

ECE 601: State Variables for Engineers

Bryan Van Scoy

December 6, 2024

Contents

I	Fundamentals	6
1	Introduction	7
1.1	Dynamical systems	7
1.2	Example: Spring–mass–damper mechanical system	7
1.3	Course overview	11
2	Systems	13
2.1	Examples	13
2.2	Properties	16
2.3	LTI systems	19
2.4	System response	21
II	Modeling	23
3	State-space models	24
3.1	State	24
3.2	State-space models	25
3.3	Block diagram	27
4	Modeling	31
4.1	Stochastic systems	31
4.2	Electrical circuits	32
4.3	Mechanical systems	38
5	Linearization	44
5.1	Jacobian linearization	44
5.2	Feedback linearization	46
6	Linear Time-Invariant Systems	48
6.1	Representations of LTI systems	48
6.2	Transfer function	48
6.3	Block diagrams and state-space realizations	50
6.4	State transformations	54
III	Analysis	55
7	Canonical Forms	56
7.1	Direct form	56
7.2	Controllable canonical form	57

7.3	Observable canonical form	58
7.4	Kalman canonical form	59
7.5	Diagonal canonical form	59
7.6	Jordan canonical form	59
8	Response	60
8.1	Types of responses	60
8.2	Solving the state equation for continuous-time LTI systems	61
8.3	Solving the state equation for discrete-time LTI systems	63
8.4	Diagonal form	64
8.5	Impulse response	65
8.6	Jordan form	66
9	Controllability	67
9.1	Definition	67
9.2	Examples of uncontrollable systems	68
9.3	Analysis	69
9.4	Controllable subspace	71
9.5	Minimum norm input signal in continuous time	75
9.6	Controllable canonical form	77
9.7	Characterizations of controllability	77
9.8	Stabilizability	78
10	Observability	79
10.1	Definition	79
10.2	Derivation of main result	79
10.3	Unobservable subspace	81
10.4	Reconstructing the initial state	81
10.5	Observable canonical form	81
10.6	Observable decomposition	82
10.7	Characterizations of observability	83
10.8	Detectability	84
11	Minimality	85
11.1	Overview	85
11.2	Proofs	87
12	Stability	89
12.1	Overview	89
12.2	Internal stability	89
12.3	External stability	95
13	Transient Response	101
13.1	Dominant modes	101
13.2	First-order systems	102
13.3	Second-order systems	103
IV	Control	107
14	Static State Feedback	108
14.1	Open-loop vs feedback control	108

14.2	Static state feedback	110
14.3	Stabilizability	114
15	Steady-state tracking	115
15.1	Open-loop control	115
15.2	Integral control	116
15.3	Pole placement with integral control	117
16	Observers	121
16.1	Luenberger observer	121
16.2	Pole placement	122
16.3	Detectability	123
17	The Separation Principle	127
17.1	Interconnected systems	127
17.2	Observer with static state feedback	130
18	Linear–Quadratic Regulator	133
18.1	Motivation	133
18.2	Problem description	135
18.3	Solution via completing the square	135
19	Kalman Filter	138
19.1	Definition	138
19.2	Interpretations	139
19.3	Example	140
19.4	Duality	141
19.5	Computation	141
19.6	Derivations	142
20	Linear–Quadratic–Gaussian Control	144
20.1	Problem statement	144
20.2	Solution	145
20.3	Extensions	145
21	Model Reduction	146
21.1	Balanced realization	147
21.2	Balanced truncation	148
22	System Identification	149
22.1	Problem description	149
22.2	Eigensystem realization algorithm	149
23	Model Predictive Control	152
23.1	Constrained optimal control	152
23.2	Overview of MPC	152
V	Appendices	154
A	Linear Algebra	155

A.1	Vector space	155
A.2	Eigenvalues and eigenvectors	159
A.3	Matrix similarity	162
A.4	Subspace	163
A.5	Diagonalization	165
A.6	Jordan form	167
A.7	Matrix exponential	168
A.8	Quadratic forms	171
B	Series expansion	172
B.1	One-dimensional functions	172
B.2	Multi-dimensional functions	172

Part I

Fundamentals

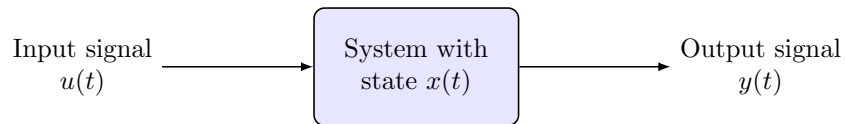
1

Introduction

This is a graduate-level course on dynamical systems that emphasizes linear state-space models in both continuous and discrete time. The material in this course is fundamental knowledge for students pursuing research in systems and control theory, signal processing, or robotics.

1.1 Dynamical systems

A *system* is described by the abstract representation:

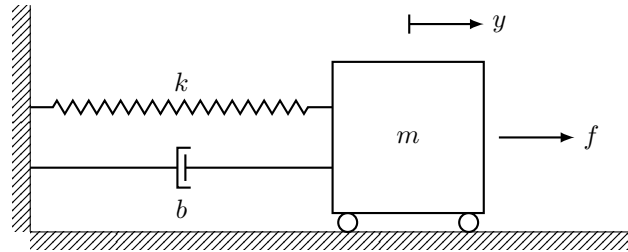


Example (DC motor).

- The input $u(t)$ is the voltage we apply to the motor, which is a function of time.
- The output $y(t)$ is the angular speed of the motor shaft, which we measure using a tachometer.
- The system is the motor itself. The state $x(t)$ is a vector containing all quantities that determine the motor's behavior. This may include the current through the windings, the angular position and speed of the shaft, the temperature, etc.

1.2 Example: Spring–mass–damper mechanical system

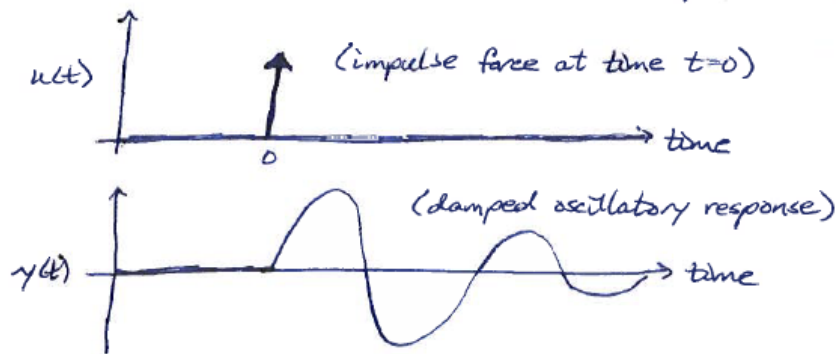
One of the simplest dynamical systems is a mechanical system consisting of a single mass that slides on a fixed horizontal surface (without friction) and is connected to a fixed vertical wall by a spring and a damper.



The *input* to the system is the force that we apply to the mass, and the *output* of the system is the horizontal position of the mass.

For this simple system, we have intuition about how it will behave.

- If we apply an impulsive force to the right, the mass will move to the right, oscillate (due to the spring), the oscillations will decay (due to the damper), and the mass will settle into a fixed position.



- If we apply a constant force to the right, the mass will move to the right, oscillate (due to the spring), the oscillations will decay (due to the damper), and the mass will settle into a fixed position.
- If we apply a constant force to the left, similar behavior will occur with the mass settling into a fixed position to the left of its at-rest position.
- If we apply a sinusoidal force, the mass will continue to oscillate back and forth.

To better understand this relationship between the applied force and the position of the mass, we can use Newton's laws to derive a *mathematical model* of the system.

Modeling

Newton's law states that the mass times the acceleration of the mass is equal to the sum of all the applied forces acting on it. There are several forces acting on the mass:

- The applied force f acts to the right.
- The spring exerts a force $-ky$ that is proportional to the displacement of the mass.
- The damper exerts a force $-by$ that is proportional to the velocity of the mass.

The last two forces are negative since they exert a force to the left for positive (to the right) displacement and velocity. Applying Newton's law gives that $m\ddot{y} = f - ky - b\dot{y}$. Rearranging to put all of the terms involving y on the same side, the equation of motion for the system is

$$m\ddot{y} + b\dot{y} + ky = f$$

This is a second-order differential equation that describes the relationship between the applied force f and the position y of the mass. The coefficients m , b , and k are parameters that describe the system.

Transfer function

One way to analyze this system is to take the *Laplace transform* of the equation of motion. Doing so and assuming the initial conditions $y(0)$ and $\dot{y}(0)$ are zero, we obtain $(ms^2 + bs + k)Y(s) = F(s)$, where $Y(s)$ is the Laplace transform of $y(t)$, and similarly for $F(s)$ and $f(t)$. Rearranging this equation, we can find the *transfer function* of the system,

$$G(s) = \frac{Y(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$$

To find the response $y(t)$ given an applied force $f(t)$, we can compute the Laplace transform $F(s)$, multiply it by the transfer function to obtain $Y(s)$, and then take the inverse Laplace transform to get back to $y(t)$.

PID control

Suppose we want to control the position of the mass. Let the signal r denote the *reference*, which is the desired position of the mass. The *error* signal is then the difference between the reference and the actual position, $e = r - y$. Feedback control constructs the input u to the system (in this case, the force f applied to the mass) based on this error signal. Perhaps the most common feedback controller is *PID control*, where the input u is proportional to the error, its integral, and its derivative. That is,

$$u(t) = \underbrace{k_p e(t)}_{\text{proportional}} + \underbrace{k_I \int_{-\infty}^t e(\tau) d\tau}_{\text{integral}} + \underbrace{k_d \frac{d}{dt} e(t)}_{\text{derivative}}$$

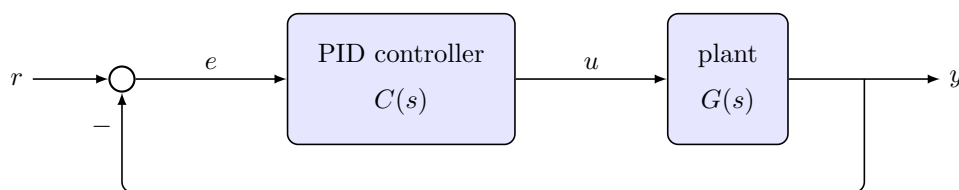
Taking the Laplace transform of this equation, we find that

$$U(s) = \left(k_p + \frac{k_I}{s} + k_d s \right) E(s)$$

where $U(s)$ and $E(s)$ are the Laplace transforms of $u(t)$ and $e(t)$, respectively. The transfer function is that of an ideal PID controller, which is not physically implementable since it is noncausal (the degree of its numerator is greater than that of its denominator). Instead, we can approximate the ideal PID controller using a first-order filter on the derivative term,

$$C(s) = k_p + \frac{k_I}{s} + \frac{k_d s}{T_f s + 1}$$

where $T_f \gg 1$ is the filter time constant. This relationship is illustrated in the following block diagram.



The closed-loop transfer function from the reference r to the output y is

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

Limitations

- Transfer functions describe the relationship between a single input signal and a single output signal (SISO systems). When there are more input or output signals, the transfer function becomes a *transfer matrix*. While this can still be used to study the systems with multiple inputs and/or multiple outputs (MIMO systems), the analysis becomes much more difficult.

$$\underbrace{\begin{bmatrix} Y_1(s) \\ \vdots \\ Y_p(s) \end{bmatrix}}_{p \text{ outputs}} = \underbrace{\begin{bmatrix} G_{11}(s) & \dots & G_{1m}(s) \\ \vdots & \ddots & \vdots \\ G_{p1}(s) & \dots & G_{pm}(s) \end{bmatrix}}_{p \times m \text{ transfer matrix}} \underbrace{\begin{bmatrix} U_1(s) \\ \vdots \\ U_m(s) \end{bmatrix}}_{m \text{ inputs}}$$

- Control design using transfer functions is often graphical and heuristic (such as PID control and root locus), which is not computationally tractable for large systems.
- Even more problematic is that transfer function apply only to *linear time-invariant* systems. Many applications (such as robotics) involve nonlinear systems for which the transfer function does not exist.

State-space model

To overcome these limitations, we will use *state-space models* in this course to study dynamical systems. Let's derive a state-space model for the spring-mass-damper system. To do so, define the *state* of the system as the two-dimensional vector

$$x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$$

so that $x_1 = y$ and $x_2 = \dot{y}$. We can then rewrite the equations of motion in terms of this state vector as follows:

$$\begin{aligned} \dot{x}_1 &= \dot{y} = x_2 \\ \dot{x}_2 &= \ddot{y} = \frac{1}{m}(f - bx_2 - kx_1) \\ y &= x_1 \end{aligned}$$

The first equation comes from the definition of the state, the second equation from the equation of motion, and the third equation describes how the output y depends on the state x . In matrix form, we have the following state-space model for the system:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f \\ y &= [1 \quad 0] x + [0] f \end{aligned}$$

At any time t_0 , the state $x(t_0)$ and the input $u(t)$ for $t \geq t_0$ *uniquely* determine the future outputs $y(t)$ for all $t \geq t_0$. Knowing the initial position $y(0)$ is not enough; we must also know the initial velocity $\dot{y}(0)$ since the system is described by a *second-order* differential equation. This is why the state vector has dimension two.

The general form of a state space model for a linear time-invariant system is

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

For the spring-mass-damper system, the state-space matrices are

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ -\frac{k}{m} & -\frac{b}{m} & \frac{1}{m} \\ \hline 1 & 0 & 0 \end{array} \right]$$

Advantages of state-space models

Some advantages of the state-space representation of a system are that:

- it is a compact and convenient for computer implementations
- it has the same form for systems with multiple inputs and multiple outputs (MIMO) as systems with single inputs and single outputs (SISO)
- the state vector reveals the internal nature of the system, not just the map from its inputs to outputs
- it generalizes to more complex systems, such as those that are nonlinear and time-varying

1.3 Course overview

In this course, you will learn how to:

- model a physical system
- linearize a nonlinear system
- find various state space realizations of a system (minimal, diagonal, controllable, observable)
- find the response of a system
- discretize a continuous-time system
- characterize internal and external stability of a system
- characterize controllability, observability, stabilizability, and detectability of a system
- design feedback controllers to obtain desired transient and steady-state characteristics of the response or to optimize a cost

Main topics

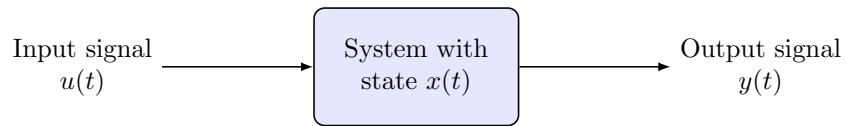
- **Fundamentals:** What is a dynamical system?
 - modeling physical systems (circuits, Newton's laws, Lagrangian and Hamiltonian mechanics)
 - linearization
 - controllability and observability
 - system response
 - realizations (diagonalization, Jordan form)
- **Analysis:** How well does a system perform?
 - stability (BIBO stability, Lyapunov theory)
 - transient response
- **Design and Feedback:** How to modify a system so that it performs well?
 - pole placement
 - observer-based compensation
 - optimal control (LQR)
 - optimal estimation (Kalman filter)
 - separation principle and LQG

- **System Identification:** How to model a system from data?
 - model reduction
 - eigensystem realization algorithm
 - observer Kalman filter detection

2

Systems

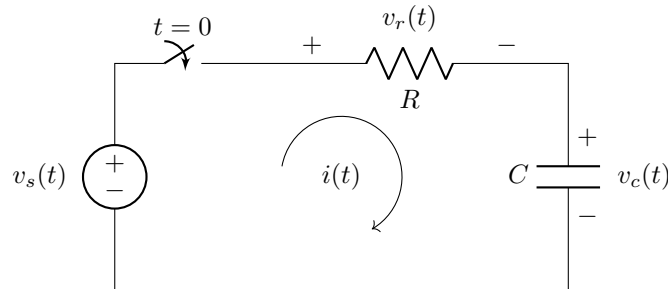
A *system* is a relationship between signals. We typically split the signals into two groups: the *input signals* are those that we can use to influence the behavior of the system, and the *output signals* are those that we can observe or want to control. Systems are often *dynamic*, meaning that the output of the system at any given time depends not only on the current value of the input, but also on previous values of the input. In other words, dynamic systems have *memory*, so the behavior now depends on what has happened to the system in the past. This memory is captured by the *state* of the system, which is a set of signals internal to the system that completely describe its current state of memory. A *continuous-time system* has input, output, and state that are continuous-time signals, while a *discrete-time system* involves discrete signals. The following diagram illustrates the main components of a general system.



2.1 Examples

Circuit

Circuits are electrical systems, where the signals that the system relates to each other are the voltages and currents in the circuit. For instance, consider the following circuit consisting of a voltage source, switch, resistor, and capacitor.



The input signal to this system is the source voltage $v_s(t)$, and the output signal is the voltage across the capacitor $v_c(t)$. Using Kirchhoff's laws, the relationship between the source and capacitor voltage is described

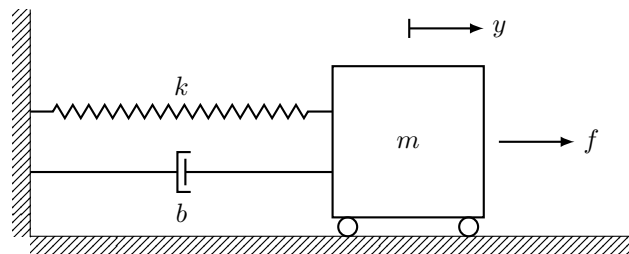
by the following first-order differential equation:

$$v_s(t) = RC \dot{v}_c(t) + v_c(t)$$

The solution to this equation gives the voltage across the capacitor $v_c(t)$, which depends on both the source voltage $v_s(t)$ and the initial capacitor voltage $v_c(0)$. The memory of this system is captured by the voltage across the capacitor, so the state is $x(t) = v_c(t)$. Together, the initial state and the input signal uniquely determine the output.

Mechanical system

Mechanical systems relate signals like the position and velocity of an object with the forces that act on it. For instance, consider the following mechanical system consisting of a mass attached to the wall by a spring and a damper.



The input signal to this system is the external force $f(t)$ applied to the mass, and the output signal is the horizontal position $y(t)$ of the mass. Using Newton's laws, the relationship between the applied force and the position of the mass is described by the following second-order differential equation:

$$f(t) = m\ddot{y}(t) + b\dot{y}(t) + ky(t)$$

Since this is a second-order differential equation, two initial conditions are needed to solve for the position. For instance, we could use the initial position $y(0)$ and the initial velocity $\dot{y}(0)$ as the initial conditions. Since these are both needed to specify the output, the state consists of both the position *and* velocity, that is, $x = (y, \dot{y})$.

Savings account

A savings account can be modeled as a discrete-time system, where the input signal is the amount $A(k)$ deposited into the account in month k and the output signal is the principal (or balance) $P(k)$ of the account. Here, the time index k corresponds to the month and takes nonnegative integer values $k = 0, 1, 2, \dots$, where month zero corresponds to when the account was established. Given that the account earns interest i , the dynamics relating the amount deposited with the principal is

$$\underbrace{P(k+1)}_{\text{balance next month}} = \underbrace{P(k)}_{\text{balance this month}} + \underbrace{\frac{i}{12}P(k)}_{\text{interest}} + \underbrace{A(k)}_{\text{deposits}}$$

While continuous-time systems are described by *differential* equations, discrete-time systems are described by *difference* equations that involve the signals at shifted times (such as k and $k+1$). For this system, the state is the principal $P(k)$.

Numerical algorithms

An iterative numerical algorithm is a sequence of instructions that may be used to solve a mathematical problem. We can interpret such algorithms as discrete-time systems. There are a numerous algorithms that can be applied to solve a variety of problems. We now discuss a few such applications.

Square root calculation

As a simple example, we can use Newton's method to compute the square root \sqrt{a} of a positive number $a > 0$. To do so, let $x(0) > 0$ be the initial estimate of the square root. Then for each iteration $k = 0, 1, 2, \dots$, update the estimate as

$$x(k+1) = \frac{1}{2} \left(x(k) + \frac{a}{x(k)} \right)$$

As this system is iterated, the iterates $x(k)$ converge to the square root \sqrt{a} in the limit as $k \rightarrow \infty$.

Fixed-point iterations

We can use numerical algorithms to solve nonlinear equations using fixed-point iterations. Consider the nonlinear equation

$$2x - e^{-x} = 1$$

To solve this equation, we first isolate one instance of x . For example, solving for the first instance of x gives

$$x = \frac{1}{2}(1 + e^{-x})$$

To turn this into a discrete-time system, we replace x on the left-hand side by $x(k+1)$ and x on the right-hand side by $x(k)$. This gives the discrete-time system

$$x(k+1) = \frac{1}{2}(1 + e^{-x(k)})$$

We can start at some estimate $x(0)$ of the solution and iterate for $k = 0, 1, 2, \dots$. If the iterates of the system converge, then this must be a solution to the original nonlinear equation.

Instead of isolating the first instance of x in the equation, we could have isolated the second instance to obtain the iteration

$$x(k+1) = -\ln(2x(k) - 1)$$

But this iteration does not converge! In general, there is no guarantee that a fixed-point iteration will converge to a solution of the nonlinear equation, so you may need to try isolating a different instance of x , or start the iteration from a different initial condition $x(0)$.

Optimization

Consider the problem of finding the value x that minimizes a function $f(x)$. Some examples of optimization problems are the following:

- x may be the number of widgets that a company produces and $f(x)$ the associated production costs.
- x may be the route that you take to campus and $f(x)$ the amount of time that you spend commuting.

Such optimization problems are typically described mathematically as

$$\underset{x}{\text{minimize}} \quad f(x)$$

If the objective function $f(x)$ is differentiable (meaning that its derivative exists), we can use the gradient descent algorithm to find the minimizer:

$$x(k+1) = x(k) - \alpha \frac{d}{dx} f(x(k))$$

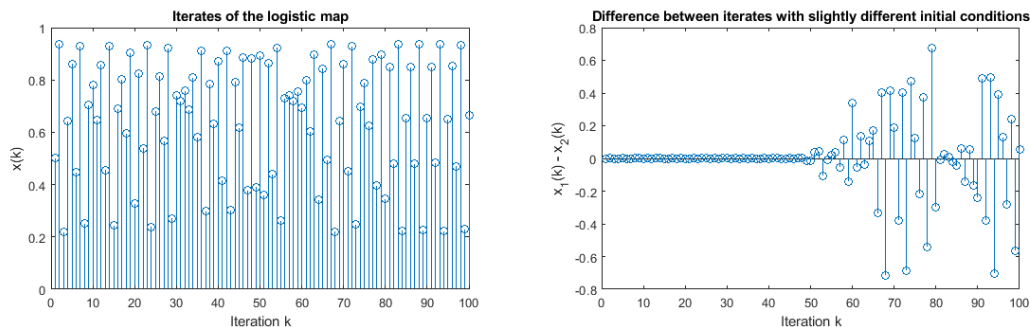
where $\alpha > 0$ is the stepsize. Whether this algorithm converges to the optimal solution or not depends on the properties of the objective function $f(x)$ and the choice of stepsize α .

Logistic map

An interesting discrete-time system is the *logistic map*, which is defined by the iterations

$$x(k+1) = r x(k) (1 - x(k))$$

where $r \in [0, 4]$ is a constant parameter. The dynamics of this system highly depend on the parameter r . For instance, the iterates with $r = 3.75$ and initial condition $x(0) = 0.5$ are shown on the left.



One of the interesting properties of this system is that it is *chaotic*, meaning trajectories starting arbitrarily close together eventually diverge far apart. To illustrate this, suppose we instead initialize the system with $x(0) = 0.500001$. The difference between the iterates using this initialization and the initialization $x(0) = 0.5$ from before is shown on the right. While the trajectories start out close (so that their difference is approximately zero), they eventually become very far apart. This illustrates some of the interesting behavior that can occur when the dynamics are *nonlinear*.

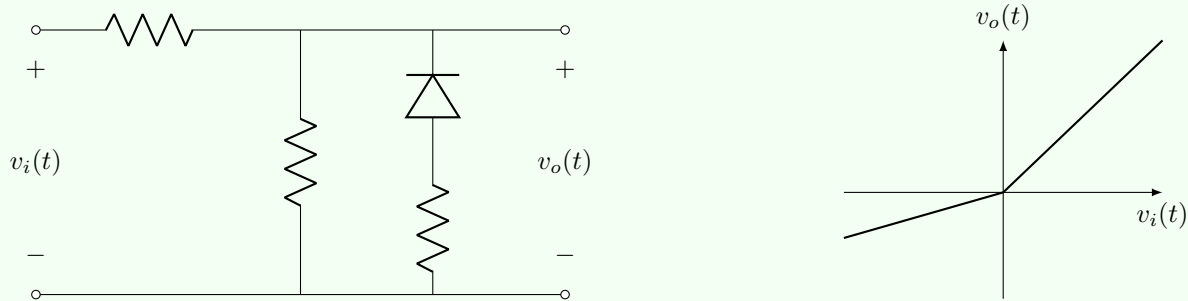
2.2 Properties

It is often useful to characterize systems based on their properties. We now define some of the basic properties that a system may have.

Memory

A system is *static* if the value of the output signal at any given time depends only on the value of the input signal *at the same time*. In other words, $y(t)$ is a function of $u(t)$, but not $u(\tau)$ for any $\tau \neq t$. Static systems have no state.

Example (Static system). An example of a static system is a circuit without any energy storage elements (such as capacitors or inductors). For instance, the output voltage $v_o(t)$ depends only on the input voltage $v_i(t)$ at the current time in the following circuit:



The plot of the output voltage as a function of the input voltage is shown on the right. Since the system has no memory, the output at time t is a function of the input at time t only.

A system is *dynamic* if the value of the output signal depends on the input signal at past (or future) times. In other words, $y(t)$ depends on $u(\tau)$ for some $\tau \neq t$. Dynamic systems have a state, and the response of the system depends not only on the input signal, but on the initial state as well. All systems that have memory are dynamic, so essentially all “interesting” systems are dynamic.

An RC circuit is an example of a dynamic system. The voltage across the capacitor depends on what currents have been applied to it in the past, and the output voltage depends not only on the supply voltage but the initial capacitor voltage as well. Other examples of dynamic systems are the spring–mass–damper system, savings account, fixed-point iterations, and logistic map.

Linearity

A system is *linear* if a weighted sum of input signals produces the same weighted sum of the corresponding output signals. In particular, a system is linear if the input $au_1 + bu_2$ produces the output $ay_1 + by_2$ for all scalars a and b and all input signals u_1 and u_2 and their corresponding output signals y_1 and y_2 .

To show that a system is nonlinear, we only need to find specific input-output pairs $u_1 \mapsto y_1$ and $u_2 \mapsto y_2$ and constants a and b such that the weighted sum of inputs $au_1 + bu_2$ does *not* produce the weighted sum of outputs $ay_1 + by_2$.

Example (Linear vs nonlinear). Examples of linear systems are the RC circuit, savings account, and the system

$$y(t) = u(t) + 3 \int_{-\infty}^t u(\tau) d\tau + 2 \dot{u}(t)$$

Examples of nonlinear systems include the following:

$$y(t) = \frac{1}{2}(u(t) + 3)^2$$

$$y(t) = \int_0^t \sqrt{u(\tau)} d\tau$$

$$\dot{y}(t) = y(t)u(t), \quad y(0) = 1$$

$$y(k+1) = \frac{1}{2} \left[y(k) + \frac{u(k)}{y(k)} \right]$$

$$y(k+1) = [y(k)]^2 + u(k)$$

The previous example suggests a general principle that we may use to quickly detect nonlinearity: if any of the input or output signals are raised to a power, multiplied, or divided, the system is nonlinear.

Time invariance

A system is *time invariant* if shifting the input signal in time shifts the output signal by the same amount in time. Suppose the input signal $u_1(t)$ produces the output signal $y_1(t)$. Then a system is time invariant if the shifted input signal $u_2(t) = u_1(t - \tau)$ produces the shifted output signal $y_2(t) = y_1(t - \tau)$ for all scalars τ and all input signals u_1 and its corresponding output signal y_1 .

A system is *time varying* if shifting the input signal in time does *not* shift the output signal by the same amount in time. In other words, the dynamics of the system change over time.

Example (Time-varying parameter). Time-varying systems often arise from systems whose parameters vary with time. For instance, the RC circuit is time invariant if the resistance and capacitance are constant. But if the resistance $R(t)$ of the resistor changes over time, then the dynamics become time varying:

$$v_s(t) = R(t) C \dot{v}_c(t) + v_c(t)$$

Likewise, the savings account system is time invariant if the interest rate parameter is constant, but it is time varying if the interest rate changes over time (which it does!):

$$P(k+1) = \left(1 + \frac{i(k)}{12}\right) P(k) + A(k)$$

Causality

A system is *causal* if the output at any given time depends only on the input at previous times. In particular, a system is causal if $y(t)$ only depends on the input $u(\tau)$ at past times $\tau \leq t$ and *not* future times $\tau > t$.

Example. All physical systems are causal, as a physical signal cannot predict the values of future input signals. However, a noncausal system can be implemented in several circumstances:

- We can implement a noncausal system if the independent variable does not actually represent time. In image processing, for example, signals are images and the independent variables correspond to dimensions in space along the image.
- Even when the independent variable does represent time, we can implement a noncausal system when the entire input signal is available before processing. For example, consider processing an audio signal that is stored on a computer. Since the entire signal is available, the algorithm need not process the audio signal in a causal manner.

Dimensionality

A system is *finite-dimensional* if the statespace (the space in which the state takes its values) is a finite-dimensional vector space such as \mathbb{R}^n . All of the systems that we will study are finite dimensional.

Example (Continuous-time delay). A relevant example of an infinite-dimensional (linear time-invariant) system is a delay in continuous time. For a delay of time T , the output is $y(t) = u(t - T)$. The state of this system is the set of all inputs over the past T seconds,

$$x(t) = \{u(\tau) \mid t - T \leq \tau \leq t\}$$

This is an infinite-dimensional space, since it is a function over a continuous interval.

- **Time.** A system evolves in *continuous time* if time t takes values in a continuous interval, such as all real numbers $(-\infty, \infty)$ or nonnegative real numbers $[0, \infty)$. In contrast, a system evolves in *discrete time* if time t takes values in a discrete set, such as all integers $\{\dots, -2, -1, 0, 1, 2, \dots\}$ or natural numbers $\{0, 1, 2, \dots\}$.
- **Linearity.** A system is *linear* if a weighted sum of input signals produces the same weighted sum of the corresponding output signals. In particular, a system is linear if the input $a u_1 + b u_2$ produces the output $a y_1 + b y_2$ for all scalars a and b and all input signals u_1 and u_2 and their corresponding output signals y_1 and y_2 .
- **Time invariance.** A system is *time invariant* if shifting the input signal in time shifts the output signal by the same amount in time. In particular, a system is time invariant if the shifted input signal $t \mapsto u(t - \tau)$ produces the shifted output signal $t \mapsto y(t - \tau)$ for all scalars τ and all input signals u and its corresponding output signal y .
- **Causality.** A system is *causal* if at each time the output at that time only depends on the input at previous times. In particular, a system is causal if $y(t)$ only depends on the $u(\tau)$ for $\tau \leq t$. All physical systems are causal.
- **Dimensionality.** A system is *finite-dimensional* if the state $x(t)$ at each time t is in a finite-dimensional vector space such as \mathbb{R}^n .

2.3 LTI systems

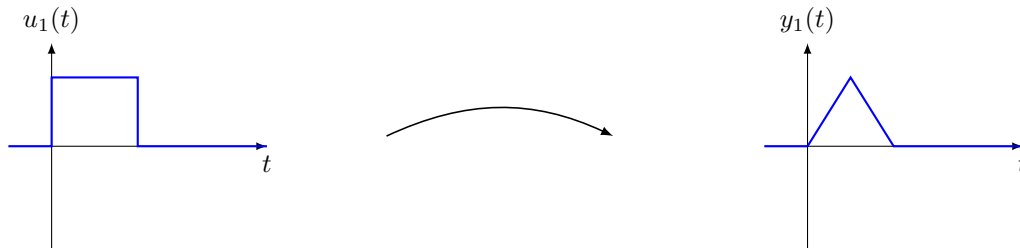
An important class of systems are those that are *linear* and *time invariant*, or LTI systems. There are several reasons to study LTI systems:

- Some systems are inherently linear and time invariant, such as RLC circuits, spring–mass–damper systems, and many mathematical operations like differentiation and integration.
- Many systems can be adequately approximated by LTI systems. Even if a system is not LTI, we can often approximate it by one that is.
- Understanding LTI systems forms a foundational basis for understanding more complex systems.
- LTI systems have a rich and elegant theory. They strike a balance between being simple enough to thoroughly understand and yet are rich enough to be broadly applicable.

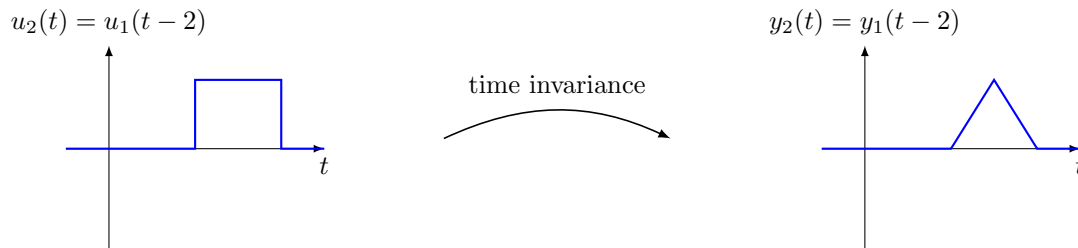
LTI systems are a rich class of systems for which we have a complete and thorough understanding of how they work, and the tools that we learn will help us in understanding more complicated systems. LTI systems are the types of systems that you study in introductory courses in circuits, mechanics, and mathematics.

Motivation

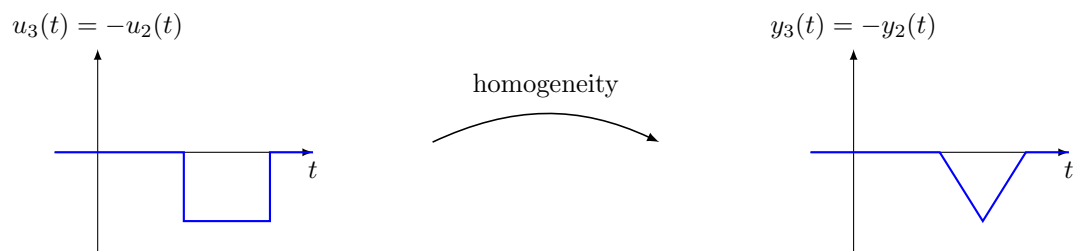
Consider an LTI system, and suppose we know a single input/output pair for the system as shown below.



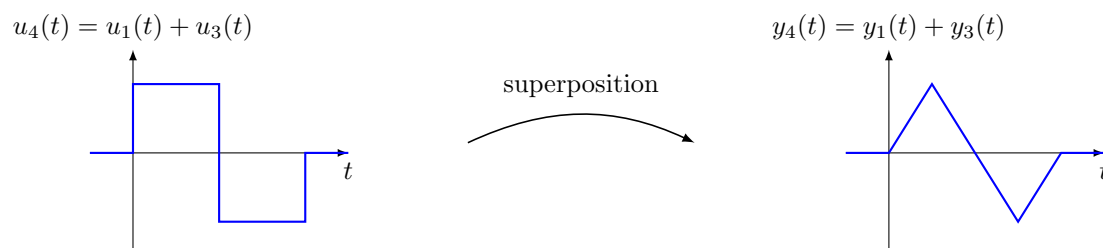
Because the system is time invariant, we can shift the input in time and the output gets shifted in time by the same amount.



Since the system is linear (and therefore also homogeneous), we can scale this shifted input and the output gets scaled by the same amount.



Also from linearity (this time using superposition), we can sum the first and third inputs and the output will be the sum of the corresponding outputs.



We could continue using the linearity and time invariance properties to construct new input/output pairs, all from knowing a single input/output pair and that the system is LTI!

Generalization

Let's now generalize the motivating example to see all the ways in which we can exploit that a system is linear and time invariant. As before, suppose we know a single input-output pair of an LTI system (this time we will work with a discrete-time system for simplicity, although the procedure is similar in continuous time).

$$u(k) \mapsto y(k)$$

Since the system is time invariant, then shifting the input signal in time shifts the output signal by the same amount in time.

$$u(k - m) \mapsto y(k - m)$$

This produces a different input-output pair for each time shift (corresponding to the value of m). Now from linearity, taking a weighted sum of inputs produces an identical weighted sum of outputs. Applying this to the above input-output pair for all values of m gives

$$\sum_m w(m) u(k - m) \mapsto \sum_m w(m) y(k - m)$$

where there is a weight $w(m)$ for each value of m .

Later, we will see that this expression is general enough to represent *any* input-output pair of the system. In other words, knowing a single input-output pair is enough to completely characterize any LTI system!

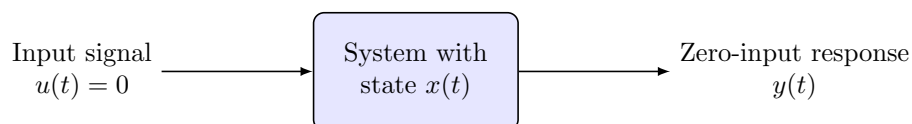
2.4 System response

The most fundamental question regarding the analysis of a system is: *how does the system respond to various excitations?* The output of a system, known as the *response*, depends on both the input signal and the initial conditions (or initial state). In an RC circuit, for instance, the voltage across the capacitor depends on both the source voltage and the initial capacitor voltage.

There are various types of responses, based on the nature of the input signal and initial condition. We now describe several important responses of a system.

Zero-input response

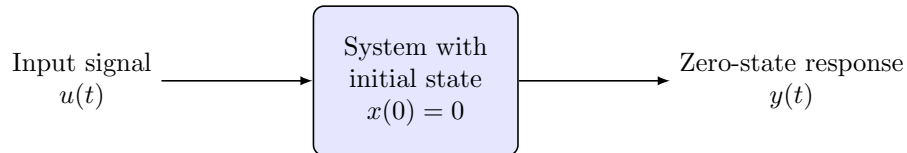
The zero-input response (ZIR) is the output of the system due to the initial conditions when the input signal is zero (at all times). The zero-input response depends only on the initial state of the system.



Example (ZIR of spring–mass–damper system). Going back to our example of a spring–mass–damper mechanical system, the zero-input response is the position of the mass when no force is applied. In this case, the response depends on only the initial position and velocity of the mass.

Zero-state response

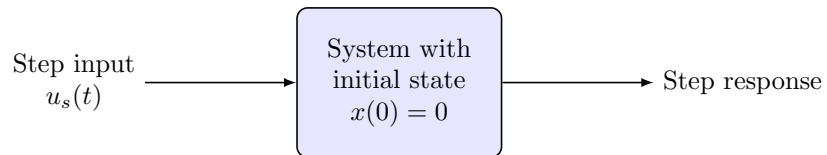
The *zero-state response* is the output of the system due to the input signal when the initial state is zero. By “initial state”, we typically mean the state at time zero, in which case it is assumed that the input signal is zero for times before the initial time.



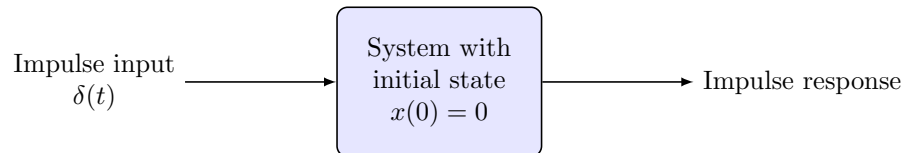
Example (ZSR of spring–mass–damper system). Continuing our example, the zero-state response of the mechanical system is the position of the mass when it starts at rest. In this case, the response depends on only the force applied to the mass.

The zero-state response depends on the particular input signal. Two common zero-state responses are the following.

- The **step response** is the zero-state response due to a unit step input signal.



- The **impulse response** is the zero-state response due to a unit impulse input signal.



Part II

Modeling

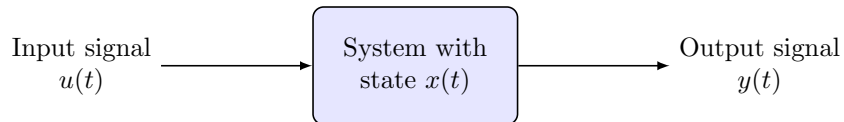
3

State-space models

Dynamical systems have various representations, from schematic diagrams to differential equations and transfer functions. In this chapter, we will study models that explicitly describe the state of the system, which is a signal internal to the system that contains all information about the system at each point in time. State-space representations are useful since they can represent any system (even nonlinear ones), while concepts such as the transfer function and convolution are limited to linear time-invariant systems. Additionally, there exist a vast amount of analysis and design methods for state-space models.

3.1 State

Recall that a *system* is described by the abstract representation:



The symbol t denotes *time*, and the input u , output y , and state x are all *signals* that are functions of time. The defining property of the state is as follows.

Definition (State). The state is a signal such that, at any time t_0 , the state $x(t_0)$ and the input $u(t)$ for $t \geq t_0$ *uniquely* determine the future outputs $y(t)$ for $t \geq t_0$.

The state contains all relevant information about the system at each point in time so that, along with the input signal, it can be used to determine the output. Some examples of systems and their associated states are the following:

- In a mechanical system, the state may consist of the positions and velocities of all masses.
- In a circuit, the state may consist of the voltages across all capacitors and current through all inductors.

An important properties of the state (as we will see) is that the state is *not* unique. Also, the true state of a system may include many quantities such as the ambient temperature, pressure, humidity, time of day, season, etc, as these may affect the dynamics of the system (possibly to a very small extent). Modeling a system involves a trade-off between the complexity of the model and how accurately it represents the desired system. In the rest of this chapter, we will study system models that explicitly represent the state.

3.2 State-space models

General model

State-space models explicitly describe how the state $x(t)$ changes in time, which depends on the current value of the state, the system input, and possibly time itself. For causal continuous-time and discrete-time systems, respectively, the *state equation* that describes how the state evolves is given by

$$\dot{x}(t) = f(x(t), u(t), t) \quad \text{or} \quad x(t+1) = f(x(t), u(t), t). \quad (\text{state equation})$$

The state equation is a differential equation in continuous time and a difference equation in discrete time. The function f is called the *state transition function* and describes how the state changes. Since the state completely describes the system at each time, the output is a function of the current state (and possibly the current input and time as well),

$$y(t) = g(x(t), u(t), t). \quad (\text{output equation})$$

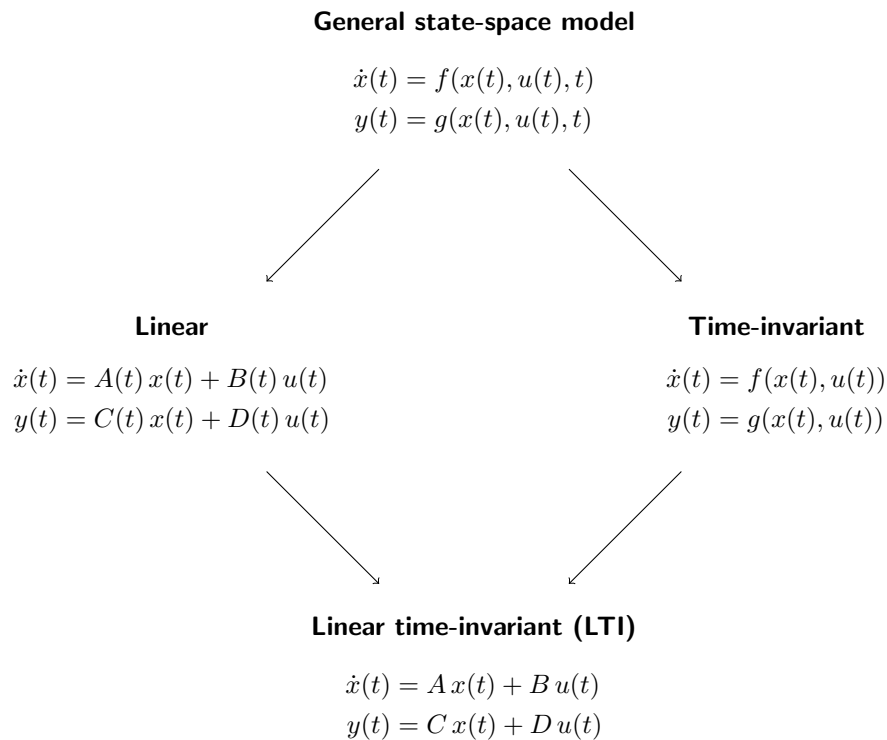
This is called the *output equation* and is the same in both continuous and discrete time.

