Course Companion

University Electrical Engineering

Every topic handout in one volume

exercises.lol

Contents

1. Circuit Analysis I (DC) 3

2. Mathematical Methods for EE 16

3. Digital Logic Design 28

4. Circuit Analysis II (AC) 38

5. Electronics I 48

6. Signals and Systems 58

7. Electromagnetics 69

8. Microprocessors and Embedded Systems 80

9. Electrical Machines and Power Systems 91

10. Control Systems 105

11. Topic 11 124

12. Digital Signal Processing 134

Circuit Analysis I (DC)

University Electrical Engineering

What this course is, and how to use it

This is the first course in University Electrical Engineering: **direct-current circuit analysis**. A DC circuit is one where currents and voltages settle to steady values – batteries, sensors, logic supplies, and the bias networks inside every amplifier.

The course is a ladder, and every rung is used by the next:

- Units 1.1 and 1.2 need only algebra: quantities, sign conventions, Ohm's law and Kirchhoff's laws.
- Units 1.3 to 1.5 build systematic methods – node and mesh analysis, network theorems, and the ideal op-amp – that turn long circuits into short calculations.
- Units 1.6 and 1.7 add the energy-storing elements and the first **transient response**, which is where calculus begins to matter.

How to read each unit: a short story says why it matters; a picture or a rule of thumb comes before any formula; then the precise statement; then worked examples that get harder, one step at a time, each with a check at the end; then the mistakes people actually make; then a three-line summary. Every number in the examples is something you can verify a second way – a different loop, an unused node, a limiting case – and you should build that habit here: in later courses the check is the difference between a caught error and a burnt resistor.

Quantities, units and measurement

A phone charger labelled "20 W" and a kettle labelled "2000 W" both move charge. Why is one a hundred times more powerful? To answer, you need the six quantities every circuit question is asked in – and one convention that keeps their signs honest.

Learn them as a chain, not a list: charge moves; its rate is current; current through a difference in potential transfers energy; the rate of that transfer is power.

- **Charge** q is measured in **coulombs** (C). One coulomb is about 6.24×10^{18} elementary charges. Charge is **conserved**: it is never created or consumed in a circuit, only moved.
- **Current** i is the rate of charge flow, in **amperes** (A): $i = dq/dt$, so 1 A = 1 C/s.
- **Voltage** v is energy per unit charge, in **volts** (V): 1 V = 1 J/C. It is always measured **between** two points; "5 V at a node" means 5 V between that node and the reference (ground) node.

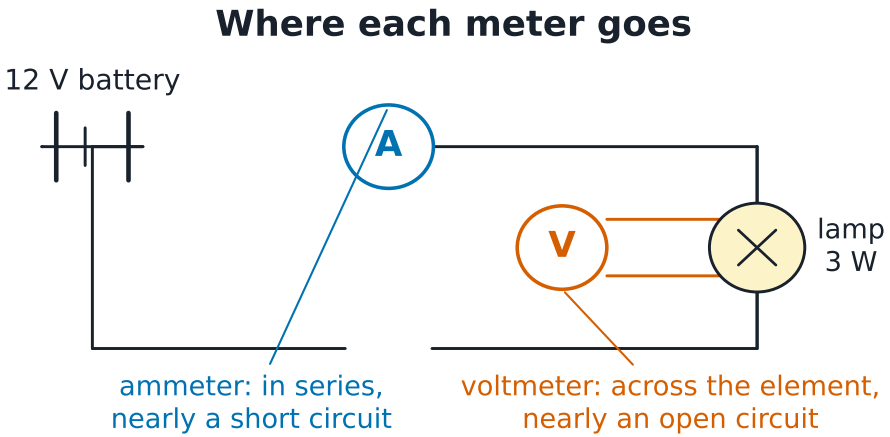
exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

- **Power** p is energy per unit time, in **watts** (W): $p = vi$, and 1 W = 1 J/s.
- **Energy** w is power integrated over time, $w = \int p\,dt$, in **joules** (J). Electricity meters sell the **kilowatt-hour**: 1 kWh = 3.6 MJ.

Deriving the power formula. Move a charge dq through a potential difference v and the energy transferred is $dw = v \cdot dq$. Divide by dt: $p = dw/dt = v \cdot (dq/dt) = vi$. Power is voltage times current – nothing deeper.

The **passive sign convention** removes all ambiguity. Pick a direction for the current through an element and a polarity for its voltage; when the current arrow enters the + terminal, $p = vi$ is the power the element **absorbs**. A positive result means the element consumes power; a negative result means it supplies power. Getting this backwards is the most common source of wrong signs in the whole course.



Worked example 1 (power and energy). A 12 V battery drives a lamp that draws 0.25 A, with the current arrow drawn into the lamp's + terminal. For the lamp, $p = vi = 12 \times 0.25 = 3$ W absorbed. Over 2 minutes the lamp takes $w = pt = 3 \times 120 = 360$ J. With the same arrow drawn for the battery, the battery's current enters its − terminal, so its absorbed power is −3 W: it supplies exactly what the lamp takes, and the bookkeeping closes.

exercises.lol exercises.com.cn

Worked example 2 (the meter reads the wrong number). A 12 V ideal source feeds two 10 MΩ resistors in series, so each should have 6 V across it. A real voltmeter with 10 MΩ input resistance is connected across the lower one to "check". The meter is now a second 10 MΩ resistor in parallel: the lower part combines to 5 MΩ, and the meter reads $\frac{5}{10+5} \times 12 = 4$ V instead of 6 V. The check itself changed the circuit. (Across the ideal source itself the meter would read 12 V correctly: an ideal source holds its voltage whatever is connected.) Rule of thumb: the meter's resistance must be at least a hundred times the resistance it measures. Escalate further: an **ammeter** in current mode is the mirror case – a series meter with non-negligible resistance steals voltage and reads low current.

Common mistakes.

- Reporting "the voltage at point A" without saying what the other point is. Voltage is always between two points; name them.
- Mixing units: minutes with watts, mA with kΩ. Convert first; then the answer's unit falls out by itself (V/Ω = A).
- Trusting a meter without asking what resistance it adds.

Summary: six quantities, one chain (charge to current to energy to power); power is vi under the passive sign convention; meters must be nearly invisible to the circuit (series ammeter = near short, parallel voltmeter = near open).

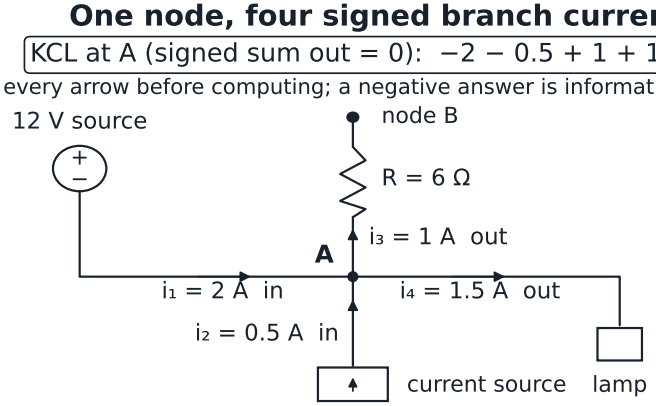
Ohm's law and Kirchhoff's laws

Two resistors arrive at a node and one leaves. How much current takes each exit? The answer needs the two laws named after Kirchhoff – and they are not new physics. One is "charge does not pile up"; the other is "energy books balance". Both were hiding in unit 1.1.

Ohm's law links a **resistor**: $v = Ri$, with R in **ohms** (Ω). **Conductance** $G = 1/R$ (siemens) is sometimes handier. Power in a resistor is always absorbed, and combining $v = Ri$ with $p = vi$ gives the three equivalent forms $p = vi = i^2R = v^2/R$ – use whichever variable you already know.

For **series** elements the resistances add and one current is shared; for **parallel** elements the conductances add and one voltage is shared. Two resistors in parallel combine as the product over the sum, $R_1 \parallel R_2 = \frac{R_1 R_2}{R_1+R_2}$ – and the result is always smaller than either.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn



Deriving the divider. Two resistors in series across a source carry one current $i = v_s/(R_1+R_2)$. The voltage on R_2 is that current times R_2, so $v_2 = v_s \cdot R_2/(R_1+R_2)$ – the **voltage divider**. The mirror argument on two parallel resistors fed by a current source gives the **current divider**: $i_1 = i_s \cdot R_2/(R_1+R_2)$. Notice the swap: each branch takes the share proportional to the **other** resistor.

Kirchhoff's current law (KCL) restates conservation of charge: the signed currents leaving any closed surface sum to zero. A **node** is a junction of two or more elements; every connection of the same wire is one node, however many branches meet there.

Kirchhoff's voltage law (KVL) restates energy conservation: the signed voltages around any closed **loop** sum to zero. Walk a loop and tally the rises and falls: a label that goes up in one traversal is the same label that goes down when you return, so the total must be zero – you cannot gain energy by walking in a circle.

Worked example 1 (divider, checked two ways). A 9 V source feeds $R_1 = 6$ kΩ in series with $R_2 = 3$ kΩ. The current is $i = 9$ V/9 kΩ = 1 mA, so $v_2 = 1$ mA × 3 kΩ = 3 V – the same answer as the divider formula $9 \times 3/9$. Check by KVL: $-9 + 6 + 3 = 0$. Check by power: source 9×1 mA = 9 mW; resistors 6 mW +3 mW = 9 mW.

Worked example 2 (current divider, then the node it feeds). A 2 A source feeds 6 Ω in parallel with 3 Ω. The parallel pair is $6 \times 3/9 = 2$ Ω, so both see 4 V, and the currents are $4/6 = 0.667$ A and $4/3 = 1.333$ A – the divider formula gives the same shares, and $0.667 + 1.333 = 2$ A at the node, KCL closed. Now escalate: at that node a further branch of unknown current i_4 also leaves; if another 0.5 A flows **in** from elsewhere, KCL reads $-0.667 - 1.333 - 0.5 + i_4 = 0$, so $i_4 = 2.5$ A out. Charge did not vanish; it re-routed.

exercises.lol exercises.com.cn

Common mistakes.

- Drawing KCL with some arrows in and some out, then adding magnitudes. Write "sum of currents leaving = 0" and let the signs speak.
- Treating two separate blobs of wire as one node (they are one node only if connected).
- Parallel resistors adding to something **bigger** – the combination is always smaller than the smallest.

Summary: Ohm's law plus two conservation laws solve every DC resistive circuit; dividers come from them, not from memory; label every arrow first, then trust the algebra – a negative answer is information, not a mistake.

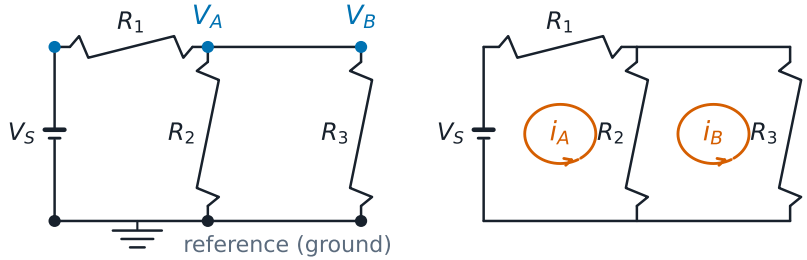
Nodal and mesh analysis

A circuit with four nodes and six branches can be solved by clever guesses – once. The same circuit appears a hundred times in a design, and the guess-work has to become a recipe a computer could run. Two recipes exist; both end in a few simultaneous equations.

Nodal analysis uses KCL. Choose a **reference node** (ground), label the remaining $n-1$ node voltages, and write one KCL equation per unknown node, with each branch current written as a voltage difference over a resistance. This method is the workhorse of circuit simulators (SPICE solves exactly these equations, millions of them).

Mesh analysis uses KVL. Assign a circulating **mesh current** to each window of a planar circuit and write one KVL equation per mesh; a resistor shared by two meshes carries the **difference** of the two mesh currents. Pick the method with fewer equations, and remember: a voltage source fixes a node-voltage difference (nodal gets easier), while a current source fixes a branch current (mesh gets easier).

Nodal: node voltages **Mesh: loop currents**



one KCL equation per unknown node shared branch carries $i_A - i_B$

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

Deriving the nodal equation. At a node A with resistors R_1 to a source node V_s and R_2 to ground, KCL (sum of currents leaving) gives $\frac{V_A - V_s}{R_1} + \frac{V_A - 0}{R_2} = 0$. Multiply through and collect: the equation is linear in V_A – always. Every nodal analysis is this one equation repeated per node.

> **Worked example 1 (one unknown node).** A 12 V source connects through 2 Ω to node A; from A, a 6 Ω resistor goes to ground and a 1 A source pushes current into A. KCL at A, currents out:

$$\frac{V_A - 12}{2} + \frac{V_A}{6} - 1 = 0$$

Multiply by 6: $3(V_A - 12) + V_A - 6 = 0$, so $4V_A = 42$ and $V_A = 10.5$ V. Check with the source branch: $(12 - 10.5)/2 = 0.75$ A flows from the source, plus 1 A from the current source; the 6 Ω carries $0.75 + 1 = 1.75$ A, and $10.5/6 = 1.75$ A – the unused route confirms the answer.

> **Worked example 2 (two meshes, shared resistor).** Mesh A: a 12 V source (rising in the clockwise direction) with a 2 Ω resistor. Mesh B: a 6 Ω resistor with a 3 V source (opposing). The meshes share a 4 Ω resistor. KVL with clockwise mesh currents i_A, i_B:

$$-12 + 2i_A + 4(i_A - i_B) = 0 \qquad 4(i_B - i_A) + 6i_B + 3 = 0$$

Collect: $6i_A - 4i_B = 12$ and $-4i_A + 10i_B = -3$. Solving, $i_A = 2.455$ A and $i_B = 0.682$ A, so the shared resistor carries $i_A - i_B = 1.773$ A downward in mesh A's sense. Check by substituting back into either original equation: both return zero to within rounding.

Common mistakes.

- Forgetting the shared resistor's current is a difference of mesh currents.
- Choosing a method without counting equations: current sources usually fit nodal analysis directly, while voltage sources usually fit mesh analysis directly.
- Assuming the reference node must be a source's negative terminal. Any node can be the reference; choosing a voltage-source endpoint may simplify the equations.

Summary: nodal = KCL per unknown node (simulators' choice); mesh = KVL per window; shared elements see current differences; choose by counting equations.

Source transformations and network theorems

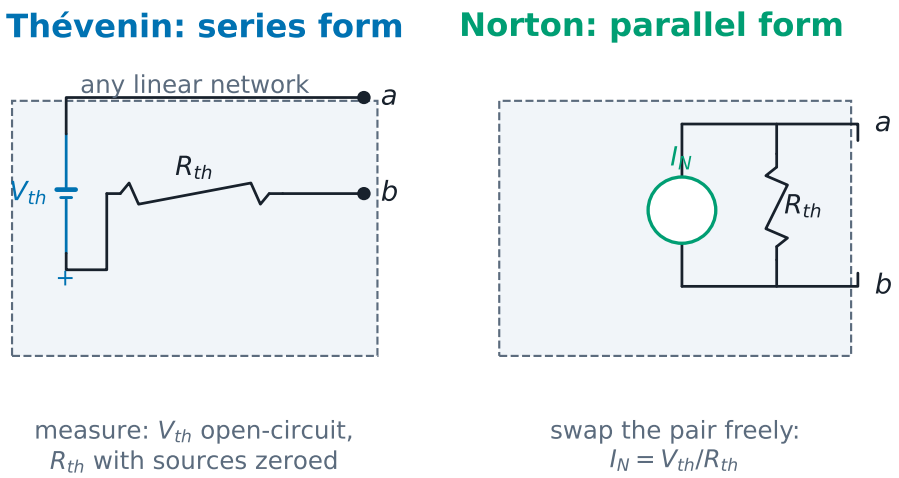
You replace a sensor board and the lamp it feeds brightens slightly. What did the board look like **to the lamp**? Not its schematic – one source and one resistance. Three theorems make that replacement exact, and they all say one thing: a linear network is indistinguishable from its behaviour at two terminals.

- **Thévenin's theorem** : any linear two-terminal network of sources and resistors behaves, at its terminals, like one voltage source V_{th} in series with one

exercises.lol exercises.com.cn

resistance R_{th}. V_{th} is the **open-circuit voltage** ; R_{th} is the resistance seen with sources zeroed (voltage sources shorted, current sources opened).

- **Norton's theorem** is the same statement as a current source $I_N = V_{th}/R_{th}$ in parallel with R_{th}. The two are the **source transformation** : swap between them whenever the arithmetic is friendlier.
- **Superposition** : in a linear circuit with several independent sources, the response is the sum of the responses to each source acting alone (others zeroed). It is a computing strategy, not a property of any one component.



Deriving maximum power transfer. A Thévenin source V_{th}, resistance R_{th}, feeds R_L. The load power is $P = \left(\frac{V_{th}}{R_{th}+R_L}\right)^2 R_L$. Differentiate with respect to R_L and set the result to zero: the maximum sits at $R_L = R_{th}$, where $P_{max} = V_{th}^2/(4R_{th})$ – and the source resistance then burns exactly as much as the load delivers: 50% efficiency, no more.

Maximum power transfer is a *signal* rule, not an *energy* rule: audio and RF systems match impedances to keep signals strong; power grids never do (50% efficiency would double every bill). A Thévenin equivalent does not preserve the internal losses of the original network; its efficiency must be calculated from the actual sources.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Worked example 1 (find the equivalent, then load it). A 10 V source and a 4 Ω resistor form one series branch across a 6 Ω resistor. Find the equivalent seen by an external load connected across that 6 Ω resistor. With the external load disconnected, the terminal voltage is $V_{th} = 10 \times 6/10 = 6$ V (a divider). Zeroing the source leaves 4 Ω parallel with 6 Ω: $R_{th} = 2.4$ Ω. With $R_L = R_{th} = 2.4$ Ω, the loop current is $6/(2.4 + 2.4) = 1.25$ A and the load power is $1.25^2 \times 2.4 = 3.75$ W. Check against the formula: $V_{th}^2/(4R_{th}) = 36/9.6 = 3.75$ W.

Worked example 2 (superposition, checked head-on). A 12 V source through 2 Ω feeds node A; a 3 A source also injects into A; 4 Ω runs from A to ground. By superposition: the 12 V source alone (3 A opened) drives the 4 Ω through a $2 + 4$ divider: $V_{A1} = 12 \times 4/6 = 8$ V. The 3 A source alone (12 V shorted) sees $2\ \Omega \parallel 4\ \Omega = 1.333\ \Omega$: $V_{A2} = 3 \times 1.333 = 4$ V. Total $V_A = 8 + 4 = 12$ V. Direct check by one nodal equation: $(V_A - 12)/2 + V_A/4 - 3 = 0$ gives $V_A = 12$ V – the two methods agree exactly.

Common mistakes.

- Computing R_{th} with independent sources still active. Deactivate them: short ideal independent voltage sources and open ideal independent current sources. Retain dependent sources and use a test source when needed.
- Applying superposition to **power** (it is linear in voltage and current, not in their squares – compute responses first, then square).
- Believing the 50%-efficiency point is a design target for power delivery.

Summary: measure two things (V_{th} open-circuit, R_{th} zeroed) and any linear one-port becomes a battery and a resistor; swap series↔parallel forms freely; add responses, never powers; match resistance for maximum signal power, not for economy.

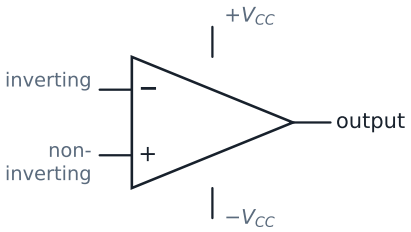
Ideal operational amplifiers

A microphone produces 10 mV. A speaker wants 2 V. Something must multiply by 200 – with a dial to set exactly how much. That something is the **operational amplifier**, and the whole unit is about one trick: feedback makes an absurdly high-gain device behave like an exact, mild-gain one.

An **operational amplifier** is a differential voltage amplifier with very large gain, drawn as a triangle with + and − inputs and one output. Real op-amps have a **gain-bandwidth product** and finite input currents; the **ideal op-amp** model used in analysis assumes infinite open-loop gain, infinite input resistance and zero output resistance.

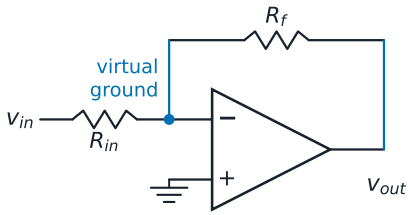
exercises.lol exercises.com.cn

Five terminals



ideal: infinite gain,
infinite input resistance

Inverting amplifier



the − node sits at 0 V, so
all of v_{in}/R_{in} flows through R_f

Why the golden rules follow from feedback. With **negative feedback** (a path from output back to the − input), the output does whatever is needed to make the input difference nearly zero – with gain above 10^5, an output of volts needs an input difference of microvolts, which rounds to zero. Two **golden rules** hold:

1. No current flows into either input (infinite input resistance).
2. The voltage difference between the inputs is zero – a **virtual short** . The inputs track each other without being connected.

Deriving the inverting gain. The workhorse is the **inverting amplifier** : input through R_{in} to the − node, feedback through R_f, + input grounded. Rule 2 pins the − node at 0 V (a **virtual ground**). Rule 1 says the input current v_{in}/R_{in} cannot enter the chip, so it all flows through R_f:

$$\frac{v_{in}}{R_{in}} = -\frac{v_{out}}{R_f} \;\Rightarrow\; \frac{v_{out}}{v_{in}} = -\frac{R_f}{R_{in}}$$

The gain is set by two resistors – the chip's enormous gain cancels out of the answer. The **non-inverting amplifier** puts the signal at + and a divider at −: $v_{out}/v_{in} = 1 + R_f/R_{in}$, always at least unity (at unity it is the **voltage follower** , a buffer).

> **Worked example 1 (inverting stage, fully accounted).** $R_{in} = 10\,\text{k}\Omega$, $R_f = 100\,\text{k}\Omega$, $v_{in} = +0.2$ V. Gain is $-100/10 = -10$, so $v_{out} = -2$ V, comfortably inside ±15 V supplies. The feedback current is $0.2/10\text{k} = 20\ \mu$A and flows into the output; the ideal input carries no current. Check by the virtual ground: the − node sits at 0 V, so the drop on R_f is $0 - (-2) = 2\text{ V} = 20\ \mu\text{A} \times 100\ \text{k}\Omega$.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

> **Worked example 2 (a mixer, and a saturation trap).** Extend the inverting stage to two inputs ($R_1 = R_2 = 10\ \text{k}\Omega$, same $R_f = 100\ \text{k}\Omega$): the virtual ground collects both currents, so $v_{out} = -\left(\frac{R_f}{R_1}v_1 + \frac{R_f}{R_2}v_2\right)$ – a **summing amplifier** . With $v_1 = 0.2$ V and $v_2 = -0.05$ V: $v_{out} = -(10 \times 0.2 + 10 \times (-0.05)) = -(2 - 0.5) = -1.5$ V. Now the trap: swap R_f for 1 MΩ (gain −100 per input) and the same inputs ask for $-(100 \times 0.2 + 100 \times (-0.05)) = -15$ V. The **saturation** rule stops it: the output cannot leave its supply rails, and a ±15 V op-amp rests near −13.5 V. The golden rules have silently stopped applying – report the clamp, not the impossible voltage.

Common mistakes.

- Forgetting the inverting stage's minus sign (or "correcting" it away).
- Applying rule 2 when there is no negative feedback path – no feedback, no virtual short.
- Pushing the answer past the rails and reporting it anyway.

Summary: infinite gain tamed by feedback gives two golden rules; the inverting stage turns them into $-R_f/R_{in}$, the non-inverting into $1 + R_f/R_{in}$; virtual ground makes mixers; saturation is a real output state, not a rounding error.

Capacitors, inductors and energy

Every flash in a phone camera collects energy slowly and releases it in a millisecond. No arrangement of resistors can hold anything – resistors burn energy; the two new elements **store** it, and that single change is what makes circuits dynamic and worth a calculus course.

A **capacitor** stores charge separated by an insulator: two plates with electric field between them. Its **capacitance** C, in **farads** (F), is charge per volt, $q = Cv$. Differentiate: current is the rate of change of voltage,

$$i = C\frac{dv}{dt}$$

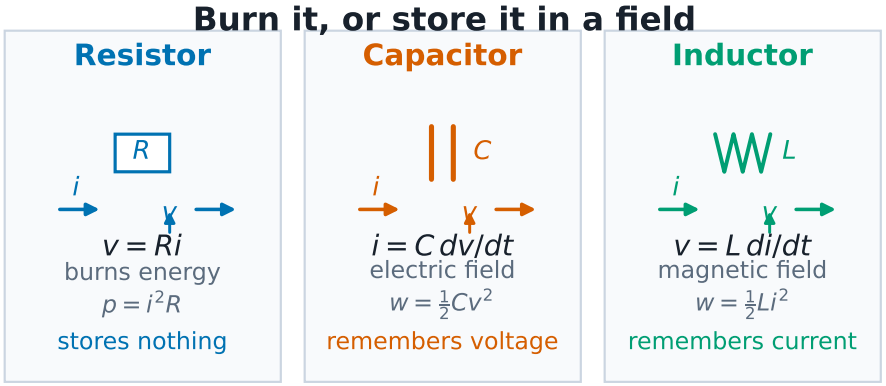
so a capacitor carries current only while its voltage changes: steady DC is an open circuit to it. Integrating the same law gives the stored energy $w = \frac{1}{2}Cv^2$ – a state memory that depends only on the present voltage, not on how it got there.

An **inductor** stores energy in a magnetic field around a coil. Its **inductance** L, in **henrys** (H), gives the mirror law,

$$v = L\frac{di}{dt}$$

so an inductor develops voltage only while its current changes: steady DC is a short circuit to it. Stored energy is $w = \frac{1}{2}Li^2$.

exercises.lol exercises.com.cn



Why the combinations mirror the resistor rules. Parallel capacitors share one voltage, so charges add: $C_{eq} = C_1 + C_2$ (areas add). Series inductors share one current, so flux linkages add: $L_{eq} = L_1 + L_2$ (lengths add). Series capacitors and parallel inductors combine by the reciprocal rule – the exact mirror of resistor arithmetic, because the storage laws swap the roles of v and i.

> **Worked example 1 (two energies, two units).** A 100 µF capacitor charged to 12 V holds $w = \frac{1}{2} \times 100 \times 10^{-6} \times 12^2 = 7.2$ mJ. A 10 mH inductor carrying 2 A holds $w = \frac{1}{2} \times 0.01 \times 2^2 = 20$ mJ. Both are small by kettle standards, but the capacitor's 7.2 mJ delivered in 1 ms is 7.2 W – and in 1 µs it would be 7.2 kW. Stored energy is modest; **rate** makes it dramatic, which is why large capacitors are discharged before you touch them.

> **Worked example 2 (slope law and combinations).** A 10 µF capacitor's voltage rises at 100 V/s; the current is $i = C\,dv/dt = 10 \times 10^{-6} \times 100 = 1$ mA – constant, because the slope is constant. Now combine: a 2 µF and a 3 µF capacitor in series give $2 \times 3/5 = 1.2$ µF (each holds the same charge, so the "capacity per volt" shrinks); in parallel they give 5 µF. A 10 mH and 20 mH inductor in series give 30 mH; the same two in parallel give $10 \times 20/30 = 6.67$ mH.

△ An inductor's current is continuous: opening a switch in an inductor's path forces di/dt to be huge, and the induced voltage $L\,di/dt$ can arc across the opening contacts. This is real, not theoretical – it is how old ignition coils made their spark.

Common mistakes.

- Using $w = \frac{1}{2}Cv^2$ with volts swapped for something else – check the units: farads times volts squared is joules.
- Adding parallel capacitances as reciprocals (that is the series rule).
- Believing a capacitor "blocks" DC instantly – it blocks **steady** DC; during changes it carries current gladly.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Summary: capacitors remember voltage, inductors remember current; $i = C\,dv/dt$ and $v = L\,di/dt$; energies $\frac{1}{2}Cv^2$, $\frac{1}{2}Li^2$; combinations mirror resistor arithmetic with v and i swapped; never open an inductor's path suddenly.

First-order RC and RL response

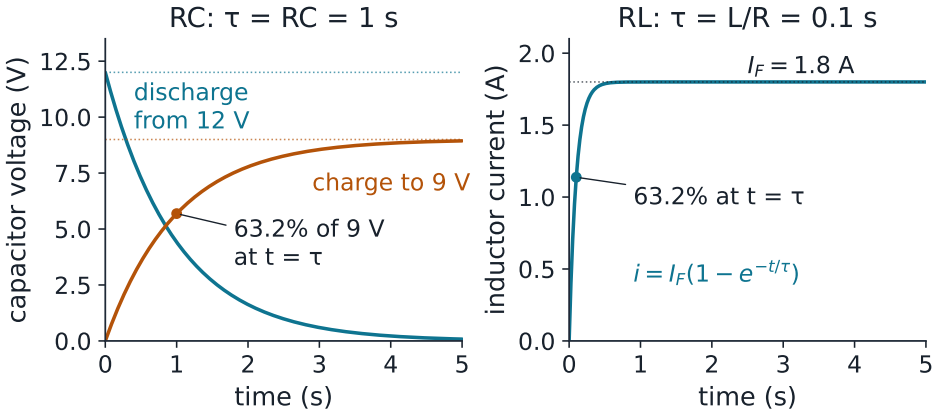
You press a button and the light fades up over a second – or the sensor reading creeps to its new value over milliseconds. Switch a source, an energy-storing element and a resistor into a new arrangement, and the stored quantity cannot jump: it slides to its new steady value. A circuit with one storage element is **first order**, and every such circuit has the same shape of answer – learn the shape once.

Define the **natural response** as the decay from an initial state with the source removed, and the **step response** as the move from an initial state to a new forced value V_F. For a capacitor voltage,

$$v(t) = V_F + (V_0 - V_F)\,e^{-t/\tau}$$

Deriving it. After the switch, a capacitor C discharges through R: KCL gives $C\,dv/dt + v/R = 0$. Separate variables: $dv/v = -dt/(RC)$, integrate: $\ln v = -t/(RC) + \text{const}$, so $v = V_0 e^{-t/RC}$ – an exponential with **time constant** $\tau = RC$. Add a source and the same equation gains a constant solution V_F; the general answer is the constant plus a decaying difference: exponential to the new value, always. The RL mirror replaces $\tau = RC$ with $\tau = L/R$.

At $t = \tau$ the gap has closed to 37% of its starting size; after 5τ the circuit is within 1% and is treated as settled. The initial value V_0 comes from the stored energy (voltage continuity for C, current continuity for L), and V_F from the circuit the switch left behind, with the storage element treated as its steady-state open (C) or short (L).



exercises.lol exercises.com.cn

Worked example 1 (RC discharge). A 100 µF capacitor charged to 12 V discharges through 10 kΩ. $\tau = RC = 10\,000 \times 100 \times 10^{-6} = 1$ s, so $v(t) = 12e^{-t}$ volts. After 2 s, $v = 12 \times e^{-2} = 1.62$ V. After 5 s (5τ) it is $12e^{-5} = 0.08$ V – settled. Check the energy story: the initial 7.2 mJ has all gone into the resistor by then.

Worked example 2 (charging between two non-zero values). The same RC network, but the switch moves the resistor from a 2 V supply to a 9 V supply with the capacitor already at 2 V. $V_0 = 2$, $V_F = 9$, so $v(t) = 9 + (2 - 9)e^{-t} = 9 - 7e^{-t}$ volts. At $t = 1$ s ($= \tau$) it reaches $9 - 7 \times 0.368 = 6.42$ V – the gap (7 V) has closed by 63.2%, exactly the definition. Endpoint checks: $v(0) = 2$ V and $v(5) = 9 - 0.047 \approx 9$ V.

Worked example 3 (the RL mirror, fast). A 20 mH inductor in series with 100 Ω switches onto a 10 V supply. $\tau = L/R = 0.02/100 = 0.2$ ms, final current $I_F = 10/100 = 0.1$ A, so $i(t) = 0.1(1 - e^{-t/0.2\text{ms}})$. At $t = 0.6$ ms (three time constants) it has reached 95%: 0.095 A. The same shape as the capacitor voltage – one law, two elements.

Method for every switching problem: (1) find V_0 from the old circuit; (2) find V_F from the new circuit; (3) compute τ from the new circuit with the source zeroed; (4) write the one-line answer and check it against both endpoints.

Common mistakes.

- Using the old circuit for V_F or the new one for V_0 – the values come from different snapshots, deliberately.
- Computing τ with the source still present – zero the source to see the true resistance the storage element faces.
- Believing 5τ is a physical constant – it is a 1% engineering convention.

Summary: one storage element means one exponential; V_0 from continuity, V_F from the final circuit, τ from the zeroed-source resistance; $\tau = RC$ or L/R; 5τ is settled for practical purposes, and every answer checks at both ends.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

Mathematical Methods for EE

University Electrical Engineering

What this course is, and how to use it

This is the second course in University Electrical Engineering: **mathematical methods** – the toolkit the rest of the degree runs on. Wave 1's DC circuits needed only algebra; from AC circuits onward, every method rests on something taught here.

- Units 2.1–2.3 refresh and extend the algebra: calculus for signals, complex numbers, linear algebra.
- Units 2.4–2.6 are the dynamics toolkit: differential equations, Laplace transforms, Fourier series.
- Units 2.7–2.8 handle randomness and fields: probability for noise, vector calculus for electromagnetics.

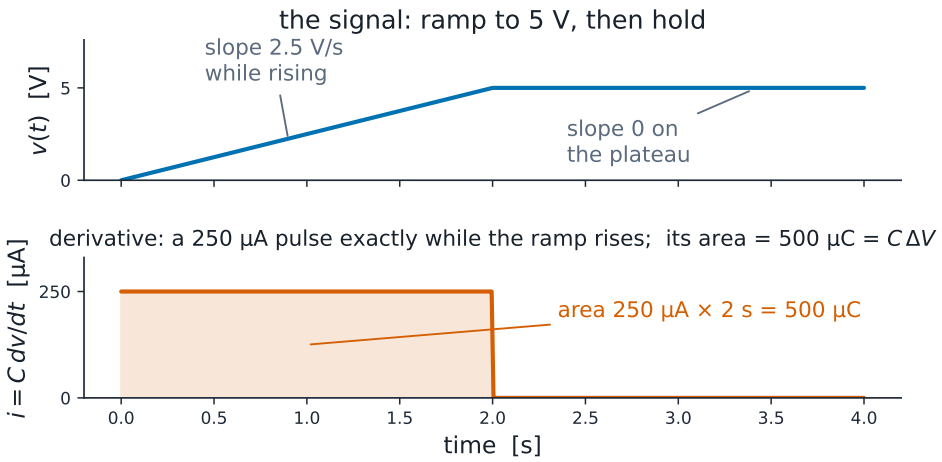
Each unit opens with a story, shows a picture before the formal idea, defines terms precisely, derives what later courses quote, works two escalating examples whose numbers are machine-verified, names the classic mistakes, and closes with a three-line summary. The habit is the same one the whole degree uses: check every symbolic answer numerically at a concrete value – an integral by a sum, a solved system by substitution, a transform pair by computing both sides.

Calculus review for signals

A scope trace rises smoothly from 0 to 5 V and flattens. The current into a capacitor at every instant depends on *how fast* the trace moves – slope – and the charge delivered over an interval is the *area* under the current trace. Signals are functions of time, and calculus reads them: the **derivative** dv/dt is the slope – instantaneous rate of change; the **integral** $\int i\,dt$ is accumulated area.

$$\frac{d}{dt}t^n = n\,t^{n-1} \qquad \int_0^T t^n\,dt = \frac{T^{n+1}}{n+1}$$

exercises.lol exercises.com.cn



The same signal, read twice: slope on top, area below

For the EE work ahead, three facts do most of the work:

- The derivative of e^{at} is $a \cdot e^{at}$ – exponential rates are their own slopes.
- The integral of a **rectangular pulse** (height A, width τ) is area $A\tau$.
- A triangle's area is half the enclosing rectangle – and $\int_0^\infty Ae^{-t/\tau}\,dt = A\tau$ (the exponential decays over its time constant, so the "effective width" is τ).

Deriving e^{at}'s self-slope. The limit definition: $\frac{d}{dt}e^{at} = \lim_{h\to 0}\frac{e^{a(t+h)}-e^{at}}{h} = e^{at}\lim_{h\to 0}\frac{e^{ah}-1}{h} = e^{at}\cdot a$ – the exponential factors out of its own difference quotient, which is the whole reason circuits that vary as e^{at} stay that way under differentiation.

> **Worked example 1 (the linear ramp, verified).** A capacitor charging linearly from 0 to 5 V in 2 s ($v = 2.5t$): with $C = 100\ \mu F$, $i = C\,dv/dt = 100\mu \times 2.5 = 250\ \mu A$ – constant, because the ramp's slope is constant. The charge delivered is $\int i\,dt = 250\mu \times 2 = 500\ \mu C$, which also equals $C \times \Delta V = 100\mu \times 5$. Two methods, one answer – that is the checking habit.

> **Worked example 2 (the exponential pulse, escalated).** A sensor pulse decays as $i(t) = 5e^{-t/0.2}$ mA (peak 5 mA, time constant $\tau = 0.2$ s). The total charge is the area $\int_0^\infty 5e^{-t/0.2}dt = 5 \times 0.2 = 1.0$ mC – the "effective width" rule in action. The peak slope is at $t = 0$: $\frac{di}{dt} = -\frac{5}{0.2} = -25$ mA/s. Check by crude summation: one rectangle of height 5 mA over the width 0.2 s gives exactly the same 1.0 mC – which is not a coincidence but the meaning of τ.

Common mistakes. Differentiating a ramp's *value* instead of its slope (constant 5 V does not mean constant current – only constant *slope* does). Forgetting that integration is signed: area below the axis subtracts. Dropping units inside the integral ($\mu A \times s = \mu C$, always).

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

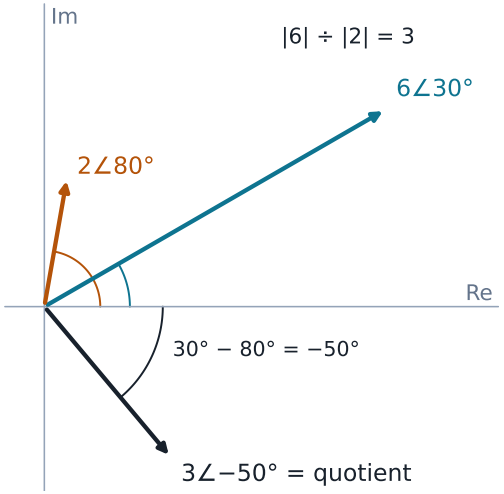
Summary: derivative = slope, integral = signed area; exponentials are their own slopes; a pulse's area is height × effective width (τ for a decaying exponential) – and every answer has a unit.

Complex numbers and phasor algebra

An AC voltage is 230 V *and* some phase – one number cannot hold both, but a point on a plane can. A **complex number** $z = a + jb$ (with $j^2 = -1$) has a real part, an imaginary part, a **modulus** $|z| = \sqrt{a^2+b^2}$ and an **argument** $\theta = \text{atan2}(b, a)$. The two forms are interchangeable:

$$a + jb = r(\cos\theta + j\sin\theta) = r\,e^{j\theta}$$

– **Euler's identity** , the bridge EE uses everywhere. Multiplication adds arguments: $r_1e^{j\theta_1} \cdot r_2e^{j\theta_2} = r_1r_2e^{j(\theta_1+\theta_2)}$ – a rotation and a scaling, which is exactly what a sinusoid's phase shift and amplitude change look like.



Division in polar form: $6 \div 2 = 3$ and $30° - 80° = -50°$

> **Worked example 1 (divide two phasors, checked by the inverse).** Divide $6\angle 30°$ by $2\angle 80°$: in polar form the answer is $3\angle -50°$ directly – moduli divide, arguments subtract. Doing the same in rectangular form means rationalising $(5.196 + j3)/(0.347 + j1.970)$ – possible, but the polar route is why EE works in **phasors** . A third check: $3\angle -50° \times 2\angle 80° = 6\angle 30°$ – the inverse operation must return the start.

exercises.lol exercises.com.cn

Worked example 2 (a quadratic with complex roots, escalated). A damped system's characteristic equation $s^2 + 2s + 5 = 0$ has roots $s = \frac{-2 \pm \sqrt{4-20}}{2} = -1 \pm j2$. In polar form the roots are the conjugate pair $2.236\angle \pm 116.6°$ (modulus $\sqrt{1+4} = 2.236$). Check by substitution: $(-1+j2)^2+2(-1+j2)+5 = (-3-j4)+(-2+j4)+5 = 0$ exactly. The roots' real part -1 is the decay rate and the imaginary part 2 rad/s the ringing frequency – one complex number carrying both behaviours, which is why the rest of the degree keeps them together.

Common mistakes. Adding moduli instead of adding complex numbers (moduli do not add; rectangular parts do). Using degrees inside $e^{j\theta}$ formulas that assume radians. Confusing j rotation direction: multiplying by j rotates $+90°$, by $-j$ rotates $-90°$.

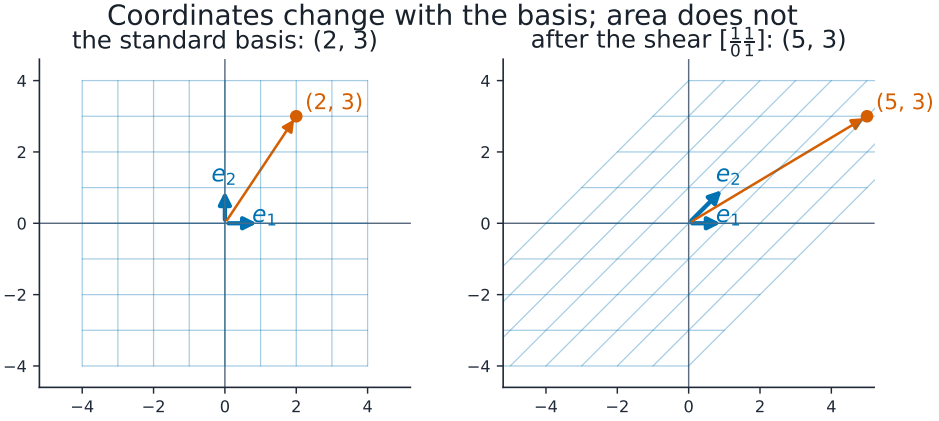
Summary: two forms, one number – rectangular for adding, polar for multiplying and dividing; Euler's identity is the bridge; roots of quadratics carry decay in the real part and frequency in the imaginary part.

Linear algebra essentials

Three mesh currents in a coupled circuit mean three equations in three unknowns – and the structure of the solution is not "clever substitution" but a matrix. A **linear system** $Ax = b$ is solved by elimination; a **matrix inverse** A^{-1} exists only when the **determinant** is non-zero. For the 2×2 case:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

The determinant $ad - bc = 0$ means the equations are **singular** – no unique solution, and in circuits a dependent source loop or a floating node.



A basis spans the space; a matrix moves between bases

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Deriving the 2×2 inverse. Multiply A by its candidate: $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \frac{1}{ad-bc} \begin{pmatrix} ad-bc & 0 \\ 0 & ad-bc \end{pmatrix} = I$ – the swap-and-negate pattern exists precisely because those cross-terms cancel.

Worked example 1 (elimination and inverse agree). Solve $x+2y = 8$, $3x+y = 9$. Elimination: subtract 3×row1 from row2 → $-5y = -15$, so $y = 3$, $x = 2$. Check by the inverse: $\det = 1\cdot1-2\cdot3 = -5$; $A^{-1} = \frac{1}{-5}\begin{pmatrix} 1 & -2 \\ -3 & 1 \end{pmatrix}$; $x = (1\cdot8-2\cdot9)/-5 = 2$ ✓, $y = (-3\cdot8+1\cdot9)/-5 = 3$ ✓. Both routes agree; each is the other's check.

Worked example 2 (the mesh system, machine-solved, escalated). The EE01 two-mesh circuit closes as one matrix equation $\begin{pmatrix} 6 & -4 \\ -4 & 10 \end{pmatrix}\begin{pmatrix} i_A \\ i_B \end{pmatrix} = \begin{pmatrix} 12 \\ -3 \end{pmatrix}$. Elimination (or the inverse; $\det = 60 - 16 = 44$) gives $i_A = 2.455$ A, $i_B = 0.682$ A – verified by substitution into both original equations to machine precision. The negative off-diagonal entries -4 are the shared resistor speaking: coupling shows up as off-diagonal terms, and the bigger the coupling, the further the determinant sits from the product of the diagonals. Now the singular case: $x + 2y = 8$, $2x + 4y = 16$ has determinant $1\cdot4 - 2\cdot2 = 0$ – the second equation is just twice the first (a whole line of solutions, not one point).

Common mistakes. Mixing the row order between A and b. Assuming a non-zero determinant guarantees an easy solution – it guarantees uniqueness, not convenience. Forgetting that singular systems can have *no* solution (parallel planes) or *infinitely many* (dependent planes): the determinant cannot tell which.

Summary: $Ax = b$ with $\det \neq 0$ has exactly one solution; elimination computes it, the inverse re-derives it, and substitution checks it; coupling lives in the off-diagonals; $\det = 0$ means the geometry collapsed.

