System Model Representation

ODE · transfer function · state space · block diagrams · Mason's rule

Abu Haqeem

MEC524 Control Systems · Handbook Chapter 2 · Syllabus Chapter 2 · Weeks 3–5

Semester 20264

How to use this chapter

Three documents, three different jobs. Use the right one.

These slides	Where every result *comes from*, and what it means. Follow the derivations here.
The handbook	The reference. Look up a result, a rule, a worked example. Chapter 2, § 2.0–2.5.
The tutorial	Where you find out whether you actually understood it. § 2.6.

Where the ODEs come from

Here every differential equation is *given*. Chapter 3 (weeks 6–7) shows how to derive them from the physics. The Laplace transform from Chapter 1 is all you need now.

Where Chapter 2 is going

Chapter 1 gave you the ODE and the Laplace transform. This chapter asks: *once a system is described by an ODE, what form should it be in?*

1. **Transfer function** — one ratio in s; poles tell you stability. §2.1–2.2
2. **State space** — the same system as a matrix equation that keeps the inside visible. §2.3
3. **Block diagram** — many named subsystems reduced to one transfer function. §2.4
4. **Signal flow graph** — Mason's rule, for any tangle of loops. §2.5

Chapters 4 and 5 start from a single $T(s)$

Transient response, steady-state error, Bode plots: every one of them assumes you can already produce the closed-loop transfer function. This is where you learn to.

A print head on a belt

A 3D printer's print head is pulled along its rail by a toothed belt.

- the head and carriage have **mass** m;
- the belt stretches, so it acts as a **spring** k;
- the rail bearings drag, roughly a **damper** c.

The engineering question

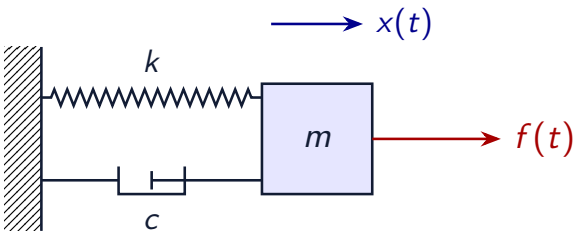
The motor applies a force $f(t)$. How does the head position $x(t)$ respond — and if we wrap a position controller around it, will it settle?

Newton's second law gives the model (Chapter 3 derives it, in week 6):

$$m\ddot{x} + c\dot{x} + kx = f(t)$$

Handbook § 2.0

The model: one mass, one spring, one damper



What a model actually does: predict the next instant

Forget the symbols for a moment. A model is a **recipe**: *tell me how things are now, and I will tell you how they will be a moment from now.*

- A weather forecast: today's pressure, wind and temperature → tomorrow's.
- Your phone's map: where you are and how fast you are moving → when you arrive.
- The print head: where it is and how fast it is moving → where it will be one millisecond later.

Where the physics comes in

Newton supplies the one missing piece — the acceleration. From the forces on the head, $a = (f - kx - c\dot{x})/m$. Speed changes by $a\,\Delta t$, position by $v\,\Delta t$. Repeat.

Every representation in this chapter is this recipe, written down differently.

Handbook § 2.0

The recipe, run by hand for three milliseconds

Print head: $m = 0.6\,\text{kg}$, $c = 6\,\text{N s/m}$, $k = 1500\,\text{N/m}$, no motor force. Released 2 mm from rest. Step $\Delta t = 1\,\text{ms}$.

t [ms]	x [mm]	v [mm/s]	$a = -(kx + cv)/m$ [m/s^2]
0	2.000	0	−5.00
1	2.000	−5.00	−4.95
2	1.995	−9.95	−4.89
3	1.985	−14.84	−4.81

Each row: $x_{\text{new}} = x + v\,\Delta t$, $v_{\text{new}} = v + a\,\Delta t$, both from the row above.

Two lessons

Exact at 3 ms: 1.978 mm. A computer takes far smaller steps, millions of them — that is *simulation*. And only x and v were carried from row to row: those two are the **state**.

numbers: Decks/ch2-anim/ch2_solve.m

The state is what the system must remember

Ask: *to predict what happens next, what must I know right now?*

Thing	Must remember	Energy stores
A mug of coffee cooling	its temperature	1 (heat)
A ball thrown upward	height *and* speed	2 (height, motion)
The print head	position *and* speed	2 (belt, mass)

Forget one and the recipe cannot run

Two print heads at the same position, one at rest and one moving at 50 mm/s, have different futures. Position alone is not a state.

One state per energy store — the same count as the order of the ODE (*How many derivatives? Count the energy stores*, two slides on).

Handbook § 2.1.2, 2.3.1

One system, four dialects

Representation	**The print head**	**Reach for it when…**
Differential equation	$m\ddot{x} + c\dot{x} + kx = f$	simulating; nonlinearity matters
Transfer function	$G(s) = \dfrac{1}{ms^2 + cs + k}$	poles, stability, loop tuning
State space	$\dot{z} = Az + Bu,\ y = Cz + Du$	many states, many inputs
Block diagram / SFG	a picture of connected parts	combining named subsystems

Equivalent means

The same output for every input, from the same initial conditions. Nothing is lost converting between them — only convenience changes.