For an n -dimensional system with m inputs and p outputs, the dimensions of the signals are as follows:

- $x(t)$ is an n -dimensional state vector
- $u(t)$ is an m -dimensional input vector
- $y(t)$ is a p -dimensional output vector

Simplified models based on system properties

The above state-space model can be used to represent a wide variety of dynamical systems. When the system has certain properties, however, we can represent the system using a simpler model that is easier to work with. For instance, the state transition function f and output function g are both linear when the system is linear, and they are independent of time when the system is time invariant. The general state-space model of a causal continuous-time system is as follows (discrete-time systems are similar, except that the time derivative $\dot{x}(t)$ is replaced with the time shift $x(t+1)$):



For finite-dimensional systems, the state $x(t)$ is in a finite-dimensional vector space, so A , B , C , and D are matrices (in the infinite dimensional setting, they are general linear operators).

Autonomous systems

An *autonomous* system is a system without input. For instance, a general autonomous system in continuous time has the form

$$\dot{x}(t) = f(x(t), t)$$

while an autonomous linear time-invariant system has the form

$$\dot{x}(t) = Ax(t).$$

Linear systems

While many systems are nonlinear, we will focus on linear models. There are several reasons for this:

- Linear models are often good approximations to nonlinear systems (within certain operating conditions).
- State-space analysis for linear systems is computationally tractable. We can leverage fast numerical linear algebra tools that scale to large systems and can be done rapidly in real time.
- State-space analysis has become a cornerstone of modern engineering since the 1980's.
- Many of the core concepts from linear systems occur in nonlinear systems as well. We must understand the (simpler) linear case to build intuition for other more complex systems.

Since LTI systems are completely described by their system matrices (A, B, C, D) , we sometimes use the following compact notation to represent the system:

$$y = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] u.$$

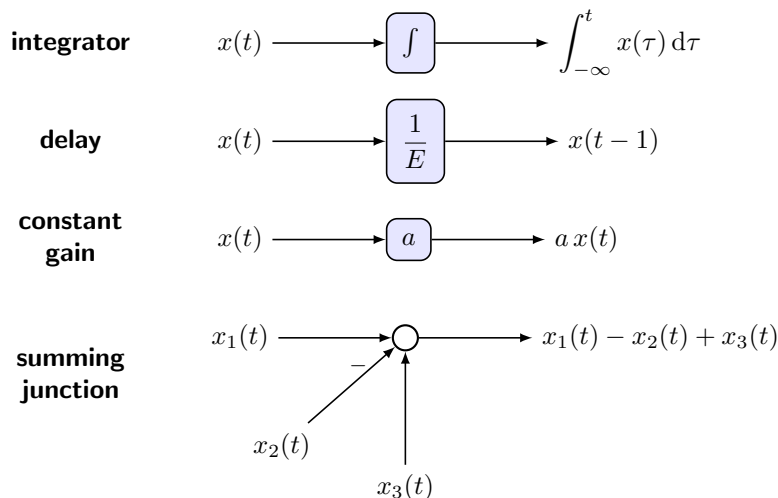
The matrices (A, B, C, D) are the state-space matrices that represent the dynamics of the system.

- A is an $n \times n$ matrix that represents how the current state affects the state evolution
- B is an $n \times m$ matrix that represents how the input affects the state evolution
- C is a $p \times n$ matrix that represents how the state affects the output
- D is a $p \times m$ matrix that represents how the input affects the output

3.3 Block diagram

The block diagram is a visual representation of a system. This representation is useful for identifying structure in the system and how the various signals interact with each other. In this section, we describe block diagram representations of LTI systems.

In the block diagram representation of a system, arrows represent signals, blocks represent systems, and circles represent summing junctions. Any LTI system can be described by a block diagram with the following types of components.

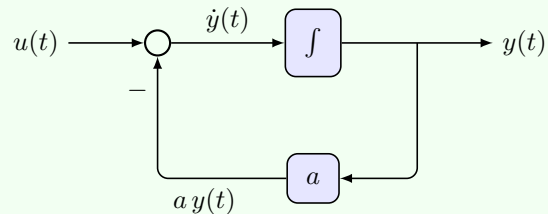


A few comments concerning the basic block diagram components:

- By default, we assume that all signals entering a summing junction (represented by a circle) are summed. To subtract a signal, we place a negative sign on the arrow just before the summing junction.
- Integrators are used for continuous-time systems, while delays are used for discrete-time systems.
- Using the fundamental theorem of calculus, another way to interpret an integrator is that, if the input signal is $\dot{x}(t)$, then the output signal is $x(t)$. Similarly, another way to interpret a delay is that, if the input signal is $x(t+1)$, then the output signal is $x(t)$.

- We can also represent block diagrams in the frequency domain. Constant gains and summing junctions are identical in the frequency domain. The Laplace transform of an integrator is multiplication by $1/s$, and the z -transform of a delay is multiplication by $1/z$.

Example. The first-order differential equation $\dot{y}(t) + a y(t) = u(t)$ can be represented by the following block diagram, where we labeled the intermediate signals for clarity. Note that the summing junction in the block diagram represents the differential equation.

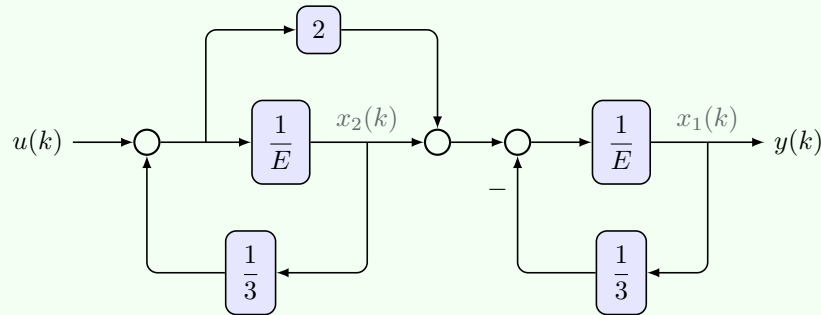


State-space realization from block diagram

We now show how to construct a state-space realization of a system from its block diagram representation. The steps are as follows:

- Label the output of each integrator block (in continuous time) or delay block (in discrete time) as a state variable $x_i(t)$. Then the input to the block is $\dot{x}_i(t)$ for an integrator and $x_i(t+1)$ for a delay.
- Express the input to each integrator or delay block in terms of the state variables $x_1(t), x_2(t), \dots, x_n(t)$ and/or the input signals $u_1(t), u_2(t), \dots, u_m(t)$. This is the state equation.
- Express the outputs $y_1(t), y_2(t), \dots, y_p(t)$ in terms of the state variables and/or the input signals. This is the output equation.

Example. Consider the discrete-time LTI system described by the following block diagram.



We labeled the output of each delay block as a stored variable, in this case $x_1(k)$ and $x_2(k)$. Now let's write down the difference equations that these intermediate variables satisfy. The first summing junction on the left side of the diagram corresponds to the equation

$$x_2(k+1) = \frac{1}{3}x_2(k) + u(k)$$

and the third summing junction on the right side of the diagram corresponds to the equation

$$x_1(k+1) = -\frac{1}{3}x_1(k) + x_2(k) + 2x_2(k+1)$$

This equation contains $x_2(k+1)$, but we already found how to express this in terms of $x_2(k)$ and $u(k)$. Substituting in the above expression,

$$x_1(k+1) = -\frac{1}{3}x_1(k) + \frac{5}{3}x_2(k) + 2u(k)$$

We now have two first-order difference equations in the variables $x_1(k)$ and $x_2(k)$, and the input signal $u(k)$ enters directly into these equations. The last equation that we need is how to find the output signal $y(k)$ from these intermediate variables, which in this case is simply

$$y(k) = x_1(k)$$

Therefore, the state-space representation corresponding to the block diagram is

$$y = \left[\begin{array}{cc|c} -\frac{1}{3} & \frac{5}{3} & 2 \\ 0 & \frac{1}{3} & 1 \\ 1 & 0 & 0 \end{array} \right] u.$$

Block diagram from state-space realization

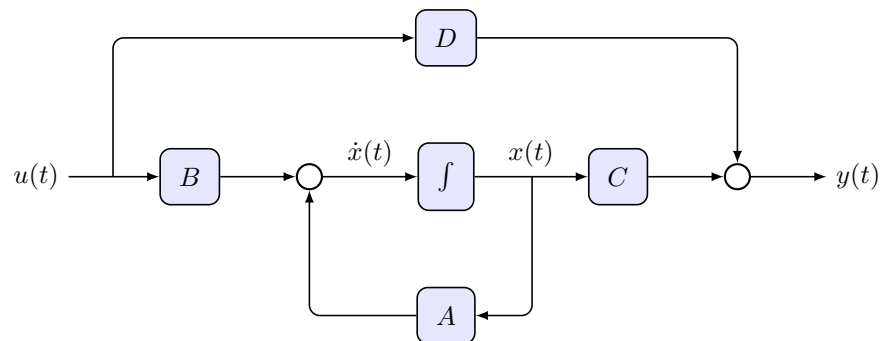
The above procedure could also be reversed. Given a state-space representation of a system, we can construct a block diagram as follows:

- Let each state variable $x_i(t)$ be the output of an integrator/delay. Then the input to the block is $\dot{x}_i(t)$ for an integrator and $x_i(t+1)$ for a delay.
- Use the state variables $x_1(t), \dots, x_n(t)$ and/or the input signals $u_1(t), \dots, u_m(t)$ to construct the input to each integrator/delay based on the state equation.
- Use the state variables $x_1(t), \dots, x_n(t)$ and/or the input signals $u_1(t), \dots, u_m(t)$ to construct the output

signals $y_1(t), \dots, y_p(t)$ based on the output equation.

General form

If we allow arrows in the block diagram to represent vector-valued signals, then we can also use the above procedure to construct the following block diagram of a general state-space realization, where the integrator block integrates each component of its input signal.



Based on the procedures for converting between state-space representations and block diagrams, we make the following observations:

- The number of state variables is the same as the number of integrator/delay blocks.
- There is a one-to-one correspondence between block diagrams and state-space representations.

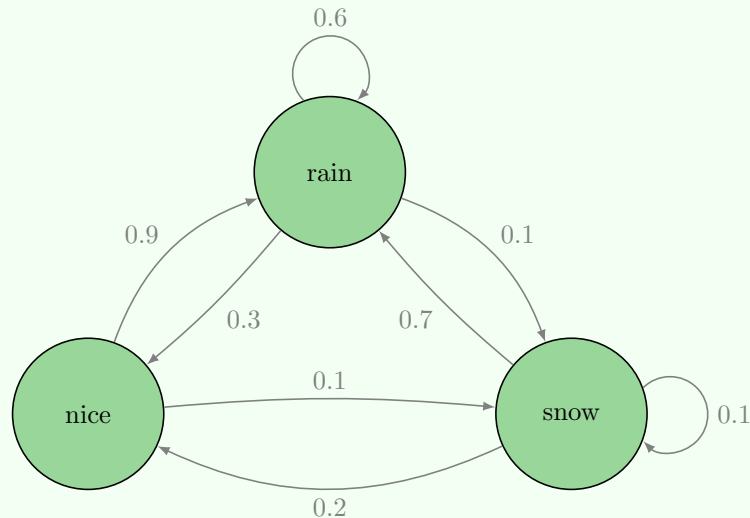
4

Modeling

Dynamical systems are described by differential equations in continuous time and difference equations in discrete time. These equations that describe the dynamics of the system are often derived from physical laws. In this chapter, we will derive models for electrical circuits and mechanical systems from first principles.

4.1 Stochastic systems

Example (London in the winter). Let the discrete time t denote the day, and suppose the probability that the weather transitions from one state (nice, rainy, or snowy) on a given day to another state the next day is as follows:



For instance, if the weather today is nice, then tomorrow the weather will rain with probability 0.9 and snow with probability 0.1 (with no probability of being nice again!). We can model the weather as a discrete-time autonomous linear time-invariant system. Define the state on day t as

$$x(t) = \begin{bmatrix} \text{Prob}(\text{nice on day } t) \\ \text{Prob}(\text{rain on day } t) \\ \text{Prob}(\text{snow on day } t) \end{bmatrix}$$

From the law of total probability, the state evolves as follows:

$$x(t+1) = \begin{bmatrix} 0 & 0.3 & 0.2 \\ 0.9 & 0.6 & 0.7 \\ 0.1 & 0.1 & 0.1 \end{bmatrix} x(t)$$

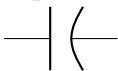

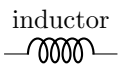
This is an autonomous discrete-time linear time-invariant dynamical system that describes the probability that the weather is nice, rainy, and snowy on any given day. Since the components of the state are probabilities, they must be nonnegative and sum to one. For instance, if the weather is nice today, then we could set $x(0) = (1, 0, 0)$ and iterate the system to observe how the probabilities change over time. This type of system that describes the sequence of probabilities over time is called a *Markov chain*.

4.2 Electrical circuits

Circuits are systems composed of electrical components. The signals in a circuit include the voltage of each node and the current through each path.

Circuit elements

Circuit elements include resistors, capacitors, inductors, op-amps, and other integrated circuits (ICs). The following table summarizes the relationship between the voltage $v(t)$ across a circuit component and the current $i(t)$ through the component as a function of time t for resistors, capacitors, and inductors.

component	voltage-current	current-voltage
capacitor 	$v(t) = \frac{1}{C} \int_{-\infty}^t i(\tau) d\tau$	$i(t) = C \frac{dv(t)}{dt}$
resistor 	$v(t) = R i(t)$	$i(t) = \frac{1}{R} v(t)$
inductor 	$v(t) = L \frac{di(t)}{dt}$	$i(t) = \frac{1}{L} \int_{-\infty}^t v(\tau) d\tau$

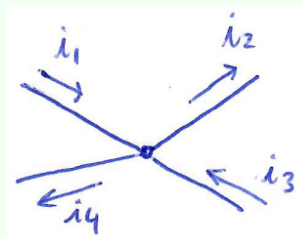
We can interpret each individual circuit element as a (simple) dynamical system that relates the voltage across the component with the current through the element. With this interpretation, a resistor is a *static* system in that the voltage and current are functions of each other at each point in time; that is, resistors have no memory. Capacitors and inductors, on the other hand, are *dynamical* systems that store energy in electric and magnetic fields.

Governing laws

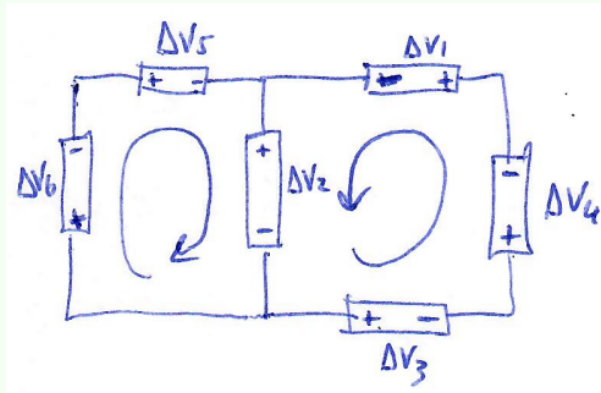
Just as Newton's laws govern mechanical systems, Kirchoff's laws describe the relationship between voltages and currents in an electrical circuit. Kirchoff's laws can be derived from Maxwell's equations in the low-frequency limit.

- **Kirchoff's Current Law (KCL):** The sum of currents at any node is zero.
- **Kirchoff's Voltage Law (KVL):** The sum of voltages around any loop is zero.

Example (KCL). The following node has four paths with two currents (i_1 and i_3) flowing into the node and two currents (i_2 and i_4) flowing out of the node. Kirchoff's current law states that $i_1 + i_3 = i_2 + i_4$.



Example (KVL). Consider the following circuit with various loops.



We can apply Kirchoff's voltage law to both of the inner loops as well as the outer loop. Doing so yields the following equations:

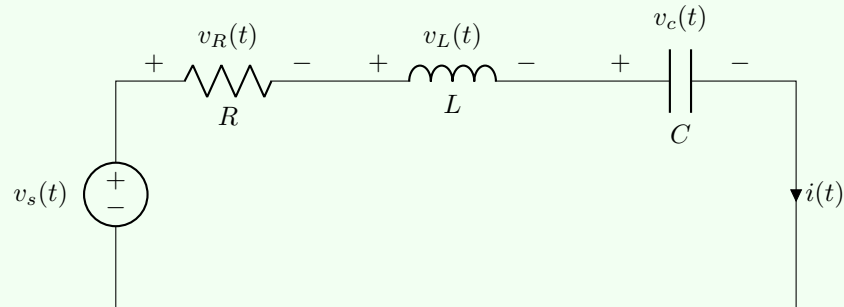
$$0 = \Delta V_2 + \Delta V_5 + \Delta V_6 \quad (\text{left inner loop})$$

$$0 = \Delta V_1 + \Delta V_2 + \Delta V_3 + \Delta V_4 \quad (\text{right inner loop})$$

$$0 = \Delta V_6 + \Delta V_5 - \Delta V_1 - \Delta V_4 - \Delta V_3 \quad (\text{outer loop})$$

Remark. For KVL, we use the convention that the voltage is positive when traversing the loop from the positive terminal (+) to the negative terminal (-). If we would instead traverse the loop in the opposite direction, all of the signs would be flipped which would yield the same governing equation. Also, only need to count loops that are independent from each other. In the above example, for instance, the equation for the outer loop is redundant and follows from summing the inner loop equations. ■

Example (Series RLC circuit). We now use Kirchoff's laws to model the following series RLC circuit.



The sum of the voltages around the loop is zero, so the source voltage is equal to the sum of the voltage drops across each of the three elements.

$$v_s = v_R + v_L + v_c$$

The same current flows through each element. Using the relationships between voltage and current for each element, we have that

$$i = \frac{1}{R} v_R = C \dot{v}_c \quad \text{and} \quad v_L = L \dot{i}$$

Combining these, we can write the source voltage in terms of the voltage across the capacitor and its derivatives as

$$v_s = LC \ddot{v}_c + RC \dot{v}_c + v_c$$

This is a second-order differential equation that describes the relationship between the supplied voltage and the voltage across the capacitor.

There are various ways we can write the governing equation. For instance, we can use that $i = C \dot{v}_c$ to write the equation in terms of the current as

$$v_s = L \dot{i} + Ri + \frac{1}{C} \int i \, dt$$

Current is the rate of change of charge ($i = \dot{q}$), so the governing equation in terms of the charge is

$$v_s = L \ddot{q} + R \dot{q} + \frac{1}{C} q$$

State-space representations

To find a state-space representation of an electrical circuit,

- Define the state as the vector of voltages across capacitors and currents through inductors.
- Express the currents through capacitors and voltages across inductors in terms of the state and inputs. These are the state equations.
- Express each output in terms of the state and inputs. These are the output equations.

Example (continued). To find a state-space representation of the RLC circuit, we first let the state vector be the capacitor voltage and inductor current,

$$x(t) = \begin{bmatrix} v_c(t) \\ i(t) \end{bmatrix}.$$

We then find expressions for the current through the capacitor and voltage across the inductor in terms of the state and input. In this case, the current through the capacitor is $i(t)$. To find the voltage across the inductor, we use KVL,

$$v_s(t) = v_R(t) + v_L(t) + v_c(t)$$

and the relationship between voltage and current for a resistor,

$$v_R(t) = Ri(t).$$

The state equations then consist of the derivatives of state, which are

$$\frac{dv_c(t)}{dt} = \frac{1}{C}i(t)$$

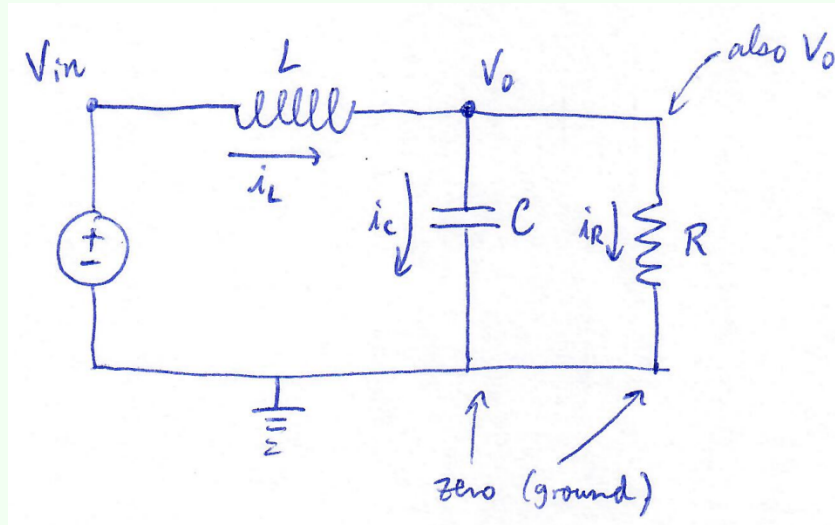
and

$$\frac{di(t)}{dt} = \frac{1}{L}v_L(t) = \frac{1}{L}(v_s(t) - Ri(t) - v_c(t)).$$

Therefore, a state equation that represents the circuit is

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{1}{L} & -\frac{R}{L} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} v_s(t).$$

Example (Parallel RLC circuit). We now use Kirchoff's laws to model the following parallel RLC circuit.



From the relationships between voltage and current for each circuit element and Kirchoff's laws, we obtain the following equations:

$$V_{in} - V_0 = L \dot{i}_L \quad (\text{inductor})$$

$$V_0 = \frac{1}{C} \int i_C dt \quad (\text{capacitor})$$

$$V_0 = R i_R \quad (\text{resistor})$$

$$i_L = i_C + i_R \quad (\text{KCL at } V_0)$$

To find a state-space representation of the circuit, we let the state consist of the voltage across the capacitor and the current through the inductor,

$$x(t) = \begin{bmatrix} V_0(t) \\ i_L(t) \end{bmatrix}.$$

The derivative of the capacitor voltage is then

$$\frac{dV_0(t)}{dt} = \frac{1}{C} i_C(t) = \frac{1}{C} \left(i_L(t) - \frac{1}{R} V_0(t) \right)$$

and the derivative of the inductor current is

$$\frac{di_L(t)}{dt} = \frac{1}{L} (V_{in}(t) - V_0(t)).$$

Therefore, a state equation that represents the circuit is

$$\dot{x}(t) = \begin{bmatrix} -\frac{1}{RC} & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V_{in}(t).$$

Remark (Analogy between mechanical and electrical systems). Recall that the governing equation of a spring-mass-damper system and the series RLC circuit in terms of the charge are

$$m\ddot{y} + b\dot{y} + ky = f \quad \text{and} \quad L\ddot{q} + R\dot{q} + \frac{1}{C}q = v_s.$$

By comparing these equations, we can interpret an electrical circuit as a mechanical system and vice versa. For instance, the supply voltage in the circuit is similar to the applied force in the mechanical system, the charge in the circuit is similar to the position of the mass, and the coefficients of the electrical components correspond to those of the mechanical components. The full analogy is as follows:

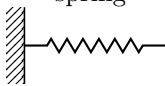
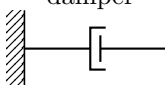
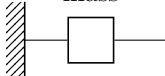
Mechanical system	Electrical circuit
applied force f	supply voltage v_s
position y	charge q
mass m	inductance L
damping coefficient b	resistance R
spring coefficient k	(inverse) capacitance $\frac{1}{C}$

We can construct a similar analogy for the parallel RLC circuit. This analogy enables us to interpret electrical circuits as mechanical systems and vice versa. ■

4.3 Mechanical systems

The equations of motion for mechanical systems can be derived using Newton's law of motion as well as variational formulations based on the energies in the system.

Elements

component	force-velocity	velocity-force
 spring	$f(t) = k \int_{-\infty}^t v(\tau) d\tau$	$v(t) = \frac{1}{k} \frac{df(t)}{dt}$
 damper	$f(t) = b v(t)$	$v(t) = \frac{1}{b} f(t)$
 mass	$f(t) = m \frac{dv(t)}{dt}$	$v(t) = \frac{1}{m} \int_{-\infty}^t f(\tau) d\tau$

Newtonian mechanics

Newton's law of motion states that the sum of the external forces applied to a mass is equal to the time derivative of momentum,

$$\frac{dp}{dt} = \sum_i f_i$$

Both the external forces and the momentum are vectors, so they have both a magnitude and a direction. For a particle, the momentum is the product of the mass and the velocity, $p = mv$. When the mass is constant, this leads to the familiar form of Newton's law which states that the sum of the external forces is equal to

the mass times the acceleration,

$$ma = \sum_i f_i$$

where the acceleration is the time derivative of velocity, $a = \frac{dv}{dt}$.

While Newton's law accurately describes the motion of a system, it can be difficult to apply to complex systems. There are several reasons for this.

- Newton's law involves *vector* quantities. These vectors must be described in an orthogonal set of coordinates, which is not always the most natural coordinates to describe a system.
- All external forces applied to an object must be solved for. When systems involve constraints (such as a mass pushing against a wall, or a mass hanging from a string), we must explicitly solve for the forces that result in this constrained motion (such as the wall or string pushing back against the mass).

An alternative formulation of these equations of motion, collectively known as *variational mechanics*, overcomes these limitations by allowing us to define a set of independent but not necessarily orthogonal *generalized coordinates*.

Variational mechanics

Variational mechanics formulates the equations of motion using the (scalar) energies in the system.

- The positions and rotations of each particle or planar rigid body is described by a set of *generalized coordinates* q_i for $i = 1, \dots, n$.
- A set of *generalized forces* Q_k act on the system. Examples of generalized forces are translational forces and rotational torques.
- The *kinetic energy* of the system is denoted T .

- For a particle with mass m and velocity v , the kinetic energy is

$$T = \frac{1}{2}mv^2$$

- For an object rotating with angular velocity ω with moment of inertia J , the kinetic energy is

$$T = \frac{1}{2}J\omega^2$$

- The *potential energy* of the system is denoted U .
- For a particle with mass m and height y with gravitational acceleration g , the potential energy is

$$U = mgy$$

- For a spring with coefficient k and displacement Δ , the potential energy is

$$U = \frac{1}{2}k\Delta^2$$

- The *Lagrangian* is the difference between the kinetic and potential energy of the system,

$$L = T - U$$

In general, the Lagrangian depends on the generalized coordinates q_i , the corresponding generalized velocities \dot{q}_i , and time t .

- The *generalized momentum* associated with the generalized coordinate q is the partial derivative of the Lagrangian with respect to the time derivative of the generalized coordinate,

$$p = \frac{\partial L}{\partial \dot{q}}$$

- The *Hamiltonian* is defined in terms of the Lagrangian as

$$H(q, p, t) = \sum_{i=1}^n p_i \dot{q}_i - L(q, \dot{q}, t)$$

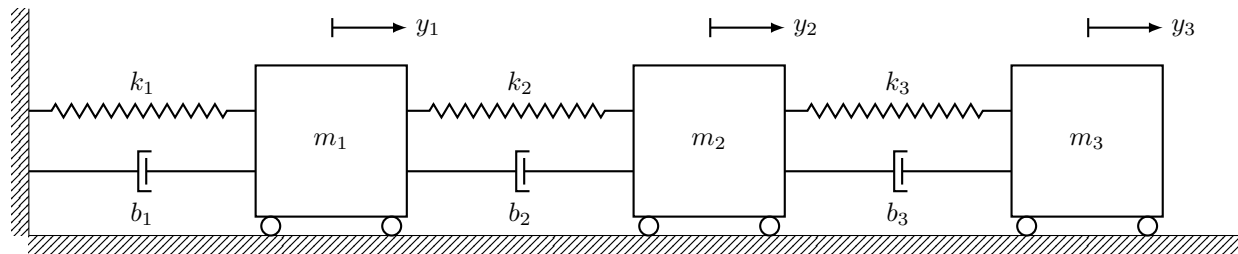
Lagrangian mechanics. For a system with n particles, Lagrange's equations of motion are the following set of n coupled second-order differential equations:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \sum_k Q_k$$

Hamiltonian mechanics. For a system with n particles, Hamilton's equations of motion are the following set of $2n$ coupled first-order differential equations:

$$\begin{aligned} \dot{q}_i &= + \frac{\partial H}{\partial p_i} \\ \dot{p}_i &= - \frac{\partial H}{\partial q_i} + \sum_k Q_k \end{aligned}$$

Triple spring–mass–damper



Newtonian mechanics. As this is a relatively simple system, we can easily derive the equations of motion using Newtonian mechanics. To do so, we first draw a free-body diagram for each mass, where we indicate all forces acting on the mass. Using Newton's law that the mass times the acceleration is equal to the sum of the forces, we obtain the following equations of motion.

$$\begin{aligned} m_1 \ddot{y}_1 &= f_1 - k_1 y_1 - b_1 \dot{y}_1 + k_2 (y_2 - y_1) + b_2 (\dot{y}_2 - \dot{y}_1) \\ m_2 \ddot{y}_2 &= f_2 - k_2 (y_2 - y_1) - b_2 (\dot{y}_2 - \dot{y}_1) + k_3 (y_3 - y_2) + b_3 (\dot{y}_3 - \dot{y}_2) \\ m_3 \ddot{y}_3 &= f_3 - k_3 (y_3 - y_2) - b_3 (\dot{y}_3 - \dot{y}_2) \end{aligned}$$

Lagrangian mechanics. We can also derive the equations of motion using Lagrangian mechanics. The kinetic energy of the system is the sum of the kinetic energies of each mass,

$$T = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 + \frac{1}{2} m_3 v_3^2$$

where $v_i = \dot{y}_i$ is the velocity of mass i for $i = 1, 2, 3$. The potential energy is the sum of the potential energies of each spring,

$$U = \frac{1}{2} k_1 y_1^2 + \frac{1}{2} k_2 (y_2 - y_1)^2 + \frac{1}{2} k_3 (y_3 - y_2)^2$$

The Lagrangian is then the difference between the kinetic and potential energies, $L = T - U$, and the equations of motion are

$$\frac{d}{dt} \frac{\partial L}{\partial v_i} - \frac{\partial L}{\partial y_i} = Q_i$$

for $i = 1, 2, 3$, where Q_i is the generalized force acting on mass i . The generalized force is sum of the applied force and the forces due to the dampers. Take the first mass for instance. The partial derivatives of the Lagrangian are

$$\frac{\partial L}{\partial v_1} = m_1 v_1 \quad \text{and} \quad \frac{\partial L}{\partial y_1} = -k_1 y_1 + k_2 (y_2 - y_1),$$

and the generalized force is $Q_1 = f_1 - b_1 v_1 + b_2 (v_2 - v_1)$. Substituting these quantities into the Lagrange equation gives

$$m_1 \dot{v}_1 + k_1 y_1 - k_2 (y_2 - y_1) = f_1 - b_1 v_1 + b_2 (v_2 - v_1)$$

which is the same as that obtained using Newtonian mechanics.

Hamiltonian mechanics. Alternatively, we can also derive the equations of motion using Hamiltonian mechanics. The Hamiltonian is the total energy of the system, $H = T + U$. Instead of the velocities v_i , Hamiltonian mechanics is formulated in terms of the conjugate momentum $p_i = \frac{\partial L}{\partial \dot{q}_i} = m_i v_i$, where $q_i = y_i$ is the generalized coordinate. In terms of the generalized coordinates and conjugate momenta, the Hamiltonian is

$$H = T + U = \frac{1}{2m_1} p_1^2 + \frac{1}{2m_2} p_2^2 + \frac{1}{2m_3} p_3^2 + \frac{1}{2} k_1 q_1^2 + \frac{1}{2} k_2 (q_2 - q_1)^2 + \frac{1}{2} k_3 (q_3 - q_2)^2$$

The equations of motion are then

$$\dot{q}_i = + \frac{\partial H}{\partial p_i} \quad \text{and} \quad \dot{p}_i = - \frac{\partial H}{\partial q_i} + Q_i \quad \text{for } i = 1, 2, 3.$$

where Q_i is the same generalized force as in the Lagrangian formulation. For instance, the equations of motion for the first mass are

$$\dot{q}_1 = \frac{1}{m_1} p_1 \quad \text{and} \quad \dot{p}_1 = -k_1 q_1 + k_2 (q_2 - q_1) + f_1 - \frac{b_1}{m_1} p_1 + \frac{b_2}{m_2} p_2 - \frac{b_2}{m_1} p_1$$

The first equation states that $v_1 = \dot{y}_1$, and the second equation is equivalent to the equation of motion for the first mass obtained from Newtonian mechanics.

Summary. All three approaches to modeling result in the same equations of motion for the system. In this case, Newton's laws are simple to apply and are the most straightforward approach to obtaining the equations, but for more complicated systems Lagrangian and Hamiltonian mechanics are simpler.

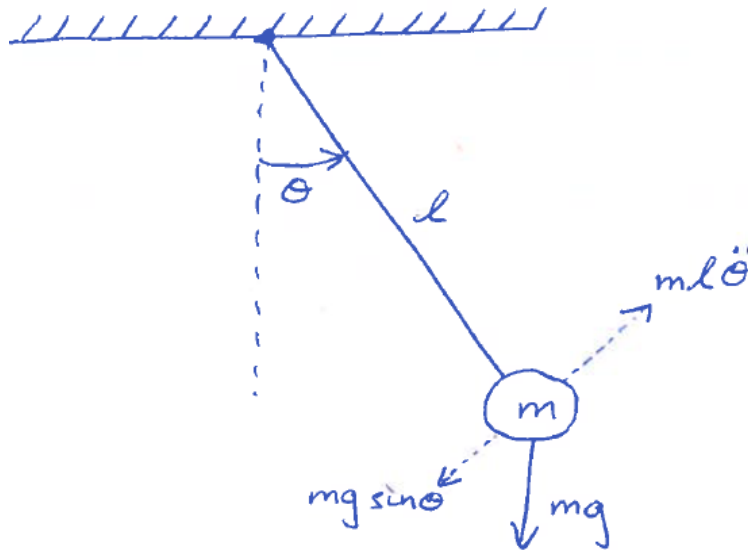
To summarize, the triple spring-mass-damper system is described by the linear time-invariant state-space

model

$$\underbrace{\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{v}_1 \\ \dot{v}_2 \\ \dot{v}_3 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{1}{m_1}(k_1 + k_2) & \frac{1}{m_1}k_2 & 0 & -\frac{1}{m_1}(b_1 + b_2) & \frac{1}{m_1}b_2 & 0 \\ \frac{1}{m_2}k_2 & -\frac{1}{m_2}(k_2 + k_3) & \frac{1}{m_2}k_3 & \frac{1}{m_2}b_2 & -\frac{1}{m_2}(b_2 + b_3) & \frac{1}{m_2}b_3 \\ 0 & \frac{1}{m_3}k_3 & -\frac{1}{m_3}k_3 & 0 & \frac{1}{m_3}b_3 & -\frac{1}{m_3}b_3 \end{bmatrix}}_A \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_1} & 0 & 0 \\ 0 & \frac{1}{m_2} & 0 \\ 0 & 0 & \frac{1}{m_3} \end{bmatrix}}_B \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}}_u$$

Simple pendulum

Consider a simple pendulum with mass m , length ℓ , acceleration due to gravity g , angle θ , and torque τ applied at the joint in the direction of θ .



Newtonian mechanics. Perhaps the simplest method to obtain the dynamics of the system are to apply a force balance on the mass of the pendulum (as shown in the above diagram). Since the sum of the forces perpendicular to the rod must be zero, we have that

$$ml\ddot{\theta} + mg \sin \theta = \frac{\tau}{\ell}$$

where the applied torque is the applied force multiplied by the length of the pendulum.

Lagrangian mechanics. We can also construct the equations of motion using variational mechanics. Using the position of the joint as the origin, the Cartesian coordinates of the mass are

$$x = \ell \sin \theta \quad \text{and} \quad y = -\ell \cos \theta.$$

Since the length of the rod is constant in time while the angle varies in time, the time derivative of the coordinates is

$$\dot{x} = \ell \dot{\theta} \cos \theta \quad \text{and} \quad \dot{y} = -\ell \dot{\theta} \sin \theta.$$

The kinetic energy due to the motion of the mass is

$$T = \frac{1}{2} m v^2 = \frac{m}{2} (\dot{x}^2 + \dot{y}^2) = \frac{m \ell^2 \dot{\theta}^2}{2}.$$

The potential energy due to the height of the mass is

$$U = mgy = -mg\ell \cos \theta.$$

The Lagrangian is then the difference between the kinetic and potential energies,

$$L = T - V = \frac{1}{2} m \ell^2 \dot{\theta}^2 + mg\ell \cos \theta.$$

Lagrange's equation of motion is the second-order differential equation

$$\begin{aligned} \tau &= \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} \\ &= \frac{d}{dt} (m \ell^2 \dot{\theta}) - (-mg\ell \sin \theta) \\ &= m \ell^2 \ddot{\theta} + mg\ell \sin \theta \end{aligned}$$

Note that this is the same equation of motion that we obtained using Newtonian mechanics.

Hamiltonian mechanics. We can also derive the equations of motion using Hamiltonian mechanics as follows. The generalized momentum corresponding to the generalized coordinate $q = \theta$ is

$$p = \frac{\partial L}{\partial \dot{q}} = m \ell^2 \dot{\theta}.$$

The Hamiltonian is then the total energy in the system,

$$H = p\dot{q} - L = T + V = \frac{1}{2} m \ell^2 \dot{\theta}^2 - mg\ell \cos \theta.$$

In order to compute the partial derivatives of the Hamiltonian, we need to express H in terms of p and q ,

$$H = \frac{1}{2m\ell^2} p^2 - mg\ell \cos q.$$

The partial derivatives of the Hamiltonian are then

$$\frac{\partial H}{\partial p} = \frac{1}{m\ell^2} p \quad \text{and} \quad \frac{\partial H}{\partial q} = mg \sin q.$$

Therefore, Hamiltonian's equations of motion are the two first-order differential equations

$$\begin{aligned} \dot{q} &= \frac{1}{m\ell^2} p \\ \dot{p} &= -mg \sin q + \tau \end{aligned}$$

We recover Lagrange's equations of motion by substituting \dot{p} from the first equation into the second.

5

Linearization

The equations of motion that describe the dynamics of a system are often *nonlinear*. Such systems are much more difficult to analyze and control than their linear counterparts. In this chapter, we study several methods for constructing linear systems from nonlinear systems. In many cases this linearization is only an approximation to the original system (as in Jacobian linearization), although it is sometimes possible to use feedback to linearize the dynamics (as in feedback linearization).

5.1 Jacobian linearization

Jacobian linearization is a method for approximating a nonlinear system by a linear system. The Jacobian linearization approximates a nonlinear system about an equilibrium point, or more general, any given nominal trajectory of the system. The linearization about a fixed equilibrium point produces a linear time-invariant system, while the linearization about a time-varying trajectory produces a linear time-varying system. This is a local approximation in that the linear model is only a good approximation to the nonlinear system if the system stays near the equilibrium (or nominal trajectory). This tool allows us to analyze and control (to some extent) nonlinear systems using their linearization.

Linearization about an equilibrium point

Equilibrium points (or fixed points) are states of a system for which the system does not move when initialized at the equilibrium.

Consider a time-invariant continuous-time system

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

where $x(t)$ is the state and $u(t)$ the input at time t . A state \tilde{x} is an *equilibrium point* of the system if there exists a constant input \tilde{u} , called the *equilibrium input*, such that

$$f(\tilde{x}, \tilde{u}) = 0$$

If we initialize the system at the equilibrium point $x(0) = \tilde{x}$ and apply the equilibrium input $u(t) \equiv \tilde{u}$ for all $t \geq 0$, then the state does not change in time since $\dot{x}(t) \equiv 0$. Therefore, the trajectory of the system is simply $x(t) \equiv \tilde{x}$ for all $t \geq 0$. In other words, the system remains at the equilibrium point for all time.

The concept of an equilibrium point is the same in discrete time, although the definition is slightly different.

Consider a discrete-time dynamical system

$$x(k+1) = f(x(k), u(k)).$$

For the state to remain the same, we require an equilibrium point \tilde{x} and its corresponding equilibrium input \tilde{u} to satisfy $\tilde{x} = f(\tilde{x}, \tilde{u})$. When the state is initialized at $x(0) = \tilde{x}$ and we apply the equilibrium input $u(k) = \tilde{u}$ for all $k \geq 0$, the state remains at the equilibrium point for all time.

Given an equilibrium point \tilde{x} and its corresponding equilibrium input \tilde{u} , we know that if we start the system from the equilibrium point \tilde{x} and apply the equilibrium input \tilde{u} , then the system will stay at the equilibrium point. But what happens as the state deviates away from the equilibrium point and we apply a slightly different input? To study this case, we will use the *deviation* variables

$$\delta_x(t) = x(t) - \tilde{x} \quad \text{and} \quad \delta_u(t) = u(t) - \tilde{u}$$

which are the differences between the actual state and input and the equilibrium.

Using the definition of the deviation variables and the fact that (\tilde{x}, \tilde{u}) is an equilibrium of the system, the Jacobian linearization of the nonlinear system about the equilibrium is

$$\dot{\delta}_x(t) = A\delta_x(t) + B\delta_u(t)$$

where

$$A = \frac{\partial f}{\partial x}(\tilde{x}, \tilde{u}) \quad \text{and} \quad B = \frac{\partial f}{\partial u}(\tilde{x}, \tilde{u})$$

The state-space matrices A and B are constant and are the Jacobian of f evaluated at the equilibrium. This is a linear time-invariant system in the deviation variables. The Jacobian linearization is the first-order Taylor series approximation for the state equation in terms of the deviation variables.