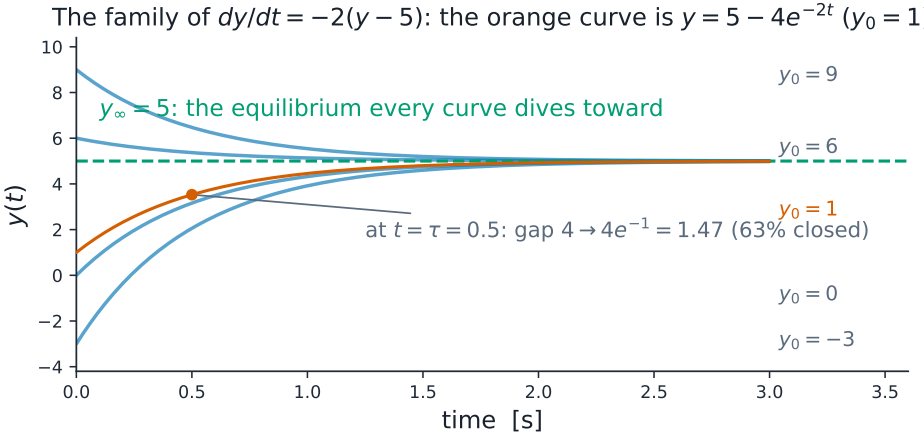
Ordinary differential equations

The switch closes at $t = 0$ and the capacitor voltage slides to its new value – wave 1's RC response was one instance of a pattern this unit makes general. The circuit equations are **ordinary differential equations** . A first-order linear ODE with constant coefficients,

$$\frac{dy}{dt} + ay = f(t),$$

has solution = **homogeneous** part Ce^{-at} + **particular** part forced by f. The homogeneous term carries the **initial condition** .

exercises.lol exercises.com.cn



One equation, a family of curves; the initial condition selects one

Why the solution has this shape. For $\frac{dy}{dt} = -a(y - y_\infty)$, rewrite as $\frac{dy}{y-y_\infty} = -a\,dt$: each side is now integrable, giving $\ln(y - y_\infty) = -at + C$, i.e. $y - y_\infty = \tilde{C}e^{-at}$. The gap to the final value decays exponentially – every first-order system slides down its own percentage curve.

> **Worked example 1 (the step-solved RC, endpoint-checked).** $dy/dt + 2y = 10$, $y(0) = 1$. Homogeneous: Ce^{-2t}. Particular (constant input → constant output): $y_p = 10/2 = 5$. So $y = 5 + Ce^{-2t}$; $y(0) = 5 + C = 1$ gives $C = -4$: $y = 5 - 4e^{-2t}$. Checks: at $t = 0$, $y = 1$ ✓; as $t \to \infty$, $y \to 5$ and $dy/dt \to 0$, so the equation reads $0 + 2(5) = 10$ ✓. This is the RC equation of unit 1.7 in general form – the method, not the circuit, is what this course teaches.

> **Worked example 2 (a decaying input, machine-verified, escalated).** Solve $\frac{dy}{dt} + 3y = 6e^{-t}$, $y(0) = 2$. Try a particular $y_p = ke^{-t}$: $-ke^{-t} + 3ke^{-t} = 2ke^{-t} = 6e^{-t}$ gives $k = 3$. So $y = 3e^{-t} + Ce^{-3t}$; $y(0) = 3 + C = 2$ gives $C = -1$: $y = 3e^{-t} - e^{-3t}$. Check at $t = 0.5$: $y = 3(0.6065) - 0.2231 = 1.596$. Substituting into the left side: $dy/dt + 3y = -1.150 + 3(1.596) = 3.639$, and the right side is $6e^{-0.5} = 3.639$ – equal to machine precision. The response follows two exponentials, the input's (-1) and the system's (-3).

Common mistakes. Solving the homogeneous equation and forgetting the particular part entirely. Letting the initial condition choose k in the *particular* guess (only the homogeneous constant is free). Checking only $t = 0$: also check the large-t limit, where the equation must balance with $dy/dt \approx 0$.

Summary: homogeneous + particular; the initial condition selects from the family; separation of variables explains the exponential shape; endpoint checks at both ends are free.

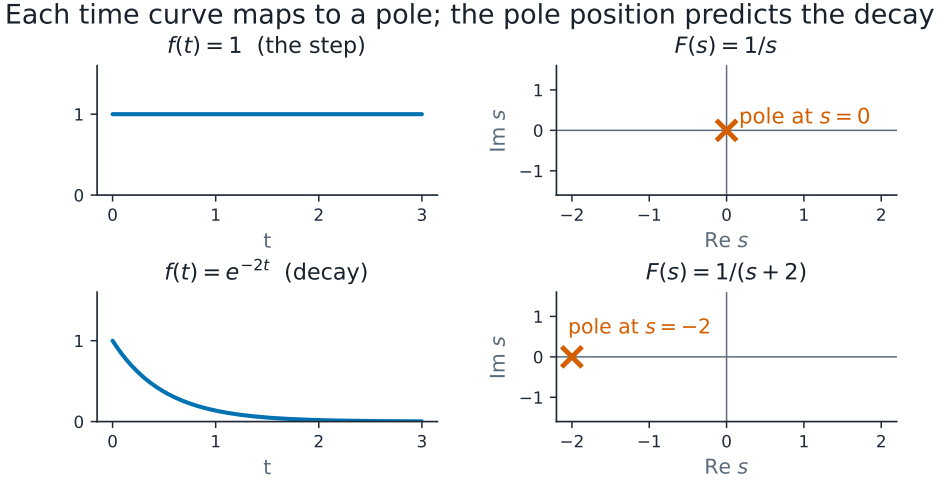
exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Laplace transforms

Solving ODEs by hand works for one equation; a circuit with three storage elements needs three. The **Laplace transform** turns differentiation into multiplication by s, converting entire systems into algebra:

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty f(t)\,e^{-st}\,dt, \qquad \mathcal{L}\{e^{at}\} = \frac{1}{s-a}$$

Deriving the key pair. $\int_0^\infty e^{at}e^{-st}dt = \int_0^\infty e^{-(s-a)t}dt = \frac{1}{s-a}$ (for $\mathrm{Re}\,s > a$) – the exponential eats its own decay, leaving a simple pole. That single computation generates most of the table.



Each time curve maps to a pole; pole position predicts decay

Three working rules: $\mathcal{L}\{dy/dt\} = sY - y(0)$ (the initial condition enters algebraically); $\mathcal{L}\{1\} = 1/s$; and **partial fractions** split a rational $F(s)$ into invertible pieces.

> **Worked example 1 (the unit-2.4 ODE, by transform).** Solve $dy/dt + 2y = 10$, $y(0) = 1$. Transformed: $sY - 1 + 2Y = 10/s$, so $Y = (10/s + 1)/(s + 2) = 10/[s(s + 2)] + 1/(s + 2)$. Partial fractions: $10/[s(s + 2)] = 5/s - 5/(s + 2)$. So $Y = 5/s - 4/(s + 2)$, and inverting term by term: $y = 5 - 4e^{-2t}$ – identical to the classical route, with the initial condition handled by bookkeeping instead of algebra at the end. The **poles** of Y (here $s = 0, -2$) predict the behaviour: a constant and a decaying exponential.

exercises.lol exercises.com.cn

Worked example 2 (second order, machine-verified, escalated). $\frac{d^2y}{dt^2}+3\frac{dy}{dt}+2y=0$, $y(0)=1$, $y'(0)=0$. Transform: $(s^2+3s+2)Y=s+3$ (the derivative rule applied twice), so $Y=\frac{s+3}{(s+1)(s+2)}=\frac{2}{s+1}-\frac{1}{s+2}$. Invert: $y=2e^{-t}-e^{-2t}$. Checks: $y(0)=2-1=1$ ✓; $y'(0)=-2+2=0$ ✓; $y(1)=0.736-0.135=0.600$ (machine-verified). The two poles $-1,-2$ *are* the two decays, with the slower pole -1 dominating the tail – the observation that later courses turn into dominant-pole approximations.

Common mistakes. Forgetting the $-y(0)$ in the derivative rule (and its second-order cousins $-y'(0)-s\,y(0)$). Inverting a Y whose fractions are not fully split: $\frac{s+3}{(s+1)(s+2)}$ must become two simple pieces before any table lookup. Reading a zero at the numerator as a pole (a zero blocks; a pole decays).

Summary: differentiation → multiplication by s, initials included; partial fractions split Y; poles predict the time behaviour; the transform and classical routes must agree – that agreement is the check.

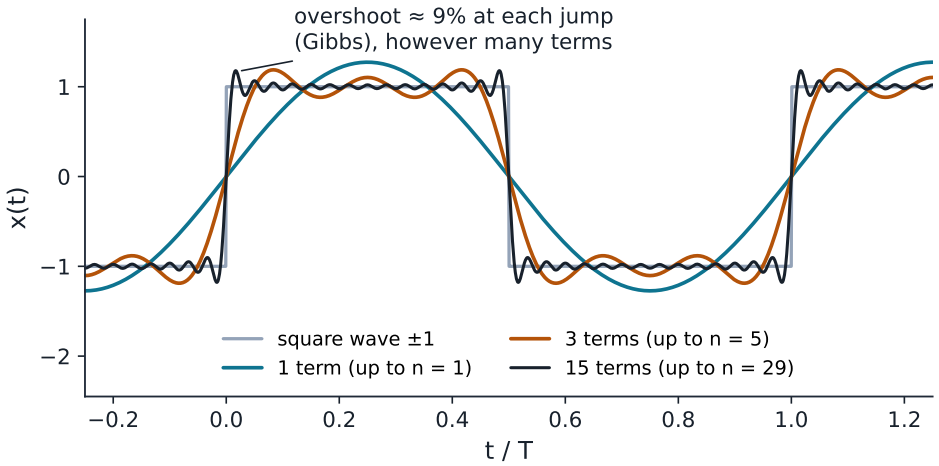
Fourier series

A distorted mains waveform heats a resistor exactly as some combination of pure sinusoids would – and which combination is not a guess but a computation. Any periodic signal is a sum of sinusoids. For period T, **fundamental** $\omega_0=2\pi/T$, the **Fourier series** is

$$f(t)=a_0+\sum_{n\geq 1}\big(a_n\cos n\omega_0 t+b_n\sin n\omega_0 t\big),\qquad a_0=\frac{1}{T}\int_T f\,dt$$

with each a_n, b_n an average of f against the matching sinusoid (**orthogonality** : distinct sinusoids average to zero against each other – so each coefficient can be picked out independently, like a bank of tuned filters).

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn



Odd harmonics $4/(n\pi)$ build the square wave; the Gibbs overshoot never goes away

Worked example 1 (the square wave, verified at the magic point). A square wave, ±1, period T, odd symmetry: all $a_n=0$, and $b_n=4/(n\pi)$ for odd n – the series $\frac{4}{\pi}(\sin\omega_0 t+\frac{1}{3}\sin 3\omega_0 t+\frac{1}{5}\sin 5\omega_0 t+\dots)$. The check that convinces: at $t=T/4$ the sum must give 1, and $\frac{4}{\pi}(1-\frac{1}{3}+\frac{1}{5}-\dots)$ is the Gregory series for $\pi/4$ scaled – it converges to 1, exactly as the square wave demands. Every **harmonic** amplitude falls as $1/n$: sharper edges need more harmonics, which is the whole of filtering.

Worked example 2 (power in the harmonics, machine-verified, escalated). A ±1 V square wave across 1 Ω delivers average power 1 W (the mean of a constant magnitude). Parseval says the harmonic powers must sum to the same 1 W: $\sum_{n\ \text{odd}}\frac{1}{2}\big(\frac{4}{n\pi}\big)^2=\frac{8}{\pi^2}\big(1+\frac{1}{9}+\frac{1}{25}+\dots\big)$. Summing the first 20 odd harmonics gives 0.990 W; the full series converges to 1 W exactly (the sum of reciprocal odd squares is $\pi^2/8$, making the total $\frac{8}{\pi^2}\cdot\frac{\pi^2}{8}=1$). The practical reading: the fundamental alone carries $\frac{8}{\pi^2}\approx 81\%$ of the power – a low-pass filter that keeps only the first harmonic still delivers most of the energy, and the 1/3-amplitude third harmonic adds just 9%.

Common mistakes. Using $1/T$ (hertz) instead of $\omega_0=2\pi/T$ (rad/s) as the fundamental frequency. Forgetting the $\frac{1}{2}$ in each harmonic's power (a sinusoid's mean square is half its peak square). Assuming even symmetry everywhere: shift the origin and coefficients change – the spectrum belongs to the function as placed.

Summary: periodic signals decompose into harmonics; orthogonality picks each coefficient independently; amplitudes fall as $1/n$ for edges; Parseval audits the power; the Gibbs overshoot never leaves.

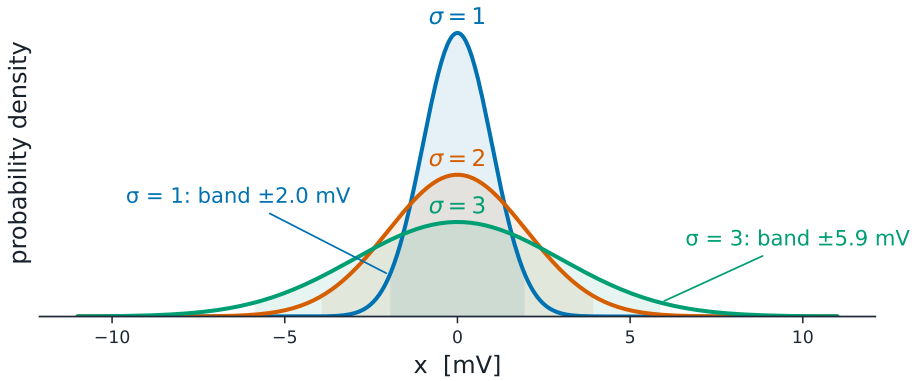
exercises.lol exercises.com.cn

Probability and random variables

Every measurement is noisy; the noise itself is a signal you cannot predict – but its *statistics* are stable, and that stability is what instruments exploit. A **random variable** has a **probability distribution**; the Gaussian (normal) $N(\mu, \sigma^2)$ models thermal noise:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

Three Gaussians, one area: each shaded band is the 95% interval ±1.96 σ



Same area, different spread: σ sets the width

Expectation $E[X]$ is the long-run average; **variance** $Var(X) = E[X^2] - \mu^2$ is the spread, and σ – the **standard deviation** – is the noise amplitude an oscilloscope shows. Two rules do most of the work: independent variances *add* ($\sigma_{total} = \sqrt{\sigma_1^2 + \sigma_2^2}$), and the average of n independent samples has standard deviation $\sigma/\sqrt{n}$.

> **Worked example 1 (the 2σ rule and averaging).** For $X \sim N(0, \sigma^2)$ with $\sigma = 2$ mV: $E[X] = 0$, $Var = 4$ (mV)^2, $\sigma = 2$ mV. The chance of exceeding $\pm 2\sigma$ is 4.55% (the two-tailed 95% band is $\pm 1.96\sigma \approx \pm 3.9$ mV). Averaging 16 independent samples yields $2/\sqrt{16} = 0.5$ mV of noise – the entire rationale of instrument averaging.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

> **Worked example 2 (two noise sources, dB accounting, escalated).** An amplifier stage adds 2 mV of noise to a signal that already carries 3 mV from the previous stage. The combined noise is not $3+2 = 5$ mV but $\sqrt{3^2 + 2^2} = \sqrt{13} = 3.61$ mV – variances add, not deviations, so the smaller source contributes proportionally less than its face value. Signal-to-noise improvement from averaging those same 16 readings: $20\log_{10}(4) = 12.0$ dB. The design reading: sixteen readings buy 12 dB, and each further 6 dB costs four times as many readings (and four times the measurement time); fixing the dominant noise source at its origin costs no measurement time – halving the 3 mV to 1.5 mV would give $\sqrt{1.5^2 + 2^2} = 2.5$ mV, a 3.2 dB improvement on every single reading.

Common mistakes. Adding standard deviations instead of variances. Confusing σ and σ^2 in tables and formulas (the scope shows σ; formulas usually use σ^2). Forgetting that $\sigma/\sqrt{n}$ improvement assumes independence – correlated noise (mains hum) does not average away.

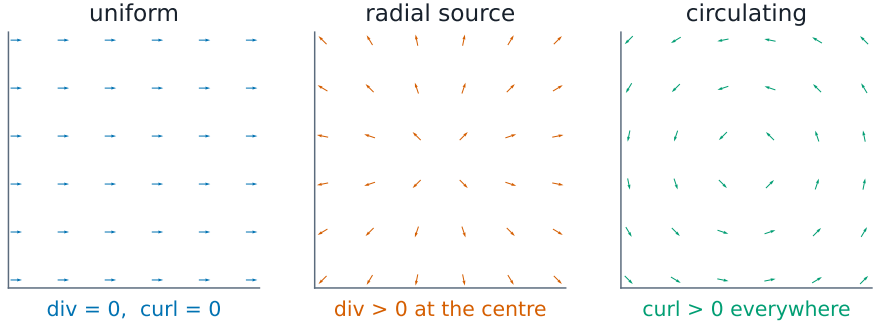
Summary: the Gaussian is the noise model; σ is the scope's noise; independent variances add, averaging divides σ by √n; attack the dominant source, then average.

Vector calculus for fields

A voltage probe reports one number at each point in space – a whole landscape of values, and EE07's Maxwell equations are statements about that landscape's geometry. Fields are vector-valued functions of position. Three operators on a field **F**:

- **Gradient** ∇V: points up the steepest slope of a scalar; the electric field is $E = -\nabla V$.
- **Divergence** $\nabla \cdot \mathbf{F}$: the net outflow per unit volume – charge density for electric fields.
- **Curl** $\nabla \times \mathbf{F}$: the circulation per unit area – zero for electrostatics, the mechanism of induction where it is not.

Read a field at a glance: outflow means divergence, swirl means curl



Read a field's story at a glance: outflow means divergence, swirl means curl

exercises.lol exercises.com.cn

Worked example 1 (the gradient predicts the drop, verified numerically). $V(x,y) = 3x^2y$ volts. $\nabla V = (6xy, 3x^2)$; at the point $(1,2)$: $\nabla V = (12,3)$, so $E = -(12,3)$ V/m. The check: move a small step $\delta = (0.01, 0)$ from $(1,2)$ – V changes by $\approx 6xy \cdot \delta x = 12 \times 0.01 = 0.12$ V, and $-E \cdot \delta = 12 \times 0.01 = 0.12$ V: the field times the step predicts the potential drop, which is what gradient means.

Worked example 2 (divergence and curl by hand, escalated). For $\mathbf{F} = (x^2,\ xy,\ 3x^2z)$: $\nabla \cdot \mathbf{F} = 2x + x + 3x^2$, which at $(1,1,1)$ equals $2 + 1 + 3 = 6$ – a net source. For the circulating field $\mathbf{G} = (-y,\ x,\ 0)$: the z-component of curl is $\frac{\partial G_y}{\partial x} - \frac{\partial G_x}{\partial y} = 1 - (-1) = 2$ – uniform rotation of strength 2 everywhere: the velocity field of a rigid body spinning at 1 rad/s, whose curl is twice its angular velocity. These two computations are the pointwise versions of EE07's Gauss and Stokes laws.

Common mistakes. Taking the gradient of a vector or the divergence of a scalar – gradient is for scalars, divergence for vectors. Sign slips in curl: the z-component is $\partial F_y/\partial x - \partial F_x/\partial y$; reversing them flips the rotation. A uniform field has zero divergence AND zero curl.

Summary: gradient is steepest-up (scalar→vector), divergence is net outflow (vector→scalar), curl is circulation (vector→vector); a step must reconcile $E \cdot \delta$ with δV.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

Digital Logic Design

University Electrical Engineering

What this course is, and how to use it

This is the third course in University Electrical Engineering: **digital logic design** – how bits are represented, combined, stored and timed inside every computer. It needs no prior electronics: the primitives are two voltage levels and a handful of gates.

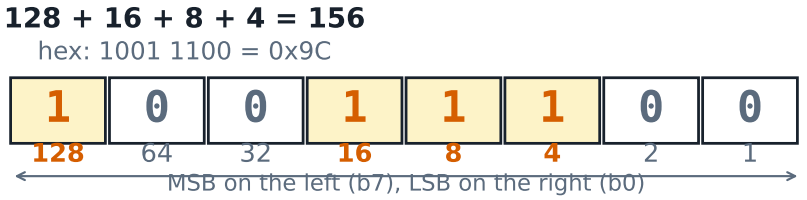
- Unit 3.1 builds the number systems: binary, hexadecimal, two's complement.
- Units 3.2–3.3 combine logic: Boolean algebra, Karnaugh maps, combinational blocks.
- Units 3.4–3.5 add time: latches, flip-flops, and clocked state machines.
- Unit 3.6 registers the patterns: counters, shift registers, memory.
- Unit 3.7 describes them in HDL and verifies with testbenches.

Each unit opens with a story, shows a picture before the formal idea, defines terms precisely, works two escalating examples whose arithmetic is machine-verified, names the classic mistakes, and closes with a three-line summary. The checking habit of this course: every Boolean answer is verified by **exhaustive evaluation** – a 3-input function has only 8 rows, so check all of them, either by hand on a truth table or by a loop in the sandbox.

Number representations and codes

A thermostat displays 22.5 °C, but the microcontroller behind it holds only bits. How does 22.5 become a bit pattern a machine can add, compare and store? Unit 3.1 is that translation, in both directions.

Computers store numbers in **binary** (base 2). Conversion is systematic: divide by 2 for integers (collect remainders), multiply by 2 for fractions. **Hexadecimal** (base 16) is binary's shorthand – each **hex** digit is exactly 4 bits: $0xB = 1011$.



Weights on the bits: read the pattern by adding the weights under each 1

exercises.lol exercises.com.cn

Deriving two's complement. We want an encoding where subtraction reuses the adder: $a-b=a+(b$ negated$)$ must hold modulo 2^N. The additive inverse of b modulo 2^N is $2^N-b=(2^N-1-b)+1$ – and 2^N-1-b is exactly "every bit of b inverted" (each 1-weight flips to 0 and back). So negate = invert + 1, and the derivation *is* the proof that addition works unchanged.

Two's complement represents signed integers in N bits: invert every bit and add 1 to negate. In 8 bits, -5 is $\overline{00000101}$ plus 1, which is 11111011. The **range** runs from -2^{N-1} up to $+2^{N-1}-1$ – that is -128 to $+127$ in 8 bits. **Overflow** occurs when the result leaves the range: adding two positives and reading a negative (or vice versa) is the sign-based detection rule – carry *out of* the MSB is not overflow; carry out differing from carry *into* the MSB is.

> **Worked example 1 (convert and check).** Convert 156 to binary: $156=128+16+8+4=10011100$. Check by re-adding the weights: $128+16+8+4=156$ ✓. The same number in hex: group the bits in fours from the right – 1001 1100 $=0x9C$ ✓. And the fraction 0.625: multiply by 2 → 1.25 (1), 0.25×2 → 0.5 (0), 0.5×2 → 1.0 (1): $0.625=0.101_2$ ✓ $(0.5+0.125)$.

> **Worked example 2 (negative numbers and overflow, escalated).** Encode -100 in 8 bits: 100 = 01100100; invert 10011011; add 1 → 10011100. Check by decoding: as unsigned that is 156, and $156-256=-100$ ✓. Now the overflow case: $100+50=150>127$, so 8-bit signed cannot hold it – the bits compute 10010110, which decodes to -106. The sign rule catches it: positive + positive read a negative ✓ (and the carry-into-MSB was 1 while carry-out was 0). The machine did not fail; the width did.

Common mistakes. Reading two's complement as unsigned (156 and -100 share bits – the interpreter decides). Forgetting the asymmetric range: -128 exists, $+128$ does not. Treating carry-out as overflow – only the carry mismatch counts.

Summary: binary↔decimal by weights and doubling; hex groups fours; two's complement = invert + 1, range $-2^{N-1}\ldots2^{N-1}-1$; overflow is detected by sign mismatch, not carry-out.

Boolean algebra and Karnaugh maps

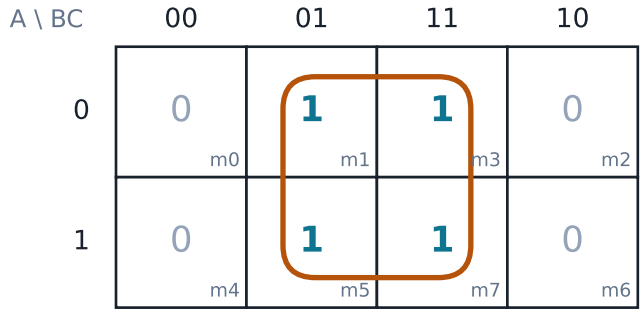
A hallway light must be switchable from either end: two switches, one bulb – the **XOR** the electrician wires is a Boolean function. **Boolean algebra** manipulates expressions with AND $(\cdot)$, OR $(+)$, NOT $(')$. The laws mirror ordinary algebra plus **absorption**: $A+A\cdot B=A$ and $A\cdot(A+B)=A$. **De Morgan's theorem** is the workhorse:

$$(A\cdot B)'=A'+B' \qquad (A+B)'=A'\cdot B'$$

– break the bar, change the sign.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

The **Karnaugh map** arranges a truth table so adjacent cells differ in one variable (Gray-code order); groups of 1, 2, 4, 8 cells (powers of two, rectangles) read off as product terms.



$F=\Sigma m(1,3,5,7)$: the group of four keeps only C, so $F=C$

> **Worked example 1 (the odd function).** The 3-variable function $F=\sum m(1,3,5,7)$ (the odd rows) maps to a column pair – every cell where $C=1$ – so $F=C$ directly: all four minterms share only C. The check: evaluate C against all 8 rows of the original; it matches exactly (machine-verified). **Don't-cares** (impossible input combinations) may join any group – use them to enlarge groups, never to create new ones.

> **Worked example 2 (the 4-variable majority, machine-verified, escalated).** The majority function $M(A,B,C,D)$ is 1 when three or more inputs are 1: its minterms are $\sum m(7,11,13,14,15)$. Grouping on the 4-variable map: four overlapping pairs, each sharing m_{15}, give $ABC+ABD+ACD+BCD$, which factors to $M=A\cdot B\cdot(C+D)+C\cdot D\cdot(A+B)$. The machine check: evaluate both the definition (count ≥ 3) and the expression on all 16 rows – they agree everywhere ✓. Notice the shape of the answer: the map gives four 3-input products instead of a flat 5-minterm sum, and factoring them into two products saves further gates.

Common mistakes. Grouping diagonals (cells must be rectangles in the Gray-code wrap-around, so edges touch). Writing a group term from variables that *change* inside the group – only the constant variables survive. Trusting an unverified simplification: the 16-row check is free.

Summary: Boolean laws plus De Morgan transform expressions; K-maps group power-of-two rectangles in Gray order; groups keep only constant variables; don't-cares enlarge but never create; verify every simplification exhaustively.

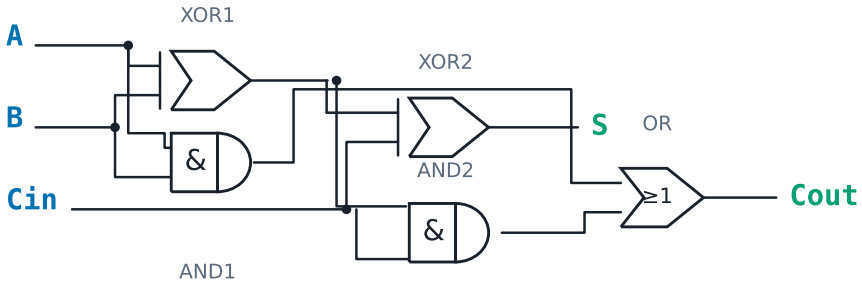
exercises.lol exercises.com.cn

Combinational building blocks

A vending machine serves one of four drinks by two button presses: that selector, the decoder behind the display, the adder inside the change logic – combinational blocks are the reusable middle layer between gates and systems. Gates compose into standard **combinational logic** blocks whose behaviour is a pure function of the present inputs:

- **Multiplexer** (MUX): 2^n-to-1 data selector; the n select bits choose one input. $Y = \sum(D_i \cdot minterm_i(selects))$.
- **Decoder** : n inputs to 2^n outputs, exactly one high – the minterm generator.
- **Adder** : a **half adder** sums two bits ($S = A \oplus B$, carry $C = A \cdot B$); a **full adder** adds three (two operands plus carry-in): $S = A \oplus B \oplus C_{in}$, $C_{out} = A \cdot B + C_{in} \cdot (A \oplus B)$.

The full adder: five gates, three inputs, two outputs



The full adder: five gates, one of the most-reused circuits in computing

Worked example 1 (the MUX and the adder, checked). A 4-to-1 MUX with select $S_1S_0 = 10$ outputs D_2 – the decimal weight of the select code names the data input. The full-adder check at $A = 1, B = 1, C_{in} = 1$: $S = 1 \oplus 1 \oplus 1 = 1$, $C_{out} = 1 + 1 \cdot 0 = 1$ – three ones in, binary 11 out (sum 1, carry 1): $3 = 2 + 1$ ✓.

Worked example 2 (a 2-bit ripple adder, machine-verified, escalated). Chain two full adders: the first adds the low bits, its carry feeds the second. Adding $11_2 + 01_2$ $(3 + 1)$: low bits $1 + 1$ give $S_0 = 0$, carry 1; high bits $1 + 0 + 1$ give $S_1 = 0$, carry 1 – output $100_2 = 4$ ✓ (machine checked through all four input pairs per stage). The pattern scales exactly: an n-bit adder is n full adders and nothing else, and the carry *ripples* through it – which is also its speed limit, fixed later by carry-lookahead.

Hazards are momentary output glitches when input changes take unequal paths; they matter because a glitch can be caught as a clock edge. The remedy is redundant grouping (keeping the hazard cover) – detect them by enumeration when the specification forbids glitches.

Common mistakes. Reading a MUX backwards (the select chooses *data*, not the output's value). Forgetting the carry chain's direction in ripple adders. Treating a hazard as a logic error – the final value is right; the *transient* is the bug.

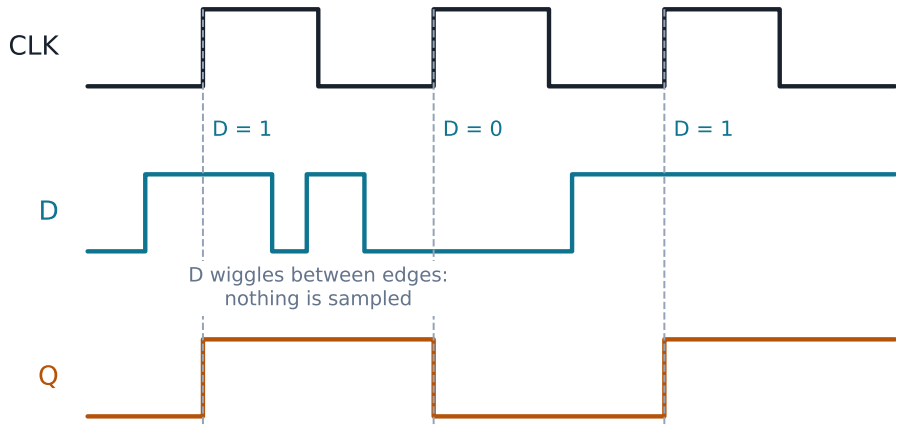
Summary: MUX selects by weighted code, decoder generates minterms, full adder = five gates summing three bits; ripple chains scale by repetition; hazards are transient, and redundant cover removes them.

Latches and flip-flops

A doorbell remembers that it was pressed after the finger leaves – the simplest memory is feedback. **Sequential logic** has state – memory. A **latch** is level-sensitive: the SR latch (NOR cross-coupled) remembers which input pulsed; the D latch follows D while ENABLE is high, freezes when low. **Flip-flops** are edge-triggered: the D flip-flop samples D **at the clock edge** and holds it until the next edge – Q after the edge equals D before it.

$$Q_{next} = D \quad \text{(D flip-flop, at each rising edge)}$$

The J-K flip-flop removes the SR forbidden state: $J = K = 1$ **toggles** (Q' next). The **excitation table** runs this backwards – given a desired transition $Q \to Q^+$, it lists the inputs required (D: $D = Q^+$; J-K: hold 0,0, set 1,0, reset 0,1, toggle 1,1).



Q copies D only at the rising clock edges

exercises.lol exercises.com.cn

Worked example 1 (read the timing diagram). A D flip-flop sees $D = 1$ at the first rising edge, $D = 0$ at the second, $D = 1$ at the third: Q reads 1, 0, 1. Between edges, D may wiggle freely – nothing is sampled. Timing diagrams make this visible: Q changes only at edge instants.

Worked example 2 (the J-K sequence, machine-traced, escalated). Starting from Q = 0, apply J, K = (1,0), (1,1), (0,1), (0,0). Step by step: (1,0) sets → 1; (1,1) toggles → 0; (0,1) resets → 0; (0,0) holds → 0. Q reads 1, 0, 0, 0 (machine-traced ✓). The design use: excitation tables run this backwards – to build a counter that goes 0 → 1 → 0 → 0 → 0, read each transition off the J-K table and the required inputs fall out (set, toggle, reset, hold). Flip-flop choice is a lookup, not a puzzle.

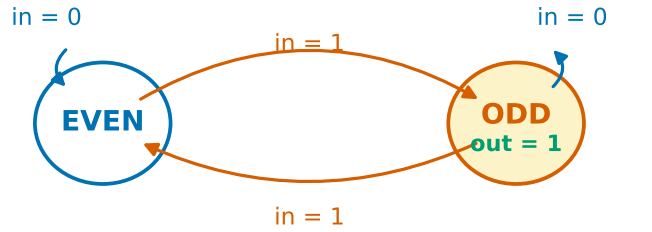
Common mistakes. Sampling on the wrong edge (falling vs rising – the specification says which). Confusing latch and flip-flop: a latch is transparent for a whole level, a flip-flop only at an instant. Expecting Q to change the moment D does – Q waits for the edge.

Summary: latches are level-transparent, flip-flops edge-triggered; Q after the edge = D before it; J-K removes the forbidden state with toggle; excitation tables invert behaviour into required inputs.

Finite state machines

A vending machine rejects a coin sequence it accepted yesterday – not because the coins changed, but because the *sequence* differs. Sequences need a machine with memory of where it is: a **finite state machine** (FSM) is states + transitions: a **state diagram** (bubbles and labelled arrows), realised as next-state logic + state registers + output logic. **Moore** machines compute outputs from state only (outputs change after the clock); **Mealy** machines compute from state and inputs (outputs can change mid-cycle, sooner but glitch-prone).

The parity detector: two states, one flip-flop



next-state logic: $Q^+ = Q \oplus in$ — the machine is a parity bit

Two states, one output: the parity detector's whole design

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Worked example 1 (the parity detector, traced). Raise an output when the number of 1s seen is odd. Two states (EVEN, ODD); each input bit moves between them; output = (state == ODD). Sequence 1, 0, 1, 1: EVEN → ODD → ODD → EVEN → ODD, so the output after the fourth bit is 1 (three 1s seen – odd ✓ machine-traced). Design procedure: state diagram → state table → (binary) state encoding → excitation table for each flip-flop → minimise next-state logic. With one state bit (0 = EVEN, 1 = ODD) the next-state logic is a single XOR: $Q^+ = Q \oplus in$ – the machine *is* a parity bit.

Worked example 2 (the 1011 detector, escalated). Detect the serial sequence 1011 (overlapping). States track the longest matched prefix: S0 (start), S1 (seen 1), S10 (seen 10), S101 (seen 101). Each transition goes to the longest tail of the input so far that is still a prefix of 1011:

State	input 0	input 1
S0	S0	S1
S1	S10	S1
S10	S0	S101
S101	S10	S1, *detect*

From S101 a 1 completes 1011: the output says *detected* and the machine goes to S1, because the trailing "1" already starts the next match (the overlap). A 0 from S101 leaves "1010", whose tail "10" is a prefix, so it goes to S10, not back to S0. Four states need two flip-flops ($2^2 = 4$ ✓). Trace the stream 1,0,1,1,0,1,1: S0→S1→S10→S101→**detect** (S1), then 0→S10, 1→S101, 1→**detect** again. The stream holds two overlapping copies, at bits 1–4 and 4–7, and the machine finds both (machine-traced ✓). A detector that restarts at S0 after a match would miss the second. **State minimisation** merges equivalent states (same outputs, same next states) – an FSM with redundant states works but wastes flip-flops and logic.

Common mistakes. Forgetting the overlap reset (a matched suffix is the next match's prefix). Encoding states before checking the state table for equivalent states. Choosing Moore when the spec needs same-cycle response (Mealy), or Mealy when outputs feed asynchronous logic (glitches).

Summary: states + transitions + outputs; Moore from state, Mealy from state and input; diagram → table → encoding → excitation → minimise; overlap detection and state merging are where marks are won and lost.

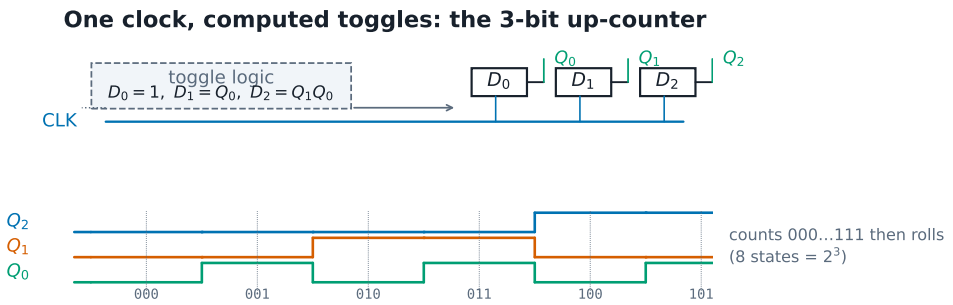
Registers, counters and memory

A scoreboard shows 7, then 8 – one number changed while the rest stood still: that selective update is a register, and counting is a register that updates itself. A **register** is a bank of flip-flops sharing a clock: load n bits per edge. A **shift register** passes bits along: serial-in serial-out delays a bit stream; serial-in parallel-

exercises.lol exercises.com.cn

out converts serial to parallel.

A **ripple counter** chains T-flip-flops (each output clocks the next); a **synchronous counter** drives all clocks from one edge and computes each toggle with logic – no cumulative delay.



One clock, computed toggles: the synchronous counter's register and trace

Worked example 1 (the 3-bit up-counter, traced). A 3-bit synchronous up-counter toggles bit 0 every clock, bit 1 when $Q_0 = 1$, bit 2 when $Q_1 \cdot Q_0 = 11$: after RESET, the counts run 000, 001, 010, 011 … 111, then roll to 000 (8 states = 2^3 ✓ – the count range check; machine-traced). The toggle conditions are exactly the AND of all lower bits – a binary counter carries when every lower bit is 1, which is what "add 1" means.

Worked example 2 (a mod-6 counter, escalated). A mod-6 counter counts 000–101 and returns to 000 – six states, not eight. For a synchronous implementation, choose next state 000 when the present word is 101, 110 or 111; otherwise add 1. Trace: 000, 001, 010, 011, 100, 101, then 000. Thus 101→000 happens at the next clock edge; either unused code 110 or 111 also recovers to 000 at the next edge. The condition $Q_2 \cdot Q_1$ detects only 110 and 111. It is 0 at 101, so using that condition alone to choose next state 000 would still allow a stored 110 after 101. Asynchronous clear is a different implementation: it can briefly reach 110 before clearing, so it must not be described as this synchronous jump. **Memory** is an organised array: k address bits select one of 2^k words of m bits – an 8K×8 memory has 13 address lines and 8 data lines ($8K = 2^{13} = 8192$ words ✓).

Common mistakes. Ripple counters *look* fine at low speed and fail at spec – the cumulative delay is per stage. Forgetting to handle the unused states in mod-N counters (they must recover to a legal state). Confusing shift direction conventions (MSB-first vs LSB-first serial).

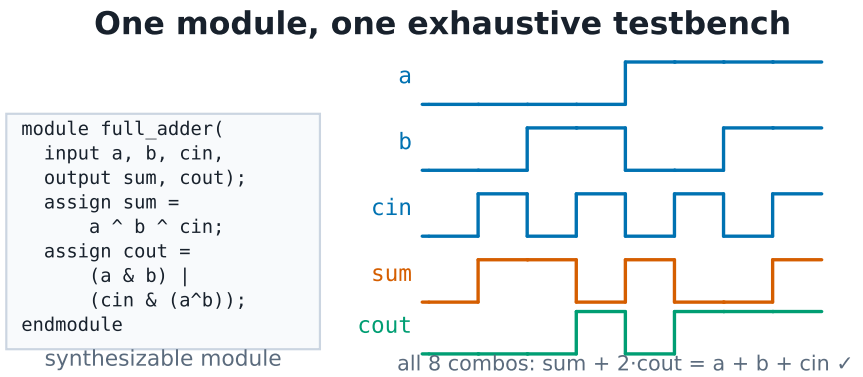
Summary: registers load per shared edge; shift registers convert between serial and parallel; synchronous counters compute toggles with logic; mod-N needs unused-state handling; memory is 2^k words of m bits, k address lines.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

HDL and verification basics

Wiring 64 gates by hand invites 64 mistakes – and the industry's answer is not more careful drawing but description plus verification. A **hardware description language** (Verilog/VHDL) describes structure and behaviour as **synthesizable** code – what can become gates – separated from the **testbench**, which simulates but never becomes hardware.

```
module full_adder(input a, b, cin, output sum, cout);
  assign sum  = a ^ b ^ cin;
  assign cout = (a & b) | (cin & (a ^ b));
endmodule
```



One module, one exhaustive testbench: description and proof together

Worked example 1 (the exhaustive testbench). The testbench drives all 8 input combinations and compares against the arithmetic truth ($a + b + cin = 2 \cdot cout + sum$) – machine-verified: all 8 pass ✓. The property is the point: the reference model is *arithmetic*, not another gate description, so the check is independent of the implementation.

Worked example 2 (a shift register in HDL, escalated). Describe a 4-bit serial-in shift register:

```
module shift4(input clk, sin, output reg [3:0] q);
  always @(posedge clk) q <= {q[2:0], sin};
endmodule
```

The concatenation `{q[2:0], sin}` *is* the shift – in goes at the right, out falls off the left. The testbench drives sin = 1,0,1,1 over four clocks and the trace reads q = 0001, 0010, 0101, 1011 ✓ (machine-simulated) – unit 3.6's shift register, now a four-line description.

exercises.lol exercises.com.cn

Common mistakes. Testbenches that restate the implementation (a copy proves nothing). Non-synthesizable constructs in hardware modules. Missing resets in `always` blocks – unit 3.6's trap again.

Summary: synthesizable module vs non-synthesizable testbench, always separate; the reference model must be independent (arithmetic or trace); exhaustive checks for small input spaces; ship modules with their proof.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

Circuit Analysis II (AC)

University Electrical Engineering

What this course is, and how to use it

This is the fourth course in University Electrical Engineering: **AC circuit analysis** – circuits driven by sinusoids, which is how almost all electrical energy is generated and moved. Wave 1 solved DC; the tools of EE02 (complex numbers, differential equations, Laplace) now pay for themselves.

- Unit 4.1 makes sinusoids arithmetic: the **phasor** and the **impedance**.
- Units 4.2–4.4 apply it: AC power, three-phase, second-order response and resonance.
- Units 4.5–4.7 widen the view: filters and Bode plots, s-domain circuit analysis, and two-port networks.

Each unit opens with a story, shows a picture before the formal idea, defines terms precisely, works two escalating examples whose numbers are machine-verified, names the classic mistakes, and closes with a three-line summary. The checking habit: every phasor answer verifies in a second representation – polar against rectangular, or the time domain against the complex arithmetic. Complex arithmetic is only bookkeeping; the checks keep it honest.

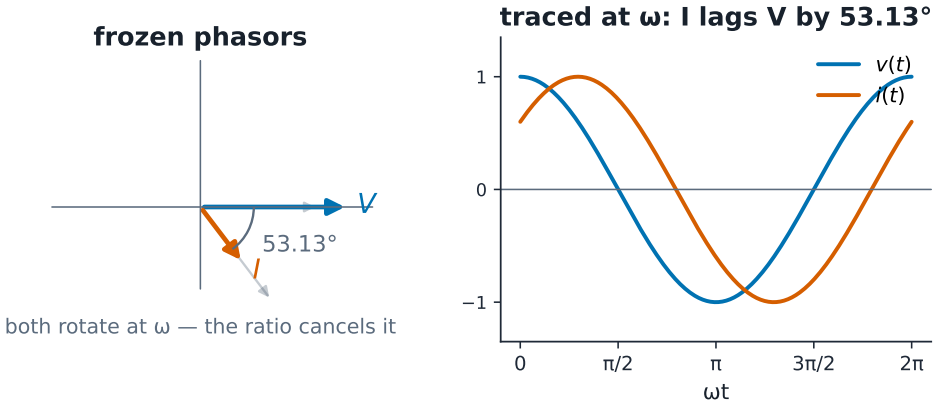
Sinusoidal steady state and phasors

The mains voltage crosses zero a hundred times a second, yet a resistor there dissipates steady power – because amplitude and phase settle into a repeating pattern. A sinusoid $v(t) = V_m \cos(\omega t + \varphi)$ becomes a **phasor** $V = V_m \angle \varphi$ – the amplitude and phase, with the frequency understood. Differentiating multiplies by $j\omega$; so at a single frequency every element gets an **impedance** :

$$Z_R = R \qquad Z_L = j\omega L \qquad Z_C = \frac{1}{j\omega C}$$

All the DC methods – dividers, node/mesh, Thévenin – carry over unchanged with complex numbers. The inductor's impedance grows with frequency (+90°); the capacitor's shrinks (−90°); the resistor is flat (0°).

exercises.lol exercises.com.cn



The rotating picture behind the frozen phasors: ratio questions cancel the rotation, so the snapshot is enough

Worked example 1 (both representations agree). $R = 30\ \Omega$ in series with L such that $\omega L = 40\ \Omega$, driven by $V = 100\angle 0°$ V at that frequency. $Z = 30 + j40 = 50\angle 53.13°\ \Omega$; $I = V/Z = (100/50)\angle(0 - 53.13°) = 2\angle -53.13°$ A. Check in rectangular: $I = 100(30 - j40)/2500 = 1.2 - j1.6$ A; $|I| = \sqrt{1.44 + 2.56} = 2$ ✓ and $\text{atan2}(-1.6, 1.2) = -53.13°$ ✓. The time-domain meaning: $i(t) = 2\cos(\omega t - 53.13°)$ A – the current lags because the inductor opposes change.

