\mathbf{x}(\cdot)$ from input-output data. A purely simulated model

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + B\mathbf{u}(t) \tag{6.2}$$

would be exact only if the initial condition and model were exact. This is not a filter in the practical sense because measured outputs do not correct the estimate. The central idea of observer design is to feed back the output mismatch, also called the *innovation*.

Definition 6.2 (Innovation).
Consider an LTI system (6.1), an estimate $\hat{\mathbf{x}}(\cdot)$ following (6.2) induces the predicted output

$$\hat{\mathbf{y}}(t) = C\hat{\mathbf{x}}(t) + D\mathbf{u}(t). \tag{6.3}$$

The signal

$$\mathbf{y}(t) - \hat{\mathbf{y}}(t) = \mathbf{y}(t) - C\hat{\mathbf{x}}(t) - D\mathbf{u}(t) \tag{6.4}$$

is called the *innovation* or *output residual*.

The innovation is not simply the sensor noise. It also contains initial-state errors, model errors, discretization effects, disturbances, and all state components that are visible through the output but not yet correctly represented by the estimate. A filter therefore has to decide how strongly this innovation should be used.

Definition 6.3 (Luenberger filter).
For the LTI system (6.1), a *Luenberger filter* is a dynamic system

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + B\mathbf{u}(t) + L\left(\mathbf{y}(t) - C\hat{\mathbf{x}}(t) - D\mathbf{u}(t)\right), \tag{6.5}$$

where L is the *observer gain*.

The sign convention in (6.5) is chosen such that the correction is added when the measured output is larger than the predicted output. Equivalently, we could write the observer in the dynamic-output-feedback form used in the observer literature:

$$\dot{\hat{\mathbf{x}}} = (A - LC)\,\hat{\mathbf{x}} + (B - LD)\,\mathbf{u} + L\mathbf{y}. \tag{6.6}$$

This form makes clear that the measured output is an input to an auxiliary dynamic system.

Definition 6.4 (Estimation error).
For a state trajectory $\mathbf{x}(\cdot)$ and an estimate $\hat{\mathbf{x}}(\cdot)$ we define the estimation error by

$$e(t) := \mathbf{x}(t) - \hat{\mathbf{x}}(t). \tag{6.7}$$

With this convention, positive error means that the true state is larger than the estimate in the corresponding direction. Subtracting (6.5) from (6.1a) gives the central error equation.

Proposition 6.5 (Luenberger error dynamics).
Given an LTI system (6.1) *and the Luenberger filter* (6.5)*, the estimation error satisfies*

$$\dot{e}(t) = (A - LC)\,e(t). \tag{6.8}$$

Proof. Insert $\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$ into (6.5). Then the innovation becomes $C(\mathbf{x}(t) - \hat{\mathbf{x}}(t))$. Using $e(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t)$, subtracting the observer dynamics from the plant dynamics yields

$$\dot{e}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) - A\hat{\mathbf{x}}(t) - B\mathbf{u}(t) - LC\left(\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right) = (A - LC)e(t).$$

Thus, known inputs cancel and the convergence of the estimate is determined solely by the observer error matrix. □

Note that the input $\mathbf{u}(\cdot)$ does not appear in the error dynamics (6.7). Proposition 6.5 gives a direct design rule. The estimate converges to the true state if $A - LC$ is Hurwitz. The role of L is therefore dual to the role of a state-feedback matrix in stabilization.

Algorithm 10 Computation of a Luenberger filter

Input: LTI system (6.1), observer gain L, initial estimate $\hat{\mathbf{x}}(0)$

procedure LUENBERGER FILTER($A, B, C, D, L, \hat{\mathbf{x}}(0)$)
  for each time instant t **do**
    Measure $\mathbf{y}(t)$ and read the known input $\mathbf{u}(t)$.
    Compute the predicted output $\hat{\mathbf{y}}(t) = C\hat{\mathbf{x}}(t) + D\mathbf{u}(t)$.
    Compute the innovation $\mathbf{y}(t) - \hat{\mathbf{y}}(t)$.
    Integrate (6.5) one step forward.
    Provide $\hat{\mathbf{x}}(t)$ to monitoring or feedback.
  end for
end procedure