Example (pendulum). The simple pendulum is described by the nonlinear differential equation

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = f(x_1, x_2) = \begin{bmatrix} x_2 \\ -\frac{g}{\ell} \sin(x_1) \end{bmatrix}$$

where $x_1 = \theta$ and $x_2 = \dot{\theta}$. The equilibrium points are the solutions to $f(\tilde{x}_1, \tilde{x}_2) = 0$, which are points of the form $(n\pi, 0)$ for any integer n . The Jacobian is

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{\ell} \cos(x_1) & 0 \end{bmatrix}$$

Therefore, the linearization about the equilibrium $(0, 0)$ (corresponding to the down position) is

$$\begin{bmatrix} \dot{\delta}_{x_1} \\ \dot{\delta}_{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{\ell} & 0 \end{bmatrix} \begin{bmatrix} \delta_{x_1} \\ \delta_{x_2} \end{bmatrix} \quad \text{where} \quad \begin{array}{l} \delta_{x_1} = x_1 \\ \delta_{x_2} = x_2 \end{array}$$

and the linearization about the equilibrium $(\pi, 0)$ (corresponding to the up position) is

$$\begin{bmatrix} \dot{\delta}_{x_1} \\ \dot{\delta}_{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{g}{\ell} & 0 \end{bmatrix} \begin{bmatrix} \delta_{x_1} \\ \delta_{x_2} \end{bmatrix} \quad \text{where} \quad \begin{array}{l} \delta_{x_1} = x_1 - \pi \\ \delta_{x_2} = x_2 \end{array}$$

Example (predator–prey). The dynamics between the populations of a predator species and prey species can be modeled using the Lotka–Volterra equations

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = f(x_1, x_2) = \begin{bmatrix} ax_1 - bx_1x_2 \\ cx_1x_2 - dx_2 \end{bmatrix}$$

where x_1 is the number of prey, x_2 is the number of predators, and the constant parameters (a, b, c, d) represent the relationship between the two species. The equilibrium points are solutions to the equation $f(\tilde{x}_1, \tilde{x}_2) = 0$. This has the trivial equilibrium $(0, 0)$, where the population of both species is zero. It also has a non-trivial equilibrium $(\frac{d}{c}, \frac{a}{b})$. The Jacobian is

$$\frac{\partial f}{\partial x} = \begin{bmatrix} a - bx_2 & -bx_1 \\ cx_2 & cx_1 - d \end{bmatrix}$$

Therefore, the linearization of the system about the non-trivial equilibrium is

$$\begin{bmatrix} \dot{\delta}_{x_1} \\ \dot{\delta}_{x_2} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{bd}{c} \\ \frac{ac}{b} & 0 \end{bmatrix} \begin{bmatrix} \delta_{x_1} \\ \delta_{x_2} \end{bmatrix} \quad \text{where} \quad \begin{aligned} \delta_{x_1} &= x_1 - \frac{d}{c} \\ \delta_{x_2} &= x_2 - \frac{a}{b} \end{aligned}$$

Linearization about an equilibrium trajectory

Instead of linearizing the system about a constant equilibrium, we can also linearize it about a given nominal trajectory. Suppose that $(\tilde{x}(t), \tilde{u}(t))$ is a trajectory of the system, meaning that it satisfies the dynamics

$$\dot{\tilde{x}}(t) = f(\tilde{x}(t), \tilde{u}(t), t)$$

The deviation from this trajectory is then

$$\delta_x(t) = x(t) - \tilde{x}(t) \quad \text{and} \quad \delta_u(t) = u(t) - \tilde{u}(t)$$

Using the fact that (\tilde{x}, \tilde{u}) is a trajectory of the system, the Jacobian linearization of the nonlinear system about the trajectory is

$$\dot{\delta}_x(t) = A(t) \delta_x(t) + B(t) \delta_u(t)$$

where

$$A(t) = \frac{\partial f}{\partial x}(\tilde{x}(t), \tilde{u}(t), t) \quad \text{and} \quad B(t) = \frac{\partial f}{\partial u}(\tilde{x}(t), \tilde{u}(t), t)$$

The state-space matrices A and B now depend on time, so this is a linear *time-varying* system in the deviation variables.

5.2 Feedback linearization

Feedback linearization is a control design technique for nonlinear control affine systems that finds an input signal that transforms the system into an equivalent linear system. In contrast to Jacobian linearization, feedback linearization is *not* an approximation.

Consider a nonlinear control affine system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned}$$

where x is the state, u is the input, and y is the output. The goal is to construct a control input

$$u = a(x) + b(x)v$$

that renders the map between the new input v and the output y linear. The original nonlinear system can then be controlled using linear control design strategies to design the auxiliary control input v , and then using this to construct the input u for the nonlinear system.

Example (feedback linearization). Consider the nonlinear dynamical system

$$\begin{aligned}\dot{x}_1 &= x_1 + x_2 \\ \dot{x}_2 &= 3x_1^2 x_2 + x_1 + u \\ y &= -x_1^3 + x_2\end{aligned}$$

We will define a transformation on the state and input such that the resulting dynamics are linear from the transformed input to the output. Define the transformed control input v and the transformed state (ξ_1, ξ_2) as

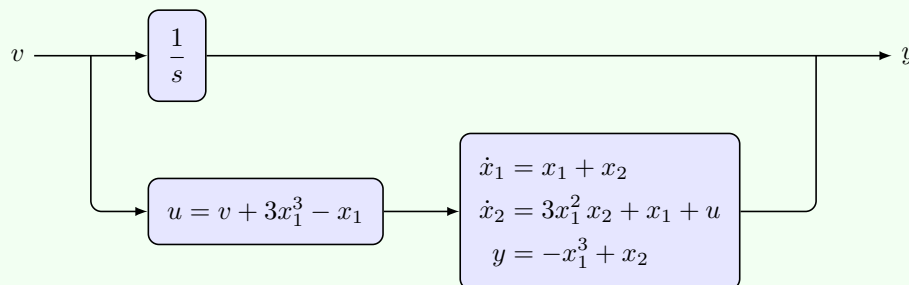
$$\begin{aligned}v &= u - 3x_1^3 + x_1 & u &= v + 3\xi_1^3 - \xi_1 \\ \xi_1 &= x_1 & \text{whose inverse is } & x_1 = \xi_1 \\ \xi_2 &= -x_1^3 + x_2 & & x_2 = \xi_1^3 + \xi_2\end{aligned}$$

(the derivation of this transformation requires more advanced tools that we will not study here). The dynamics of the transformed state in terms of the transformed input are then

$$\begin{aligned}\dot{\xi}_1 &= \xi_1 + \xi_1^3 + \xi_2 \\ \dot{\xi}_2 &= v \\ y &= \xi_2\end{aligned}$$

The state ξ_1 does not affect the output and can therefore be neglected. The dynamics from the transformed input v to the output y are linear and time invariant with transfer function $1/s$.

To control the nonlinear system, we can design a controller for the linear system to find the transformed control input v , use the input transformation to obtain the untransformed control input u , and then apply this to the original nonlinear system. This is illustrated as follows.



6

Linear Time-Invariant Systems

We have seen how to model general dynamical systems as differential equations (or difference equations in discrete time). While we can also use this type of representation for linear time-invariant systems, such systems have many other useful representations that we will study in this chapter, including their representation as block diagrams, state-space realizations, and transfer functions.

6.1 Representations of LTI systems

There are various ways of representing an LTI system, such as the following:

- transfer function (or transfer matrix)
- n^{th} -order constant coefficient differential equation
- state-space representation
- block diagram

6.2 Transfer function

We can find the transfer function of the system directly from a state-space realization. The transfer function of a system is unique, even though the state-space realization is not.

Consider the continuous-time LTI system

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

Taking the Laplace transform of the state and output equations and assuming that the initial state is zero (since the transfer function only describes how the input signal affects the output), we obtain

$$\begin{aligned}sX(s) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s)\end{aligned}$$

In the state equation, bring both terms with the state to the same side. Since the state is a vector, we need to factor out an identity matrix,

$$(sI - A)X(s) = BU(s)$$

We can now solve for the state by multiplying on the left by the inverse of the matrix $sI - A$, which gives

$$X(s) = (sI - A)^{-1}BU(s)$$

Now that we know the state, we can substitute this into the output equation to find the output in terms of the input,

$$Y(s) = CX(s) + DU(s) = [C(sI - A)^{-1}B + D]U(s)$$

Since the transfer function is such that $Y(s) = H(s)U(s)$, the quantity in square brackets must be the transfer function.

$$H(s) = C(sI - A)^{-1}B + D$$

The derivation in discrete time is identical with the Laplace variable s replaced by the z -transform variable z .

The transfer function of an LTI system with state-space matrices (A, B, C, D) is

$$H(\lambda) = C(\lambda I - A)^{-1}B + D,$$

where λ is s in continuous time or z in discrete time.

Remark. For systems with a single input and single output (SISO), the transfer function is a scalar function. But for multi-input multi-output (MIMO) systems, it is a transfer *matrix*. The transfer function is in general a $p \times m$ matrix, where p is the number of outputs and m is the number of inputs. ■

Example (spring-mass-damper system). Recall that the relationship between the applied force u and position y of the mass in the spring-mass-damper mechanical system is given by the LTI dynamical system with state-space realization

$$y = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ -\frac{k}{m} & -\frac{b}{m} & \frac{1}{m} \\ 1 & 0 & 0 \end{array} \right] f.$$

The transfer function is then

$$\frac{Y(s)}{F(s)} = [1 \quad 0] \left(z \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} = \frac{1}{ms^2 + bs + k}.$$

Poles of the transfer function and eigenvalues of A

The inverse of the matrix in the transfer function can be written

$$(sI - A)^{-1} = \frac{\text{adj}(sI - A)}{\det(sI - A)}$$

where $\text{adj}(\cdot)$ denote the adjugate, and $\det(\cdot)$ denotes the determinant. The adjugate is a polynomial matrix in s , so the only terms in the denominator are due to the determinant of $sI - A$. This determinant is precisely the characteristic polynomial of A . The roots of the characteristic polynomial are the eigenvalues of A , which are also poles of the transfer function. Therefore, any pole of the transfer function must also be an eigenvalue of the state transition matrix A .

All poles of the transfer function are also eigenvalues of the state transition matrix.

$$\text{poles of } H(s) \quad \subseteq \quad \text{eigenvalues of } A$$

Example. Going back to our spring–mass–damper system, suppose the coefficients are $k = m = 1$ and $b = 2$. Then the eigenvalues of the A matrix in the state-space representation are

$$\text{eig}\left(\begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}\right) = \{-1, -1\}$$

and the denominator of the transfer function factors as

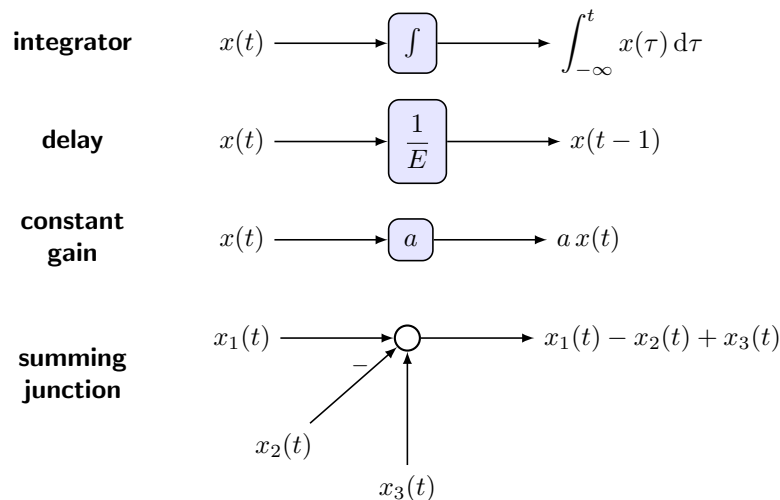
$$s^2 + 2s + 1 = (s + 1)^2$$

which has a repeated root at negative one.

6.3 Block diagrams and state-space realizations

We now study how to construct a state-space realization of an LTI system from its transfer function (or equivalently, an n^{th} -order differential equation). Corresponding to the state-space realization will also be a block diagram, which is a visual representation of the system. As we will see, the state-space representation (and therefore also the block diagram) is *not* unique. A state-space representation of a system is called a *realization*. In this section, we will see several state-space realizations of LTI systems.

A block diagram is a visual representation of a dynamical system. This representation is useful for identifying structure in the system and how the various signals interact with each other. In the block diagram representation of a system, arrows represent signals, blocks represent systems, and circles represent summing junctions. Any LTI system can be described by a block diagram with the following components.



By default, we assume that all signals entering a summing junction (represented by a circle) are summed. To subtract a signal, we place a negative sign on the arrow just before the summing junction.

For simplicity, consider a general second order system with transfer function

$$H(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^2 + a_1 s + a_0}$$

which represents the differential equation

$$\ddot{y} + a_1 \dot{y} + a_0 y = b_2 \ddot{u} + b_1 \dot{u} + b_0 u.$$

We will construct several state-space realizations and their corresponding block diagrams for this second-order system; it is straightforward to generalize the results to higher-order systems.

Direct form

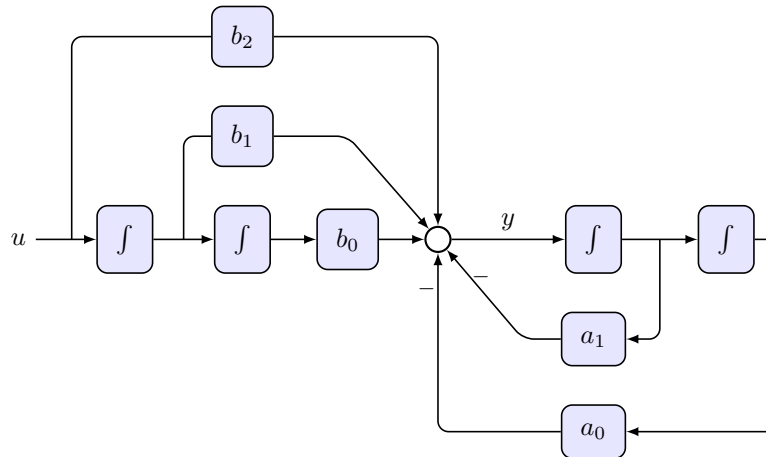
As its name suggests, the direct form is the most straightforward to construct. First, multiply the equation $Y(s) = H(s)U(s)$ by the denominator of the transfer function and divide by s^2 to obtain

$$Y(s) + \frac{a_1}{s} Y(s) + \frac{a_0}{s^2} Y(s) = b_2 U(s) + \frac{b_1}{s} U(s) + \frac{b_0}{s^2} U(s).$$

Now solve for the first $Y(s)$ to obtain

$$Y(s) = -\frac{a_1}{s} Y(s) - \frac{a_0}{s^2} Y(s) + b_2 U(s) + \frac{b_1}{s} U(s) + \frac{b_0}{s^2} U(s)$$

The direct form of the block diagram represents this equation explicitly. Since there is only one equation, the diagram has only a single summing junction.



A state-space representation of the direct form with x_1 the state of the rightmost integrator and x_4 the state of the leftmost integrator is as follows:

$$y = \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ -a_0 & -a_1 & b_0 & b_1 & b_2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline -a_0 & -a_1 & b_0 & b_1 & b_2 \end{array} \right] u.$$

While the direct form is easy to construct, notice that it uses twice as many integrators as the order of the system (in this case, four integrator blocks for a second-order system). When implementing the system, each integrator block corresponds to a variable that must be stored. It turns out that an n^{th} -order system only needs n integrator blocks to be implemented, so more efficient implementations exist (as described next).

Controllable canonical form

To construct the controllable canonical form, we will manipulate the transfer function before drawing the corresponding block diagram. In particular, let's define the intermediate signal $w(t)$ as

$$W(s) = \frac{1}{s^2 + a_1 s + a_0} U(s).$$

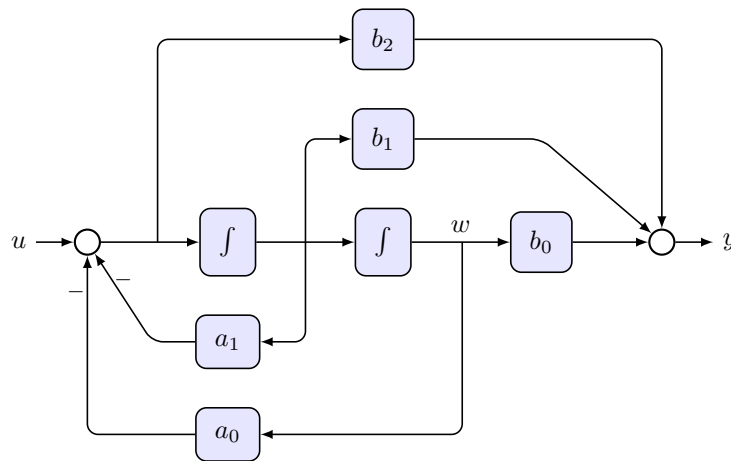
In terms of this intermediate signal, the output is

$$Y(s) = (b_2 s^2 + b_1 s + b_0) W(s)$$

and the input is

$$U(s) = (s^2 + a_1 s + a_0) W(s).$$

The direct form represents these two equations explicitly. Since there are two equations, the block diagram has two summing junctions.



This block diagram uses only two integrator blocks, which is the minimum number required for a second-order system. The state-space representation corresponding to this block diagram is as follows:

$$y = \left[\begin{array}{cccc|c} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} & 1 \\ \hline b_0 - b_n a_0 & b_1 - b_n a_1 & b_2 - b_n a_2 & \dots & b_{n-1} - b_n a_{n-1} & b_n \end{array} \right] u \quad \text{with state} \quad x = \begin{bmatrix} w \\ \dot{w} \\ \ddot{w} \\ \vdots \\ w^{(n-1)} \end{bmatrix}.$$

Remark. Later, we will see why this state-space model is called controllable, but the main idea is that we can always choose the input to make the state of the system anything we want: we can control the state. ■

Observable canonical form

To construct the observable canonical form, we first define a sequence of auxiliary variables. Define the first intermediate variable as

$$x_2 = y - b_2 u$$

so that the differential equation becomes

$$\ddot{x}_2 + a_1 \dot{y} + a_0 y = b_1 \dot{u} + b_0 u.$$

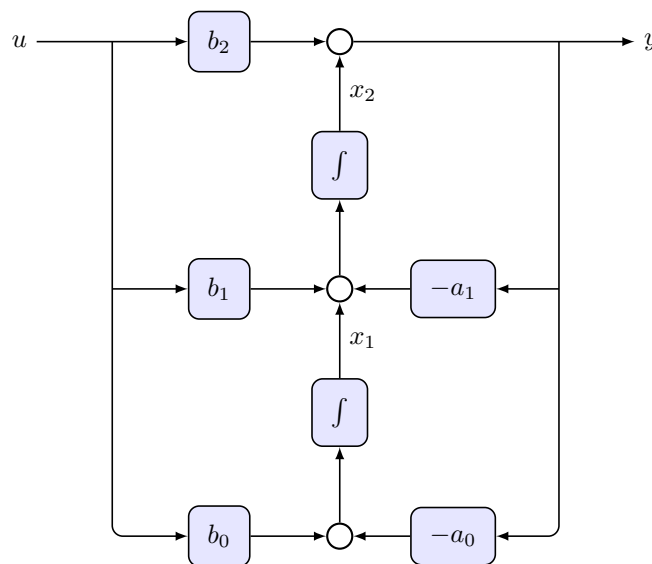
Now define the next intermediate variable as

$$x_1 = \dot{x}_2 + a_1 y - b_1 u.$$

Substituting this into the differential equation yields

$$\dot{x}_1 + a_0 y = b_0 u.$$

These equations have the following block diagram representation. As with the controllable canonical form, this block diagram also uses the minimum number of integrator blocks.



The state-space representation corresponding to this block diagram is as follows:

$$y = \left[\begin{array}{ccccc|c} 0 & 0 & \dots & 0 & -a_0 & b_0 - b_n a_0 \\ 1 & 0 & \dots & 0 & -a_1 & b_1 - b_n a_1 \\ 0 & 1 & \dots & 0 & -a_2 & b_2 - b_n a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -a_{n-1} & b_{n-1} - b_n a_{n-1} \\ \hline 0 & 0 & \dots & 0 & 1 & b_n \end{array} \right] u.$$

Remark. Later, we will see why this state-space model is called observable, but the main idea is that we can always reconstruct the state of the system if we measure the input and output of the system over a long enough period of time: we can observe the state. ■

6.4 State transformations

As we have seen, the state-space realization (A, B, C, D) of an LTI system is not unique. We now describe the relationship between various realizations.

Consider an LTI system with state-space realization

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

Now suppose we transform the state from x to z where

$$z(t) = T^{-1}x(t) \quad \text{or} \quad x(t) = Tz(t). \quad (T \text{ invertible})$$

Replacing x in the state equations with Tz and using that T is constant in time yields

$$\begin{aligned}T\dot{z} &= ATz + Bu \\ y &= CTz + Du\end{aligned}$$

Multiplying the state equation on the left by T^{-1} (which exists because T is invertible by assumption) yields

$$\begin{aligned}\dot{z} &= (A^{-1}AT)z + (T^{-1}B)u \\ y &= (CT)z + Du\end{aligned}$$

These are equivalent realizations in that they represent the same relationship between the input u and output y . The only difference is the internal representation of the state. We denote that two realizations are equivalent by

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \sim \left[\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & D \end{array} \right] \quad (T \text{ invertible})$$

Since these two realizations are equivalent for any invertible state transformation, there are an infinite number of equivalent realizations!

Fact (Controllability and observability matrices under state transformations). Consider a state-space realization (A, B, C, D) with controllability matrix P and observability matrix Q . For any state transformation matrix T , the transformed system

$$\left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array} \right] = \left[\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & D \end{array} \right]$$

has controllability matrix $\hat{P} = T^{-1}P$ and observability matrix $\hat{Q} = QT$.

Fact (State transformation between two realizations). Consider two state-space realizations

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \quad \text{and} \quad \left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array} \right]$$

of the same transfer function with states x and z , respectively. The state transformation $z = T^{-1}x$ is

- $T = P\hat{P}^{-1}$ if both realizations are controllable, and
- $T = Q^{-1}\hat{Q}$ if both realizations are observable.

Part III

Analysis

7

Canonical Forms

7.1 Direct form

For simplicity, consider a general second order system with transfer function

$$H(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^2 + a_1 s + a_0}$$

which represents the differential equation

$$\ddot{y} + a_1 \dot{y} + a_0 y = b_2 \ddot{u} + b_1 \dot{u} + b_0 u.$$

We will construct several state-space realizations and their corresponding block diagrams for this second-order system; it is straightforward to generalize the results to higher-order systems.

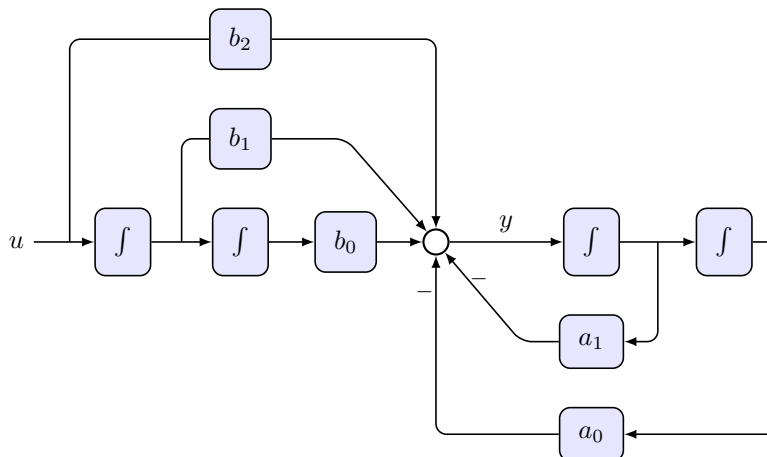
As its name suggests, the direct form is the most straightforward to construct. First, multiply the equation $Y(s) = H(s)U(s)$ by the denominator of the transfer function and divide by s^2 to obtain

$$Y(s) + \frac{a_1}{s} Y(s) + \frac{a_0}{s^2} Y(s) = b_2 U(s) + \frac{b_1}{s} U(s) + \frac{b_0}{s^2} U(s).$$

Now solve for the first $Y(s)$ to obtain

$$Y(s) = -\frac{a_1}{s} Y(s) - \frac{a_0}{s^2} Y(s) + b_2 U(s) + \frac{b_1}{s} U(s) + \frac{b_0}{s^2} U(s)$$

The direct form of the block diagram represents this equation explicitly. Since there is only one equation, the diagram has only a single summing junction.



A state-space representation of the direct form with x_1 the state of the rightmost integrator and x_4 the state of the leftmost integrator is as follows:

$$y = \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ -a_0 & -a_1 & b_0 & b_1 & b_2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline -a_0 & -a_1 & b_0 & b_1 & b_2 \end{array} \right] u.$$

While the direct form is easy to construct, notice that it uses twice as many integrators as the order of the system (in this case, four integrator blocks for a second-order system). When implementing the system, each integrator block corresponds to a variable that must be stored. It turns out that an n^{th} -order system only needs n integrator blocks to be implemented, so more efficient implementations exist (as described next).

7.2 Controllable canonical form

To construct the controllable canonical form, we will manipulate the transfer function before drawing the corresponding block diagram. In particular, let's define the intermediate signal $w(t)$ as

$$W(s) = \frac{1}{s^2 + a_1 s + a_0} U(s).$$

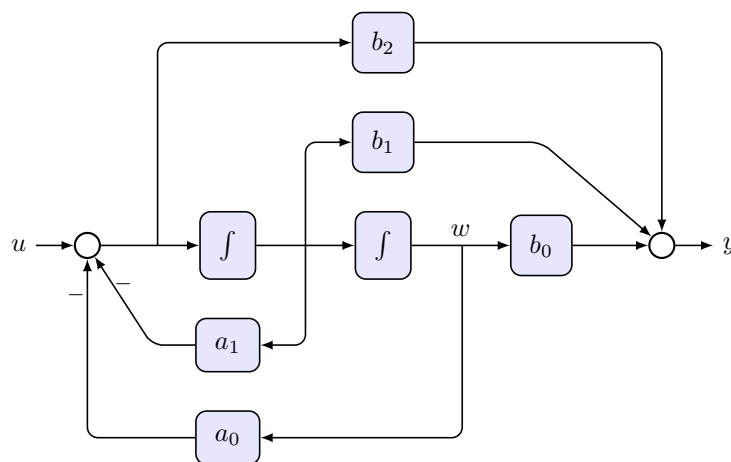
In terms of this intermediate signal, the output is

$$Y(s) = (b_2 s^2 + b_1 s + b_0) W(s)$$

and the input is

$$U(s) = (s^2 + a_1 s + a_0) W(s).$$

The direct form represents these two equations explicitly. Since there are two equations, the block diagram has two summing junctions.



This block diagram uses only two integrator blocks, which is the minimum number required for a second-order

system. The state-space representation corresponding to this block diagram is as follows:

$$y = \left[\begin{array}{cccc|c} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \\ \hline b_0 - b_n a_0 & b_1 - b_n a_1 & b_2 - b_n a_2 & \dots & b_{n-1} - b_n a_{n-1} \\ \hline \end{array} \right] u \quad \text{with state} \quad x = \begin{bmatrix} w \\ \dot{w} \\ \ddot{w} \\ \vdots \\ w^{(n-1)} \end{bmatrix}.$$

Remark. Later, we will see why this state-space model is called controllable, but the main idea is that we can always choose the input to make the state of the system anything we want: we can control the state. ■

7.3 Observable canonical form

To construct the observable canonical form, we first define a sequence of auxiliary variables. Define the first intermediate variable as

$$x_2 = y - b_2 u$$

so that the differential equation becomes

$$\dot{x}_2 + a_1 \dot{y} + a_0 y = b_1 \dot{u} + b_0 u.$$

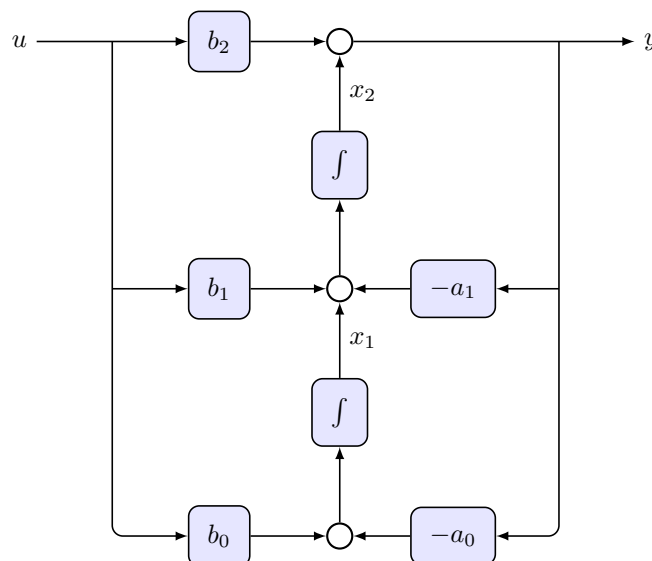
Now define the next intermediate variable as

$$x_1 = \dot{x}_2 + a_1 y - b_1 u.$$

Substituting this into the differential equation yields

$$\dot{x}_1 + a_0 y = b_0 u.$$

These equations have the following block diagram representation. As with the controllable canonical form, this block diagram also uses the minimum number of integrator blocks.



The state-space representation corresponding to this block diagram is as follows:

$$y = \left[\begin{array}{ccccc|c} 0 & 0 & \dots & 0 & -a_0 & b_0 - b_n a_0 \\ 1 & 0 & \dots & 0 & -a_1 & b_1 - b_n a_1 \\ 0 & 1 & \dots & 0 & -a_2 & b_2 - b_n a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -a_{n-1} & b_{n-1} - b_n a_{n-1} \\ \hline 0 & 0 & \dots & 0 & 1 & b_n \end{array} \right] u.$$

7.4 Kalman canonical form

7.5 Diagonal canonical form

Definition (DCF). The *diagonal canonical form* (DCF) of an LTI system is a state-space realization (A, B, C, D) of the system in which the matrix A is diagonal.

7.6 Jordan canonical form

Definition (JCF). The *Jordan canonical form* (JCF) of an LTI system is a state-space realization (A, B, C, D) of the system in which the matrix A is in Jordan form.

8

Response

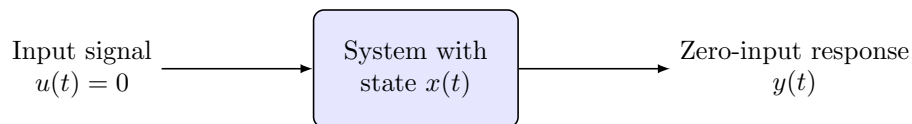
The most fundamental question regarding the analysis of a system is: *how does the system respond to various excitations?* The output of a system, known as the *response*, depends on both the input signal and the initial conditions. In an RC circuit, for instance, the voltage across the capacitor depends on both the source voltage and the initial capacitor voltage.

8.1 Types of responses

There are various types of responses, based on the nature of the input signal and initial condition. We now describe several important responses of a system.

Zero-input response

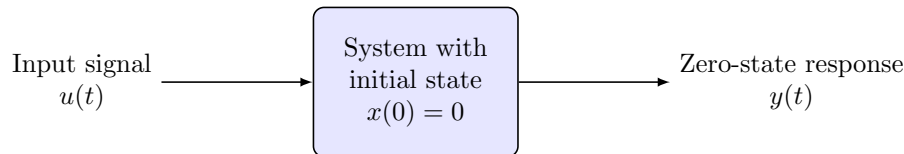
The zero-input response (ZIR) is the output of the system due to the initial conditions when the input signal is zero (at all times). The zero-input response depends only on the initial state of the system.



Example (ZIR of spring–mass–damper system). For a spring–mass–damper mechanical system, the zero-input response is the position of the mass when no force is applied. In this case, the response depends on only the initial position and velocity of the mass.

Zero-state response

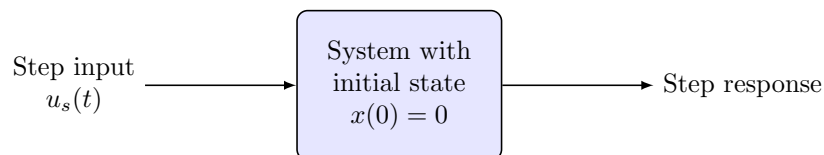
The *zero-state response* is the output of the system due to the input signal when the initial state is zero. By “initial state”, we typically mean the state at time zero, in which case it is assumed that the input signal is zero for times before the initial time.



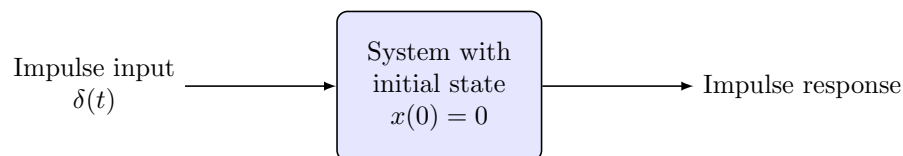
Example (ZSR of spring–mass–damper system). Continuing our example, the zero-state response of the mechanical system is the position of the mass when it starts at rest. In this case, the response depends on only the force applied to the mass.

The zero-state response depends on the particular input signal. Two common zero-state responses are the following.

- The *step response* is the zero-state response due to a unit step input signal.



- The *impulse response* is the zero-state response due to a unit impulse input signal.



8.2 Solving the state equation for continuous-time LTI systems

We now show how to find the response from a continuous-time LTI state space model. Given the input signal and initial condition, we seek to find the output signal.

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), & x(t_0) &= x_0 \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

Remark. The state equation is a differential equation (which requires some effort to solve), while the output equation is simply algebraic. Once we have the state $x(t)$, the output is simply $y(t) = Cx(t) + Du(t)$. ■

To illustrate the approach, we will first consider the case where the state dimension is one, and then solve the general case.

One-dimensional system

For a one-dimensional system, the state equation reduces to

$$\dot{x}(t) = ax(t) + bu(t), \quad x(t_0) = x_0$$

where we use lowercase symbols a and b to emphasize that they are scalars. We can find the state in the time domain or frequency domain.

Time-domain approach. Multiply both sides of the state equation by $e^{-a(t-t_0)}$ and use the product rule to obtain

$$\frac{d}{dt} \left(e^{-a(t-t_0)} x(t) \right) = e^{-a(t-t_0)} \dot{x}(t) - a e^{-a(t-t_0)} x(t) = e^{-a(t-t_0)} b u(t)$$

Now integrate from the initial time t_0 to a generic time t ,

$$\left[e^{-a(\tau-t_0)} x(\tau) \right]_{\tau=0}^{t_0} = \int_{t_0}^t e^{-a(\tau-t_0)} b u(\tau) d\tau$$

Expanding the left-hand side,

$$e^{-a(t-t_0)} x(t) - x(t_0) = \int_{t_0}^t e^{-a(\tau-t_0)} b u(\tau) d\tau$$

Now multiply both sides by $e^{a(t-t_0)}$ and rearrange to obtain the state:

$$x(t) = e^{a(t-t_0)} x_0 + \int_{t_0}^t e^{a(t-\tau)} b u(\tau) d\tau$$

Frequency-domain approach. We can obtain the same result in the frequency domain. Taking the (unilateral) Laplace transform of the state equation,

$$s X(s) - x_0 = a X(s) + b U(s)$$

Solving for the state yields

$$X(s) = \frac{x_0}{s-a} + \frac{1}{s-a} b U(s)$$

Using that $\mathcal{L}\{e^{at}\} = \frac{1}{s-a}$ and $\mathcal{L}\{f * g\} = \mathcal{L}\{f\} \mathcal{L}\{g\}$, the state is

$$x(t) = e^{at} x_0 + e^{at} * b u(t) = e^{at} x_0 + \int_0^t e^{a(t-\tau)} b u(\tau) d\tau$$

which is the same as before (with $t_0 = 0$ due to the definition of the Laplace transform).

General case

We can use the same procedure as before to find the response to a general state space model. The only difference is that, instead of the standard exponential e^{at} , we need to use the matrix exponential e^{At} .

Time-domain approach. Multiply both sides of the state equation by the matrix exponential $e^{-A(t-t_0)}$ and use the product rule to obtain

$$\frac{d}{dt} \left(e^{-A(t-t_0)} x(t) \right) = e^{-A(t-t_0)} \dot{x}(t) - A e^{-A(t-t_0)} x(t) = e^{-A(t-t_0)} B u(t)$$

Now integrate from the initial time t_0 to a generic time t ,

$$e^{-A(t-t_0)} x(t) - x(t_0) = \int_{t_0}^t e^{-A(\tau-t_0)} B u(\tau) d\tau$$

Now multiply both sides by $e^{A(t-t_0)}$ and rearrange to obtain the state:

$$x(t) = e^{A(t-t_0)} x_0 + \int_{t_0}^t e^{A(t-\tau)} B u(\tau) d\tau$$

Frequency-domain approach. We can also derive the result using the frequency domain as before, but we need the Laplace transform of the matrix exponential e^{At} , which is $(sI - A)^{-1}$. Taking the (unilateral) Laplace transform of the state equation,

$$sX(s) - x_0 = AX(s) + BU(s)$$

Solving for the state yields

$$X(s) = (sI - A)^{-1} x_0 + (sI - A)^{-1} BU(s)$$

Taking the inverse Laplace transform, we have that

$$x(t) = e^{At} x_0 + e^{At} * Bu(t)$$

which is the same as before.

Now that we have the solution to the state equation, the response is simply $y(t) = Cx(t) + Du(t)$.

The response of a continuous-time LTI system in the time domain is

$$y(t) = \underbrace{C e^{A(t-t_0)} x_0}_{\text{zero-input response}} + \underbrace{\int_{t_0}^t C e^{A(t-\tau)} B u(\tau) d\tau + D u(t)}_{\text{zero-state response}}$$

and in the frequency domain is

$$Y(s) = \underbrace{C (sI - A)^{-1} x_0}_{\text{zero-input response}} + \underbrace{[C (sI - A)^{-1} B + D] U(s)}_{\text{zero-state response}}$$

The transfer function is the ratio of the Laplace transform of the output to that of the input when the initial condition is zero, which for a general state space model is

$$\frac{Y(s)}{U(s)} = C (sI - A)^{-1} B + D$$

8.3 Solving the state equation for discrete-time LTI systems

We now show how to find the response from a discrete-time LTI state space model. Given the input signal and initial condition, we seek to find the output signal.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), & x(k_0) &= x_0 \\ y(k) &= Cx(k) + Du(k) \end{aligned}$$

In contrast to the continuous-time case, we can simply iterate the state equation in discrete time and observe the general pattern:

$$\begin{aligned}
 x(1) &= Ax_0 + Bu(0) \\
 x(2) &= A^2x_0 + ABu(0) + Bu(1) \\
 x(3) &= A^3x_0 + A^2Bu(0) + ABu(1) + Bu(2) \\
 &\vdots \\
 x(k) &= A^kx_0 + \sum_{\ell=0}^{k-1} A^{k-\ell-1} Bu(\ell)
 \end{aligned}$$

The response of a discrete-time LTI system in the time domain is

$$y(k) = \underbrace{CA^{k-k_0}x_0}_{\text{zero-input response}} + \underbrace{\sum_{\ell=k_0}^{k-1} CA^{k-\ell-1}Bu(\ell) + Du(k)}_{\text{zero-state response}}$$

and in the frequency domain is

$$Y(z) = \underbrace{zC(zI - A)^{-1}x_0}_{\text{zero-input response}} + \underbrace{[C(zI - A)^{-1}B + D]U(z)}_{\text{zero-state response}}$$

8.4 Diagonal form

Suppose A is diagonalizable. Then there exists an invertible matrix T and a diagonal matrix Λ such that

$$T^{-1}AT = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n).$$

Then applying the state transformation $x(t) = Tz(t)$ yields the state-space representation of the system

$$\begin{aligned}
 \dot{z} &= \Lambda z + \hat{B}u \\
 y &= \hat{C}z + \hat{D}u
 \end{aligned}
 \quad \text{where} \quad
 \left[\begin{array}{c|c} \Lambda & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array} \right] = \left[\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & D \end{array} \right].$$

Since Λ is diagonal, we can write this as a set of n uncoupled first-order differential equations:

$$\begin{aligned}
 \dot{z}_1 &= \lambda_1 z_1 + \hat{B}_1 u \\
 &\vdots \\
 \dot{z}_n &= \lambda_n z_n + \hat{B}_n u \\
 y &= \hat{C}z + \hat{D}u
 \end{aligned}$$

Since the states are uncoupled, we can solve for each state separately:

$$z_i(t) = e^{\lambda_i t} z_i(0) + \int_0^t e^{\lambda_i(t-\tau)} \hat{B}_i u(\tau) d\tau.$$

The output is then a weighted sum of each of these individual states along with the input signal.

Example. Consider the LTI system

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= [1 \quad 1] x\end{aligned}$$

Perform a state transformation with $x = Tz$ with state transformation matrix

$$T = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \quad \text{with inverse} \quad T^{-1} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

The transformed state-space matrices are

$$\begin{aligned}\hat{A} &= T^{-1}AT = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}, \\ \hat{B} &= T^{-1}B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \\ \hat{C} &= CT = [0 \quad 1].\end{aligned}$$

Since \hat{A} is diagonal, the transformed system is

$$\begin{aligned}\dot{z}_1 &= -z_1 + u \\ \dot{z}_2 &= -2z_2 + u \\ y &= z_2\end{aligned}$$

In this case, the first state z_1 does not affect the output, so the same input-output response is represented by the one-dimensional system

$$\begin{aligned}\dot{z}_2 &= -2z_2 + u \\ y &= z_2\end{aligned}$$

8.5 Impulse response

The *impulse response* of a system is its zero-state response due to a unit impulse input signal. Any zero-state response can be expressed as a weighted sum of shifted impulse responses.

Discrete time

The *impulse signal* in discrete time is

$$\delta(k) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}$$

which is always zero except at time $k = 0$. Recall that the general response of a discrete-time LTI system is

$$y(k) = CA^k x_0 + \sum_{\ell=0}^{k-1} CA^{k-\ell-1} Bu(\ell) + Du(k).$$

The impulse response, denoted $h(k)$, is the response when the input signal is an impulse ($u(k) = \delta(k)$) and the initial condition is zero ($x_0 = 0$),

$$h(k) = \sum_{\ell=0}^{k-1} CA^{k-\ell-1}B\delta(\ell) + D\delta(k) = \begin{cases} CA^{k-1}B & \text{if } k \geq 1 \\ D & \text{if } k = 0 \end{cases}$$

In other words, the input $\{1, 0, 0, 0, \dots\}$ produces the output $\{D, CB, CAB, CA^2B, \dots\}$.

Remark. If the system has multiple inputs and multiple outputs (MIMO), then the impulse is a matrix-valued signal with $\delta(0)$ the identity matrix. ■

8.6 Jordan form

The diagonal form only applies when the A matrix is diagonalizable. However, any matrix may be put in Jordan form. We now show how to interpret the response of the system using a state transformation so that \hat{A} is in Jordan form.

Let T be an invertible matrix that puts A in Jordan form, that is,

$$T^{-1}AT = J = \text{diag}(J_{k_1}(\lambda_1), \dots, J_{k_d}(\lambda_d)).$$

Then applying the state transformation $x(t) = Tz(t)$ yields the state-space representation of the system

$$\begin{aligned} \dot{z} &= Jz + \hat{B}u \\ y &= \hat{C}z + \hat{D}u \end{aligned} \quad \text{where} \quad \left[\begin{array}{c|c} J & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array} \right] = \left[\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & D \end{array} \right].$$

Since J is block diagonal with d blocks, we can write this as a set of d *uncoupled* differential equations:

$$\begin{aligned} \dot{z}_1 &= J_{k_1}(\lambda_1)z_1 + \hat{B}_1u \\ &\vdots \\ \dot{z}_d &= J_{k_d}(\lambda_d)z_d + \hat{B}_du \\ y &= \hat{C}z + \hat{D}u \end{aligned}$$

Since the states are uncoupled, we can solve for each state separately:

$$z_i(t) = e^{J_{k_i}(\lambda_i)t}z_i(0) + \int_0^t e^{J_{k_i}(\lambda_i)(t-\tau)}\hat{B}_i u(\tau) d\tau.$$

The output is then a weighted sum of each of these individual states along with the input signal.

9

Controllability

Controllability in dynamical systems refers to the ability to steer the system's state from any initial condition to any desired final state within a finite time interval using available inputs. In essence, it's about determining whether the system can be fully controlled.

9.1 Definition

A system is *controllable* if it can be moved from any state to any other state in a finite amount of time.

Definition (Controllability). A system is *controllable* if, given any initial state x_0 and final state x_f , there exists a time $t_f > 0$ and an input signal $u(t)$ for $0 \leq t \leq t_f$ such that $x(0) = x_0$ and $x(t_f) = x_f$.