The skill is choosing the dialect that makes *your* question easy.

Handbook § 2.0

How many derivatives? Count the energy stores

Each *independent* energy store adds one derivative to the ODE.

- mass (kinetic), spring (potential), inductor, capacitor, tank head

System	Stores	Order
RC filter, $\tau\dot{y} + y = u$	capacitor	1
Print head, $m\ddot{x} + c\dot{x} + kx = f$	mass, belt	2
Cart and pendulum, linearised	two masses, two positions	4

Order — three ways to count it

highest derivative in the ODE = degree of the TF denominator = number of state variables. They must agree: use that as a check.

Handbook § 2.1.2

Linearisation: zoom in until the curve is straight

Near an operating point (x_0, u_0), keep only the first Taylor term:

$$f(x,u) \;\approx\; f(x_0,u_0) + \left.\frac{\partial f}{\partial x}\right|_0 \Delta x + \left.\frac{\partial f}{\partial u}\right|_0 \Delta u$$

A pendulum hanging down, $\theta_0 = 0$. The slope of $\sin\theta$ there is $+1$, so $\sin\theta \approx \theta$:

$$\ddot{\theta} + \frac{g}{\ell}\sin\theta = \frac{\tau}{m\ell^2} \quad\longrightarrow\quad \ddot{\theta} + \frac{g}{\ell}\theta = \frac{\tau}{m\ell^2}$$

How small is small? At 20°, $\sin\theta$ and θ differ by about 2 %.

A linear model belongs to its operating point

Change the operating point and you get a different model — sometimes a very different one.

Handbook § 2.1.3

You have met this before: the arc rule $s = r\sin\theta \approx r\theta$

A pendulum bob of length ℓ swung to θ:

- it travels the **arc** $\ell\theta$ — exact, with θ in radians;
- it sits $\ell\sin\theta$ **sideways** of the pivot.

For small swings the two are the same length. That is all $\sin\theta \approx \theta$ says — the rule you used for levers, gears and rolling wheels in dynamics and vibrations.

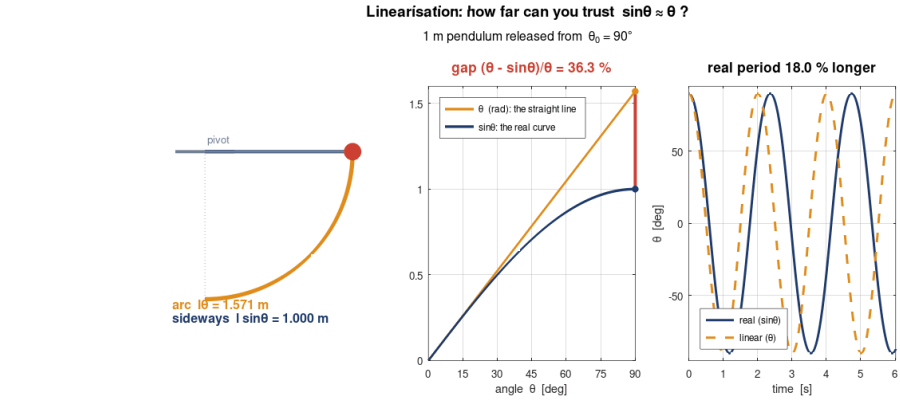
release angle	10°	20°	30°	60°	90°
gap $(\theta - \sin\theta)/\theta$	0.51 %	2.0 %	4.5 %	17.3 %	36.3 %
real period longer by	0.2 %	0.8 %	1.7 %	7.3 %	18.0 %

The engineering answer to "how small is small?"

Up to about 20° the linear model is within 2 %. At 60° it is not a model of this pendulum any more.

Handbook § 2.1.3

Watch the straight line part from the curve



A 1 m pendulum (linear period 2.006 s) released from 5° up to 90°. Left: arc $\ell\theta$ against sideways $\ell \sin\theta$.

Middle: the curve and its tangent. Right: the real swing falls behind the linear one. GIF:

`ch2-anim/MEC524-ch2-pend.gif`.

Same pendulum, balanced upright

Operating point $\theta_0 = \pi$. Measure the small deviation $\phi = \theta - \pi$.

Step 1 — the slope has changed sign.

$$\sin(\pi + \phi) = -\sin\phi \approx -\phi$$

Step 2 — linear ODE, then Laplace.

$$\ddot{\phi} - \frac{g}{\ell}\,\phi = \frac{\tau}{m\ell^2} \quad \Longrightarrow \quad G(s) = \frac{1}{m\ell^2\,(s^2 - g/\ell)}$$

Step 3 — read the poles. $s = \pm\sqrt{g/\ell}$. One is in the right half-plane: the upright pendulum is open-loop *unstable*.

Why a self-balancing robot needs a controller

Same hardware, same physics, different operating point — and the model changes from "oscillates" to "falls over".