Output: Luenberger state estimate $\hat{\mathbf{x}}(t)$

Task 6.6 (Running example: innovation in the DC motor)
Use the DC-motor state $\mathbf{x} = \begin{bmatrix} i_a & \omega & \varphi \end{bmatrix}^\top$ *from Task 4.9. Assume that the angular position is measured and that the armature voltage is known. Write the innovation used by a Luenberger filter.*

Solution to Task 6.6: If position is the measured output, one may choose $C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, $D = 0$. Hence, $\mathbf{y}(t) = \varphi(t)$ and the predicted output is $\hat{\mathbf{y}}(t) = \hat{\varphi}(t)$. The innovation is $\varphi(t) - \hat{\varphi}(t)$. The observer gain maps this scalar mismatch into corrections for estimated current, speed, and position. Thus, even if only position is measured, the filter may correct all state components if the system is observable or detectable.

In a production line, the same innovation structure occurs if a quality sensor measures a final product temperature or position while internal thermal or mechanical states are estimated. The

innovation is the difference between the measured quality variable and the model-predicted quality variable. In a vehicle function, the innovation may be a yaw-rate residual, a lateral-position residual, or a velocity residual obtained from sensor fusion. The filter gain decides how strongly this residual updates unmeasured motion states.
The preceding discussion defines the observer structure, but not yet how the gain should be chosen. We now turn from the architecture of the filter to the structural question of whether a convergent observer exists and how its error dynamics can be shaped.

6.2 Existence and Design of Luenberger Filters

The Luenberger filter exists as a convergent observer exactly under the detectability condition introduced in Chapter 4. Observability is sufficient, but it is not necessary. Unobservable modes are allowed if they are already stable, because then their estimation error decays without being corrected by the output.

Theorem 6.7 (Existence of a convergent Luenberger filter).
Consider the LTI system (6.1)*. There exists an observer gain* L *such that* $A - LC$ *is Hurwitz if and only if the pair* (A, C) *is detectable.*

Proof. By the duality result from Chapter 4, detectability of (A, C) is equivalent to stabilizability of $(A^\top, C^\top)$. Stabilizability of this dual pair means that there exists a feedback matrix F such that $A^\top + C^\top F$ is Hurwitz. Choosing $L = -F^\top$ gives

$$A - LC = \left(A^\top + C^\top F\right)^\top,$$

which has the same eigenvalues and is therefore Hurwitz. Conversely, if such a gain exists, then $A^\top - C^\top L^\top$ is a stabilizable closed-loop matrix for the dual pair. Hence, $(A^\top, C^\top)$ is stabilizable and (A, C) is detectable. □

If (A, C) is observable, the eigenvalues of $A - LC$ can be assigned arbitrarily by dual pole placement. This is the observer counterpart of state-feedback pole placement. In engineering applications, however, arbitrary fast observer poles are not automatically desirable. A very fast observer reacts strongly to noise and model mismatch. Conversely, a slow observer filters noise more effectively but may deliver delayed state information to the controller.

Corollary 6.8 (Observer pole placement).
Consider the LTI system (6.1) *and the Luenberger filter* (6.5)*. If* (A, C) *is observable, then for*

every set of desired eigenvalues that is symmetric with respect to the real axis there exists a real observer gain L such that $A - LC$ has exactly these eigenvalues.

Proof. Observability of (A, C) is equivalent to controllability of the dual pair $(A^\top, C^\top)$. The pole-placement theorem for state feedback can therefore be applied to the dual pair. It gives a dual feedback matrix whose closed-loop eigenvalues are the prescribed observer poles. Transposition and the sign convention $L = -F^\top$ yield the desired observer gain. □

Algorithm 11 Design of a Luenberger filter by dual pole placement

Input: LTI system (6.1) and desired observer poles

1: **procedure** DESIGN LUENBERGER GAIN(A, C)
2: Check observability or at least detectability of (A, C) by Theorems 4.29 or 4.36.
3: Form the dual pair $(A^\top, C^\top)$.
4: Compute state-feedback F for the dual pair such that $A^\top + C^\top F$ has the desired poles.
5: Set $L = -F^\top$.
6: Verify that $A - LC$ is Hurwitz.
7: **end procedure**

Output: Observer gain L

Task 6.9 (Choosing observer speed for the DC motor)
For the DC motor with a position sensor, explain qualitatively what happens if the observer poles are chosen much faster than the closed-loop controller poles.