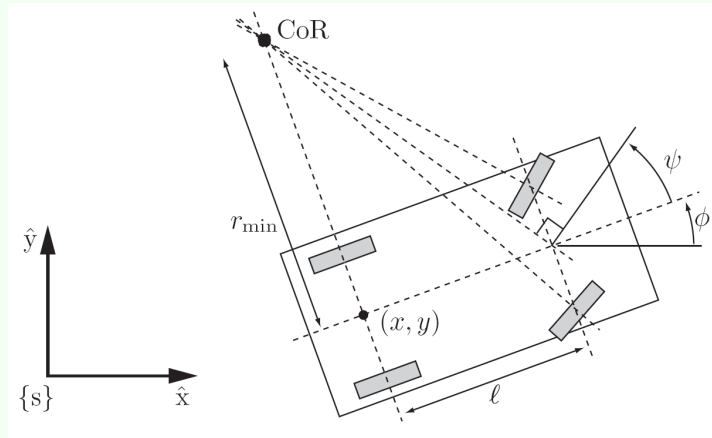
Comments

- Controllability does not involve the output of the system; it only has to do with the input's ability to control the *state*.
- For nonlinear systems, the notion of controllability depends on the particular starting and ending states. In that case, *controllability* typically refers to the input's ability to drive the state to the origin, while *reachability* is the input's ability to drive the system from the origin to another state. For linear systems, however, all notions of controllability and reachability are equivalent.

The main questions with regards to controllability of a system are the following:

- Is a system controllable?
- If a system is controllable, how do we drive the system from a given initial state to a given final state? How much time does it take? How much "energy" does it take?
- If a system is uncontrollable, what states can and cannot be controlled?

Example (Car-like dynamics). Consider the simplified car-like dynamics in the following diagram:



Suppose that the control inputs are the instantaneous velocities of the back wheels and the instantaneous angular velocity of the front wheels. Assuming that the wheels do not slip, the motion of the car must be parallel to the direction of the wheels. So is the car controllable, that is, can we move it into any position and orientation using appropriate inputs? The answer is yes. While we cannot move the car instantaneously in any given direction, we can maneuver the car through appropriate combinations of inputs. For instance, to move the car sideways (as in parallel parking), we can perform the sequence of commands:

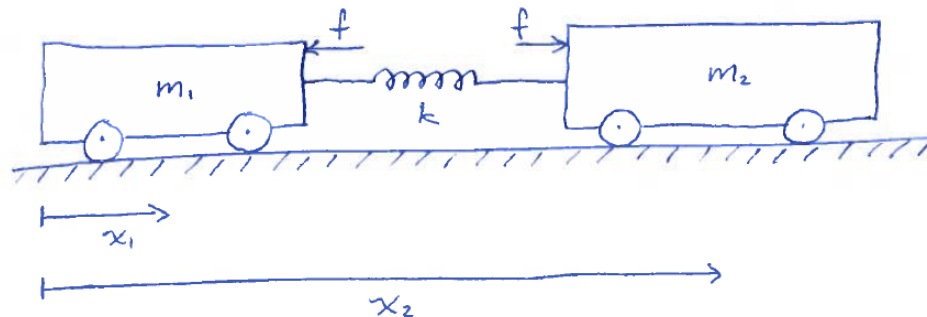
steer \rightarrow drive \rightarrow reverse steer \rightarrow reverse drive.

Doing so repetitively in small amounts moves the car sideways, enabling one to parallel park.

9.2 Examples of uncontrollable systems

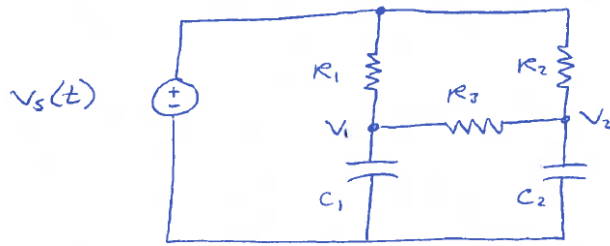
We now provide some examples of uncontrollable systems to understand where this situation can arise.

- *Physical system with no external forces*



The center of mass cannot be moved by an internal force.

- *Balanced bridge*



balanced condition:

$$R_1 C_1 = R_2 C_2$$

The voltage $v_1 - v_2$ is uncontrollable when balanced.

- Pole/zero cancellation

Uncontrollable states result in pole/zero cancellations in the transfer function (as we will see).

- Redundant state variables

If the system $\dot{x} = Ax + Bu$ has an additional state variable $z = Fx$, then the combined system is

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A & 0 \\ FA & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ FB \end{bmatrix} u$$

for which the transformed state $z - Fx \equiv 0$ is uncontrollable.

9.3 Analysis

We now describe how to analyze controllability. We do so for discrete-time systems, as the derivation is a little simpler than in continuous time.

Suppose we want to control a discrete-time LTI system from some initial state x_0 to some final state x_f in N time steps. From the solution to the state equations, the state at time N is

$$x(N) = A^N x(0) + \sum_{k=0}^{N-1} A^{N-k-1} B u(k).$$

Therefore, the relationship between the initial state x_0 , final state x_f , and control input u is

$$x_f - A^N x_0 = \begin{bmatrix} B & AB & A^2B & \dots & A^{N-1}B \end{bmatrix} \begin{bmatrix} u(N-1) \\ u(N-2) \\ \vdots \\ u(0) \end{bmatrix}.$$

This is linear system of equations. Recall the following fact from linear algebra concerning when a linear system of equations has a solution.

Fact. The linear system of equations $Ax = b$ has a solution x for all b if and only if A has full row rank.

Therefore, the system is controllable if and only if the matrix

$$\begin{bmatrix} B & AB & A^2B & \dots & A^{N-1}B \end{bmatrix}$$

has full row rank for some $N > 0$. But this is difficult to check, as we may need to make N arbitrarily large. The main result is that it suffices to take $N = n$. To show this, we need the following important result from linear algebra.

Theorem (Caley–Hamilton). Every square matrix satisfies its own characteristic equation.

Specifically, let $A \in \mathbb{R}^{n \times n}$ be a square matrix, and denote its characteristic polynomial as

$$p(\lambda) = \det(\lambda I - A) = \sum_{k=0}^n p_k \lambda^k$$

where p_0, p_1, \dots, p_n are the coefficients. The Cayley–Hamilton theorem then says that the matrix A satisfies this equation:

$$0 = p(A) = \sum_{k=0}^n p_k A^k.$$

Proof. The general proof of the Cayley–Hamilton theorem is quite complicated. Note that the following “proof” is incorrect:

$$p(A) = \det(AI - A) = \det(0) = 0. \quad (\text{incorrect!})$$

The left-hand side is a matrix while the right-hand side is a scalar, so the dimensions of this equation do not even make sense. The proof is quite straightforward when the matrix is diagonalizable, so let’s consider that case. Suppose $A = X\Lambda X^{-1}$ where Λ is diagonal. Then, the characteristic polynomial evaluated at the matrix A is

$$p(A) = \sum_{k=0}^n p_k A^k = X \left(\sum_{k=0}^n p_k \Lambda^k \right) X^{-1} = X \begin{bmatrix} \sum_k p_k \lambda_1^k & & \\ & \ddots & \\ & & \sum_k p_k \lambda_n^k \end{bmatrix} X^{-1} = 0$$

where the last equality is due to the fact that each eigenvalue λ_k satisfies $p(\lambda_k) = 0$ since eigenvalues are roots of the characteristic polynomial (by definition). Since not every matrix is diagonalizable, the general proof uses the Jordan form, which we omit. ■

A useful consequence of the Cayley–Hamilton theorem is that, if $A \in \mathbb{R}^{n \times n}$, then any power of A can be written as a polynomial of degree $n - 1$ in A . That is, there exist scalars $\alpha_i(k)$ such that

$$A^k = \sum_{i=0}^{n-1} \alpha_i(k) A^i.$$

To see why, note that the case $k = n$ follows directly from the Cayley–Hamilton theorem, and then we can use induction to show that if the result holds for k , it also holds for $k + 1$.

Therefore, for any vector b , the vector $A^k b$ is a linear combination of the set of vectors $\{b, Ab, \dots, A^{n-1}b\}$.

In particular, for a matrix $B = [b_1 \ b_2 \ \dots \ b_m]$,

$$\text{rank} [B \ AB \ A^2B \ \dots \ A^k B] = \text{rank} [B \ AB \ A^2B \ \dots \ A^{n-1}B].$$

So when testing for controllability, there is no need to use $k \geq n$, as making the matrix larger cannot increase the rank. We summarize this result as follows.

Definition (Controllability matrix). The controllability matrix associated with a pair (A, B) with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ is the $n \times nm$ matrix

$$P = [B \ AB \ A^2B \ \dots \ A^{n-1}B].$$

■

Fact (Discrete-time controllability). The pair (A, B) is controllable if and only if the controllability matrix has full row rank (that is, rank n).

While the controllability matrix P depends on the state-space matrices A and B , whether or not a system is controllable is independent of the particular state-space realization due to the following result.

Fact. The controllability matrix transforms under a state transformation $\hat{x} = T^{-1}x$ as $\hat{P} = T^{-1}P$.

Proof. The controllability matrix of the transformed system is

$$\begin{aligned}\hat{P} &= [\hat{B} \quad \hat{A}\hat{B} \quad \dots \quad \hat{A}^{n-1}\hat{B}], \\ &= [T^{-1}B \quad (T^{-1}AT)(T^{-1}B) \quad \dots \quad (T^{-1}AT)^{n-1}(T^{-1}B)], \\ &= [T^{-1}B \quad T^{-1}AB \quad \dots \quad T^{-1}A^{n-1}B], \\ &= T^{-1}P.\end{aligned}$$

Fact. The rank of the controllability matrix is invariant under state transformations.

Proof. The previous result implies that $\text{rank}(\hat{P}) = \text{rank}(T^{-1}P) = \text{rank}(P)$ since T is invertible. ■

9.4 Controllable subspace

Based on our analysis of controllability, we can find the input that drives the system between two given states as follows.

Fact. For a discrete-time LTI system, the input signal that drives the system from an initial state x_0 to final state x_f is the solution to the linear system of equations

$$x_f - A^n x_0 = P \begin{bmatrix} u(n-1) \\ \vdots \\ u(0) \end{bmatrix}$$

where P is the controllability matrix.

The system is controllable if and only if P has full row rank, in which case this linear system of equations has a solution u for any initial state x_0 and final state x_f . When the system is *not* controllable, however, we can still find an input that drives the system between some states (but not all states). We now characterize which states the input is able to control. To do so, recall the following result from linear algebra.

Fact (Existence of solution to linear system of equations). The linear system of equations $Ax = b$ has a solution x if and only if b is in the column space of A .

The controllable subspace is defined as the set of states that can be reached from the origin by suitable choice of input. From the above result, we have the following.

Fact (Controllable subspace). The controllable subspace is the column space of the controllability matrix:

$$\mathcal{C} = \text{col}(P).$$

This set of states forms a subspace in that it is closed under linear combinations, meaning that if x_1 and x_2 are in the controllable subspace, then $ax_1 + bx_2$ is also in the controllable subspace for any scalars a and b .

Using the controllable subspace, we can define a more refined notion of controllability for each state.

Definition (Controllability of a state). A state is controllable iff it is in the controllable subspace:

$$x \text{ is controllable} \iff x \in \mathcal{C}.$$

These definitions are consistent in that the following are equivalent:

- (A, B) is controllable
- P has full column rank
- the controllable subspace is all of \mathbb{R}^n
- all states are controllable

Example. Consider the LTI system with state-space matrices

$$A = \begin{pmatrix} 1 & 5 \\ 8 & 4 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -2 \\ 2 \end{pmatrix}.$$

The controllability matrix is

$$P = [B \quad AB] = \begin{bmatrix} -2 & 8 \\ 2 & -8 \end{bmatrix}$$

which has rank one since the second column is a multiple of the first column. The controllable subspace is

$$\mathcal{C} = \text{col}(P) = \text{span} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mid x_2 = -x_1 \right\}.$$

For instance, to drive the system from the origin to the final state $x_f = (10, -10)$, we find the appropriate control input as

$$x_f = Pu \implies \begin{bmatrix} 10 \\ -10 \end{bmatrix} = \begin{bmatrix} -2 & 8 \\ 2 & -8 \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \end{bmatrix} \implies \begin{bmatrix} u(1) \\ u(0) \end{bmatrix} = \begin{bmatrix} -0.2941 \\ 1.1765 \end{bmatrix}.$$

Iterating the system with the input from the origin, we find that the $x(2) = x_f$ as desired.

Controllable decomposition

We can make the controllable subspace explicit in the state-space realization using an appropriate state transformation.

Suppose (A, B) is not controllable. Then $q = \text{rank}(P) < n$, so the controllability matrix P has q linearly independent columns, say $t_1, \dots, t_q \in \mathbb{R}^n$. These columns form a basis for the controllable subspace,

$$\mathcal{C} = \text{span}(t_1, \dots, t_q).$$

To form a basis for the orthogonal complement of the controllable subspace, we can choose the vectors

$t_{q+1}, \dots, t_n \in \mathbb{R}^n$ such that the following state transformation matrix is invertible:

$$T = [T_1 \quad T_2] \in \mathbb{R}^{n \times n}$$

where

$$T_1 = [t_1 \quad \dots \quad t_q] \in \mathbb{R}^{n \times q} \quad (\text{basis for } \mathcal{C})$$

$$T_2 = [t_{q+1} \quad \dots \quad t_n] \in \mathbb{R}^{n \times (q-n)} \quad (\text{basis for } \mathcal{C}^\perp)$$

Our claim is that, using this T to perform a state transformation, the state equations separate the controllable states. To see this, we first need the following result.

Definition (Invariant subspace). A subspace S is A -invariant if $AS \subseteq S$, where $AS = \{Ax \mid x \in S\}$.

Fact. The controllable subspace is A -invariant.

Proof. The result follows from

$$AC = \{Ax \mid x \in \text{col}(P)\} = \{Ax \mid x = Pz\} = \{APz\} = \{[AB \quad A^2B \quad \dots \quad A^n B] z\} \subseteq \mathcal{C}$$

where the subset inclusion follows from the Cayley–Hamilton theorem. ■

Fact. Suppose a subspace S is A -invariant, and define T_S to have columns that span S and let $T_{\bar{S}}$ complete this basis so that $T = [T_S \quad T_{\bar{S}}]$ is square and invertible. Then, there exist A_{11} , A_{12} , and A_{22} such that

$$T^{-1}AT = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}.$$

Proof. Since the columns of T_S are a basis for S , for any x we have that $T_S x \in S$. Then from A -invariance of S , we have that $AT_S x \in S$. Therefore, $AT_S = T_S A_{11}$ for some A_{11} . Then,

$$AT = A [T_S \quad T_{\bar{S}}] = [T_S \quad T_{\bar{S}}] \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}. \quad \blacksquare$$

Since the controllable subspace is A -invariant, the transformed A matrix is

$$\hat{A} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}.$$

Moreover, since the column space of B is contained in the controllable subspace, there exists a matrix \hat{B}_1 such that

$$B = [T_1 \quad T_2] \begin{bmatrix} \hat{B}_1 \\ 0 \end{bmatrix}, \quad \text{which implies that} \quad \hat{B} = \begin{bmatrix} \hat{B}_1 \\ 0 \end{bmatrix}.$$

Therefore, the transformed state-space matrices have the structure

$$\hat{A} = T^{-1}AT = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ 0 & \hat{A}_{22} \end{bmatrix} \quad \text{and} \quad \hat{B} = T^{-1}B = \begin{bmatrix} \hat{B}_1 \\ 0 \end{bmatrix}.$$

The transformed dynamics are

$$\begin{aligned} z_1(k+1) &= \hat{A}_{11}z_1(k) + \hat{A}_{12}z_2(k) + \hat{B}_1u(k) \\ z_2(k+1) &= \hat{A}_{22}z_2(k) \end{aligned}$$

Note that the state z_2 is not affected by the input at all and is therefore uncontrollable. The value of the second state is simply

$$z_2(k) = \hat{A}_{22}^k z_2(0)$$

which only depends on the initial condition. If we include the output (which has no particular structure under the state transformation), then

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \xrightarrow{T} \left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array} \right] = \left[\begin{array}{cc|c} \hat{A}_{11} & \hat{A}_{12} & \hat{B}_1 \\ 0 & \hat{A}_{22} & 0 \\ \hline \hat{C}_1 & \hat{C}_2 & \hat{D} \end{array} \right].$$

The transfer function of the transformed system is

$$\begin{aligned} \hat{H}(s) &= [\hat{C}_1 \quad \hat{C}_2] \left(sI - \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ 0 & \hat{A}_{22} \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{B}_1 \\ 0 \end{bmatrix} + \hat{D} \\ &= [\hat{C}_1 \quad \hat{C}_2] \left(\begin{bmatrix} sI - \hat{A}_{11} & -\hat{A}_{12} \\ 0 & sI - \hat{A}_{22} \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{B}_1 \\ 0 \end{bmatrix} + \hat{D} \\ &= [\hat{C}_1 \quad \hat{C}_2] \left(\begin{bmatrix} (sI - \hat{A}_{11})^{-1} & (sI - \hat{A}_{11})^{-1} \hat{A}_{12} (sI - \hat{A}_{22})^{-1} \\ 0 & (sI - \hat{A}_{22})^{-1} \end{bmatrix} \right) \begin{bmatrix} \hat{B}_1 \\ 0 \end{bmatrix} + \hat{D} \\ &= \hat{C}_1 (sI - \hat{A}_{11})^{-1} \hat{B}_1 + \hat{D} \end{aligned}$$

which is the same as the transfer function for the reduced system without the second state,

$$\left[\begin{array}{c|c} \hat{A}_{11} & \hat{B}_1 \\ \hline \hat{C}_1 & \hat{D} \end{array} \right].$$

This is because the transfer function assumes that the state is zero, so $z_2(k) = 0$ for all k . So we do not change the transfer function by removing this (uncontrollable) state.

Example. Consider the LTI system with state-space matrices

$$A = \begin{pmatrix} 1 & 5 \\ 8 & 4 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -2 \\ 2 \end{pmatrix}.$$

The controllability matrix is

$$P = [B \quad AB] = \begin{bmatrix} -2 & 8 \\ 2 & -8 \end{bmatrix}$$

which has rank one since the second column is a multiple of the first column. A basis for the columns of P is $\{t_1\}$ where

$$t_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

To transform the state to separate the controllable subspace, we need to complete this basis. We can do this in many ways. Let's use

$$t_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

so that the transformation matrix and its inverse is

$$T = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad T^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

The transformed state-space matrices are

$$\hat{A} = \begin{bmatrix} -4 & 5 \\ 0 & 9 \end{bmatrix} \quad \text{and} \quad \hat{B} = \begin{bmatrix} -2 \\ 0 \end{bmatrix}.$$

So under the transformation $x = Tz$, the transformed dynamics are

$$\begin{aligned} z_1(k+1) &= -4z_1(k) + 5z_2(k) - 2u(k) \\ z_2(k+1) &= 9z_2(k) \end{aligned}$$

If $z_2(0) = 0$, then $z_2(k) = 0$ for all k and the system simplifies to

$$z_1(k+1) = -4z_1(k) - 2u(k).$$

9.5 Minimum norm input signal in continuous time

If (A, B) is controllable, then the system can be driven from any initial state to any final state. For discrete-time systems this can be done in n iterations, while it can be done in an arbitrary amount of time for continuous-time systems. We now describe the “smallest” input signal that drives the system between two states in a given amount of time.

Definition (Controllability Gramian). The controllability Gramian is the matrix-valued function of time

$$W_c(t) = \int_0^t \exp(A\tau) B B^\top \exp(A^\top \tau) d\tau$$

in continuous time and

$$W_c(k) = \sum_{m=0}^{k-1} A^m B B^\top (A^\top)^m$$

in discrete time.

Fact (Minimum norm input in continuous time). A control input u that drives the system (A, B) from initial state x_0 to final state x_f in time $t_f > 0$ is

$$u(t) = -B^\top \exp(A^\top(t_f - t)) W_c(t_f)^{-1} (\exp(At_f)x_0 - x_f)$$

where $W_c(t)$ is the controllability Gramian. Moreover, this input signal uses the minimal amount of energy to do so, and the energy used by this control law is

$$\int_0^{t_f} \|u(t)\|^2 dt = (\exp(At_f)x_0 - x_f)^\top W_c(t_f)^{-1} (\exp(At_f)x_0 - x_f).$$

Remark. We can use the characterization of the minimum norm input signal to characterize which states are “easy” to control and what states are “hard” to control. Suppose we choose the initial state as the origin ($x_0 = 0$) and the final state as an eigenvector of the controllability Gramian with eigenvalue λ , that is,

$$W_c(t_f)x_f = \lambda x_f \quad \text{which implies} \quad W_c(t_f)^{-1}x_f = \frac{1}{\lambda}x_f.$$

The minimal input energy is then

$$\int_0^{t_f} \|u(t)\|^2 dt = \frac{\|x_f\|^2}{\lambda}.$$

The state can be driven to x_f with small energy when the eigenvalue λ is large, while the input has large energy when λ is small. In the limit as $\lambda \rightarrow 0$, the energy of the input goes to infinity and the system becomes uncontrollable. ■

Proof. Since the system is controllable, the controllability Gramian is positive definite and therefore invertible, so the input signal is well-defined. From the solution of the state equation, the state at the final time is

$$x(t_f) = \exp(At_f)x_0 + \int_0^{t_f} \exp(A(t_f - \tau)) B u(\tau) d\tau.$$

Substituting the expression for the controlled input,

$$x(t_f) = \exp(At_f)x_0 + \int_0^{t_f} \exp(A(t_f - \tau)) B \left(-B^\top \exp(A^\top(t_f - \tau)) W_c(t_f)^{-1} (\exp(At_f)x_0 - x_f) \right) d\tau.$$

Since the last few terms inside the integral do not depend on the integrand, we can move them outside the integral to obtain

$$x(t_f) = \exp(At_f)x_0 - \left(\int_0^{t_f} \exp(A(t_f - \tau)) B B^\top \exp(A^\top(t_f - \tau)) d\tau \right) W_c(t_f)^{-1} (\exp(At_f)x_0 - x_f).$$

The expression inside the large parentheses is the controllability Gramian $W_c(t_f)$, so it cancels with its inverse to obtain

$$x(t_f) = \exp(At_f)x_0 - (\exp(At_f)x_0 - x_f) = x_f.$$

Therefore, the controlled input drives the state to x_f at time t_f as desired. To compute the energy, the integral of the squared-norm of the controlled input signal is

$$\int_0^{t_f} (\exp(At_f)x_0 - x_f)^\top W_c(t_f)^{-1} \exp(A(t_f - t)) B B^\top \exp(A^\top(t_f - t)) W_c(t_f)^{-1} (\exp(At_f)x_0 - x_f) dt.$$

Moving the integral inside only the terms that depend on t ,

$$(\exp(At_f)x_0 - x_f)^\top W_c(t_f)^{-1} \left(\int_0^{t_f} \exp(A(t_f - t)) B B^\top \exp(A^\top(t_f - t)) dt \right) W_c(t_f)^{-1} (\exp(At_f)x_0 - x_f).$$

The term inside the large parentheses is the controllability Gramian at time t_f , which cancels with its inverse. Therefore, the energy of the input simplifies to

$$(\exp(At_f)x_0 - x_f)^\top W_c(t_f)^{-1} (\exp(At_f)x_0 - x_f)$$

which proves the result. ■

9.6 Controllable canonical form

Given a proper transfer function

$$H(s) = d + \frac{b_{n-1}s^{n-1} + \dots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0}$$

the *controllable canonical form* (CCF) is the state-space realization

$$\left(\begin{array}{c|c} A_{CCF} & B_{CCF} \\ \hline C_{CCF} & D_{CCF} \end{array} \right) = \left(\begin{array}{cccccc|c} 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ \hline -a_0 & -a_1 & -a_2 & \dots & -a_{n-2} & -a_{n-1} & 1 \\ \hline b_0 & b_1 & b_2 & \dots & b_{n-2} & b_{n-1} & d \end{array} \right)$$

Fact. Any state-space representation in controllable canonical form is controllable.

In fact, the controllability matrix is always invertible, and its inverse has the simple expression

$$P_{CCF}^{-1} = \begin{bmatrix} a_1 & a_2 & \dots & a_{n-1} & 1 \\ a_2 & a_3 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}.$$

9.7 Characterizations of controllability

While analyzing the rank of the controllability matrix is one way to determine controllability, there are many equivalent characterizations. The following result summarizes many of the equivalent characterizations of controllability (some of which may not make sense yet).

Theorem (Controllability). Given matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, the following statements are equivalent.

- (A, B) is controllable.
- (B^T, A^T) is observable.
- The controllability matrix has full row rank.
- The controllability Gramian is positive definite for all positive times.
- The matrix $[A - \lambda I \quad B]$ has full row rank for all $\lambda \in \mathbb{C}$.
- If $x \in \mathbb{C}^n$ is a left-eigenvector of A with eigenvalue $\lambda \in \mathbb{C}$ (that is, $x^*A = x^*\lambda$), then $x^*B \neq 0$.
- The eigenvalues of $A + BK$ may be arbitrarily assigned by suitable choice of K (as long as they are chosen in complex conjugate pairs).
- The controllable subspace is \mathbb{R}^n .

Testing controllability using the condition that $[A - \lambda I \quad B]$ has full row rank for all λ is known as the Popov–Belevitch–Hautus (PBH) test. Also, note that we actually only need to check the eigenvalues of A , because the matrix always has full row rank when λ is not an eigenvalue.

Proof. See handwritten notes. ■

9.8 Stabilizability

A system is *stabilizable* if all of its unstable modes are controllable (but the stable modes need not be controllable). In this case, we can construct a state-feedback controller such that the system is stable; we can assign the closed-loop eigenvalues of the controllable modes, but the uncontrollable modes cannot be changed.

Theorem (Stabilizability). Given matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, the following statements are equivalent.

- (A, B) is stabilizable
- (B^T, A^T) is detectable
- There exists a matrix K such that $A + BK$ is stable.

10

Observability

A system is *observable* if the initial state can be uniquely determined by observing the output for a sufficiently long time. Observability is dual to controllability, so we can use the tools for one to study the other.

10.1 Definition

Definition (Observability). A system is *observable* if, for some time $t_f > 0$ sufficiently large, the input signal $u(t)$ and output signal $y(t)$ for $t \in [0, t_f]$ uniquely determine the initial state $x(0)$.

Remark. Observability does not involve the input to the system; it only has to do with the output's ability to observe the *state*. ■

Example. Consider the position of a mass subject to some applied force. If we can only measure the velocity, then it is impossible to uniquely determine the initial position of the object (we can only infer relative positions).

The main questions with regards to observability of a system are the following:

- Is a system observable?
- If a system is observable, how do we determine the initial state from the observed output? How much time does it take?
- If a system is unobservable, what states can and cannot be observed?

10.2 Derivation of main result

To study observability, consider a discrete-time LTI system

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k), \\y(k) &= Cx(k) + Du(k).\end{aligned}$$

The response at each time k is given by

$$y(k) = CA^k x(0) + \sum_{\ell=0}^{k-1} CA^{k-\ell-1} Bu(\ell) + Du(k).$$

Suppose that the input signal $u(k)$ and output signal $y(k)$ are known for all times $k = 0, 1, \dots, N$ for some $N > 0$. Then we also know the quantity

$$\tilde{y}(k) = y(k) - \sum_{\ell=0}^{k-1} CA^{k-\ell-1}Bu(\ell) - Du(k) = CA^kx(0).$$

To find the initial state, we need to solve the system of linear equations

$$\tilde{y}(k) = CA^kx(0), \quad k = 0, 1, \dots, N$$

for the initial state $x(0)$. We can write this system of equations in matrix form as

$$\begin{bmatrix} \tilde{y}(0) \\ \tilde{y}(1) \\ \vdots \\ \tilde{y}(N) \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^N \end{bmatrix}}_{Q_N} x(0).$$

This system of equations has a solution $x(0)$ for all \tilde{y} if and only if Q_N has full column rank. (Otherwise, there is some linear combinations of columns that is zero, so $Q_N v = 0$ for some v , so $x(0) + \alpha v$ is a solution for all scalars α .) If Q_N has full column rank (that is, rank n), then the initial state is

$$x(0) = Q_N^+ \tilde{y}$$

where $Q_N^+ = (Q_N^T Q_N)^{-1} Q_N^T$ is the pseudo inverse.

The rank of Q_N may in general depend on the time horizon N . From the Cayley–Hamilton theorem, however, it suffices to take $N = n - 1$. One way to see this is to convert the observability problem to a controllability problem. Notice that the transpose

$$Q_N^T = [C^T \quad A^T C^T \quad \dots \quad (A^N)^T C^T] = P_N$$

is the controllability matrix of (A^T, C^T) . We then have the following equivalent statements:

- The pair (C, A) is observable.
- The matrix Q_N has rank n for some $N > 0$.
- The matrix P_N has rank n for some $N > 0$.
- The controllability matrix P has rank n .
- The pair (A^T, C^T) is controllable.

We can then directly apply our characterization of controllability to study observability.

Definition (Observability matrix). The observability matrix is

$$Q = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}.$$

Fact (Observability). The pair (C, A) is observable if and only if the observability matrix has full column rank.

10.3 Unobservable subspace

The unobservable subspace is the set of states that cannot be observed by observing the output. The unobservable subspace is the nullspace of the observability matrix, which is the subspace $\ker(Q)$. We say that a *state* is unobservable if it is in the unobservable subspace, that is, the state x is unobservable if $x \in \ker(Q)$. A system (A, C) is observable if and only no states are unobservable, in which case the unobservable subspace is the empty set in \mathbb{R}^n .

Fact (Unobservable subspace). The set of unobservable states is the nullspace of the observability matrix:

$$\mathcal{O} = \text{null}(Q).$$

Definition (Observable state). A state is unobservable if and only if it is in the unobservable subspace:

$$x \text{ is unobservable} \iff x \in \mathcal{O}.$$

Remark. The controllable states form a subspace, meaning that if x_1 and x_2 are controllable, then so is $ax_1 + bx_2$. Likewise, the unobservable states form a subspace. The uncontrollable states and observable states, however, do *not* form subspaces since they are not closed under linear combinations. ■

10.4 Reconstructing the initial state

To reconstruct the initial state, define the signal

$$\tilde{y}(k) = y(k) - \sum_{\ell=0}^{k-1} CA^{k-\ell-1}Bu(\ell) - Du(k)$$

and then solve the linear system of equations

$$\begin{bmatrix} \tilde{y}(0) \\ \tilde{y}(1) \\ \vdots \\ \tilde{y}(n-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} x(0)$$

for the initial state $x(0)$. When the input signal is zero, $\tilde{y}(k)$ reduces to $y(k)$.

10.5 Observable canonical form

Given a proper transfer function

$$H(s) = d + \frac{b_{n-1}s^{n-1} + \dots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0}$$

the *observable canonical form* (OCF) is the state-space realization

$$\left(\begin{array}{c|c} \frac{A_{\text{OCF}}}{C_{\text{OCF}}} & \frac{B_{\text{OCF}}}{D_{\text{OCF}}} \end{array} \right) = \left(\begin{array}{cccccc|c} 0 & 0 & \dots & 0 & 0 & -a_0 & b_0 \\ 1 & 0 & \dots & 0 & 0 & -a_1 & b_1 \\ 0 & 1 & \dots & 0 & 0 & -a_2 & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & -a_{n-2} & b_{n-2} \\ 0 & 0 & \dots & 0 & 1 & -a_{n-1} & b_{n-1} \\ \hline 0 & 0 & \dots & 0 & 0 & 1 & 0 \end{array} \right)$$

The observable canonical form is always observable. In fact, the observable matrix is always invertible, and its inverse has the simple expression

$$Q_{\text{OCF}}^{-1} = \begin{bmatrix} a_1 & a_2 & \dots & a_{n-1} & 1 \\ a_2 & a_3 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}$$

10.6 Observable decomposition

When the system is not observable, we can apply a state transformation to separate the unobservable states.

Suppose $\text{rank}(Q) = q < n$. Then (C, A) is not observable, so (A^T, C^T) is not controllable. Moreover, the controllability matrix P of this pair is the transpose of the observability matrix for the original system, that is, $P = Q^T$. Define the state transformation matrix

$$T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

where the rows of T_1 form a basis for the row space of the observability matrix Q , and the remaining rows are chosen such that T is invertible. Then, the state-space matrices under this state transformation have the form

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \xrightarrow{T} \left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array} \right] = \left[\begin{array}{cc|c} \hat{A}_{11} & 0 & \hat{B}_1 \\ \hat{A}_{21} & \hat{A}_{22} & \hat{B}_2 \\ \hline \hat{C}_1 & 0 & \hat{D} \end{array} \right].$$

In these new coordinates,

$$\begin{aligned} z_1(k+1) &= \hat{A}_{11}z_1(k) + \hat{B}_1u(k) \\ z_2(k+1) &= \hat{A}_{21}z_1(k) + \hat{A}_{22}z_2(k) + \hat{B}_2u(k) \\ y(k) &= \hat{C}_1z_1(k) + \hat{D}u(k) \end{aligned}$$

The state z_2 cannot be observed since the output only depends on z_1 and u , and z_1 only depends on z_1 and u . As before, the transfer function is

$$C(sI - A)^{-1}B + D = \hat{C}_1(sI - \hat{A}_{11})^{-1}\hat{B}_1 + \hat{D}$$

Example (Position vs velocity). Consider applying a force f to a rigid object with mass m , position p , and velocity $v = \dot{p}$. From Newton's laws, the equations of motion are $m\dot{v} = f$. In state-space form,

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f.$$

We will analyze the observability of this system under two different measurements.

- First, suppose we can measure the position. Then the output is

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix}.$$

The observability matrix for this output is

$$Q = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

which has full column rank, so the system is observable. Knowing the position and applied force is sufficient to uniquely determine the initial position and velocity.

- Now suppose we can measure the velocity. Then the output is

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix}.$$

The observability matrix for this output is

$$Q = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

which does not have full column rank, so the system is unobservable. The observable subspace is the null space of the observability matrix,

$$\text{null}(Q) = \text{span} \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The position is unobservable, as this cannot be uniquely determined from the velocity measurements and the applied force.

10.7 Characterizations of observability

Theorem (Observability). Given matrices $A \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{p \times n}$, the following statements are equivalent.

- (C, A) is observable.
- (A^T, C^T) is controllable.
- The observability matrix has full column rank.
- The continuous-time observability Gramian

$$W_o(t) = \int_0^t \exp(A^T \tau) C^T C \exp(A \tau) d\tau$$

is positive definite for all times $t > 0$.

- The discrete-time observability Gramian

$$W_o(k) = \sum_{m=0}^{k-1} (A^T)^m C^T C A^m$$

is positive definite for all times $k > 0$.

- The matrix $\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}$ has full column rank for all $\lambda \in \mathbb{C}$.
- If $x \in \mathbb{C}^n$ is a right-eigenvector of A with eigenvalue $\lambda \in \mathbb{C}$ (that is, $Ax = \lambda x$), then $Cx \neq 0$.
- The eigenvalues of $A + LC$ may be arbitrarily assigned by suitable choice of L (as long as they are chosen in complex conjugate pairs).
- The unobservable subspace is $\{0\}$.

Comments

- The condition that the matrix $\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}$ has full column rank for all λ is known as the PBH test for observability.

10.8 Detectability

A system is *detectable* if all of its unstable modes are observable (but the stable modes need not be observable). In this case, we can construct an observer that asymptotically estimates the state; we can assign the closed-loop eigenvalues of the observable modes, but the unobservable modes cannot be changed.

Theorem (Detectability). Given matrices $A \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{p \times n}$, the following statements are equivalent.

- (C, A) is detectable
- (A^T, C^T) is stabilizable
- There exists a matrix K such that $A + LC$ is stable.

11

Minimality

Minimality refers to a state-space realization have the smallest possible number of states that are needed to accurately represent the input-output behavior of a dynamical system. In this chapter, we study how minimality of a state-space realization is related to pole/zero cancellations of the transfer function, controllability, and observability.

11.1 Overview

There are two main ways of representing an LTI system.

- **Input-output representation.** The transfer function is an input-output representation of a system that describes how the input and output signals are related, but has no notion of a state.
- **Internal representation.** The state-space representation is an internal representation that describes how the input signal influences the state, and how the output signal depends on the state.

Recall that the transfer function and state-space realization are related as

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \quad \text{has transfer function} \quad H(s) = C(sI - A)^{-1}B + D.$$

Moreover, the transfer function of the system is unique while the state-space realization is not, meaning that there are many internal representations that have the same input-output representation. We now discuss the relationship between these various representations. To do so, we first define irreducibility for a transfer function and minimality of a state-space realization.

Definition (Irreducible). A SISO transfer function is *irreducible* if it has no pole-zero cancellations.

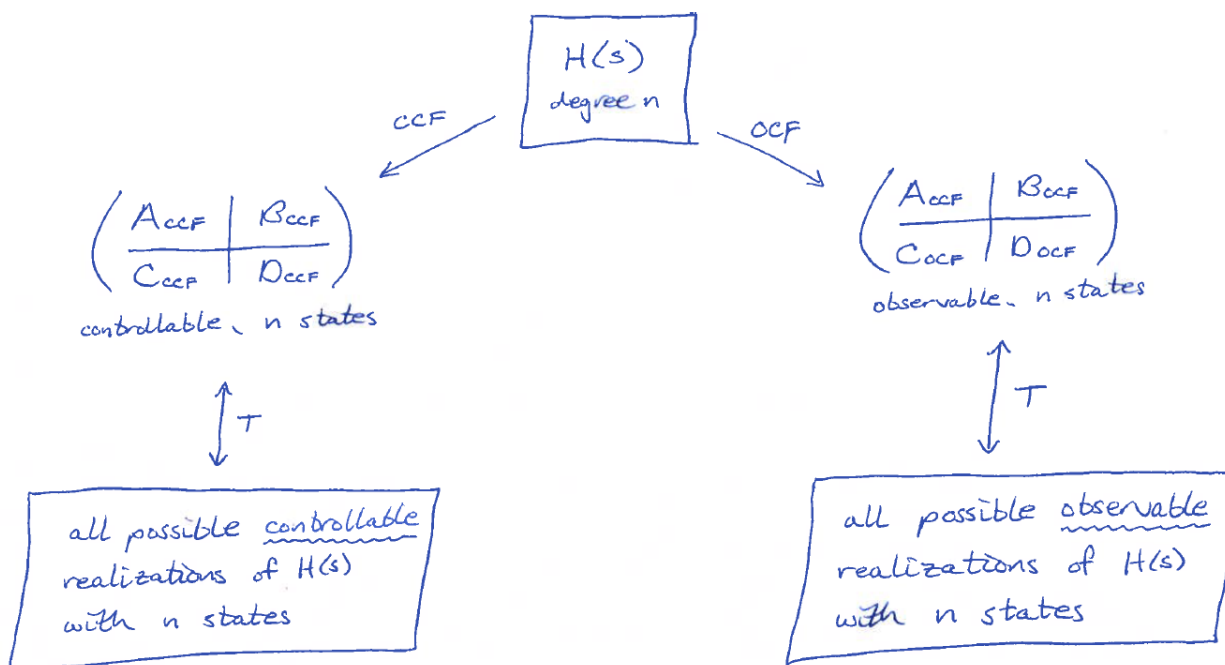
Definition (Minimal). A realization is *minimal* if there exists no other realization with the same transfer function and fewer states.

In other words, a minimal realization uses the smallest internal representation necessary to represent the input-output relationship of a given transfer function. The following result summarizes the relationship between these two concepts.

Fact. Consider a transfer function $H(s)$ with state-space realization (A, B, C, D) , and suppose the denominator of $H(s)$ has degree n and A has dimensions $n \times n$. Then the following are equivalent:

- the transfer function $H(s)$ is irreducible
- the state-space realization (A, B, C, D) is minimal
- (A, B) is controllable and (C, A) is observable
- all n -state realizations of $H(s)$ are controllable and observable
- the CCF of $H(s)$ is observable
- the OCF of $H(s)$ is controllable

The following figure illustrates this result. For any transfer function $H(s)$ of degree n , we can construct both its controllable canonical form (CCF) and observable canonical form (OCF). Applying a state transformation to each of these realizations yields *all* possible controllable and observable realizations, respectively, of the transfer function with n states. If the transfer function is irreducible, then all n state realizations are minimal (that is, both controllable and observable), so any state transformation applied to the CCF or OCF yields a minimal state-space realization.



Example (Minimality). Consider the transfer function

$$H(s) = \frac{s^2 + 2s + 1}{s^3 + 6s^2 + 11s + 6}.$$

The denominator has degree three. The controllable canonical form is

$$\left[\begin{array}{ccc|c} A_{CCF} & B_{CCF} & & \\ \hline C_{CCF} & D_{CCF} & & \end{array} \right] = \left[\begin{array}{ccc|c} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -6 & -11 & -6 & 1 \\ \hline 1 & 2 & 1 & 0 \end{array} \right]$$

and the observable canonical form is

$$\left[\begin{array}{ccc|c} A_{OCF} & B_{OCF} & & \\ \hline C_{OCF} & D_{OCF} & & \end{array} \right] = \left[\begin{array}{ccc|c} 0 & 0 & -6 & 1 \\ 1 & 0 & -11 & 2 \\ 0 & 1 & -6 & 1 \\ \hline 0 & 0 & 1 & 0 \end{array} \right].$$

The CCF is not observable since its observability matrix is

$$Q_{CCF} = \begin{bmatrix} 1 & 2 & 1 \\ -6 & -10 & -4 \\ 24 & 38 & 14 \end{bmatrix}$$

is singular. Similarly, the OCF is not ocontrollable since $P_{OCF} = Q_{CCF}^T$. However, the transfer function has a pole-zero cancellation:

$$H(s) = \frac{(s+1)^2}{(s+1)(s+2)(s+3)} = \frac{s+1}{(s+2)(s+3)}.$$

The reduced system has CCF

$$\left[\begin{array}{ccc|c} 0 & 1 & 0 & \\ \hline -6 & -5 & 1 & \\ 1 & 1 & 0 & \end{array} \right] \quad \text{with observability matrix} \quad Q = \begin{bmatrix} 1 & 1 \\ -6 & -4 \end{bmatrix}.$$

The observability matrix has full rank, so the realization is both controllable and observable (and therefore minimal). Note that the minimal realization has the same number of states as the number of poles of the transfer function after any cancellations.

11.2 Proofs

We now prove the main result connecting minimal realizations, irreducible transfer functions, controllability, and observability.

Fact. The transfer function is irreducible iff the CCF is observable.

Proof. The CCF is observable iff, for any $x \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}$ that satisfy $A_{CCF}x = \lambda x$, we have that $C_{CCF}x = 0$. This implies that $x_2 = \lambda x_1, \dots, x_n = \lambda x_{n-1}$, and $a_0 x_1 + \dots + a_{n-1} x_n + \lambda_n x_n = 0$. Substituting,

we obtain $x_1(a_0 + \lambda a_1 + \dots + \lambda^{n-1}a_{n-1}) = 0$. Also, $x_1 \neq 0$ since this would make $x = 0$. Therefore,

$$C_{\text{CCF}}x \neq 0 \quad \iff \quad x_1(b_0 + b_1\lambda + \dots + b_{n-1}\lambda^{n-1}) \neq 0$$

λ cannot be a root of both $a(s)$ and $b(s)$, so $H(s)$ is irreducible. ■

Similarly, $H(s)$ is irreducible iff the OCF is controllable.

Fact. If there is an n -state realization of $H(s)$ that is controllable and observable, then so are *all* n -state realizations.

Proof. For any realizations (A, B, C, D) and $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$,

$$CA^k B = \hat{C}\hat{A}^k\hat{B} \quad \text{for all } k$$

since they have the same impulse response. Then they also have the same Hankel matrix

$$\mathcal{H} = \begin{bmatrix} CB & CAB & \dots & CA^{n-1}B \\ CAB & CA^2B & \dots & CA^nB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{n-1}B & CA^nB & \dots & CA^{2n-2}B \end{bmatrix} = QP.$$

That is, $\hat{Q}\hat{P} = \hat{\mathcal{H}} = \mathcal{H} = QP$. So Q and P are both invertible iff \hat{Q} and \hat{P} are both invertible. ■

Therefore, a realization is minimal iff it is both controllable and observable.

