

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

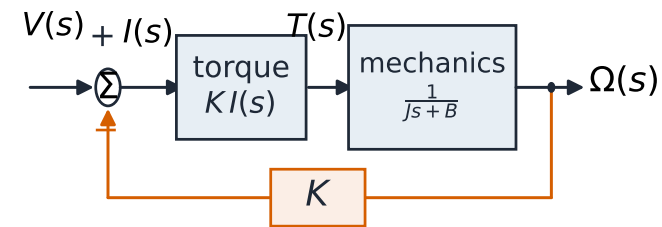
$$(\text{how fast } x \text{ changes}) = (\text{drive in}) - (\text{leakage out}).$$

A **first-order model** has the form $\tau \frac{dx}{dt} + x = K u(t)$. 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/s² 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 m³/h with h in m): the slope $\frac{dq}{dh} = 1/\sqrt{h}$ 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.

$$\begin{array}{ll} \text{torque law} & \text{Newton} \\ T = Ki & J\dot{\omega} = T - B\omega \end{array}$$



$$\text{KVL: } V = Ri + K\omega$$

$$\text{with } L \approx 0: \frac{\Omega(s)}{V(s)} = \frac{K}{(Js + B)R + K^2} = \frac{50}{s + 35}$$

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

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

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

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 $\cdot$ m², friction $B = 0.1$ N $\cdot$ m $\cdot$ s, motor constant $K = 0.5$ (in N $\cdot$ m/A, numerically the same as V $\cdot$ 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:

- Torque law: current makes torque, $T = K i$, with T in $\text{N} \cdot \text{m}$ 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 $V = Ri + K\omega$. In the s -domain, with zero initial conditions, the two laws become

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

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

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

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 $s = -35 \text{ rad/s}$. The time constant is $1/35 = 28.6 \text{ ms}$: fast. The DC gain is $50/35 = 1.43 \text{ rad/s}$ of final speed per volt. Sense check at DC: with everything constant the motor equations are pure algebra, and indeed $\frac{K}{BR + K^2} = \frac{0.5}{0.35} = 1.43$. 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: $\frac{K}{(Js + B)R} = \frac{50}{s + 10}$ 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 $h = 4 \text{ m}$ and 0.33 at $h = 9 \text{ m}$; one slope cannot serve both.
- Unit slips: J in $\text{kg} \cdot \text{m}^2$, B in $\text{N} \cdot \text{m} \cdot \text{s}$, 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.

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.

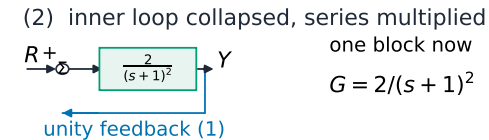
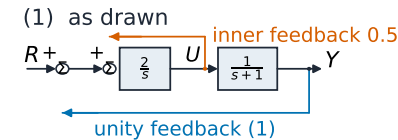
The idea. The transfer function $G(s)$ is the ratio of output to input in the s -domain, with zero initial conditions. When blocks are connected, arithmetic replaces algebra:

- 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, $T = \frac{G}{1 + GH}$.

The feedback formula deserves its one-line derivation. The error is $e = r - Hy$ and the output is $y = Ge$. Combine them: $y = G(r - Hy)$, move the GHy 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.



(3) close the outer loop: $T = G/(1 + G)$

read it: $\omega_n = \sqrt{3} = 1.73 \text{ rad/s}$, $\zeta = 0.577$

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 $G_1 = \frac{1}{s + 1}$ and $G_2 = \frac{1}{s + 2}$ in series with gain $K = 6$, under unity feedback ($H = 1$):

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

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** $\zeta = \frac{3}{2 \times 2.83} = 0.530$. That sits inside the standard design region $0.4 \leq \zeta \leq 0.7$, and unit 10.3 turns it into an overshoot of about 14%.