Solution to Task 6.9: Fast observer poles make the estimated current and speed converge quickly after an initial-state error. This is attractive because the controller receives useful estimates soon after start-up. At the same time, the gain L usually becomes larger. Then position measurement noise is injected more strongly into all estimated states, in particular into the speed estimate. In practice, the observer should be faster than the relevant controller dynamics but not so fast that sensor noise and unmodeled high-frequency dynamics dominate the estimate.

For production systems, the same trade-off appears in tension, temperature, or position observers. A fast observer reacts quickly to process changes but may amplify noisy quality measurements. A slow observer gives smoother estimates but may hide rapid disturbances. In AD/ADAS functions, overly fast observer poles may turn sensor noise into steering or braking activity, whereas overly slow poles may delay the detection of lateral motion, road grade, or tire-force changes.

The deterministic Luenberger filter is a standard tool because it is transparent, model-based, and easy to implement. Its strengths and limitations are summarized in Table 6.1.

Table 6.1: Advantages and limitations of Luenberger filters

Advantage	Limitation
✓ Gives a clear deterministic construction for state estimation.	✗ Requires a state-space model and a detectable sensor configuration.
✓ Observer poles can be placed by duality to state feedback.	✗ Pole locations do not directly encode noise or uncertainty levels.
✓ The error dynamics are autonomous for known inputs.	✗ Large gains may amplify measurement noise and unmodeled dynamics.
✓ Easy to implement as a dynamic system driven by input and output.	✗ Tuning is often heuristic if no stochastic or optimization model is used.

A convergent observer is useful only if its estimate can be connected safely to the feedback laws designed earlier. The next step is therefore to analyze what happens when a state-feedback controller is driven by the estimated state rather than by the true state.

6.3 Observer-based Feedback and Separation

Once a state estimate is available, a natural implementation of a state-feedback law is

$$\mathbf{u}(t) = F\hat{\mathbf{x}}(t). \tag{6.9}$$

Here, F may be an LQR, H_2, H_∞, or pole-placement feedback from Chapter 5. The resulting controller is a dynamic output feedback because the signal $\hat{\mathbf{x}}(\cdot)$ is generated by the observer dynamics.

For clarity, we first ignore direct feedthrough by setting $D = 0$. Combining

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + BF\hat{\mathbf{x}}(t),\\ \dot{\hat{\mathbf{x}}}(t) &= A\hat{\mathbf{x}}(t) + BF\hat{\mathbf{x}}(t) + L(C\mathbf{x}(t) - C\hat{\mathbf{x}}(t))\end{aligned}$$

with $e(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t)$ gives the triangular representation

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A + BF & -BF \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ e(t) \end{bmatrix}. \tag{6.10}$$

This form shows why controller and observer can be designed separately in the nominal linear case.

Theorem 6.10 (Separation principle).
Consider the LTI system (6.1) *with* $D = 0$*. Suppose that* F *is chosen such that* $A + BF$ *is Hurwitz and that* L *is chosen such that* $A - LC$ *is Hurwitz. Then the observer-based feedback* (6.9) *yields an exponentially stable closed loop.*

Proof. The coordinate transformation from $(\mathbf{x}, \hat{\mathbf{x}})$ to $(\mathbf{x}, e)$ gives the block upper-triangular system matrix in (6.10). The eigenvalues of a block triangular matrix are the union of the eigenvalues of its diagonal blocks. Hence, the closed-loop eigenvalues are those of $A + BF$ and those of $A - LC$. Both matrices are Hurwitz by assumption, so the complete observer-based closed loop is exponentially stable. □

The theorem is powerful but should be interpreted correctly. It is a nominal linear stability result. It does not say that arbitrary observer tuning is harmless. A noisy estimate may still degrade performance, and actuator constraints may still be violated. The result says that, under the model assumptions, stabilization and observation can be designed as two separate linear problems.