12

Stability

Stability is a fundamental concept in the study of dynamical systems that describes the behavior of the system trajectories over time. A system is said to be stable if its trajectories remain bounded or converge to a fixed point as time progresses. This means that small perturbations or disturbances to the system do not cause its behavior to change drastically. Stability is particularly important in the field of control, where it ensures that a controlled system behaves predictably and does not exhibit undesirable behavior.

12.1 Overview

There are two main notions of stability for dynamical systems.

- *Internal (or Lyapunov) stability* is a property of the internal state of the system that characterizes the zero-input response. A system is internally stable if, when the state starts “close” to an equilibrium point, then it stays “close” to the equilibrium point as the system evolves in time.
- *External (or input-output) stability* is a property of input-output response of the system that characterizes the zero-state response. A system is input-output stable if, whenever the input is “small”, the output is also “small”.

The following table summarizes the two main types of stability.

	Internal stability	External stability
Alternative names	Lyapunov stability	input-output stability
Type of response characterized	zero-input response	zero-state response
Stability with respect to	an equilibrium	an input signal

There are also other types of stability, such as input-to-state stability (ISS), that characterize stability of the state with respect to the input signal.

12.2 Internal stability

Internal stability is a property of the state of a dynamical system. Since this property does not concern the input, we will assume that the system has no input signal.

Motivation

Imagine a simple pendulum consisting attached to a rod that is free to swing in a vertical plane. When the pendulum is at rest, hanging downward, it is in a stable equilibrium position. If the pendulum is slightly displaced from this position, it will oscillate back and forth around the equilibrium point, eventually coming to rest again.

However, if the pendulum is inverted (upside down), it is in an unstable equilibrium position. A small disturbance in this state will cause the pendulum to swing further away from the equilibrium point, eventually leading to the pendulum falling uncontrollably.

This example illustrates the concept of stability in a dynamical system. The stable equilibrium of the pendulum corresponds to a system where small disturbances do not lead to drastic changes in behavior, while the unstable equilibrium demonstrates how a system can exhibit unstable behavior, where small disturbances lead to large changes in the system's state.

Definition

Consider the nonlinear system

$$\dot{x}(t) = f(x(t)), \quad x(0) = x_0.$$

Internal stability is defined with respect to an equilibrium point (or more generally, an equilibrium trajectory). Recall that an equilibrium point \tilde{x} is a point such that $f(\tilde{x}) = 0$ (for continuous-time systems). If the system is initialized at the equilibrium point, then it stays in equilibrium for all future time.

An equilibrium point \tilde{x} is:

- **stable** if, for all $\varepsilon > 0$, there exists $\delta > 0$ such that

$$\|x_0 - \tilde{x}\| < \delta \implies \|x(t) - \tilde{x}\| < \varepsilon \text{ for all } t \geq 0.$$

- **asymptotically stable** if there exists $\delta > 0$ such that

$$\|x_0 - \tilde{x}\| < \delta \implies \lim_{t \rightarrow \infty} \|x(t) - \tilde{x}\| = 0.$$

- **exponentially stable** if there exists $\delta > 0$, $K > 0$, and $\lambda > 0$ such that

$$\|x_0 - \tilde{x}\| < \delta \implies \|x(t) - \tilde{x}\| \leq K e^{-\lambda t} \|x_0 - \tilde{x}\| \text{ for all } t \geq 0.$$

Moreover, any of the properties is *global* if it holds for all initial conditions (that is, $\delta = \infty$), and a system is *unstable* if it is not stable.

Stability means that, whenever the state starts sufficiently close to the equilibrium point, it remains arbitrarily close to the equilibrium point for all time. Notice, however, that the two measures of closeness (one for the initial state and the other for the state at all future times) need not be the same. No matter how close we want the state to stay to the equilibrium (described by ε), there exists some set of initial conditions (described by δ) for which the state will remain that close.

Asymptotic stability is similar to stability except that the state eventually converges to the equilibrium, and exponential stability means that the state converges to the equilibrium exponentially fast.

The definitions for internal stability are identical in discrete time.

Internal stability of LTI systems

Now consider the LTI systems:

$$\dot{x}(t) = Ax(t) \quad \text{and} \quad x(k+1) = Ax(k).$$

Internal stability satisfies the following for LTI systems:

- The origin ($\tilde{x} = 0$) is always an equilibrium point.
- Asymptotic and exponential stability are equivalent.
- All stability properties are global.

Therefore, the following results completely characterize internal stability for LTI systems.

Fact (Internal stability of continuous-time LTI systems). For the continuous-time LTI system $\dot{x}(t) = Ax(t)$, the origin is

- globally exponentially stable if and only if all eigenvalues of A have strictly negative real part:

$$\operatorname{Re}(\lambda) < 0$$

- stable if and only if all eigenvalues of A have nonpositive real part and, for any eigenvalue with zero real part, the algebraic and geometric multiplicities are equal:

$$\operatorname{Re}(\lambda) \leq 0 \quad \text{and} \quad n_\lambda = \dim(E_\lambda) \quad \text{whenever} \quad \operatorname{Re}(\lambda) = 0$$

This is because, if $\lambda = a + bi$, then

$$e^{\lambda t} = e^{at} (\cos(bt) + i \sin(bt))$$

where the first term is exponential growth or decay, while the second term is sinusoidal oscillations. Therefore, the sign of $a = \operatorname{Re}(\lambda)$ determines whether the exponential grows ($a > 0$), decays ($a < 0$), or is constant ($a = 0$).

We have to be careful when $a = 0$ because, if $\dim(E_\lambda) < n_k$, then the matrix exponential contains polynomial terms that grow without bound.

Example. Consider the system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x$$

The solution to the state equations is

$$x(t) = e^{At} x(0) \quad \text{where} \quad e^{At} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}.$$

Due to the polynomial term in the matrix exponential, the first state grows without bound whenever the second state has nonzero initialization.

Such behavior does not occur when the algebraic and geometric multiplicities are equal because the Jordan blocks have size one.

The results in discrete time are similar.

Fact (Internal stability of discrete-time LTI systems). For the discrete-time LTI system $x(k+1) = Ax(k)$, the origin is

- globally exponentially stable if and only if all eigenvalues of A have magnitude strictly less than one:

$$|\lambda| < 1$$

- stable if and only if all eigenvalues of A have magnitude less than or equal to one and, for any eigenvalue with unit magnitude, the algebraic and geometric multiplicities are equal:

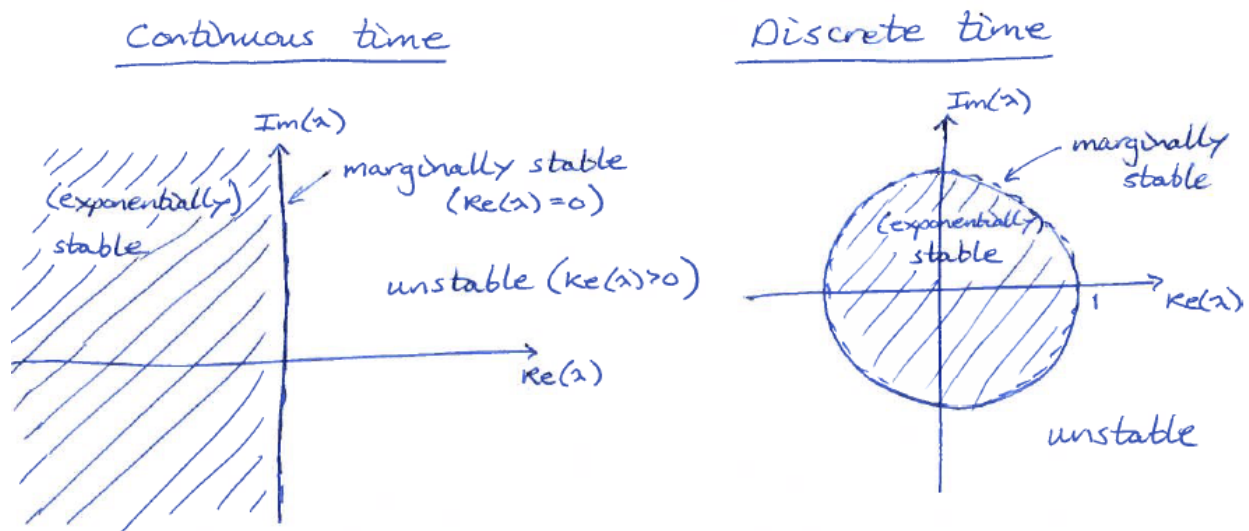
$$|\lambda| \leq 1 \quad \text{and} \quad n_\lambda = \dim(E_\lambda) \quad \text{whenever} \quad |\lambda| = 1$$

This is because, if $\lambda = r e^{j\theta}$, then

$$\lambda^k = r^k (\cos(k\theta) + i \sin(k\theta))$$

where the first term is exponential growth or decay, while the second term is sinusoidal oscillations. Therefore, the magnitude of $r = |\lambda|$ determines whether the exponential grows ($r > 1$), decays ($r < 1$), or is constant ($r = 1$).

The stability regions are as follows.



Remark. For LTI systems, internal stability is completely characterized by the eigenvalues (and eigenvectors for marginally stable eigenvalues) of A . ■

Lyapunov's direct method

While we can characterize internal stability in terms of the eigenstructure of A , characterizing stability of nonlinear systems is in general much more difficult. Instead of using eigenvalues, we will search for a function that represents the (generalized) energy of the system. Similar to the energy of physical systems, this energy function must be nonnegative and decreasing over time. Moreover, we will require that the equilibrium point is the minimum energy state. Such a function is called a *Lyapunov function* and is characterized as follows.

Theorem (Lyapunov stability). Consider the nonlinear system

$$\dot{x}(t) = f(x(t)), \quad x_0 = x(0) \in \mathbb{R}^n.$$

Suppose we can find a function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ that maps states to real numbers such that V has continuous partial derivatives and

- V is positive definite, meaning that $V(0) = 0$ and $V(x) > 0$ whenever $x \neq 0$.
- V is decreasing along system trajectories, meaning that $\dot{V}(x) \leq 0$ whenever x satisfies the system dynamics $\dot{x} = f(x)$.

Then, the system is stable about the equilibrium $\tilde{x} = 0$.

Comments

- *Interpretation.* We interpret the function V as an “energy” of the system, in which case the condition that $\dot{V} \leq 0$ is that the energy dissipates or is conserved over time.
- *Local vs global stability.* If the two conditions ($V > 0$ and $\dot{V} \leq 0$) hold for all states, then we can conclude *global* stability. If the conditions only hold near the equilibrium, then we can conclude *local* stability.
- *Proof idea.* The idea of the proof is that, roughly,

$$V(x(t)) = V(x(0)) + \int_0^t \dot{V}(x(\tau)) \, d\tau \leq V(x(0)),$$

so $V(x(t))$ is bounded. Moreover, as $x(0) \rightarrow 0$, we have that $V(x(0)) \rightarrow 0$ and $V(x(t)) \rightarrow 0$.

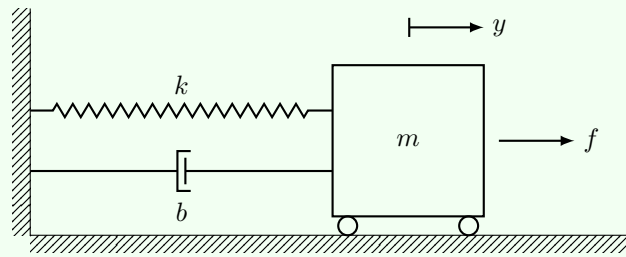
- *Computing \dot{V} .* The derivative of the function V is a derivative with respect to time. Since V depends on the state x which depends on time, we need to use the chain rule:

$$\begin{aligned} \dot{V}(x) &= \frac{\partial V}{\partial x_1} \dot{x}_1 + \frac{\partial V}{\partial x_2} \dot{x}_2 + \dots + \frac{\partial V}{\partial x_n} \dot{x}_n \\ &= (\nabla V(x))^\top \dot{x} \\ &= (\nabla V(x))^\top f(x) \end{aligned}$$

where the gradient of V is the vector-valued function of partial derivatives,

$$\nabla V(x) = \begin{bmatrix} \frac{\partial V}{\partial x_1} \\ \vdots \\ \frac{\partial V}{\partial x_n} \end{bmatrix}.$$

Example (Lyapunov stability of spring–mass–damper system).



Consider the spring–mass–damper mechanical system shown above. As we have seen, the position of the mass is described by the second-order differential equation

$$m\ddot{y} + b\dot{y} + ky = f.$$

To study Lyapunov stability of the system, suppose the applied force is zero ($f = 0$). The state of the system consists of the position and velocity of the mass, $x = (y, \dot{y})$. Since this is a physical system, we can check if the physical energy of the system is a Lyapunov function. The total energy is the sum of the kinetic energy due to the velocity of the mass and the potential energy that is stored in the spring,

$$V(x) = \frac{1}{2}m\dot{y}^2 + \frac{1}{2}ky^2$$

The function V is positive definite since it is nonnegative and is zero only if $x = 0$. Moreover, the time derivative of V is decreasing along system trajectories since

$$\begin{aligned} \dot{V}(x) &= \frac{\partial V}{\partial y} \frac{dy}{dt} + \frac{\partial V}{\partial \dot{y}} \frac{d\dot{y}}{dt} && \text{(chain rule)} \\ &= (ky)\dot{y} + (m\dot{y})\ddot{y} \\ &= ky\dot{y} + m\dot{y}\left(-\frac{b}{m}\dot{y} - \frac{k}{m}y\right) && \text{(using system dynamics)} \\ &= -b\dot{y}^2 \\ &\leq 0 \end{aligned}$$

Therefore, V is a Lyapunov function that proves global stability of the origin.

There are two cases, depending on whether or not the system contains damping.

- If there is no damping ($b = 0$), then $\dot{V} \equiv 0$, so the energy is conserved.
- If there is damping ($b > 0$), then $\dot{V}(x) < 0$ whenever $\dot{y} \neq 0$, so energy is dissipated whenever the mass is moving.

Lyapunov's indirect method

Given a nonlinear system $\dot{x} = f(x)$ with equilibrium point \tilde{x} , we have seen how to linearize the system to obtain its linearization

$$\dot{\delta}_x = A\delta_x \quad \text{where} \quad A = \frac{\partial f}{\partial x}(\tilde{x}) \quad \text{and} \quad \delta_x = x - \tilde{x}.$$

Since the linearization is an LTI system, we can analyze its stability properties by computing the eigenstructure of the Jacobian matrix A . In some cases, we can conclude stability properties of the *nonlinear* system

from stability properties of its linearization. Since the linearization is only a good approximation to the nonlinear system near the equilibrium, however, we can only conclude *local* stability properties using this method.

Fact (Lyapunov's indirect method). If the linearization is exponentially stable, then the equilibrium \tilde{x} of the nonlinear system is locally asymptotically stable. Otherwise, the linearization fails to provide stability properties of the nonlinear system.

Example (Van der Pol oscillator). Instability of the linearization does *not* in general imply instability of the nonlinear system, as one might suspect. To illustrate this, consider the nonlinear system

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 + \mu(1 - x_1^2)x_2\end{aligned}$$

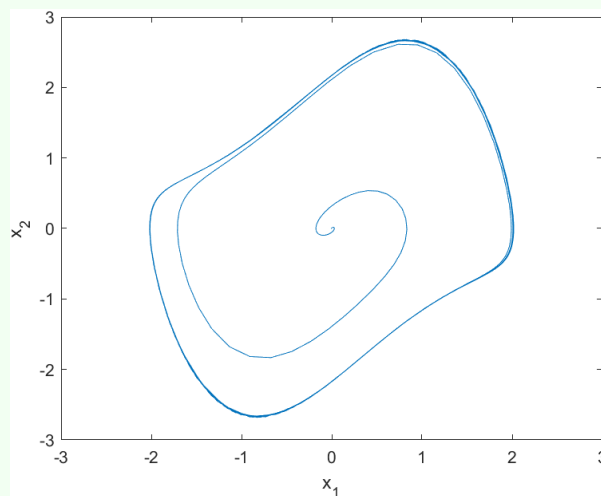
where $\mu > 0$ is a positive parameter. This system has an equilibrium point at the origin. The linearization about the origin is

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & \mu \end{bmatrix} x.$$

The system matrix has characteristic polynomial $\lambda^2 - \mu\lambda + 1$ which has roots

$$\lambda = \frac{\mu \pm \sqrt{\mu^2 - 4}}{2}.$$

The eigenvalues have positive real part for μ sufficiently small, so the linearization is unstable. While the origin of the nonlinear system is in fact unstable, the system does have a stable limit cycle as shown below.



12.3 External stability

External (input-output) stability depends on the inputs and outputs of a system, but *not* the state. Roughly, a system is input-output stable if “small” inputs produce “small” outputs. A common way to specify what we mean by “small” is whether or not a signal is bounded, in which case this is called bounded-input

bounded-output (BIBO) stability. A system is externally stable if there exists a constant $\eta > 0$ such that

$$\|y\| \leq \eta \|u\| \quad \text{for all } u \text{ and } y.$$

The constant η characterizes how large the output can be based on the size of the input, where the norm $\|\cdot\|$ characterizes the size of the signal. As we will see, the characterization of stability will depend on the particular norm.

Vector norms

Definition (Norm). For a vector space V , a function $\|\cdot\| : V \rightarrow \mathbb{R}$ is a norm if it satisfies the following properties for all vectors $x, y \in V$ and all scalars $\alpha \in \mathbb{R}$.

- *Absolutely scalable.* $\|\alpha x\| = |\alpha| \|x\|$
- *triangle inequality.* $\|x + y\| \leq \|x\| + \|y\|$
- *Nonnegativity.* $\|x\| \geq 0$
- *Definiteness.* $\|x\| = 0$ implies that $x = 0$

Remark. The nonnegativity condition is redundant since it follows from the first two properties as follows:

$$0 = \|0\| = \|x + (-x)\| \leq \|x\| + \|-x\| = \|x\| + |-1| \|x\| = 2 \|x\|.$$

■

p -Norm

For the vector space $V = \mathbb{R}^n$, a common norm is the p -norm, which is defined as

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \quad \text{for } p \geq 1$$

(this is not a norm when $0 < p < 1$ since it violates the triangle inequality). Some common examples are as follows.

- When $p = 1$, this is called the taxicab or Manhattan norm.

$$\|x\|_1 = |x_1| + |x_2| + \dots + |x_n|$$

- When $p = 2$, this is the standard Euclidean norm.

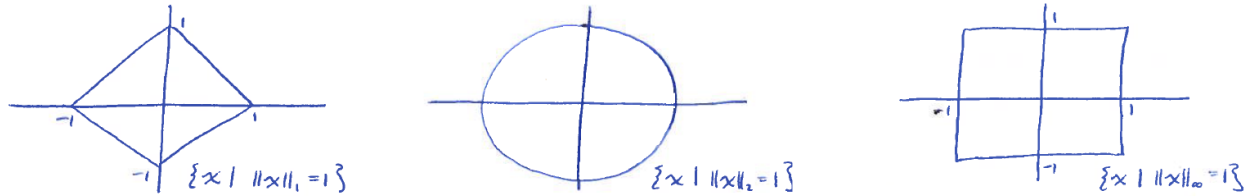
$$\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} = \sqrt{x^\top x}$$

(when unspecified, this is typically what $\|x\|$ means).

- When $p = \infty$, this is the infinity-norm or max-norm.

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

We can visualize p -norms by plotting their level sets (the set of vectors with unit norm) in two dimensions:



For *finite* vector spaces like \mathbb{R}^n , all norms are equivalent in that, if $\|x\|$ is bounded for some norm, then it is bounded for any norm. For instance, we have the following bounds:

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \leq \sqrt{n} \|x\|_2 \leq n \|x\|_\infty$$

Matrix norms

A matrix is an element of the vector space $V = \mathbb{R}^{m \times n}$. We could therefore apply any of the vector norms described previously. However, matrices have other norms that are useful.

Induced norms

An induced norm is a matrix norm that is derived from a vector norm. Given a vector norm, the corresponding matrix norm is

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

where the left-hand side is a matrix norm and the right-hand side contains two (possibly different) vector norms. Typically, $\|A\|$ stands for the induced norm using the vector 2-norm, though we could choose any vector norms on the right-hand side to obtain different matrix norms.

As a special case, the induced 2-norm or *spectral norm* of a matrix is

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \sqrt{\lambda_{\max}(A^T A)} = \sigma_{\max}(A),$$

where λ_{\max} denotes the largest eigenvalue and σ_{\max} the largest singular value.

Frobenius norm

The Frobenius norm is defined as

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |A_{ij}|^2} = \sqrt{\text{tr}(A^T A)} = \sqrt{\sum_{i=1}^r \sigma_i^2(A)}$$

where $r = \min\{m, n\}$. The Frobenius norm is *not* an induced norm.

Nuclear norm

The nuclear norm is defined as

$$\|A\|_* = \text{tr}(\sqrt{A^T A}) = \sum_{i=1}^r \sigma_i(A)$$

Schatten p -norms

All of the above matrix norms are equivalent to taking the vector p -norm of the singular values of the matrix A . These are called Schatten p -norms.

$$\begin{aligned} \text{spectral norm} &= \infty\text{-norm of singular values} \\ \text{Frobenius norm} &= 2\text{-norm of singular values} \\ \text{nuclear norm} &= 1\text{-norm of singular values} \end{aligned}$$

As with vectors, all matrix norms are equivalent:

$$\|A\|_2 \leq \|A\|_F \leq \|A\|_* \leq \sqrt{r} \|A\|_F \leq r \|A\|_2$$

This means that it does not matter what norm we use in describing sequences that converge to zero, since

$$\lim_{k \rightarrow \infty} x_k = 0 \quad \text{means that} \quad \lim_{k \rightarrow \infty} \|x_k\| = 0$$

for any norm, and similarly for any sequence of matrices A_k .

Infinite-dimensional vector spaces

We can interpret a signal as an infinite-dimensional vector indexed by time and a linear system as an infinite-dimensional matrix. Therefore, we now consider norms on infinite-dimensional vector spaces.

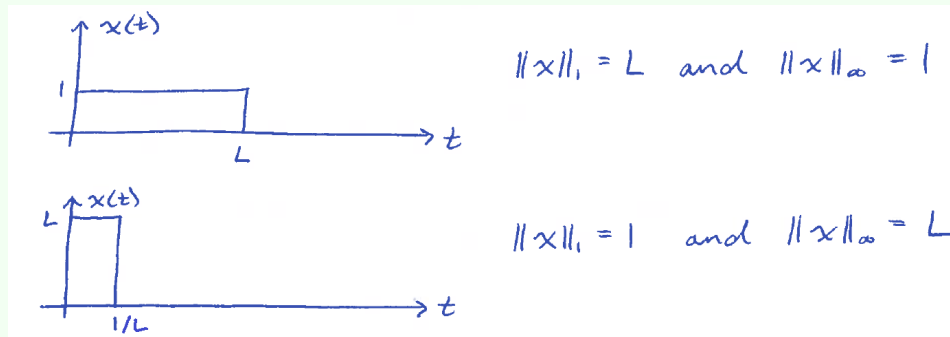
For continuous-time signals, time takes values in the set of nonnegative real numbers $\mathbb{R}_{\geq 0}$. For discrete-time signals, time takes values in the set of nonnegative integers $\mathbb{Z}_{\geq 0}$. The corresponding p -norms are as follows:

Continuous time	Discrete time
$\ x\ _1 = \int_0^{\infty} x(t) dt$	$\ x\ _1 = \sum_{k=0}^{\infty} x_k $
$\ x\ _2 = \sqrt{\int_0^{\infty} x(t) ^2 dt}$	$\ x\ _2 = \sqrt{\sum_{k=0}^{\infty} x_k ^2}$
$\ x\ _{\infty} = \text{ess sup}_{t \geq 0} x(t) $	$\ x\ _{\infty} = \sup_{k \geq 0} x_k $

(ess sup denotes the essential supremum, which is the maximum excluding sets of measure zero).

While all vector (and matrix) norms in finite-dimensional spaces are equivalent, norms in infinite-dimensional spaces are not in general equivalent.

Example (Non-equivalent norms). Consider the following two signals:



Since this holds for any $L > 0$, there cannot exist a constant $c > 0$ such that, for all x ,

$$\|x\|_1 \leq c \|x\|_\infty \quad \text{or} \quad \|x\|_\infty \leq c \|x\|_1.$$

Similarly, we can show that $\|x\|_1$ and $\|x\|_2$ are not equivalent.

This example illustrates that signal norms are not equivalent, so minimizing one norm is generally not equivalent to minimizing another norm.

BIBO stability

Definition (BIBO stability). A system is BIBO stable if there exists a constant $\eta > 0$ such that, for any input signal u , the zero-state response y satisfies $\|y\|_\infty \leq \eta \|u\|_\infty$.

Fact (BIBO stability of LTI system). The LTI system

$$\begin{aligned} \dot{x} &= Ax + Bu, & x(0) &= 0 \\ y &= Cx + Du \end{aligned}$$

is BIBO stable if and only if the impulse response

$$H(t) = Ce^{At}B + D\delta(t)$$

satisfies

$$\int_0^\infty \|H(t)\|_2 dt < \infty.$$

Proof. See handwritten notes. ■

Fact. For single-input single-output linear time-invariant systems, the following are equivalent:

- The system is BIBO stable.
- The impulse response $h(t)$ is absolutely integrable:

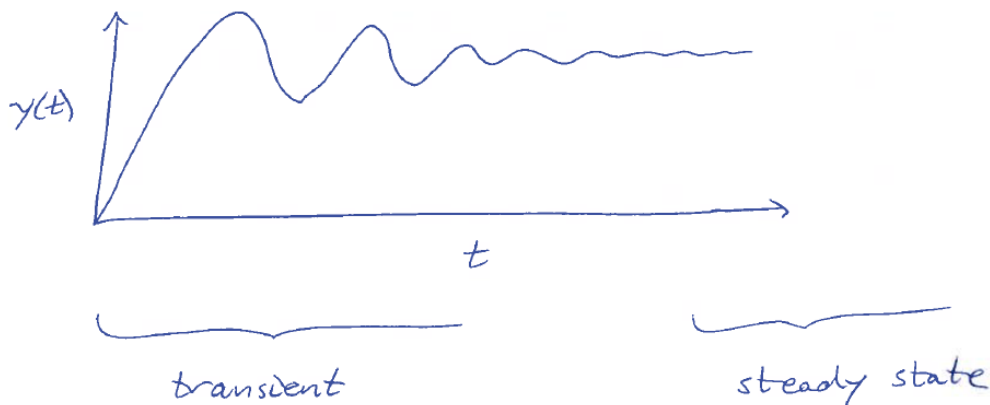
$$\int_0^{\infty} |h(t)| dt < \infty$$

- The transfer function $H(s)$ has all poles in the open left-half plane.

13

Transient Response

Stability and system norms (such as \mathcal{H}_2 and \mathcal{H}_∞) are asymptotic properties of the system. In contrast, the transient response describes the initial behavior of the system. In this chapter, we characterize the transient response of the system.



13.1 Dominant modes

The transient properties of a system are more difficult to characterize in general than asymptotic properties, so we often work with approximations.

Definition (Dominant poles). A pole of the transfer function is dominant if it is separated to the right from the other poles in real part by a factor of 10 or more.

Example (Dominant poles). The transfer function

$$H(s) = \frac{1}{(s+2)(s+30)}$$

has a dominant pole at $s = -2$, and the transfer function

$$H(s) = \frac{1}{(s^2 + 2s + 5)(s+15)(s+100)}$$

has a pair of dominant complex conjugate poles at $s = -1 \pm j2$.

For the purpose of transient analysis (not steady-state analysis!), systems with dominant poles behave similar to how they would if we neglected all other poles. Therefore, we assume that the transfer function can be appropriately approximated by a low-order system consisting of only one or two dominant poles. For systems with more dominant poles, the transient analysis is much more complicated.

13.2 First-order systems

Without loss of generality, the transfer function of a first-order system has the form

$$H(s) = \frac{K}{\tau s + 1}$$

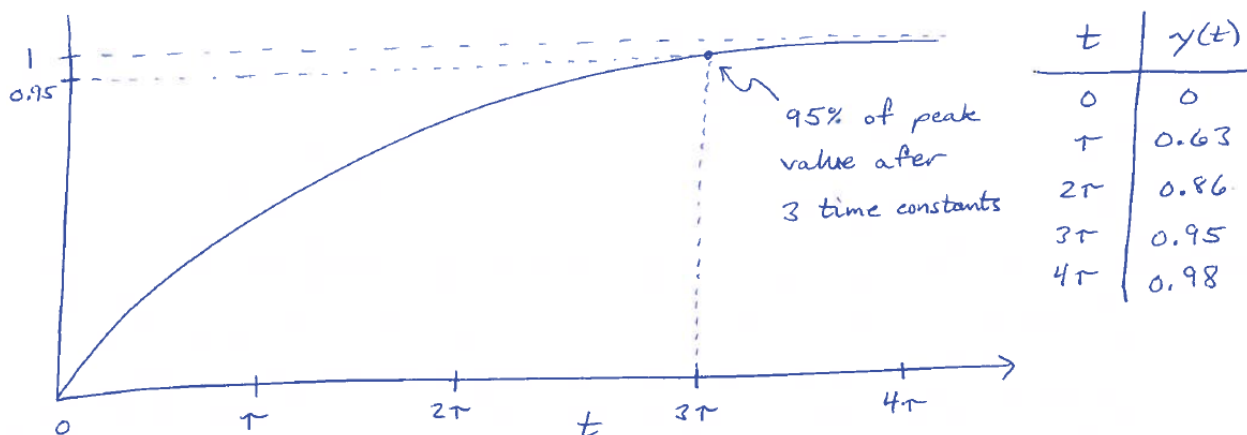
where K is the dc gain and τ is the time constant (in units of seconds). The step response is

$$Y(s) = H(s)U(s) = \frac{K}{\tau s + 1} \cdot \frac{1}{s} = K \left(\frac{1}{s} - \frac{\tau}{\tau s + 1} \right),$$

which has inverse Laplace transform

$$y(t) = K(1 - e^{-t/\tau}) \quad \text{for } t \geq 0.$$

The step response reaches 95% of its final value K after three time constants as illustrated below.



13.3 Second-order systems

Without loss of generality, the transfer function of a second-order system has the form

$$H(s) = \frac{K \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where ω_n is the natural frequency (in units of radians per second) and ζ is the damping ratio (unitless).

The poles of the system are $s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$. The transient response changes shape based on the value of the damping ratio.

	damping ratio	poles	step response
overdamped	$\zeta > 1$	real and distinct (both stable)	slow
critically damped	$\zeta = 1$	real and equal (stable)	fastest without overshoot
underdamped	$0 < \zeta < 1$	complex conjugate (stable)	overshoot and oscillation
undamped	$\zeta = 0$	imaginary	undamped oscillation
unstable	$\zeta < 0$	real and distinct (both unstable)	unbounded

The cases $\zeta < 0$ and $\zeta > 1$ are not that interesting because we can split the system up into two first-order systems. Therefore, we consider the underdamped case.

Underdamped

Suppose the system is underdamped, meaning that $0 < \zeta < 1$. The step response is

$$Y(s) = G(s)U(s) = \frac{K \omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} = K \left(\frac{1}{s} - \frac{s + 2\zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right).$$

To obtain the inverse Laplace transform, we split the second term as

$$Y(s) = K \left(\frac{1}{s} - \frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + (1 - \zeta^2)\omega_n^2} - \frac{\zeta\omega_n}{(s + \zeta\omega_n)^2 + (1 - \zeta^2)\omega_n^2} \right).$$

Recall the following Laplace transform pairs:

$$\mathcal{L}\{e^{-at} \cos(bt)\} = \frac{s + a}{(s + a)^2 + b^2} \quad \text{and} \quad \mathcal{L}\{e^{-at} \sin(bt)\} = \frac{b}{(s + a)^2 + b^2}.$$

Therefore, the step response is

$$y(t) = K \left(1 - \left[\cos(\sqrt{1 - \zeta^2}\omega_n t) + \frac{\zeta}{\sqrt{1 - \zeta^2}} \sin(\sqrt{1 - \zeta^2}\omega_n t) \right] \right) e^{-\zeta\omega_n t} \quad \text{for } t \geq 0.$$

Define the damped frequency

$$\omega_d = \omega_n \sqrt{1 - \zeta^2}$$

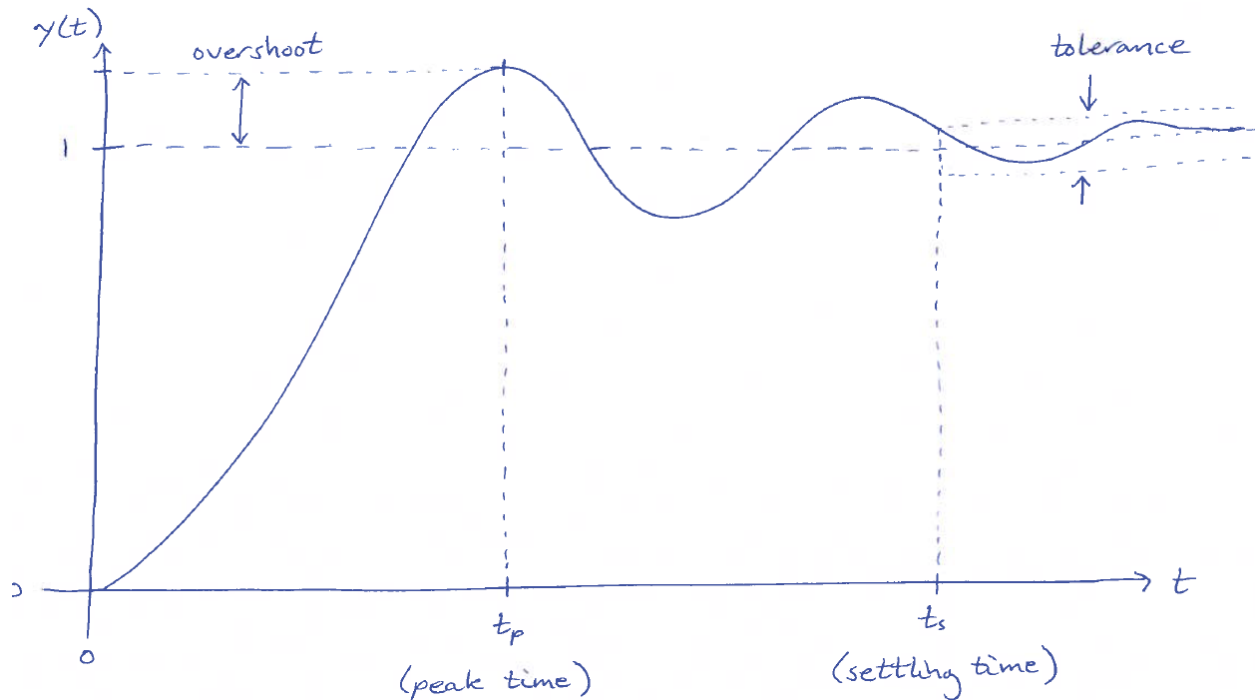
and let θ be the angle such that $\zeta = \sin \theta$ and $\sqrt{1 - \zeta^2} = \cos \theta$. The step response then simplifies to

$$y(t) = K \left(1 - \frac{1}{\sqrt{1 - \zeta^2}} \left[\cos \theta \cos(\omega_d t) + \sin \theta \sin(\omega_d t) \right] \right) e^{-\zeta\omega_n t} \quad \text{for } t \geq 0.$$

Simplifying further,

$$y(t) = K \left(1 - \frac{1}{\sqrt{1 - \zeta^2}} \cos(\omega_d t - \theta) e^{-\zeta\omega_n t} \right) \quad \text{for } t \geq 0.$$

The step response is illustrated below.



Transient specifications

- *Overshoot* is the amount the step response exceeds its final value.
- *Percent overshoot (PO)* is

$$\frac{(\text{overshoot}) - (\text{steady-state value})}{(\text{steady-state value})} \times 100\%.$$

- *Peak time t_p* is the time for the step response to reach the first (highest) peak.
- *Settling time t_s* is the shortest time until the response stays within some percent of the final value (usually 2% or 5%).

Computing performance specifications

For an underdamped system, the performance specifications can be computed as follows.

- The peak time is the first time for which $\dot{y}(t) = 0$.

$$t_p = \frac{\pi}{\omega_d}$$

- The percent overshoot can be computed from the value of the step response at the peak time.

$$\text{PO} = 100 \exp\left(-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right)$$

- The settling time is the time when $|1 - y(t)| \leq \varepsilon$. Since $|\cos \theta| \leq 1$, the magnitude of the step response is bounded by

$$|1 - y(t)| \leq \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t}$$

which yields the following bound on the ε -settling time:

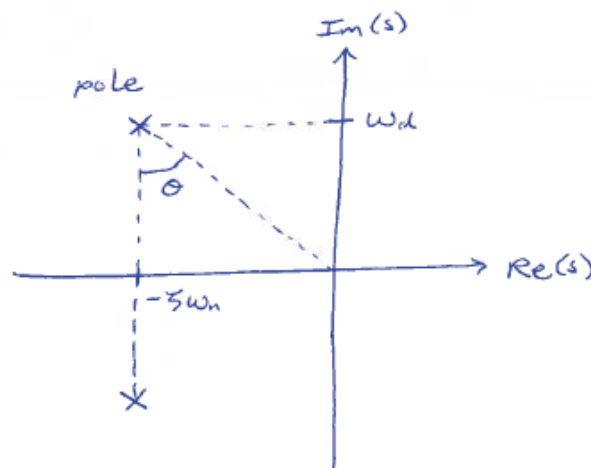
$$t_s \geq \frac{1}{\zeta\omega_n} \log\left(\frac{1}{\varepsilon\sqrt{1-\zeta^2}}\right).$$

For ζ not too close to one, the settling time can be approximated as

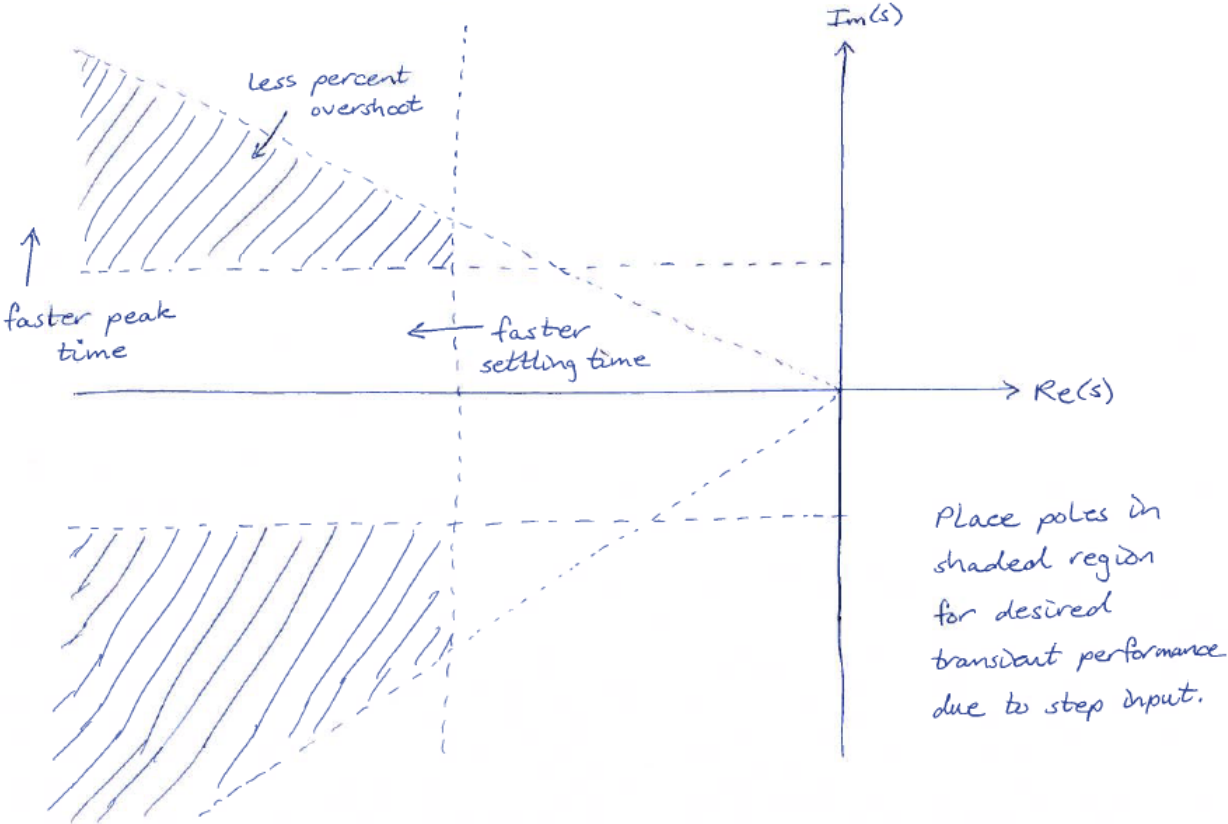
$$t_s^{2\%} \approx \frac{4}{\zeta\omega_n} \quad \text{and} \quad t_s^{5\%} \approx \frac{3}{\zeta\omega_n}.$$

Intuitive picture

The poles are at $s = -\zeta\omega_n \pm j\omega_d$.



- The peak time is $t_p = \pi/\omega_d$ which depends only on the imaginary part of the poles.
- The settling time $t_s \approx k/\zeta\omega_n$ depends only on the real part of the poles.
- The percent overshoot $PO = 100\exp(-\pi \tan \theta)$ depends only on the angle of the poles.