Handbook § 2.6.1

From ODE to $G(s)$: why derivatives become powers of s

The Laplace transform of a derivative:

$$\mathcal{L}\{\dot{y}\} = sY(s) - y(0), \qquad \mathcal{L}\{\ddot{y}\} = s^2 Y(s) - s\,y(0) - \dot{y}(0)$$

Set every initial condition to zero: each d^k/dt^k becomes s^k.

$$(a_n s^n + \cdots + a_1 s + a_0)\, Y(s) = (b_m s^m + \cdots + b_0)\, U(s)$$

$$\boxed{G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + \cdots + b_0}{a_n s^n + \cdots + a_0}}$$

Why zero initial conditions?

With them, $G(s)$ depends only on the coefficients — the plant — not on the input or on how the experiment started. That is what makes it a *property of the system*.

Handbook § 2.2.1

Worked example — ODE to transfer function (1 of 2)

Find $Y(s)/U(s)$ and classify the system:

$$\dddot{y} + 12\,\ddot{y} + 32\,\dot{y} = 7\,\dot{u} + 32\,u$$

Step 1 — transform with zero initial conditions.

$$(s^3 + 12s + 32)\,Y(s) = (7s + 32)\,U(s)$$

There is no s^2 term, because there is no $\ddot{y}$. Do not invent one.

Step 2 — divide.

$$G(s) = \frac{7s + 32}{s^3 + 12s + 32}$$

Step 3 — classify by reading, not computing. Order 3 (denominator degree) · relative degree $3 - 1 = 2$ · type 0 (no factor of s below) · DC gain $G(0) = 32/32 = 1$.

Handbook § 2.2.5

Why the poles decide stability

Expand any output in partial fractions and invert term by term:

$$Y(s) = \sum_i \frac{r_i}{s - p_i} \quad \Longrightarrow \quad y(t) = \sum_i r_i \, e^{p_i t}$$

Every pole contributes its own $e^{p_i t}$:

- $\operatorname{Re} p_i < 0$ — the term dies away;
- $\operatorname{Re} p_i = 0$ — it persists (marginal);
- $\operatorname{Re} p_i > 0$ — it grows without bound.

Stability from pole location

Stable if and only if *every* pole is in the open left half-plane. One right-half-plane pole ruins it; the others cannot compensate.

Handbook § 2.2.4

Worked example — ODE to transfer function (2 of 2)

Step 4 — find a root. Try small integers in $s^3 + 12s + 32$. At $s = -2$: $-8 - 24 + 32 = 0$. ✓

Step 5 — divide it out.

$$s^3 + 12s + 32 = (s+2)(s^2 - 2s + 16) \quad \Longrightarrow \quad s = -2, \qquad s = 1 \pm j\sqrt{15} = 1 \pm j3.87$$

Step 6 — conclude. Two poles have real part $+1$: the system is **unstable**.

A DC gain of 1 does not mean it settles at 1

$G(0)$ predicts the steady state *only if the system is stable*. Here the output grows without bound, and the final-value theorem does not apply. Find the poles before you quote a steady-state value.

Handbook § 2.2.4–2.2.5

Why a transfer function is not enough

- A quadrotor: four rotor speeds in, six motion coordinates out. As transfer functions that is a grid of 24, and none of them shows what is happening inside.
- Even for the print head, $G(s)$ is a black box: input and output only.

State

The smallest set of variables that, together with the future input, fixes all future behaviour.

For the print head: **position and velocity**. Position alone does not tell you whether the head is about to hit the end stop; velocity alone does not tell you where it is.

Handbook § 2.3.1

Building a state model from the ODE

$m\ddot{x} + c\dot{x} + kx = f$. Choose $z_1 = x$ and $z_2 = \dot{x}$.

$\dot{z}_1 = z_2$ the definition of z_2

$\dot{z}_2 = \ddot{x} = -\frac{k}{m}z_1 - \frac{c}{m}z_2 + \frac{1}{m}f$ the ODE, solved for its highest derivative

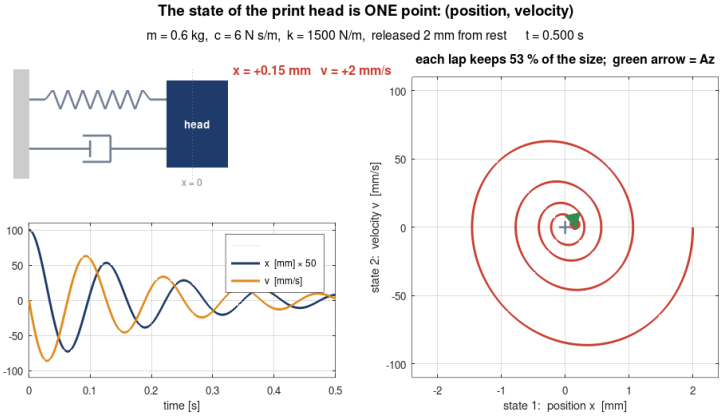
$$\dot{z} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix}}_{A} z + \underbrace{\begin{bmatrix} 0 \\ 1/m \end{bmatrix}}_{B} f, \qquad x = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} z, \quad D = 0$$

The recipe

One state per derivative *below* the highest. Every row but the last is a definition; the last row is the ODE.