Corollary 6.11 (Solvability of stabilization with measured output).
For the LTI system (6.1) *with* $D = 0$*, the observer-based stabilization problem is solvable if* (A, B) *is stabilizable and* (A, C) *is detectable.*

Proof. Stabilizability of (A, B) implies that a feedback F exists with $A + BF$ Hurwitz. Detectability of (A, C) implies by Theorem 6.7 that an observer gain L exists with $A - LC$ Hurwitz. The separation principle then gives exponential stability of the observer-based closed loop. □

In the DC-motor example, one may first design an LQR for the full state and then replace the true state by the estimated state. If the position sensor makes the motor model detectable and the voltage input stabilizes the relevant states, the nominal observer-based controller is stable. In production automation, this justifies using an observer to estimate hidden process variables before applying an optimal feedback. In vehicle functions, it justifies a modular architecture in which estimation and motion control are tuned separately at first and then validated together.

The separation principle tells us that a stabilizing observer may be combined with a stabilizing controller. It does not, however, prescribe a particular observer gain. The next section therefore replaces direct pole placement by a Riccati-based design that mirrors the optimal-control constructions of Chapter 5.

6.4 Optimal Observation by Duality

The Luenberger filter gives a convergent estimate once the observer poles have been selected. The remaining question is how to select the gain systematically. The optimal-control viewpoint from Chapter 5 gives one answer. Instead of placing poles directly, we penalize estimation error and correction effort. Hence, one approach is to interpret optimal observation as the dual counterpart of optimal stabilization.

To see the idea, consider the error dynamics with an artificial correction input,

$$\dot{e}(t) = Ae(t) + \mathbf{d}(t), \qquad \mathbf{y}_e(t) = Ce(t). \tag{6.11}$$

The signal $\mathbf{d}(\cdot)$ represents a model correction used to explain the measured output mismatch. If large corrections are cheap, the estimate may follow measurements aggressively. If large corrections are expensive, the estimate remains closer to the model prediction.

Following the deterministic infinite-horizon construction used in LQR, one may view state estimation as a least-squares approximation problem over past measurements. In this interpretation, the estimator uses past output data to reconstruct the present state. By inverting time, however, this problem has the structure of an LQR for the dual system. We formulate the resulting steady-state design directly.

Definition 6.12 (Quadratic observer design problem).
Let $\hat{Q} = \hat{Q}^\top \succeq 0$ and $\hat{R} = \hat{R}^\top \succ 0$. The quadratic observer design problem is to choose a linear innovation gain L such that the error dynamics

$$\dot{e}(t) = (A - LC)e(t)$$

is stable and the trade-off between output-error reduction and correction intensity, represented by $\hat{Q}$ and $\hat{R}$, is optimized in the dual LQ sense.

Theorem 6.13 (Riccati design of an optimal observer gain).
Consider an LTI system (6.1) *with* $D = 0$. *Assume that* (A, C) *is detectable,* $(A, \hat{Q}^{1/2})$ *is*

stabilizable, $\hat{Q} \succeq 0$, and $\hat{R} \succ 0$. Then there exists a unique positive semidefinite stabilizing solution $P = P^\top \succeq 0$ of the algebraic Riccati equation

$$AP + PA^\top - PC^\top \hat{R}^{-1} CP + \hat{Q} = 0. \tag{6.12}$$

Moreover,

$$L = PC^\top \hat{R}^{-1} \tag{6.13}$$

is an optimal steady-state observer gain and $A - LC$ is Hurwitz.