Part IV

Control

14

Static State Feedback

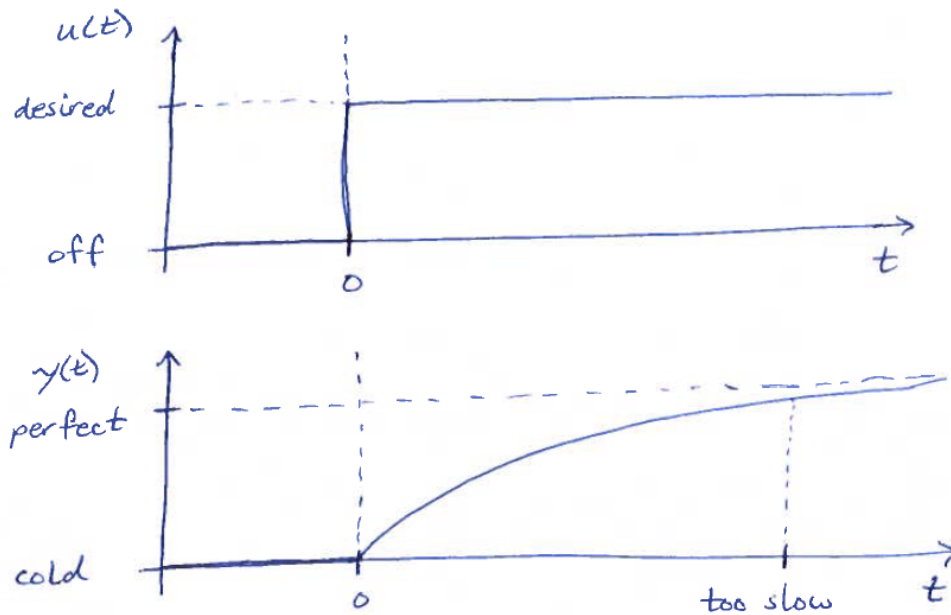
14.1 Open-loop vs feedback control

Given a system, we may want to choose the input u in an automatic way to improve the system performance. The two main types of performance criteria are:

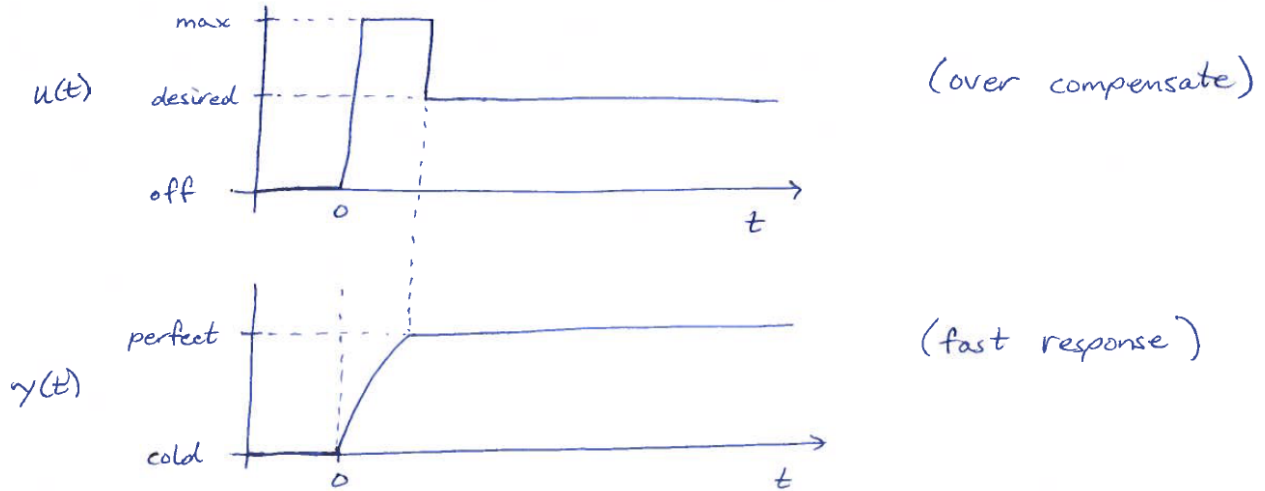
- **Asymptotic:** stability, steady-state error
- **Transient:** overshoot, rise time, settling time

Motivating example

Consider controlling the water temperature of the shower. The control input u is the faucet position, and the measured output y is the water temperature (for example, using your hand). The open-loop step response looks something like this:

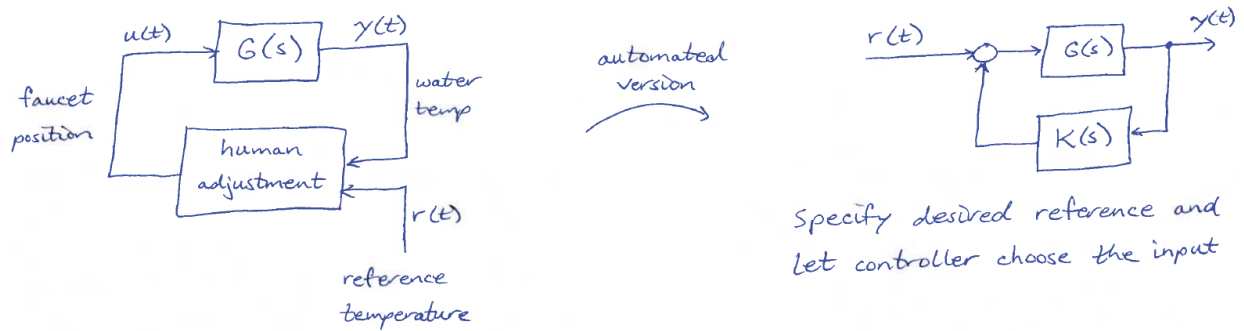


When you are in a rush, however, this response may be considered too slow. To make the response faster, we could modify the input to first turn the faucet to the maximum position, and then as soon as the temperature reaches the desired temperature, move the faucet instantaneously to the desired position to maintain the perfect temperature.



The input signal in this case is an *open-loop* controller because the value of the input signal is determined beforehand and then applied to the system without using any measurements. Such controllers are very risky and typically fail because they require perfect models of the system and perfect actuation.

Instead, a more robust control method is to use sensors to provide *feedback* about the state of the system, which is then used to automatically adjust the input signal. Continuing our example, one method of feedback is to use your hand to measure the temperature of the water and then adjust the faucet position accordingly until the water temperature is just right.



This approach for selecting the input is called *feedback control* and is the approach that we will study in this chapter.

14.2 Static state feedback

One type of feedback control is *static state feedback*. Here, the control input is chosen as a static function of the state and a reference input signal:

$$u(t) = Kx(t) + r(t).$$

The constant matrix K is called the *state feedback gain* and the signal r is a reference input. There are two main cases:

- When the reference is zero ($r(t) = 0$ for all t), this is called a *regulator* and the goal is to make the state converge to the origin quickly.
- When the reference is nonzero, this is called a *tracker* and the goal is to make the output follow the reference as closely as possible.

Remark. State feedback control assumes we have access to the entire state $x(t)$ at each time t when selecting the control input $u(t)$. This is not always possible, as it may require many costly sensors to measure the state completely. Later, we will see how to use state feedback in this scenario. ■

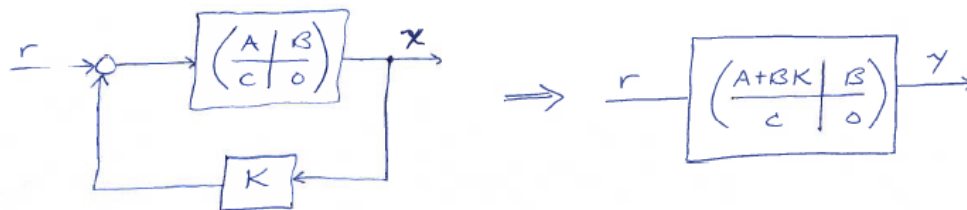
Consider applying the static state feedback controller to the continuous-time LTI system:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}$$

Substituting the static state feedback controller for the input signal, the dynamics become

$$\begin{aligned}\dot{x} &= (A + BK)x + Br \\ y &= Cx\end{aligned}$$

We can visualize this as the following block diagram:



Recall that stability only depends on the state transition matrix.

- the open-loop system is governed by matrix A
- the closed-loop system is governed by matrix $A + BK$

Therefore, we would like to choose the state feedback gain K so that the closed-loop system matrix $A + BK$ has desirable properties, such as being stable, having fast decay, etc.

Pole placement

The following result characterizes precisely when it is possible to arbitrarily place the closed-loop eigenvalues (or poles).

Fact (Pole placement). The following statements are equivalent:

- a) (A, B) is controllable
- b) the eigenvalues of $A + BK$ may be arbitrarily assigned by suitable choice of K , as long as they are chosen in complex conjugate pairs

Example. Consider the following system:

$$\dot{x} = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u.$$

This system has the form of the controllable decomposition, from which we can conclude (by inspection) that the first state is controllable while the second state is not. Let's see what happens when we apply static state feedback. Setting $u(t) = Kx(t)$ with $K = [k_1 \quad k_2]$, the closed-loop system is

$$\dot{x} = \begin{bmatrix} 1 + k_1 & 1 + k_2 \\ 0 & 2 \end{bmatrix} x$$

which has characteristic polynomial

$$\det(\lambda I - (A + BK)) = (\lambda - 1 - k_1)(\lambda - 2).$$

The closed-loop eigenvalues are $1 + k_1$ and 2. We can choose k_1 to move the eigenvalue at 1 arbitrarily, but we cannot move the uncontrollable state by any choice of K .

Now suppose the input directly affects the second state so that the system is

$$\dot{x} = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u.$$

The system is now controllable, and the closed-loop system is

$$\dot{x} = \begin{bmatrix} 1 & 1 \\ k_1 & 2 + k_2 \end{bmatrix} x$$

which has characteristic polynomial

$$\begin{aligned} \det(\lambda I - (A + BK)) &= (\lambda - 1)(\lambda - 2 - k_2) - k_1, \\ &= \lambda^2 + (-3 - k_2)\lambda + (2 - k_1 + k_2). \end{aligned}$$

We can choose k_1 and k_2 to achieve *any* polynomial of degree two. For instance, if we want both closed-loop poles to be at -1 , then the desired closed-loop characteristic polynomial is

$$(\lambda + 1)^2 = \lambda^2 + 2\lambda + 1.$$

Equating coefficients, we find that $k_2 = -5$ and $k_1 = -4$.

Bass–Guara formula

When the system is controllable, it is possible to place the closed-loop eigenvalues wherever we would like. We now study one way to do so.

As a first step, suppose the system is in controllable canonical form so that the state-space matrices are

$$A_{\text{CCF}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-2} & -a_{n-1} \end{bmatrix} \quad \text{and} \quad B_{\text{CCF}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

The coefficients in the last row of A_{CCF} are the coefficients in the characteristic polynomial,

$$\det(\lambda I - A_{\text{CCF}}) = \lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_1\lambda + a_0.$$

Suppose we want the closed-loop characteristic polynomial to be

$$\lambda^n + \alpha_{n-1}\lambda^{n-1} + \dots + \alpha_1\lambda + \alpha_0.$$

Then we want to choose K_{CCF} so that the closed-loop system matrix is

$$A_{\text{CCF}} + B_{\text{CCF}}K_{\text{CCF}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \dots & -\alpha_{n-2} & -\alpha_{n-1} \end{bmatrix}.$$

Therefore, choose the state feedback gain as

$$K_{\text{CCF}} = [a_0 - \alpha_0 \quad a_1 - \alpha_1 \quad \dots \quad a_{n-1} - \alpha_{n-1}].$$

The state feedback gain is simply the coefficients of the open-loop characteristic polynomial minus the coefficients of the desired closed-loop characteristic polynomial in order from lowest to highest degree.

The state feedback gain K_{CCF} assumes that the system is in controllable canonical form. In the state coordinates for the controllable canonical form, the system is

$$\dot{z}(t) = (A_{\text{CCF}} + B_{\text{CCF}}K_{\text{CCF}})z(t).$$

Since the transformed state is $z(t) = T^{-1}x(t)$, in the original coordinates the system dynamics are

$$\begin{aligned} \dot{x}(t) &= T\dot{z}(t) \\ &= T(A_{\text{CCF}} + B_{\text{CCF}}K_{\text{CCF}})T^{-1}x(t) \\ &= (A + BK)x(t) \end{aligned}$$

where the state feedback matrix in the original coordinates is

$$K = K_{\text{CCF}}T^{-1}.$$

Recall that T is the state transformation matrix that puts the system in controllable canonical form, which is given explicitly by $T = PP_{\text{CCF}}^{-1}$, where P is the controllability matrix of the original system (A, B) and P_{CCF} is the controllability matrix of the system in controllable canonical form, whose inverse has the simple expression

$$P_{\text{CCF}}^{-1} = \begin{bmatrix} a_1 & a_2 & \dots & a_{n-1} & 1 \\ a_2 & a_3 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}.$$

We summarize the Bass–Guara formula for pole placement as follows.

Fact (Bass–Guara formula for pole placement). Consider a realization (A, B) , and denote the characteristic polynomial of A as

$$\lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_1\lambda + a_0.$$

Then, the state feedback matrix K such that the characteristic polynomial of $A + BK$ is

$$\lambda^n + \alpha_{n-1}\lambda^{n-1} + \dots + \alpha_1\lambda + \alpha_0$$

is given by

$$K = K_{\text{CCF}}T^{-1} \quad \text{where} \quad K_{\text{CCF}} = [a_0 - \alpha_0 \quad a_1 - \alpha_1 \quad \dots \quad a_{n-1} - \alpha_{n-1}]$$

is the state feedback gain in controllable canonical form, $T = PP_{\text{CCF}}^{-1}$ is the state transformation matrix that transforms the system to controllable canonical form, P is the controllability matrix of (A, B) , and the (inverse of) the controllability matrix in controllable canonical form is the Hankel matrix

$$P_{\text{CCF}}^{-1} = \begin{bmatrix} a_1 & a_2 & \dots & a_{n-1} & 1 \\ a_2 & a_3 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}.$$

Example. Consider the system

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 15 & 7 & -1 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

The eigenvalues of A are the roots of

$$\det(\lambda I - A) = \lambda^3 + \lambda^2 - 7\lambda - 15$$

which are 3 and $-2 \pm j$. Let's choose K to move all the eigenvalues to -1 . The desired characteristic polynomial is then

$$(\lambda + 1)^3 = \lambda^3 + 3\lambda^2 + 3\lambda + 1.$$

We want the closed-loop system matrix to be

$$A + BK = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 & -3 \end{bmatrix}.$$

Therefore, choose $K = [-16 \quad -10 \quad -2]$.

Invariance of controllability under static state feedback

Fact. Controllability of a system is invariant under static state feedback.

Proof. First, note that

$$\begin{bmatrix} I & 0 \\ K & I \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -K & I \end{bmatrix},$$

so the matrix is invertible for any K . Then using the PBH test for controllability, the following are equivalent:

- a) $(A + BK, B)$ is controllable
- b) $[A + BK - \lambda I \quad B]$ has full row rank for all complex λ
- c) $[A - \lambda I \quad B] \begin{bmatrix} I & 0 \\ K & I \end{bmatrix}$ has full row rank for all complex λ
- d) $[A - \lambda I \quad B]$ has full row rank for all complex λ
- e) (A, B) is controllable

■

14.3 Stabilizability

A system is *stabilizable* if all of its unstable modes are controllable (but the stable modes need not be controllable). In this case, we can construct a state-feedback controller such that the system is stable; we can assign the closed-loop eigenvalues of the controllable modes, but the uncontrollable modes cannot be changed.

Theorem (Stabilizability). Given matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, the following statements are equivalent.

- (A, B) is stabilizable.
- $[\lambda I - A \quad B]$ has full row rank for all $\lambda \in \mathbb{C}$ with $\text{Re}(\lambda) \geq 0$.
- If $w^* A = \lambda w^*$ with $0 \neq w \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}$ with $\text{Re}(\lambda) \geq 0$, then $w^* B \neq 0$.
- There exists a matrix K such that $A + BK$ is stable.

Example. Consider the uncontrollable system

$$A = \begin{bmatrix} -3 & 2 \\ -2 & 2 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

The eigenvalues of A are $\lambda_1 = 1$ and $\lambda_2 = -2$, and the corresponding left eigenvectors are

$$w_1 = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad \text{and} \quad w_2 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}.$$

For the unstable eigenvalue λ_1 , we have that $w_1^* B = -3$, so the eigenvalue is controllable. For the stable eigenvalue λ_2 , we have that $w_2^* B = 0$, so the eigenvalue is uncontrollable. Even though the system is uncontrollable, all unstable eigenvalues are controllable, so the system is stabilizable, meaning that we can find a state feedback matrix such that the closed-loop system is stable. The closed-loop system is

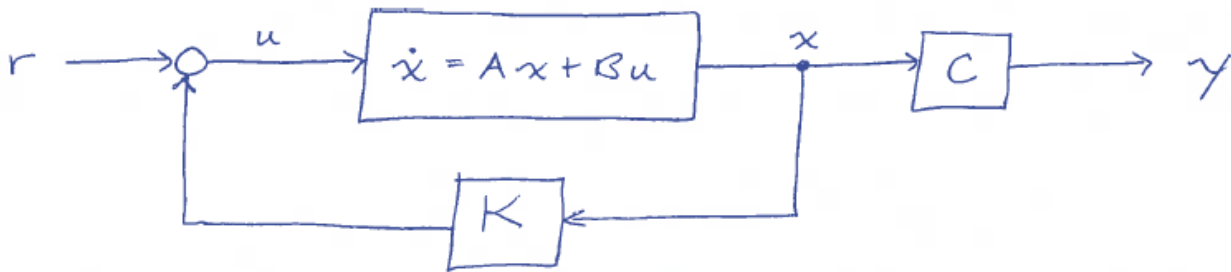
$$A + BK = \begin{bmatrix} -3 + k_1 & 2 + k_2 \\ -2 + 2k_1 & 2 + 2k_2 \end{bmatrix}$$

which has eigenvalues -2 and $1 + k_1 + 2k_2$. We cannot move the eigenvalue at -2 since it is uncontrollable, but we can choose K to move the unstable but controllable eigenvalue at 1.

15

Steady-state tracking

Recall the regulation problem in which we use static state feedback with the goal of having the output track the reference:



How do we ensure that the output $y(t)$ converges to the reference $r(t)$ in the limit as $t \rightarrow \infty$? We want

$$\lim_{t \rightarrow \infty} r(t) - y(t) = 0.$$

15.1 Open-loop control

We first consider an open-loop approach that relies on choosing the state feedback gain K based on the steady-state gain of the system.

Steady-state gain

Consider the system

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}$$

Suppose the system is internally stable (so the eigenvalues of A are in the stability region) and the input signal is constant. Then the system converges to an equilibrium point. We now find the steady-state gain, which is how much the input is amplified at the output. We can do so in either the time domain or frequency domain (we get the same answer in both cases).

Time domain

In the time domain, the equilibrium point must satisfy $\dot{x}(t) = 0$. Since A is stable, it is also invertible. We can therefore solve for the steady-state value of the output as

$$y_\infty = -CA^{-1}Bu_\infty$$

Frequency domain

In the frequency domain, we can use the final value theorem to find the steady-state value of the output as

$$\lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sY(s)$$

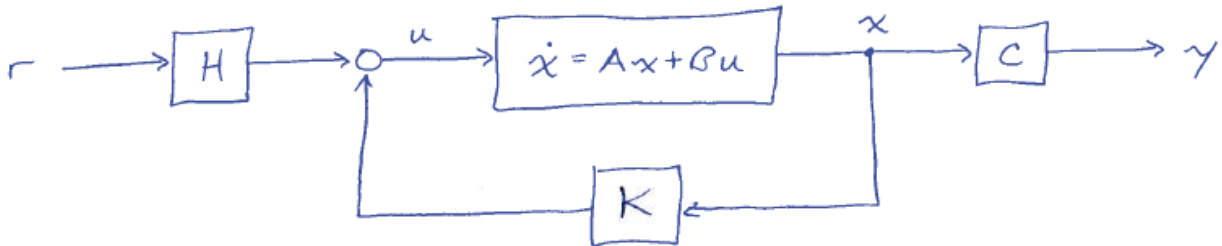
where $Y(s)$ is the Laplace transform of $y(t)$. The Laplace transform of a constant input is $U(s) = u_\infty/s$, so

$$\lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sG(s)U(s) = G(0)u_\infty.$$

Therefore, the steady-state gain is $G(0) = -CA^{-1}B$, which is the same as obtained in the time domain.

Regulation using steady-state gain

As a first approach to the regulation problem, we could achieve zero steady-state error by scaling the reference so that the steady-state gain is one.



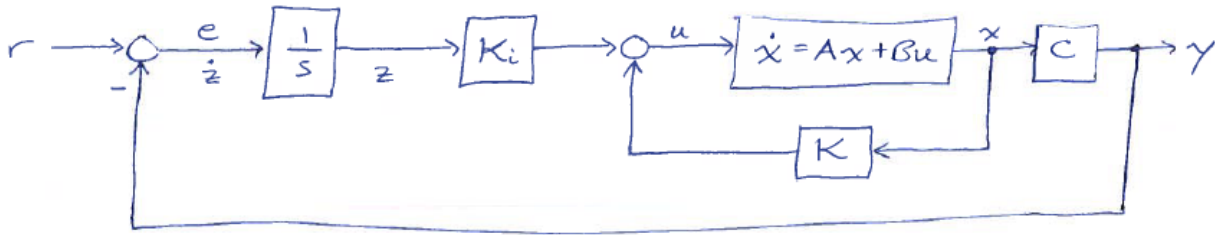
The steady-state gain of the closed-loop system is $-C(A + BK)^{-1}BH$, so we need to choose

$$H = -[C(A + BK)^{-1}B]^{-1}.$$

The problem with this approach, however, is that this requires *exact* knowledge of the system matrices (A, B, C). Any error in modeling will result in nonzero steady-state error. A more robust solution is to use feedback to ensure that the steady-state error is zero, even when the model of the system is imperfect.

15.2 Integral control

To achieve perfect steady-state tracking without perfect knowledge of the system matrices, we can augment the input with the integral of the error, $e(t) = r(t) - y(t)$.



The state equations represented by the block diagram are as follows:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ \dot{z} &= r - y \\ u &= Kx + K_i z \\ y &= Cx\end{aligned}$$

We can check that this system does indeed have zero steady-state error. Assuming that the closed-loop system is stable, the trajectories converge to an equilibrium point. Since this equilibrium is constant, it must satisfy $0 = \dot{z} = e = r - y$, so the output must be equal to the reference at steady state. Therefore, we have perfect tracking as long as the closed-loop system is stable.

15.3 Pole placement with integral control

Integral control produces zero steady-state error as long as the closed-loop system is stable. We previously saw how to place the closed-loop poles using static state feedback to achieve desired performance characteristics, such as stability and transient performance. We now consider pole placement when using integral control with static state feedback.

Closed-loop system

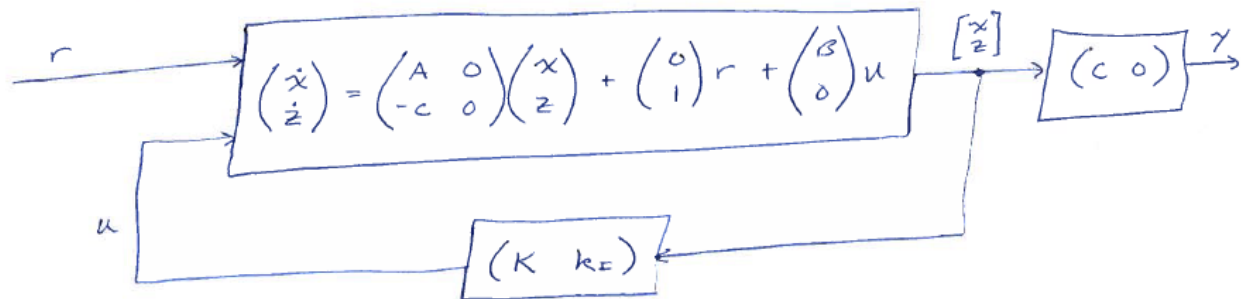
The closed-loop system from the reference input r to the output y is

$$\begin{aligned}\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} &= \begin{bmatrix} A + BK & BK_i \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \\ y &= [C \quad 0] \begin{bmatrix} x \\ z \end{bmatrix}\end{aligned}$$

The closed-loop system matrix is an affine function of the state feedback matrix and the integral gain,

$$\begin{bmatrix} A + BK & BK_i \\ -C & 0 \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} [K \quad K_i].$$

The block diagram of the closed-loop system is as follows.



From our previous results on pole placement, we can choose the state feedback matrix K and the integral control gain K_i to place the poles of the closed-loop system wherever we like if and only if

$$\left(\begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \begin{bmatrix} B \\ 0 \end{bmatrix} \right) \text{ is controllable.}$$

The following result shows that this is the case as long as the plant (A, B) is controllable and has no zeros at the origin (which would cancel out the integrator).

Fact. Suppose the plant (A, B) satisfies the following:

- (A, B) is controllable
- A has no eigenvalue at zero
- $C(sI - A)^{-1}B$ has no zero at $s = 0$

Then, the following augmented system is controllable:

$$\left(\begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \begin{bmatrix} B \\ 0 \end{bmatrix} \right).$$

Proof. See handwritten notes. ■

Under these conditions, the augmented system is controllable, so we can choose the state feedback matrix K and the integral gain K_i to place the closed-loop poles. We now consider two methods for designing such a controller.

Method 1

One method is to use the general procedure for pole placement applied to the augmented system. Note that P is then the controllability matrix of the augmented system.

Method 2

Suppose the plant (A, B, C) is in controllable canonical form. Then, the closed-loop system matrix is

$$\begin{bmatrix} A+BK & BK_i \\ -C & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ k_1 - a_0 & k_2 - a_1 & k_3 - a_2 & \dots & k_{n-1} - a_{n-2} & k_n - a_{n-1} & K_i \\ -c_1 & -c_2 & -c_3 & \dots & -c_n & 0 & 0 \end{bmatrix}$$

where

$$K = [k_1 \quad k_2 \quad \dots \quad k_n] \quad \text{and} \quad C = [c_1 \quad c_2 \quad \dots \quad c_n].$$

The characteristic polynomial of the closed-loop system is

$$s \left(s^n + (a_{n-1} - k_n)s^{n-1} + \dots + (a_1 - k_2)s + (a_0 - k_1) \right) + K_i (c_n s^{n-1} + \dots + c_2 s + c_1).$$

We can therefore choose K_i to match the constant term with that of the desired characteristic polynomial, and then choose the elements k_1, \dots, k_n of K to match the higher-order coefficients.

Example. Design an integral state-feedback controller for the system

$$H(s) = \frac{1}{s^3 + 2s^2 + 15s + 18}$$

so that the closed-loop poles are the roots of the polynomial

$$p(s) = s^4 + 4.2s^3 + 13.6s^2 + 21.6s + 16.$$

To use the second method, let's use the controllable canonical form of the system, which is

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] = \left[\begin{array}{ccc|c} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -18 & -15 & -2 & 1 \\ \hline 1 & 0 & 0 & 0 \end{array} \right].$$

Now form the augmented system

$$\begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -18 & -15 & -2 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} B \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

The closed-loop system is then

$$\begin{bmatrix} A + BK & BK_i \\ -C & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ k_1 - 18 & k_2 - 15 & k_3 - 2 & K_i \\ -1 & 0 & 0 & 0 \end{bmatrix}$$

which has characteristic polynomial

$$s(s^3 + (2 - k_3)s^2 + (15 - k_2)s + (18 - k_1)) + K_i.$$

Comparing coefficients with those of $p(s)$, we obtain

$$K = [-3.6 \quad 1.4 \quad -2.2] \quad \text{and} \quad K_i = 16.$$

16

Observers

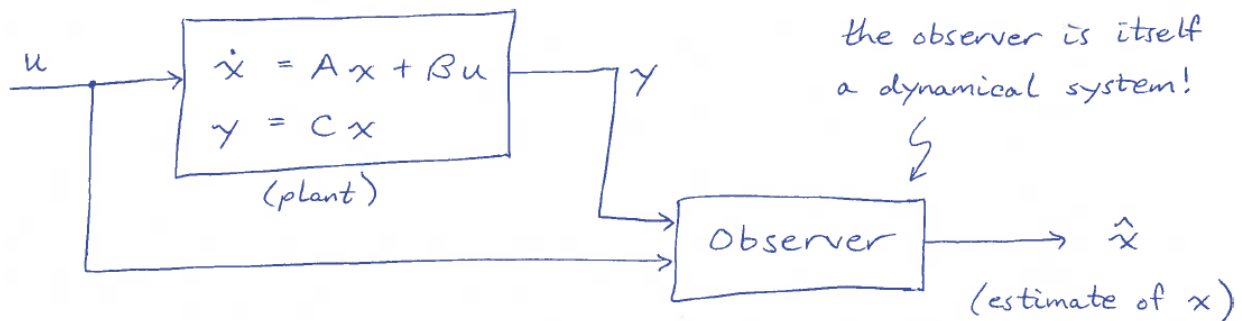
We saw previously that if an LTI system is controllable and we can measure the full state, then we can place the closed-loop poles anywhere we like using static state feedback.

Measuring the full state, however, may require many (costly) sensors, so we would like to control the system using only the output measurements $y = Cx$.

16.1 Luenberger observer

We will use the following control strategy:

- Use u and y to estimate x with an observer.
- Use the estimate of x to perform full state feedback control as if it were the actual state.



We want to design the observer such that

$$\|x(t) - \hat{x}(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty$$

for any initial condition $x_0 = x(0)$. We will use a linear observer whose structure is similar to the plant:

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu - L(y - \hat{y}) \\ \hat{y} &= C\hat{x}\end{aligned}$$

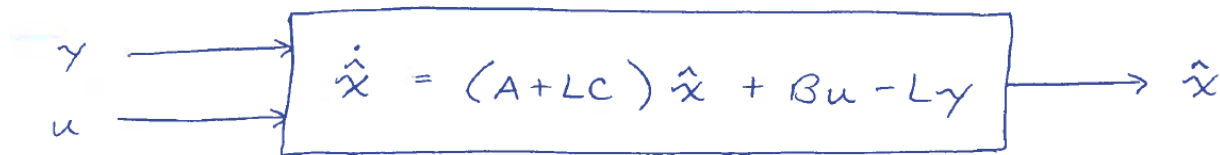
Comments

- If we initialize the observer to be the state of the plant, $\hat{x}(0) = x(0)$, then we will have $\hat{y} = y$ and $\hat{x} = x$ for all time. (But we do not know the initial state, since this is the purpose of using an observer.)
- The observer uses the same system matrices (A, B, C) as the plant, so this requires knowing the model of the plant.
- This type of observer is called a *Luenberger observer*. There are other types of observers (such as nonlinear).

The error of the observer is $e(t) = x(t) - \hat{x}(t)$. The dynamics of the error are

$$\begin{aligned}\dot{e} &= \dot{x} - \dot{\hat{x}} \\ &= (Ax + Bu) - (A\hat{x} + Bu - L(Cx - C\hat{x})) \\ &= (A + LC)e\end{aligned}$$

This is just an autonomous LTI system! For the error to converge to zero, we should choose the observer gain L such that the matrix $A + LC$ is stable (meaning that all of its eigenvalues are in the left-half complex plane).



16.2 Pole placement

Can we choose the eigenvalues of L to place the eigenvalues of $A + LC$ in the left-half plane?

Fact (Pole placement). The following are equivalent:

- (C, A) is observable
- the eigenvalues of $A + LC$ may be arbitrarily assigned by suitable choice of L , as long as they are chosen in complex conjugate pairs

Proof. Instead of proving the result from scratch, we can use our result on pole placement with controllability. Consider the following statements:

- (C, A) is observable
- (A^T, C^T) is controllable
- can arbitrarily place eigenvalues of $A^T + C^T L^T$ (where $K = L^T$)
- can arbitrarily place eigenvalues of $A + LC$

The first two statements are equivalent by duality of controllability and observability. The second and third statements are equivalent from pole placement for controllability. And finally, the last two statements are equivalent since eigenvalues are invariant under the transpose operation. Therefore, observability is equivalent to pole placement of $A + LC$. ■

There are several methods to find the observer gain L . One method is to set $L = K^T$ where

$$K = K_{\text{CCF}}(PP_{\text{CCF}}^{-1})^{-1}$$

where P is the controllability matrix of (A^T, C^T) . We can also find L using the observable canonical form,

$$\frac{b_{n-1}s^{n-1} + \dots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0} = \left(\begin{array}{cccccc|c} 0 & 0 & \dots & 0 & 0 & -a_0 & b_0 \\ 1 & 0 & \dots & 0 & 0 & -a_1 & b_1 \\ 0 & 1 & \dots & 0 & 0 & -a_2 & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & -a_{n-2} & b_{n-2} \\ 0 & 0 & \dots & 0 & 1 & -a_{n-1} & b_{n-1} \\ \hline 0 & 0 & \dots & 0 & 0 & 1 & 0 \end{array} \right)$$

which is just the transpose of the controllable canonical form.

Fact (Pole placement). Consider a realization (C, A) , and denote the characteristic polynomial of A as

$$\lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_1\lambda + a_0.$$

Then, the observer gain L such that the characteristic polynomial of $A + LC$ is

$$\lambda^n + \alpha_{n-1}\lambda^{n-1} + \dots + \alpha_1\lambda + \alpha_0$$

is given by

$$L = (Q_{\text{OCF}}^{-1}Q)^{-1}L_{\text{OCF}} \quad \text{where} \quad L_{\text{OCF}} = [a_0 - \alpha_0 \quad a_1 - \alpha_1 \quad \dots \quad a_{n-1} - \alpha_{n-1}]^T$$

is the observer gain in observable canonical form, $T = Q_{\text{OCF}}^{-1}Q$ is the state transformation matrix that transforms the system to observable canonical form, Q is the observability matrix of (C, A) , and the (inverse of) the observability matrix in observable canonical form is the Hankel matrix

$$Q_{\text{OCF}}^{-1} = \begin{bmatrix} a_1 & a_2 & \dots & a_{n-1} & 1 \\ a_2 & a_3 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}.$$

16.3 Detectability

If the system is not observable, we can still design an observer that asymptotically tracks the state if all unobservable modes are stable.

Theorem (Detectability). Given matrices $A \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{p \times n}$, the following statements are equivalent.

- (C, A) is detectable.
- (A^T, C^T) is stabilizable
- $\begin{bmatrix} \lambda I - A \\ C \end{bmatrix}$ has full column rank for all $\lambda \in \mathbb{C}$ with $\text{Re}(\lambda) \geq 0$.
- If $Av = \lambda v$ with $0 \neq v \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}$ with $\text{Re}(\lambda) \geq 0$, then $Cv \neq 0$.
- There exists a matrix L such that $A + LC$ is stable.

Example. Consider the system

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 0 & -2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} u \\ y &= [1 \ 0 \ 0] x \end{aligned}$$

The second and third states are unobservable. However, using $L = (\ell_1, \ell_2, \ell_3)$, we have that

$$A + LC = \begin{bmatrix} 1 + \ell_1 & 0 & 0 \\ 1 + \ell_2 & -1 & 1 \\ \ell_3 & 0 & -2 \end{bmatrix}$$

which has eigenvalues $1 + \ell_1$, -1 , and -2 . The unobservable modes cannot be moved, but they are stable, so the system is detectable.

Example. Consider the system

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 0 & -6 \\ 1 & 0 & -11 \\ 0 & 1 & -6 \end{bmatrix} x \\ y &= \begin{bmatrix} 0 & 1 & -3 \end{bmatrix} x\end{aligned}$$

a) Is the system observable?

The observability matrix is

$$Q = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -3 \\ 1 & -3 & 7 \\ -3 & 7 & -15 \end{bmatrix}$$

which has rank two, so the system is not observable.

b) Is the system detectable?

The eigenvalues of A are -1 , -2 , and -3 . Since all eigenvalues are stable, the system is detectable.

c) Which eigenvalues can be moved?

We can apply the PBH test for detectability to each eigenvalue of A .

- For the eigenvalue -1 ,

$$\text{rank} \begin{bmatrix} A + 1I \\ C \end{bmatrix} = \text{rank} \begin{bmatrix} 1 & 0 & -6 \\ 1 & 1 & -11 \\ 0 & 1 & -5 \\ 0 & 1 & -3 \end{bmatrix} = 3$$

which is full rank, so the eigenvalue can be moved.

- For the eigenvalue -2 ,

$$\text{rank} \begin{bmatrix} A + 2I \\ C \end{bmatrix} = \text{rank} \begin{bmatrix} 2 & 0 & -6 \\ 1 & 2 & -11 \\ 0 & 1 & -4 \\ 0 & 1 & -3 \end{bmatrix} = 3$$

which is full rank, so the eigenvalue can be moved.

- For the eigenvalue -3 ,

$$\text{rank} \begin{bmatrix} A + 3I \\ C \end{bmatrix} = \text{rank} \begin{bmatrix} 3 & 0 & -6 \\ 1 & 3 & -11 \\ 0 & 1 & -3 \\ 0 & 1 & -3 \end{bmatrix} = 2$$

which is not full rank, so the eigenvalue cannot be moved.

Example (continued).

- c) Choose the observer gain L to place all eigenvalues of $A + LC$ at -3 .

The characteristic polynomial of A is

$$\lambda^3 + 6\lambda^2 + 11\lambda + 6$$

and the desired closed-loop characteristic polynomial is

$$(\lambda + 3)^3 = \lambda^3 + 9\lambda^2 + 27\lambda + 27.$$

Therefore, the observer gain in observable canonical form is

$$L_{\text{OCF}} = \begin{bmatrix} -21 \\ -16 \\ -3 \end{bmatrix}$$

and the inverse of the observability matrix in observable canonical form is

$$Q_{\text{OCF}}^{-1} = \begin{bmatrix} 11 & 6 & 1 \\ 6 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Using the observability matrix found previously, the state transformation matrix is

$$T = Q_{\text{OCF}}^{-1}Q = \begin{bmatrix} 3 & 0 & -6 \\ 1 & 3 & -11 \\ 0 & 1 & -3 \end{bmatrix}.$$

This transformation matrix, however, is not invertible (its eigenvalues are 0, 1, and 2). This is due to the fact that the system was not observable. Since we chose the closed-loop poles to include the unobservable pole, however, the formula still holds using the pseudoinverse instead of the standard inverse, in which case the observer gain is

$$L = T^+ L_{\text{OCF}} = \begin{bmatrix} -3.7143 \\ 1.9286 \\ 1.6429 \end{bmatrix}.$$

Keep in mind that using the pseudoinverse yields an incorrect observer gain if the closed-loop poles do not include all unobservable poles (since in this case such an L does not exist).

17

The Separation Principle

The separation principle is a fundamental idea in control that appears in a variety of contexts. In this chapter, we study one instance of the separation principle which states that the design of a dynamic output feedback controller separates into the design of an observer and a static state feedback controller, both of which we have studied previously.

17.1 Interconnected systems

Before describing dynamic output feedback controllers and the separation principle, let's review how to combine systems to form more complex interconnected systems.

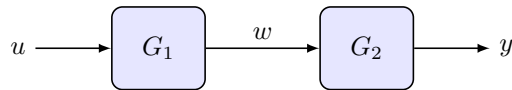
Suppose we have two systems:

$$G_1 = \left(\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array} \right) \quad \text{and} \quad G_2 = \left(\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array} \right).$$

There are various ways these two systems can be combined to form an interconnected system.

Series connection

Consider the series connection of the two systems G_1 and G_2 show below.



If we consider G_1 and G_2 as operators that map the input signal to the output signal, then the block diagram indicates that $w = G_1u$ and $y = G_2w$. Eliminating the intermediate signal w , the output is

$$y = G_2G_1u.$$

Let's now find a state-space realization of the series interconnection using the state-space realizations of the individual systems. In state-space form,

$$\begin{aligned} \dot{x}_1 &= A_1x_1 + B_1u \\ w &= C_1x_1 + D_1u \end{aligned} \quad \text{and} \quad \begin{aligned} \dot{x}_2 &= A_2x_2 + B_2w \\ y &= C_2x_2 + D_2w \end{aligned}$$

Combining the systems, a state-space realization of the series connection is

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} A_1 & 0 \\ B_2C_1 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2D_1 \end{bmatrix} u \\ y &= [D_2C_1 \quad C_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + (D_2D_1)u \end{aligned}$$

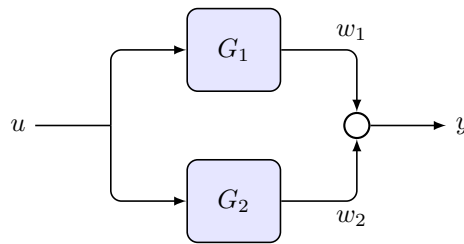
Here, we chose the state of the interconnected system to be $x = (x_1, x_2)$. We also could have chosen the state in the reverse order, (x_2, x_1) , which would have lead to a similar state-space realization. Therefore, the series interconnection has either of the following state-space realizations:

$$G_2G_1 = \left[\begin{array}{cc|c} A_1 & 0 & B_1 \\ B_2C_1 & A_2 & B_2D_1 \\ \hline D_2C_1 & C_2 & D_2D_1 \end{array} \right] = \left[\begin{array}{cc|c} A_2 & B_2C_1 & B_2D_1 \\ 0 & A_1 & B_1 \\ \hline C_2 & D_2C_1 & D_2D_1 \end{array} \right],$$

where the first realization has state (x_1, x_2) and the second has state (x_2, x_1) .

Parallel connection

Consider the parallel connection of the two systems G_1 and G_2 show below.



If we consider G_1 and G_2 as operators that map the input signal to the output signal, then the block diagram indicates that $w_1 = G_1u$, $w_2 = G_2u$, and $y = w_1 + w_2$. Eliminating the intermediate signals w_1 and w_2 , the output is

$$y = (G_1 + G_2)u.$$

Let's now find a state-space realization of the parallel interconnection using the state-space realizations of the individual systems. In state-space form,

$$\begin{aligned} \dot{x}_1 &= A_1x_1 + B_1u & \text{and} & & \dot{x}_2 &= A_2x_2 + B_2u \\ w_1 &= C_1x_1 + D_1u & & & w_2 &= C_2x_2 + D_2u \end{aligned}$$

Combining the systems, a state-space realization of the parallel connection is

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y &= [C_1 \quad C_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + (D_1 + D_2)u \end{aligned}$$

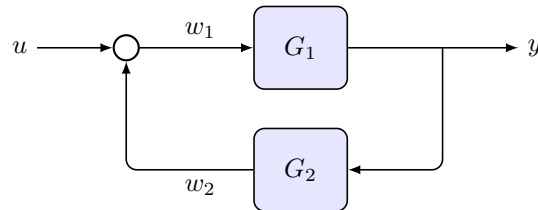
Here, we chose the state of the interconnected system to be $x = (x_1, x_2)$. We also could have chosen the state in the reverse order, (x_2, x_1) , which would have lead to a similar state-space realization. Therefore, the parallel interconnection has either of the following state-space realizations:

$$G_1 + G_2 = \left[\begin{array}{cc|c} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ \hline C_1 & C_2 & D_1 + D_2 \end{array} \right] = \left[\begin{array}{cc|c} A_2 & 0 & B_2 \\ 0 & A_1 & B_1 \\ \hline C_2 & C_1 & D_2 + D_1 \end{array} \right],$$

where the first realization has state (x_1, x_2) and the second has state (x_2, x_1) .

Feedback connection

Consider the feedback connection of the two systems G_1 and G_2 show below.