Handbook § 2.3.2–2.3.3

Watch the state move: one point, one spiral



The print head from the hand-run recipe, released 2 mm from rest. Right: $z_1 = x$ across, $z_2 = \dot{x}$ up. The whole present is one point; the green arrow is $\dot{z} = Az$, where it goes next. GIF: `ch2-anim/MEC524-ch2-state.gif`.

Where $C(sI - A)^{-1}B + D$ comes from

Laplace-transform both equations with zero initial conditions:

$$sZ(s) = AZ(s) + BU(s), \qquad Y(s) = CZ(s) + DU(s)$$

Collect Z. Write sI, not s — you cannot subtract a matrix from a scalar:

$$(sI - A)\,Z(s) = BU(s) \quad \Longrightarrow \quad Z(s) = (sI - A)^{-1}B\,U(s)$$

Substitute into the output equation:

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

Where the poles went

$(sI - A)^{-1} = \operatorname{adj}(sI - A)/\det(sI - A)$. So the denominator of $G(s)$ is $\det(sI - A)$: **the poles are the eigenvalues of A.**

Handbook § 2.3.3

Reading the spiral from A

For the print head, $m = 0.6$, $c = 6$, $k = 1500$:

$$A = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2500 & -10 \end{bmatrix}, \qquad \det(sI - A) = s^2 + 10s + 2500$$

Eigenvalues: $s = -5 \pm j\,49.75$ — the poles of $G(s)$, as the last slide promised.

Imaginary part $49.75\,\mathrm{rad/s}$
how fast the point goes round: one lap every $2\pi/49.75 = 0.126\,\mathrm{s}$.

Real part $-5\,\mathrm{s}^{-1}$
how fast it falls inward: each lap keeps $e^{-5\times 0.126} = 53\,\%$ of its size.

So $\zeta = 0.10$ and $\omega_n = 50\,\mathrm{rad/s}$, seen as a picture

A real part of 0 would be a circle traced for ever; a positive real part, an outward spiral — unstable. Chapter 4 reads these numbers off poles.

Handbook § 2.3.3

Worked example — transfer function to state space

Convert $G(s) = \dfrac{4}{s^2 + 3s + 2}$.

Step 1 — back to the ODE. $\ddot{y} + 3\dot{y} + 2y = 4u$

Step 2 — states $x_1 = y$, $x_2 = \dot{y}$. $\dot{x}_1 = x_2$, $\dot{x}_2 = -2x_1 - 3x_2 + 4u$

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0$$

Step 3 — check, don't trust. $\det(sI - A) = s^2 + 3s + 2$, so the eigenvalues are -1 and -2 — the poles of $G(s)$. And $C(sI - A)^{-1}B$ returns $4/(s^2 + 3s + 2)$. ✓

The 4 could equally sit in $C = [4 \;\; 0]$ with $B = [0 \;\; 1]^T$. A state model is not unique; the transfer function is.

Handbook § 2.3.3

Worked example — a numerator mixes the states

Convert $G(s) = \dfrac{s^2 + 7s + 2}{s^3 + 9s^2 + 26s + 24}$.

Step 1 — split the fraction into two blocks in series.

$$U \;\to\; \boxed{\frac{1}{s^3 + 9s^2 + 26s + 24}} \;\to\; X_1 \;\to\; \boxed{s^2 + 7s + 2} \;\to\; Y$$

Step 2 — the denominator block, companion form, $z_1 = x_1$, $z_2 = \dot{x}_1$, $z_3 = \ddot{x}_1$:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -24 & -26 & -9 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Step 3 — the numerator block just reads the states.

$y = \ddot{x}_1 + 7\dot{x}_1 + 2x_1 = 2z_1 + 7z_2 + z_3$, so $C = [2 \;\; 7 \;\; 1]$, $D = 0$.

Last row of A: denominator backwards and negated. C: numerator backwards. Eigenvalues $-2, -3, -4$.

Handbook § 2.6.2

Worked example — state space to transfer function (1 of 2)

Find $G(s)$ for $A = \begin{bmatrix} -2 & 0 \\ K & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 1 \end{bmatrix}$, $D = 0$.

Step 1 — form $sI - A$.

$$sI - A = \begin{bmatrix} s+2 & 0 \\ -K & s-1 \end{bmatrix}$$

Step 2 — the 2×2 inverse: swap the diagonal, negate the off-diagonal, divide by the determinant.

$$(sI - A)^{-1} = \frac{1}{(s+2)(s-1)} \begin{bmatrix} s-1 & 0 \\ K & s+2 \end{bmatrix}$$

Step 3 — the poles are already visible. $\det(sI - A) = (s+2)(s-1)$: eigenvalues -2 and $+1$, *whatever K is*.