Worked example 2: nested loops. An inner loop $\frac{2}{s}$ with feedback gain 0.5 sits in series with $\frac{1}{s+1}$, 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 $\zeta = \frac{2}{2 \times 1.73} = 0.577$. 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 $\frac{G}{1+G}$ 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.

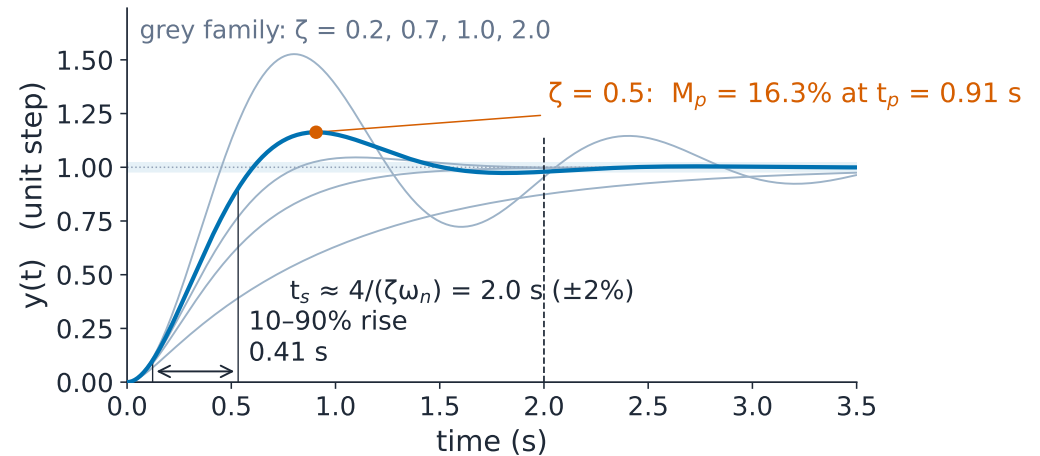
Time-domain response

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.

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

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

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

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

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**

$K_v = \lim_{s \rightarrow 0} sG(s)$ gives the ramp error $\frac{1}{K_v}$. 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:

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

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

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 $G = \frac{K}{s(s+2)}$ in unity feedback. The closed loop is $\frac{K}{s^2 + 2s + K}$, so matching the prototype gives $2\zeta\omega_n = 2$ and $\omega_n = \sqrt{K}$, hence $\zeta = 1/\sqrt{K}$. Demand $\zeta = 0.45$:

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

Predict the response: $M_p = e^{-\pi(0.45)/0.893} = 0.205$, so 20.5%; peak time $t_p = \frac{\pi}{2.22 \times 0.893} = 1.58 \text{ s}$; settling $t_s \approx \frac{4}{0.45 \times 2.22} = 4.0 \text{ s}$. Now the error constants, read off the open loop: $K_v = \frac{4.94}{2} = 2.47$, so a unit ramp leaves a steady error $\frac{1}{K_v} = 0.405$, 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.

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 $s^4 + 2s^3 + 3s^2 + 4s + 5$.

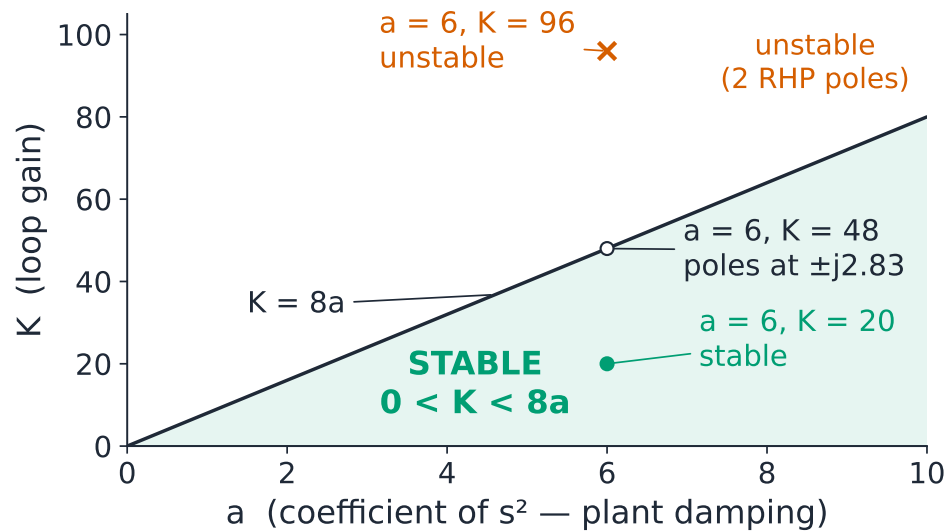
row	first column	second column	third column
s^4	1	3	5
s^3	2	4	0
s^2	$\frac{2 \cdot 3 - 1 \cdot 4}{2} = 1$	$\frac{2 \cdot 5 - 1 \cdot 0}{2} = 5$	
s^1	$\frac{1 \cdot 4 - 2 \cdot 5}{1} = -6$	0	
s^0	5		