If we consider G_1 and G_2 as operators that map the input signal to the output signal, then the block diagram indicates that $w_1 = u + w_2$, $w_2 = G_2 y$, and $y = G_1 w_1$. Eliminating the intermediate signals w_1 and w_2 , the output is

$$y = (I - G_1 G_2)^{-1} G_1 u.$$

Remark (MIMO systems). For multi-input multi-output systems, G_1 and G_2 are *matrices* of transfer functions, so $G_1 G_2 \neq G_2 G_1$ in general, as matrices do not commute. When the systems are SISO, however, the order of multiplication and division are irrelevant, so we can write the output as

$$y = \frac{G_1}{1 - G_1 G_2} u.$$

In general, however, the inverse must be on the left-hand side. ■

Let's now find a state-space realization of the feedback interconnection using the state-space realizations of the individual systems. In state-space form,

$$\begin{aligned} \dot{x}_1 &= A_1 x_1 + B_1 w_1 & \text{and} & & \dot{x}_2 &= A_2 x_2 + B_2 y \\ y &= C_1 x_1 + D_1 w_1 & & & w_2 &= C_2 x_2 + D_2 y \end{aligned}$$

To combine the state-space realizations, first expand the output as

$$\begin{aligned} y &= C_1 x_1 + D_1 w_1, \\ &= C_1 x_1 + D_1 (u + w_2), \\ &= C_1 x_1 + D_1 (u + C_2 x_2 + D_2 y). \end{aligned}$$

The output y now occurs on both sides of the equation due to the feedback loop. Solving for the output, we obtain

$$y = \Delta^{-1} (C_1 x_1 + D_1 C_2 x_2 + D_1 u) \quad \text{where} \quad \Delta = I - D_1 D_2.$$

Now that we have the output in terms of the states and input, we can expand the first state equation as

$$\begin{aligned} \dot{x}_1 &= A_1 x_1 + B_1 w_1, \\ &= A_1 x_1 + B_1 (u + w_2), \\ &= A_1 x_1 + B_1 (u + C_2 x_2 + D_2 y), \\ &= A_1 x_1 + B_1 (u + C_2 x_2 + D_2 \Delta^{-1} (C_1 x_1 + D_1 C_2 x_2 + D_1 u)). \end{aligned}$$

We now have the state update in terms of the states x_1 and x_2 and the input u . Similarly, the second state equation expands as

$$\begin{aligned}\dot{x}_2 &= A_2x_2 + B_2y, \\ &= A_2x_2 + B_2\Delta^{-1}(C_1x_1 + D_1C_2x_2 + D_1u),\end{aligned}$$

which also only depends on the states and the input. Therefore, a state-space realization of the feedback connection is

$$\begin{aligned}\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} A_1 + B_1D_2\Delta^{-1}C_1 & B_1C_2 + B_1D_2\Delta^{-1}D_1C_2 \\ B_2\Delta^{-1}C_1 & A_2 + B_2\Delta^{-1}D_1C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 + B_1D_2\Delta^{-1}D_1 \\ B_2\Delta^{-1}D_1 \end{bmatrix} u \\ y &= [\Delta^{-1}C_1 \quad \Delta^{-1}D_1C_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + (\Delta^{-1}D_1)u\end{aligned}$$

Therefore, the feedback interconnection has the following state-space realization with state (x_1, x_2) :

$$(I - G_1G_2)^{-1}G_1 = \left[\begin{array}{cc|c} A_1 + B_1D_2\Delta^{-1}C_1 & B_1C_2 + B_1D_2\Delta^{-1}D_1C_2 & B_1 + B_1D_2\Delta^{-1}D_1 \\ B_2\Delta^{-1}C_1 & A_2 + B_2\Delta^{-1}D_1C_2 & B_2\Delta^{-1}D_1 \\ \hline \Delta^{-1}C_1 & \Delta^{-1}D_1C_2 & \Delta^{-1}D_1 \end{array} \right].$$

Remark (Well-posedness). The feedback connection requires that $\Delta = I - D_1D_2$ is invertible. This is needed for the system to be *well posed*, meaning that for any input signal u , there exist signals (w_1, w_2, y) that satisfy the interconnection. A system that is not well posed is like an inconsistent system of equations, such as $\{x + y = 1, 2x + 2y = 3\}$. ■

Example (Well-posedness). Consider the feedback interconnection with systems

$$G_1 = \frac{s-1}{s+1} \quad \text{and} \quad G_2 = \frac{s+3}{s+2}.$$

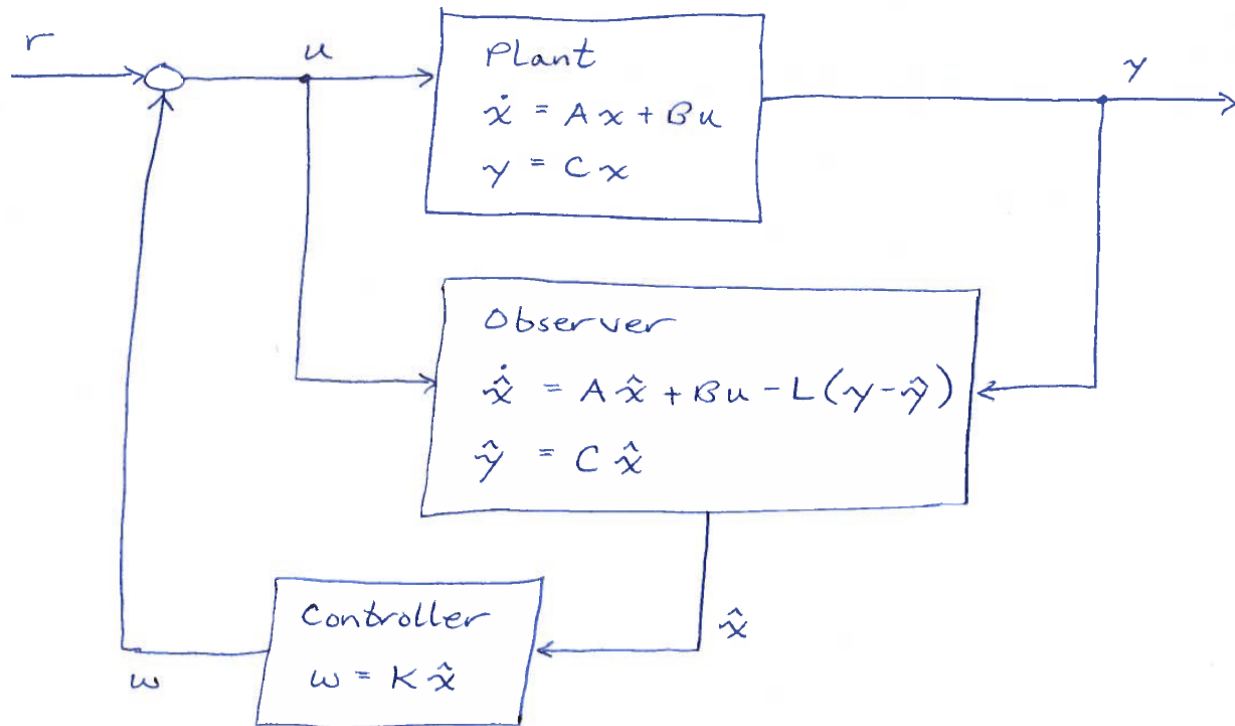
The transfer function of the closed-loop system is

$$\frac{Y(s)}{U(s)} = (I - G_1G_2)^{-1}G_1 = \frac{\frac{s-1}{s+1}}{1 - \frac{s-1}{s+1} \cdot \frac{s+3}{s+2}} = \frac{(s-1)(s+2)}{s+6}$$

The transfer function is not proper, which implies that the system is noncausal. This system is not well posed since $D_1 = 1$ and $D_2 = 1$, so $I - D_1D_2 = 0$, which is not invertible.

17.2 Observer with static state feedback

We now return to the problem of using an observer with static state feedback. Let's consider what happens when we use an observer to estimate the state and then apply static state feedback to the estimate as shown in the following block diagram.



To find the closed-loop equations from the reference input r to the measured output y , eliminate the controlled input u to obtain

$$\dot{x} = Ax + B(K\hat{x} + r)$$

and

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + B(K\hat{x} + r) - LC(x - \hat{x}), \\ &= (A + BK + LC)\hat{x} - LCx + Br.\end{aligned}$$

Therefore, the combined system is

$$\begin{aligned}\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} &= \begin{bmatrix} A & BK \\ -LC & A + BK + LC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} r \\ y &= [C \ 0] \begin{bmatrix} x \\ \hat{x} \end{bmatrix}\end{aligned}$$

Stability of the closed-loop system depends on the eigenvalues of the closed-loop system matrix, which are not easy to find in this form. Instead of using the observer state \hat{x} , let's use a state transformation to write the closed-loop dynamics in terms of the error $e(t) = x(t) - \hat{x}(t)$. This is equivalent to applying a state transformation with

$$T = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} = T^{-1}.$$

Applying the state transformation yields

$$\begin{aligned}\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} &= \begin{bmatrix} A + BK & -BK \\ 0 & A + LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} r \\ y &= [C \ 0] \begin{bmatrix} x \\ e \end{bmatrix}\end{aligned}$$

Therefore, the eigenvalues of the closed-loop dynamics are those of $A + BK$ and $A + LC$. If the plant (A, B, C) is minimal (meaning that (A, B) is controllable and (C, A) is observable), then we can place all eigenvalues of the closed-loop system wherever we like by suitable choices of K and L .

Separation principle: The closed-loop poles of a system in feedback with an observer-based controller separate into those of the observer and those of the plant with static state feedback.

Comments

- The closed-loop dynamics have the form of the controllable decomposition, from which we can conclude that the observer error is uncontrollable (it is not affected by the reference or the state of the plant). If we cancel this uncontrollable state, the closed-loop dynamics simplify to

$$\left[\begin{array}{cc|c} A + BK & -BK & B \\ 0 & A + LC & 0 \\ \hline C & 0 & 0 \end{array} \right] = \left[\begin{array}{c|c} A + BK & B \\ \hline C & 0 \end{array} \right],$$

which is precisely the closed-loop map we would have obtained if we had used full state feedback! While the observer error is uncontrollable, we cannot cancel these uncontrollable modes since the plant and controller are physically separate systems.

- The eigenvalues of $A + BK$ dictate how fast the output y tracks the reference r when the observer is initialized with the state of the plant.
- The eigenvalues of $A + LC$ dictate how fast the estimate \hat{x} of the observer tracks the plant state x , which is independent of $A + BK$.
- The observer dynamics are

$$\dot{\hat{x}} = (A + BK + LC)\hat{x} + Br - Ly,$$

which may be unstable, even if $A + B$ and $A + LC$ are both stable. For example, the observer must be unstable to track the state of an unstable plant.

18

Linear–Quadratic Regulator

18.1 Motivation

We have now seen how to use dynamic output feedback to place the closed-loop poles wherever we want (when the system is minimal) by combining static state feedback with an observer.

We have some intuition into where to place the poles from our analysis of second-order systems. But this only tells us how the pole locations affect the transient response and says nothing about the energy required to control the system.

Consider the continuous-time LTI system $\dot{x} = Ax + Bu$. Using static state feedback $u = Kx$, the energy of the control input is

$$\begin{aligned}\text{control energy} &= \int_0^\infty \|u(t)\|^2 dt \\ &= \int_0^\infty x(t)^\top K^\top K x(t) dt \\ &= x_0^\top \left(\int_0^\infty e^{(A+BK)^\top t} K^\top K e^{(A+BK)t} dt \right) x_0 \\ &= x_0^\top P x_0\end{aligned}$$

where P is the solution to the Lyapunov equation

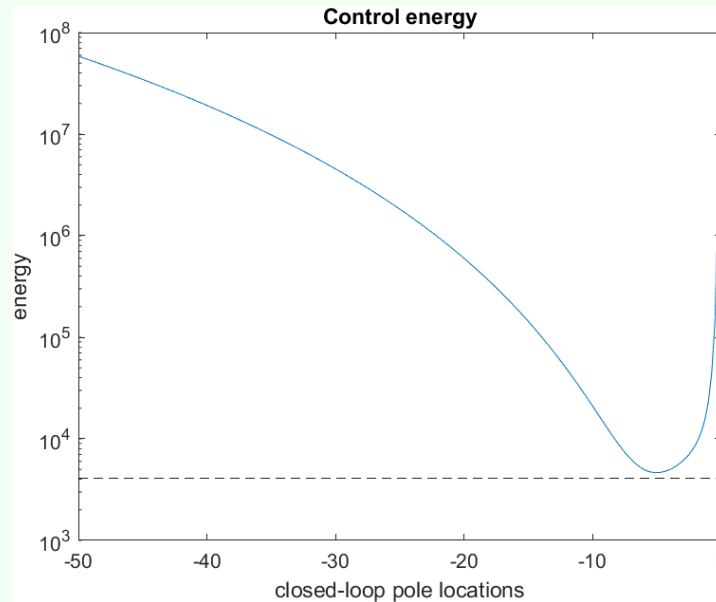
$$(A + BK)^\top P + P(A + BK) + K^\top K = 0.$$

The energy therefore is a quadratic function of the initial state. The initial state that requires the most energy of the control input is the eigenvector of P corresponding to its largest eigenvalues (see Matlab demo).

Example (Control energy). Consider the continuous-time LTI system

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 78 & -20 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u.$$

Suppose we use static state feedback $u = Kx$ to place all three closed-loop poles at the same (real) location. The following plot shows the resulting control energy as a function of the closed-loop pole locations.



The minimum control energy is achieved for pole locations near -5 . We can do slightly better, however, if we design the static state feedback gain to minimize the control energy. The energy of this optimal control input is the dashed line. In this chapter, we learn how to design static state feedback controllers based on what we want them to optimize.

Instead of pole placement, we can choose the controller to optimize some cost. Some typical costs are the tracking error

$$J_x = \int_0^{\infty} x(t)^T Q x(t) dt$$

and the control effort

$$J_u = \int_0^{\infty} u(t)^T R u(t) dt.$$

These are quadratic functions of the state and input, where Q and R are symmetric matrices that characterize how much to weight the different components of the state and input in the cost. For the cost to be sensible, Q and R must be positive semidefinite, as otherwise there would be states or inputs that could make the cost arbitrarily negative. Moreover, we typically assume that R is strictly positive definite so that all inputs are penalized, as otherwise some inputs could be made arbitrarily large without affecting the cost. For example, we could set $R = \mu I$ for some $\mu > 0$ so that the value of μ allows us to adjust how much the input energy affects the cost.

18.2 Problem description

The linear-quadratic regulator (LQR) problem is to choose the input signal u to minimize the cost

$$J(x_0) = \int_0^{\infty} (x(t)^\top Q x(t) + u(t)^\top R u(t)) dt$$

subject to the dynamics

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0.$$

Comments

- The name LQR comes from the fact that the dynamics are *linear*, the cost is *quadratic*, and this is a *regulation* problem.
- This formulation is the infinite-horizon continuous-time LQR problem. There is a similar version in discrete time (in which the integral in the cost becomes a summation) and for a finite horizon (in which the upper limit of integration becomes some finite time $T > 0$).
- We might expect the optimal (in the sense of minimizing J) control strategy to depend on the initial state x_0 . It turns out that the optimal cost J depends on x_0 , but the control input $u(t)$ that optimizes J does *not* depend on x_0 .

18.3 Solution via completing the square

There are many ways to solve the LQR problem. Here, we use an approach based on simple algebra and calculus. First, let's review how to complete the square.

Completing the square for scalars

Consider the function

$$f(x, u) = qx^2 + 2sxu + ru^2$$

with parameters $q \geq 0$ and $r > 0$. To minimize the function with respect to u , we can complete the square as follows:

$$\begin{aligned} f(x, u) &= qx^2 + r\left(\frac{2s}{r}xu + u^2\right) \\ &= qx^2 + r\left(u^2 + 2\left(\frac{sx}{r}\right)u + \left(\frac{sx}{r}\right)^2\right) - r\left(\frac{sx}{r}\right)^2 \\ &= \left(q - \frac{s^2}{r}\right)x^2 + r\left(u + \frac{sx}{r}\right)^2 \end{aligned}$$

The first term does not depend on u , while the second term is nonnegative and can be made zero with $u = -sx/r$. To summarize, the optimal u and the corresponding value of the function is

$$u_{\text{opt}} = -\frac{sx}{r} \quad \text{and} \quad f(x, u_{\text{opt}}) = \left(q - \frac{s^2}{r}\right)x^2.$$

We could also obtain this by setting the derivative equal to zero:

$$0 = \frac{d}{du} f(x, u) = 2sx + 2ru \quad \implies \quad u = -\frac{sx}{r}.$$

Example (Completing the square). $17x^2 + 12ux + 3u^2 = 5x^2 + 3(u + 2x)^2$

Completing the square for vectors

Similar to before, we can also complete the square to minimize a vector function. Consider the function

$$f(x, u) = x^T Q x + 2x^T S u + u^T R u$$

with parameters $Q \succeq 0$ and $R \succ 0$. To minimize the function with respect to u , we can complete the square as follows:

$$\begin{aligned} f(x, u) &= x^T Q x + (u^T R u + u^T S^T x + x^T S u) \\ &= x^T Q x + (u + R^{-1} S^T x)^T R (u + R^{-1} S^T x) - x^T S R^{-1} S^T x \\ &= x^T (Q - S R^{-1} S^T) x + (u + R^{-1} S^T x)^T R (u + R^{-1} S^T x). \end{aligned}$$

Once again, the first term does not depend on u , while the second term is nonnegative and can be made zero with $u = -R^{-1} S^T x$. To summarize, the optimal u and the corresponding value of the function is

$$u_{\text{opt}} = -R^{-1} S^T x \quad \text{and} \quad f(x, u_{\text{opt}}) = x^T (Q - S R^{-1} S^T) x.$$

We could also obtain this by setting the derivative equal to zero using vector calculus:

$$0 = \frac{d}{du} f(x, u) = 2S^T x + 2R u \quad \implies \quad u = -R^{-1} S^T x.$$

LQR solution

We now solve the LQR problem by completing the square. Assuming the closed-loop system is stable (so that $x(t) \rightarrow 0$ as $t \rightarrow \infty$) and introducing a symmetric matrix P , we have from the fundamental theorem of calculus that

$$x_0^T P x_0 + \int_0^\infty \frac{d}{dt} (x^T P x) dt = 0$$

where the derivative term inside the integral expands as

$$\frac{d}{dt} (x^T P x) = \dot{x}^T P x + x^T P \dot{x} = (Ax + Bu)^T P x + x^T P (Ax + Bu).$$

Since this term is zero, we can add it to the cost without changing its value. Doing so, we obtain

$$\begin{aligned} J &= \int_0^\infty (x^T Q x + u^T R u) dt \\ &= x_0^T P x_0 + \int_0^\infty \left(\frac{d}{dt} (x^T P x) + x^T Q x + u^T R u \right) dt \\ &= x_0^T P x_0 + \int_0^\infty \left((Ax + Bu)^T P x + x^T P (Ax + Bu) + x^T Q x + u^T R u \right) dt \\ &= x_0^T P x_0 + \int_0^\infty \left(x^T (A^T P + P A + Q) x + 2x^T P B u + u^T R u \right) dt. \end{aligned}$$

The term inside the integral is quadratic in the state and input. Completing the square,

$$J = x_0^T P x_0 + \int_0^\infty \left[x^T (A^T P + P A + Q - P B R^{-1} B^T P) x + (u + R^{-1} B^T P x)^T R (u + R^{-1} B^T P x) \right] dt.$$

The first and second terms do not depend on the input u , and the third term is minimized by the input $u = -R^{-1}B^T Px$. While the second term does not depend on the input, it does depend on the parameter P that we introduced (and therefore get to choose). Since this term is The following result shows that we can make this term zero for appropriate choice of P .

Definition (Algebraic Riccati equation). Given matrices A , B , Q , and R , the matrix equation

$$A^T P + PA + Q - PBR^{-1}B^T P = 0$$

in the matrix variable P is called the (continuous-time) *algebraic Riccati equation* (ARE).

The ARE is a matrix version of the quadratic equation. There is no equivalent quadratic formula, but the equation can be solved efficiently.

Theorem. Suppose $Q \succeq 0$, $R \succ 0$, (A, B) is stabilizable, and (Q, A) is detectable. Then,

- the ARE has a unique solution P that is symmetric and positive definite, and
- the matrix $A - BR^{-1}B^T P$ is Hurwitz (has all eigenvalues in the left-half plane) for this P .

Under the conditions in the theorem, the solution to the LQR problem is the static state feedback controller

$$u(t) = Kx(t) \quad \text{with} \quad K = -R^{-1}B^T P,$$

where P is the unique positive definite solution to the algebraic Riccati equation. The optimal cost is the quadratic function of the initial state,

$$J = x_0^T P x_0.$$

Moreover, the closed-loop system matrix $A + BK$ is stable.

19

Kalman Filter

The Kalman filter, also known as the linear-quadratic estimator (LQE), is a recursive algorithm used for estimating the state of a dynamic system from noisy measurements. It operates by predicting the system's next state and corresponding uncertainty based on a mathematical model, then updating this prediction using new observations to refine the estimate. The filter assumes the system is governed by linear equations and that noise in the dynamics and measurements follows a Gaussian distribution. At each iteration, it computes a weighted average of the predicted state and the measured state, with weights determined by their respective uncertainties. The Kalman filter is widely used in fields such as robotics, navigation, and signal processing due to its ability to provide accurate, real-time estimates even in the presence of noise.

19.1 Definition

Definition (Continuous-time steady-state Kalman filter). Given an LTI system (A, B, C, D) and symmetric matrices W and V , the continuous-time steady-state Kalman filter is the state observer

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) - L(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C\hat{x}(t)\end{aligned}$$

where L is the Kalman gain

$$L = -PC^T V^{-1}$$

and the symmetric matrix P is the solution to the continuous-time algebraic Riccati equation

$$AP + PA^T + W - PC^T V^{-1} CP = 0.$$

Definition (Discrete-time steady-state Kalman filter). Given an LTI system (A, B, C, D) and symmetric matrices W and V , the discrete-time steady-state Kalman filter is the state observer

$$\begin{aligned}\hat{x}_{k+1} &= A\hat{x}_k + Bu_k - L(y_k - \hat{y}_k) \\ \hat{y}_k &= C\hat{x}_k\end{aligned}$$

where L is the Kalman gain

$$L = -\Sigma C^T (C\Sigma C^T + V)^{-1}$$

and the symmetric matrix Σ is the solution to the discrete-time algebraic Riccati equation

$$A\Sigma A^T - \Sigma + W - A\Sigma C^T (C\Sigma C^T + V)^{-1} C\Sigma A^T = 0.$$

Note that the Kalman filter depends on the problem data (A, C, W, V) and *not* the input matrix B . In the following section, we will see how to interpret the parameters W and V .

19.2 Interpretations

We now describe several interpretations of the Kalman filter as an optimal state observer. In each case, we consider an LTI system in which the state and output equations are perturbed:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + w(t) & \text{or} & & x_{k+1} &= Ax_k + Bu_k + w_k \\ y(t) &= Cx(t) + v(t) & & & y_k &= Cx_k + v_k \end{aligned}$$

The signal w is called the *process noise* and represents disturbances to the system such as unmodeled dynamics like wind or friction. The signal v is called the *measurement noise* and represents errors in the measurements such as round-off errors or sensor noise.

Deterministic interpretation

Suppose we have measured the input $u(\tau)$ and output $y(\tau)$ for $-\infty < \tau \leq t$. Also, suppose the matrices W and V are positive definite (and therefore invertible). The output $\hat{x}(t)$ of the Kalman filter is then the optimal estimator of the state $x(t)$ along with the process noise $w(t)$ and measurement noise $v(t)$ that minimize the quadratic cost

$$\int_{-\infty}^t (w(\tau)^T W^{-1} w(\tau) + v(\tau)^T V^{-1} v(\tau)) d\tau \quad \text{or} \quad \sum_{m=-\infty}^k (w_m^T W^{-1} w_m + v_m^T V^{-1} v_m)$$

subject to the system dynamics. We use the inverses of W and V in the cost as a matter of convention (so that the solution is consistent with the stochastic interpretation). In general, “larger” W results in “larger” w , and similarly for V and v .

Stochastic interpretation

A *stochastic process* is a signal whose value at each point in time is a random variable. Stochastic processes often represent noise in the system. *White* noise means the distribution of the signal is constant in time, and *zero-mean* means that the expectation is zero.

Suppose the system and the noise satisfy the following.

- The initial state is a Gaussian random variable with known distribution.

$$x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$$

- At each iteration, the process and measurement noise are also Gaussian random variables with zero mean and known covariance.

$$w_t \sim \mathcal{N}(0, W) \quad \text{and} \quad v_t \sim \mathcal{N}(0, V)$$

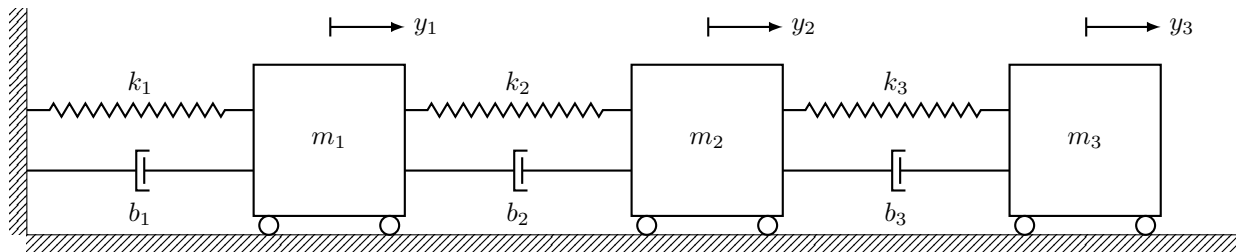
- The initial state, process noise, and measurement noise are all mutually independent.

$$x_0, w_0, w_1, \dots, v_0, v_1, \dots \quad \text{mutually independent}$$

Given the control signal u and measured signal y , the output \hat{x} of the Kalman filter is the minimum mean square error (MMSE) estimator of the state x . This means that, at each time t , the estimate $\hat{x}(t)$ minimizes the estimation error $\mathbb{E} \|x(t) - \hat{x}(t)\|^2$. Moreover, the Kalman filter is a recursive algorithm whose complexity does not grow over time.

19.3 Example

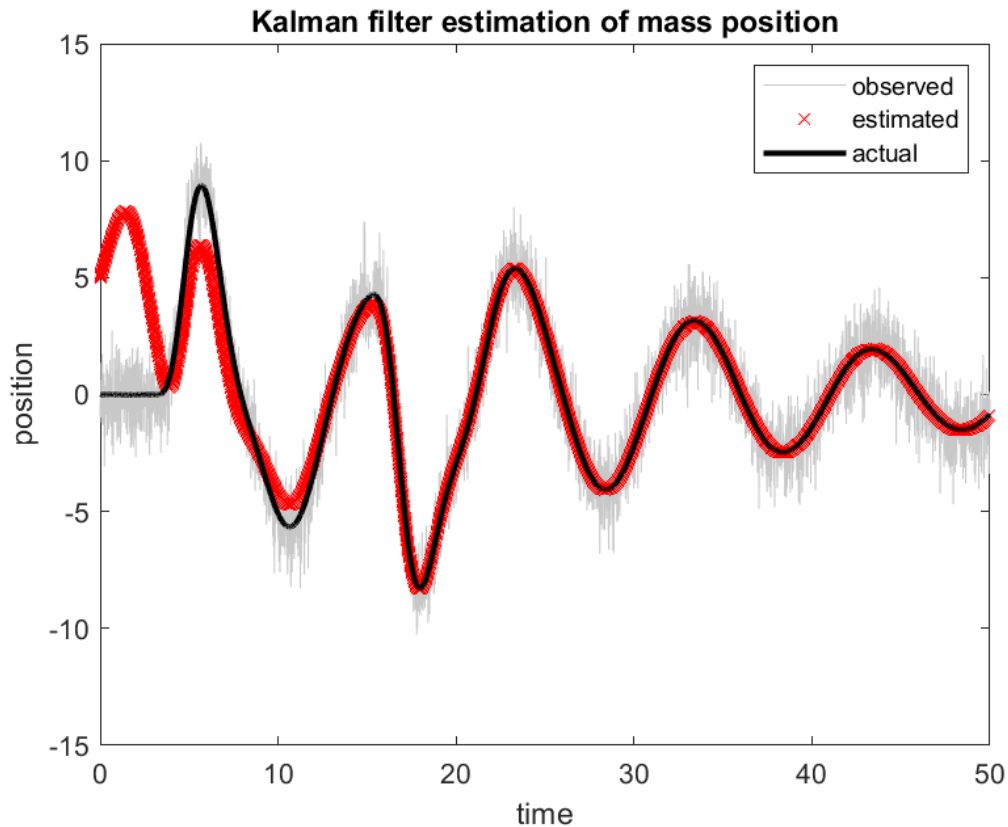
Consider the spring–mass–damper system shown below with three masses each with mass $m = 1$, damping coefficient $b = 0.5$, and spring constant $k = 1$.



Suppose we apply a force to the first mass and measure the position of the third mass. Suppose the dynamics are perturbed by random noise with process covariance $W = 0.1I_n$ and measurement covariance $V = I_p$, where $n = 6$ is the number of states and $p = 1$ is the number of measured outputs. The corresponding Kalman gain is

$$L = -(0.1213 \quad 0.2441 \quad 0.3750 \quad 0.0130 \quad 0.0338 \quad 0.0203)^T.$$

Suppose input force on the first mass is zero except for a value of one hundred on the time interval $[3, 3.2]$ and negative one hundred on $[15, 15.2]$. The measured, estimated, and actual position of the third mass are shown below.



Despite having noisy measurements (light gray), the estimate from the Kalman filter (red) converges to the actual position (black) of the third mass.

19.4 Duality

The problems of optimal estimation and optimal control are duals of each other. In particular, each problem can be transformed into the other, so the solution to one problem can be used to compute the solution to the other.

To observe this duality, recall that (in continuous time) the optimal control gain is

$$K = -R^{-1}B^T P \quad \text{where} \quad A^T P + PA + Q - PBR^{-1}B^T P = 0.$$

Similarly, the optimal estimator gain is

$$L = -\Sigma C^T V^{-1} \quad \text{where} \quad A\Sigma + \Sigma A^T + W - \Sigma C^T V^{-1} C \Sigma = 0.$$

These solutions are quite similar. Denote the solutions to the Kalman filtering and LQR problems as

$$L = \text{KF}(A, C, W, V) \quad \text{and} \quad K = \text{LQR}(A, B, Q, R).$$

Then we have the following results:

$$L^T = \text{LQR}(A^T, C^T, W, V) \quad \text{and} \quad K^T = \text{KF}(A^T, B^T, Q, R).$$

Fact (Duality). The following are equivalent:

- The system (A, B) with cost matrices (Q, R) has optimal control gain K with solution P .
- The system (B^T, A^T) with covariance matrices (Q, R) has Kalman gain K^T with solution P .

Moreover, the following are also equivalent:

- The system (C, A) with covariance matrices (W, V) has Kalman gain L with solution Σ .
- The system (A^T, C^T) with cost matrices (W, V) has optimal control gain L^T with solution Σ .

19.5 Computation

Using that optimal estimation is dual to optimal control, we obtain the following characterization of the algebraic Riccati equation which implies that, under mild conditions, the observer dynamics are stable.

Theorem. Suppose $W \succeq 0$, $V \succ 0$, (A, C) is detectable, and (A, W) is stabilizable. Then,

- the ARE has a unique solution Σ that is symmetric and positive definite, and
- the matrix $A + LC$ with the Kalman gain L is stable for this solution.

Comments

- To compute the solution to the algebraic Riccati equation in MATLAB, you can use the functions `icare` and `idare`. Equivalently, you can use the functions `lqr` and `dlqr` along with the fact that the Kalman filter is dual to the LQR problem. Whenever using MATLAB functions, be sure to check the documentation to see how to format the inputs (and outputs) properly.
- The discrete-time algebraic Riccati equation can also be solved by iteration. In particular, the following sequence converges to the unique positive definite solution when (C, A) is observable:

$$\Sigma_{k+1} = A\Sigma_k A^T + W - A\Sigma_k C^T (C\Sigma_k C^T + V)^{-1} C\Sigma_k A^T \quad \text{with} \quad \Sigma_0 = W.$$

19.6 Derivations

We now show how to construct the Kalman filter as the optimal state estimator in the stochastic setting in discrete time.

To produce an estimate of the state, we will construct a probability distribution of the state and update this distribution over time (both as the system evolves and as we make measurements).

Definition (Belief). The belief $b_{t,s}$ is the probability distribution of the state at time t given the inputs and measurements up to time s .

$$b_{t,s} = \text{Prob}(x_t \mid u_0, u_1, \dots, u_s, y_0, y_1, \dots, y_s)$$

By assumption, the initial belief is the Gaussian random variable $b_{0,0} \sim \mathcal{N}(\mu_0, \Sigma_0)$. Under the given assumptions (linear dynamics and Gaussian noise), it turns out that the belief at any time is also Gaussian. This is precisely why the Kalman filter is so useful — we can represent the belief using only its mean and covariance, which are finite-dimensional quantities. When these assumptions do not hold (the dynamics are nonlinear or the noise is non-Gaussian), the belief is a more general probability distribution that is much more difficult to represent!

We denote the mean and covariance of the belief $b_{t,s}$ as $\mu_{t,s}$ and $\Sigma_{t,s}$, respectively. Since the belief at each time is Gaussian (as we will show), we have that

$$b_{t,s} \sim \mathcal{N}(\mu_{t,s}, \Sigma_{t,s})$$

We can compute the belief at any time using induction. Suppose that $b_{t,t-1} \sim \mathcal{N}(\mu_{t,t-1}, \Sigma_{t,t-1})$. We will construct the belief at the next time in two steps: first we will include the measurement y_t to find $b_{t,t}$, and then we will use the state update to find $b_{t+1,t}$.

- a) **Measurement update.** First, we use the belief $b_{t,t-1}$ of the state *before* applying the measurement y_t to find the belief $b_{t,t}$ after the measurement has been applied. Since the measurement is correlated with the true state, the belief $b_{t,t}$ will have less uncertainty than the belief $b_{t,t-1}$ before the measurement update.

Since the measurement noise is independent of the previous process noise, measurement noise, and initial state (by assumption), the belief and measurement noise have the joint distribution

$$\begin{bmatrix} b_{t,t-1} \\ v_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{t,t-1} \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_{t,t-1} & 0 \\ 0 & V \end{bmatrix}\right).$$

The measurement is an affine function of the belief and measurement noise,

$$\begin{bmatrix} b_{t,t-1} \\ y_t \end{bmatrix} = \begin{bmatrix} I & 0 \\ C & I \end{bmatrix} \begin{bmatrix} b_{t,t-1} \\ v_t \end{bmatrix} + \begin{bmatrix} 0 \\ D \end{bmatrix} u_t.$$

To construct the update, we will use the following result, which says that an affine transformation applied to a Gaussian random variable is also Gaussian.

Proposition. If $x \sim \mathcal{N}(\mu, \Sigma)$, then $Ax + b \sim \mathcal{N}(A\mu + b, A\Sigma A^\top)$. ■

Therefore, the joint distribution of the belief and the measurement is

$$\begin{bmatrix} b_{t,t-1} \\ y_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{t,t-1} \\ C\mu_{t,t-1} + Du_t \end{bmatrix}, \begin{bmatrix} \Sigma_{t,t-1} & \Sigma_{t,t-1}C^\top \\ C\Sigma_{t,t-1} & C\Sigma_{t,t-1}C^\top + V \end{bmatrix}\right).$$

The belief after the measurement is the previous belief conditioned on the measurement, $b_{t,t} = b_{t,t-1} \mid y_t$. We can use the following result to compute the conditional expectation of two jointly Gaussian random variables.

Proposition.

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix}\right) \implies x | y \sim \mathcal{N}(\mu_x + \Sigma_{xy}\Sigma_y^{-1}(y - \mu_y), \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{yx})$$

■

Therefore, the belief after the measurement update is

$$b_{t,t} \sim \mathcal{N}(\mu_{t,t-1} + \Sigma_{t,t-1}C^\top(C\Sigma_{t,t-1}C^\top + V)^{-1}(y_t - C\mu_{t,t-1} - Du_t), \Sigma_{t,t-1} - \Sigma_{t,t-1}C^\top(C\Sigma_{t,t-1}C^\top + V)^{-1}C\Sigma_{t,t-1}).$$

Denote the updated mean as $\mu_{t,t}$ and covariance as $\Sigma_{t,t}$.

- b) **State update.** Now that we know the belief $b_{t,t} \sim \mathcal{N}(\mu_{t,t}, \Sigma_{t,t})$, we will construct the belief after the state update. Since the state update contains noise, the belief $b_{t+1,t}$ will have more uncertainty than the belief $b_{t,t}$ before the state update.

Using the formula for an affine transformation applied to a Gaussian random variable, the state update without noise is

$$Ax_t + Bu_t \sim \mathcal{N}(A\mu_{t,t} + Bu_t, A\Sigma_{t,t}A^\top).$$

The actual state update also has additive noise. To include the noise, we will use the following result.

Proposition. If $x \sim \mathcal{N}(\mu_x, \Sigma_x)$ and $y \sim \mathcal{N}(\mu_y, \Sigma_y)$ are independent, then $x + y \sim \mathcal{N}(\mu_x + \mu_y, \Sigma_x + \Sigma_y)$.

■

Using the distribution of the process noise, the full state update (with process noise) is then

$$b_{t+1,t} = Ax_t + Bu_t + w_t \sim \mathcal{N}(A\mu_{t,t} + Bu_t, A\Sigma_{t,t}A^\top + W).$$

Denote the updated mean as $\mu_{t+1,t}$ and covariance as $\Sigma_{t+1,t}$.

To summarize, we have shown that if the belief $b_{t,t-1}$ is Gaussian, then the belief $b_{t+1,t}$ is also Gaussian. Since the initial belief is Gaussian (by assumption), we have from induction that the belief remains Gaussian at each time step, and the above formulas indicate how the mean and covariance evolve over time.

Define the mean and covariance of the belief at time t given information up to time $t-1$ as

$$\hat{x}_t := \mu_{t,t-1} \quad \text{and} \quad \Sigma_t := \Sigma_{t,t-1}.$$

The above equations then simplify to the following recursion for the mean:

$$\begin{aligned} \hat{x}_{t+1} &= \mu_{t+1,t} \\ &= A\mu_{t,t} + Bu_t \\ &= A[\mu_{t,t-1} + \Sigma_{t,t-1}C^\top(C\Sigma_{t,t-1}C^\top + V)^{-1}(y_t - C\mu_{t,t-1} - Du_t)] + Bu_t \\ &= A\hat{x}_t + Bu_t + A\Sigma_tC^\top(C\Sigma_tC^\top + V)^{-1}(y_t - C\hat{x}_t - Du_t) \end{aligned}$$

and the following recursion for the covariance:

$$\begin{aligned} \Sigma_{t+1} &= \Sigma_{t+1,t} \\ &= A\Sigma_{t,t}A^\top + W \\ &= A[\Sigma_{t,t-1} - \Sigma_{t,t-1}C^\top(C\Sigma_{t,t-1}C^\top + V)^{-1}C\Sigma_{t,t-1}] + W \\ &= A\Sigma_tA^\top + W - A\Sigma_tC^\top(C\Sigma_tC^\top + V)^{-1}C\Sigma_tA^\top \end{aligned}$$

To simplify the formulas, define the *Kalman gain*

$$L_t = -A\Sigma_tC^\top(C\Sigma_tC^\top + V)^{-1}.$$

In terms of the Kalman gain, the recursions for the mean and covariance of the belief simplify to

$$\begin{aligned} \hat{x}_{t+1} &= A\hat{x}_t + Bu_t - L_t(y_t - C\hat{x}_t - Du_t) \\ \Sigma_{t+1} &= (A + L_tC)\Sigma_tA^\top + W. \end{aligned}$$

20

Linear–Quadratic–Gaussian Control

The linear–quadratic–Gaussian (LQG) control problem is to design a controller that minimizes the expected quadratic cost of a linear time-invariant system subject to Gaussian noise.

20.1 Problem statement

The LQG control problem can be formulated in either continuous or discrete time, and the objective can be optimized over a finite or infinite time horizon. Here, we consider the infinite-horizon discrete-time LQG problem.

Consider the discrete-time LTI system

$$\begin{aligned}x_{t+1} &= Ax_t + Bu_t + w_t \\ y_t &= Cx_t + v_t\end{aligned}$$

where w_t and v_t are white Gaussian noise processes with covariance matrices W and V , respectively. Our goal is to determine the control inputs u_t to minimize the expected quadratic cost

$$J = \min_u \lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=0}^{T-1} (x_t^\top Q x_t + u_t^\top R u_t) \right],$$

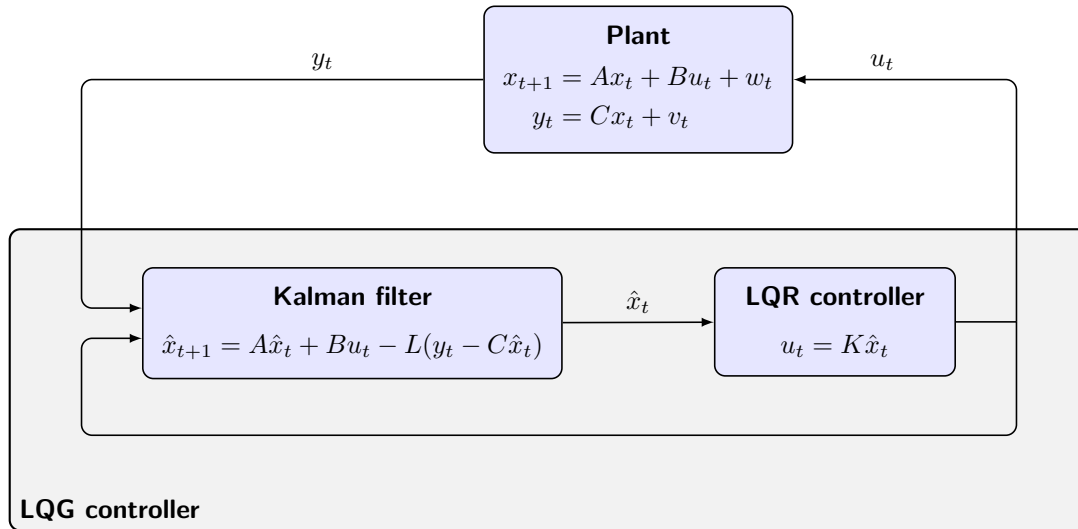
where $Q \succeq 0$ and $R \succ 0$. The expectation in the cost is over the process noise, measurement noise, and initial state conditioned on the observed output.

Separation principle. The optimal control policy for the LQG problem satisfies the *separation principle*, which states that the optimal controller separates into optimal state estimation followed by optimal static state feedback applied to the estimate. That is,

- a) first use a Kalman filter to find the MMSE estimate of the state x_t given all currently available information, and then
- b) apply the optimal LQR policy to the state estimate, treating the estimate as if it were the true state.

20.2 Solution

The control input u_t that minimizes the expected cost J consists of the LQR controller applied to the state estimate produced by the Kalman filter. This controller structure is shown below.



The steady-state LQR gain is

$$K = -(B^T P B + R)^{-1} B^T P A$$

and the steady-state Kalman filter gain is

$$L = -A \Sigma C^T (C \Sigma C^T + V)^{-1}$$

where P and Σ are the solutions to the algebraic Riccati equations

$$P = A^T P A + Q - A^T P B (B^T P B + R)^{-1} B^T P A$$

and

$$\Sigma = A \Sigma A^T + W - A \Sigma C^T (C \Sigma C^T + V)^{-1} C \Sigma A^T.$$

The optimal cost using the LQG controller is

$$J = \underbrace{\text{tr}(P W)}_{\text{LQR cost}} + \underbrace{\text{tr}(\Sigma (A^T P A - P + Q))}_{\text{cost of estimation}}$$

which separates into the cost of the optimal LQR controller and the cost of estimating the state. When there is no noise, our estimate of the state is exact and $\Sigma = 0$ so the second term vanishes.

20.3 Extensions

The LQG controller can also be constructed for continuous-time systems, over a finite horizon, and for time-varying systems. Recall that we often obtain the linear system as the linearization of a nonlinear system. When the linearization is about an equilibrium point, this results in a linear *time-invariant* system. But when the linearization is about an equilibrium trajectory, the result is a linear *time-varying* system. The LQG controller can also be applied to such a scenario.

21

Model Reduction

In some applications, high-fidelity models may be too complex for efficient use in tasks such as real-time control, optimization, or simulation. Given a (possibly high-dimensional and complex) model of a system, model reduction seeks to simplify the model while preserving its essential characteristics. Benefits of using a reduced model include reduced computational burden, enhanced insight via interpretable representations of system behavior, and reduced computational resources for implementation.

Example. Let $\varepsilon > 0$ be a small number, and consider the system

$$\begin{aligned}\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \varepsilon \\ \varepsilon^{-1} \end{bmatrix} u, \\ y &= \begin{bmatrix} \varepsilon^{-1} & \varepsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.\end{aligned}$$

While the system is minimal (both controllable and observable), the first state is barely controllable while the second state is barely observable. We can see this clearly from the controllability and observability Gramians,

$$W_c = \frac{1}{2} \begin{bmatrix} \varepsilon^2 & 1 \\ 1 & \varepsilon^{-2} \end{bmatrix} \quad \text{and} \quad W_o = \frac{1}{2} \begin{bmatrix} \varepsilon^{-2} & 1 \\ 1 & \varepsilon^2 \end{bmatrix}.$$

Using the change of coordinates $z_1 = \varepsilon^{-1}x_1$ and $z_2 = \varepsilon x_2$, however, the system becomes

$$\begin{aligned}\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} &= \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u, \\ y &= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}.\end{aligned}$$

In these coordinates, both states are equally controllable and observable. Precisely, the controllability and observability Gramians in the transformed coordinates are both equal,

$$\tilde{W}_c = \tilde{W}_o = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

21.1 Balanced realization

Definition (Balanced realization). A state-space realization is *balanced* if the controllability and observability Gramians are equal and diagonal. ■

Definition (Hankel singular values). For a balanced realization, the Hankel singular values are the diagonal elements of the controllability and observability Gramians. ■

The Hankel singular values are then the values $\sigma_1, \dots, \sigma_n$ such that

$$W_c = W_o = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

ordered such that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$.