Worked example 2 (parallel combination, escalated). The same $R = 30\ \Omega$ now in **parallel** with the $\omega L = 40\ \Omega$ inductor: $Z = \frac{30 \times j40}{30 + j40} = \frac{30 \times j40 \times (30 - j40)}{2500} = \frac{48\,000 + j36\,000}{2500} = 19.2 + j14.4 = 24\angle 36.87°\ \Omega$ (verified: $|Z| = \sqrt{19.2^2 + 14.4^2} = 24$ ✓, and $\cos 36.87° = 0.8$ ✓). Note the parallel impedance is *smaller* than either branch magnitude ($24 < 30 < 40$) and its angle (36.87°) lies *below* the inductor's 90°, pulled toward the resistor – every parallel pair trades magnitude and angle in this direction.

Common mistakes. Adding magnitudes of parallel impedances (only complex addition is legal). Mixing degree-mode calculator settings into radian formulas. Forgetting that the phasor carries *amplitude* (or RMS – pick one convention per problem) consistently for V and I.

Summary: one frequency → phasors; Z_R flat, Z_L up 90°, Z_C down 90°; every DC method carries over; verify each answer in the other representation.

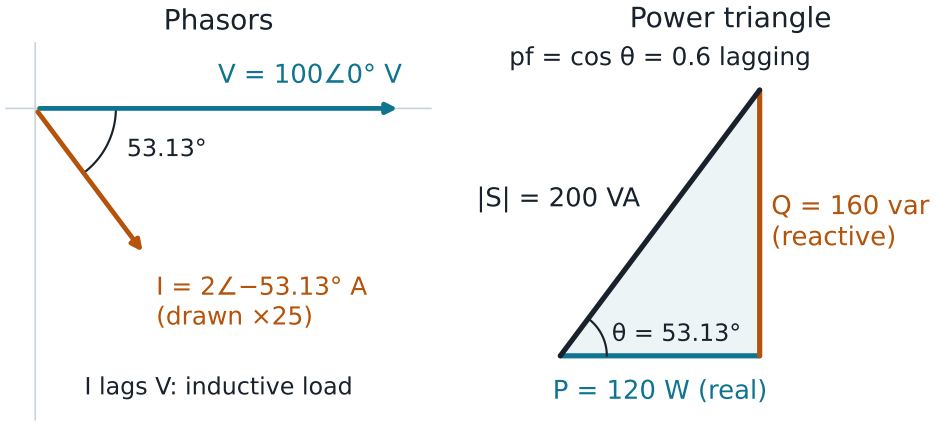
exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

AC power and three-phase

A 1 kW heater and a 1 kVA capacitor both draw "1 k-something" from the line, but only the heater's share bills as energy. With phasor V and current I, **complex power** $S = V \cdot I^*$:

$$P = VI\cos\theta \quad \text{(real, W)}, \qquad Q = VI\sin\theta \quad \text{(reactive, var)}$$

with θ the angle between V and I. The **power factor** $\cos\theta$ is the fraction of apparent power $|S| = VI$ that does real work.



The RL load: current lags by 53.13°, so $P = 120$ W, $Q = 160$ var and pf = 0.6 lagging

Worked example 1 (the RL load, checked two ways). The load above: $P = 100 \times 2 \times \cos 53.13° = 120$ W, $Q = 100 \times 2 \times \sin 53.13° = 160$ var, $|S| = 200$ VA, pf = 0.6 lagging. Check: $P = |I|^2 R = 4 \times 30 = 120$ ✓ and $Q = |I|^2 X = 4 \times 40 = 160$ ✓ – the loss-formula check. **Power-factor correction** adds parallel capacitance to supply the reactive power locally: to correct 160 var at 100 V (50 Hz) needs $C = Q/(\omega V^2) = 160/(2\pi \cdot 50 \cdot 10^4) = 50.9\ \mu F$.

exercises.lol exercises.com.cn

Worked example 2 (a three-phase motor, escalated). A balanced star-connected motor on a 415 V (line), 50 Hz supply has phase impedance $10+j10\ \Omega$ per phase. Phase voltage $= 415/\sqrt{3} = 239.6$ V; phase current $= 239.6/\sqrt{10^2+10^2} = 239.6/14.14 = 16.94$ A (equal to line current in star); phase angle 45° lagging. Total power $P = 3 \times 239.6 \times 16.94 \times \cos 45° = 8611$ W – and the line formula $\sqrt{3} \times 415 \times 16.94 \times 0.7071 = 8611$ W ✓ the two routes agree. The per-phase route also gives the missing pieces: $Q = 3 \times 239.6 \times 16.94 \times \sin 45° = 8611$ var, pf 0.707 lagging, and (for a delta-connected motor of the same impedance) the phase current would be $415/14.14 = 29.3$ A with line current $\sqrt{3} \times 29.3 = 50.8$ A – the star/delta factor is exactly the $\sqrt{3}$ bookkeeping.

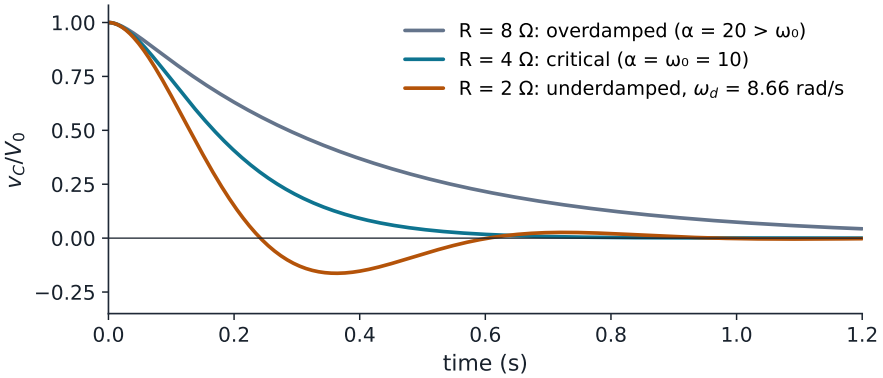
Common mistakes. Using line voltage in the per-phase formula (divide by $\sqrt{3}$ first). Forgetting the motor's pf applies to *both* formulas. Correcting power factor at the wrong voltage – the capacitor sees the voltage at its own connection point.

Summary: $S = VI^*$ splits into P (billed) and Q (borrowed); loss formulas $|I|^2R$, $|I|^2X$ are the check; capacitive correction supplies Q locally; three-phase per-phase analysis with $V_{line} = \sqrt{3}V_{phase}$ (star) is the workhorse.

Second-order RLC circuits

Close a switch on a charged capacitor in series with an inductor and the current cannot jump – nor can it arrive instantly – so the response bends through at least one derivative more than RC ever needed. A series RLC has the differential equation $L\,d^2i/dt^2 + R\,di/dt + i/C = 0$, with **natural frequency** $\omega_0 = 1/\sqrt{LC}$ and **damping** $\alpha = R/(2L)$. Three regimes:

- **Overdamped** ($\alpha > \omega_0$): two real decays, no oscillation.
- **Critically damped** ($\alpha = \omega_0$): fastest non-oscillating return.
- **Underdamped** ($\alpha < \omega_0$): decaying oscillation at $\omega_d = \sqrt{\omega_0^2 - \alpha^2}$.



One circuit, three resistors: the same $\omega_0 = 10$ rad/s with $\alpha = 20$, 10 and 5

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Worked example 1 (regime arithmetic). $R = 4\ \Omega$, $L = 0.2$ H, $C = 0.05$ F: $\omega_0 = 1/\sqrt{0.2 \times 0.05} = 10$ rad/s; $\alpha = 4/(2 \times 0.2) = 10$ – critically damped ($\alpha = \omega_0$). Change R to $2\ \Omega$: $\alpha = 5 < 10$, underdamped with $\omega_d = \sqrt{100 - 25} = 8.66$ rad/s. The regime test is one comparison; the **Q factor** $Q = \omega_0/(2\alpha)$ is 0.5 at critical damping and 1 at $R = 2\ \Omega$ – lightly damped (high Q) rings longer.

Worked example 2 (the overdamped exponents, machine-verified, escalated). With $R = 8\ \Omega$ (same L, C): $\alpha = 20 > 10$ – overdamped, and the two decay exponents are $s = -\alpha \pm \sqrt{\alpha^2 - \omega_0^2} = -20 \pm \sqrt{300}$, i.e. -2.68 and $-37.32\ \text{s}^{-1}$ (machine-verified). The capacitor voltage released from 1 V is $v_C(t) = Ae^{-2.68t} + Be^{-37.32t}$ with $A + B = 1$ and $dv_C/dt(0) = 0$ (no instantaneous current), giving $A = 37.32/(37.32 - 2.68) = 1.077$, $B = -0.077$: a fast initial drop onto the slow decay. The reading: the *fast* root only sets the first moments; the *slow* root (-2.68, the one nearer the origin) owns the tail – exactly the dominant-pole rule EE02 met in Laplace.

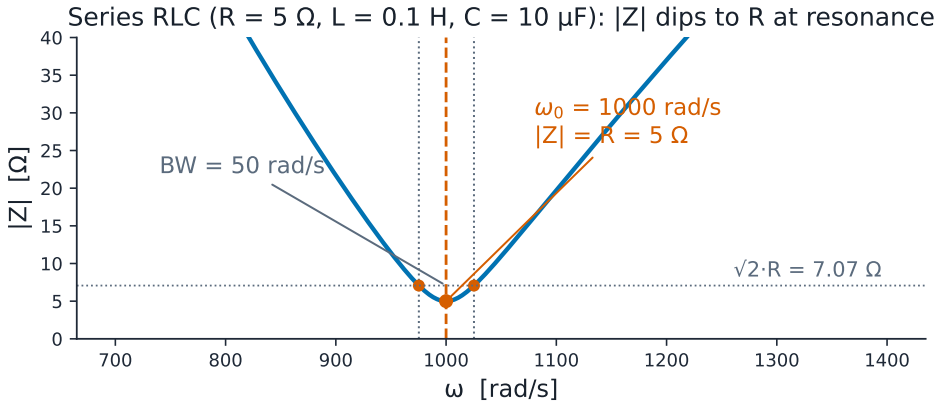
Common mistakes. Comparing R with $2\sqrt{L/C}$ instead of α with ω_0 – the same test, but write it as stated. Using ω_0 where ω_d belongs (the ringing you *see* is the damped frequency). Forgetting the second initial condition: two energy stores give two constants.

Summary: $\omega_0 = 1/\sqrt{LC}$, $\alpha = R/2L$, three regimes by one comparison; ω_d is the visible ringing; the slow root owns the tail; two stores → two constants → two endpoint checks.

Resonance

A radio's tuning circuit picks one station out of a crowded band by making one frequency special: at **resonance** the imaginary parts cancel, $\omega_0 L = 1/(\omega_0 C)$, i.e. $\omega_0 = 1/\sqrt{LC}$. There $Z = R$ (purely resistive, minimum), current is maximum, and the **bandwidth** $BW = R/L = \omega_0/Q$. The halves of the impedance cancel even though each alone can be large – V_L and V_C can each exceed the source voltage by the factor **Q**.

exercises.lol exercises.com.cn



|Z| dips to R at ω_0; the half-power points sit where |Z| = √2 · R

Worked example 1 (voltage magnification, checked). $R = 5\ \Omega$, $L = 0.1$ H, $C = 10\ \mu F$: $\omega_0 = 1/\sqrt{0.1 \times 10^{-5}} = 1000$ rad/s; $Q = \omega_0 L/R = 1000 \times 0.1/5 = 20$. At a source of 10 V, the current at resonance is $10/5 = 2$ A, and $V_L = \omega_0 L I = 200$ V – twenty times the source. Check: $V_C = I/(\omega_0 C) = 2/0.01 = 200$ V ✓; both are 200 V, 180° apart, their sum zero.

Worked example 2 (the half-power band, escalated). The same circuit's bandwidth: $BW = \omega_0/Q = 1000/20 = 50$ rad/s, and the half-power frequencies compute to $\omega_1 = 975.3$, $\omega_2 = 1025.3$ rad/s – their difference is 50.0 ✓ (machine-verified). They sit almost evenly either side of 1000; exactly, $\omega_1\omega_2 = \omega_0^2$ (symmetric on a logarithmic axis). At the band edges $|Z| = \sqrt{2}R = 7.07\ \Omega$, the current is $1/\sqrt{2}$ of maximum, and each reactive element holds $Q/\sqrt{2} \approx 14$ times the source voltage – still amplification, just past its peak. This is the selectivity a tuner sells: at 900 rad/s (a neighbouring station two bandwidths away) the current is down to $10/\sqrt{R^2 + (\omega L - 1/\omega C)^2} = 10/|5 + j(90 - 111.1)| = 10/21.7 = 0.46$ A – under a quarter of the resonant 2 A.

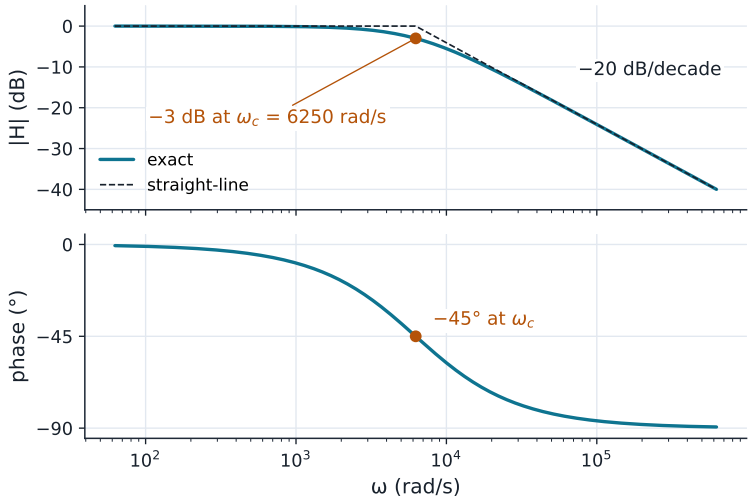
Common mistakes. Reporting Q-amplification without checking the other element (both must match). Using f_0 where ω_0 belongs (factor 2π). Believing resonance maximises *power* at any source impedance – it minimises |Z|, which maximises current only for a stiff source.

Summary: $\omega_0 = 1/\sqrt{LC}$, |Z| = R there, BW = ω_0/Q between the half-power points; each reactive voltage is Q · source; check the amplification on both elements.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Filters and Bode plots

A bass boost is a filter: one band up, the rest untouched. A **filter** passes a band and attenuates the rest; its transfer function $H(j\omega)$ has magnitude and phase. The **Bode plot** plots $20\log_{10}|H|$ in dB against log frequency: each pole contributes −20 dB/decade above its **corner frequency** $\omega_c = 1/(RC)$; each zero +20 dB/decade.



RC low-pass with its corner at 6250 rad/s: exact curves against the straight-line asymptotes

Worked example 1 (the low-pass, endpoints checked). A low-pass RC with $R = 1.6$ kΩ, $C = 0.1\ \mu F$: $\omega_c = 1/(1600 \times 10^{-7}) = 6250$ rad/s (995 Hz). At ω_c the gain is $1/\sqrt{2} = -3$ dB and the phase −45°; at $10\omega_c$ the straight-line approximation reads −20 dB (exact −20.04). Check the endpoints: at low frequency $H = 1$ (0 dB), at $100\omega_c \approx -40$ dB – each decade past the corner loses 20 dB, which is the **decibel** arithmetic $20\log_{10} 10 = 20$.

Worked example 2 (the high-pass mirror, machine-verified, escalated). Swapping output to the resistor makes a high-pass with the same $\omega_c = 6250$ rad/s: $H(j\omega) = (j\omega/\omega_c)/(1 + j\omega/\omega_c)$. At $0.1\omega_c$: −20.04 dB at +84.3°; at ω_c: −3.01 dB at +45°; at $10\omega_c$: −0.04 dB at +5.7° (all machine-verified) – the exact mirror of the low-pass, with phase *leading* and the stop-band below instead of above. The design reading: a steeper 40 dB/decade roll-off needs two cascaded sections, and (with the loading between them accounted for) the cascade multiplies gains, so the dB add – the same chaining logic as the ABCD matrices of unit 4.7.

Common mistakes. Reading −3 dB as "half power lost in the filter" at the corner

exercises.lol exercises.com.cn

(it is half *power delivered to the load versus mid-band*). Mixing ω and f on the same axis. Forgetting phase: a filter that is flat in magnitude can still wreck a pulse by its phase slope.

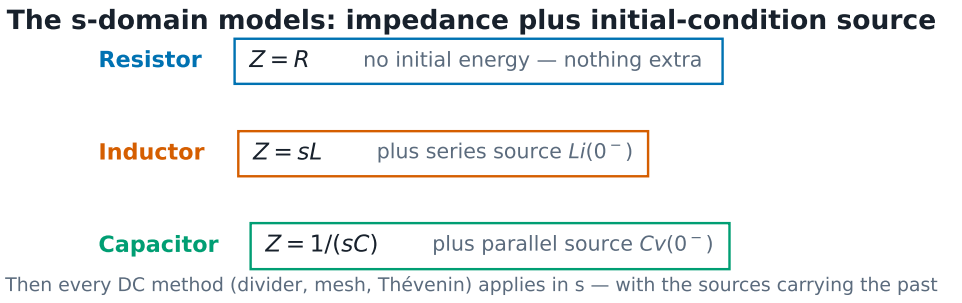
Summary: poles -20 dB/decade above their corner, zeros $+20$; -3 dB and $\pm 45°$ at the corner; the asymptotes are the design tool, the exact curve the check.

Laplace circuit analysis

EE02's Laplace machinery becomes a circuit method: each element gets an s-domain model with initial conditions as sources –

- Resistor: $Z = R$.
- Inductor: $Z = sL$ **plus a series voltage** $Li(0^-)$ (or parallel current source $i(0^-)/s$).
- Capacitor: $Z = 1/(sC)$ **plus a parallel current source** $Cv(0^-)$ (or series voltage $v(0^-)/s$).

Then every DC method applies in s. The response's **poles** carry the transient; the input's poles carry the steady state.



The s-domain models: impedance plus initial-condition source – one picture for every transient problem

> **Worked example 1 (the wave-1 RC, by algebra).** The wave-1 RC step with initial voltage: $R = 10\,\text{k}\Omega$, $C = 100\,\mu F$ ($\tau = 1$ s), source 9 V, $v_C(0) = 1$ V. s-domain KVL: $9/s = IR + v_C(0)/s + I/(sC) \to I = 8/(R(s + 1/(RC))) = 8/(10^4(s + 1))$; $V_C = v_C(0)/s + I/(sC) = 1/s + 8/(s(s+1)) = 9/s - 8/(s+1)$, so $v_C(t) = 9 - 8e^{-t}$ V – exactly the wave-1 answer, now derived by algebra. The check: $v_C(0) = 9 - 8 = 1$ ✓, $v_C(\infty) = 9$ ✓.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

> **Worked example 2 (a source-free RLC, poles matched, escalated).** The series RLC of unit 4.3 with $R = 2\ \Omega$, $L = 0.2$ H, $C = 0.05$ F, released with $i(0) = 1$ A, $v_C(0) = 0$. s-domain mesh (impedances plus the inductor's $Li(0)$ source): $(0.2s + 2 + 20/s)I = 0.2$, so $I(s) = s/(s^2 + 10s + 100)$ and the poles are $-5 \pm j8.66$ – exactly the $\alpha = 5$, $\omega_d = 8.66$ the time-domain regime arithmetic gave ✓. The two routes are the same physics in two notations; the s-domain route delivers the exponents without memorising a form, and partial fractions (as in EE02) finish the waveform.

Common mistakes. Dropping the initial-condition sources – the answer then claims the circuit starts at rest. Cancelling s blindly (a factor of s in numerator and denominator can hide an impulse). Forgetting the final-value theorem's condition (pole strictly in the left half-plane).

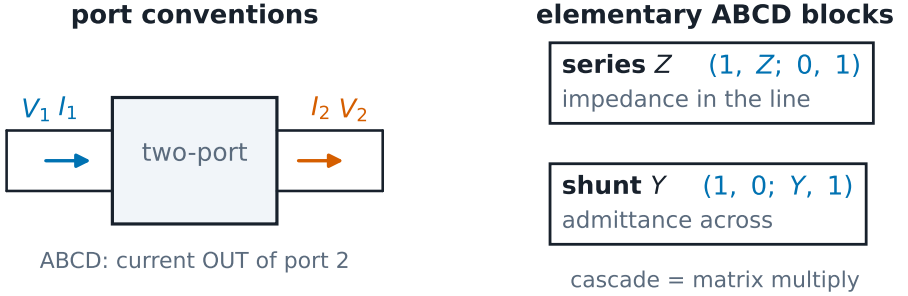
Summary: impedances become R, sL, $1/sC$ plus initial-condition sources; DC methods in s; the poles *are* α and ω_d; endpoint checks still free.

Two-port networks

An amplifier module, a matching network, a length of cable – each is a box with two wire pairs, and the designer only needs what the box does at its terminals. A **two-port** has input and output terminal pairs; the outside world sees only the port behaviour. Parameter sets relate the port variables:

$$\begin{pmatrix} V_1 \\ I_1 \end{pmatrix} = [ABCD] \begin{pmatrix} V_2 \\ -I_2 \end{pmatrix}, \qquad Z\text{:}\ \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = [z] \begin{pmatrix} I_1 \\ I_2 \end{pmatrix}$$

Conversion is algebra (e.g. from z-parameters: $A = z_{11}/z_{21}$, $B = (z_{11}z_{22} - z_{12}z_{21})/z_{21}$). **Cascade** connections multiply ABCD matrices – the reason transmission lines and filter chains use this set.



Port conventions and the two elementary ABCD blocks every chain composes

exercises.lol exercises.com.cn

Worked example 1 (the L-section cascade, determinant checked). The series impedance Z has $ABCD = (1, Z; 0, 1)$; the shunt admittance Y has $(1, 0; Y, 1)$. Cascading shunt-Y then series-Z: $(1, 0; Y, 1)(1, Z; 0, 1) = (1, Z; Y, 1 + YZ)$. Check with $Z = j10$, $Y = -j0.1$ (an L section at one frequency): $1 + YZ = 1 + (-j0.1)(j10) = 2$, $B = j10$ – and the determinant $AD - BC = 2 - j10 \cdot (-j0.1) = 1$ equals 1, as every ABCD matrix of a reciprocal network must.

Worked example 2 (into z-parameters, machine-verified, escalated). The same L-section's z-parameters from the conversion $z_{11} = A/C$, $z_{12} = \Delta/C$, $z_{21} = 1/C$, $z_{22} = D/C$ with $C = Y = -j0.1$: $z_{11} = z_{12} = z_{21} = 1/(-j0.1) = j10\ \Omega$, $z_{22} = 2/(-j0.1) = j20\ \Omega$ (machine-verified). The symmetry $z_{12} = z_{21}$ is the reciprocity condition – visible in the numbers, not assumed – and $z_{22} \neq z_{11}$ records the asymmetry of the L (shunt at the input side). Read back: with port 2 open ($I_2 = 0$), $V_1 = j10 I_1$ – the series arm carries no current, so the input sees only the shunt arm, $1/Y = 1/(-j0.1) = j10\ \Omega$.

Common mistakes. Sign of I_2: ABCD uses $-I_2$ (current *out* of port 2); z-parameters use it *into* the port – pick the convention and stay. Multiplying cascades in the wrong order (matrix product is not commutative). Expecting $z_{12} = z_{21}$ for a network with controlled sources inside – reciprocity requires linearity *without* dependent sources.

Summary: ports hide the interior; ABCD cascades by multiplication (reciprocal networks: $AD - BC = 1$); z-parameters suit series connections and expose reciprocity as $z_{12} = z_{21}$; convert only by algebra, verify by one determinant.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

Electronics I

University Electrical Engineering

What this course is, and how to use it

This is the fifth course in University Electrical Engineering: **electronics I** – the first course on devices. Wave 1 and EE04 analysed circuits made of ideal components; here the components themselves are the subject, and the core skill is **modelling** : choosing the simplest device model that answers the question at hand.

- Unit 5.1 introduces semiconductors and the diode's ladder of models.
- Unit 5.2 applies diodes: rectifiers, supplies and zener regulators.
- Units 5.3–5.4 bias the two transistor families (BJT, MOSFET).
- Unit 5.5 linearises about the bias point: small-signal amplifiers.
- Units 5.6–5.7 add feedback and the frequency-domain view.

The checking habit: every bias point is verified against the device's operating regions (the assumption that made the equation legal), every small-signal answer at a limiting case. Each unit ends with the common mistakes and a three-line summary.

Semiconductors and diode models

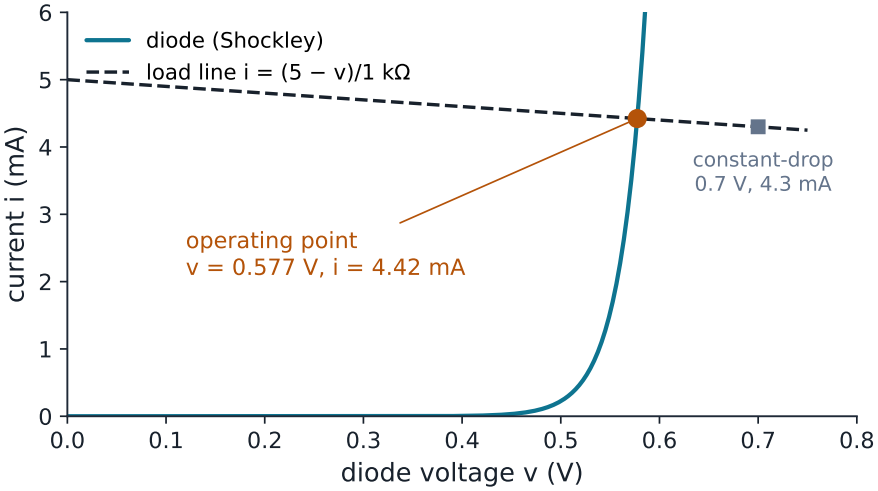
Pure silicon conducts poorly; **doping** adds carriers – donors make n-type (electrons), acceptors make p-type (holes). A **pn junction** PN conducts one way: forward bias (p side positive) lets current flow exponentially; reverse bias widens the depletion region and blocks.

The diode has a ladder of models – pick by the question:

- **Ideal** : 0 V when conducting, open when off.
- **Constant-drop** : 0.7 V when conducting (silicon).
- **Piecewise-linear** : 0.7 V + slope resistance $r_d = nV_T/I_D$.
- The **exponential** Shockley law $i = I_S(e^{v/nV_T} - 1)$ when the curve shape matters, with $V_T = kT/q \approx 26$ mV at room temperature.

Why the ladder works: differentiate the Shockley law at the operating point and the diode's dynamic resistance is $r_d = nV_T/I_D$ – a few ohms at milliamps. And because $i \propto e^{v/nV_T}$, ten times the current needs only $\ln 10 \times nV_T = 2.303 \times 26 = 59.9$ mV more voltage (n = 1): the curve is a near-vertical wall, so the *resistor* in the loop, not the diode, sets the current – which is exactly when the constant-drop model is accurate.

exercises.lol exercises.com.cn



The load line fixes the current; the steep diode curve barely moves the voltage

Worked example 1 (model selection, three ways). A 5 V source feeds a diode in series with 1 kΩ. Ideal: 5 mA. Constant-drop: $(5-0.7)/1000 = 4.3$ mA. Shockley truth ($I_S = 10^{-12}$ A, $nV_T = 26$ mV, bisection on $I_S(e^{v/26\text{m}} - 1) = (5 - v)/1000$): $v_D = 0.5775$ V, $i = 4.423$ mA. The constant-drop answer is within 2.8% – because the resistor dominates the loop. Check why: the constant-drop model is off by $0.7 - 0.5775 = 0.12$ V in the diode voltage, and the resistor turns that into only $0.12/1000 = 0.12$ mA of current error – 2.8% of 4.42 mA.

Worked example 2 (the dynamic resistance, escalated). At that operating point $r_d = nV_T/I_D = 26\,\text{mV}/4.42\,\text{mA} = 5.9\ \Omega$. Now let the source wiggle by ±0.1 V (small signal): the wiggle current is $0.1/(1000+5.9) = 99.4\ \mu$A – within 0.6% of the resistor-only 100 µA. The 5 V worth of DC sits across the pair almost unchanged; the 0.1 V of signal sees an ordinary 1 kΩ + 5.9 Ω loop. This is the whole small-signal idea of unit 5.5, met first at the diode: split the problem into a bias (where the nonlinear law sets the point) and a wiggle (where the linearised resistance takes over).

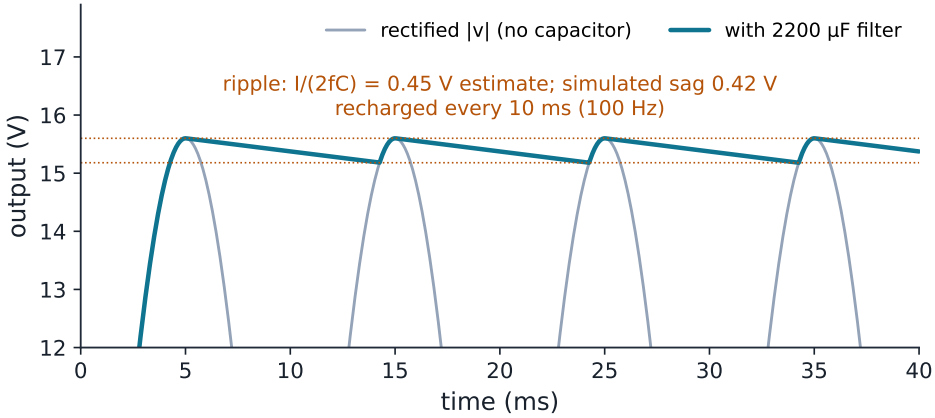
Common mistakes. Using the exponential model when the resistor dominates (needless bisection; the constant-drop model is the *right* tool there). Quoting "$r_d = 26$ mV" – that is V_T; the resistance is V_T/I_D, current in the denominator. Forgetting n = 2 at high current (59.9 mV/decade becomes 120 mV/decade).

Summary: a diode is a one-way valve with a near-vertical forward curve; the loop resistor sets the current; $r_d = nV_T/I_D$ is the linearised view; pick the model the question needs and check the answer against the next model up.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Rectifiers and dc supplies

The **half-wave rectifier** conducts one half-cycle; the **full-wave bridge** conducts both (four diodes, two conducting at a time, so the diode drops cost 1.4 V and each diode must survive a **peak inverse voltage** of one secondary peak). A **filter capacitor** holds the output between peaks: the **ripple** $\Delta V \approx I_{load}/(f \cdot C)$ (half-wave) or $I_{load}/(2f \cdot C)$ (full-wave) – the capacitor must supply the load for the whole gap between recharge peaks.



Full-wave bridge with 2200 µF at 100 mA: the capacitor bridges the gaps between peaks

Worked example 1 (the bridge supply, checked). A bridge rectifies a 12 V RMS, 50 Hz secondary; the load draws 100 mA and the capacitor is 2200 µF. Peak output: $12\sqrt{2} - 1.4 = 17.0 - 1.4 = 15.6$ V. Ripple (full wave, recharged every 10 ms): $\Delta V = 0.1/(100 \times 2200\mu) = 0.455$ V. So the output lives between 15.6 and 15.1 V. Checks: ripple ≪ peak (2.9% ✓ – the formula's premise); half-wave would double the ripple to 0.91 V for the same capacitor – the bridge buys a 2× for four diodes; PIV per diode ≈ 17.0 V (one secondary peak) – rate the diodes above that.

Worked example 2 (a zener regulator designed at both extremes, escalated). Regulate 12 V down to 5.1 V with a 5.1 V **zener**; the load varies 0–10 mA and the zener needs at least 2 mA to stay in breakdown. Worst case for the zener current is *full* load: $R \le (12 - 5.1)/(10\text{m} + 2\text{m}) = 575\ \Omega \to$ pick 560 Ω. Now check the *other* extreme – no load: $I_Z = (12 - 5.1)/560 = 12.3$ mA, so $P_Z = 5.1 \times 12.3\text{m} = 63$ mW and the resistor dissipates $(12 - 5.1)^2/560 = 85$ mW worst case. Full load leaves $I_Z = 12.3 - 10 = 2.3\ \text{mA} \ge 2$ mA ✓. Both extremes verified – that is a complete regulator design: one inequality from each end of the load range.

exercises.lol exercises.com.cn

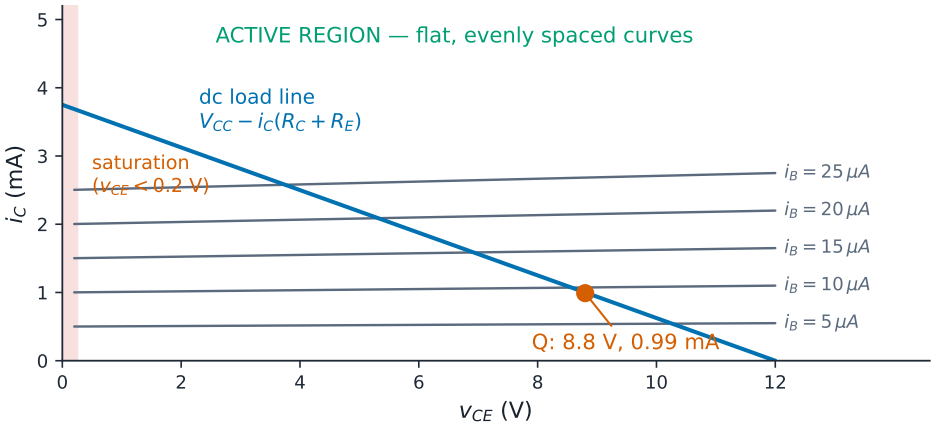
Common mistakes. Using f instead of 2f for the bridge ripple (the capacitor recharges every half-cycle). Forgetting the 1.4 V of two diode drops in the peak output. Sizing the zener resistor at the *typical* load only – the zener current must be checked at BOTH extremes, and its power at the no-load extreme.

Summary: bridge = both half-cycles, 1.4 V cost, PIV one peak; ripple $I/(2fC)$ set by the gap between recharges; the zener holds the voltage while the series resistor absorbs the surplus – designed at the two load extremes.

BJT biasing

The **bipolar junction transistor** has three regions – cut-off, **active** (the amplifier region: $I_C = \beta I_B$, $V_{BE} \approx 0.7$ V), and saturation ($V_{CE} < 0.2$ V, both junctions forward). Biasing means placing the quiescent point (the **bias point**, Q-point) firmly in the active region and making it *independent of* β – the parameter that varies 2:1 between devices of the same part number.

Divider bias on the output plane: the Q-point and its check



The 12 V dc load line ($RC + RE = 3.2$ kΩ) with the divider-bias Q-point at 0.99 mA, 8.8 V

Worked example 1 (voltage-divider bias, exact). $V_{CC} = 12$ V; $R_1 = 68$ kΩ (to V_{CC}), $R_2 = 12$ kΩ (to ground), $R_C = 2.2$ kΩ, $R_E = 1$ kΩ, $\beta = 100$. Thévenin the divider: $V_B = 12 \times 12/80 = 1.8$ V, $R_B = 68\text{k} \parallel 12\text{k} = 10.2$ kΩ. Quick estimate (ignore base current): $V_E = 1.8 - 0.7 = 1.1$ V $\rightarrow I_E \approx 1.1$ mA. Exact (base loop with $I_E = (\beta + 1)I_B$): $I_B = (1.8 - 0.7)/(10.2\text{k} + 101 \times 1\text{k}) = 9.9\ \mu$A $\rightarrow I_C = 0.989$ mA, $V_{CE} = 12 - 0.99\text{m} \times 2.2\text{k} - 0.99 = 8.8$ V – active ✓ ($V_{CE} \gg 0.2$ V). The quick estimate was 11% high; the culprit is base current loading the divider – exactly what the exact loop accounts for.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Worked example 2 (β doubled, the point of the circuit, escalated). Same circuit, $\beta = 200$: $I_B = 1.1/(10.2\text{k} + 201\text{k}) = 5.2\ \mu$A $\rightarrow I_C = 1.04$ mA, $V_{CE} = 8.7$ V – still active ✓. So $\beta \times 2$ moved the collector current by 5% (0.99 $\rightarrow$ 1.04 mA) and the Q-point barely budged. The mechanism: any collector-current rise pushes the emitter voltage up, which *reduces* V_{BE} and throttles the base current – negative feedback inside the bias, provided by R_E. With $R_E = 0$, the active-region formula $I_C = \beta(1.8 - 0.7)/10.2\text{k}$ predicts 10.8 mA at $\beta = 100$ and 21.6 mA at $\beta = 200$. The 2.2 kΩ collector resistor cannot pass either current from 12 V, so the active assumption fails: these are candidates, not actual saturated currents. The formula's β sensitivity is still clear, but the region check must come before accepting a Q-point. Both error sources are now quantified: 11% from divider loading (fixable with a stiffer divider), 5% from β spread (absorbed by R_E).

Common mistakes. Checking $V_{CE} > 0$ instead of > 0.2 V (saturation starts at about 0.2 V). Computing V_C from the *estimated* 1.1 mA and then "checking" against a Q-point computed from the exact current – carry one chain, then check it. Believing the divider alone makes the bias independent of β – it is R_E (with the divider stiff enough to hold V_B) that does the work.

Summary: Thévenin the divider, solve the base loop exactly, check the region; R_E buys β-independence at the price of headroom; estimate-then-verify is fine when the two are labelled as such.

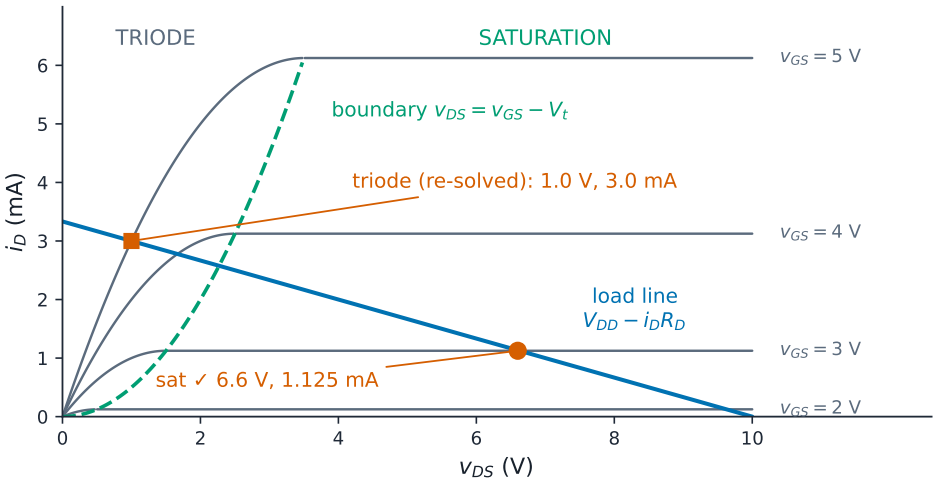
MOSFET biasing

The **MOSFET** is voltage-controlled: the gate draws no DC current, and in **saturation** (the amplifier region – the opposite naming to the BJT!) the **square-law model** gives

$$i_D = \tfrac{1}{2} k_n \frac{W}{L}(v_{GS} - V_t)^2, \qquad v_{DS} \geq v_{GS} - V_t,$$

with **threshold** V_t. The three regions: cut-off ($v_{GS} < V_t$), triode/ohmic ($v_{DS} < v_{GS} - V_t$ – a resistor, not an amplifier), saturation (the inequality holds – a current source).

exercises.lol exercises.com.cn



The boundary parabola vDS = vGS − Vt is the region test; both worked examples sit on this map

Worked example 1 (assume saturation, then check). $V_{DD} = 10$ V, $\frac{1}{2}k_n(W/L) = 0.5$ mA/V^2, $V_t = 1.5$ V, $R_D = 3$ kΩ, gate held at $V_G = 3$ V (divider – no DC gate current). Assume saturation: $I_D = 0.5 \times (3-1.5)^2 = 1.125$ mA; $V_{DS} = 10 - 1.125\text{m} \times 3\text{k} = 6.6$ V. Check: $6.6 \geq 3 - 1.5 = 1.5$ ✓ – saturation stands, and the Q-point is (1.125 mA, 6.6 V).

Worked example 2 (the check that fails, escalated). Raise the gate to $V_G = 5$ V, same circuit. Assume saturation: $I_D = 0.5 \times 3.5^2 = 6.1$ mA, $V_{DS} = 10 - 6.1\text{m} \times 3\text{k} = -8.4$ V – impossible (negative drain voltage), so the assumption is dead: the resistor cannot drop 18 V from a 10 V rail. Re-solve in triode with $i_D = 0.5(2 \times 3.5\, v_{DS} - v_{DS}^2)$ and the load line $10 = 3\text{k}\, i_D + v_{DS}$: the quadratic gives $v_{DS} = 1.00$ V, $i_D = 3.00$ mA. Checks: region $1.0 < 3.5$ ✓ triode; the triode equation returns $0.5(2 \times 3.5 \times 1 - 1) = 3.0$ mA ✓ consistent with the load line; and saturation at this gate would have demanded 18.4 V of supply. The lesson generalises: *the model is an assumption – the region check is what makes it legal.*

Common mistakes. Trusting the saturation current without the region check (the failed example above sails through to a wrong answer if you only check $V_{DS} > 0$). Mixing up the region names with the BJT (MOSFET saturation = amplifier; BJT saturation = ruined). Forgetting the gate current is zero at DC – the divider is unloaded, which is precisely why MOSFET dividers can use megohm resistors.

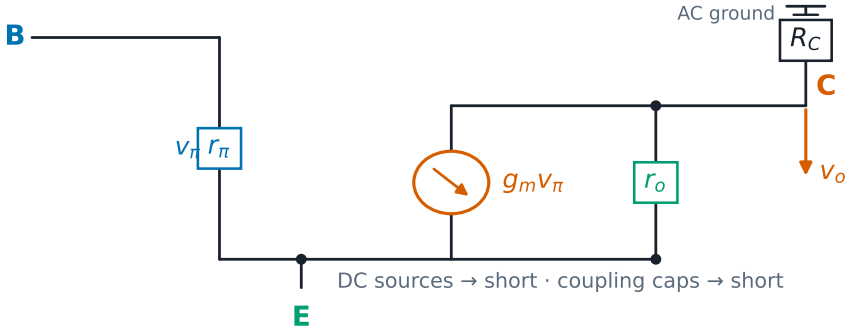
Summary: square law in saturation, boundary $v_{DS} = v_{GS} - V_t$ is the test; assume, solve, check – and when the check fails, the triode quadratic takes over; gate current zero at DC.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Small-signal amplifiers

Amplification is linear only about a **bias point** (the Q-point of unit 5.3). The **small-signal** model replaces each device by its linearised parameters at that point: BJT – $g_m = I_C/V_T$ (differentiate the exponential), $r_\pi = \beta/g_m$, r_o; MOSFET – $g_m = k(W/L)(V_{GS} - V_t) = 2I_D/(V_{GS} - V_t)$ (differentiate the square law), r_o. DC sources become shorts; coupling capacitors become shorts at signal frequency.

The hybrid-π model of the common-emitter stage



One linear circuit answers every small-signal question about the stage

Worked example 1 (the common-emitter stage). At the unit-5.3 bias point $I_C = 0.99$ mA, $\beta = 100$: $g_m = 0.99\text{m}/26\text{m} = 38.1$ mS; $r_\pi = 100/38.1\text{m} = 2.6$ kΩ. Voltage gain with R_E bypassed and $r_o = 50$ kΩ: $A_v = -g_m(R_C \parallel r_o) = -38.1\text{m} \times 2.11\text{k} = -80$. Limiting checks: $r_o \to \infty$ gives $-g_m R_C = -84$ (r_o costs 5%); and the gain must be negative – the CE stage inverts.

Worked example 2 (degeneration and the MOSFET comparison, escalated). Lift the bypass so $R_E = 1$ kΩ participates: the exact gain is $-g_m R_C/(1 + g_m R_E) = -84/(1 + 38.1) = -2.14$, against the $-R_C/R_E = -2.2$ textbook form – within 2.4% because $g_m R_E = 38 \gg 1$ (that is the form's validity condition, now visible). Degeneration traded 84 for 2.2 – linearity and a stable gain bought with the same R_E that stabilised the bias. Same story in MOSFET land: the unit-5.4 Q-point gives $g_m = 2I_D/V_{ov} = 2 \times 1.125\text{m}/1.5 = 1.5$ mS and a common-source gain $-g_m R_D = -4.5$. The BJT at the same 1.125 mA has 43.3 mS – 29× the MOSFET's – transconductance and current gain versus infinite input resistance: the fundamental trade.