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:

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

Stability needs $\frac{30 - K}{5} > 0$ 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.



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 $\frac{8a - K}{a}$, 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

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

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

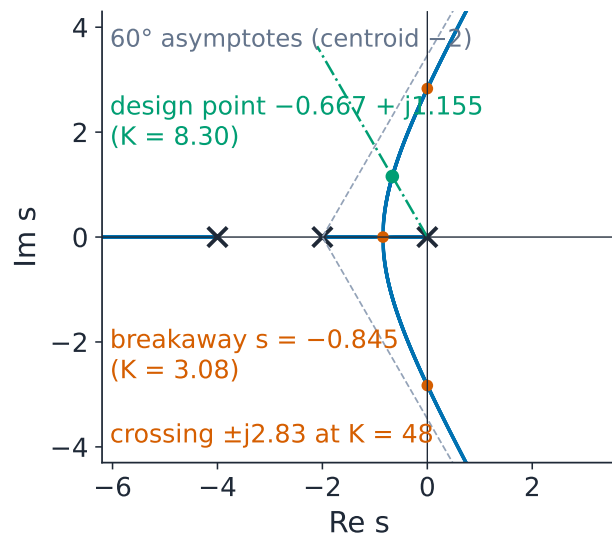
Each branch starts at an open-loop pole (at $K = 0$) and ends at a zero or at infinity. The standard sketch rules all follow from that one condition:

- **Real-axis segments:** a segment carries the locus when an odd number of poles-plus-zeros lies to its right.
- **Asymptote** angles $\frac{180^\circ(2k + 1)}{n - m}$, centred on the centroid of the poles minus the zeros.
- The **breakaway point**, where branches leave the real axis, solves $\frac{d}{ds} \left[\frac{\prod(s - p_i)}{\prod(s - z_i)} \right] = 0$, 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 $L = \frac{K}{s(s + 2)(s + 4)}$. Three poles at $0, -2, -4$ and no zeros, so $n - m = 3$ and the asymptotes leave the centroid $\frac{(0) + (-2) + (-4)}{3} = -2$ at $\pm 60^\circ$ and 180° . The real-axis locus is the segment $[-2, 0]$ and the half-line $(-\infty, -4]$. The breakaway solves

$$\frac{d}{ds}(s^3 + 6s^2 + 8s) = 3s^2 + 12s + 8 = 0,$$

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.



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 $\arccos 0.5 = 60^\circ$ 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 $120^\circ + 40.9^\circ + 19.1^\circ = 180^\circ$ exactly, so the design point is

$$s^* = -0.667 + j1.155.$$

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 $\frac{0.667}{1.334} = 0.500$, 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 .

- 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, break-away, crossing. Design means walking the locus to a damping ray and reading off the gain.