Theorem (Balanced realization).

- Let W_c and W_o denote the controllability and observability Gramians of a minimal realization.
- Compute Cholesky factorizations

$$W_c = W^T W \quad \text{and} \quad W_o = Z^T Z.$$

- Compute a singular value decomposition

$$WZ^T = U\Sigma V^T$$

with Σ diagonal and U and V orthogonal.

- Perform a state transformation with

$$T = W^T U \Sigma^{-1/2} \quad \text{and} \quad T^{-1} = \Sigma^{-1/2} V^T Z.$$

Then, the resulting realization is balanced with controllability and observability Gramians equal to Σ . ■

Proof. Recall that the observability Gramian is

$$W_o = \int_0^\infty e^{A^T t} C^T C e^{At} dt.$$

Using the state transformation $x = Tz$, the Gramian of the transformed system is

$$W_o = \int_0^\infty e^{(T^{-1}AT)^T t} (CT)^T (CT) e^{(T^{-1}AT)t} dt = T^T W_o T.$$

So for the transformation with $T = W^T U \Sigma^{-1/2}$, the observability Gramian becomes

$$\begin{aligned} \tilde{W}_o &= T^T W_o T \\ &= (\Sigma^{-1/2} V^T Z)^{-T} (Z^T Z) (\Sigma^{-1/2} V^T Z)^{-1} \\ &= \Sigma^{1/2} V^T (Z^{-T} Z) (Z Z^{-1}) V \Sigma^{1/2} \\ &= \Sigma^{1/2} V^T V \Sigma^{1/2} \\ &= \Sigma \end{aligned}$$

where we used that V is orthogonal (so $V^T V = I$). The controllability Gramian transforms similarly. ■

We can use the construction of the balanced realization to construct other realizations as well.

- **Input normal realization.** Using the state transformation matrix $T\Sigma^{1/2}$ produces the input normal realization in which all states are equally controllable with $W_c = I$ and $W_o = \Sigma^2$.
- **Output normal realization.** Using the state transformation matrix $T\Sigma^{-1/2}$ produces the output normal realization in which all states are equally observable with $W_c = \Sigma^2$ and $W_o = I$.

21.2 Balanced truncation

One method for model reduction is balanced truncation, where we put the state-space realization in balanced form and then discard states that are not very controllable/observable (those will small Hankel singular values).

Partition the system as

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] = \left[\begin{array}{cc|c} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & C_2 & D \end{array} \right]$$

where A is $k \times k$. Then, the reduced model of order k is the subsystem

$$\left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & D \end{array} \right].$$

Example. If the Gramians are $W_c = W_o = \text{diag}(3, 2, 1, 10^{-6}, 10^{-6}, 10^{-8})$, then the system can be approximated by a third-order system.

System Identification

All of the techniques that we have learned require a *model* of the system (such as the state space matrices or transfer function). If we do not have a model, we can learn one from input-output data using *system identification*.

22.1 Problem description

Given the input-output data (u_k, y_k) for $k = 0, 1, \dots, T - 1$, our goal is to find the state space matrices (A, B, C, D) along with the state x_k such that

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k\end{aligned}$$

where w_k is the process noise and v_k is the measurement noise. Ideally, we would like our model to fit the data exactly (meaning that the noise is zero). When this is not possible, we instead seek to fit the data using “small” amount of noise.

Comments

- Recall that the state space representation of a system is not unique. So which representation should we use in our model (CCF, OCF, diagonal, ...)?
- How many states should our model have? More states allows us to better match the data (use “smaller” noise), but may over-fit to noise in the data and results in a more complex model.

A common solution to these issues is to use the balanced realization in which each state is equally controllable and observable, and then perform balanced truncation to remove states that are not very controllable/observable.

There are many algorithms for system identification, such as dynamic mode decomposition, autoregressive moving average, least squares, etc. Some relevant Matlab commands include `n4sid`, `ssest`, and `ssregest`.

22.2 Eigensystem realization algorithm

The eigensystem realization algorithm (ERA) produces a state-space realization from the impulse response data of a system.

- This is also known as the Ho–Kalman algorithm.
- ERA balances the empirical Gramians (i.e., those computed from the data).
- For general input-output data (not just the impulse response), we can combine the ERA algorithm with the observer Kalman filter identification (OKID) algorithm.

Recall that the impulse response of a discrete-time system is

$$h(k) = \sum_{\ell=0}^{k-1} CA^{k-\ell-1}B\delta(\ell) + D\delta(k) = \begin{cases} CA^{k-1}B & \text{if } k \geq 1, \\ D & \text{if } k = 0. \end{cases}$$

The coefficients D , CB , CAB , CA^2B , \dots of the impulse response are called the *Markov parameters* of the system.

Given the Markov parameters, the goal is to reconstruct the realization (A, B, C, D) . For example, we trivially have that $D = h(0)$.

Given the impulse response, we can form the Hankel matrix:

$$H = \begin{bmatrix} h(1) & h(2) & \dots & H(T) \\ h(2) & h(3) & \dots & h(T+1) \\ \vdots & \vdots & \ddots & \vdots \\ h(T) & h(T+1) & \dots & h(2T-1) \end{bmatrix}.$$

The Hankel matrix has the following expression in terms of the state-space matrices, which factors in terms of the controllability and observability matrices:

$$H = \begin{bmatrix} CB & CAB & \dots & CA^{T-1}B \\ CAB & CA^2B & \dots & CA^TB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{T-1}B & CA^TB & \dots & CA^{2T-2}B \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{T-1} \end{bmatrix} [B \ AB \ \dots \ A^{T-1}B] = QP.$$

While we do not have access to (A, B, C, D) , we can form the Hankel matrix from the impulse response data, which suggests how we might recover the state-space matrices. In particular, factor the Hankel matrix using the singular value decomposition:

$$H = U\Sigma V^T$$

where U and V are orthogonal matrices (meaning that $U^T U = I$ and $V^T V = I$), and Σ is a diagonal matrix with nonnegative real numbers on the diagonal (called singular values). Then, the controllability and observability matrices are

$$Q = U\Sigma^{1/2} \quad \text{and} \quad P = \Sigma^{1/2}V^T.$$

Note that the input matrix B is the first column of the controllability matrix P and the output matrix C is the first row of the observability matrix Q ,

$$B = P \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{and} \quad C = [1 \ 0 \ \dots \ 0] Q.$$

Therefore, all that is left is to find the state transition matrix A . To do so, construct the shifted Hankel

matrix

$$\begin{aligned}
 \tilde{H} &= \begin{bmatrix} h(2) & h(3) & \dots & h(T+1) \\ h(3) & h(4) & \dots & h(T+2) \\ \vdots & \vdots & \ddots & \vdots \\ h(T+1) & h(T+2) & \dots & h(2T) \end{bmatrix} \\
 &= \begin{bmatrix} CAB & CA^2B & \dots & CA^TB \\ CA^2B & CA^3B & \dots & CA^{T+1}B \\ \vdots & \vdots & \ddots & \vdots \\ CA^TB & CA^{T+1}B & \dots & CA^{2T-1}B \end{bmatrix} \\
 &= \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{T-1} \end{bmatrix} A [B \quad AB \quad \dots \quad A^{T-1}B] \\
 &= QAP.
 \end{aligned}$$

We already know A and P from the SVD of the original Hankel matrix, so just solve for A :

$$A = Q^+ \tilde{H} P^+$$

where $(\cdot)^+$ denotes the pseudoinverse.

Comments

- If $h(k)$ is the exact impulse response, then (A, B, C, D) is a minimal realization.
- If the dimension n of the state-space realization is not known in advance, then use the Hankel singular values (the singular values of H) to estimate n . The dimension of the realization can then be set by truncating the SVD,

$$H = [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \approx U_1 \Sigma_1 V_1^T$$

(remove small singular values in Σ_2).

- The controllability and observability Gramians are

$$W_c = \sum_{k=0}^{\infty} A^k B B^T (A^T)^k \quad \text{and} \quad W_o = \sum_{k=0}^{\infty} (A^T)^k C^T C A^k.$$

Truncating the summation yields the approximations

$$W_c \approx \sum_{k=0}^{2T-2} A^k B B^T (A^T)^k = P P^T = (\Sigma^{1/2} V^T) (V \Sigma^{1/2}) = \Sigma,$$

so ERA balances the empirical Gramians, and the realization becomes balanced in the limit as $T \rightarrow \infty$.

23

Model Predictive Control

We previously studied the LQR problem, which constructs the state feedback controller that minimizes a quadratic cost. Realistic control problems, however, also have constraints on the allowable states and/or inputs. One heuristic for solving such problems is model predictive control (MPC), also known as receding horizon control (RHC). The key idea is to solve a constrained optimization problem to find the optimal control subject to the constraints over a *finite* horizon, apply the *first* control input to the system, and then repeat the entire process at the next iteration.

23.1 Constrained optimal control

Given sets X and U , suppose we want our state to satisfy the constraint $x_t \in X$ and $u_t \in U$ for all times t . The corresponding infinite-horizon constrained LQR problem is to find the states x_1, x_2, \dots and control inputs u_0, u_1, \dots to solve the following optimization problem:

$$\begin{aligned} J_\infty(x_0) &= \text{minimize} && \sum_{t=0}^{\infty} (x_t^\top Q x_t + u_t^\top R u_t) \\ &\text{subject to} && x_{t+1} = A x_t + B u_t \quad \text{for } t = 0, 1, \dots \\ &&& x_t \in X, \quad u_t \in U \quad \text{for } t = 0, 1, \dots \end{aligned}$$

As in LQR, we assume that the cost matrices satisfy $Q \succeq 0$ and $R \succ 0$. Unlike standard LQR, however, the constrained optimization problem is not typically tractable.

23.2 Overview of MPC

Model predictive control is a heuristic solution that finds the optimal solution over a fixed horizon N , applies the first control input from the optimal solution, and then repeats. For a planning horizon N , terminal cost Q_f and terminal set X_f , the finite-horizon optimization problem is

$$\begin{aligned} J_N(z) &= \text{minimize} && \sum_{t=0}^{N-1} (x_t^\top Q x_t + u_t^\top R u_t) + x_N^\top Q_f x_N \\ &\text{subject to} && x_{t+1} = A x_t + B u_t, \quad t = 0, 1, \dots, N-1 \\ &&& x_t \in X, \quad u_t \in U, \quad t = 0, 1, \dots, N-1 \\ &&& x_0 = z, \quad x_N \in X_f \end{aligned}$$

This finite-horizon planning problem serves as an approximation to the infinite-horizon problem, but is computationally tractable since it involves a finite number of variables and constraints. Let $u_0^*(z)$ be the optimal value of u_0 in the above problem for $J_N(z)$.

The steps of MPC are then as follows:

- a) At time t , observe the current state x_t and solve $J_N(x_t)$.
- b) Choose the control action $u_0^*(x_t)$.
- c) Move to time step $t + 1$ and repeat.

Part V

Appendices

A

Linear Algebra

Linear algebra is the branch of mathematics concerned with linear equations and linear maps, along with their representations in vector spaces and through matrices.

$$\begin{aligned} a_{11} x_1 + a_{12} x_2 = b_1 \\ a_{21} x_1 + a_{22} x_2 = b_2 \end{aligned} \iff \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \iff Ax = b$$

The main mathematical objects in linear algebra are scalars, vectors, and matrices.

A.1 Vector space

A vector space is a set of objects, called *vectors*, over another set of objects, called *scalars*, that satisfy certain properties. For instance, vectors can be added together, and a vector can be scaled (or multiplied) by a scalar to produce another vector.

Scalars

A scalar is a number, such as 2, $\sqrt{5}$, or π . Scalars may be real numbers or complex numbers. We can do standard algebraic operations with scalars, such as addition, subtraction, multiplication, and division.

Vectors

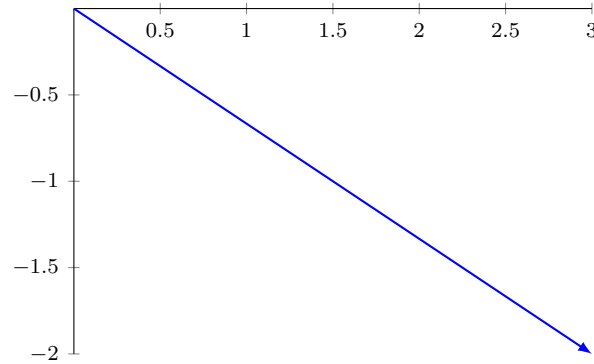
A vector is a one-dimensional array of numbers, such as

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} \sqrt{3} \\ -2 \end{bmatrix}, \quad \text{or} \quad \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

The dimension of a vector is its number of elements. For example,

$$v = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$$

is a two-dimensional vector since it has two elements. In two dimensions, we can visualize a vector as an arrow extending from the origin to its coordinates in the plane.



The inner product of two vectors with the same dimension is the scalar

$$\langle a, b \rangle = \sum_i a_i b_i$$

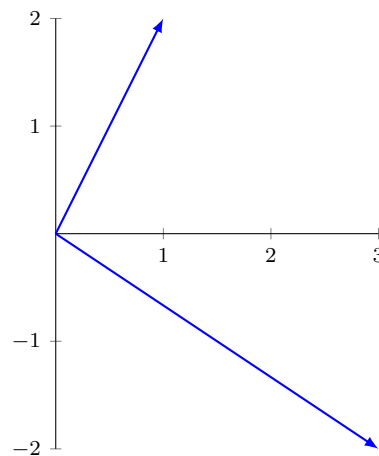
For example,

$$\left\langle \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} \right\rangle = (1)(2) + (2)(0) + (3)(1) = 5$$

Two vectors of the same dimension are *orthogonal* if their inner product is zero, $\langle a, b \rangle = 0$. For example,

$$\left\langle \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 4 \\ -2 \end{bmatrix} \right\rangle = (1)(4) + (2)(-2) = 0$$

Orthogonality is a generalization of two lines being perpendicular. In two dimensions, vectors are orthogonal if they are perpendicular to each other.



The *length* of a vector is the square root of its inner product with itself.

$$|a| = \sqrt{\langle a, a \rangle}$$

For example,

$$\left| \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \right| = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$$

In two dimensions, the length is the standard length of the vector from the origin (this is the Pythagorean theorem).

Matrices

A matrix is a rectangular array of numbers, such as

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} -2 & \sqrt{3} \\ 1 & 5 \end{bmatrix}, \quad \text{or} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Notation. Matrices are typically denoted by uppercase letters, while their individual elements are represented by the same lowercase letter. For example, we might write $A = [a_{ij}]$ to denote a matrix A with elements a_{11} , a_{12} , and so on. ■

The *dimensions* of a matrix are the number of rows and columns. For instance, the matrix

$$\begin{bmatrix} 1 & 0 & 2 \\ 4 & 3 & 4 \end{bmatrix}$$

has dimensions 2×3 , meaning that it has two rows and three columns.

The transpose of a matrix is denoted A^T , and is a matrix with the rows and columns switched. For example,

$$\begin{bmatrix} 1 & 0 & 2 \\ 4 & 3 & 4 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 \\ 0 & 3 \\ 2 & 4 \end{bmatrix}$$

The matrix has dimensions 2×3 while its transpose has dimensions 3×2 . The first row of the matrix becomes the first column of its transpose, and so on.

The sum of two matrices of the same dimensions is

$$A + B = [a_{ij} + b_{ij}]$$

Each element of the sum is the sum of the corresponding elements of the two matrices. For example,

$$\begin{bmatrix} 1 & 0 & 2 \\ 4 & 3 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 2 \\ 4 & 4 & 4 \end{bmatrix}$$

A matrix can be multiplied by a scalar α , which multiplies every element of the matrix by the scalar.

$$\alpha A = [\alpha a_{ij}]$$

For example,

$$3 \begin{bmatrix} 1 & 0 & 2 \\ 4 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 6 \\ 12 & 9 & 12 \end{bmatrix}$$

Matrix multiplication

Consider a matrix A with dimensions $m \times n$ and a matrix B with dimensions $p \times q$. We can multiply A times B only if $n = p$, in which case the product is the $m \times q$ matrix

$$C = AB = [c_{ij}] \quad \text{where} \quad c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

For example,

$$\begin{bmatrix} 1 & 0 & 2 \\ 4 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 7 & 4 \end{bmatrix}$$

Unlike scalar multiplication, matrix multiplication in general is not commutative! So $AB \neq BA$. For example,

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 3 & 4 \end{bmatrix} \neq \begin{bmatrix} 1 & 3 \\ 3 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

The product may not even have the same dimensions,

$$\begin{bmatrix} 1 & 0 & 2 \\ 4 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 7 & 4 \end{bmatrix} \neq \begin{bmatrix} 5 & 3 & 7 \\ 1 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 4 & 3 & 4 \end{bmatrix}$$

The product of two matrices may exist when multiplied in one order but not in the other.

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 1 & 2 \\ 0 & 0 \end{bmatrix} \neq \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \text{ does not exist!}$$

Determinant

The determinant of a square matrix is a scalar, denoted $\det(A)$. While the precise formula for the determinant is complicated, for a two-dimensional matrix it is simply the product of the diagonal terms minus the product of the off-diagonal terms:

$$\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11} a_{22} - a_{12} a_{21}$$

Special matrices

The identity matrix, typically denoted I , is a square matrix with ones on the diagonal and zeros on the off-diagonal. For example, the identity matrix in three dimensions is

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse

The inverse of a square matrix A is a square matrix A^{-1} of the same dimensions such that its product with the original matrix (in any order) is the identity matrix.

$$AA^{-1} = A^{-1}A = I$$

A matrix has an inverse only if its determinant is not zero. For a two-dimensional matrix with nonzero determinant, its inverse is the two-dimensional matrix

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{a_{11} a_{22} - a_{12} a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

A.2 Eigenvalues and eigenvectors

Definition. If multiplying a square matrix A by a nonzero vector x scales the vector by a (possibly complex) scalar λ , then λ is an *eigenvalue* of A with *eigenvector* x . That is, an eigenvalue and eigenvector satisfy the relationship

$$Ax = \lambda x.$$

A pair (λ, x) of an eigenvalue and its corresponding eigenvector is an *eigenpair*.

Remark (Dimensions). If A has dimensions $n \times n$, then an eigenvector x must have dimension n , and A must be square so that the vector Ax also has dimension n . ■

Example. Consider the 2×2 matrix

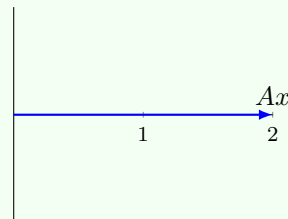
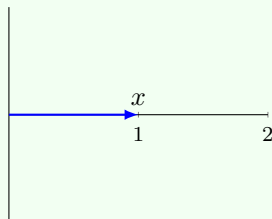
$$A = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$$

One eigenvector and eigenvalue pair is

$$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \lambda = 2$$

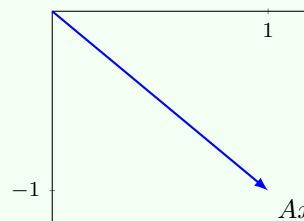
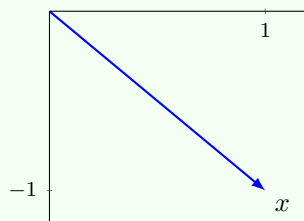
We can directly verify that this satisfies the eigen equation

$$Ax = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \lambda x$$



Another eigenvector and eigenvalue pair is

$$x = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \lambda = 1$$



Note that Ax points in the same direction as x and is scaled by λ .

Example (Differential). Signals are infinite-dimensional vectors, in which case the matrix A can be a general linear operator. For instance, the differential operator $\frac{d}{dt}$ maps a continuous-time signal to another continuous-time signal. Moreover, one of the basic results in calculus is that this operation is linear:

$$\frac{d}{dt}(ax(t) + by(t)) = a\frac{d}{dt}x(t) + b\frac{d}{dt}y(t).$$

The eigenvectors (also called *eigenfunctions*) of the differential operator are the exponential signals $e^{\lambda t}$ with corresponding eigenvalue λ since they satisfy

$$\frac{d}{dt}e^{\lambda t} = \lambda e^{\lambda t}.$$

Note that this is true for *any* λ , so there are an infinite number of eigenpairs!

Computing eigenvalues

If (λ, x) is an eigenpair of A , then they satisfy $(\lambda I - A)x = 0$, so the columns of $\lambda I - A$ are linearly dependent, which implies that its determinant is zero. Therefore, the eigenvalues of A are the solutions to the polynomial equation

$$\det(\lambda I - A) = 0.$$

This is called the *characteristic equation* of A , and the polynomial $\det(\lambda I - A)$ is its *characteristic polynomial*.

Remark (Number of eigenvalues). If A is an $n \times n$ matrix, then its characteristic polynomial is a polynomial in λ of degree n , so it always has exactly n roots (that may be complex and/or repeated). ■

Example.

$$A = \begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix}$$

The characteristic equation is

$$\begin{aligned} 0 &= \det(\lambda I - A) = \det\left(\lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix}\right) \\ &= \det \begin{bmatrix} \lambda - 3 & -4 \\ -2 & \lambda - 1 \end{bmatrix} \\ &= (\lambda - 3)(\lambda - 1) - (-4)(-2) \\ &= \lambda^2 - 4\lambda - 5 \\ &= (\lambda + 1)(\lambda - 5) \end{aligned}$$

Therefore, the eigenvalues are -1 and 5 .

Example (complex eigenvalues).

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

The characteristic equation is

$$0 = \det(\lambda I - A) = \det \begin{bmatrix} \lambda & -1 \\ 1 & \lambda \end{bmatrix} = \lambda^2 + 1$$

Therefore, the eigenvalues are $\pm j$. Since the eigenvalues are the roots of a polynomial, they may be complex.

Computing eigenvectors

Given an $n \times n$ matrix A and a (possibly complex) scalar eigenvalue λ , we can find the n -dimensional eigenvector x associated with the eigenvalue by solving the eigen equation

$$(\lambda I - A)x = 0$$

Example.

$$A = \begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix} \implies \lambda_1 = 5, \quad \lambda_2 = -1$$

For the eigenvalue λ_1 , we can find the associated eigenvector $x_1 = \begin{bmatrix} a \\ b \end{bmatrix}$ by solving the equation

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 5-3 & -4 \\ -2 & 5-1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 2 & -4 \\ -2 & 4 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

which implies that

$$\begin{aligned} 0 &= 2a - 4b \\ 0 &= -2a + 4b \end{aligned}$$

The second equation is simply the negative of the first equation, so it provides no additional information. The first equation implies that $a = 2b$, so the eigenvector is

$$x_1 = \begin{bmatrix} 2b \\ b \end{bmatrix} \quad \text{for any } b$$

For the second eigenvalue λ_2 , the corresponding eigenvector $x_2 = \begin{bmatrix} a \\ b \end{bmatrix}$ is the solution to

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1-3 & -4 \\ -2 & -1-1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -4 & -4 \\ -2 & -2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

The second equation is simply half the first equation, so again it provides no additional information. The first equation implies that $a = -b$, so the second eigenvector is

$$x_2 = \begin{bmatrix} -b \\ b \end{bmatrix} \quad \text{for any } b$$

The eigenvectors are not unique since they can be scaled. We usually just pick a convenient scaling. For instance,

$$\lambda_1 = 5, \quad x_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \text{and} \quad \lambda_2 = -1, \quad x_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

A.3 Matrix similarity

The mapping $A \mapsto T^{-1}AT$ is called a *similarity transformation*, and two matrices A and B are *similar* if there exists an invertible matrix T such that $A = T^{-1}AT$, which is denoted $A \sim B$.

The following result provides several useful properties of similar matrices.

Proposition (Similar matrices). If $A \sim B$, then

- a) $A^k \sim B^k$ for $k = 0, 1, 2, \dots$
- b) $e^{At} \sim e^{Bt}$
- c) $(sI - A)^{-1} \sim (sI - B)^{-1}$
- d) $\det(sI - A) = \det(sI - B)$
- e) A and B have the same eigenvalues

■

Proof. If A and B are similar, then there exists an invertible matrix T such that $B = TAT^{-1}$.

a) $B^k = (TAT^{-1})(TAT^{-1}) \cdots (TAT^{-1}) = TA^kT^{-1} \sim A^k$.

b) $e^{Bt} = \sum_{k=0}^{\infty} \frac{1}{k!} B^k t^k = \sum_{k=0}^{\infty} \frac{1}{k!} TA^kT^{-1} t^k = Te^{At}T^{-1} \sim e^{At}$

c) $(sI - B)^{-1} = (sI - TAT^{-1})^{-1} = (T(sI - A)T^{-1})^{-1} = T(sI - A)^{-1}T^{-1} \sim (sI - A)^{-1}$

d) $\det(sI - B) = \det(T(sI - A)T^{-1}) = \det(T) \det(sI - A) \det(T^{-1}) = \det(sI - A) \det(TT^{-1}) = \det(sI - A)$

e) This follows from the fact that A and B have the same characteristic polynomials (item (d)). ■

A.4 Subspace

Definition (Subspace). A *subspace* S of a vector space V is a subset of V that is itself a vector space. Equivalently, a subspace S is a subset of V that:

- contains zero: $0 \in S$
- is closed under addition: if $x, y \in S$, then $x + y \in S$
- is closed under scalar multiplication: if $x \in S$ and a is a scalar, then $ax \in S$

There are four fundamental subspaces associated with a matrix A .

Definition (Fundamental subspaces). The four fundamental subspaces associated with an $m \times n$ real matrix A are the following:

$$\text{col}(A) = \{Ax \mid x \in \mathbb{R}^n\} \quad (\text{column space})$$

$$\text{row}(A) = \{A^T y \mid y \in \mathbb{R}^m\} \quad (\text{row space})$$

$$\text{null}(A) = \{x \in \mathbb{R}^n \mid Ax = 0\} \quad ((\text{right}) \text{ null space})$$

$$\text{null}(A^T) = \{y \in \mathbb{R}^m \mid A^T y = 0\} \quad (\text{left null space})$$

The row space and null space are subspaces of \mathbb{R}^n , while the column space and left null space are subspaces of \mathbb{R}^m . The interpretations of these subspaces are as follows. The column space is the set of all linear combinations of the columns of the matrix, while the row space is the set of all linear combinations of its rows. The right null space is the set of all vectors that are mapped to zero when multiplied by A on the right, while the left null space is the set of all vectors that are mapped to zero when multiplied by A on the left.

Another important subspace is the span of a set of vectors.

Definition (Span). The *span* of a set of vectors is the subspace of all linear combinations of the vectors:

$$\text{span}(x_1, x_2, \dots, x_n) = \{a_1x_1 + \dots + a_nx_n \mid a_1, \dots, a_n \text{ scalars}\}.$$

Example. Consider the matrix

$$A = \begin{bmatrix} 1 & 0 & 2 & 3 \\ 0 & 1 & 4 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The column space and left null space are

$$\text{col}(A) = \text{span} \left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right) \quad \text{and} \quad \text{null}(A^T) = \text{span} \left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right),$$

and the row space and right null space are

$$\text{row}(A) = \text{span} \left(\begin{bmatrix} 1 \\ 0 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 4 \\ 5 \end{bmatrix} \right) \quad \text{and} \quad \text{null}(A) = \text{span} \left(\begin{bmatrix} -2 \\ -4 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -3 \\ -5 \\ 0 \\ 1 \end{bmatrix} \right).$$

Definition (Subspace dimension). The *dimension* of a subspace S , denoted $\dim(S)$, is the number of vectors in any basis for that subspace.

If the vectors are linearly independent, then the dimension of their span is the number of vectors.

Definition (Orthogonal complement). The orthogonal complement of a subspace S is the set of vectors that are orthogonal to all vectors in the subspace:

$$S^\perp = \{v \mid u^T v = 0 \text{ for all } u \in S\}.$$

Fact. The orthogonal complement is an *involution*, meaning that the orthogonal complement of the orthogonal complement is the original subspace:

$$(S^\perp)^\perp = S.$$

Definition (Sum and direct sum of subspaces). The sum of two subspaces U and V is the subspace

$$U + V = \{u + v \mid u \in U \text{ and } v \in V\}.$$

Moreover, this is called the *direct sum*, denoted $U \oplus V$, when the decomposition is unique.

Fact. The orthogonal complement of the four fundamental subspaces are as follows:

$$\begin{aligned} \text{row}(A)^\perp &= \text{null}(A), \\ \text{col}(A)^\perp &= \text{null}(A^T), \\ \text{null}(A)^\perp &= \text{row}(A), \\ \text{null}(A^T)^\perp &= \text{col}(A). \end{aligned}$$

Fact. If V is an inner product space and U is a subspace of V , then V is the direct sum of U and its orthogonal complement,

$$U \oplus U^\perp = V.$$

In particular, for the four fundamental spaces associated with a matrix $A \in \mathbb{R}^{m \times n}$,

$$\text{row}(A) \oplus \text{null}(A) = \mathbb{R}^n \quad \text{and} \quad \text{col}(A) \oplus \text{null}(A^\top) = \mathbb{R}^m.$$

A.5 Diagonalization

Suppose (λ_k, x_k) is a set of eigenpairs for $k = 1, \dots, n$, and define the matrices

$$X = [x_1 \quad x_2 \quad \dots \quad x_n] \quad \text{and} \quad \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}.$$

Then, the matrix form of the eigen equation is

$$\begin{aligned} AX &= A [x_1 \quad x_2 \quad \dots \quad x_n] \\ &= [\lambda_1 x_1 \quad \lambda_2 x_2 \quad \dots \quad \lambda_n x_n] \\ &= [x_1 \quad x_2 \quad \dots \quad x_n] \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \\ &= X\Lambda. \end{aligned}$$

If all eigenvectors are linearly independent, then the matrix X is invertible, in which case we can solve the eigen equation for the matrix of eigenvalues as

$$X^{-1}AX = \Lambda.$$

So applying a similarity transformation with the matrix of eigenvectors produces the diagonal matrix of eigenvalues. In this case, we say that the matrix A is *diagonalizable*, and the matrix X diagonalizes A .

Definition (Diagonalizable). A matrix is *diagonalizable* if it is similar to a diagonal matrix.

As we saw above, if the eigenvectors are all linearly independent, then the matrix of eigenvectors is invertible and diagonalizes the matrix. The following result describes when the eigenvectors are all linearly independent.

Fact. If all eigenvalues are distinct, then all eigenvectors are linearly independent.

This implies that a matrix is diagonalizable when all of its eigenvalues are distinct. But in general, some eigenvalues may be repeated. The characteristic polynomial then has the form

$$\det(\lambda I - A) = (\lambda - \lambda_1)^{n_1} (\lambda - \lambda_2)^{n_2} \dots (\lambda - \lambda_d)^{n_d}$$

where $n_1 + \dots + n_d = n$ (the size of A) and d is the number of distinct eigenvalues.

Definition (Eigenspace). The *eigenspace* associated with a complex number λ is the subspace

$$E_\lambda = \text{null}(\lambda I - A).$$

There are several cases for the eigenspace depending on the value of λ .

- If λ is not an eigenvalue, then $\lambda I - A$ is invertible, so the eigenspace is the trivial subspace $E_\lambda = \{0\}$.
- If λ is an eigenvalue, then the eigenspace is the subspace of all the associated eigenvectors.

Definition (Eigenvalue multiplicity). Each complex number λ has two associated multiplicities:

- The *algebraic multiplicity* of λ is the number of times it is repeated as a root of the characteristic polynomial (denoted n_k).
- The *geometric multiplicity* of λ is the dimension of its corresponding eigenspace, $\dim(E_\lambda)$, which is the number of linearly independent eigenvectors with eigenvalue λ .

Fact. The eigenspace satisfies the following:

- $1 \leq \dim(E_{\lambda_k}) \leq n_k$
- If $x_i \in E_{\lambda_i}$ are eigenvectors in distinct eigenspaces, then $\{x_i\}$ are linearly independent.

Fact. A matrix is diagonalizable if and only if the algebraic multiplicity of each eigenvalue is equal to its geometric multiplicity:

$$n_k = \dim(E_{\lambda_k}) \quad \text{for all } k.$$

Intuitively, this means that each eigenspace must have “enough” linearly independent eigenvectors.

Example. The matrix

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

has eigenvalues at 3 and -1 with corresponding eigenspaces

$$E_3 = \text{span}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) \quad \text{and} \quad E_{-1} = \text{span}\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right).$$

Each eigenspace has dimension one (which is the same as the multiplicity of each eigenvalue), so the matrix is diagonalizable. To diagonalize the matrix, perform a similarity transformation with the matrix of eigenvectors:

$$T = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{with inverse} \quad T^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

The diagonal form is then

$$T^{-1}AT = \begin{bmatrix} 3 & 0 \\ 0 & -1 \end{bmatrix},$$

which is a diagonal matrix containing all of the eigenvalues on the diagonal.

Example. The matrix $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ has two eigenvalues at zero, but the corresponding eigenspace is

$$E_0 = \text{span} \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)$$

which has dimension one, so the matrix is *not* diagonalizable.

A.6 Jordan form

While not every matrix is diagonalizable, any square matrix is similar to a matrix in Jordan form, which is a generalization of a diagonal matrix. The Jordan form of a matrix is a decomposition of the matrix into simple blocks, which is often useful to understand properties of the matrix. If the matrix is diagonalizable, then the Jordan form is equivalent to the eigendecomposition of the matrix.

A *Jordan matrix* is a block-diagonal matrix of the form

$$J = \begin{bmatrix} J_{k_1}(\lambda_1) & & & \\ & J_{k_2}(\lambda_2) & & \\ & & \ddots & \\ & & & J_{k_d}(\lambda_d) \end{bmatrix}$$

with $k_1 + k_2 + \dots + k_d = n$, where each diagonal block $J_{k_i}(\lambda_i)$ is a *Jordan block* of the form

$$J_{k_i}(\lambda_i) = \begin{bmatrix} \lambda_i & 1 & & & \\ & \lambda_i & 1 & & \\ & & \lambda_i & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda_i \end{bmatrix},$$

which is a $k_i \times k_i$ matrix where all unspecified entries are zero. Each Jordan block has the form $J_{k_i}(\lambda_i) = \lambda_i I + N$, where N is a matrix of zeros except for ones on the superdiagonal; such a matrix is called *nilpotent*. Intuitively, a Jordan matrix is “almost” diagonal.

The Jordan form of a square matrix A is

$$A = T^{-1}JT$$

where T is an invertible matrix and J is a Jordan matrix.

Example (Jordan form). Some examples of Jordan forms are as follows:

$$\underbrace{\begin{bmatrix} -1 & -9 \\ 1 & 5 \end{bmatrix}}_A = \underbrace{\begin{bmatrix} 3 & 5 \\ -1 & -2 \end{bmatrix}}_{T^{-1}} \underbrace{\begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}}_J \underbrace{\begin{bmatrix} 2 & 5 \\ -1 & -3 \end{bmatrix}}_T$$

$$\underbrace{\begin{bmatrix} 6 & -4 & -1 & -11 & -1 \\ -7 & 3 & 1 & 11 & 1 \\ -5 & 1 & -2 & 6 & 1 \\ 7 & -4 & -1 & -12 & -1 \\ 5 & -2 & 0 & -7 & -2 \end{bmatrix}}_A = \underbrace{\begin{bmatrix} 0 & -1 & -1 & 1 & 1 \\ 3 & 1 & 2 & -1 & -1 \\ -2 & 0 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -2 & 0 & 1 \end{bmatrix}}_{T^{-1}} \underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & -2 \end{bmatrix}}_J \underbrace{\begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ -9 & 4 & 1 & 12 & 2 \\ -2 & 1 & 0 & 3 & 0 \\ -5 & 3 & 1 & 8 & 1 \\ -5 & 2 & 0 & 7 & 1 \end{bmatrix}}_T$$

Fact. The Jordan form of a matrix can be used to determine the multiplicities of its eigenvalues:

- a) The algebraic multiplicity of an eigenvalue is the sum of the sizes of all Jordan blocks corresponding to that eigenvalue.
- b) The geometric multiplicity of an eigenvalue is the number of Jordan blocks corresponding to that eigenvalue.

Therefore, a matrix is diagonalizable if and only if all of its Jordan blocks have dimension one.

A.7 Matrix exponential

Recall that the series expansion for the exponential of a scalar x is

$$e^x = \sum_{n=0}^{\infty} \frac{1}{n!} x^n = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \dots$$

In the same way, we define the *matrix exponential* of a square matrix X in terms of its series expansion,

$$e^X = \sum_{n=0}^{\infty} \frac{1}{n!} X^n = I + X + \frac{1}{2}X^2 + \frac{1}{6}X^3 + \dots$$

Remark. The matrix exponential is defined by its series expansion, which is *not* the same as taking the exponential of each component of the matrix! ■

Example (diagonal matrix). Consider the diagonal matrix

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}.$$

The powers of this matrix are as follows:

$$A^k = \begin{bmatrix} \lambda_1^k & 0 & 0 \\ 0 & \lambda_2^k & 0 \\ 0 & 0 & \lambda_3^k \end{bmatrix}.$$

Therefore, the matrix exponential is the diagonal matrix whose elements are the (standard) exponentials of each diagonal element:

$$e^{At} = \begin{bmatrix} e^{\lambda_1 t} & 0 & 0 \\ 0 & e^{\lambda_2 t} & 0 \\ 0 & 0 & e^{\lambda_3 t} \end{bmatrix}.$$

Example (nilpotent matrix). Consider the matrix that is all zero except with ones on the superdiagonal:

$$N = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The powers of this matrix are as follows:

$$N^2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad N^3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad N^n = 0 \text{ for all } n \geq 4.$$

This type of matrix is called *nilpotent* since it is zero when raised to a high enough power (in this case, four). Since only a finite number of powers are nonzero, the series expansion is finite.

$$e^{Nt} = I + Nt + \frac{1}{2}N^2t^2 + \frac{1}{6}N^3t^3 = \begin{bmatrix} 1 & t & \frac{1}{2}t^2 & \frac{1}{6}t^3 \\ 0 & 1 & t & \frac{1}{2}t^2 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Properties

The following properties hold for any square matrices A and B , and any scalars t , t_1 , and t_2 .

- e^{At} is the unique matrix satisfying $\frac{d}{dt}(e^{At}) = Ae^{At}$
- $e^{A(t_1+t_2)} = e^{At_1}e^{At_2}$
- $(e^{At})^{-1} = e^{-At}$, so the matrix exponential is always invertible
- $Ae^{At} = e^{At}A$, so the matrix exponential commutes with the matrix in the exponent
- $(e^{At})^T = e^{A^T t}$, so transposing the matrix exponential transposes the matrix in the exponent
- $e^{(A+B)t} = e^{At}e^{Bt}$ for all t if and only if A and B commute (that is, $AB = BA$); this always holds if A and B are scalars

Laplace transform of the matrix exponential. Recall that the Laplace transform of a scalar exponential e^{at} is $\frac{1}{s-a}$. The Laplace transform of a matrix exponential generalizes as follows:

$$\mathcal{L}(e^{At}) = (sI - A)^{-1}$$

Proof.

$$\begin{aligned}
 \mathcal{L}(e^{At}) &= \mathcal{L}\left(\sum_{n=0}^{\infty} \frac{1}{n!} A^n t^n\right) && \text{(definition of matrix exponential)} \\
 &= \sum_{n=0}^{\infty} \frac{1}{n!} A^n \mathcal{L}(t^n) && \text{(linearity of Laplace transform)} \\
 &= \sum_{n=0}^{\infty} \frac{1}{n!} A^n \frac{n!}{s^{n+1}} \\
 &= s^{-1} \sum_{n=0}^{\infty} A^n s^{-n} \\
 &= s^{-1} \sum_{n=0}^{\infty} (s^{-1}A)^n \\
 &= s^{-1} (I - s^{-1}A)^{-1} && \text{(geometric series)} \\
 &= (sI - A)^{-1}
 \end{aligned}$$

Note that the Laplace transform is only defined in its region of convergence, which are the values of s for which the geometric series converges. ■

Computation

To compute the matrix exponential, first transform the matrix to Jordan form

$$A = T^{-1}JT$$

where T is invertible and $J = \text{diag}(J_1, J_2, \dots, J_m)$ is a Jordan matrix in which each J_i a Jordan block:

$$J_i = \begin{bmatrix} \lambda_i & 1 & 0 & \dots & 0 & 0 \\ 0 & \lambda_i & 1 & \dots & 0 & 0 \\ 0 & 0 & \lambda_i & \ddots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_i & 1 \\ 0 & 0 & 0 & \dots & 0 & \lambda_i \end{bmatrix}$$

Each Jordan block has the form $J_i = \lambda_i I + N$, where N is nilpotent. The powers of the matrix are then

$$A^n = (T^{-1}JT)^n = \underbrace{(T^{-1}JT)(T^{-1}JT)\dots(T^{-1}JT)}_{n \text{ times}} = T^{-1}J^nT$$

So we can raise the matrix A to a power simply by raising its Jordan matrix to the power. The matrix exponential is then

$$e^{At} = \sum_{n=0}^{\infty} \frac{t^n}{n!} A^n = \sum_{n=0}^{\infty} \frac{t^n}{n!} T^{-1}J^nT = T^{-1} \left(\sum_{n=0}^{\infty} \frac{t^n}{n!} J^n \right) T = T^{-1}e^{Jt}T$$

The matrix exponential of a block matrix is the block matrix exponentials:

$$\exp\left(\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}\right) = \begin{bmatrix} e^{A_1} & 0 \\ 0 & e^{A_2} \end{bmatrix}$$

And the matrix exponential of a single Jordan block is simple:

$$e^{Jt} = \begin{bmatrix} e^{\lambda t} & t e^{\lambda t} & \frac{1}{2} t^2 e^{\lambda t} & \dots & \frac{1}{(k-1)!} t^{k-1} e^{\lambda t} \\ 0 & e^{\lambda t} & t e^{\lambda t} & \dots & \frac{1}{(k-2)!} t^{k-2} e^{\lambda t} \\ 0 & 0 & e^{\lambda t} & \dots & \frac{1}{(k-3)!} t^{k-3} e^{\lambda t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & e^{\lambda t} \end{bmatrix}$$

A.8 Quadratic forms

A quadratic form is a homogeneous quadratic function. Given a symmetric matrix A , the corresponding quadratic form is

$$f(x) = x^T A x.$$

Minimizing quadratic forms

Ellipses

B

Series expansion

B.1 One-dimensional functions

The Taylor series expansion of a scalar function $f : \mathbb{R} \rightarrow \mathbb{R}$ about a point $\tilde{x} \in \mathbb{R}$ is the infinite summation

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(\tilde{x})}{n!} (x - \tilde{x})^n$$

Explicitly, the first few terms are

$$f(x) = f(\tilde{x}) + \frac{f'(\tilde{x})}{1!} (x - \tilde{x}) + \frac{f''(\tilde{x})}{2!} (x - \tilde{x})^2 + \frac{f'''(\tilde{x})}{3!} (x - \tilde{x})^3 + \dots$$

B.2 Multi-dimensional functions

For multi-dimensional functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$, the general formula for the Taylor series expansion is quite complicated. However, we often only use the first two terms to obtain a linear approximation of a nonlinear function. The first-order Taylor series expansion about a point $\tilde{x} \in \mathbb{R}^n$ is

$$f(x) \approx f(\tilde{x}) + \frac{\partial f(\tilde{x})}{\partial x} (x - \tilde{x})$$

where the *Jacobian matrix* of the function f is the matrix-valued function of partial derivatives

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$