Common mistakes. Computing g_m from the *estimated* bias current after the exact solve exists – one chain, one set of parameters. Using $-R_C/R_E$ where $g_m R_E \sim 1$ (the form needs $g_m R_E \gg 1$). Believing the small-signal model is a different circuit – it is the same circuit, differentiated at the Q-point.

exercises.lol exercises.com.cn

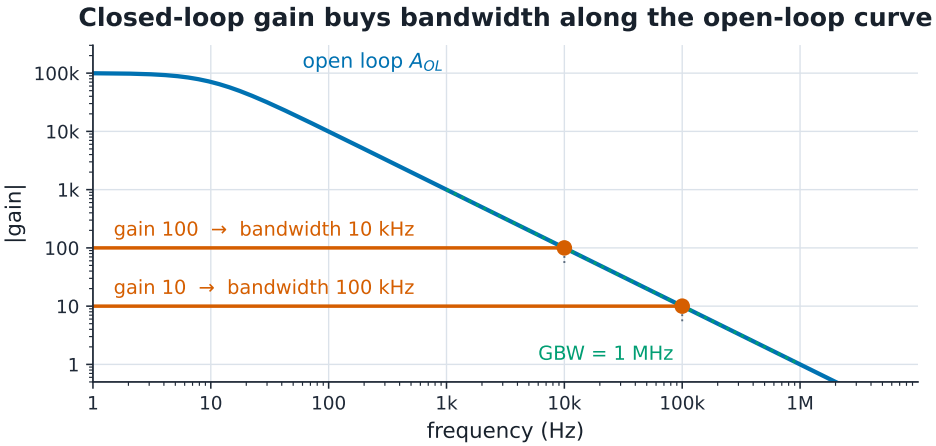
Summary: differentiate the device law at the Q-point to get g_m; $r_\pi = \beta/g_m$; bypassed gain $-g_m R_C$, degenerated $-R_C/R_E$ when $g_m R_E \gg 1$; the BJT wins gm-per-milliamp, the MOSFET wins the gate.

Op-amp feedback circuits

Wave 1's golden rules return with the finite-gain view. Negative feedback closes a loop; what matters is the **loop gain** $A_{OL}\beta_{fb}$. Derivation (inverting stage): if the op-amp can hold $v_- \approx v_+$, the input resistor carries $(v_{in} - 0)/R_{in}$, the feedback resistor carries the same current, and $v_o = -R_f v_{in}/R_{in}$. Writing the exact node equation with open-loop gain A_{OL} and solving gives

$$A_{CL} = -\frac{R_f/R_{in}}{1 + 1/(A_{OL}\beta_{fb})}, \qquad \beta_{fb} = \frac{R_{in}}{R_{in} + R_f},$$

so the golden-rule answer is exact in the limit $A_{OL}\beta_{fb} \to \infty$ and off by the factor $1/(1 + A_{OL}\beta_{fb})$ in real life.



The gain-bandwidth hyperbola: closed-loop gain buys bandwidth along the open-loop curve

Worked example 1 (the four topologies and GBW). Inverting $-R_f/R_{in}$ (input resistance R_{in}); non-inverting $1 + R_f/R_{in}$ (input resistance $\to \infty$); **transimpedance** $v_o = -I_p R_f$ (photodiode: 2 μA $\times$ 1 MΩ = 2 V); summing $-(v_1 R_f/R_1 + v_2 R_f/R_2)$. An op-amp with **gain-bandwidth product** GBW = 1 MHz at closed-loop 40 dB (100×) has bandwidth 10 kHz; at 20 dB (10×), 100 kHz ✓ (the product stays 1 MHz along the hyperbola).

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

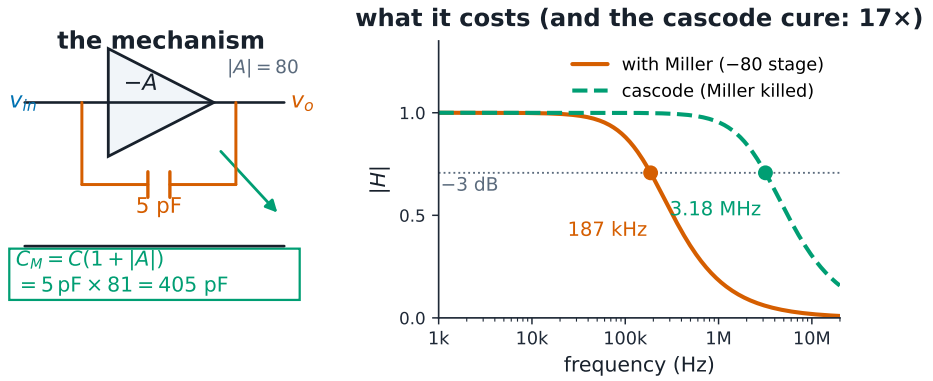
Worked example 2 (how wrong the golden rules are, escalated). The $1/(1 + A_{OL}\beta_{fb})$ error factor, computed. Closed-loop gain 100 ($\beta_{fb} = 0.01$): with $A_{OL} = 10^5$, $A_{CL} = 100/(1 + 1/1000) = 99.90$ – a 0.1% error, invisible on a scope. With a tired or high-frequency op-amp where A_{OL} has fallen to 10^3 (remember unit 5.7: the open-loop gain is already rolling off): $A_{CL} = 100/1.1 = 90.9$ – a 9.1% error. So the golden rules are a *loop-gain statement*: where the loop gain is large they are exact, and near the corner they quietly die. Check: at 10 kHz with GBW 1 MHz the open-loop gain has fallen to $10^6/10^4 = 100$ – equal to the closed-loop gain, loop gain 1, and indeed that is the −3 dB corner found in example 1.

Common mistakes. Using the inverting formula for a non-inverting stage ($1 + R_f/R_{in}$ differs by exactly 1 – at unity gain it is everything). Forgetting the transimpedance stage's feedback resistor sets gain *from current*, so the photodiode's capacitance still shapes the noise gain.

Summary: loop gain $A_{OL}\beta_{fb}$ is the truth meter for the golden rules; error = $1/(1 + A_{OL}\beta_{fb})$; GBW is a hyperbola – closed-loop gain times bandwidth is conserved until the corner, where the rules die.

Frequency response of amplifiers

Every amplifier is a band-pass: **coupling capacitors** set the low corner $f_L = 1/(2\pi R_{eq} C)$ per stage (R_{eq} = what the capacitor sees, sources zeroed), and at the high end **Miller capacitance** plus internal poles take over. The Miller derivation: a capacitor C from output to input of an inverting stage with gain $-A$ carries $i = C\, d(v_{in} - v_o)/dt = C(1 + A)\, dv_{in}/dt$ – the input node behaves as if it saw $C_M = C(1 + A)$ to ground. The **dominant pole** approximation: one pole does most of the rolling, $f_H \approx 1/(2\pi R_{eq} C_{eq})$.



The Miller multiplication and the bandwidth it costs; the cascode reclaims 17×

exercises.lol exercises.com.cn

Worked example 1 (the full band-pass, both corners). The unit-5.5 CE stage: the coupling capacitor sees $R_{eq} = R_B \parallel r_\pi = 10.2\text{k} \parallel 2.6\text{k} = 2.08$ kΩ, so a 10 µF coupling gives $f_L = 1/(2\pi \times 2.08\text{k} \times 10\mu) = 7.7$ Hz. High end: $C_{gb} = 5$ pF between base and collector where the stage gain is −80: $C_M = 5\text{p} \times 81 = 405$ pF, plus 20 pF of stray → $f_H = 1/(2\pi \times 2\text{k} \times 425\text{p}) = 187$ kHz. The stage passes 7.7 Hz to 187 kHz at gain −80. Checks: the coupling cap sees the *parallel* combination (sources zeroed), not the sum – the classic f_L slip; and the −80 inside the Miller factor is the mid-band gain, already falling near f_H – the dominant-pole answer is optimistic by a few percent, the honest reading.

Worked example 2 (the cascode and the cascade, escalated). A cascode holds the collector nearly constant, killing the gain *across* the 5 pF: $f_H = 1/(2\pi \times 2\text{k} \times 25\text{p}) = 3.18$ MHz – 17× for one transistor: Miller explains the disease and the cure. Cascade instead: two identical 187 kHz stages in series are each 3 dB down there, so the pair is 6 dB down – the overall −3 dB point sits at $187 \times \sqrt{2^{1/2} - 1} = 120$ kHz. A multistage amplifier pays for its gain twice – once in Miller, once in cascade.

Common mistakes. Summing instead of paralleling resistances when finding what the coupling capacitor sees; reporting the per-stage corner as the system corner (the cascade shrinks it by $\sqrt{2^{1/n} - 1}$).

Summary: f_L from coupling caps against zeroed-source resistance; Miller multiplies by $(1 + |A|)$; the cascode removes it; cascades shrink corners by $\sqrt{2^{1/n} - 1}$.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

Signals and Systems

University Electrical Engineering

What this course is, and how to use it

This is the sixth course in University Electrical Engineering: **signals and systems** . EE04 asked "what does THIS circuit do at THIS frequency"; this course asks the deeper question: what does a system do to ANY input – answered through the three transforms (convolution, Fourier, z) that the rest of the degree (communications, DSP, control) is built on.

- Unit 6.1 classifies signals and systems.
- Unit 6.2 makes **convolution** the universal time-domain answer.
- Units 6.3–6.4 move to frequency: Fourier series/transforms and sampling.
- Unit 6.5 reuses EE02's Laplace machinery as the system function.
- Units 6.6–6.7 go discrete: difference equations and the z-transform.

The running story is one audio channel: a microphone signal, a filter, a sampler, a digital equaliser. Every transform below moves that channel forward, and every answer is checked the way a signals engineer checks it – by deriving each transform pair at least once, verifying every convolution two ways (graphically and by the Fourier product when available), and testing every discrete result by direct iteration.

Signal representations and system properties

Why it matters. "Is it linear? Is it time-invariant?" are the first two questions of every analysis in this course – the answers decide which of all the machinery is even allowed.

The idea, with a picture. A **signal** is a function of time: continuous-time $x(t)$ or discrete-time $x[n]$. A **system** maps input signals to output signals. Four properties carry the course, and each is a test you run on ONE carefully chosen input, not a feeling:

- **Linearity** : $ax_1 + bx_2 \to ay_1 + by_2$ – feed the sum, check the sum of the outputs.
- **Time invariance** : shift the input by T; the output must shift by exactly T and nothing else.
- **Causality** : the output at time t depends only on x up to t.
- **BIBO stability** BIBO : every bounded input gives a bounded output; for an LTI system this collapses to $\int |h(t)|\, dt < \infty$ – the impulse response must be absolutely integrable.

exercises.lol exercises.com.cn

linearity

feed x1 + x2
output = y1 + y2?

sum in, sum out

time invariance

shift x by T
output shifts by T only?

shift in, shift out

causality

output at t = 0
depends on x(t > 0)?

no output before input

BIBO stability

bounded x in
bounded y out?

integral |h| < infinity

Each property is one experiment: pick the probe input, compare the two outputs

The derivation. Time invariance fails whenever the system's clock scales: for $y(t) = x(2t)$, shifting the input by T gives $y_{shift}(t) = x(2t - T) = x(2(t - T/2))$ – the output is the original system's output delayed by only $T/2$. Comparing "shift then system" ($x(t - T) \to y(t)$ at argument $2t$, i.e. $x(2t - T)$) against "system then shift" ($y(t - T) = x(2t - 2T)$) exposes the mismatch whenever it exists – that comparison IS the test.

> **Worked example 1 – the time-scaling trap.** Is $y(t) = x(2t)$ time-invariant? With $x(t) = u(t)$: shift-then-system gives $x(t-1) = u(t-1) \to u(2t-1) = u(t-\frac{1}{2})$; system-then-shift gives $y(t-1) = u(2(t-1)) = u(t-1)$. Half shift versus full shift – time-varying ✓. A 2× fast-playback stage in the running audio channel fails exactly this way.

> **Worked example 2 – stability by the integral.** The channel's room-echo model $h(t) = e^{-t}u(t)$: $\int_0^\infty e^{-t}\,dt = 1 < \infty$ ✓ stable. The same filter with $h(t) = u(t)$ (an integrator) gives $\int_0^\infty 1\,dt = \infty$ – a bounded step input produces an unbounded ramp output, so the integrator is BIBO-unstable however harmless it looks. One integral, one verdict.

Common mistakes. Testing linearity with one input (you need two, plus scaling). Declaring time invariance from a diagram (run the shift test). And confusing causality with stability: $h(t) = e^{-t}u(t)$ is both, but $h(t) = -e^{t}u(-t)$ (anti-causal, decaying) is stable and NOT causal – the two properties are independent.

Summary. Linearity + time invariance unlock convolution; causality orders cause and effect; BIBO stability is $\int |h| < \infty$. Every property is a one-probe experiment.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

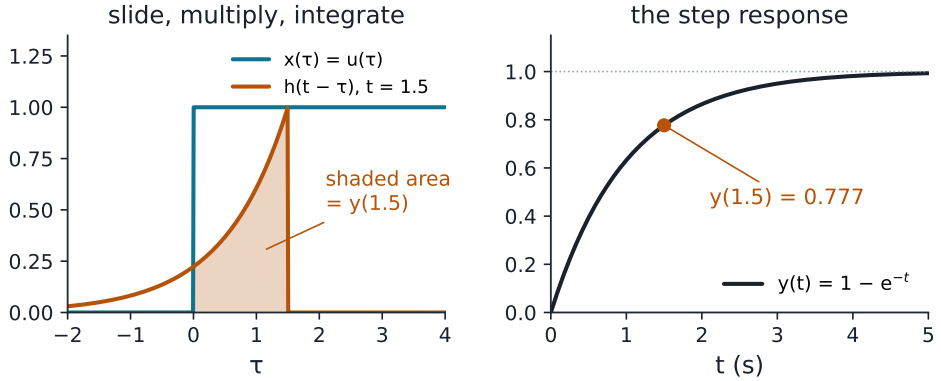
LTI systems and convolution

Why it matters. Convolution is the ONE formula that gives the output of any linear time-invariant system to ANY input – no transform, no transfer function, just the impulse response and an integral.

The idea, with a picture. For a **linear time-invariant** system, the **impulse response** $h(t)$ determines everything: decompose the input into delayed, scaled impulses; each impulse $\delta(t - \tau)\,x(\tau)\,d\tau$ produces $h(t - \tau)$ scaled; superposition integrates them all:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)\,h(t - \tau)\,d\tau = (x * h)(t).$$

Convolution is slide–multiply–integrate: flip h, slide it past x, and each output value is the area of the overlap.



Each value of $y(t)$ is the overlap area as h slides past x

The derivation. Sifting: $x(t) = \int x(\tau)\delta(t - \tau)\,d\tau$. Linearity moves the integral outside; time invariance turns $\delta(t - \tau) \to h(t - \tau)$; the result is the convolution integral – three properties, one formula.

> **Worked example 1 – the classic step response.** The channel's filter $x(t) = u(t)$ into $h(t) = e^{-t}u(t)$: for $t \geq 0$ the overlap is $\tau \in [0, t]$, so

$$y(t) = \int_0^t e^{-(t-\tau)}\,d\tau = e^{-t}(e^{t} - 1) = 1 - e^{-t}.$$

Check the ends: $y(0) = 0$ ✓ (no output before any input arrives); $y(\infty) = 1$ ✓ (DC gain 1 – and indeed $H(0) = 1$). At $t = 1.5$, $y = 1 - e^{-1.5} = 0.777$ – the shaded overlap area in the figure.

exercises.lol exercises.com.cn

Worked example 2 – escalate to the pulse. Now a 2-second rectangular input of height 1 (a held note) into the same filter. Two checkpoints by superposition: the step ON at $t = 0$ plus a step OFF at $t = 2$, so $y(t) = (1-e^{-t})u(t) - (1-e^{-(t-2)})u(t-2)$. At $t = 1$: $1 - e^{-1} = 0.632$. At $t = 3$: $(1-e^{-3}) - (1-e^{-1}) = e^{-1} - e^{-3} = 0.318$ – the release tail. Check the peak at $t = 2^-$: $1 - e^{-2} = 0.865$, and at $t = 2^+$ the second term subtracts exactly its start, so the curve is continuous with a slope discontinuity of -1 – the kink the ear hears as the note-off.

Common mistakes. Forgetting the flip (convolution is NOT $\int x(\tau)h(\tau)\,d\tau$). Dropping the limits when the signals start at 0 (the overlap window, not $(-\infty, \infty)$, does the work). And assuming $x * h = x \cdot h$ – multiplication is what happens in the FREQUENCY domain, never in time.

Summary. $y = x * h$; flip, slide, integrate; widths add ($W_1 + W_2$); commutative; cascades convolve. Every transform later exists to avoid doing this integral.

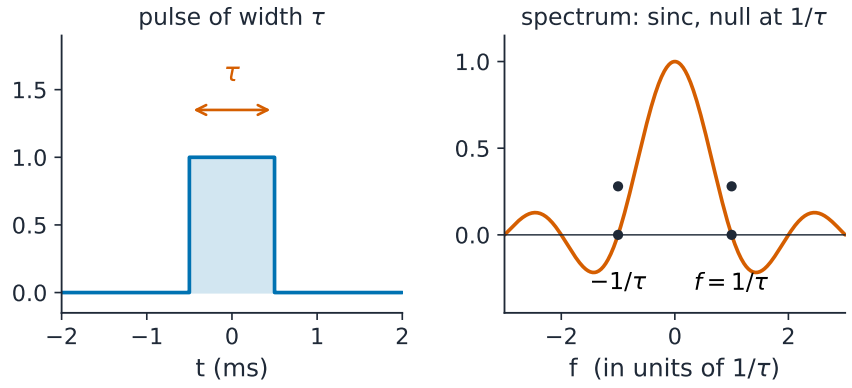
Fourier series and transforms

Why it matters. The channel's noise, its harmonics, its equaliser bands – all live in frequency. The Fourier transform is the bridge, and its convolution-to-multiplication property is why filters are designed there.

The idea, with a picture. Periodic signals expand into harmonics (EE02's series); the **Fourier transform** extends the idea to aperiodic signals:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)\, e^{-j\omega t}\, dt, \qquad x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)\, e^{j\omega t}\, d\omega.$$

Convolution in time becomes multiplication in frequency: $Y(\omega) = X(\omega)H(\omega)$ – the property the whole subject turns on.



The sinc spectrum of a rectangular pulse: null at $1/\tau$, narrower in time means wider in frequency

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

The derivation. For a rectangular pulse of amplitude A and width τ centred at 0:

$$X(\omega) = \int_{-\tau/2}^{\tau/2} A\, e^{-j\omega t}\, dt = A\, \frac{e^{j\omega\tau/2} - e^{-j\omega\tau/2}}{j\omega} = A\tau\, \frac{\sin(\omega\tau/2)}{\omega\tau/2},$$

the sinc function. At $\omega = 0$ the transform equals the AREA $A\tau$ (and $\mathrm{sinc}(0) = 1$ returns it ✓); the first null sits at $\omega\tau/2 = \pi$, i.e. $f = 1/\tau$ – the **time–bandwidth trade-off** .

Worked example 1 – the channel's pulse budget. A 1 ms click ($A = 1$, $\tau = 10^{-3}$ s): $X(f) = 10^{-3}\,\mathrm{sinc}(f \cdot 10^{-3})$, first null at $f = 1$ kHz. A ten-times-shorter click (0.1 ms) pushes the null to 10 kHz – the shorter the event, the wider the spectrum, which is why a click irritates every band of the equaliser at once. Check the area: the value at $f = 0$ is $A\tau$ – 10^{-3} for the 1 ms click and 10^{-4} for the 0.1 ms one ✓; the shorter click puts less into each band but spreads it over ten times as many.

Worked example 2 – escalate to the ideal filter's price. The channel wants a brick-wall low-pass: gain 1 to 4 kHz, 0 above. Its impulse response is the inverse transform of the passband rectangle: $h(t) = \frac{\sin(\omega_c t)}{\pi t}$ with $f_c = 4$ kHz – a sinc in TIME, stretching infinitely in both directions, so the ideal filter is non-causal. Truncate it to $|t| \le 5$ ms (a 10 ms window) and the passband gains ripple (Gibbs), with ripples about $1/(10\text{ ms}) = 100$ Hz apart, and the edge can be no sharper than about that same 100 Hz. The trade-off is honest: steepness in frequency is paid in time-domain length, the same law from the other side.

Common mistakes. Evaluating sinc at ω rather than $\omega\tau/2$ (dimensionless argument, always). Forgetting the $1/2\pi$ on the inverse. And reading $|X(\omega)|$ as "the signal" – the transform is complete only with its phase.

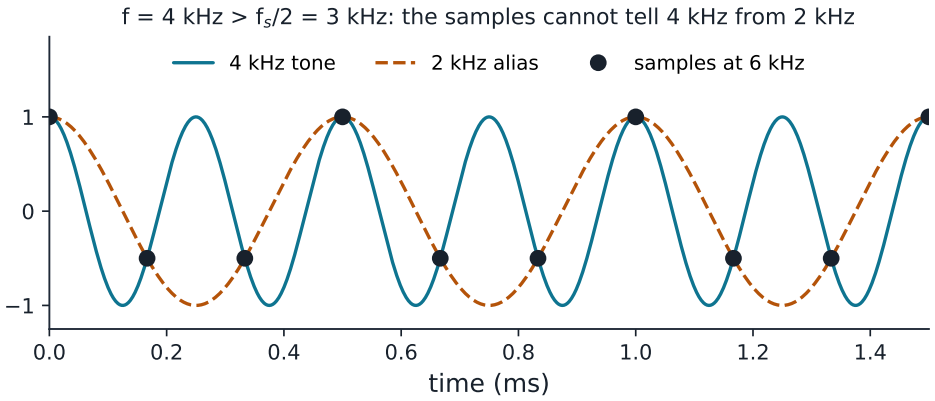
Summary. Transform pairs by the integral; pulse ↔ sinc with null at $1/\tau$; convolution ↔ multiplication. Steepness in one domain is length in the other.

Sampling

Why it matters. The channel goes digital at one point: the sampler. Everything downstream – the equaliser, the storage, the reconstruction – inherits whatever the sampling step did, and it can never be undone afterwards.

The idea, with a picture. Sampling multiplies the signal by an impulse train; in frequency, the spectrum replicates at multiples of f_s. The **Nyquist criterion** : a signal band-limited to f_{max} is perfectly recoverable if $f_s > 2f_{max}$. Below that, replicas overlap and **aliasing** folds high frequencies into the band as false lows.

exercises.lol exercises.com.cn



Sampled at 6 kHz, a 4 kHz tone is indistinguishable from 2 kHz

The derivation. A sampler output $x_s(t) = x(t)\sum_n \delta(t-nT)$ has, by the convolution-to- multiplication duality (multiplication in time = convolution in frequency), the spectrum $(1/T)\sum_k X(\omega - k\omega_s)$ – replicas at every multiple of the sampling frequency. If X occupies $|\omega| < \omega_s/2$, the replicas never touch; any narrower spacing overlaps them, and within the overlap no filter can tell a true low frequency from a folded high one.

> **Worked example 1 – the 4 kHz alias.** A 4 kHz cosine sampled at 6 kHz: $4 > 6/2 = 3$, so the sequence is indistinguishable from $|f - f_s| = 2$ kHz. Check sample by sample: $\cos(2\pi\cdot\frac{4}{6}n) = \cos(2\pi\cdot\frac{2}{3}n)$ and $\cos(2\pi\cdot\frac{2}{6}n) = \cos(2\pi\cdot\frac{1}{3}n)$; the two sequences agree because $\cos(2\pi\cdot\frac{2}{3}n) = \cos(2\pi(\frac{2}{3}-1)n) = \cos(2\pi\cdot(-\frac{1}{3})n) = \cos(2\pi\cdot\frac{1}{3}n)$ ✓ (cosine is even). A 4 kHz tone sampled at 6 kHz IS a 2 kHz sequence.

> **Worked example 2 – escalate to the anti-alias budget.** The channel's microphone carries content to 20 kHz; the sampler runs at 44.1 kHz ($f_s/2 = 22.05$ kHz). A 25 kHz interference tone would alias to $44.1 - 25 = 19.1$ kHz – inside the stated 20 kHz band. The **anti-alias filter** must cut it before the sampler: at 44.1 kHz CD sampling the guard band is only $22.05 - 20 = 2.05$ kHz wide, which is why professional capture oversamples at 96 or 192 kHz (guard bands of 28 and 76 kHz) and filters gently. The fix is always before the sampler, never after – a folded frequency carries no flag saying where it came from.

Common mistakes. Counting $f_s = 2f_{max}$ as safe (the inequality is strict; at equality a sine at f_{max} can be sampled at its zero crossings and vanish). Filtering after sampling (the fold is already inside the band). And assuming digital upsampling recovers what a low f_s missed – the information was never recorded.

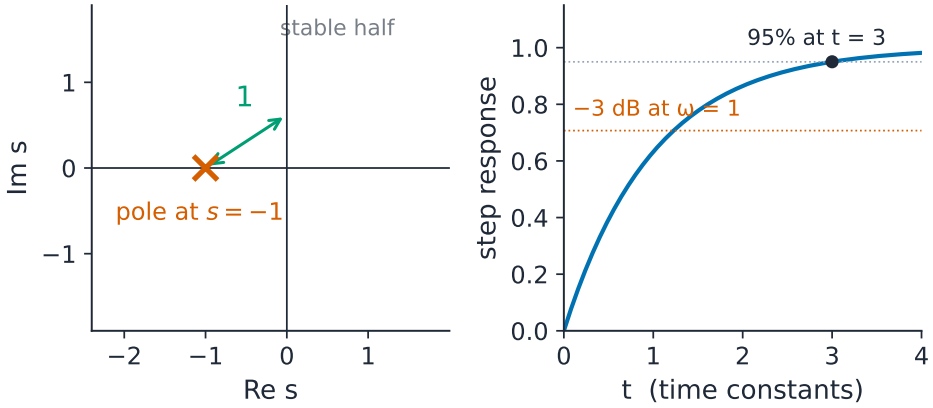
Summary. Sampling replicates the spectrum at f_s; $f_s > 2f_{max}$ or aliases fold; the anti-alias filter goes before the sampler; the guard band between f_{max} and $f_s/2$ is the design budget.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Laplace transform and system functions

Why it matters. The channel's analogue stage – coupling capacitors, tone controls – is a rational system, and the poles of its system function predict ringing, speed and stability at a glance.

The idea, with a picture. EE02's Laplace transform becomes the **system function** $H(s) = Y(s)/X(s)$; for an LTI system, $h(t) \leftrightarrow H(s)$ as a transform pair. Rational $H(s)$ factors into **poles and zeros** : each pole contributes a natural mode e^{pt}; the region of convergence (ROC) decides which; causal stability requires all poles in the open left half-plane.



One pole at −1: the decay rate in time, the −3 dB frequency in the response – one number, three readings

The derivation. $\mathcal{L}\{e^{-at}u(t)\} = 1/(s+a)$ for $\mathrm{Re}(s) > -a$ (one integration by parts). The unit step $1/s$. So for $H(s) = 1/(s+1)$ driven by a step:

$$Y(s) = \frac{1}{s(s+1)} = \frac{1}{s} - \frac{1}{s+1} \implies y(t) = 1 - e^{-t},$$

partial fractions doing the convolution of Unit 6.2 with three lines of algebra – and both routes land on the same curve ✓.

> **Worked example 1 – read the pole three ways.** $H(s) = 1/(s+1)$: the pole at $s = -1$ IS the e^{-t} in the impulse response (time constant 1 s); the step response settles at $1 - e^{-t}$ reaching 95% at $t = 3$ s; the **frequency response** $H(j\omega)$ has $|H(j\cdot 1)| = 1/\sqrt{2} = 0.707$ – the −3 dB point sits exactly at the pole's distance from the origin, and the phase there is −45°.

exercises.lol exercises.com.cn

Worked example 2 – escalate to the two-pole tone control. A bass boost with a notch, $H(s) = \dfrac{s^2 + 2s + 100}{(s+1)(s+10)}$: zeros at $-1 \pm j9.95$ (close to the imaginary axis near ± 10 rad/s, so they carve a notch there: $|H(j10)| = 20/(10.05 \times 14.14) = 0.14$, about -17 dB), poles at -1 and -10 (both left – stable ✓). DC gain $100/10 = 10$ (20 dB of bass lift ✓); high-frequency gain 1 (the numerator's s^2 matches the denominator's). Check the mid-dip: at $\omega = 0$, 20 dB; at $\omega \to \infty$, 0 dB; the transition lives between the pole frequencies – the whole equaliser curve is legible from five marked points on the s-plane.

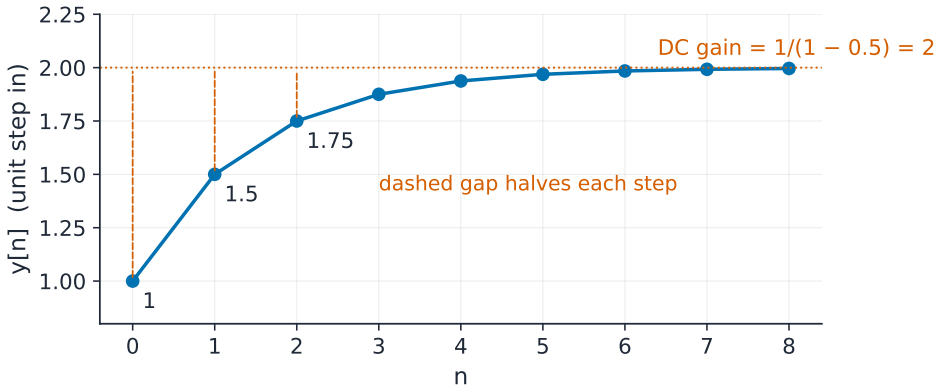
Common mistakes. Reading stability from the poles without the ROC (a pole at -1 with a left-extending ROC is anti-causal and unstable). Forgetting $H(s)$ only exists with its ROC. And trusting a pole-zero plot scaled wrong: the -3 dB point is the pole's DISTANCE from the origin only for a first-order factor with no nearby zeros.

Summary. $H(s) = Y/X$; partial fractions turn transforms back into exponentials; poles set the modes, the ROC the causality, the left half-plane the stability, and $s = j\omega$ the frequency response.

Discrete-time systems

Why it matters. Once sampled, the channel's equaliser is a difference equation running on a processor – iterating it by hand is how its behaviour is actually understood.

The idea, with a picture. The discrete mirror of convolution: $y[n] = \sum_k x[k]\,h[n-k]$. Systems described by **difference equations** such as $y[n] = 0.5\,y[n-1] + x[n]$ – an accumulator with decay. The homogeneous mode is a^n where a is the root of the characteristic equation; iterating one step at a time is the direct solution.



Each step halves the remaining gap to the limit – a geometric approach you can verify on your fingers

The derivation. For $y[n] = a\,y[n-1] + x[n]$ with step input and zero initial condition, the first samples are $1,\ 1+a,\ 1+a+a^2,\ \ldots$ – partial sums of a geometric series, so

$$y[n] = \frac{1 - a^{n+1}}{1 - a} \to \frac{1}{1 - a} \quad (|a| < 1),$$

and the limit $1/(1-a)$ is the DC gain: the discrete echo of the continuous $\int h < \infty$ condition.

Worked example 1 – iterate and converge. $y[n] = 0.5y[n-1] + x[n]$, $y[-1] = 0$, unit step: $y[0] = 1$, $y[1] = 1.5$, $y[2] = 1.75$, $y[3] = 1.875$ – closing on the DC gain $1/(1-0.5) = 2$ ✓. The gap to the limit halves each step (the residue is 0.5^{n+1}), so after 10 steps the gap is $0.5^{11} \approx 0.05\%$ – one line of arithmetic predicting the whole table.

Worked example 2 – escalate to an oscillating root. Set $a = -0.8$: $y[n] = -0.8\,y[n-1] + x[n]$ on a step gives $y[n] = \frac{1}{1.8}(1 - (-0.8)^{n+1})$ – the approach to 0.556 oscillates: $y[0] = 1$, $y[1] = 0.2$, $y[2] = 0.84$, $y[3] = 0.328$. Check $|a| = 0.8 < 1$ – still stable (the oscillation decays), and the sign flips every sample are the fingerprint of a negative root, the discrete analogue of a complex pole pair ringing at half the sampling rate ($z = -0.8$ maps to $s = (\ln 0.8 \pm j\pi)/T$).

Common mistakes. Iterating with a wrong $y[-1]$ (state the initial condition before the first line). Declaring stability from the first few samples (only $|a| < 1$ decides). And reading the DC gain from $n = 0$ alone – it is $1/(1-a)$, the LIMIT, visible only as the samples settle.

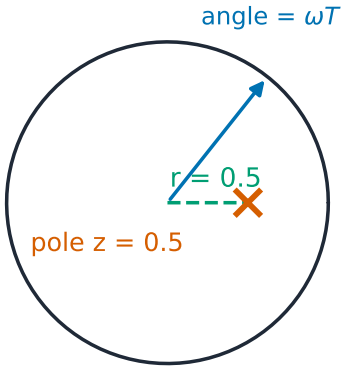
Summary. Difference equation → iterate; geometric partial sums; DC gain $1/(1-a)$; root inside the unit circle = stability; negative or complex roots = oscillation.

The z-transform

Why it matters. The equaliser ships as coefficients a_k, b_k of a difference equation; the z-transform is how those numbers are chosen, and the **unit circle** is where their frequency response is read off.

The idea, with a picture. The **z-transform** $X(z) = \sum_n x[n]z^{-n}$ plays Laplace's role in discrete time. The fundamental pair: $a^n u[n] \leftrightarrow 1/(1 - az^{-1})$ with ROC $|z| > |a|$. For a causal system, poles INSIDE the unit circle mean stability. The inverse transform is long division or partial fractions.

exercises.lol exercises.com.cn



the circle is the frequency axis; radius 0.5 = damping -693 s^{-1} at T = 1 ms

The unit circle is the frequency axis; the pole radius is the damping – one picture joins the two domains

The derivation. $\sum_{n=0}^{\infty} a^n z^{-n}$ is geometric with ratio az^{-1}; it converges to $1/(1-az^{-1})$ exactly when $|az^{-1}| < 1$, i.e. $|z| > |a|$ – the ROC is the exterior of a circle through the pole. Stability needs the ROC to include the unit circle (the frequency axis $z = e^{j\omega}$), which forces $|a| < 1$ – the unit-circle condition in one line.

> **Worked example 1 – the accumulator again, by transform.** $H(z) = 1/(1 - 0.5z^{-1})$, pole at $z = 0.5$ – inside the unit circle, stable ✓. Step response: $Y(z) = H(z)\,\dfrac{z}{z-1} = \dfrac{z^2}{(z-0.5)(z-1)}$; partial fractions give $y[n] = 2(1 - 0.5^{n+1})$. Check $n = 0$: $2(1 - 0.5) = 1$ ✓ matches the iteration of Unit 6.6, and $n \to \infty$ gives 2 ✓ – the DC gain again. Two methods, one sequence.

> **Worked example 2 – the plane-to-plane dictionary.** The mapping $s \to z = e^{sT}$ sends $s = j\omega$ to the unit circle $e^{j\omega T}$ (frequency = angle), and a continuous decay rate σ to a radius $e^{\sigma T}$. The accumulator's pole at $z = 0.5$ with $T = 1$ ms corresponds to $\sigma = \ln(0.5)/0.001 = -693$ s^{-1} – a time constant of $1/693 = 1.44$ ms. Check against Unit 6.6's iteration: the gap halves per SAMPLE ($0.5 = e^{-0.693}$), and $0.693/T$ per second is the same -693 s^{-1} ✓. The two transforms are one subject seen from two banks of the same river.

Common mistakes. Quoting $H(z)$ without its ROC (the same rational function can describe different systems when the ROC changes). Reading frequency response off the real axis (it lives on $|z| = 1$). And confusing z and z^{-1} conventions – for the causal sequence $a^n u[n]$, the ROC is $|z| > |a|$; a left-sided sequence has a different ROC.

Summary. $a^n \leftrightarrow 1/(1 - az^{-1})$, ROC exterior; for a causal system, poles inside the unit circle = stability; $z = e^{sT}$ maps decay to radius and frequency to angle – and the

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

DC gain $1/(1 - a)$ closes the loop with the iteration two units ago.

exercises.lol exercises.com.cn

Electromagnetics

University Electrical Engineering

What this course is, and how to use it

This is the seventh course in University Electrical Engineering: **electromagnetics** – the physics under every circuit, antenna and motor. Circuits hide the fields inside wires; this course opens the box.

- Unit 7.1 reviews the vector calculus toolkit (EE02's operators, now with coordinates and symmetries).
- Units 7.2–7.4 build statics: electrostatics, magnetostatics, and the assembled Maxwell equations.
- Units 7.5–7.6 propagate waves and feed them down transmission lines.
- Unit 7.7 opens waveguides and antennas.

The running story is one link: a 3 GHz radio connection between two boards on a bench – its fields, its wave, its cable, its guide, its antennas. Every field answer below is computed two ways (Gauss vs direct integration; Ampère vs Biot-Savart), and every wave answer is checked against a limiting case (free space, perfect conductor, DC).

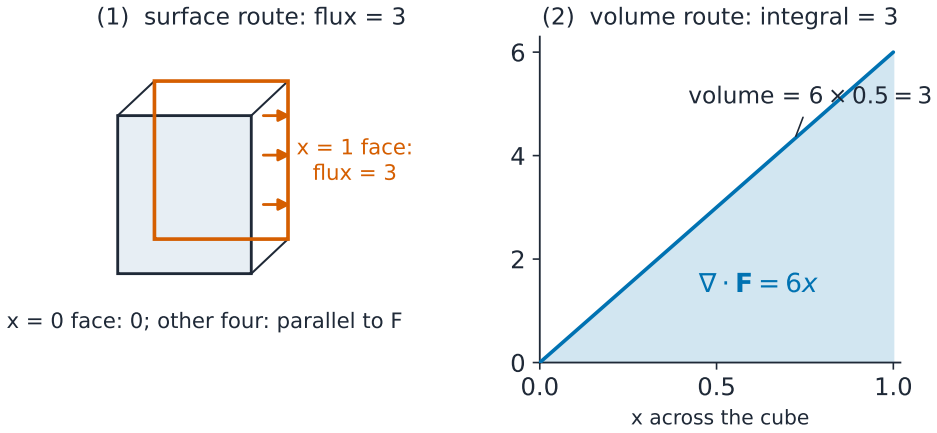
Vector field tools

Why it matters. A phone's GPS receiver hears −130 dBm; whether that works is decided by divergence and curl. The operators of EE02 are the grammar; here they take coordinates and learn the symmetry shortcut that makes real geometry tractable.

The idea, with a picture. A field has an outflow per volume (the **divergence**) and a circulation per area (the **curl**). Before touching an integral, test the symmetry of the source: full spherical symmetry → Gauss's law on a sphere ($dA = r^2 \sin\theta \, d\theta \, d\varphi$); cylindrical → a coaxial cylinder ($dA = r \, d\varphi \, dl$); none → direct integration along the field formula.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn



One cube, two routes: integrate 6x over the box, or count the flux through its faces

The derivation. The divergence theorem

$$\oint_S \mathbf{F} \cdot d\mathbf{A} = \int_V (\nabla \cdot \mathbf{F}) \, dV$$

says the field's net outflow through a closed surface equals the volume integral of its sources. For a symmetric source, the surface integral collapses to $E \times A$; solving for the field is then one line – that one line is every Gauss-law example in this course.

> **Worked example 1 – divergence with coordinates.** Take $\mathbf{F} = (3x^2, 0, 0)$: $\nabla \cdot \mathbf{F} = \partial(3x^2)/\partial x = 6x$, so at the point $(2, 0, 0)$ the divergence is 12 per unit volume. Check at the point $(-2, 0, 0)$: the field still points along $+x$ ($3x^2$ is never negative), yet the divergence is -12 – the sign tracks how the field changes (the source density), not which way it points.

> **Worked example 2 – the theorem on a cube.** The flux of that $\mathbf{F}$ out of the unit cube $[0, 1]^3$, by the volume route: $\int_0^1 \int_0^1 \int_0^1 6x \, dx \, dy \, dz = 6 \cdot \frac{1}{2} = 3$. By the surface route: only the $x = 1$ face carries flux ($3 \cdot 1 \cdot 1 = 3$; the $x = 0$ face has $3x^2 = 0$, the four others are parallel to $\mathbf{F}$) – total 3 ✓. Both routes, one number: the theorem is an accounting identity you can always afford to check.

Common mistakes. Integrating r^2 but differentiating $1/r^2$ (most Gauss examples hide one such step – write both explicitly). Forgetting the sphere's $dA = r^2 \sin\theta \, d\theta \, d\varphi$ has an r^2 that cancels the field's $1/r^2$ – that cancellation IS the method. And applying Gauss's law to asymmetric sources: no symmetry, no $E \times A$, no shortcut.

Summary. Divergence = outflow density, curl = circulation density; symmetry chooses the surface; the divergence theorem is the two-route check that makes every answer here self-verifying.

exercises.lol exercises.com.cn

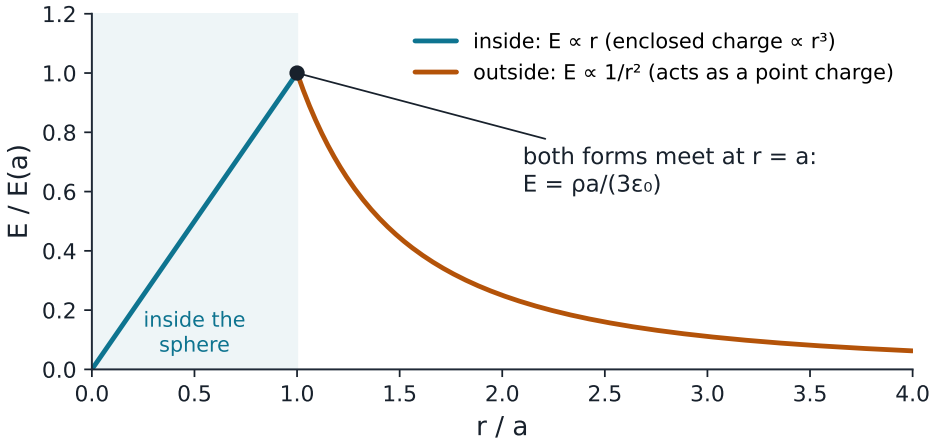
Electrostatics

Why it matters. Every capacitor, every insulation failure, every touch sensor is an electrostatics problem wearing a component label.

The idea, with a picture. The Coulomb field of a point charge, $E = q/(4\pi\varepsilon_0 r^2)\,\hat{r}$, gathers into **Gauss's law** :

$$\oint \mathbf{E}\cdot d\mathbf{A} = \frac{Q_{enc}}{\varepsilon_0},$$

and symmetry converts the integral into arithmetic: for a spherically symmetric source, $E \cdot 4\pi r^2 = Q_{enc}/\varepsilon_0$.



Uniform sphere: $E \propto r$ inside, $E \propto 1/r^2$ outside, continuous at the surface

The derivation. Inside a uniformly charged sphere of radius a and density ρ, the enclosed charge grows as r^3 while the Gaussian sphere's area grows as r^2:

$$E(r)\cdot 4\pi r^2 = \frac{\rho\cdot\frac{4}{3}\pi r^3}{\varepsilon_0} \;\Longrightarrow\; E = \frac{\rho r}{3\varepsilon_0}\ (r<a), \qquad E = \frac{\rho a^3}{3\varepsilon_0 r^2}\ (r>a).$$

The **potential** stacks the field: $V(r) = -\int_\infty^r \mathbf{E}\cdot d\mathbf{l}$; the **capacitance** of an isolated sphere follows, $C = Q/V = 4\pi\varepsilon_0 a$ – about 1.1 pF per cm of radius, why free-space objects barely couple. At a dielectric interface the **boundary conditions** are: tangential E continuous, normal $D = \varepsilon E$ jumping by any free surface charge.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Worked example 1 – the sphere, both regions. With $\rho = 1\ \mu\text{C/m}^3$ and $a = 2$ cm: at the surface $r = a$, both formulas give $E = \rho a/3\varepsilon_0 = (10^{-6}\cdot 0.02)/(3\times 8.854\times 10^{-12}) = 753$ V/m ✓ one value, two routes. At $r = 4$ cm (double the radius) the outside form gives $753/4 = 188$ V/m – the $1/r^2$ falloff, one octave out.

Worked example 2 – escalate to the dielectric. Keep the same free charge density and let a homogeneous, linear dielectric fill all space, inside and outside the charged region. With $\varepsilon_r = 4$, the field everywhere drops by $4\times$ and the surface value becomes 188 V/m. The **polarisation** changes E, while $D = \varepsilon E$ is unchanged for the same free charge. The bound charge inside the sphere is $Q_{bound} = -(1-1/\varepsilon_r)Q_{free} = -0.75Q_{free}$. It is distributed through the charged volume. There is no extra bound surface charge at $r = a$, because the dielectric is the same on both sides and the polarisation is continuous there. A dielectric sphere surrounded by vacuum is a different boundary-value problem.

Common mistakes. Using $4\pi r^2$ where the geometry is a cylinder ($2\pi rl$). Mixing ε_0 and $\varepsilon = \varepsilon_0\varepsilon_r$ mid-problem – pick one convention per problem. And applying the outside (point-charge) form inside the sphere: the enclosed charge, not the total charge, is what Gauss sees.