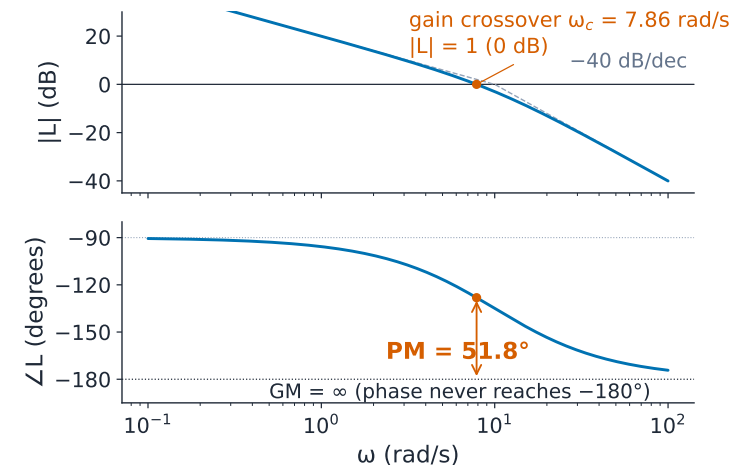
Bode and Nyquist analysis

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 $\angle L = -180^\circ$), 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.)

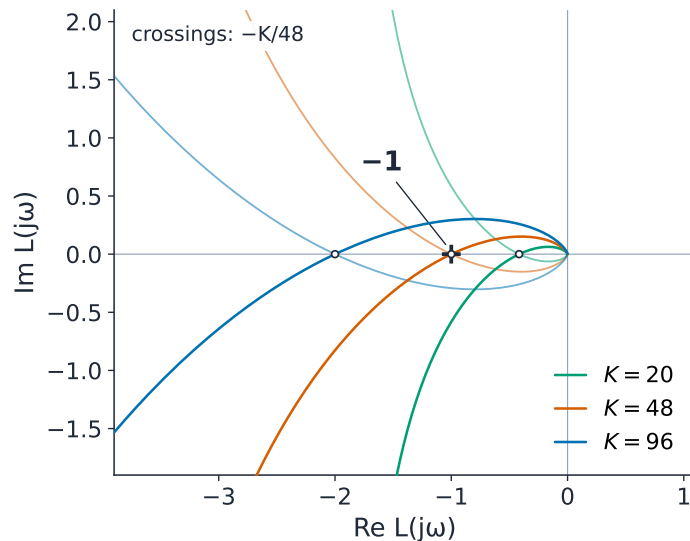


Read PM at the gain crossover; the phase never reaches -180° , so GM is infinite

Worked example 1: margins by arithmetic. Take $L = \frac{100}{s(s+10)}$. 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

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

The phase there is $-90^\circ - \arctan(7.86/10) = -90^\circ - 38.2^\circ = -128.2^\circ$, so the phase margin is $PM = 180^\circ - 128.2^\circ = 51.8^\circ$. The gain margin is infinite: the phase only approaches -180° asymptotically. Cross-check in the time domain: the closed loop is $\frac{100}{s^2 + 10s + 100}$, 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 $L = \frac{20}{s(s+2)(s+4)}$, 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 $\omega = \sqrt{8} = 2.83 \text{ rad/s}$, and there