Proof. Apply the continuous-time LQR result from Chapter 5 to the dual pair $(A^\top, C^\top)$ with state weighting $\hat{Q}$ and input weighting $\hat{R}$. The assumptions of the theorem are precisely the stabilizability and detectability conditions required for the dual Riccati equation to have a unique positive semidefinite stabilizing solution. Written in transpose form, this Riccati equation is exactly (6.12). The optimal feedback gain of the dual problem is $\hat{R}^{-1}CP$. Thus the dual closed-loop matrix is $A^\top - C^\top \hat{R}^{-1} CP$. Transposing this matrix and using $L = PC^\top \hat{R}^{-1}$ gives the observer error matrix $A - LC$. Since the dual Riccati solution is stabilizing, this matrix is Hurwitz. □

The matrices in (6.12) should be interpreted carefully.

Remark 6.14
The matrix $\hat{Q}$ describes how strongly model uncertainty or state-error directions should be represented in the observer design. The matrix $\hat{R}$ penalizes reliance on the innovation. In the stochastic Kalman interpretation below, these matrices become process- and measurement-uncertainty descriptions.

In a deterministic teaching setting, we can use these matrices as tuning parameters.

Algorithm 12 Riccati-based observer-gain design

Input: LTI system (6.1) with $D = 0$, weights $\hat{Q}$ and $\hat{R}$
1: **procedure** DESIGN RICCATI OBSERVER($A, C, \hat{Q}, \hat{R}$)
2: Check detectability of (A, C).
3: Choose $\hat{Q} = \hat{Q}^\top \succeq 0$ and $\hat{R} = \hat{R}^\top \succ 0$.
4: Solve (6.12) for the stabilizing solution $P = P^\top \succeq 0$.
5: Compute $L = PC^\top \hat{R}^{-1}$.
6: Verify that $A - LC$ is Hurwitz.
7: **end procedure**
Output: Observer gain L

Task 6.15 (Running example: Riccati tuning for the DC motor)
For a DC-motor observer with position measurement, explain qualitatively what happens if the weight $\hat{R}$ assigned to the measurement innovation is increased.

Solution to Task 6.15: A larger $\hat{R}$ makes innovation feedback more expensive in the Riccati design. The gain L tends to become smaller, so the observer trusts the model prediction more and the measured position less. The estimates become smoother and less sensitive to position noise, but the convergence from an incorrect initial state or an unmodeled load disturbance becomes slower.

For production systems, the Riccati viewpoint is useful if different sensors and model components have different reliability. A noisy temperature sensor should not dominate the estimate of slow thermal states. A well-calibrated tension sensor may be allowed to correct the model more strongly. In a vehicle function, high-quality inertial measurements can receive strong weight, while camera-based or environment-dependent signals may be weighted more cautiously depending on delay, confidence, and operating conditions.
The Riccati observer design above can be read as a deterministic dual design. The Kalman filter gives the same innovation structure a stochastic interpretation by assigning reliability levels to the model and to the measurements.

6.5 Kalman Filter as Optimal Observation

The Kalman filter uses the same innovation structure as the Luenberger filter, but gives the gain an uncertainty-based interpretation. In continuous time, the steady-state version is often called the Kalman–Bucy filter. We use the model

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) + \mathbf{d}(t), \tag{6.14a}$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) + \mathbf{y_d}(t). \tag{6.14b}$$

Here, $\mathbf{d}(\cdot)$ represents process uncertainty and $\mathbf{y_d}(\cdot)$ represents measurement uncertainty. If the process model is uncertain, the filter should react more strongly to measurements. If the measurements are uncertain, the filter should rely more strongly on the model.

Definition 6.16 (Steady-state Kalman filter problem).
For the system (6.14), assume process and measurement uncertainties with weights $\hat{Q} \succeq 0$ and

$\hat{R} \succ 0$, respectively. The steady-state Kalman filter problem is to compute a causal estimate $\hat{\mathbf{x}}(t)$ from $\mathbf{u}(\tau)$ and $\mathbf{y}(\tau)$, $0 \leq \tau \leq t$, such that the estimation error is minimized according to the assumed uncertainty weights.

The word causal is important. The estimate at time t may only use data available up to time t. This is why the Kalman filter is implemented recursively: prediction by the model is followed by correction through the innovation.