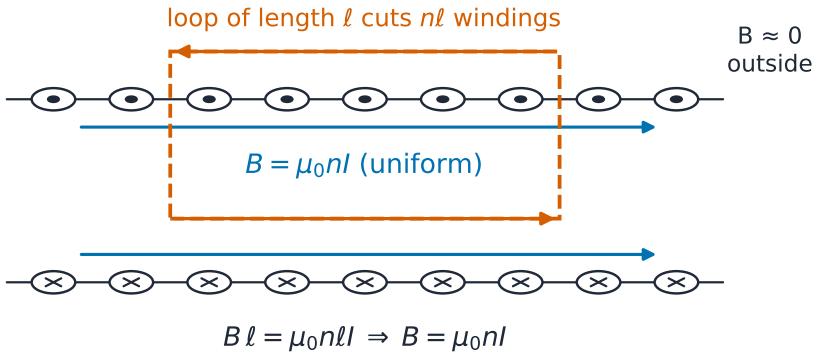
Summary. Gauss's law plus symmetry: $E \times A = Q_{enc}/\varepsilon$; the sphere grows linearly inside, falls as $1/r^2$ outside, agrees at the surface. A dielectric divides fields by ε_r and stores the difference as bound charge.

Magnetostatics

Why it matters. Motors, relays, MRI magnets and every inductor begin with the same law pair: Biot-Savart for the general case, Ampère when symmetry is kind.

The idea, with a picture. Currents make fields. **Biot-Savart law** integrates every current element's contribution; **Ampère's law** $\oint \mathbf{B}\cdot d\mathbf{l} = \mu_0 I_{enc}$ wraps the field around the current. The symmetry test from Unit 7.1 decides which to use.

exercises.lol exercises.com.cn



The rectangular Amperian loop cuts the windings; the outside long side contributes zero

The derivation. For an infinite solenoid with n turns per metre, take a rectangular Amperian loop with one long side inside (length l), one outside, and the short sides crossing the windings. The enclosed current is nlI; the outside field of an infinite solenoid vanishes by cancellation, and the short sides contribute nothing (field perpendicular to them). So $B \cdot l = \mu_0 nlI$:

$$B = \mu_0 nI \quad \text{(inside, uniform)}, \qquad B \approx 0 \ \text{(outside)}.$$

Magnetic flux $\Phi = \int \mathbf{B} \cdot d\mathbf{A}$ leads to **inductance** $L = N\Phi/I$; for the solenoid, $L/l = \mu_0 n^2 A$. The **Lorentz force** $\mathbf{F} = q\mathbf{v} \times \mathbf{B}$ bends moving charge into a circle of radius $r = mv/qB$ – the cyclotron orbit.

> **Worked example 1 – the solenoid priced.** With $n = 1000$ turns/m and $I = 2$ A: $B = 4\pi \times 10^{-7} \times 1000 \times 2 = 2.51$ mT inside, uniform along the axis. A Biot–Savart check would integrate a thousand loops – this is why the symmetry test comes first: one line instead of a page.

> **Worked example 2 – the orbit and the force.** A 1 μC charge at 10^6 m/s moving perpendicular to a 1 T field feels $F = qvB = 10^{-6} \times 10^6 \times 1 = 1$ N – and circles at radius $r = mv/qB$: for an electron ($m = 9.11 \times 10^{-31}$ kg), $r = (9.11 \times 10^{-31} \times 10^6)/(1.6 \times 10^{-19} \times 1) = 5.7$ μm – the cyclotron radius that sets magnetron and spectrometer geometry. Check the speed: an electron at 10^6 m/s carries 2.85 eV – comfortably non-relativistic, so the formula is honest.

Common mistakes. Forgetting Ampère's law needs the TOTAL enclosed current (loop with N turns $\to NI$). Assuming **B** is continuous across a current sheet (its tangential component jumps by $\mu_0 K$). And using the solenoid formula for a short coil: "infinite" means length $\gg$ diameter, or the field is weaker and non-uniform.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Summary. Ampère plus symmetry: $B \cdot l = \mu_0 I_{enc}$; the infinite solenoid concentrates flux, $B = \mu_0 nI$, $L/l = \mu_0 n^2 A$; moving charge circles at $r = mv/qB$.

Maxwell's equations

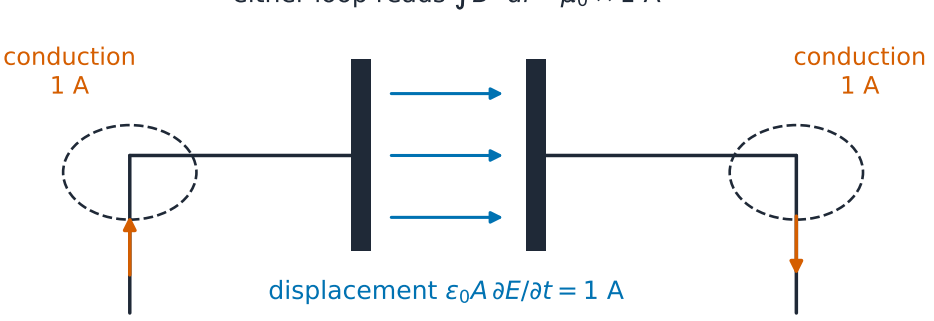
Why it matters. Every radio, every photon, every charger's wireless coupling rides on the one term Maxwell added to close the loop.

The idea, with a picture. Statics + **displacement current** completes the set:

$$\nabla \cdot \mathbf{E} = \rho/\varepsilon_0 \qquad \nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t \qquad \nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \, \partial \mathbf{E}/\partial t$$

The four boxes are a loop: circulating **E** makes changing **B**, circulating **B** makes changing **E** – self-sustaining waves.



The loop integral cannot depend on which surface you span it with – displacement current makes it true

The derivation. Charge conservation $\nabla \cdot \mathbf{J} = -\partial\rho/\partial t$ forces the correction: take the divergence of the original Ampère law ($\nabla \times \mathbf{B} = \mu_0 \mathbf{J}$) and the left side is identically zero while the right side is $-\mu_0 \partial\rho/\partial t$ – nonzero whenever charge piles up. Adding $\mu_0 \varepsilon_0 \partial \mathbf{E}/\partial t$ repairs it, because Gauss's law turns that term's divergence into exactly $+\mu_0 \partial\rho/\partial t$. The loop closes.

> **Worked example 1 – the capacitor's missing ampere.** A parallel-plate capacitor charges at $dQ/dt = 1$ A. Between the plates no conduction current exists, yet an Ampèrian loop around the wire must read the same 1 A whichever surface you span it with. Between the plates $E = Q/(\varepsilon_0 A)$, so $\varepsilon_0 A \cdot \partial E/\partial t = dQ/dt = 1$ A ✓ – the displacement current IS the conduction current, numerically.

exercises.lol exercises.com.cn

Worked example 2 – escalate to the wave speed. In source-free space ($\rho = 0$, $\mathbf{J} = 0$), take the curl of the **E** curl equation and substitute the **B** one:

$$\nabla^2\mathbf{E} = \mu_0\varepsilon_0\,\partial^2\mathbf{E}/\partial t^2,$$

the wave equation with speed $c = 1/\sqrt{\mu_0\varepsilon_0}$. Insert the numbers: $1/\sqrt{\mu_0\varepsilon_0} = 3.0 \times 10^8$ m/s – light, from two constants measured on a bench. The link budget of the running story begins here.

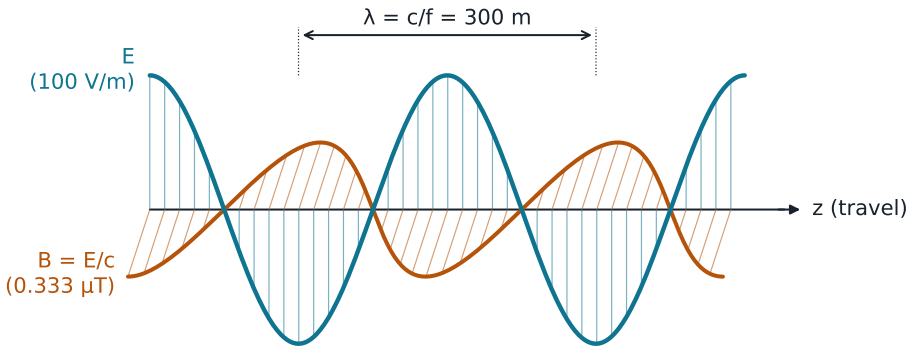
Common mistakes. Treating displacement current as optional bookkeeping (it is what makes the equations consistent and what carries the wave). Forgetting which equation has the minus sign (Faraday's: it is Lenz's law, the induced field opposing the change). And applying the static forms to antennas: radiation IS the correction term doing its work.

Summary. Four equations, one loop; displacement current closes charge conservation and, in vacuum, yields the wave equation at $c = 1/\sqrt{\mu_0\varepsilon_0}$.

Plane waves

Why it matters. Wi-Fi, radar, sunlight and the bench link's 3 GHz carrier are all the same solution: the plane wave.

The idea, with a picture. In source-free space the wave equation's simplest solution is a **plane wave** : **E**, **B** and the propagation direction mutually perpendicular, in phase, with $B = E/c$. The **intrinsic impedance** $\eta_0 = \sqrt{\mu_0/\varepsilon_0} = 377\ \Omega$ of free space sets the ratio of E to H; the **Poynting vector** $\mathbf{S} = \mathbf{E} \times \mathbf{H}$ points along the propagation with $S = E^2/\eta_0$ (instantaneous; with rms values of E this is the average power density, while a sinusoid of amplitude E_0 averages $E_0^2/2\eta_0$).



E and B in phase and at right angles; λ = 300 m at 1 MHz

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

The derivation. For $\mathbf{E} = E_0\cos(\omega t - kz)\,\hat{x}$, Faraday's law gives $\partial B/\partial t = -\partial E/\partial z$, integrating to $B = (k/\omega)E = E/c$. The power density follows from the impedance: $H = E/\eta_0$, so $S = E \times H = E^2/\eta_0$ – one number per wave.

Worked example 1 – reading the numbers off. A 100 V/m (rms) wave at 1 MHz: $B = 100/(3 \times 10^8) = 0.333\ \mu$T; wavelength $\lambda = c/f = 300$ m; power density $S = 100^2/377 = 26.5$ W/m^2. Check the impedance route: $H = 100/377 = 0.265$ A/m, and $E \times H = 26.5$ ✓ – the two routes agree because they are the same ratio.

Worked example 2 – escalate to the conductor's skin. In a **good conductor** the wave decays as $e^{-\alpha z}$ with **skin depth** $\delta = \sqrt{2/(\omega\mu\sigma)}$. Copper ($\sigma = 5.8 \times 10^7$): at 50 Hz, $\delta = \sqrt{2/(2\pi \cdot 50 \cdot 4\pi \times 10^{-7} \cdot 5.8 \times 10^7)} = 9.3$ mm; at 1 GHz, $\delta = 2.1\ \mu$m – about 4500 times thinner for twenty million times the frequency ($\delta \propto 1/\sqrt{f}$ ✓). This one number explains why shields work, why RF current rides the surface, and why the bench link's 3 GHz traces need wide, smooth copper: at 3 GHz the current lives in the top 1.2 µm, so surface roughness and any plating there set the loss (gold plating protects against corrosion; it conducts worse than copper).

Common mistakes. Using c in a dielectric – the speed there is $1/\sqrt{\mu\varepsilon}$ (≈ 0.67c in the coax below). Mixing E/H (377 Ω in space) with E/B (c – a speed, not an impedance). And forgetting the wave's fields are in phase in a LOSSLESS medium only.

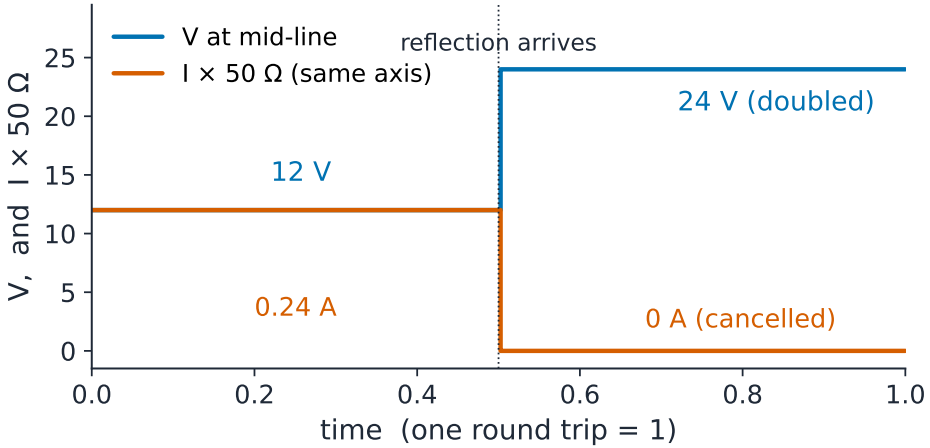
Summary. Plane wave: $E \perp B \perp \hat{k}$, $B = E/c$, $S = E^2/\eta_0$; conductors eat waves as $e^{-z/\delta}$ with $\delta = \sqrt{2/\omega\mu\sigma}$.

Transmission lines

Why it matters. The bench link's two boards are joined by 40 cm of coax; at 3 GHz that is six wavelengths – a circuit no more, a transmission line.

The idea, with a picture. The **telegrapher equations** treat the line as distributed inductance L' and capacitance C' per metre: $\partial v/\partial x = -L'\partial i/\partial t$ and $\partial i/\partial x = -C'\partial v/\partial t$. Their solutions are voltage/current waves with $v_p = 1/\sqrt{L'C'}$ and the line's characteristic impedance $Z_0 = \sqrt{L'/C'}$.

exercises.lol exercises.com.cn



The open-end signature: voltage doubles, current goes to zero

The derivation. Differentiate the first equation in x and the second in t and eliminate $\partial^2 i/\partial x \partial t$: a wave equation for v with speed $1/\sqrt{L'C'}$. At a load Z_L, matching gives the reflection coefficient $\Gamma = (Z_L - Z_0)/(Z_L + Z_0)$: a short ($\Gamma = -1$) inverts, an open ($\Gamma = +1$) doubles the voltage and cancels the current – the **bounce diagram** tracks every bounce.

Worked example 1 – a real coax. RG-316-class coax with $L' = 250$ nH/m and $C' = 100$ pF/m: $Z_0 = \sqrt{250\times10^{-9}/100\times10^{-12}} = \sqrt{2500} = 50\ \Omega$ ✓ (the standard cable – the numbers were chosen to make it so); $v_p = 1/\sqrt{250\times10^{-9}\times100\times10^{-12}} = 2\times10^8$ m/s $= 0.67c$. The bench link's 40 cm of it is SIX wavelengths at 3 GHz ($\lambda = v_p/f = 6.7$ cm) – proving the "line, not circuit" point with the parts on the bench.

Worked example 2 – the open-ended step. Drive that 50 Ω line with a 12 V step, open-ended. The incident wave carries 12 V and $12/50 = 0.24$ A. At the open, $\Gamma = +1$: the reflected +12 V doubles the load voltage to 24 V and the reflected current −0.24 A cancels the forward current to zero – an open circuit draws no steady current ✓, and the VSWR tends to infinity, the open-circuit signature the standing-wave meter reads.

Common mistakes. Using Z_0 as a physical resistance (it is a ratio; a lossless 50 Ω line dissipates nothing). Computing λ with c instead of v_p (30% error in PTFE). And forgetting the source-side reflection: a mismatched driver re-reflects, and the bounce diagram must include it.

Summary. $Z_0 = \sqrt{L'/C'}$, $v_p = 1/\sqrt{L'C'}$; $\Gamma = (Z_L - Z_0)/(Z_L + Z_0)$; opens double voltage, shorts cancel it; long lines are waves, and the bounce diagram is the book-keeping.

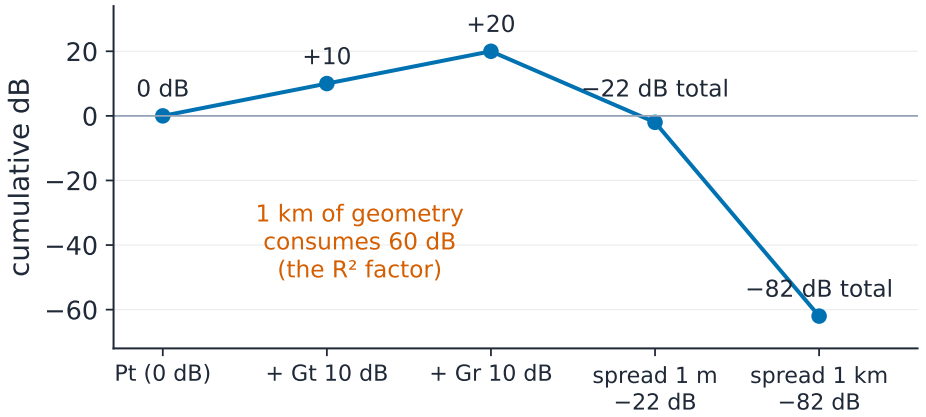
exercises.lol exercises.com.cn

Waveguides and antennas

Why it matters. Above ~10 GHz coax loses too much; the bench link's backhaul (and every radar) moves into hollow pipe, and the last metre of any radio link is always an antenna.

The idea, with a picture. A hollow conducting pipe guides waves only above the **cutoff** frequency: the rectangular guide's dominant mode cuts off at $f_c = c/(2a)$ (a = wide dimension). Above cutoff the guide **wavelength** stretches: $\lambda_g = \lambda/\sqrt{1-(f_c/f)^2}$ – the wave zigzags, so fewer axial cycles fit per metre of pipe. Antennas close the link: the half-wave **dipole** has gain 2.15 dBi and input resistance ≈ 73 Ω; the **Friis equation** prices the whole link:

$$P_r/P_t = G_r G_t \left(\frac{\lambda}{4\pi R}\right)^2.$$



Every dB in the chain is a design decision; the last term is pure geometry

The derivation. Between the plates of the guide, the field must fit a standing half-wave across a: $a = \lambda_c/2$ at cutoff, so $f_c = c/2a$. Along the guide, the propagation constant satisfies $k_g^2 = k^2 - (\pi/a)^2$; solving for the wavelength gives the stretch formula above. The Friis chain multiplies: transmit power × both gains × the spreading factor $(\lambda/4\pi R)^2$ – an area ratio, the antenna aperture over the sphere's area.

Worked example 1 – WR-90 at X-band. WR-90 waveguide has $a = 22.86$ mm: $f_c = 3\times10^8/(2\times0.02286) = 6.56$ GHz – X-band (8.2–12.4 GHz) rides above cutoff. At 10 GHz ($\lambda = 30$ mm): $\lambda_g = 30/\sqrt{1-0.656^2} = 39.8$ mm – 33% longer than free space. Check the limit: as $f \to f_c$, $\lambda_g \to \infty$ (the wave stops); as $f \to \infty$, $\lambda_g \to \lambda$ (the zigzag vanishes) ✓.

Worked example 2 – the bench link's budget. Two 10-dBi (G = 10) antennas, 1 m apart on the bench, at 3 GHz ($\lambda = 0.1$ m): $P_r/P_t = 10 \times 10 \times (0.1/(4\pi \times 1))^2 = 100 \times 6.33 \times 10^{-5} = 6.3 \times 10^{-3}$ – 22 dB of loss, still easy on the bench. Scale to the same boards 1 km apart: $(0.1/(4\pi \times 1000))^2 = 6.33 \times 10^{-11}$, times $100 = 6.3 \times 10^{-9}$ (-82 dB) – the link budget that decides dish sizes. One formula, two scales, and the ratio between them is pure geometry (R^2, a factor of 10^6 ✓).

Common mistakes. Using the guide's NARROW dimension for f_c (cutoff is set by the wide one). Applying λ_g as if it were λ in Friis (the free-space wavelength belongs there). And quoting dipole gain in dB without the "i": 2.15 dBi is relative to isotropic; relative to a dipole it is zero.

Summary. Guide: $f_c = c/2a$, $\lambda_g = \lambda/\sqrt{1-(f_c/f)^2}$ (stretched, infinite at cutoff). Antennas: Friis $P_r/P_t = G_r G_t(\lambda/4\pi R)^2$ – geometry decides the last term every time.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

Microprocessors and Embedded Systems

University Electrical Engineering

What this course is, and how to use it

This is the eighth course in University Electrical Engineering: **microprocessors and embedded systems** . EE03 built the logic; this course builds the machine that runs programs, and the discipline of making hardware and software meet.

- Unit 8.1 maps the architecture: CPU, registers, memory, and the fetch-decode-execute cycle.
- Units 8.2–8.3 program it: embedded C and assembly, then memory-mapped I/O.
- Units 8.4–8.5 add time: interrupts/timers and the serial protocols.
- Units 8.6–8.7 close with analog interfaces and debugging method.

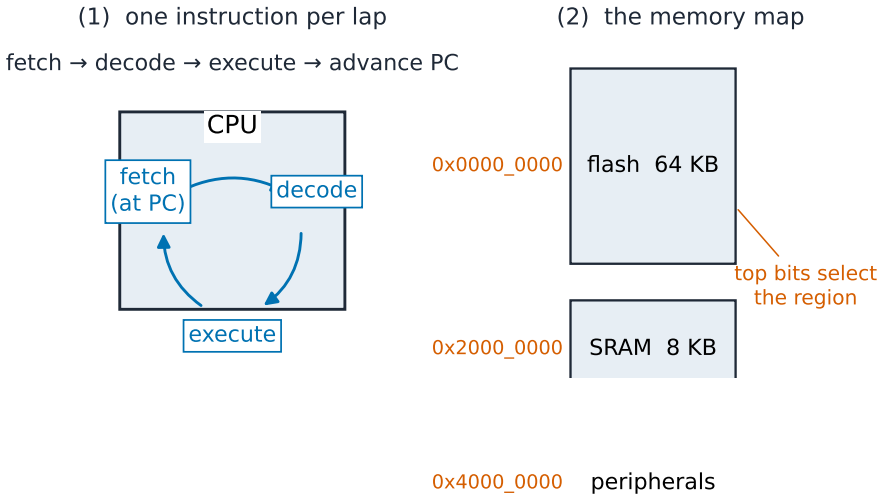
The running story is one small product: a battery-powered sensor node with a button, an LED, a serial link and a 12-bit sensor. Every unit below moves that node forward, and every number is checked the way a firmware engineer checks it – by bit widths and limits (does the value fit?), by the clock it assumes (every timing closes against 84 MHz), and by tracing the exact state changes of every code fragment.

Microcontroller architecture

Why it matters. The thermostat on the wall, the anti-lock brake controller and the earbud on the desk each run one of these: a complete computer on a chip the size of a grain of rice, costing less than the postage that ships it.

The idea, with a picture. A **microcontroller** integrates three things on one chip: flash memory holding the program, SRAM holding the data, and a CPU that runs the **fetch-decode-execute cycle** : the **program counter** (PC) holds the address of the next instruction, the control unit decodes it, the datapath executes it, and the PC advances (or jumps). Every transfer rides the **bus** – address lines saying where, data lines carrying what, control lines saying read or write – sequenced by the **clock** .

exercises.lol exercises.com.cn



One instruction per lap: fetch at the PC, decode, execute, advance

The derivation. Addresses are unsigned integers, so an N-bit address field addresses 2^N locations exactly. The whole memory map follows:

$$\text{size} = 2^N \text{ bytes} \iff N = \log_2(\text{size}) \text{ address lines.}$$

Flash at `0x0000_0000` and SRAM at `0x2000_0000` never collide because their top bits differ – the address decoder just looks at bits [31:16].

> **Worked example 1 – sizing the memory map.** The node's chip has 64 KB flash and 8 KB SRAM. Flash: $64\,\text{KB} = 2^{16}$ bytes → 16 address lines, so it occupies `0x0000_0000`–`0x0000_FFFF`. SRAM: $8\,\text{KB} = 2^{13}$ bytes → 13 lines, at `0x2000_0000`–`0x2000_1FFF`. Check: any address with bits [31:16] = `0x0000` and [15:0] ≤ `0xFFFF` lands in flash; `0x2000_…` lands in SRAM; the regions cannot overlap because their top bits differ ✓.

> **Worked example 2 – how fast is the loop itself?** The node's CPU runs at 84 MHz with a nominal one instruction per cycle (a Cortex-M class core): 84×10^6 instructions per second, so one instruction takes $\dfrac{1}{84 \times 10^6} = 11.9$ ns. A 100-instruction control pass therefore costs 1.19 µs – and a 1 ms control loop has room for about 840 passes if nothing else intervenes. The budget arithmetic that starts here is reused in every later unit.

Common mistakes. Mixing address bits with data bits (the address lines count locations, the data lines their width). Forgetting that the map is fixed in the linker script: the same chip boots at whatever address the hardware vendor chose. And quoting program size in bits when flash sizes are in bytes – a factor of eight that sinks the part choice.

Summary. One chip: flash + SRAM + CPU cycling fetch, decode, execute. An N-bit address field addresses 2^N bytes; regions with distinct top bits cannot overlap; the clock turns instruction count into seconds.

Embedded C and assembly

Why it matters. The node's firmware is a few thousand lines of C touching hardware registers directly. Nothing between your line of code and a transistor – which is exactly why the mapping from C to machine must be precise.

The idea, with a picture. A **memory-mapped register** is just an address; C reaches it through a `volatile` pointer (volatile tells the compiler: this memory changes outside your view, touch it every time). The compiler turns C into loads, stores, arithmetic and branches; **assembly** is the same program written in those instructions by hand, one line each.

C

```
*GPIO_OUT |= (1u << 5);          set bit 5 only
```

assembly: load, mask, store

```
LDR  r0, =0x20000100     address of GPIO_OUT
LDR  r1, [r0]            bus transfer 1: read
ORR  r1, r1, #0x20       inside the CPU, no bus
STR  r1, [r0]            bus transfer 2: write
```

register transfer

```
GPIO_OUT ← GPIO_OUT | 0x20        other bits unchanged
```

One C line, one instruction sequence, one register transfer – three views of one act

The derivation. Read-modify-write is the fundamental act: to change one pin without disturbing its neighbours, the CPU must load the register, mask the bit locally, and store it back. That is three operations but only two bus transfers: one read and one write. This example uses a plain read/write word register; some real peripherals also provide atomic set/clear registers:

```
volatile uint32_t *const GPIO_OUT = (uint32_t *)0x40000000u;
*GPIO_OUT |= (1u << 5);   // load, OR, store - set bit 5 only
```


exercises.lol exercises.com.cn

Worked example 1 – trace the counter. The C loop

```
uint32_t count = 0;
while (1) { count = count + 1; }
```

An illustrative load-increment-store plus branch implementation traces $0, 1, 2, 3, \ldots$. A compiler may instead keep this non-volatile counter in a CPU register. Check the limit: the largest value is $2^{32} - 1 = 4\,294\,967\,295$; the next increment wraps to zero after 2^{32} increments from zero. Assuming four clock cycles per iteration at 84 MHz, the wrap takes $\dfrac{4\,294\,967\,296 \times 4}{84 \times 10^6} \approx 204$ s – arithmetic, not waiting three minutes with a stopwatch.

> **Worked example 2 – the bit mask, escalated.** To toggle bits 5 and 6 and clear bit 3 in one expression: the mask is $(1 \ll 5) \,|\, (1 \ll 6) = 0x60$. Then `*GPIO_OUT = (*GPIO_OUT & ~0x08) ^ 0x60` clears bit 3 and flips 5 and 6 in one read-modify-write. Verify on paper with `GPIO_OUT = 0xCE`: `0xCE & ~0x08 = 0xC6`, then `0xC6 ^ 0x60 = 0xA6` – bit 3 now 0 (was 1), bits 6 and 5 flipped from 1, 0 to 0, 1 ✓. The hex check on paper is the fastest debugger in this unit.

Common mistakes. Dropping `volatile` (the compiler caches the register in a CPU register and your loop reads it forever). Assuming C evaluation order between two register writes when the hardware needs them in sequence – insert a barrier or read-back. And writing bit 5 as `*GPIO_OUT = 0x20`, which clears every other bit on the port.

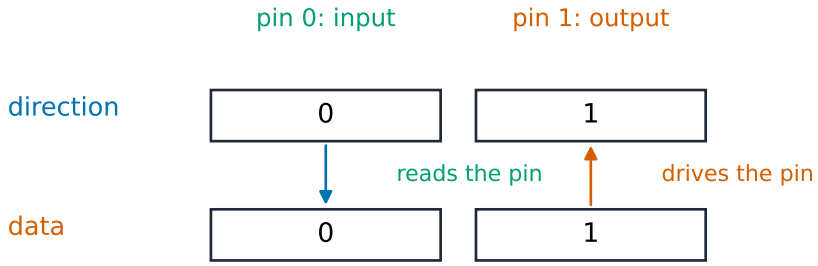
Summary. Registers are addresses; `volatile` keeps the compiler honest; this plain word-register example uses load-mask-store. Trace state on paper in hex before trusting the silicon.

Memory and I/O

Why it matters. The node's button, LED and sensor all live outside the CPU. How they connect decides whether the firmware is a clean table of addresses or a pile of magic numbers.

The idea, with a picture. Memory-mapped I/O makes a peripheral register look exactly like a memory location: reading a data register receives a byte; writing one sends it. A **GPIO** port has two such registers – direction (input or output, one bit per pin) and data (the pin's level) – so configure once, then read or write forever.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn



configure direction once, then read or write data

Direction says which way the pin speaks; data says what it says

The derivation. Polling is the honest baseline: the status register's done-bit is a memory read in a loop. Each lap costs the loop's instructions; waiting T seconds burns

$$T \times f_{\text{clock}}$$

CPU cycles, or approximately $T f_{\text{clock}}/C$ complete laps when each lap costs C clock cycles. The factor C changes the number of checks, not the CPU time occupied by a continuous busy wait. Detection and timeout tests are quantised to the loop's check times.

> **Worked example 1 – the cost of asking.** The node polls a sensor-ready bit at 84 MHz; each loop lap is 4 cycles (load, AND, branch, increment), so the CPU performs $84 \times 10^6/4 = 21$ million laps per second. Waiting 1 ms is ~21 000 laps = 84 000 cycles: 100% of the CPU for that millisecond, with zero useful work.

> **Worked example 2 – the whole node's map.** The node's peripherals sit at `0x4000_0000` (GPIO), `0x4001_0000` (UART), `0x4002_0000` (timer), `0x4003_0000` (ADC). A firmware constant table maps each driver to its base; a typo in one entry (`0x4001_0000` vs `0x4001_0004`) lands four bytes into the UART's register set – which is why every driver file opens with named offsets, never raw addresses. The check: after init, read each peripheral's ID register back; a wrong base reads garbage on the first try.

Common mistakes. Forgetting the direction register (the pin stays an input and the write does nothing). Polling with no timeout – a dead sensor hangs the whole product. And mixing up a port's bit number with the pin number on the package: the datasheet's pin table, not the header's convenience macro, is the truth.

Summary. Peripherals are memory: base plus named offsets. Configure direction

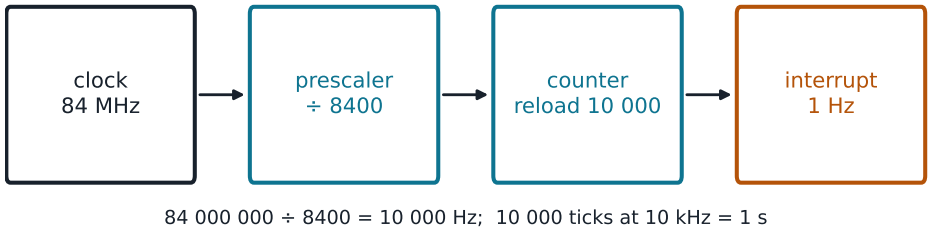
exercises.lol exercises.com.cn

once, poll with a timeout, and let a read-back ID check prove the map before any real work runs.

Interrupts and timers

Why it matters. The node runs on a battery for a year. Polling burns the battery asking questions; interrupts let the CPU sleep until the world needs it – the difference between a week and a year.

The idea, with a picture. An **interrupt** suspends the main program where it stands: the hardware pushes the return state, **vectors** to an **interrupt service routine** (ISR) chosen by the **NVIC** (the nested vectored interrupt controller, which also ranks priorities), the ISR runs and returns. A **timer** is just a counter clocked through a divider – the **prescaler** . In the STM32-style upcounter used here, an update event occurs when CNT wraps from ARR back to zero; an enabled interrupt can then call the ISR.



Two divisions turn 84 MHz into a 1 Hz interrupt

The derivation. Timer frequency divides twice:

$$f_{\text{interrupt}} = \frac{f_{\text{clock}}}{PR}, \qquad P = PSC + 1, \quad R = ARR + 1,$$

where PSC and ARR are the stored 16-bit register values, at most 65,535. Each divider factor can reach 65,536 because zero is counted too. A design is valid when the stored values fit AND $PR = f_{\text{clock}}/f_{\text{interrupt}}$. Here ARR is nonzero and the counter runs continuously.

> **Worked example 1 – a 1 Hz heartbeat.** From 84 MHz: divider factor 8400 gives $84 \times 10^6/8400 = 10$ kHz; another factor of 10,000 gives exactly 1 Hz. Write PSC = 8399 and ARR = 9999. Both stored values fit 16 bits, and $8400 \times 10\,000 = 84 \times 10^6$ ✓ closes the design. A divider factor of 84,000 would need PSC = 83,999, which does not fit.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

> **Worked example 2 – interrupt latency vs polling, priced.** A GPIO interrupt with 12 cycles of entry, a 30-instruction ISR and 10 cycles of return costs $12 + 30 + 10 = 52$ cycles $= 0.62$ µs at 84 MHz – against the 21 000 laps the polled wait burned. But the ISR's cost is paid on every event: at 1 kHz of events that is 52 000 cycles/s = 0.06% of the CPU; at 1 MHz of events it is 62% – past some rate, polling a fast peripheral is cheaper than interrupting. The crossover arithmetic is the design decision.

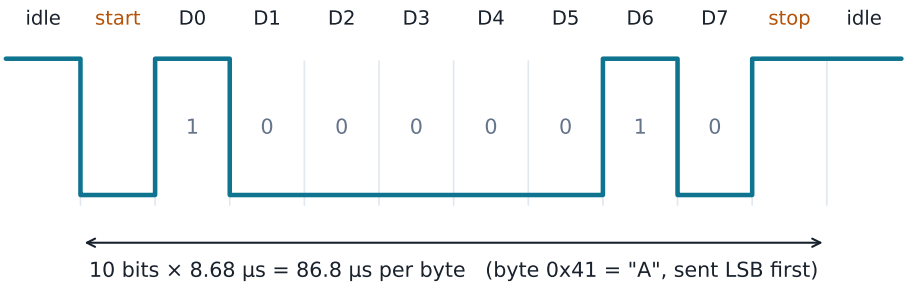
Common mistakes. An ISR that forgets to clear its interrupt flag – the CPU returns and immediately re-enters, forever. Long ISRs (do the minimum, set a flag, let the main loop work). And racing the timer: reading a 16-bit timer on an 8-bit bus needs the high byte read twice or latched, or a carry between reads corrupts the value.

Summary. Interrupt = vector + short ISR + flag clear; timer = $(PSC+1)(ARR+1)$ against the clock, with both stored values inside their registers. Interrupts win at low event rates; polling wins past ~1 MHz.

Serial protocols

Why it matters. The node's readings leave through three wires or fewer. Choosing UART, SPI or I2C – and pricing the wire against the speed – is a weekly decision in embedded work.

The idea, with a picture. **UART** sends asynchronous bytes: no shared clock, just a start bit, 8 data bits, a stop bit, at a **baud rate** both ends agreed on. **SPI** shares a clock: the master drives SCLK and chip-select, data flows on MOSI out and MISO in – full duplex. **I2C** needs two wires (SDA data, SCL clock), addresses devices with 7 bits, and wires them open-drain so any device can hold the line low.



An 8N1 frame: 10 bit times per byte, 86.8 µs at 115 200 baud

The derivation. UART has no clock line, so both ends reconstruct time from the agreed baud: a byte in 8N1 framing occupies exactly 10 bit times,

exercises.lol exercises.com.cn

$$T_{\text{byte}} = \frac{10}{\text{baud}}, \qquad \text{throughput} = \frac{\text{baud}}{10}\ \text{bytes/s}.$$

SPI instead sends one bit per complete clock cycle, so a byte takes 8 clocks plus setup: $T_{\text{byte}} \approx 8/f_{\text{SCLK}}$ – the clock is the speed.

> **Worked example 1 – sizing UART.** At 115 200 baud, 8N1: throughput is $115\,200/10 = 11\,520$ bytes/s; one byte takes $10/115\,200 = 86.8$ µs. Against the node's 1 ms control loop: 11 bytes just fit (0.95 ms) ✓; a blocking 100-byte burst (8.68 ms) blows the budget – split it or raise the baud. The sizing decides, not preference.

> **Worked example 2 – the three-way price.** SPI at 10 MHz clocks a byte in $8/10\,\text{MHz} = 0.8$ µs – about 108× faster than UART here – but costs a chip-select wire per device. I2C at 400 kHz needs 9 clocks for 8 data bits plus ACK: $9/400\,\text{kHz} = 22.5$ µs, before address, start and stop overhead. At these chosen rates it lies between SPI and UART, but two wires serve the whole board (7-bit address = up to 112 devices at 1 address per device). Sensor on I2C, flash on SPI, console on UART – the standard split, each priced by this arithmetic.

Common mistakes. Mismatched baud (the frame still 'works' and the bytes are garbage – check the divisor math against the peripheral clock, not the core clock). Reading an I2C address as 8 bits: the 7-bit address plus the read/write bit packs into one byte, and datasheets quote both forms. And forgetting SPI mode (clock polarity and phase): the four modes exist because devices differ; the wrong one clocks every byte one bit off.

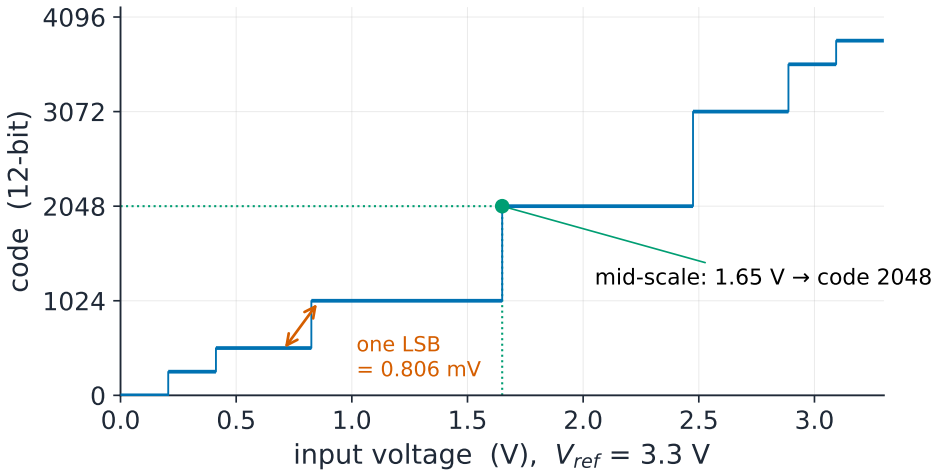
Summary. UART: 10 bit times per byte, no clock, both ends agree the baud. SPI: one bit per clock, a wire per device. I2C: two wires, 7-bit addresses, open-drain. Price bytes-per-millisecond against wires before choosing.

ADC, DAC and sensors

Why it matters. The physical world speaks volts; the CPU speaks numbers. The node's temperature reading is only as good as the converter and its wiring – and the errors here are quiet, not loud.

The idea, with a picture. An **ADC** quantises: an N-bit converter spanning a reference voltage V_{ref} has a **resolution** of one **LSB** $= V_{\text{ref}}/2^N$. The choice of N is the choice of how finely the world is measured; the wiring decides whether that fineness survives.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn



Each code owns a one-LSB band of input voltage; mid-scale is code 2048

The derivation. Quantisation maps a voltage to the code whose band contains it: code $= \lfloor V_{\text{in}}/V_{\text{ref}} \times 2^N \rfloor$ (with variants at the edges). Reconstructing each input band at its midpoint gives an ideal quantisation-error bound of half an LSB within the input range; an individual input can have zero error.

> **Worked example 1 – the 12-bit sensor.** With $V_{\text{ref}} = 3.3$ V, one LSB is $3.3/4096 = 0.806$ mV. A 1.65 V input reads $1.65/3.3 \times 4096 = 2048$ – mid-scale exactly ✓. If the node must resolve 0.1 °C and the sensor gives 10 mV/°C, it needs 1 mV of resolution – a 12-bit converter (0.806 mV) passes, a 10-bit one (3.22 mV) fails: the requirement, not the catalogue, picks N.

> **Worked example 2 – the source-impedance trap.** The ADC's sample-and-hold capacitor (≈ 7 pF on this chip) charges through the source resistance, and settling to ½ LSB needs a sampling time of at least $R_{\text{source}} C \ln(2^{N+1})$. For a 50 kΩ source that is $50\,000 \times 7\,\text{pF} \times 9.01 \approx 3.2$ µs for a full-scale input jump in this simple RC model. Shrink the window to 1 µs and the same bound allows only 16 kΩ. A 100 kΩ thermistor divider (50 kΩ Thevenin) therefore needs at least a 3.2 µs window – or halve the divider values. A **DAC** walks the same staircase in reverse: writing 2048 to a 12-bit DAC with $V_{\text{ref}} = 3.3$ V produces 1.650 V.

Common mistakes. Quoting resolution in bits and forgetting the reference (0.806 mV is only true at 3.3 V). Treating the ADC input as a voltmeter: its sampling current loads high-impedance sensors, as the RC bound above shows. And averaging to 'buy' bits – averaging 16 samples of white noise gains about 2 bits of effective resolution, but it cannot repair a systematic offset or a drooped sample.

Summary. One LSB $= V_{\text{ref}}/2^N$; the requirement picks N. Midpoint reconstruction

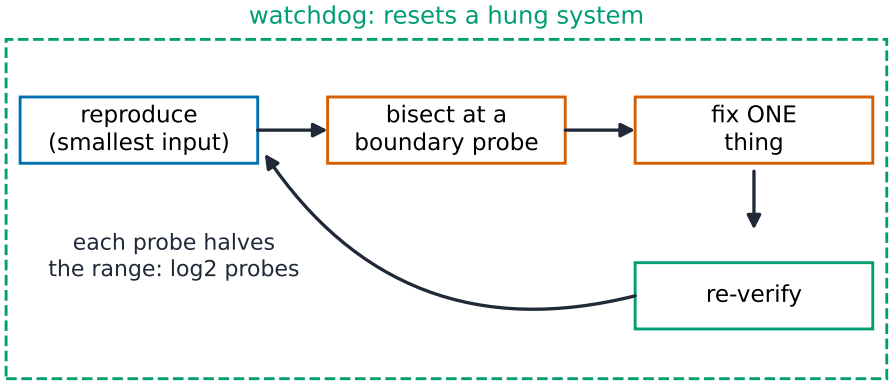
exercises.lol exercises.com.cn

gives a half-LSB quantisation-error bound; the RC bound $t_s \geq RC\ln(2^{N+1})$ trades source impedance against sampling time – the allowed source impedance depends on the model and acquisition window.

Debugging and integration

Why it matters. Firmware is tested by running it on the exact hardware it will ship on, and every bug arrives as a rumour ("it froze twice last night"). Method is the difference between fixing it and renaming it.

The idea, with a picture. The loop is: reproduce with the smallest input; bisect the boundary between trusted and suspected code (an LED, a breakpoint, or one UART print); fix ONE thing; re-verify. A **watchdog** is the safety net that resets a hung system – kick the counter before it expires, or the product reboots by design instead of freezing by accident.



Reproduce, bisect, fix one thing, re-verify – and a watchdog outside the loop

The derivation. Bisection is the algorithm: if the fault reproduces with the boundary probe placed at the midpoint of the suspected range, the bug lives in the first half; else in the second. Each probe halves the range, so $\log_2$(suspicious instructions) probes suffice – ten probes for a thousand instructions. The same halving that runs the ADC ladder runs the debug loop.

> **Worked example 1 – debounce as a state trace.** The node's button reads 1, 0, 0, 1 across four 10 ms samples (a bouncing contact). Rule: require two consecutive equal readings. Trace: t = 0 ms reads 1 (no verdict); t = 10 ms reads 0 (one, no verdict); t = 20 ms reads 0 again → stable 0 declared. The 1 at t = 30 ms is one sample, not yet stable, and is correctly ignored. **Debounce** = N consecutive samples = a timer tick + a GPIO read + a state decision: the whole course in one trace.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

> **Worked example 2 – bisecting the freeze.** The node freezes within an hour of a specific log sequence; the suspected region is 1 024 instructions of the comms driver. Make instruction 512 light the LED: if the node freezes with the LED still dark, the bug is in the first half; if the LED lights before the freeze, it is in the second. Each run halves the region: 512 → 256 → 128 → … – ten runs bound it to one instruction. The arithmetic says why bisection beats rereading the code hoping to see it: $2^{10} = 1024$.

Common mistakes. Fixing two things at once (one of them worked, one broke something else – now there are three bugs). Debugging on a different board revision than the failing one. And disabling the watchdog 'for now': the field failure the watchdog would have survived is exactly the one that ships.

Summary. Reproduce, bisect, fix one, re-verify; $\log_2$ probes bound the hunt. Debounce turns noise into events by N consecutive samples. The watchdog is the last line: kick it in the main loop only, never in an ISR.

exercises.lol exercises.com.cn

Electrical Machines and Power Systems

University Electrical Engineering

What this course is, and how to use it

This is the ninth course in University Electrical Engineering: **electrical machines and power systems** . EE07 wrote the field equations; this course puts them to work – the machines that convert mechanical and electrical energy, and the networks that move it.

- Unit 9.1 builds the magnetic circuit and the **transformer** .
- Units 9.2–9.4 cover the three machine families: DC, induction, synchronous.
- Units 9.5–9.6 widen to the network: per-unit arithmetic and power flow.
- Unit 9.7 opens power electronics.