Handbook § 2.3.3

Worked example — state space to transfer function (2 of 2)

Step 4 — multiply right to left. B picks the first column:

$$(sI - A)^{-1}B = \frac{1}{(s+2)(s-1)} \begin{bmatrix} s-1 \\ K \end{bmatrix}$$

Step 5 — $C = [1 \;\; 1]$ adds the two rows.

$$\boxed{G(s) = \frac{s + K - 1}{(s+2)(s-1)}}$$

Step 6 — test a special value. At $K = 0$, A is diagonal and only z_1 is driven, so G must be $1/(s+2)$. The formula gives $(s-1)/\big((s+2)(s-1)\big) = 1/(s+2)$. ✓

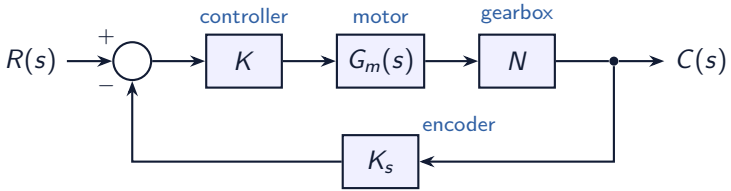
What $G(s)$ just hid

At $K = 0$ the $(s-1)$ cancels and $G(s)$ looks stable — but A still has eigenvalue $+1$: z_2 grows from any nonzero start.

Handbook § 2.3.3; 2.6 Problem 5

A position servo, drawn as blocks

A camera pan axis: a controller drives a motor, a gearbox turns the camera, and an encoder reports the angle back.



Every block comes from a different place: K you design, $G_m(s)$ from the motor model of Chapter 3 (week 7), N from the drawing, K_s from the encoder calibration.

The engineering question

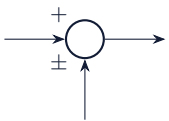
The reference steps to a new angle. What does the camera do? Before Chapter 4 can answer, we need **one** transfer function $C(s)/R(s)$.

Handbook § 2.4.1

Three primitives: block, summing junction, pickoff



Block



Summing junction



Pickoff point

Label every wire; series and parallel follow

The habit: name every summer output E_i and every block output V_i. A pickoff carries the *same* signal on both branches. Then write one equation per label.

Series. $V = G_1 X$ and $Y = G_2 V$, so

$$Y = (G_1 G_2)\, X \qquad \text{blocks in series multiply}$$

Parallel. Both blocks see X; the summer combines them:

$$Y = G_1 X \pm G_2 X = (G_1 \pm G_2)\, X \qquad \text{blocks in parallel add, with the summer's sign}$$

Both are plain substitution

Feedback is the first rule where the output appears on both sides of its own equation — so you have to *solve*, not just substitute.

Handbook § 2.4.2

Why feedback gives G/(1+GH)

Label the summer output E. One equation per label:

$$E = R - HC \quad \text{(summer)}, \qquad C = GE \quad \text{(block)}$$

Substitute the first into the second:

$$C = G(R - HC) = GR - GHC$$

C is on both sides. Collect it:

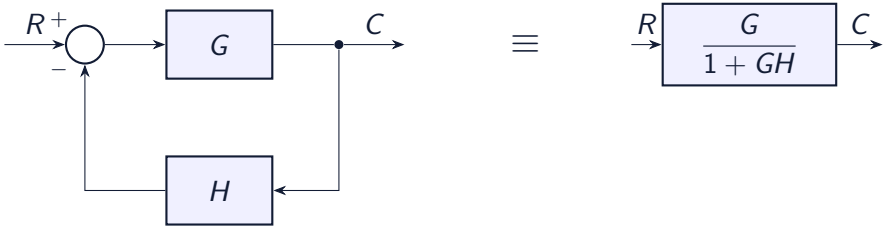
$$C\,(1 + GH) = GR \quad \Longrightarrow \quad \boxed{\frac{C}{R} = \frac{G}{1 + GH}}$$

A $+$ at the summer gives $G/(1 - GH)$. The $+$ in the negative-feedback formula comes *from the minus at the summer.*

The servo: $G = KG_mN$, $H = K_s$, so $\dfrac{C}{R} = \dfrac{KG_mN}{1 + KG_mNK_s}$.

Handbook § 2.4.2

Rule 3 in one picture



The camera servo with numbers

Give the pan servo of the earlier slide real blocks: $K = 4\,\mathrm{V/rad}$, $G_m(s) = \dfrac{20}{s(0.1s+1)}$ (motor angle per volt), $N = 0.1$, $K_s = 1$.