Theorem 6.17 (Continuous-time steady-state Kalman filter).
Consider the noisy LTI system (6.14)*. Assume that* (A, C) *is detectable and that* $\left(A, \hat{Q}^{1/2}\right)$ *is stabilizable. Let* $\hat{Q} = \hat{Q}^\top \succeq 0$ *denote the process uncertainty weighting and* $\hat{R} = \hat{R}^\top \succ 0$ *denote the measurement uncertainty weighting. Then the algebraic Riccati equation*

$$AP + PA^\top - PC^\top \hat{R}^{-1} CP + \hat{Q} = 0 \tag{6.15}$$

has a unique positive semidefinite stabilizing solution $P = P^\top \succeq 0$*. With*

$$L = PC^\top \hat{R}^{-1}, \tag{6.16}$$

the steady-state Kalman filter is given by

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + B\mathbf{u}(t) + L\left(\mathbf{y}(t) - C\hat{\mathbf{x}}(t) - D\mathbf{u}(t)\right). \tag{6.17}$$

The matrix $A - LC$ *is Hurwitz, and the filter minimizes the steady-state estimation-error covariance with respect to the assumed process and measurement uncertainty weights.*

Proof. The result follows from the continuous-time Riccati result applied to the dual observer problem. Detectability of (A, C) and stabilizability of $\left(A, \hat{Q}^{1/2}\right)$ are precisely the conditions ensuring the existence of a unique positive semidefinite stabilizing solution of (6.15). With this solution, the optimal steady-state correction is proportional to the innovation, with gain $L = PC^\top \hat{R}^{-1}$. Substitution gives the filter equation (6.17). Since the Riccati solution is stabilizing, the corresponding estimation-error matrix $A - LC$ is Hurwitz. When the uncertainty weights are interpreted as process and measurement noise covariances, P is the corresponding steady-state estimation-error covariance. □

The theorem makes explicit that the steady-state Kalman filter has the same innovation structure as a Luenberger filter. The difference is not the architecture, but the way the observer gain is selected. In a Luenberger filter, the gain is usually chosen to assign desired error dynamics. In

the Kalman filter, the gain is computed from a Riccati equation that balances the assumed process uncertainty against the assumed measurement uncertainty.
This interpretation is important for tuning. A larger process uncertainty weighting means that the model prediction is regarded as less reliable; the filter therefore reacts more strongly to the innovation. A larger measurement uncertainty weighting means that the measured output is regarded as less reliable; the filter therefore relies more on the model and uses a smaller correction. Thus, Kalman tuning is best understood as specifying the relative trust in model and sensor information rather than as an abstract choice of matrix weights.

Algorithm 13 Computation of a steady-state Kalman filter

Input: LTI system (6.14), weights $\hat{Q}$ and $\hat{R}$
1: **procedure** GENERATE KALMAN FILTER($A, B, C, D, \hat{Q}, \hat{R}$)
2: Check detectability of (A, C).
3: Interpret/tune $\hat{Q}$ as process- and $\hat{R}$ as measurement-uncertainty weight.
4: Solve the algebraic Riccati equation (6.15) for the stabilizing solution P.
5: Compute $L = PC^\top \hat{R}^{-1}$.
6: Implement the recursive filter (6.17).
7: Validate the estimate under representative noise, disturbance, and model-mismatch scenarios.
8: **end procedure**
Output: Kalman gain L and estimate $\hat{\mathbf{x}}(\cdot)$

Task 6.18 (Running example: Kalman tuning for the DC motor)
For the DC-motor observer, explain what happens if the assumed position-sensor uncertainty is increased. Explain also what happens if the assumed process uncertainty in the mechanical equation is increased because the load torque is poorly known.

Solution to Task 6.18: If the position measurement is assumed to be less reliable, the filter uses the position innovation less strongly. The position and speed estimates become smoother, but they react more slowly to actual deviations. If the mechanical process uncertainty is increased, the filter trusts the mechanical model less. Then deviations in the measured output are interpreted more readily as evidence that the speed or load-related state estimate should be corrected.