The running story is one small workshop: a 10 kVA transformer feeds a distribution board; an induction motor drives a compressor; a synchronous generator stands behind the grid; a buck converter charges the battery backup. Every machine answer below is checked through a second route – loss accounting, the equivalent circuit at a limit, or the power balance – and every per-unit answer by converting back to real units.

Magnetic circuits and transformers

Why it matters. The charger that fills your phone in an hour contains a transformer smaller than a coin, and it works because magnetic flux, like current, flows in loops through materials that resist it. The same picture, scaled up a million times, moves power between continents.

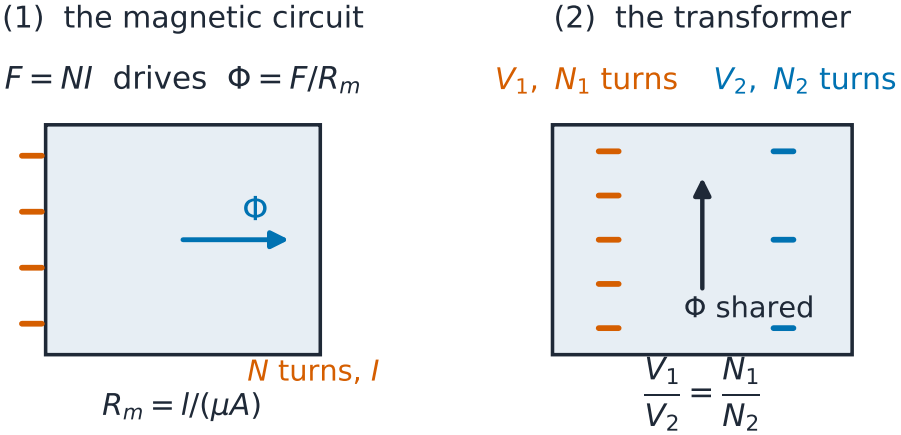
The idea, with a picture. A coil of N turns carrying current I is a pump for **flux** : it creates **magnetomotive force** $F = NI$ (ampere-turns). Flux Φ (weber) flows around an iron loop of length l and cross-section A, opposed by **reluctance** $R_m = \dfrac{l}{\mu A}$, where $\mu = \mu_0 \mu_r$ is the permeability. This is the magnetic mirror of Ohm's law:

$$\Phi = \frac{F}{R_m}, \qquad \text{just as} \qquad I = \frac{V}{R}.$$

The flux density $B = \Phi / A$ (tesla) is the quantity iron cares about – power-frequency cores work between about 1.0 and 1.5 T.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn



One flux loop, two coils: the transformer is just a magnetic circuit shared by two windings

The derivation. Ampère's law around the core gives $H\,l = NI$, with H the field intensity in A/m. With $B = \mu H$ and $\Phi = BA$:

$$\Phi = \mu A \cdot \frac{NI}{l} = \frac{NI}{l/(\mu A)} = \frac{F}{R_m}.$$

Nothing deeper – the whole unit is bookkeeping on this line, plus one warning: air has $\mu_r = 1$, four thousand times weaker than the core below, so a tiny air gap can dominate the reluctance of the entire loop.

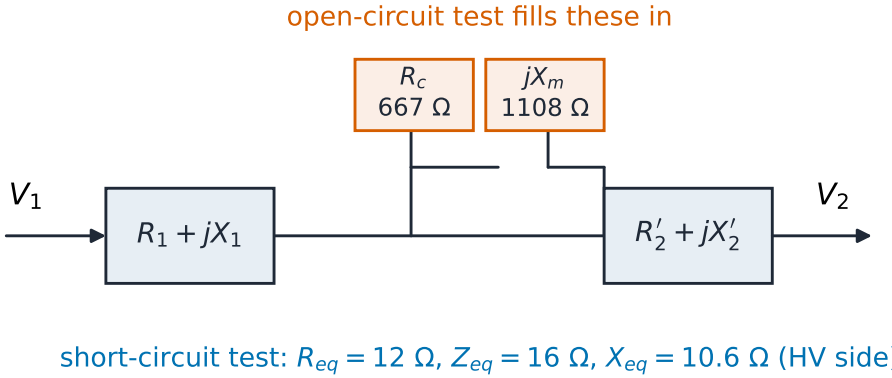
> **Worked example 1 – a core on its own.** A core of mean length 0.5 m, cross-section 10 cm^2 $= 10^{-3}$ m^2, $\mu_r = 4000$, wound with 200 turns at 0.5 A. The magnetomotive force is $F = 200 \times 0.5 = 100$ At. The reluctance is $R_m = \dfrac{0.5}{4\pi \times 10^{-7} \times 4000 \times 10^{-3}} = 99\,472$ At/Wb, so the flux is $\Phi = \dfrac{100}{99\,472} = 1.005$ mWb and the density $B = \dfrac{1.005 \times 10^{-3}}{10^{-3}} = 1.005$ T. Second route: $H = NI/l = 200$ A/m, so $B = \mu_0 \mu_r H = 4\pi \times 10^{-7} \times 4000 \times 200 = 1.005$ T ✓ – a sensible power-frequency density, which is itself the sanity check that the arithmetic stayed in range.

Now cut a 0.5 mm gap in the same core. The gap adds $R_{gap} = \dfrac{0.5 \times 10^{-3}}{4\pi \times 10^{-7} \times 10^{-3}} = 397\,887$ At/Wb – four times the whole core's reluctance. Flux collapses to $\dfrac{100}{497\,359} = 0.201$ mWb, one fifth, from a gap a hair thick. This is why transformer cores are stacked and laminated tight, and why gapped cores must be designed on purpose.

exercises.lol exercises.com.cn

Worked example 2 – the transformer, measured. A transformer links two windings through one shared flux, so ideally $\frac{V_1}{V_2} = \frac{N_1}{N_2}$ and $\frac{I_1}{I_2} = \frac{N_2}{N_1}$: volts trade against amps, power stays. Real ones lose a little in the winding resistance (**copper loss** I^2R) and in the iron (**core loss** : hysteresis plus **eddy currents**). The two losses are separated by two tests:

- **Open-circuit test** (LV side, rated voltage): 400 V draws 0.7 A and 240 W. The 240 W is core loss. Its loss current is $I_c = 240/400 = 0.6$ A, so the magnetising branch is $R_c = 400/0.6 = 667\ \Omega$ and, from $I_m = \sqrt{0.7^2 - 0.6^2} = 0.361$ A, $X_m = 400/0.361 = 1108\ \Omega$ (both referred to the LV side).
- **Short-circuit test** (HV side, reduced voltage, rated current): 80 V draws the rated 5 A (= 10 000/2000) and 300 W. That 300 W is the full-load copper loss, and the series branch is $R_{eq} = 300/5^2 = 12\ \Omega$, $Z_{eq} = 80/5 = 16\ \Omega$, $X_{eq} = \sqrt{16^2 - 12^2} = 10.6\ \Omega$ (HV side).



Two tests, two branches: the open circuit fills in the shunt pair, the short circuit the series pair

Full-load efficiency at unity power factor: $\eta = \frac{10\,000}{10\,000 + 240 + 300} = 94.9\%$. At 0.8 power factor the output is 8000 W with the same losses, so $\eta = \frac{8000}{8540} = 93.7\%$ – fixed core loss hurts more as the useful power falls. Efficiency peaks where the load-dependent copper loss equals the fixed core loss: $x = \sqrt{240/300} = 0.894$ of full load, a number the workshop should recognise on its meter.

Common mistakes. Forgetting μ_0 inside μ (a factor of nearly a million, since $1/\mu_0 \approx 8 \times 10^5$). Referring OC values to the HV side or SC values to the LV side without squaring the turns ratio. Expecting the ideal-transformer current ratio to survive heavy loading – the series drop $I \times Z_{eq}$ is exactly what bends it. And treating an air gap as a small detail: the example above shows a hair-thick gap is four fifths of the story.

Summary. $F = NI$ drives $\Phi = F/R_m$ through reluctance $l/(\mu A)$; density $B = \Phi/A$ must stay under about 1.5 T. Two tests on a real transformer separate shunt (core, from the open circuit) from series (copper, from the short circuit), and efficiency balances output against both.

DC machines

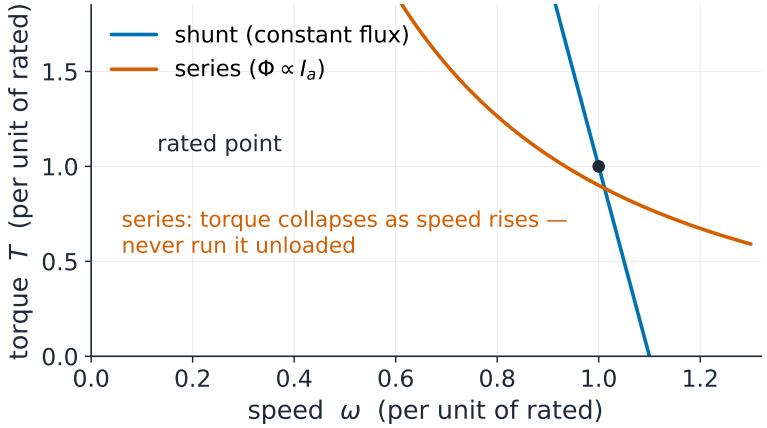
Why it matters. The motor in a cordless drill, an electric train's axle motor and the starter in every car with a combustion engine are DC machines: torque you can command with a current, from two lines of physics.

The idea, with a picture. A conductor of length l in field B carrying current I feels force $F = BIl$; the same conductor moving at speed v induces EMF $e = Blv$. A DC machine wires many such conductors around a rotor (the **armature**) and commutates them so each sees a fixed field direction – the **commutator** is a mechanical switch that keeps the torque unidirectional. Collect the bookkeeping into one constant $k\Phi$ and the whole machine is two equations:

$$E = k\Phi\omega \quad \text{(EMF)}, \qquad T = k\Phi I_a \quad \text{(torque)},$$

with $k\Phi$ set by the field flux alone. Multiplying them, $T\omega = EI_a$: the same air-gap power seen from both sides. Mechanical and electrical language are one language.

The field connection chooses the personality. A **shunt motor** (field in parallel, constant flux) holds nearly constant speed; a **series motor** (field in series, flux $\propto I_a$) makes torque $\propto I_a^2$ and must never run unloaded.



Shunt: a flat speed for any load. Series: huge torque at low speed, runaway risk at no load

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

The derivation. From $e = Blv$ with $v = \omega r$ and the geometry bundled into k: $E = k\Phi\omega$. From $F = BIl$ with the lever arms bundled the same way: $T = k\Phi I_a$. In a shunt machine Φ is fixed by the supply, so speed follows $\omega = \dfrac{V - I_a R_a}{k\Phi}$: the armature resistance drop is the only sag. In a series machine $\Phi \propto I_a$ (until iron saturates), so $T \propto I_a^2$ and $\omega \propto \dfrac{V - I_a R_a}{I_a}$ – speed rises sharply as load disappears.

> **Worked example 1 – the rating point.** A shunt motor on 220 V with armature resistance $R_a = 0.2\ \Omega$ draws $I_a = 50$ A at 1500 rpm. The EMF is $E = 220 - 50 \times 0.2 = 210$ V. At 1500 rpm $= 157.1$ rad/s, $k\Phi = \dfrac{210}{157.1} = 1.337$ V · s, so torque $T = 1.337 \times 50 = 66.9$ N · m. Check both routes through the air gap: $T\omega = 66.9 \times 157.1 = 10.5$ kW and $EI_a = 210 \times 50 = 10.5$ kW ✓ – they must agree; they are the same conversion.

Starting the same motor from rest is the trap: at switch-on $E = 0$, so the inrush would be $I_a = \dfrac{220}{0.2} = 1100$ A, twenty times rated. A starter resistance limiting current to 100 A needs $R_{start} = \dfrac{220}{100} - 0.2 = 2.0\ \Omega$, cut out in steps as E builds.

> **Worked example 2 – field weakening.** Keep the load torque of 66.9 N · m and cut the field flux 20%, to $k\Phi = 1.070$. Torque demands $I_a = \dfrac{66.9}{1.070} = 62.5$ A, so the EMF settles at $E = 220 - 62.5 \times 0.2 = 207.5$ V and the speed at $\omega = \dfrac{207.5}{1.070} = 193.9$ rad/s $= 1852$ rpm – about 23% faster for one fifth less flux. Power check: $T\omega = 66.9 \times 193.9 = 12.96$ kW against $EI_a = 207.5 \times 62.5 = 12.97$ kW ✓. Above base speed this is the standard trick – **field weakening** : trade flux for speed, pay in armature current.

Common mistakes. Feeding rpm into $E = k\Phi\omega$ (it wants rad/s – a factor of 2π). Expecting a shunt motor to speed up under load (the drop across R_a makes it slow, slightly). Running a series motor unloaded – with no current there is no flux to limit the speed. And forgetting that at standstill the only thing limiting current is resistance.

Summary. $E = k\Phi\omega$ and $T = k\Phi I_a$, joined by $T\omega = EI_a$. Shunt wiring holds speed; series wiring multiplies torque at low speed. Start through resistance, weaken field for range.

Induction machines

Why it matters. Around seven of every ten industrial motors are **squirrel-cage** induction machines: no brushes, no commutator, one moving part – because the rotating field does the switching for free.

The idea, with a picture. Three coils spaced 120° apart, fed by three currents spaced 120° apart in time, add to a **rotating field** of constant magnitude that rotates at **synchronous speed** $n_s = \dfrac{120f}{p}$ rpm (f in Hz, p poles). The rotor chases it and never quite catches up; the shortfall is **slip**

$$s = \frac{n_s - n}{n_s}.$$

Slip is the gear ratio of the machine: the rotor bars see the field slip past at rate s, so the rotor frequency is sf and every rotor quantity follows from it.

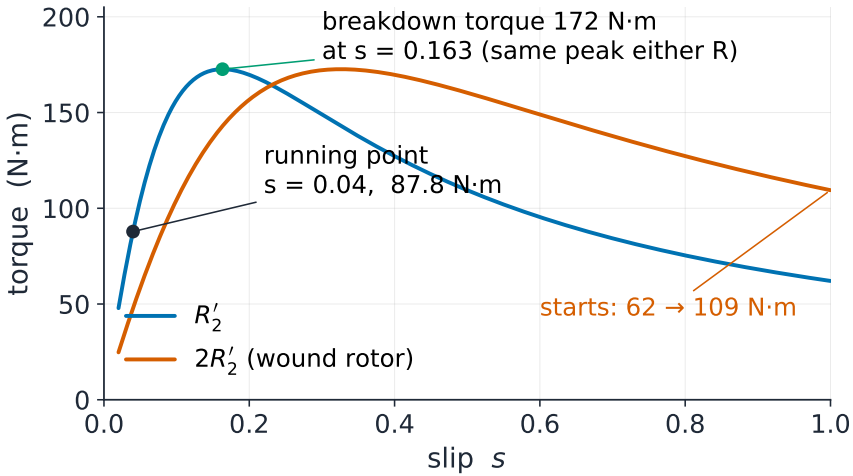
The derivation. In the equivalent circuit, all quantities are referred to the stator and the rotor branch appears as R_2'/s – the slip-scaled resistance. Split it:

$$\frac{R_2'}{s} = R_2' + R_2'\,\frac{1-s}{s},$$

and the second term is not a resistance at all: multiplied by I^2 it is exactly the mechanical output power, while $I^2 R_2'$ is the rotor copper loss. Slip divides the air-gap power $P_{ag} = T\omega_s$ into copper loss sP_{ag} and shaft power $(1-s)P_{ag}$. With a Thevenin source V_{th}, $Z_{th} = R_{th} + jX_{th}$ behind the magnetising branch, torque is

$$T = \frac{3V_{th}^2\,(R_2'/s)}{\omega_s\left[(R_{th} + R_2'/s)^2 + (X_{th} + X_2')^2\right]},$$

peaking (differentiate, set to zero) where $R_2'/s_{max} = \sqrt{R_{th}^2 + (X_{th} + X_2')^2}$ – the **breakdown torque** . Extra rotor resistance moves the peak towards starting: the **wound rotor** machine and the double-cage rotor sell exactly this.



Same peak torque, different address: rotor resistance chooses where the curve peaks

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

Worked example 1 – reading the nameplate. A 4-pole, 50 Hz motor: $n_s = \dfrac{120 \times 50}{4} = 1500$ rpm. Running at 1440 rpm: $s = \dfrac{60}{1500} = 0.04$. The rotor frequency is $sf = 2$ Hz – why rotor iron needs only light laminations – and the slip term R_2'/s is $25 \times R_2'$: at small slip the rotor behaves almost purely resistively, which is why the machine runs efficiently near synchronism.

Worked example 2 – the whole torque chain. The same motor, 400 V line (231 V per phase), $R_1 = 0.5$, $X_1 = 1.2$, $R_2' = 0.4$, $X_2' = 1.2\ \Omega$, magnetising branch large (taken out of the way: $V_{th} = 231$ V, $Z_{th} = R_1 + jX_1$). Synchronous speed $\omega_s = 157.1$ rad/s. At $s = 0.04$, $R_2'/s = 10\ \Omega$:

$$Z = 10.5 + j2.4, \qquad |Z|^2 = 116.0, \qquad T = \frac{3 \times 231^2 \times 10}{157.1 \times 116.0} = 87.8\ \text{N m}.$$

Follow the power: $P_{ag} = T\omega_s = 13.8$ kW, rotor copper loss $sP_{ag} = 552$ W, shaft power $T\omega_s(1-s) = 13.2$ kW ✓ – one chain of multiplications, three balances closed. The peak: $s_{max} = \dfrac{0.4}{\sqrt{0.5^2 + 2.4^2}} = 0.163$, and the breakdown torque there is 172 N · m – twice the running point, the margin the compressor start needs. Locked rotor ($s = 1$): $T_{start} = 62$ N · m. Double the rotor resistance and T_{start} rises to 109 N · m while the running point (same slip) loses efficiency – the wound-rotor trade in one number.

Common mistakes. Mixing rpm and rad/s inside the torque formula (ω_s is mechanical, rad/s). Assuming torque rises with slip – it peaks at s_{max} and falls both sides. Charging the whole air-gap power to copper loss: copper gets only the fraction s. And expecting the rotor frequency to be 50 Hz – it is sf, 2 Hz here.

Summary. $n_s = 120f/p$, slip $s = (n_s - n)/n_s$, rotor frequency sf, rotor branch R_2'/s whose second half is the shaft. Torque peaks where R_2'/s matches the magnitude of $R_{th} + j(X_{th} + X_2')$; rotor resistance buys starting torque at running cost.

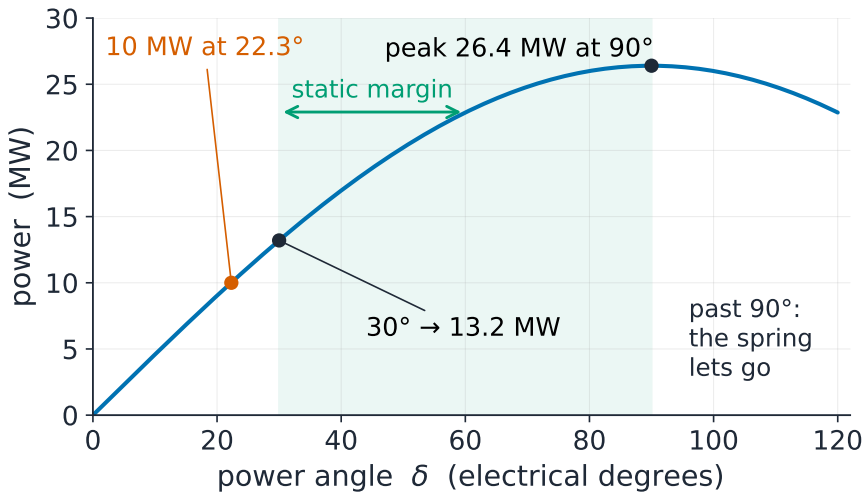
Synchronous machines

Why it matters. Virtually every watt from a power station flows out of a **synchronous machine** : a magnet spun by a turbine, locked to the grid's rotating field, delivering power through an angle.

The idea, with a picture. The rotor – an electromagnet or permanent magnet – locks to the rotating field, so speed is exactly $n_s = 120f/p$, no slip. The induced EMF has the transformer form $E = 4.44\, f N \Phi k_w$ (the winding factor k_w tidies the coil geometry). What the machine exchanges with the grid is set by the **power angle** δ: the angle by which the rotor's EMF leads the bus voltage. The connection behaves like a torsion spring – deflect it, feel torque:

$$P = \frac{3VE}{X_s} \sin\delta.$$

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn



A sine-shaped spring: power grows with angle to 90 degrees, then lets go – that is loss of synchronism

The derivation. Per phase, the bus is $V\angle 0$, the internal EMF $E\angle\delta$, between them only the synchronous reactance X_s:

$$I = \frac{E\angle\delta - V}{jX_s}, \qquad P = 3\,\mathrm{Re}(VI^*) = \frac{3VE}{X_s}\sin\delta, \qquad Q = \frac{3V(E\cos\delta - V)}{X_s}.$$

The Q line carries the second lesson: raising the field (**excitation**) raises E and pushes Q positive – the machine exports reactive power like a capacitor; lowering it imports. Grid operators schedule this daily.

Worked example 1 – the rating and its margin. A generator on an **infinite bus** at 11 kV line, $X_s = 5\ \Omega$ per phase, EMF 12 kV line. Per phase $V = 6351$ V, $E = 6928$ V, so

$$P = \frac{3 \times 6351 \times 6928}{5} \sin\delta = 26.4 \sin\delta\ \text{MW}.$$

At $\delta = 30°$: $P = 13.2$ MW. The **static margin** is $\dfrac{26.4 - 13.2}{13.2} = 100\%$ – the machine could double its load before the spring lets go. Operators keep real machines nearer 30° for exactly this cushion.

Worked example 2 – real and reactive together. Same machine, schedule 10 MW: $\sin\delta = \dfrac{10}{26.4} = 0.379$, $\delta = 22.3°$. Then

$$Q = \frac{3 \times 6351 \times (6928\cos 22.3° - 6351)}{5} = 0.23\ \text{MVAr}.$$

exercises.lol exercises.com.cn

Over-excited, it quietly supports the bus voltage while delivering the megawatts. Under-excite to $E = 10$ kV line (5774 V per phase) and hold 10 MW: $\sin\delta = \dfrac{10\times5\times10^6}{3\times6351\times5774} = 0.455$, $\delta = 27.0°$, and $Q = \dfrac{3\times6351\times(5774\times0.890-6351)}{5} = -4.6$ MVAr: the same megawatts, but the machine now absorbs reactive power. One dial (excitation) moves Q far more than P.

Common mistakes. Using line voltages per phase (divide by $\sqrt{3}$ first). Believing more excitation gives more real power – the turbine sets P; excitation sets Q. Pushing δ past 90°, where $\dfrac{dP}{d\delta}$ goes negative and the machine slips a pole. And forgetting that δ here is electrical: with p poles the mechanical angle is $2\delta/p$.

Summary. Speed locked to the grid, power $P = \dfrac{3VE}{X_s}\sin\delta$, reactive $Q = \dfrac{3V(E\cos\delta - V)}{X_s}$. Turbine owns the angle, field owns the VARs.

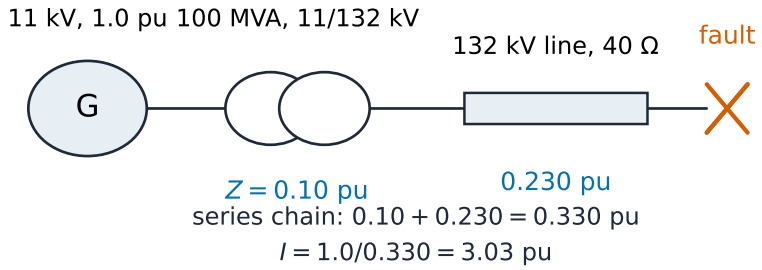
Power networks and the per-unit system

Why it matters. A grid calculation mixing ohms at 11 kV with ohms at 132 kV drowns in turns ratios. Power engineers long ago switched to a language where every voltage is near 1.0 and every impedance is a percentage: the **per-unit** system.

The idea, with a picture. Choose a **base value** pair: base power S_{base} and base voltage V_{base} at each level; everything else follows:

$$Z_{base} = \frac{V_{base}^2}{S_{base}}, \qquad I_{base} = \frac{S_{base}}{\sqrt{3}\,V_{base}}, \qquad Z_{pu} = \frac{Z_{actual}}{Z_{base}}.$$

Impedances move between voltage levels by the turns ratio squared – per-unit absorbs that automatically, because each level has its own Z_{base}. A transformer's nameplate "10%" means: at its own rated current, 10% of rated voltage drops inside it – which is exactly $Z_{pu} = 0.10$.



One line, one chain of per-unit impedances: the fault sees them all in series

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

The derivation. Base changes are pure bookkeeping. An impedance given as Z_{old} pu on S_{old}, V_{old} becomes, on new bases,

$$Z_{new} = Z_{old}\times\frac{S_{new}}{S_{old}}\times\left(\frac{V_{old}}{V_{new}}\right)^2,$$

and if the voltage bases already match the transformer ratio (as they do in a consistent study), only the S ratio survives.

Worked example 1 – a fault at the end of the line. A 100 MVA, 11/132 kV transformer with $Z = 10\%$ on its own base feeds a 132 kV line of 40 Ω; a three-phase fault closes at the line's far end with the generator held at 1.0 pu. Line base: $Z_{base} = \dfrac{132^2}{100} = 174.2$ Ω, so the line is $\dfrac{40}{174.2} = 0.230$ pu; the transformer is already 0.10 pu on 100 MVA. Total 0.330 pu, so the **fault current** is $\dfrac{1.0}{0.330} = 3.03$ pu. Back to real units – the check that proves the method: base current at 132 kV is $I_{base} = \dfrac{100\times10^6}{\sqrt{3}\times132\,000} = 437$ A, so $I_{fault} = 3.03\times437 = 1327$ A. The same 3.03 pu read on the 11 kV side with that level's base current $\dfrac{100\times10^6}{\sqrt{3}\times11\,000} = 5249$ A gives $3.03\times5249 = 15\,926$ A – one dimensionless number serves both voltage levels ✓.

Worked example 2 – changing bases. The study grows: add a 60 MVA, 66/11 kV transformer rated 8% and a generator with $X_d = 0.25$ pu on 80 MVA, 11 kV. On the study's 100 MVA base the transformer becomes $0.08\times\dfrac{100}{60} = 0.133$ pu and the generator $0.25\times\dfrac{100}{80} = 0.313$ pu (voltage bases already agree). In series between the generator's EMF and a fault on the transformer's 66 kV terminals: $0.313 + 0.133 = 0.446$ pu, so the fault current is $\dfrac{1.0}{0.446} = 2.24$ pu. On the 66 kV side the base current is $\dfrac{100\times10^6}{\sqrt{3}\times66\,000} = 875$ A, so $2.24\times875 = 1962$ A flows into the fault, while the generator carries $2.24\times5249 = 11\,780$ A on its 11 kV side – the same pu, two real currents. Two multiplications replaced a page of ratio chasing.

Common mistakes. Using line-to-line volts in a per-phase base (or vice versa) – pick the convention $Z_{base} = V_{LL}^2/S_{3\phi}$ and stay with it. Forgetting the square on the voltage-base ratio when changing bases. And treating "%Z" as a loss figure – it is a voltage-drop figure at rated current, which is why a 10% transformer can still be 99% efficient at full load.

Summary. Choose S_{base} once and V_{base} per level; convert everything to pu; series, parallel and fault arithmetic go dimensionless; convert back with each level's own base current to prove the answer.

exercises.lol exercises.com.cn

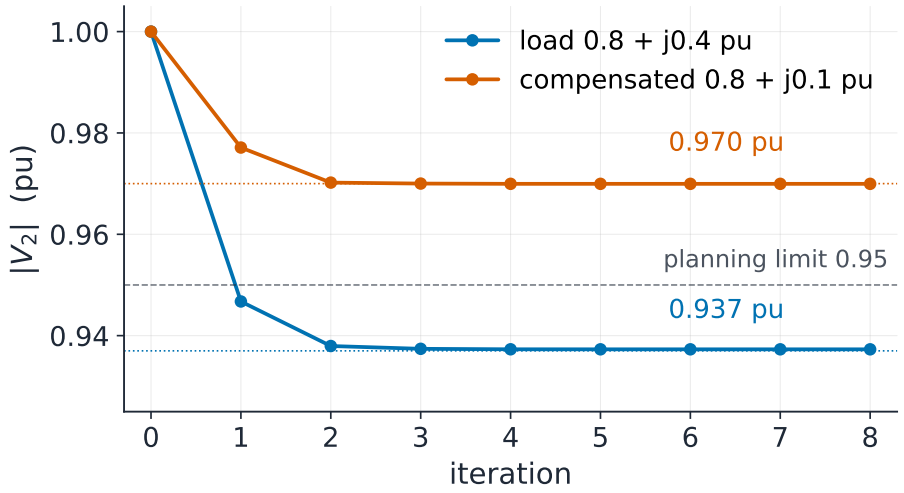
Per-unit and power flow

Why it matters. "Will the voltage at the end of the feeder stay above 0.95?" is the question every grid planner answers thousands of times a day with a **power flow** – and the two-bus version fits on one page.

The idea, with a picture. Bus 1 is the **slack bus** : held at $1.0\angle 0°$ pu, absorbing whatever the losses turn out to be. A line $Z = R + jX$ feeds a load drawing $S = P + jQ$ at bus 2. The load fixes current through its own voltage, and the voltage fixes current through the line – a loop. So guess V_2, run the loop once, and repeat:

$$V_2^{(k+1)} = V_1 - Z \cdot \mathrm{conj}\left(\frac{S}{V_2^{(k)}}\right),$$

the fixed-point (Gauss–Seidel) iteration. The conjugate is not decoration: with $S = VI^*$, the current is $I = \mathrm{conj}(S/V)$, and skipping it sends the iteration to nonsense.



Guess, correct, repeat: both cases settle within a few iterations – and compensation settles higher

The derivation. From $V_2 = V_1 - ZI$ and $I^* = S/V_2$, each pass computes the current the present V_2 implies and the voltage that current produces. Linearise once around $V \approx 1$ and the famous approximation falls out:

$$\Delta V \approx \frac{PR + QX}{|V_2|},$$

whose second term explains the unit's headline: it is the reactive flow through reactance that eats voltage, which is why capacitors – **reactive compensation** – are a voltage tool.

> **Worked example 1 – the base case.** $Z = 0.02 + j0.10$ pu, load $0.8 + j0.4$ pu, $V_1 = 1.0\angle 0°$. Starting from $V_2 = 1.0$ the iteration converges to $V_2 = 0.937\angle -4.4°$ pu, with line current $I = 0.954\angle -31°$ pu. Checks: the approximation gives $\Delta V \approx \dfrac{0.8 \times 0.02 + 0.4 \times 0.10}{0.937} = 0.060$ pu against the exact $1 - 0.937 = 0.063$ ✓ honest at this loading; of the drop, $QX = 0.040$ dwarfs $PR = 0.016$ – reactive dominates. The sending-end balance: $P_s = 0.8 + |I|^2 R = 0.8 + 0.954^2 \times 0.02 = 0.818$ pu ✓ and $Q_s = 0.4 + 0.954^2 \times 0.10 = 0.491$ pu: the line itself eats 0.091 of reactive.

> **Worked example 2 – compensation.** Install capacitors at bus 2 cutting the net load to $0.8 + j0.1$ pu and iterate again: V_2 settles at $0.970\angle -4.6°$ pu. The drop falls from 0.063 to 0.030 pu – the QX term dropped from 0.040 to 0.010 and the voltage followed – while the angle stays near 4.5°, because the angle is carried by P, which did not change. In per-unit the design sentence is one line: 0.3 pu of capacitors at the load lifts V_2 as far as halving the line impedance would (0.970 pu either way), and capacitors cost far less than a second line.

Common mistakes. Dropping the conjugate (the iteration then diverges or converges to a wrong operating point quietly). Letting the slack bus fix more than it should – it sets the angle reference and swallows losses; loads elsewhere must still balance in the converged answer. And trusting $\Delta V \approx (PR + QX)/V$ when the angle across the line grows: the approximation quietly ignores the quadrature component $\dfrac{PX - QR}{V}$, fine here at 4.4°, misleading past about 15°.

Summary. Slack at one end, load at the other, iterate $V_2 = V_1 - Z\,\mathrm{conj}(S/V_2)$; verify with the $PR + QX$ approximation and close the power balance. Reactive flow is the voltage problem; compensation is the cheap half of the fix.

Power electronics introduction

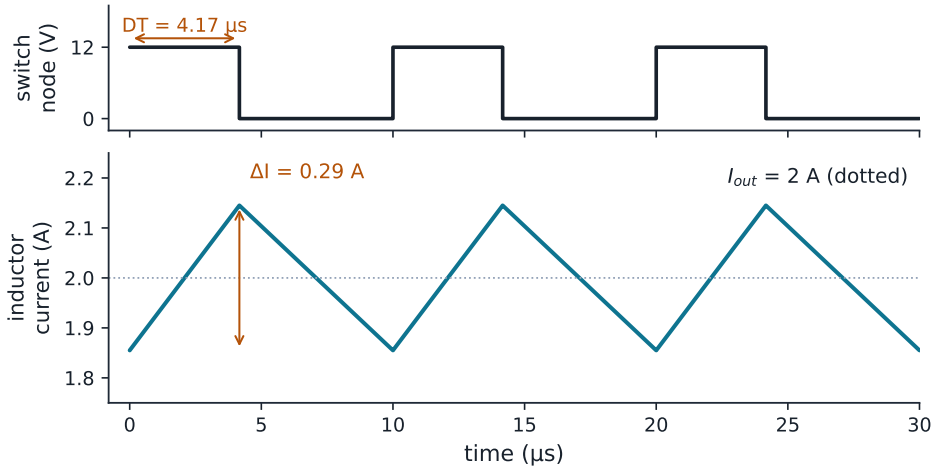
Why it matters. The same charger opens with a switch that flips a hundred thousand times a second. Switching converters are why a 30-watt adapter is now smaller than its own plug – and why this unit ends the course: machines meet electronics at the drive.

The idea, with a picture. A **buck** converter is a switch, a diode, an inductor and a capacitor. While the switch is ON the input feeds the inductor; while it is OFF the inductor keeps current flowing through the diode. The inductor and capacitor do the averaging; the switch only chooses the ratio of on-time to period, the **duty cycle** D. **Volt-second balance** on the inductor – the average of v_L over a period must be zero in steady state – gives, in one line:

$$(V_{in} - V_{out})\,DT = V_{out}\,(1 - D)\,T \quad \Longrightarrow \quad V_{out} = D\,V_{in},$$

exercises.lol exercises.com.cn

and the **boost** dual gives $V_{out} = \dfrac{V_{in}}{1-D}$. **Continuous conduction** (CCM) means the inductor current never reaches zero – the small-ripple world all these formulas live in.



Inductor current ramps up while the switch is on and down while it is off

The derivation. In **continuous conduction**, the buck ON phase the inductor sees $V_{in} - V_{out}$ for DT seconds; in the OFF phase $-V_{out}$ for $(1-D)T$. Setting the volt-second sum to zero yields $V_{out} = DV_{in}$ – independent of L, C and load in CCM, which is the converter's charm. The triangular current swing follows from $v = L\,di/dt$ on either phase:

$$\Delta I_L = \frac{V_{out}(1-D)}{Lf}, \qquad \Delta V_{out} = \frac{\Delta I_L}{8Cf},$$

the ripple pair every datasheet quotes.

> **Worked example 1 – a 12 V to 5 V buck.** Twelve volts in, 5 V at 2 A out, switching at 100 kHz with $L = 100$ µH. Duty $D = \dfrac{5}{12} = 0.417$. Ripple current $\Delta I_L = \dfrac{5 \times 0.583}{100\mu \times 100k} = 0.29$ A – 15% of the load, a standard design point. CCM holds because $\Delta I_L/2 = 0.15$ A is far below 2 A. With $C = 220$ µF the output ripple is $\Delta V_{out} = \dfrac{0.29}{8 \times 220\mu \times 100k} = 1.7$ mV: the converter's output is cleaner than the supply it runs from.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

> **Worked example 2 – a 12 V to 24 V boost.** The boost runs the same parts re-arranged: switch across the inductor, diode to the output. For 24 V at 1.5 A from 12 V, $D = 1 - \dfrac{12}{24} = 0.5$. Power conserved (ideal): $P_{in} = 36$ W, so the input current is 3 A – not 1.5 A; current trades against voltage, power does not. Ripple $\Delta I_L = \dfrac{V_{in}D}{Lf} = \dfrac{12 \times 0.5}{100\mu \times 100k} = 0.6$ A, CCM comfortable against 3 A. The output capacitor sees the diode's chopped current – pulses only during $(1-D)$ of each period – so boost converters need bigger, lower-ESR output capacitors than bucks of the same power: the ripple RMS, not the average, sizes the part.

Common mistakes. Light load in a "CCM" design: below the boundary current $\Delta I_L/2$ the converter drops into discontinuous conduction and V_{out} rises above DV_{in}. Ignoring the diode drop in low-voltage outputs – 0.4 V in a 5 V rail is 8%, which is why **synchronous rectification** (a MOSFET for the diode) is routine below about 12 V. And equating input and output current because the topology looks symmetric – only power is conserved.

Summary. Volt-second balance: $V_{out} = DV_{in}$ (buck) or $V_{out} = V_{in}/(1-D)$ (boost) in CCM; ripples $\Delta I_L = \dfrac{V_{out}(1-D)}{Lf}$ and $\Delta V_{out} = \dfrac{\Delta I_L}{8Cf}$. The switch sets the ratio; the passives set the quality; the losses live in the switch and the diode.

exercises.lol exercises.com.cn

Control Systems

University Electrical Engineering

What this course is, and how to use it

This is the tenth course in University Electrical Engineering: **control systems**. EE06 gave you the system function. This course closes a **feedback** loop around it and answers the engineer's question: how do I make the output follow the input quickly, accurately, and without ringing?

- Units 10.1 and 10.2 build models of the **plant** (the thing being controlled) and reduce them to one **transfer function**.
- Units 10.3 to 10.5 predict and shape the time response: specifications, stability, root locus.
- Units 10.6 and 10.7 work in the frequency domain: margins, Nyquist, PID and compensators.
- Unit 10.8 opens state space: models for many states, and a design method built on linear algebra.

One habit runs through the whole course. Every design is checked in both domains. A phase margin predicts a damping. A pole placement predicts an overshoot. Every algebraic stability claim is confirmed by a second method: Routh against the root locus, Nyquist against Routh. When two roads lead to the same number, you can trust the number.

Modelling of dynamic systems

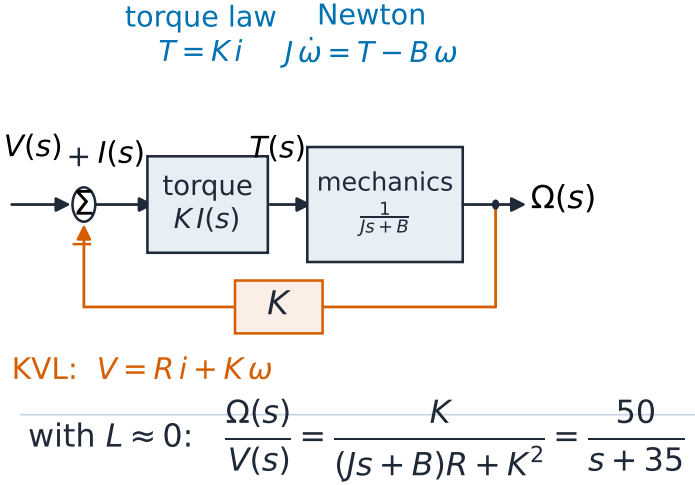
Why it matters. A cruise-control car holds 100 km/h on a hill. A 3D printer places a droplet of plastic within 0.05 mm. Both start the same way: write down what the physical system does, as an equation you can compute with.

The idea. Almost every physical domain stores, flows, and loses energy in the same pattern. A hot tank stores heat, gains heat from a heater, and loses heat to the room. A capacitor stores charge, gains it from a source, and leaks it through a resistor. Written as one **state variable** x, each of them says

$$(\text{how fast } x \text{ changes}) \;=\; (\text{drive in}) \;-\; (\text{leakage out}).$$

A **first-order model** has the form $\tau \dfrac{dx}{dt} + x = K\,u(t)$. The Greek letter τ (tau) is the **time constant**, measured in seconds: it says how long the storage takes to respond. The symbol K is the DC gain: the final value of x per unit of steady input u. With a step input of size U, the solution is $x(t) = KU\,(1 - e^{-t/\tau})$, which reaches 63.2% of its final value at one time constant.

Linearization. Real systems are not straight lines. A pendulum obeys $\ddot{\theta} = -(g/l)\sin\theta$, which has no single gain. The fix is to pick an **operating point** (the state you work near) and replace the curve by its slope there. Near rest, $\sin\theta \approx \theta$, so the pendulum behaves like a spring with **natural frequency** $\omega_n = \sqrt{g/l}$. For a swing of length $l = 0.9$ m, take $g = 9.81$ m/s^2 and get $\omega_n = \sqrt{9.81/0.9} = 3.30$ rad/s, about 0.53 Hz: a slow playground swing. The same idea works for a tank draining at $q = 2\sqrt{h}$ (in m^3/h with h in m): the slope $\dfrac{dq}{dh} = 1/\sqrt{h}$ equals 0.5 near $h = 4$ m, and 1/3 near $h = 9$ m. The linear model is honest only near its operating point, and the slope tells you how honest.



The armature-controlled DC motor: two physical laws, one inner feedback loop

Worked example 1: an RC circuit. Take $R = 2$ kΩ and $C = 100$ µF. The resistor carries the current $\dfrac{v_{in} - v}{R}$ and the capacitor integrates it, so

$$C\frac{dv}{dt} = \frac{v_{in} - v}{R}, \qquad \text{which rearranges to} \qquad RC\frac{dv}{dt} + v = v_{in}.$$

This is the first-order model with $\tau = RC = 2000 \times 100 \times 10^{-6} = 0.2$ s and $K = 1$. Step the input from 0 to 5 V and the capacitor reaches $5(1 - e^{-1}) = 3.16$ V, which is 63.2%, after 0.2 s. Sense check: a bigger R or a bigger C means a slower circuit, and the product RC has units of seconds, so the model is consistent.

Worked example 2: the armature-controlled DC motor. This model carries through the whole course. The motor has rotor inertia $J = 0.01$ kg · m^2, friction $B = 0.1$ N · m · s, motor constant $K = 0.5$ (in N · m/A, numerically the same as V · s/rad), and armature resistance $R = 1$ Ω. The inductance is small, so we neglect it. Two physical laws couple the electrical and mechanical sides:

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

- Torque law: current makes torque, $T = K\,i$, with T in N · m and i in A.
- Newton's law for the rotor: $J\,\dot{\omega} = T - B\,\omega$, where ω is the shaft speed in rad/s.

The coupling runs both ways. The spinning armature also generates a **back-EMF** voltage $K\omega$, so Kirchhoff's voltage law (KVL) gives $V = R\,i + K\omega$. In the s-domain, with zero initial conditions, the two laws become

$$(Js + B)\,\Omega(s) = K\,I(s), \qquad I(s) = \frac{V(s) - K\,\Omega(s)}{R}.$$

Substitute the second equation into the first and collect the Ω terms:

$$\left[(Js + B)R + K^2\right]\Omega(s) = K\,V(s), \qquad \text{so} \qquad \frac{\Omega(s)}{V(s)} = \frac{K}{(Js + B)R + K^2}.$$

Put in the numbers. The denominator is $0.01s + (0.1)(1) + 0.5^2 = 0.01s + 0.35$, so

$$\frac{\Omega(s)}{V(s)} = \frac{0.5}{0.01s + 0.35} = \frac{50}{s + 35}.$$

The model has one pole at $s = -35$ rad/s. The time constant is $1/35 = 28.6$ ms: fast. The DC gain is $50/35 = 1.43$ rad/s of final speed per volt. Sense check at DC: with everything constant the motor equations are pure algebra, and indeed $\dfrac{K}{BR + K^2} = \dfrac{0.5}{0.35} = 1.43$. The whole transient came from three physical constants, and the back-EMF appears in the diagram as a feedback loop the motor builds for free. The two variables here, i and ω, are exactly the state variables that unit 10.8 formalises.

Common mistakes.

- Dropping the back-EMF term $K\omega$ from KVL. The result looks simpler and is badly wrong: $\dfrac{K}{(Js + B)R} = \dfrac{50}{s + 10}$ predicts 5 rad/s per volt instead of 1.43 and a time constant of 100 ms instead of 28.6 ms. The back-EMF's built-in feedback is what makes the real motor fast.
- Using a linearization far from its operating point. The tank slope is 0.5 at $h = 4$ m and 0.33 at $h = 9$ m; one slope cannot serve both.
- Unit slips: J in kg · m^2, B in N · m · s, ω in rad/s. Check that the motor constant has the same value in both laws.

Summary. Storage, flow, and dissipation give $\tau\dot{x} + x = Ku$. Pick the operating point, take the slope, and the world becomes linear near it. The motor model $50/(s + 35)$ is our running example from here on.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Transfer functions and block diagrams

Why it matters. A **block diagram** is how engineers speak about systems: a poster of a car's cruise control is a block diagram, not a page of equations.