Step 1 — the forward path is three blocks in series: multiply.

$$G = K\,G_m\,N = \frac{4 \times 20 \times 0.1}{s(0.1s+1)} = \frac{80}{s(s+10)}$$

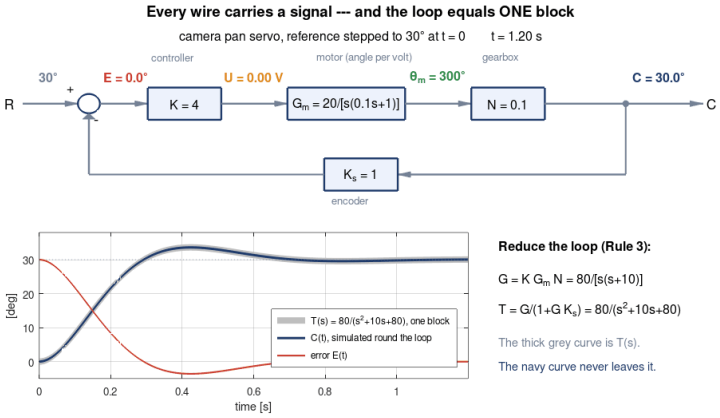
Step 2 — close the loop with Rule 3, $H = K_s = 1$:

$$\frac{C}{R} = \frac{G}{1+GH} = \frac{80}{s(s+10)+80} = \boxed{\frac{80}{s^2+10s+80}}$$

Step 3 — check, don't trust. $T(0) = 80/80 = 1$: a 30° command ends at 30°, the motor having turned $30/0.1 = 300°$. Poles $-5 \pm j\,7.42$: stable.

Handbook § 2.4.2

Watch every wire during one step



Reference stepped to 30°. The controller starts at $U = 4 \times 0.524 = 2.09\,\text{V}$ and the error dies away. Simulated round the loop, the camera peaks at 33.6° near 0.42 s and settles at 30° — on top of the single block $T(s)$ the whole time. GIF: `ch2-anim/MEC524-ch2-servo.gif`.

Moving a summer or a pickoff

Sometimes no rule fits until you move a junction. The rule for moving it: *every signal must stay the same.*

Summer forward past G. Before: $Y = G(X + U)$. Expand: $Y = GX + GU$. The moved input must pass through G too.

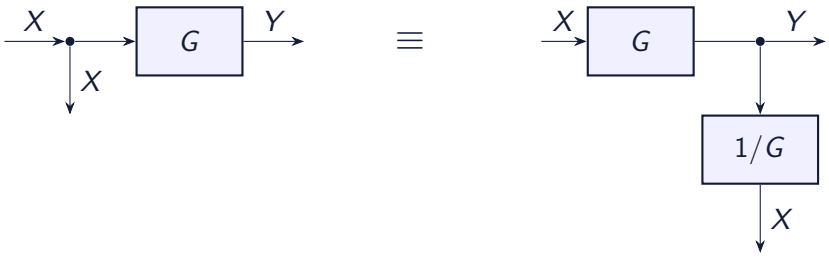
Pickoff forward past G. The tap used to carry X; after the block the wire carries GX. To get X back the branch needs $1/G$.

	forward past G	**backward past G**
summer: moved input	$\times G$	$\times 1/G$
pickoff: moved branch	$\times 1/G$	$\times G$

Summers and pickoffs go opposite ways.

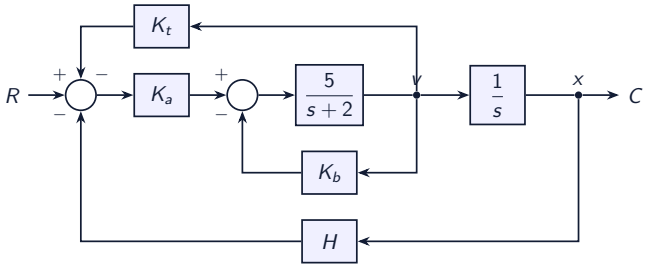
Handbook § 2.4.3

Move C: a pickoff moved forward past G



A workshop gate with three feedback wires

A sliding gate at the workshop door. Drive amplifier $K_a = 5$; motor and gate $G_m = 5/(s+2)$, gate speed v per volt; back emf $K_b = 0.2$ into the drive's summer; tacho $K_t = 0.2$ (from v) and position sensor $H = 1$ (from x) into the controller's summer.



Reason before you calculate

The K_b loop is innermost: close it first. The K_t and H loops enter one summer but leave from *different* wires, v and x: no rule fits. Move the tacho pickoff past $1/s$ so both leave from x.