In production systems, the Kalman interpretation is useful when sensors and models have known or estimated reliability. A temperature sensor with strong noise should not produce high-frequency actuator-state corrections. A process model with uncertain material parameters should be corrected more strongly by reliable measurements. In vehicle functions, the same idea appears in

sensor fusion: inertial sensors, wheel-speed sensors, cameras, radar, and map information may have different uncertainty levels and update rates. The Kalman gain encodes how these signals are balanced in the state estimate.

The main strengths and limitations of the Kalman filter are summarized in Table 6.2.

Table 6.2: Advantages and limitations of Kalman filters

Advantage	**Limitation**
✓ Provides an optimal innovation gain for a specified uncertainty model.	✗ Result depends strongly on the assumed process/measurement weights.
✓ Has the same implementable structure as a Luenberger filter.	✗ Incorrect uncertainty descriptions may cause poor or overconfident estimates.
✓ Naturally supports sensor-fusion interpretations.	✗ The continuous-time filter still requires numerical integration and validation.
✓ The Riccati equation gives a systematic tuning framework.	✗ Linearity and stationarity assumptions may be restrictive for strongly nonlinear systems.

This chapter has developed state observation as the counterpart of state-feedback design. A Luenberger filter corrects a model-based state estimate by feeding back the innovation between measured and predicted output. Detectability is the structural condition that guarantees that the estimation error can be made asymptotically stable. Observability, on the other hand, gives the stronger property needed for arbitrary observer pole placement.

The separation principle shows that, for nominal LTI systems, stabilizing state feedback and convergent state observation can be designed independently and then combined in an observer-based feedback loop. This justifies the use of full-state controllers from Chapter 5 even when the state is not directly measured, but it does not replace engineering validation under noise, delays, constraints, and model mismatch.

Last, the Kalman filter has been introduced as an optimal innovation filter. It has the same structure as a Luenberger filter, but its gain is computed from a Riccati equation that balances assumed process uncertainty against assumed measurement uncertainty. The central message is that observer design cannot create missing information. Instead, the model and sensor configuration determine what can be reconstructed, while Luenberger and Kalman design determine how the available information is used.

BIBLIOGRAPHY

[1] ANDERSON, B.D.O. ; MOORE, J.B.: *Optimal control: linear quadratic methods*. Courier Corporation, 2007

[2] FÖLLINGER, O.: *Regelungstechnik*. 13. überarbeitete Auflage. VDE-Verlag, 2022

[3] HINRICHSEN, D. ; PRITCHARD, A.J.: *Mathematical system theory I: Modeling, state space analysis and robustness*. Springer, 2010

[4] ISIDORI, A.: *Nonlinear Control Systems*. 3rd edition. Springer, 1995

[5] KALMAN, R.E.: Contributions to the theory of optimal control. In: *Boletin de la Sociedad Matematica Mexicana* 5 (1960), No. 2, pp. 102–119

[6] LUNZE, J.: *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*. 11. überarbeitete und ergänzte Auflage. Springer, 2016

[7] LUNZE, J.: *Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung*. 9. überarbeitete Auflage. Springer, 2016

[8] LUNZE, J.: *Automatisierungstechnik*. 5. Auflage. DeGruyter, 2020

[9] SKOGESTAD, S. ; POSTLETHWAITE, I.: *Multivariable feedback control: analysis and design*. John Wiley & Sons, 2005

[10] SONTAG, E.D.: *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer, 1998

[11] UNBEHAUEN, H.: *Regelungstechnik I*. Vieweg / Teubner, 2007

Jürgen Pannek
Institute for Intermodal Transport and Logistic Systems
Hermann-Blenck-Str. 42
38519 Braunschweig

During summer term 2026 I give the lecture to the module *Control Engineering 2 (Regelungstechnik 2)* at the Technical University of Braunschweig. To structure the lecture and support my students in their learning process, I prepared these lecture notes. The aim of the lecture notes is to provide participating students with knowledge of terms of system theory and control engineering. Moreover, students shall be enabled to understand complex control structures, apply control schemes and analyze control systems. After successfully completing the module, students shall additionally be able to apply the discussed methods within real life applications and be able to assess results.