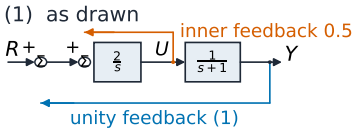
The idea. The transfer function $G(s)$ is the ratio of output to input in the s-domain, with zero initial conditions. When blocks are connected, arithmetic replaces algebra:

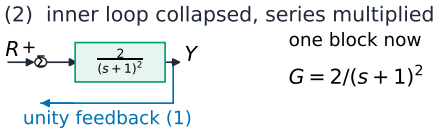
- Series (one block after the other): multiply, $G = G_1 G_2$.
- Parallel (both blocks feed the same sum): add, $G = G_1 + G_2$.
- Feedback: divide by the loop, $T = \dfrac{G}{1 + GH}$.

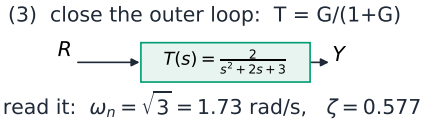
The feedback formula deserves its one-line derivation. The error is $e = r - H\,y$ and the output is $y = G\,e$. Combine them: $y = G(r - Hy)$, move the $GH\,y$ term across, and $y(1 + GH) = Gr$. The equation

$$1 + GH = 0$$

is the **characteristic equation**. Its roots are the closed-loop poles, so it holds the whole stability story. Reduce inner loops first, and keep the denominator visible at every step.







Reduce the inner loop first, multiply the series, close the outer loop last

Worked example 1: one loop, read like a second-order system. Put $G_1 = \dfrac{1}{s + 1}$ and $G_2 = \dfrac{1}{s + 2}$ in series with gain $K = 6$, under unity feedback ($H = 1$):

$$T(s) = \frac{6}{(s + 1)(s + 2) + 6} = \frac{6}{s^2 + 3s + 8}.$$

exercises.lol exercises.com.cn

Now read the second-order parameters by matching against $s^2 + 2\zeta\omega_n s + \omega_n^2$. Here $\omega_n = \sqrt{8} = 2.83$ rad/s and the **damping ratio** $\zeta = \dfrac{3}{2 \times 2.83} = 0.530$. That sits inside the standard design region $0.4 \le \zeta \le 0.7$, and unit 10.3 turns it into an overshoot of about 14%.

> **Worked example 2: nested loops.** An inner loop $\dfrac{2}{s}$ with feedback gain 0.5 sits in series with $\dfrac{1}{s+1}$, all inside a unity-feedback outer loop. Step 1, collapse the inner loop:
>
> $$\frac{2/s}{1 + (2/s)(0.5)} = \frac{2}{s+1}.$$
>
> Step 2, multiply the series blocks:
>
> $$G = \frac{2}{s+1} \cdot \frac{1}{s+1} = \frac{2}{(s+1)^2}.$$
>
> Step 3, close the outer unity loop:
>
> $$T(s) = \frac{2}{(s+1)^2 + 2} = \frac{2}{s^2 + 2s + 3}.$$
>
> Read it: $\omega_n = \sqrt{3} = 1.73$ rad/s and $\zeta = \dfrac{2}{2 \times 1.73} = 0.577$. A tangled diagram became two numbers. For genuinely tangled diagrams, Mason's gain rule does all of this in one formula: sum the forward path gains, subtract the products of loops that do not touch. Reach for it when block algebra genuinely tangles, not before; on simple diagrams the three rules above are faster.

Common mistakes.

- Closing a loop as $\dfrac{G}{1+G}$ when $H \neq 1$. The return path carries GH, not G.
- Losing a minus sign at a feedback summer. The error is $r - Hy$. If you add instead, the characteristic equation becomes $1 - GH = 0$ and every stability claim flips.
- Reducing the outer loop first. It still works, but the algebra explodes; inner first is the discipline.

Summary. Series multiplies, parallel adds, feedback divides by $1 + GH$, and the denominator of the result is the characteristic equation. Inner loops first.

Time-domain response

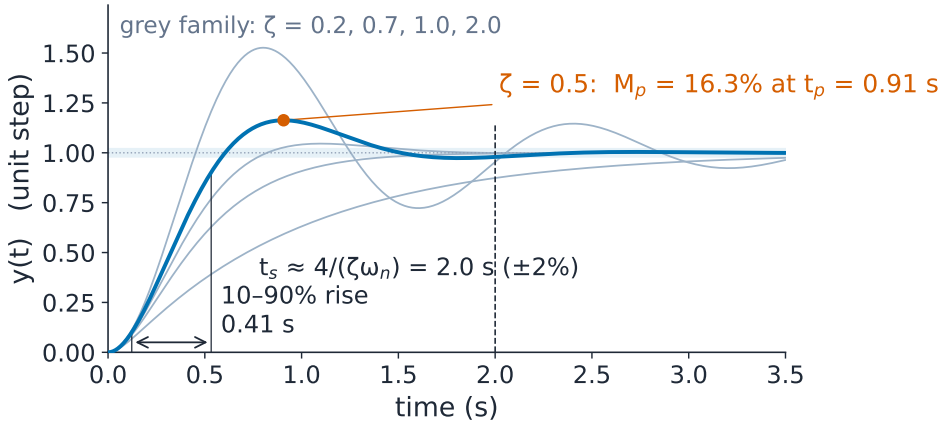
Why it matters. When a drone maker promises "settles within half a second, overshoots less than 10%", those are time-domain numbers. This unit is the dictionary that turns pole positions into those numbers.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

The idea. The second-order prototype carries the vocabulary:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}.$$

Its poles sit at $s = -\zeta\omega_n \pm j\,\omega_n\sqrt{1-\zeta^2}$. Read the picture: the distance from the origin is ω_n (how fast), the imaginary part rings, and the real part $-\zeta\omega_n$ decays. The damping ratio is the cosine of the pole angle, so a ray from the origin at angle $\arccos\zeta$ is a line of constant damping: one picture, one number.



One damping ratio, one shape: the whole family at $\omega_n = 4$ rad/s

The **step response** is the output when the input steps from 0 to 1. With a unit step input, the standard results are

$$M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}}, \qquad t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}, \qquad t_s \approx \frac{4}{\zeta\omega_n} \quad (2\%\text{ band}).$$

Here M_p is the **overshoot** (peak above the final value, as a fraction), t_p the **peak time**, and t_s the **settling time**. The overshoot formula is not magic: the step response is

$$y(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\sin\!\big(\omega_d t + \phi\big), \qquad \omega_d = \omega_n\sqrt{1-\zeta^2},\ \phi = \arccos\zeta,$$

and the first peak comes where the derivative is zero, which lands at $\omega_d t_p = \pi$; putting that back into the envelope gives M_p.

Steady-state error needs one more idea. Count the integrators in the open loop: that count is the **system type**. A type-1 system (one integrator) has zero steady-state error to a step, and its **velocity error constant**

exercises.lol exercises.com.cn

$K_v = \lim_{s\to 0} sG(s)$ gives the ramp error $\dfrac{1}{K_v}$. Read the constants off the loop before computing any transient.

> **Worked example 1: one pair of numbers, four specifications.** Take $\omega_n = 4$ rad/s and $\zeta = 0.5$. Then $\omega_d = 3.46$ rad/s and the poles are $-2 \pm j3.46$. Consistency check: the distance from the origin is $\sqrt{2^2 + 3.46^2} = 4 = \omega_n$, as it must be. The specifications:

$$M_p = e^{-\pi(0.5)/0.866} = e^{-1.814} = 0.163 \quad (16.3\%), \qquad t_p = \frac{\pi}{3.46} = 0.907 \text{ s},$$

$$t_s \approx \frac{4}{0.5 \times 4} = 2.0 \text{ s (2\% band)}, \qquad t_s \approx \frac{3}{2} = 1.5 \text{ s (5\% band)}.$$

The 10 to 90% **rise time** computes numerically from the step response as 0.41 s. Sense check across domains: $\zeta = 0.5$ should come with a phase margin near 50 degrees, and unit 10.6's working rule $\zeta \approx PM/100$ closes that loop.

> **Worked example 2: choose the gain to meet a damping spec.** The plant is $G = \dfrac{K}{s(s+2)}$ in unity feedback. The closed loop is $\dfrac{K}{s^2 + 2s + K}$, so matching the prototype gives $2\zeta\omega_n = 2$ and $\omega_n = \sqrt{K}$, hence $\zeta = 1/\sqrt{K}$. Demand $\zeta = 0.45$:

$$K = \frac{1}{0.45^2} = 4.94, \qquad \omega_n = 2.22 \text{ rad/s}.$$

Predict the response: $M_p = e^{-\pi(0.45)/0.893} = 0.205$, so 20.5%; peak time $t_p = \dfrac{\pi}{2.22 \times 0.893} = 1.58$ s; settling $t_s \approx \dfrac{4}{0.45 \times 2.22} = 4.0$ s. Now the error constants, read off the open loop: $K_v = \dfrac{4.94}{2} = 2.47$, so a unit ramp leaves a steady error $\dfrac{1}{K_v} = 0.405$, while the step error is exactly zero because the plant is type 1. Machine check: the simulated closed-loop step settles to 1.000 and its first peak is 1.205.

Common mistakes.

- Mixing ω_n with ω_d. Peak time uses the damped frequency; distance-from-origin uses ω_n.
- Quoting t_s without its band. The 2% and 5% estimates differ by a factor of $\frac{4}{3}$.
- Applying second-order formulas to a system with zeros or extra poles without checking that one pole pair really dominates.

Summary. Poles in, specifications out: M_p, t_p, t_s, and the error constants. A damping ray on the pole plane is a constant-overshoot line.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Routh stability

Why it matters. Stability is the one property you must have before any other design matters. A loop that rings itself into saturation is not badly tuned; it is broken.

The idea. The closed loop is stable exactly when every root of the characteristic polynomial has a negative real part. The **Routh array** counts those roots without solving anything. Write the coefficients in two rows, then fill downwards. Each new entry is a cross-multiply-and-divide of the two rows above it, and the number of sign changes in the first column equals the number of right-half-plane (RHP) roots. A first column with no sign changes means no RHP roots: stable.

> **Worked example 1: count RHP roots.** Test $s^4 + 2s^3 + 3s^2 + 4s + 5$.

row	first column	second column	third column
s^4	1	3	5
s^3	2	4	0
s^2	$\frac{2\cdot 3 - 1\cdot 4}{2} = 1$	$\frac{2\cdot 5 - 1\cdot 0}{2} = 5$	
s^1	$\frac{1\cdot 4 - 2\cdot 5}{1} = -6$	0	
s^0	5		

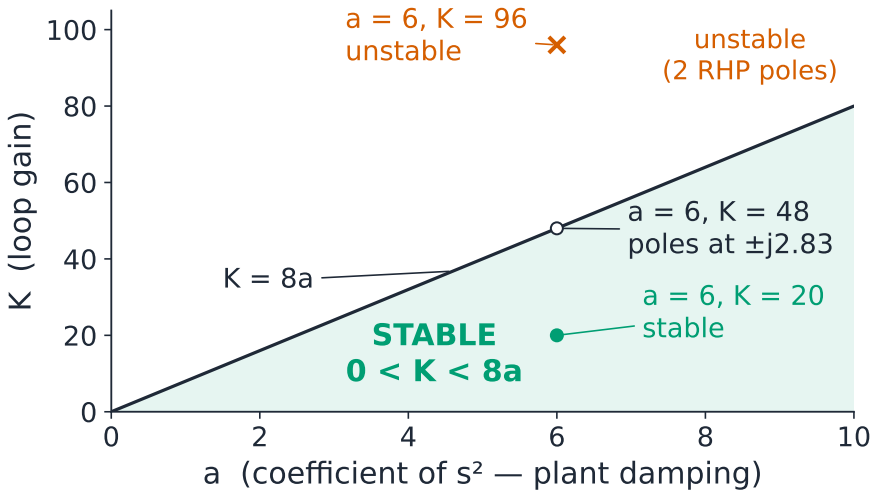
The first column reads 1, 2, 1, −6, 5: two sign changes (+1 to −6, then −6 to +5), so two RHP roots. Check by the exact roots, which are $-1.29 \pm j0.86$ and $+0.29 \pm j1.42$: indeed two in the right half plane, and the array said so without solving the quartic.

> **Worked example 2: find the gain range.** For what K is $s^3 + 5s^2 + 6s + K$ stable? Build the array:

$$s^3 :\ 1,\ 6; \qquad s^2 :\ 5,\ K; \qquad s^1 :\ \frac{5\cdot 6 - K}{5} = \frac{30 - K}{5}; \qquad s^0 :\ K.$$

Stability needs $\dfrac{30 - K}{5} > 0$ and $K > 0$, so $0 < K < 30$. At exactly $K = 30$ the s^1 row vanishes. That is not an error: a whole zero row means symmetric roots, found from the **auxiliary equation** built from the row above, $5s^2 + 30 = 0$, so $s = \pm j2.45$. The system sits on the boundary: **marginal stability**, a sustained oscillation at 2.45 rad/s that neither grows nor dies. The next two units reach this same boundary by two other roads.

exercises.lol exercises.com.cn



The Routh test draws the stable design space before any root is solved

A design picture. For the family $s^3 + as^2 + 8s + K$, the same cross-product gives the s^1 entry $\dfrac{8a - K}{a}$, so stability is exactly $0 < K < 8a$: a straight boundary in the (a, K) plane. The running plant of this course, $s(s+2)(s+4)$, has $a = 6$ and limit $K = 48$: every later unit lands on that same point.

Common mistakes.

- Feeding the array an incomplete polynomial. The coefficients must cover every power from s^n down; a missing power gets coefficient 0.
- Dividing a row by a negative number to simplify. Rescaling by a positive number is free; a negative rescaling flips the very signs you are counting.
- Treating a single zero in the first column as failure. Replace it with a small positive ε and continue; a whole zero row needs the auxiliary equation.
- Calling the $K = 30$ case "stable". Poles on the axis mean marginal, not stable: the oscillation never dies.

Summary. Build the array, count first-column sign changes, and the RHP root count falls out with no root-finding. Parameter ranges read off one row.

Root locus

Why it matters. Gain is the knob every amplifier, every controller, every tuning session turns. The root locus tells you where the closed-loop poles go as you turn it, before you build anything.

The idea. The **root locus** is the set of points s where the angle condition

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

holds for the loop $L(s) = K\,\dfrac{\prod(s - z_i)}{\prod(s - p_i)}$:

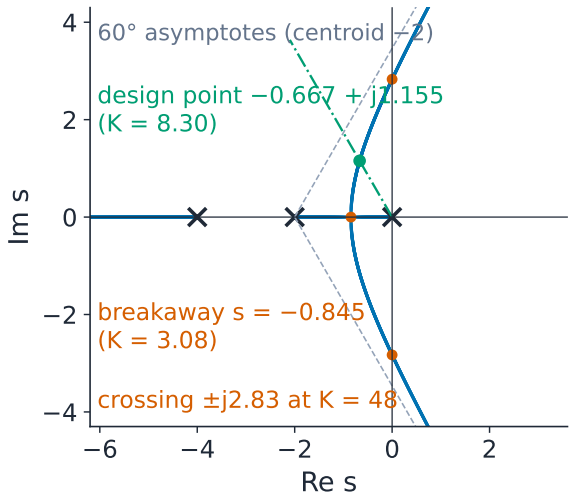
$$\sum_i \angle(s - z_i) \;-\; \sum_i \angle(s - p_i) \;=\; 180^\circ \cdot (2k + 1).$$

Each branch starts at an open-loop pole (at $K = 0$) and ends at a zero or at infinity. The standard sketch rules all follow from that one condition:

- Real-axis segments: a segment carries the locus when an odd number of poles-plus-zeros lies to its right.
- **Asymptote** angles $\dfrac{180^\circ(2k+1)}{n-m}$, centred on the centroid of the poles minus the zeros.
- The **breakaway point**, where branches leave the real axis, solves $\dfrac{d}{ds}\Big[\dfrac{\prod(s-p_i)}{\prod(s-z_i)}\Big] = 0$, and only roots on the locus count.
- The $j\omega$-axis crossing: put $s = j\omega$ into $1 + KL_0 = 0$ and solve the real and imaginary parts.

> **Worked example 1: sketch the running plant.** Take $L = \dfrac{K}{s(s+2)(s+4)}$. Three poles at $0, -2, -4$ and no zeros, so $n - m = 3$ and the asymptotes leave the centroid $\dfrac{(0) + (-2) + (-4)}{3} = -2$ at $\pm 60^\circ$ and 180°. The real-axis locus is the segment $[-2, 0]$ and the half-line $(-\infty, -4]$. The breakaway solves
>
> $$\frac{d}{ds}\big(s^3 + 6s^2 + 8s\big) = 3s^2 + 12s + 8 = 0,$$
>
> giving $s = -0.845$ (on the locus: keep it) and $s = -3.155$ (between -4 and -2, off the locus: discard it). At the breakaway the gain is $K = |s(s+2)(s+4)| = 0.845 \times 1.155 \times 3.155 = 3.08$. The $j\omega$ crossing: with $s = j\omega$, the real part gives $K = 6\omega^2$ and the imaginary part gives $\omega^2 = 8$, so $\omega = 2.83$ rad/s and $K = 48$. The same 48 that the Routh boundary drew.

exercises.lol exercises.com.cn



Poles leave, meet at −0.845, bend to the asymptotes, and cross into instability at K = 48

Worked example 2: design for a damping spec on the locus. Require $\zeta = 0.5$. The poles must sit on the ray at $\arccos 0.5 = 60°$ from the negative real axis, that is on $s = -\sigma + j\sqrt{3}\,\sigma$ for some $\sigma > 0$. Search along the ray for the point that satisfies the angle condition. At $\sigma = 0.667$ the three pole angles are $120° + 40.9° + 19.1° = 180°$ exactly, so the design point is

$$s^* = -0.667 + j\,1.155.$$

The gain comes from the magnitude condition:

$$K = |s^*| \cdot |s^* + 2| \cdot |s^* + 4| = 1.334 \times 1.764 \times 3.527 = 8.30.$$

Check the geometry: $|s^*| = \sqrt{0.667^2 + 1.155^2} = 1.334$ and $\dfrac{0.667}{1.334} = 0.500$, exactly the damping asked for. Check the stability room: $K = 8.30$ sits far below the limit 48, so damping, not stability, is what the ζ-ray buys. The third pole at this gain sits at -4.67, well to the left of the pair, which is why the pair dominates and the second-order predictions hold.

Common mistakes.

- Keeping an off-locus breakaway root. The quadratic gives two answers; only points on the real-axis locus are physical.
- Reading damping as distance from the origin. ζ is the cosine of the pole angle: circles are constant ω_n, rays are constant ζ.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

- Assuming the asymptote crossing is the $j\omega$ crossing. Compute the crossing from $1 + KL_0 = 0$; the asymptote is only a sketch aid.

Summary. Angle condition in, sketch out: segments, centroid, asymptotes, breakaway, crossing. Design means walking the locus to a damping ray and reading off the gain.

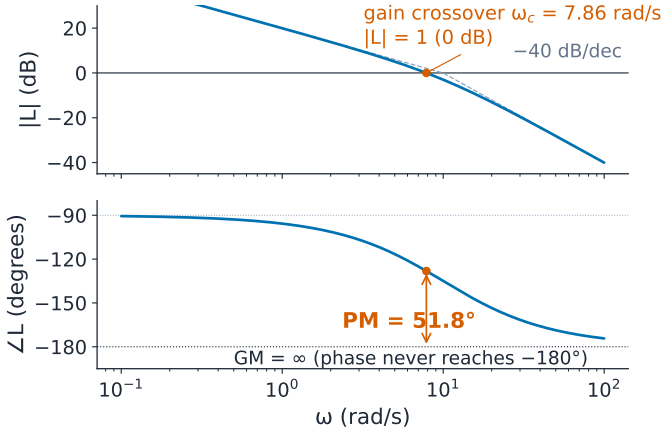
Bode and Nyquist analysis

Why it matters. In industry you rarely know the exact pole positions; you measure a frequency response. Margins are the engineering tolerance stated in that language: how close to instability is this loop, in decibels and degrees?

The idea. A Bode plot draws the loop gain $L(j\omega)$ as magnitude (in dB, against log frequency) and phase (in degrees). Each pole contributes a -20 dB/decade slope and up to $-90°$ of phase; each zero contributes the opposite. Two numbers read off the plot carry the stability information:

- **Phase margin** PM: at the **gain crossover** frequency (where $|L| = 1$), how many degrees the phase sits above $-180°$.
- **Gain margin** GM: at the **phase crossover** (where $\angle L = -180°$), the factor by which $|L|$ is below 1.

The working rule that ties the domains together is $\zeta \approx PM/100$: a 50° margin behaves like $\zeta \approx 0.5$, an overshoot near 16%. The **Nyquist criterion** makes stability counting exact. Plot $L(j\omega)$ for ω from $-\infty$ to $+\infty$, count clockwise **encirclements** N of the point -1, and the closed loop has $Z = N + P$ right-half-plane poles, where P counts open-loop RHP poles. (A pole at the origin, as in our running plant, is indented to the right on the Nyquist contour; its infinite arc stays far from -1 and changes nothing here.)



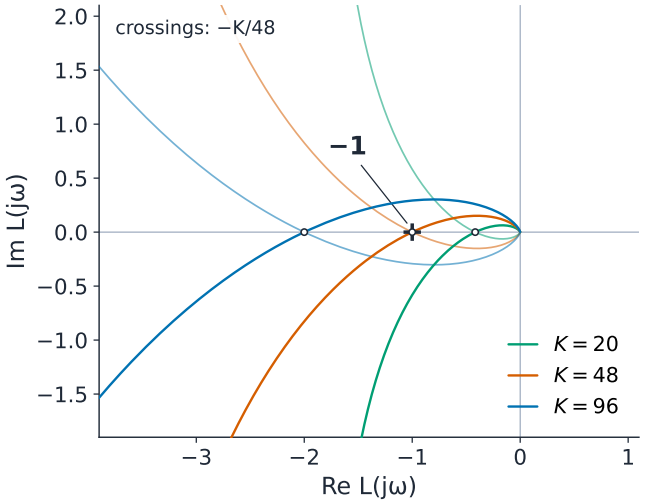
Read PM at the gain crossover; the phase never reaches −180°, so GM is infinite

exercises.lol exercises.com.cn

Worked example 1: margins by arithmetic. Take $L = \dfrac{100}{s(s+10)}$. The gain crossover solves $|L| = 1$, that is $\omega\sqrt{\omega^2+100} = 100$. Squaring gives $\omega^4 + 100\omega^2 - 10^4 = 0$, so

$$\omega_c^2 = \frac{-100 + \sqrt{100^2 + 4\times 10^4}}{2} = 61.8, \qquad \omega_c = 7.86 \text{ rad/s}.$$

The phase there is $-90° - \arctan(7.86/10) = -90° - 38.2° = -128.2°$, so the phase margin is $PM = 180° - 128.2° = 51.8°$. The gain margin is infinite: the phase only approaches $-180°$ asymptotically. Cross-check in the time domain: the closed loop is $\dfrac{100}{s^2+10s+100}$, exactly the prototype with $\omega_n = 10$ and $\zeta = 0.5$, an overshoot of 16.3%. The rule $\zeta \approx PM/100$ predicted 0.52 against an exact 0.5: a working estimate, not a theorem.



One curve shape; gain slides it through −1. At $K = 96$ it wraps twice around the point

Worked example 2: the running plant, three ways. Take $L = \dfrac{20}{s(s+2)(s+4)}$, the $\zeta = 0.5$ design with the gain more than doubled. The phase reaches $-180°$ where the imaginary part of $s(s+2)(s+4)$ vanishes, at $\omega = \sqrt{8} = 2.83$ rad/s, and there

$$|L| = \frac{20}{2.83 \times 3.46 \times 4.90} = \frac{20}{48} = 0.417, \qquad GM = \frac{48}{20} = 2.4 \ \ (7.6 \text{ dB}).$$

The gain crossover solves $\omega\sqrt{\omega^2+4}\sqrt{\omega^2+16} = 20$, giving $\omega_c = 1.73$ rad/s. The phase there is $-154.3°$, so $PM = 25.7°$. The rule predicts $\zeta \approx 0.26$ and a ringing

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

response; simulating the exact closed loop gives 48% overshoot, the behaviour the margin promised. Now the Nyquist check. The magnitude at the $-180°$ frequency is $K/48$, so at $K = 20$ the curve passes 0.42 to the right of -1: no encirclement, stable. At $K = 48$ it passes exactly through -1: marginal. At $K = 96$ it wraps twice clockwise around -1, so $Z = N + P = 2 + 0 = 2$ closed-loop RHP poles. Routh on $s^3 + 6s^2 + 8s + K$ says unstable for $K > 48$, with two RHP roots at $K = 96$. Three methods, one answer.

Common mistakes.

- Reading margins off the closed-loop response. Margins live on the loop gain L, before the loop closes.
- Mixing up the two crossovers. PM belongs to the gain crossover; GM to the phase crossover.
- Counting encirclements of the origin. The criterion counts crossings of the point -1, and with this sign convention clockwise is the positive direction.
- Forgetting P. If the plant itself has RHP poles, $Z = N + P$ needs that count; N alone is not the answer.

Summary. Bode gives PM and GM in seconds; $\zeta \approx PM/100$ carries them into the time domain; Nyquist counts RHP poles exactly, and all three must agree with Routh.

PID control and compensation

Why it matters. Most industrial loops run a PID. It is the first controller you will tune, and the one you will tune most often.

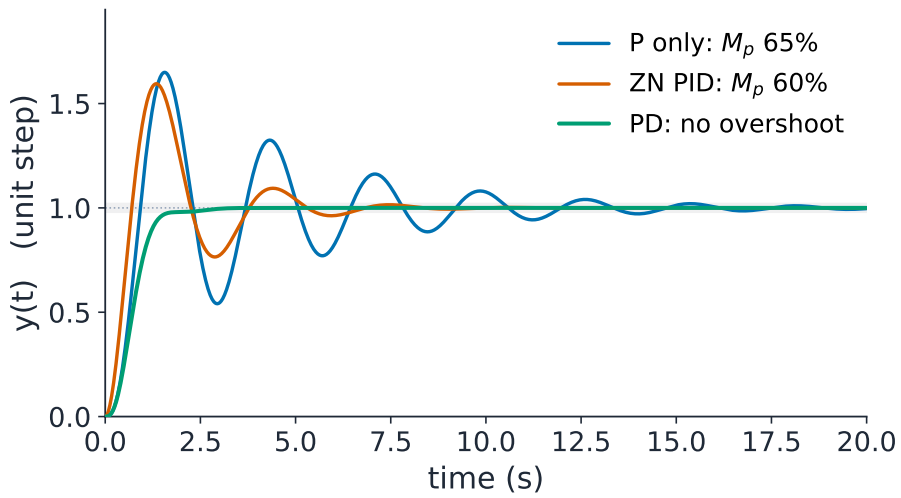
The idea. A **PID controller** PID acts on the error $e = r - y$ three ways at once:

$$u(t) = K_p\, e \;+\; K_i \int e\, dt \;+\; K_d \frac{de}{dt}.$$

The **proportional** term reacts now. The **integral** term remembers: it removes steady-state error by raising the system type. The **derivative** term anticipates, adding damping proportional to how fast the error moves. In the series (ideal) form, $u = K_p\big(e + \frac{1}{T_i}\int e\, dt + T_d\dot{e}\big)$, the tuning constants are the integral time T_i and the derivative time T_d, both in seconds.

The classical first cut is Ziegler-Nichols ultimate-gain tuning: raise the proportional gain until the loop just oscillates, record the ultimate gain K_u and oscillation period T_u, and look up the row of the table. The price list matters as much as the gains. P alone never removes steady-state error. I adds phase lag. D amplifies measurement noise, and it must act on the measured output rather than the reference, or every setpoint change kicks the actuator.

exercises.lol exercises.com.cn



One plant, three tunings; every curve simulated from the closed-loop equations

Worked example 1: Ziegler-Nichols on the running plant. The loop $\dfrac{K}{s(s+2)(s+4)}$ oscillates at $K_u = 48$ with $\omega_u = \sqrt{8} = 2.83$ rad/s (the Routh and Nyquist boundary), so the period is $T_u = \dfrac{2\pi}{2.83} = 2.22$ s. The PID row of the table gives

$$K_p = 0.6\,K_u = 28.8, \qquad T_i = T_u/2 = 1.11 \text{ s}, \qquad T_d = T_u/8 = 0.278 \text{ s}.$$

What do you actually get? With P alone at $K_p = 28.8$, the closed loop $s^3 + 6s^2 + 8s + 28.8$ has poles -5.50 and $-0.25 \pm j2.27$: 65% overshoot, settling around 15 s, and a permanent ramp error $\dfrac{1}{K_v} = \dfrac{8}{K_p} = 0.278$. The Ziegler-Nichols PID removes that error completely, because the integrator raises the type, and its first overshoot is 60%, decaying to about a sixth each cycle: the recipe aims at a quarter-decay, not at comfort. The honest lesson: the table lands you in the neighbourhood in one experiment, and it does not finish the job.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Worked example 2: finish with derivative, and size a lead. Keep $K_p = 28.8$ and add derivative. The PD loop's characteristic polynomial is $s^3 + 6s^2 + (8 + K_pT_d)s + K_p$: the D term literally adds to the damping coefficient. Raise T_d from the Ziegler-Nichols value until the ringing stops. At $T_d = 0.5$ s the polynomial is $s^3+6s^2+22.4s+28.8$ with poles -2 and $-2\pm j3.22$. Where derivative acts matters: on the **measurement** (the right practice, as below) the reference step meets no zero, and the simulated step shows no overshoot at all, settling in 2.1 s. On the **error** the PD zero at -2 cancels the real pole and the surviving pair, $\zeta = 2/3.79 = 0.53$, still rings at about 14%: damping either way, but only the measurement form is kick-free. The ramp error stays 0.278: derivative buys damping, not accuracy. For accuracy and speed together you combine the terms and retune, which is exactly the exercise on sheet 10.7.

When the plant has no clean ultimate gain, frequency-domain lead compensation does the same job. A **lead compensator** $\dfrac{1+\tau s}{1+\alpha\tau s}$ with $0 < \alpha < 1$ puts a zero below a pole and buys phase at one frequency. The maximum lift is

$$\sin\phi_m = \frac{1-\alpha}{1+\alpha} \qquad \text{at} \qquad \omega_m = \frac{1}{\tau\sqrt{\alpha}}.$$

For a $+45°$ lift, $\alpha = \dfrac{1-\sin 45°}{1+\sin 45°} = 0.172$. The same network also lifts the high-frequency gain by $\dfrac{1}{\alpha} = 5.8$, which the design must then absorb, usually by lowering the main gain. A lag network is the same pair with $\alpha > 1$: it buys low-frequency gain for error reduction, at the price of phase lag.

Anti-windup belongs in every real implementation. When the actuator saturates, the integrator keeps accumulating error it cannot spend, and on recovery it dumps the stored action as one large overshoot. The two standard cures: clamp the integrator while the output saturates, or back-calculate, feeding a gain times the saturation excess back to reduce the integral. Without one of these, the first real saturation eats every margin you designed.

Common mistakes.

- Running the ultimate-gain method on a plant that never oscillates. A first-order or heavily damped plant has no K_u; use the step-response variant of the table instead.
- Taking derivative of the error instead of the measurement. A setpoint step has an infinite slope, and the D term kicks the actuator.
- Confusing the tuning forms. The series constants K_p, T_i, T_d are not the parallel gains: $K_i = K_p/T_i$ and $K_d = K_pT_d$.
- Designing a lead network and forgetting its $\dfrac{1}{\alpha}$ high-frequency gain lift, which erodes the very margin you bought.

Summary. P reacts, I remembers, D anticipates; each buys one behaviour and charges

exercises.lol exercises.com.cn

a known price. Use the table for the first cut, margins for the refinement, and anti-windup for reality. The interactive lab on the course page lets you turn exactly these two knobs, gain and damping, and watch overshoot, settling and stability move.

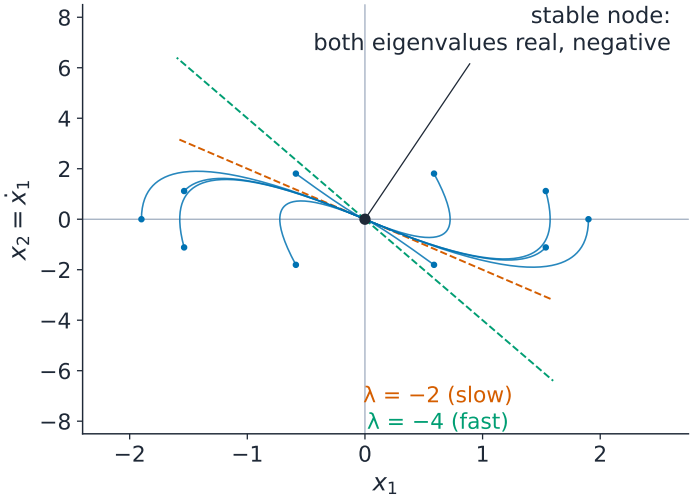
State-space introduction

Why it matters. Transfer functions describe one input and one output. Modern controllers, like a balancing robot or a drone running four loops at once, live on state space, where several internal variables are modelled and fed back together.

The idea. State space writes the system as n first-order equations instead of one n-th-order polynomial:

$$\dot{x} = Ax + Bu, \qquad y = Cx + Du,$$

where x is the state vector, u the input, y the measured output, and A, B, C, D are matrices of constants. Intuition: the states are n drawers of memory. The matrix A says how each drawer pushes the others, B says where the input enters, and C says what you measure. The **eigenvalues** of A are the poles of the system, because $\det(sI - A) = 0$ is exactly the characteristic equation. With zero input, the solution is $x(t) = \Phi(t)\,x(0)$, where the **transition matrix** $\Phi(t) = e^{At}$ is the matrix exponential, computed as the infinite series $I + At + \dfrac{(At)^2}{2!} + \cdots$, which converges for every square matrix.



Trajectories straighten onto the slow eigenvector: the tail of every response is the slow pole

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Worked example 1: read a model, verify its decay. Take $G(s) = \dfrac{8}{(s+2)(s+4)} = \dfrac{8}{s^2+6s+8}$. In companion form,

$$A = \begin{bmatrix} 0 & 1 \\ -8 & -6 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 8 & 0 \end{bmatrix}, \quad D = 0.$$

The DC gain checks the construction: $-CA^{-1}B = \frac{8}{8} = 1$, matching the plant gain at $s = 0$. The eigenvalues of A are -2 and -4, exactly the poles. Now the transition matrix at $t = 0.5$ s, evaluated by the series:

$$\Phi(0.5) = e^{0.5A} = \begin{bmatrix} 0.600 & 0.116 \\ -0.930 & -0.097 \end{bmatrix}.$$

Its eigenvalues are $e^{-1} = 0.368$ and $e^{-2} = 0.135$: each pole becomes its own decay factor per half-second. That is the deep identity: the matrix exponential is the pole, made real. Starting from $x(0) = [1,\ 0]^T$, the state after 0.5 s is the first column, $[0.600,\ -0.930]^T$. In the phase plane every trajectory straightens onto the slow eigenvector $[1, -2]^T$, which is why the $\lambda = -2$ pole owns the tail of the response.

Worked example 2: pole placement by linear algebra. With full-state feedback $u = -K_{fb}\,x$ (here K_{fb} is a row vector of gains, not the loop gain), the closed-loop matrix is $A - BK_{fb}$ and you choose its characteristic polynomial directly. Demand both poles at -10, that is $s^2+20s+100$. With $B = [0,\ 1]^T$ and $K_{fb} = [k_1,\ k_2]$,

$$A - BK_{fb} = \begin{bmatrix} 0 & 1 \\ -8-k_1 & -6-k_2 \end{bmatrix},$$

whose characteristic polynomial is $s^2 + (6+k_2)s + (8+k_1)$. Match coefficients: $k_1 = 92$ and $k_2 = 14$. The check is one line: the eigenvalues of $A - BK_{fb}$ compute to -10 and -10 exactly. What the root-locus unit did by searching along a ray, state feedback does by solving linear equations, and it can place poles that no cascade of compensators reaches, provided the input can actually drive every state direction. That property is **controllability**, checked by the rank of the matrix $[\,B \quad AB \quad \cdots\,]$, which is 2 here, so both poles are placeable. When the states are not measured, an observer estimates them; that is the door to the next course.

Common mistakes.

- Forgetting C. The states are not the output; the transfer function is $C(sI - A)^{-1}B + D$, and dropping C quietly changes the DC gain.
- Taking e^{At} element by element. The matrix exponential is defined by the series, and its off-diagonal entries are not exponentials of the entries.
- Assuming pole placement always works. If $[\,B \quad AB \quad \cdots\,]$ loses rank, some directions are unreachable and no K_{fb} exists; check controllability first.

exercises.lol exercises.com.cn

- Mixing conventions. This course uses column states and a row K_{fb}; a textbook with row states flips every product.

Summary. A holds the dynamics, its eigenvalues are the poles, e^{At} is the free response, and $u = -K_{fb}x$ turns design into linear algebra, guaranteed by controllability.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

Topic 11

University Electrical Engineering

What this course is, and how to use it

This is the eleventh course in University Electrical Engineering: **communication systems** . EE06 built the signal tools; this course sends signals across real channels – modulated onto carriers, quantised into bits, protected by codes.

- Units 11.1–11.3 are analogue: spectra, AM, FM.
- Units 11.4–11.6 are the digital chain: noise and SNR, PCM, modulation.
- Unit 11.7 sets the limits: entropy, capacity, coding.

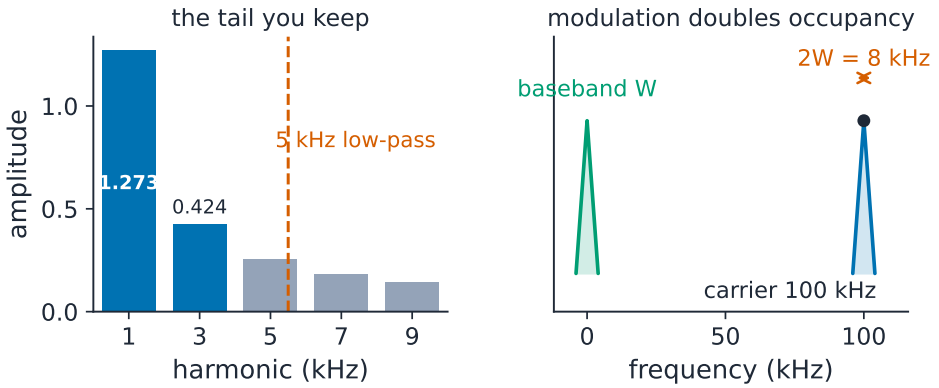
The running story is one radio link: a microphone, a transmitter, a noisy channel, a receiver, and a data modem sharing the same wire. Every unit below moves one block of that link forward, and every number is checked the way a communications engineer checks it – every dB claim recomputed as a ratio, every power budget closed (carrier + sidebands = total), and every bandwidth claim tested against the channel it must fit.

Signals, spectra and bandwidth

Why it matters. A radio regulator sells spectrum by the kilohertz; whether a signal fits its licence is decided by its spectrum, and the spectrum is decided by the waveform's shape.

The idea, with a picture. A real signal occupies spectrum. The square wave is odd harmonics at $4/\pi \cdot 1/k$ – the 3rd harmonic at 1/3 amplitude, the 5th at 1/5 – so **bandwidth** is a design choice about how much of the tail you keep. For modulated signals the **sideband** structure rules: mixing shifts the spectrum, it never creates new shape, so a baseband signal of width W on a carrier occupies $2W$ of radio spectrum.

exercises.lol exercises.com.cn



Left: the tail you keep is the bandwidth you pay for. Right: modulation doubles the occupancy

The derivation. A Fourier series writes the square wave as $\frac{4}{\pi}\sum_{k\ \text{odd}}\frac{1}{k}\sin(k\omega_0 t)$, so a low-pass at B keeps the harmonics with $kf_0 \le B$, and the power fraction kept follows from Parseval: each harmonic's power is its amplitude squared, so the fundamental alone carries $1.273^2/(1.273^2+0.424^2+0.255^2+\ldots)\approx 81\%$.

> **Worked example 1 – a 1 kHz square through a 5 kHz low-pass.** The filter passes the 1st harmonic (full amplitude $4/\pi = 1.273$) and the 3rd ($1.273/3 = 0.424$), blocks the 5th (0.255) and beyond. Check the power kept: $(1.273^2+0.424^2)/(1.273^2+0.424^2+0.255^2+\ldots)\approx 90\%$ – the ear hears the rounding, the meter barely moves, which is why the 300–3400 Hz telephone channel passes speech intelligibly at all.

> **Worked example 2 – escalate to the radio licence.** A 4 kHz music signal is placed on a 100 kHz carrier by mixing: the spectrum becomes the carrier plus mirror sidebands spanning 96 kHz to 104 kHz – $2W = 8$ kHz of occupied bandwidth. If the licence grants 8 kHz, amplitude modulation exactly fills it; grant only 4 kHz and you must transmit one sideband (SSB, unit 11.2's theme). Check: no mix of filters can compress the occupancy below W without deleting signal – the $2W$ law is geometry, not electronics.

Common mistakes. Quoting "bandwidth" without a definition (−3 dB? 98% power? first null? – state which). Believing a filter can narrow a modulated signal below its message width. And forgetting that harmonics of a square wave fall as $1/k$ in AMPLITUDE but $1/k^2$ in POWER – the tail is weaker than it looks.

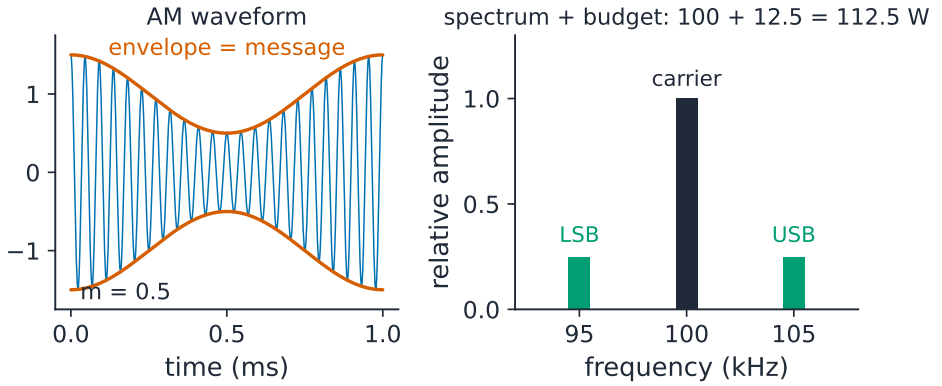
Summary. Spectra come from shapes; a low-pass tail is a design choice measured in power fraction; modulation shifts and doubles. $2W$ is the occupancy law every licence is written against.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Amplitude modulation

Why it matters. AM is the simplest carrier system that still closes its power budget on paper – and the arithmetic of that budget explains why every broadcast engineer moved on, and why the envelope detector still ships in every radio.

The idea, with a picture. In **amplitude modulation** the carrier's envelope carries the message: $s(t) = A[1 + m\cdot x(t)]\cos(\omega_c t)$ with **modulation index** $m \le 1$. The spectrum is the carrier plus two sidebands at $f_c \pm f_m$, and the power budget is exact: sideband power $= (m^2/2)P_c$, total $P_c(1 + m^2/2)$.



The envelope is the message; the budget bars show the carrier taking most of the power

The derivation. With a single-tone message $x = \cos(\omega_m t)$: $s(t) = A\cos(\omega_c t) + \frac{mA}{2}\cos((\omega_c+\omega_m)t) + \frac{mA}{2}\cos((\omega_c-\omega_m)t)$ – write the product of cosines as a sum and the two sidebands appear with amplitude $mA/2$, hence power $(mA/2)^2$ each and $\frac{m^2}{2}P_c$ together.

> **Worked example 1 – the budget at m = 0.5.** A 100 W carrier: sidebands total $\frac{0.25}{2}\times 100 = 12.5$ W, transmission 112.5 W, efficiency $12.5/112.5 = 11.1\%$ ✓. At the legal maximum $m = 1$: 50 W in sidebands, 150 W total, 33.3% – two thirds of the transmitter power goes into a carrier that carries no information. Both budgets close on $P_c(1 + m^2/2)$.

exercises.lol exercises.com.cn

Worked example 2 – escalate to the detector design. The **envelope detector** (a diode and an RC) must discharge slowly enough to follow the envelope's fastest dips but fast enough not to bridge carrier peaks: $1/\omega_c \ll \tau \ll 1/\omega_m$. For $f_c = 100$ kHz and $f_m = 5$ kHz, τ must sit between 1.6 μs and 31.8 μs; $\tau = 10$ μs works with margin at both ends ✓. Push m toward 1 and the upper bound tightens to $\dfrac{\sqrt{1-m^2}}{m\,\omega_m}$ – at $m = 0.9$ it is $\dfrac{\sqrt{0.19}}{0.9} \times \dfrac{1}{2\pi \cdot 5\,\text{kHz}} = 15.4$ μs: the same RC, less than half the slack, which is why deep modulation distorts before anything else fails.