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

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

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

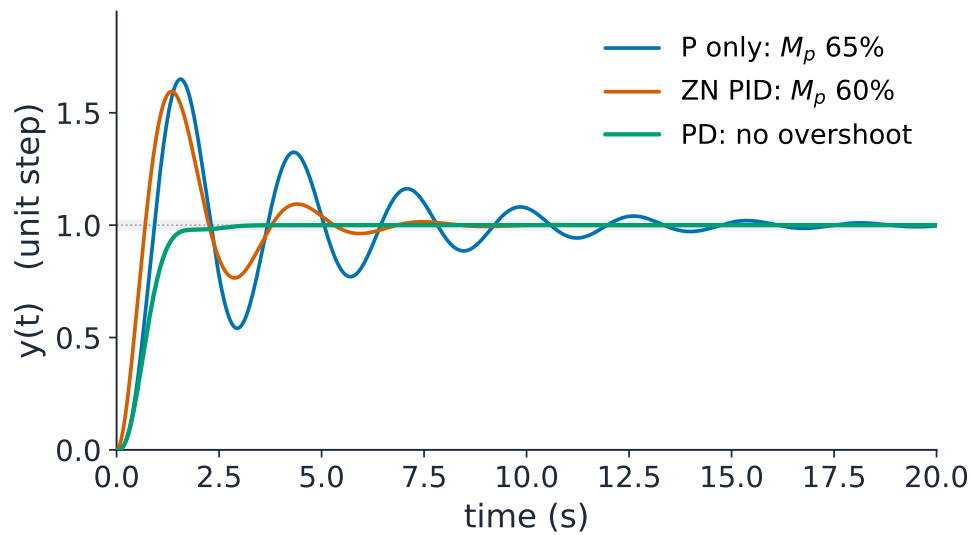
Why it matters. Most industrial loops run a PID. It is the first controller you will tune, and the one you will tune most often.

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, $u = K_p(e + \frac{1}{T_i} \int e dt + T_d \dot{e})$, 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.



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

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

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

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 $\frac{1}{K_v} = \frac{8}{K_p} = 0.278$. 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.

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 $s^3 + 6s^2 + (8 + K_p T_d)s + K_p$: 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** $\frac{1 + \tau s}{1 + \alpha \tau s}$ with $0 < \alpha < 1$ puts a zero below a pole and buys phase at one frequency. The maximum lift is

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

For a $+45^\circ$ lift, $\alpha = \frac{1 - \sin 45^\circ}{1 + \sin 45^\circ} = 0.172$. The same network also lifts the high-frequency gain by $\frac{1}{\alpha} = 5.8$, 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 $K_d = K_p T_d$.
- Designing a lead network and forgetting its $\frac{1}{\alpha}$ high-frequency gain lift, which erodes the very margin you bought.

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

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.

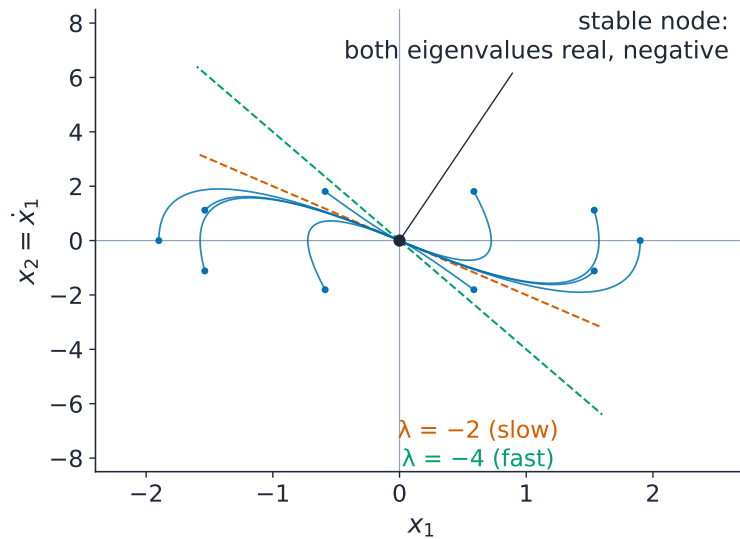
State-space introduction

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:

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

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 $I + At + \frac{(At)^2}{2!} + \dots$, which converges for every square matrix.



Trajectories straighten onto the slow eigenvector: the tail of every response is the slow pole

Worked example 1: read a model, verify its decay. Take $G(s) = \frac{8}{(s+2)(s+4)} = \frac{8}{s^2 + 6s + 8}$. In companion form,

$$A = \begin{bmatrix} 0 & 1 \\ -8 & -6 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [8 \ 0], \quad D = 0.$$

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 $[B \ AB \ \dots]$, 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 $[B \ AB \ \dots]$ loses rank, some directions are unreachable and no K_{fb} exists; check controllability first.

- 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.