Handbook § 2.4.3

Reducing the gate, one rule at a time

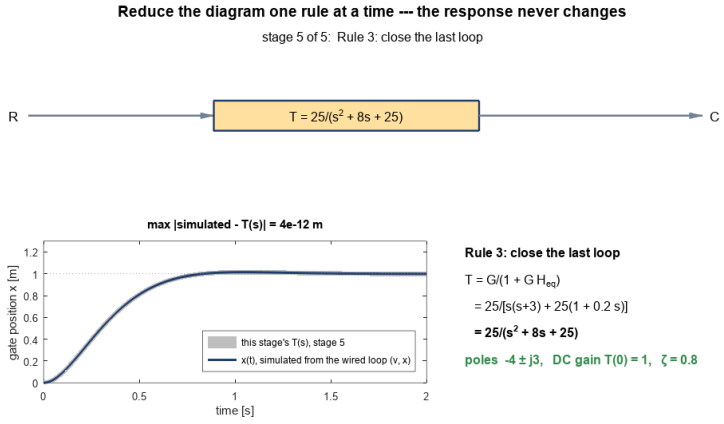
1. **Rule 3 on the back-emf loop:** $G_i = \dfrac{G_m}{1 + G_m K_b} = \dfrac{5}{s + 2 + 1} = \dfrac{5}{s + 3}$
2. **Move C, tacho pickoff forward past** $1/s$: the branch gains $1/G$, so $K_t/\frac{1}{s} = 0.2s$, now taken from x.
3. **Rule 2, the two branches in parallel:** $H + K_t s = 1 + 0.2s$.
4. **Rule 1, series:** $G = K_a\, G_i\, \dfrac{1}{s} = \dfrac{25}{s(s+3)}$
5. **Rule 3 on the last loop:**

$$\frac{C}{R} = \frac{25}{s(s+3) + 25(1 + 0.2s)} = \boxed{\frac{25}{s^2 + 8s + 25}}$$

Check: $T(0) = 25/25 = 1$, so the gate stops where it was sent. Poles $-4 \pm j3$, $\zeta = 0.8$. Forget the $\times s$ in step 2 and the loop becomes $25/(s^2 + 3s + 30)$: DC gain 0.83 — a different machine.

Handbook § 2.4.3

Watch the response survive every move



Grey: the step response of the diagram at each stage, from its own blocks. Navy: the gate as wired, simulated from its two states v and x. They coincide at all five stages; peak 1.015 m near 1.05 s. Red dashed: the move without its $\times s$. GIF: `ch2-anim/MEC524-ch2-reduce.gif`.

Two strategies for a multi-loop diagram

1. Rule-based reduction

- close the innermost loop (Rule 3);
- collapse series and parallel;
- repeat, working outward.

Use when loops nest cleanly inside each other.

2. Node labelling

- label every E_i and V_i;
- one equation per label;
- eliminate until only R and C remain.

Use when one pickoff feeds two summers, or there is a feedforward path.

Not sure which? Label the nodes

You lose nothing, and the labels are exactly the nodes of the signal flow graph in § 2.5.

Handbook § 2.4.4

Worked example 1 — nested feedback

The diagram, in words. The forward path is $R \to \Sigma_1 \to G_1 \to \Sigma_2 \to G_2 \to G_3 \to C$.

- H_1 takes the output of G_2 back to Σ_2, negative;
- H_2 takes C back to Σ_1, negative.

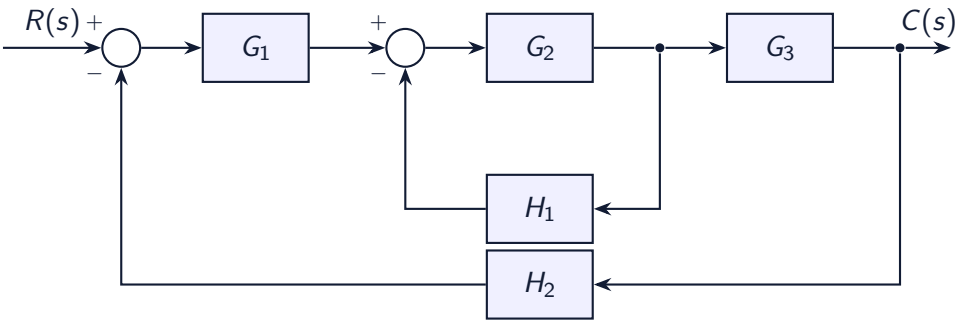
Find $C(s)/R(s)$.

Reason before you calculate

Which loop is innermost? The H_1 loop contains only G_2 and has no other loop inside it. Close it first. The H_2 loop contains everything, so it is closed last.

Handbook § 2.4.5

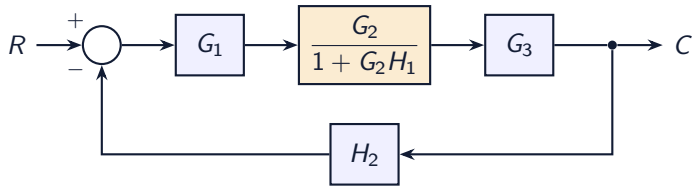
Worked example 1 — the diagram



Worked example 1 — step 1: close the inner loop

Rule 3 on G_2 with feedback H_1 replaces that loop with one block:

$$G_{\text{in}} = \frac{G_2}{1 + G_2 H_1}$$

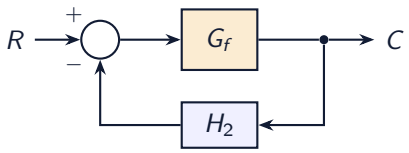


What is left: three blocks in series, inside one loop. Both of those have a rule.