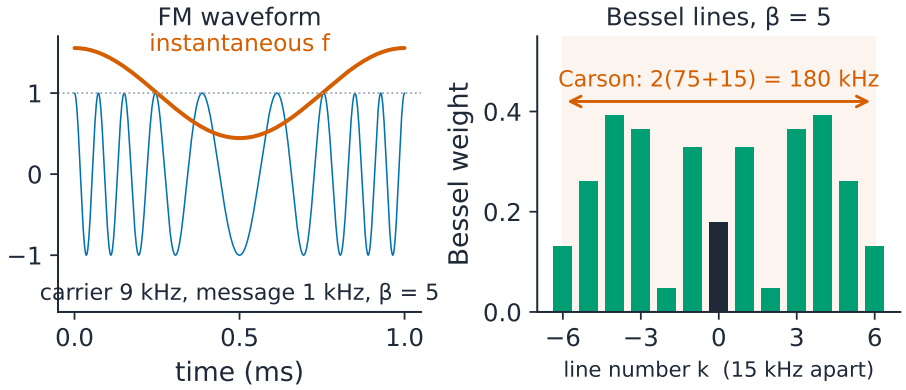
Common mistakes. Setting $m > 1$ (envelope folds, phase reverses, detector produces garbage). Computing efficiency against total power one way and against carrier power another (state the denominator). And choosing τ by the carrier alone: the carrier sets only the lower bound; the message's highest frequency and its depth m set the upper one.

Summary. AM: envelope = message; sidebands $m^2/2 \times P_c$; efficiency peaks at 33%; the detector's RC lives between $1/\omega_c$ and $\sqrt{1-m^2}/(m\,\omega_m)$.

Angle modulation (FM/PM)

Why it matters. FM broadcasts through thunderstorms because it trades bandwidth for noise immunity, and the size of that trade – Carson's rule on one side, the SNR gain on the other – is the course's cleanest engineering bargain.

The idea, with a picture. Frequency modulation puts the message in the **instantaneous frequency** : $f_i = f_c + k_f x(t)$. The **frequency deviation** Δf against the highest message frequency f_m gives the **modulation index** $\beta = \Delta f / f_m$, and **Carson's rule** $B \approx 2(\Delta f + f_m)$ bounds the meaningful spectrum.



The message bends the instantaneous frequency; Carson's rule fences the spectral spread

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

The derivation. Integrating f_i gives the phase $\phi(t) = 2\pi f_c t + \beta \sin(\omega_m t)$; expanding $A\cos(\phi)$ with the Jacobi–Anger identity produces spectral lines at $f_c \pm k f_m$ with Bessel weights $J_k(\beta)$ – significant lines stop near $k \approx \beta + 1$, which is exactly Carson's $2(\Delta f + f_m)$. Carson's rule estimates a bandwidth containing roughly 98% of the power in the usual FM model; it is not a strict spectral cutoff.

Worked example 1 – FM broadcast. $\Delta f = 75$ kHz, $f_m = 15$ kHz: $\beta = 5$, $B = 2 \times 90 = 180$ kHz – and the 200 kHz channel spacing leaves a 20 kHz guard band ✓. Narrowband ($\beta \ll 1$): B collapses to about $2f_m$, the walkie-talkie regime. Check the limit: $\Delta f \to 0$ gives $\beta \to 0$, $J_0 \to 1$ – the carrier alone, as it must.

Worked example 2 – escalate to the bargain's price. Wideband FM buys post-detection SNR: $(S/N)_{out} \approx (3\beta^2/2)\,(S/N)_{ref}$ in the high-SNR regime, where the reference $(S/N)_{ref}$ is the received power over the noise in the message band f_m (what baseband transmission at the same power would give). For $\beta = 5$: $10\log_{10}(37.5) = 15.7$ dB of improvement – paid for with 180 kHz of bandwidth (6× the 30 kHz that AM would occupy for the same 15 kHz audio) and lost entirely below the FM threshold near a carrier-to-noise ratio of about 10 dB in the 180 kHz channel, where click noise and capture take over. The ledger: 15.7 dB of gain for 7.8 dB of spectrum.

Common mistakes. Using Carson's rule as if it were exact (it is a 98%–power bound). Mixing up Δf (a property of the modulator) with B (a property of the spectrum). And forgetting PM: the same machinery with the message in the phase, $\beta_{PM} = k_p A_m$ – frequency-independent, which is why FM won broadcasting while PM lives inside data modems.

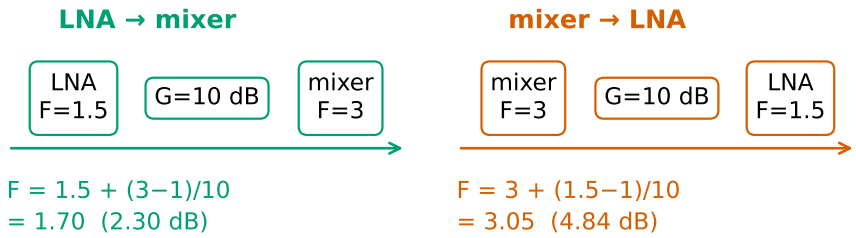
Summary. $\beta = \Delta f / f_m$; $B \approx 2(\Delta f + f_m)$; the wideband SNR gain is $3\beta^2/2$, real only above threshold.

Noise and SNR

Why it matters. The link budget of the running story ends at a noise floor that no amplifier can dig below – the thermal floor of the universe – and every receiver stage decides how far above it the signal arrives.

The idea, with a picture. Every receiver adds noise. The **noise figure** F measures how much, and Friis's cascade law $F = F_1 + (F_2 - 1)/G_1$ says the FIRST stage decides everything: its gain divides every later stage's contribution.

exercises.lol exercises.com.cn



Same blocks, swapped order, 2.5 dB different – the front-end gain IS the protection

The derivation. Per-stage input-referred noise adds at the input of each stage, and each earlier gain scales the later noise down by its gain, so the cascade's total referred noise is $F_1 + (F_2 - 1)/G_1 + (F_3 - 1)/(G_1 G_2) + \ldots$ – after two stages of healthy gain the tail terms vanish. Thermal floor: $-174\ dBm/Hz + 10\log_{10} B$.

> **Worked example 1 – the LNA wins.** LNA with $F_1 = 1.76$ dB ($F = 1.5$) and $G_1 = 10$ dB ahead of a mixer with $F_2 = 4.77$ dB ($F = 3$): the chain computes $F = 1.5 + (3-1)/10 = 1.70 = 2.30$ dB ✓ – the mixer's 4.77 dB collapsed to a 0.5 dB penalty. Swap the blocks: $3 + (1.5-1)/10 = 3.05 = 4.84$ dB – 2.5 dB worse for the same parts. In a 180 kHz FM channel the floor sits at $-174 + 10\log_{10}(180\,000) = -121.4$ dBm.

> **Worked example 2 – escalate to the link budget.** The transmitter delivers −20 dBm at the receiver's antenna after path loss; the channel is 180 kHz wide. Signal −20 dBm against noise −121.4 dBm is an input $S/N = 101.4$ dB – enormous. The receiver's own noise raises the floor by its 2.3 dB noise figure to −119.1 dBm, so the carrier-to-noise ratio after the front end is $-20 - (-119.1) = 99.1$ dB, far above the FM threshold. The FM gain of unit 11.3 is quoted against the message-band reference: noise in the 15 kHz audio band is $-174 + 10\log_{10}(15\,000) + 2.3 = -129.9$ dBm, so $(S/N)_{ref} = 109.9$ dB, and the demodulator's 15.7 dB lifts the post-detection S/N to about 125.6 dB in the linear regime – far above hi-fi; the practical limit becomes stereo processing, not noise. One budget, three units, closed.

Common mistakes. Adding noise figures in dB (convert to linear F first, add ratios, convert back once). Forgetting the floor scales with bandwidth – 10 dB per decade of B. And quoting a cascade F for a mismatched reference temperature or impedance; F is defined at 290 K.

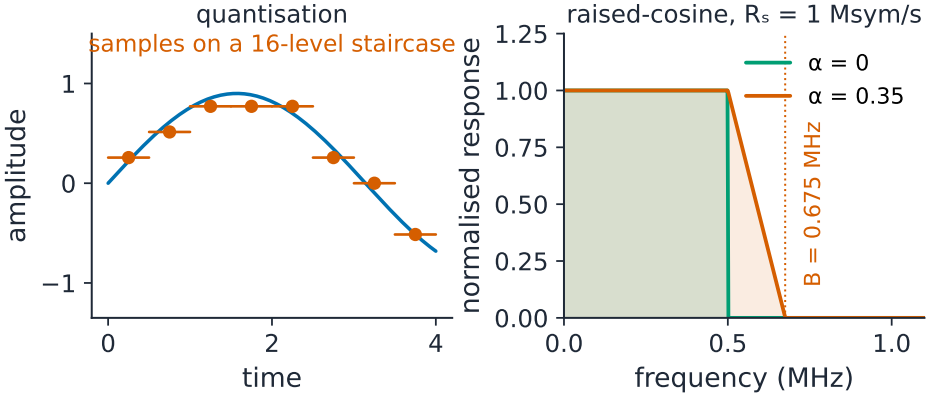
Summary. $F_{total} = F_1 + (F_2 - 1)/G_1 + \ldots$; first stage rules; floor $= -174\ dBm/Hz + 10\log_{10} B$; every budget closes in dB arithmetic once the ratios are linear.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

PCM and baseband transmission

Why it matters. The running link goes digital in this unit: the microphone's voltage becomes bits, the bits become pulses, and the pulse shape decides how fast the wire can carry them.

The idea, with a picture. Quantisation trades precision for bits: n bits give 2^n levels and $SQNR \approx 6.02n + 1.76$ dB. Sampling at $f_s > 2W$ (Nyquist) then encoding is **PCM** – the whole telephone network's foundation. At baseband, the **raised-cosine** filter with roll-off α trades bandwidth for timing slack: $B = (1+\alpha)R_s/2$.



Sampling freezes time, quantisation freezes amplitude, shaping fences the spectrum

The derivation. A uniform quantiser with step Δ rounds each sample to the nearest level; the error is uniform on $[-\Delta/2, \Delta/2]$ with variance $\Delta^2/12$. Against a full-scale sine of power $(2^n\Delta)^2/8$, the ratio is $1.5 \cdot 2^{2n}$ – $10\log_{10}$ of which is $6.02n + 1.76$ dB. The 6.02 is one bit = one doubling = $10\log_{10} 4$.

> **Worked example 1 – the telephone number.** 4 kHz voice sampled at 8 kHz with 8 bits per sample: $8\,000 \times 8 = 64$ kbps; $SQNR = 6.02 \times 8 + 1.76 = 49.9$ dB ✓ (256 levels). Every 1 dB of extra SQNR costs 1/6 of a bit per sample – the 8-bit choice is the toll-quality settlement of that price.

> **Worked example 2 – escalate to the wire.** At $R_s = 1$ Msym/s with $\alpha = 0.35$: $B = 1.35 \times 1/2 = 0.675$ MHz ✓ – the number every modem datasheet quotes. Cut the roll-off to $\alpha = 0$ and the spectrum touches the $R_s/2$ ideal but the pulse's zero-crossings become knife-edge timing. Manchester coding doubles the bandwidth to buy a strong clock (the classic Ethernet choice): its first spectral null sits at $2R$ against NRZ's R, and NRZ has a DC term and no clock. Each line code is a point on the bandwidth-versus-clock plane.

Common mistakes. Sampling exactly at $2W$ (the inequality is strict; at equality the

exercises.lol exercises.com.cn

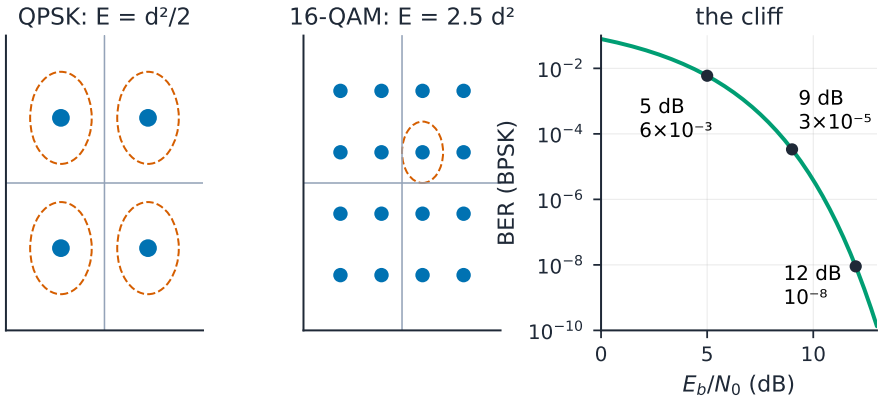
guard band is zero and any filter ripple aliases). Confusing $SQNR$ with the analog SNR of the whole chain. And forgetting $\alpha = 0$ is unbuildable – it is the bound that real filters approach.

Summary. $SQNR = 6.02n + 1.76$ dB; Nyquist $f_s > 2W$; raised-cosine $B = (1+\alpha)R_s/2$; line codes trade spectrum for clock strength.

Digital modulation

Why it matters. The link's bits must ride a carrier, and the constellation you draw decides how many of them fit per hertz and how easily noise knocks them off their points.

The idea, with a picture. Bits ride carriers as **constellations** : ASK moves amplitude, PSK rotates phase, QAM does both. More points per symbol = more bits/Hz and more sensitivity to noise – the two-dimensional trade is the whole story.



Equal minimum distance d: 16-QAM packs 2× the bits per symbol for 5× the average energy

The derivation. For equal minimum distance d, the average symbol energy is the mean of the squared radii, $E_{avg} = \overline{r^2}$, computed directly per constellation: QPSK's four points sit on a circle of radius $r = d/\sqrt{2}$, so $E_{avg} = r^2 = d^2/2$; 16-QAM's grid (4 inner points, 8 edge points, 4 corner points) averages the squared radii to $E_{avg} = (4 \cdot 0.5 + 8 \cdot 2.5 + 4 \cdot 4.5)\, d^2/16 = 2.5d^2$ – a factor 5, i.e. 7.0 dB, for 2× the bits per symbol.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

Worked example 1 – the energy ledger. QPSK: 2 bits/symbol at $E = 0.5d^2$. 16-QAM: 4 bits/symbol at $2.5d^2$ – five times the symbol energy, 7.0 dB. Per bit the ledger softens: $0.25d^2$ against $0.625d^2$ – factor $2.5 = 4.0$ dB, because 16-QAM carries two bits per symbol ✓. Check with the datasheet number: the classic 16-QAM-vs-QPSK gap in E_b/N_0 at 10^{-6} BER is quoted near 4 dB – the per-bit geometry, measured.

Worked example 2 – escalate to the cliff. Under AWGN, BPSK's BER is $Q(\sqrt{2E_b/N_0})$. At $E_b/N_0 = 5$ dB: $\sqrt{2 \times 3.16} = 2.51$, $Q(2.51) \approx 6 \times 10^{-3}$. At 9 dB ($\sqrt{2 \times 7.94} = 3.98$): $Q(3.98) \approx 3 \times 10^{-5}$. At 12 dB: $\approx 10^{-8}$. Three points, one cliff – 4 dB buys two orders of magnitude and the next 3 dB three more, which is why link budgets obsess over the last dB.

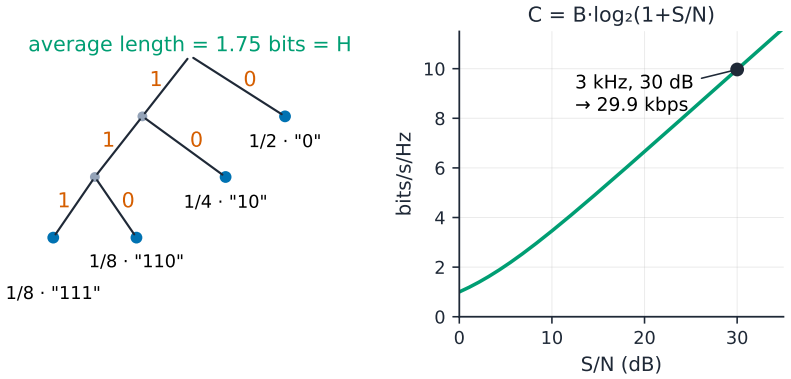
Common mistakes. Comparing constellations at equal average energy when the fair comparison is equal minimum distance (or equal peak). Reading E_b/N_0 curves without checking the normalisation. And bits-per-symbol gains scale as $\log_2 M$ while energy grows far faster in M – the cliff gets steeper.

Summary. Equal-d ledger: QPSK $d^2/2$, 16-QAM $2.5d^2$ (7 dB, 2× bits). BPSK BER $Q(\sqrt{2E_b/N_0})$; the curve is a cliff – budget the dB before the link goes out.

Information and coding introduction

Why it matters. The last block draws the map's edges: bits a source deserves, bits a channel allows, and how codes move practice toward theory.

The idea, with a picture. Entropy $H = -\sum p \log_2 p$ measures a source's surprise in bits/symbol; the gap to fixed-length coding is redundancy. Huffman removes it for the source; **channel codes** spend redundancy on protection instead. The ceiling is **channel capacity** $C = B \log_2(1 + S/N)$.



Left: optimal lengths average exactly the entropy for dyadic probabilities. Right: capacity, approached but never beaten

exercises.lol exercises.com.cn

The derivation. For a dyadic source (p's are negative powers of 2), Huffman's greedy merge produces lengths $\log_2(1/p)$ exactly, so the average equals H – met, not approached; non-dyadic sources sit within one bit.

> **Worked example 1 – the quartile source.** Probabilities 1/2, 1/4, 1/8, 1/8: the four terms $p\log_2(1/p)$ are 0.5, 0.5, 0.375, 0.375, so $H = 1.75$ bits ✓ against 2 bits fixed-length – 87.5% efficient; Huffman lengths 1, 2, 3, 3 average exactly 1.75 ✓ – dyadic, bound met. **Hamming code** (7,4): 4 data + 3 parity, distance 3, corrects any single error: encode 1011 → 0110011; flip bit 5, the syndrome computes to $101_2 = 5$ ✓. Rate 4/7 (43% redundancy).

> **Worked example 2 – escalate to the modem ceiling.** A 3 kHz channel at 30 dB SNR: $C = 3000 \times \log_2(1001) = 29.9$ kbps – the dial-up ceiling. V.90/V.92 hit 56 kbps downstream only by going digital to the exchange – inside the law. Check the direction: 10 dB more SNR buys $\log_2(10) \approx 3.3$ bits/Hz – linear in bandwidth, logarithmic in power.

Common mistakes. Computing H with the wrong log base (bits = $\log_2$). Believing Huffman always meets H (only dyadic). And reading C as a guarantee – codes approach it, none reach it.

Summary. $H = \sum p\log_2(1/p)$, met by Huffman on dyadic sources; Hamming (7,4) fixes one error at 43% redundancy; $C = B\log_2(1 + S/N)$ is the ceiling.

exercises.lol exercises.com.cn

exercises.lol exercises.com.cn

Digital Signal Processing

University Electrical Engineering

What this course is, and how to use it

This is the twelfth course in University Electrical Engineering: **digital signal processing** – the last of the EE strand. EE06 built the continuous-time theory; this course re-founds it in samples: the transforms that live on a computer, the filters you can actually ship, and the multirate machinery that moves between rates.

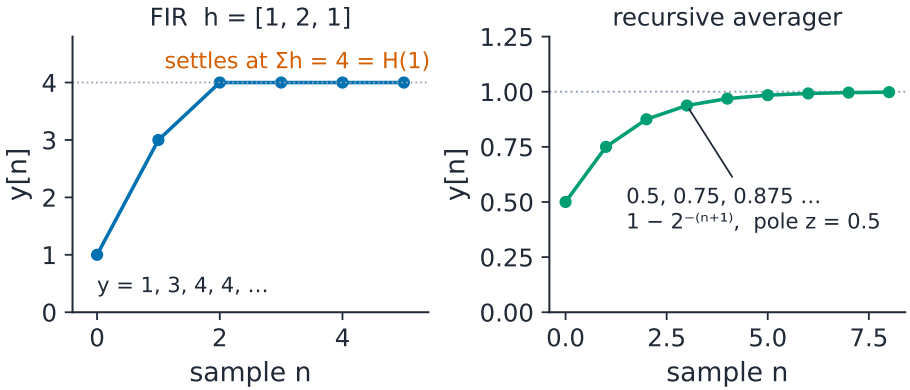
- Units 12.1–12.3 are the foundations: discrete systems, DTFT/DFT, FFT.
- Units 12.4–12.5 design filters: FIR by windows, IIR by bilinear transform.
- Units 12.6–12.7 change rates and build a small verified pipeline.

The checking habit: every filter design is verified against its specification (measured ripple, measured transition, measured stopband), every frequency-mapping claim is prewarped explicitly, and every speedup claim is an operation count, not folklore.

Discrete-time signals and systems review

Why it matters. Everything in this course – every filter, every spectrum, every speedup – is eventually a program operating on a list of numbers. The whole behaviour of that program is decided by one list: the impulse response.

The idea, with a picture. A discrete system is LTI when it is fully described by its **impulse response** $h[n]$: the output is **convolution** $y[n] = \sum_k h[k] \cdot x[n-k]$ – slide the taps across the input and add one diagonal at a time.



The FIR settles in three samples; the recursive averager never exactly settles – the tap sum is the destination in both

Causality means $h[n] = 0$ for $n < 0$ (no output before an input); **BIBO**

exercises.lol exercises.com.cn

stability BIBO means $\sum |h[n]|$ converges. Every claim in this course reduces to one of those two sentences.

The derivation. Feed the complex exponential $x[n] = e^{j\omega n}$ through the convolution sum: each term picks up the factor $h[k]e^{j\omega(n-k)} = e^{j\omega n}h[k]e^{-j\omega k}$, so the output is $e^{j\omega n}\sum_k h[k]e^{-j\omega k}$ – the exponential is an eigenfunction, and the eigenvalue $H(e^{j\omega}) = \sum h[n]e^{-j\omega n}$ is the **frequency response** . The frequency response is just the DTFT of the taps: unit 12.2 builds on exactly this.

> **Worked example 1 – the sliding table.** $h = [1, 2, 1]$ (a 3-tap smoother) on a step $x = [1, 1, 1, 1, \ldots]$: slide and add – $y = 1, 3, 4, 4, \ldots$ The steady value is $\sum h = 4$ ✓ (a step through any FIR settles at the tap sum – that is the DC gain $H(1)$). A finite pulse $[1, 1, 1]$ gives $1, 3, 4, 3, 1$, mirroring the taps: the output reads the taps forwards then backwards.

> **Worked example 2 – the recursive cousin.** The running average $y[n] = (y[n-1] + x[n])/2$ has an infinite impulse response $h[n] = 0.5^{n+1}$: its step response computes to $0.5, 0.75, 0.875, 0.9375$ – halving the distance to 1 each step, the exponential $1 - 2^{-(n+1)}$ ✓, because its single pole sits at $z = 0.5$. Compare the two smoothers: the FIR's memory is exactly 3 samples; the recursive one never fully forgets. $|pole| < 1$ is stability – unit 12.5 makes a career of that.

Common mistakes. Flipping the wrong signal in the table (convolution is commutative, but a hand table must stay consistent or the diagonals lie). Quoting $\sum h$ as "the gain" without saying it is the gain at DC only. And expecting a recursive smoother to settle – it approaches, never arrives.

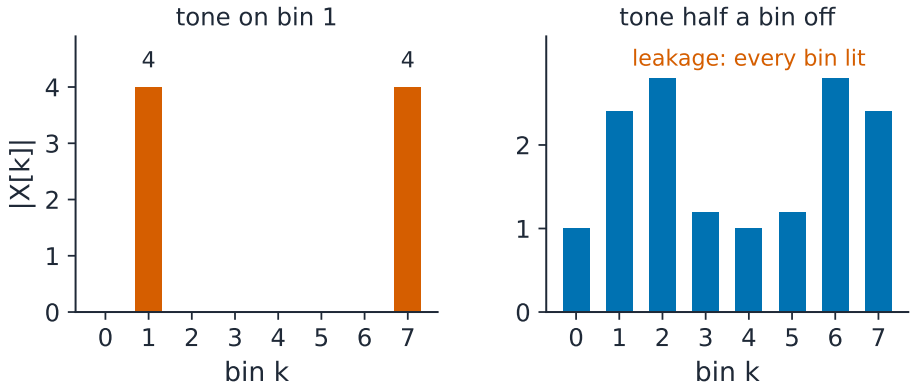
Summary. LTI means convolution by $h[n]$; causality is $h[n] = 0$ for $n < 0$; stability is $\sum |h| < \infty$ (poles inside the unit circle); exponentials are eigenfunctions with eigenvalue $H(e^{j\omega})$.

DTFT and DFT

Why it matters. A computer holds N samples, not an integral – the spectrum must become a finite list of numbers. How that list lies to you is the single most practical topic in the course.

The idea, with a picture. The **DTFT** $X(e^{j\omega}) = \sum_n x[n]e^{-j\omega n}$ is periodic in ω with period 2π. Sampling it at N points gives the **DFT** – and sampling in frequency means the time signal is treated as periodic: that single fact explains **leakage** .

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn



On-bin: two spikes. Half a bin off: the rectangular window's sidelobes paint every bin

Resolution is a time decision: bins are $\Delta f = f_s/N$ wide, so 1 Hz resolution at $f_s = 1$ kHz means $N = 1000$ – one second of data, no exceptions.

The derivation. Sampling the DTFT at $\omega_k = 2\pi k/N$ is multiplying it by a frequency comb; by duality the time signal becomes periodic with period N. If your tone sits exactly on a bin, the periodic copies join seamlessly – two clean spikes. If it sits between bins, the copies disagree and the disagreement smears energy into every bin. Windowing the data with $w[n]$ convolves the true spectrum with $W(e^{j\omega})$: the rectangle's sidelobes are only -13 dB down, so the smear is loud; the **Hamming window** suppresses its sidelobes to -43 dB, buying a clean spectrum at the price of a wider main lobe.

> **Worked example 1 – the smear, quantified.** An 8-point DFT of a cosine at exactly bin 1 ($f = f_s/8$): $|X| = 4$ at bins 1 and 7, zero elsewhere ✓ (all 8 samples describe whole periods). Slide the tone half a bin ($f = 1.5 \cdot f_s/8$): the magnitudes compute to $2.40, 2.80, 1.19, 1.0, \ldots$ across every bin ✓ – a tone between bins is visible everywhere, and a small signal beside a large one vanishes into its sidelobes. That is leakage's real cost.

> **Worked example 2 – the resolution budget.** At $f_s = 1$ kHz: $N = 100$ gives 10 Hz bins – two tones 6 Hz apart are ONE bump, however clean the plot looks. $N = 1000$ gives 1 Hz bins for one second of data. Zero-padding to 4000 points draws the same bump with $40\times$ more ink: padding interpolates the same spectrum, it resolves nothing new. Resolution was decided when you chose how long to measure.

Common mistakes. Reading bin numbers as hertz – the map is $f = k \cdot f_s/N$. Believing zero-padding buys resolution. And forgetting the DFT assumes your N samples repeat forever: the edge discontinuity between copy and copy is itself a broadband signal.

Summary. $X[k]$ = DTFT sampled at $2\pi k/N \Leftrightarrow$ time periodic with period N; off-bin

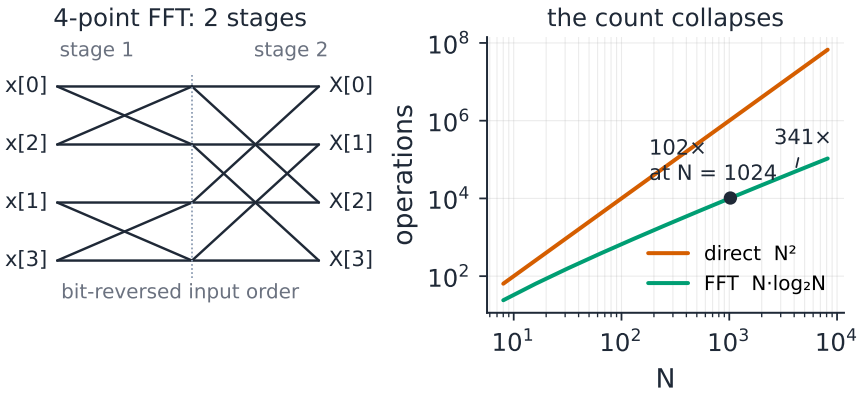
exercises.lol exercises.com.cn

tones leak through the window's sidelobes (rectangle −13 dB, Hamming −43 dB); $\Delta f = f_s/N$ is bought with measurement time, never with padding.

The FFT

Why it matters. The DFT costs N^2 complex operations – a 1024-point transform is a million multiplies. The **FFT** is the algorithm that made digital signal processing a subject at all.

The idea, with a picture. Factor an N-point DFT into two $N/2$-point DFTs plus $N/2$ combining multiplies – and recurse. The **butterfly** is the combining unit: $\log_2 N$ stages of $N/2$ butterflies, each one multiply-and-add.



Same answer, 102× fewer operations at $N = 1024$ – and the gap widens with N

The derivation. Split $x[n]$ into even and odd samples: $X[k] = \sum_{even} x[n]W_N^{kn} + W_N^k \sum_{odd} x[n]W_N^{kn}$ with the **twiddle factor** $W_N = e^{-j2\pi/N}$. Each sum is an $N/2$-point DFT (Danielson–Lanczos); the odd arm needs only the factor W_N^k to recombine. Recurse $\log_2 N$ deep. The even/odd splitting runs backwards through the address bits – hence **bit-reversed** input order for the decimation-in-time variant.

> **Worked example 1 – the count.** Counting complex multiply–add operations: direct $N^2 = 1024^2 = 1\,048\,576$; FFT $N \log_2 N = 1024 \times 10 = 10\,240$ – **102× fewer** ✓. At $N = 4096$: 4096^2 against $4096 \times 12 =$ **341×** ✓. The gap widens with N: the FFT is $O(N \log N)$, and the whole order of the computation changes.

> **Worked example 2 – the spectrum analyser.** 1 Hz bins at $f_s = 1$ kHz needs $N \geq 1000$: choose $N = 1024$ (a power of two), giving $\Delta f = 1000/1024 = 0.977$ Hz ✓ just enough. At ten nanoseconds per operation the transform finishes in about 100 μs – real time against a 1.024 s record. Bin 100 of that DFT is frequency $100 \times 0.977 = 97.66$ Hz ✓ – the bin-to-hertz mapping is $k \cdot f_s/N$, and forgetting it is the classic lab error.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

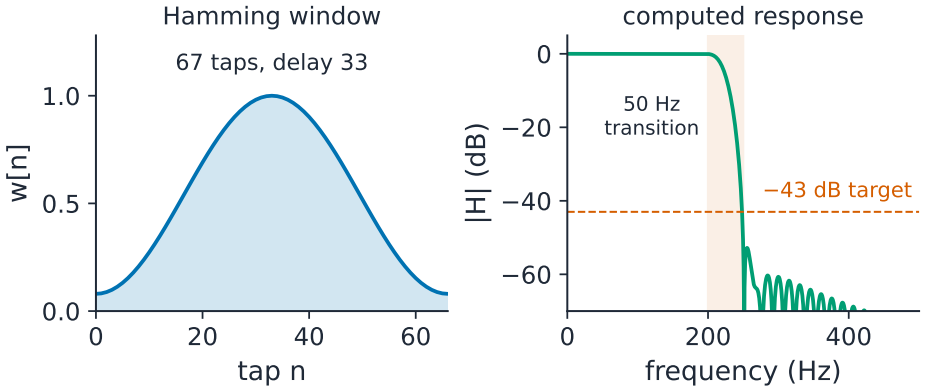
Common mistakes. Assuming N must be a power of two (mixed-radix and Bluestein handle any N, the power-of-two case is just the fastest). Expecting the FFT to change the answer – it computes the same DFT, only faster. And reading bins as hertz, again.

Summary. Even/odd split + twiddle factors, $\log_2 N$ stages, $N \log_2 N$ operations: a 102× speedup at $N = 1024$ that grows with N; the mathematics is the DFT, the engineering is the butterfly.

FIR filter design

Why it matters. The **FIR** filter is the workhorse you can prove things about: unconditionally stable, and – with symmetric taps – exactly linear phase. Frequencies in a flat passband keep their relative phases; amplitude filtering can still change a waveform.

The idea, with a picture. Start from the ideal low-pass's impulse response (a sinc), truncate it to N taps with a window $w[n]$, and the window's spectral shape IS the filter's fate: its main-lobe width becomes your transition band, its sidelobes become your stopband.



The window guides the design; compute the actual filter response to verify its passband and stopband

Definitions with numbers. Peak sidelobes of the windows themselves: rectangular −13 dB, Hann −31 dB, Hamming −43 dB, Blackman −58 dB. Transition width $\approx 3.3 \cdot f_s/N$ for Hamming. Real symmetric taps $h[n] = h[N-1-n]$ give **linear phase** with group delay $(N-1)/2$ samples wherever the response is nonzero. Phase may jump by π when the real amplitude factor changes sign; phase and group delay are undefined at response zeros.

The derivation. Truncating $h[n]$ is multiplying by $w[n]$; in frequency the true response convolves with $W(e^{j\omega})$. So the implemented response is the ideal brick wall smeared by the window: passband ripple, a transition of one main-lobe width, and

exercises.lol exercises.com.cn

a stopband whose attenuation depends on the convolution, cutoff and tap count. A window sidelobe value is not the resulting filter's stopband guarantee. Use the window table and transition estimate to choose a starting design, then compute its response. (The alternation theorem behind Parks–McClellan makes the equiripple design optimal where windows are merely convenient.)

> **Worked example 1 – size the filter.** A Hamming FIR at $f_s = 1$ kHz must pass up to 200 Hz and attenuate from 250 Hz. Choose a 225 Hz ideal cutoff in the 50 Hz transition: $N = 3.3 \cdot 1000/50 = 66$ ✓ – take 67, odd, for a Type-I symmetric filter with delay $(67-1)/2 = 33$ samples (33 ms). At $N = 31$ the same window gives $3.3 \cdot 1000/31 = 106$ Hz transition: the two designs bracket the specification; the delay is a cost to record, not a footnote.

> **Worked example 2 – verify the design.** Form the 67 taps from $h[n] = 0.45\,\mathrm{sinc}(0.45(n-33))w[n]$, where $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$ and its value at zero is 1. Normalise the tap sum to 1. Evaluate $H(f) = \sum_{n=0}^{66} h[n]e^{-j2\pi fn/1000}$. On a 0.025 Hz grid, the computed passband (0–200 Hz) ranges from −0.034 to +0.0045 dB, and the largest stopband level (250–500 Hz) is about −52.3 dB. These sampled checks meet the targets of ±0.05 dB ripple and at most −43 dB stopband level; a finer grid or extremum search checks between the samples. If a design instead gives −37 dB, it fails that target: inspect its taps, cutoff, window, tap count and measurement band rather than guessing a unique cause.

Common mistakes. Designing N from the cutoff and never measuring. Choosing an even N when a Type-I (odd, symmetric) design was intended. Forgetting the $(N-1)/2$ delay in a control loop. And asking windows for optimality – they are convenient, not best.

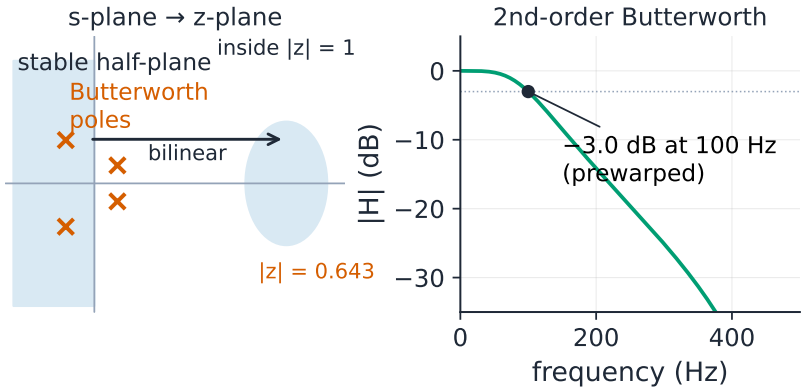
Summary. Window the ideal sinc: the sidelobe table guides window choice, $3.3f_s/N$ fixes the transition, symmetry fixes linear phase at delay $(N-1)/2$; then measure the response against every promised number.

IIR filter design

Why it matters. The same 100 Hz cutoff that costs an FIR 67 taps costs a biquad **IIR** 5 coefficients – because it inherits forty years of analog filter lore through one substitution.

The idea, with a picture. The **bilinear transform** $s = \frac{2}{T} \cdot \frac{1-z^{-1}}{1+z^{-1}}$ maps the whole analog left half-plane inside the unit circle: every stable analog filter becomes a stable digital one, automatically.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn



Stability comes free; the price is frequency warp, repaid by prewarping

The price is frequency **warp**: the analog axis $-\infty \ldots \infty$ wraps once around the unit circle, compressively, so every critical frequency must be **prewarped** $\omega_a = \frac{2}{T}\tan(\omega_d T/2)$ before use.

The derivation. Set $s = j\omega_a$ in the transform: it solves to $z = \frac{1+sT/2}{1-sT/2}$, whose magnitude is 1 exactly when $\mathrm{Re}\, s = 0$ – the analog imaginary axis maps onto the unit circle, left half-plane strictly inside. The tangent relation is the warp: near dc $\tan\theta \approx \theta$ (negligible distortion), but a digital frequency approaching Nyquist compresses toward the analog infinity. Butterworth, Chebyshev and elliptic tables transplant intact because only the frequency axis bent – the polynomial geometry survived.

> **Worked example 1 – the biquad.** A 2nd-order Butterworth low-pass at 100 Hz, $f_s = 1$ kHz: prewarp $\omega_a = 2000\tan(\pi \cdot 100/1000) = 649.8$ rad/s ✓. With s normalised by ω_a, the substitution constant is $K = 1/\tan(0.1\pi) = 3.078$ and the prototype $H(s) = 1/(s^2 + 1.4142s + 1)$ transforms to

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{14.82 - 16.94z^{-1} + 6.12z^{-2}}$$

exercises.lol exercises.com.cn

Worked example 2 – verify, then pay. DC gain: the denominator sums to $14.82 - 16.94 + 6.12 = 4.00$, so $H(1) = 4/4 = 1.000$ ✓ (Butterworth is flat at dc). The response at exactly 100 Hz evaluates to -3.0 dB ✓ – the prewarp put the -3 dB point precisely on the promised frequency, which is the entire purpose of prewarping. The poles sit at $|z| = 0.643$, inside the unit circle because the mapping guarantees it ✓. Skip the prewarp – design the analog filter at $2\pi \times 100 = 628.3$ rad/s – and the -3 dB point lands at 96.9 Hz; set $T = 1$ as well ($K = 1$) and it cuts at 250 Hz, reading a complacent -0.05 dB at 100 Hz while missing the specification entirely – the numbers always show the difference. The bill: 5 coefficients against the FIR's 67 taps, but the phase is nonlinear near the cutoff and, in fixed point, the denominator coefficients are the fragile ones – a rounding of 6.12 moves the cutoff measurably. Efficiency, phase and sensitivity trade in one line.

Common mistakes. Skipping the prewarp (the cutoff lands low – 96.9 Hz for 100 Hz at 1 kHz is small, 2.86 kHz for 4 kHz at 10 kHz is fatal). Setting $T = 1$ while still quoting analog hertz. Testing stability by luck instead of trusting the mapping. And expecting linear phase – IIR phase near its poles is the opposite of flat.

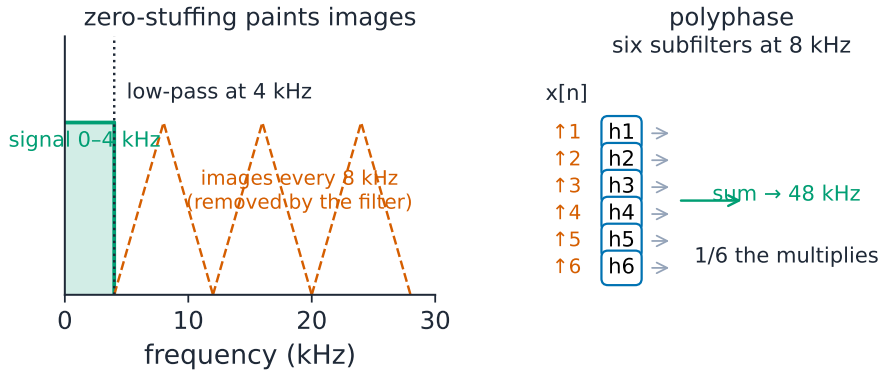
Summary. Bilinear: LHP → unit disk (stability free), frequencies warped by $\frac{2}{T}\tan(\cdot)$ (prewarp every critical frequency); the Butterworth biquad lands as $H(z) = \frac{1+2z^{-1}+z^{-2}}{14.82-16.94z^{-1}+6.12z^{-2}}$ (dc 1.000, -3.0 dB at 100 Hz, poles at 0.643).

Multirate processing

Why it matters. Sample rate is compute budget: a filter running at 48 kHz costs six times the same filter at 8 kHz. Multirate is the art of doing the work where it is cheap.

The idea, with a picture. Decimation by M (keep every Mth sample) must first **low-pass** below $f_s/(2M)$ – or the folded spectrum aliases, irreversibly. **Interpolation** by L (stuff $L-1$ zeros between samples) must then low-pass below the old Nyquist to kill the images.

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn



Upsampling paints images every 8 kHz; one low-pass erases them; polyphase does the same arithmetic for one-sixth the multiplies

The derivation. Zero-stuffing by L in time makes $X_{up}(e^{j\omega}) = X(e^{j\omega L})$ – the same spectrum replicated L times around the circle: those are the images, and they are exactly as real as the signal. Keeping 1 sample in M wraps M spectral copies onto each other: whatever lives above the new Nyquist folds back into the band. The two low-passes are not bureaucracy; each erases a mathematical certainty. **Polyphase** is then pure algebra: the interpolation filter's L phase-sets each see only every Lth input, so one filter at f_sL becomes L subfilters at f_s – identical arithmetic, $1/L$ the multiplies.

Worked example 1 – 8 kHz speech to 48 kHz. Interpolate by 6, then a low-pass at 4 kHz (the old Nyquist) removes the images at 8k, 16k, 40k… Run naively, that filter executes at 48 kHz – six times the work. Polyphase reorganises it into 6 subfilters each running at 8 kHz: the same arithmetic, 1/6 the multiplies ✓ – and this is not an approximation, it is the same computation scheduled differently.

Worked example 2 – one converter, not two. Resampling 48 kHz audio up to 72 kHz needs $L = 3$, $M = 2$ ($48 \times 3/2 = 72$ ✓). Two passes would need an interpolator, a filter, a decimator and another filter. One rational converter does it: upsample by 3 to 144 kHz, apply a single low-pass at 24 kHz (the input's own Nyquist – it stops the ↑ 3 images and the ↓ 2 aliases at once), keep every second sample. One filter, one pass, the same result.

Common mistakes. Decimating before filtering (the alias is then not noise, it is permanent). Forgetting the $1/L$ amplitude loss of zero-stuffing (compensate with a gain of L in the filter). And treating polyphase as an approximation – it is exact bookkeeping.

Summary. ↓ M needs a low-pass at $f_s/2M$ first; ↑ L needs a low-pass at the old Nyquist after; polyphase delivers the same arithmetic for $1/L$ the multiplies; rational

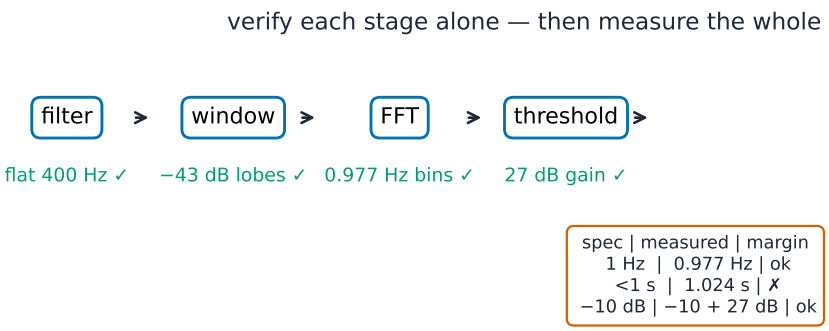
exercises.lol exercises.com.cn

L/M converters merge into one filter at the intermediate rate.

Signal-processing project

Why it matters. The course closes the way practice opens: with a specification, an implementation, and a measurement – and the honest gap between the three.

The idea, with a picture. Build a tone detector as a pipeline, each stage verified alone before the next exists: anti-alias filter → window → FFT → threshold. A defect found in a stage costs an hour; found in the integrated system, a week of bisection.



Specify first, verify each stage alone, then measure the whole against every spec line

The method. Write the specification first – a tone detector: 1 Hz bins, under 1 s latency, detection at −10 dB SNR – implement in verified stages, and report **measured performance** against every line, failures included.

> **Worked example 1 – the spec bites back.** 1 Hz bins at $f_s = 1$ kHz needs $N \geq 1000$ samples – that is already 1.0 s of data, and the spec says under 1 s latency. The conflict surfaces at design time, which is when you want it: $N = 1024$ gives 0.977 Hz ✓ but a 1.024 s record ×; $N = 1000$ gives 1.000 Hz – exactly on the line. The margin table records the trade; nothing is silent.

> **Worked example 2 – can −10 dB even be detected?** A sinusoid's coherent energy lands in one bin while white noise spreads over all $N/2$: the bin **processing gain** is $10\log_{10}(N/2)$ – at $N = 1024$ that is 27 dB ✓. An input SNR of −10 dB becomes about +17 dB in the decision bin: detectable with margin – the window costs a few dB of that gain and it still clears – and the number came from the transform, not from hope.

Common mistakes. Measuring only the happy path. Silent goals – spec lines that exist in the code but not in the report. Tuning the detector to the test tone (overfitting is lying with extra steps). And skipping per-stage checks: integration hides what verification would show.

Summary. Specify first, verify stages alone, trust measured margins; processing gain

exercises.lol exercises.com.cn
exercises.lol exercises.com.cn

$10\log_{10}(N/2)$ sets feasibility.

exercises.lol exercises.com.cn