Handbook § 2.4.5

Worked example 1 — steps 2 and 3: series, then the outer loop

Step 2 — series (Rule 1). $G_f = G_1 \cdot \dfrac{G_2}{1 + G_2 H_1} \cdot G_3 = \dfrac{G_1 G_2 G_3}{1 + G_2 H_1}$



Step 3 — Rule 3, then clear the fraction (multiply top and bottom by $1 + G_2 H_1$):

$$\frac{C}{R} = \frac{G_f}{1 + G_f H_2} = \frac{\dfrac{G_1 G_2 G_3}{1 + G_2 H_1}}{1 + \dfrac{G_1 G_2 G_3 H_2}{1 + G_2 H_1}} = \boxed{\frac{G_1 G_2 G_3}{1 + G_2 H_1 + G_1 G_2 G_3 H_2}}$$

Not $(1 + G_2 H_1)(1 + G_1 G_2 G_3 H_2)$: the two loops share G_2, so each adds *one* term.

Handbook § 2.4.5

Worked example 1 — cross-check by node labelling

Node	Equation
Σ_1	$E_1 = R - H_2 C$
G_1	$V_1 = G_1 E_1$
Σ_2	$E_2 = V_1 - H_1 V_2$
G_2	$V_2 = G_2 E_2$
G_3	$C = G_3 V_2$

Eliminate V_2. $V_2 = G_2(V_1 - H_1 V_2)$, so

$$V_2(1 + G_2 H_1) = G_2 V_1$$

Push through G_3 and G_1.

$$C = \frac{G_1 G_2 G_3}{1 + G_2 H_1} E_1$$

Close with $E_1 = R - H_2 C$.

$$C(1 + G_2 H_1) = G_1 G_2 G_3 (R - H_2 C)$$

Same answer, same steps

The graphical moves and the algebra are one calculation written two ways.

Handbook § 2.4.5

Worked example 2 — a feedforward path

The diagram, in words. $R \to \Sigma_1 \to G_1 \to \Sigma_2 \to G_2 \to \Sigma_3 \to G_3 \to C$.

- the output of G_1 *also* goes straight to Σ_3 $(+)$;
- H_1 takes C back to Σ_2 $(-)$; H_2 takes C back to Σ_1 $(-)$.

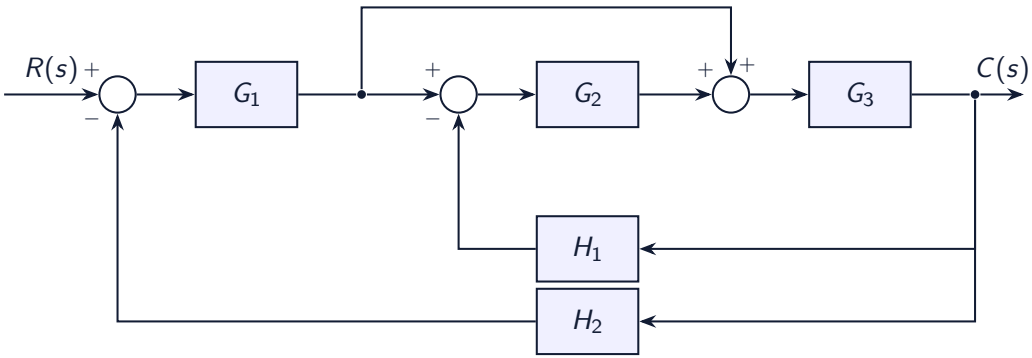
Why rule-based reduction is painful here

The pickoff after G_1 feeds two summers. No loop can be closed without first moving something, and every move adds a G or $1/G$ block to track.

So: **node labelling**. One equation per label, and let the algebra find the order.

Handbook § 2.4.6

Worked example 2 — the diagram



Worked example 2 — one equation per node

Node	Equation	Reads as
Σ_1 output	$E_1 = R - H_2 C$	reference minus outer feedback
G_1 output (pickoff)	$V_1 = G_1 E_1$	
Σ_2 output	$E_2 = V_1 - H_1 C$	
G_2 output	$V_2 = G_2 E_2$	
Σ_3 output	$E_3 = V_2 + V_1$	main path plus feedforward
G_3 output	$C = G_3 E_3$	

V_1 appears twice — in E_2 and in E_3

One pickoff, one signal, one symbol. Giving the feedforward branch a new variable is how the extra term gets lost.

Handbook § 2.4.6

Worked example 2 — eliminate, then collect C

Step 1 — expand E_3. $E_3 = G_2(V_1 - H_1 C) + V_1 = (1 + G_2)\, V_1 - G_2 H_1 C$

Step 2 — substitute $C = G_3 E_3$ and $V_1 = G_1(R - H_2 C)$.

$$C = G_3\big[(1 + G_2)\, G_1(R - H_2 C) - G_2 H_1 C\big]$$

Step 3 — every C to the left.

$$C\big[1 + G_2 G_3 H_1 + G_1 G_3 (1 + G_2) H_2\big] = G_1 G_3 (1 + G_2)\, R$$

$$\boxed{\frac{C}{R} = \frac{G_1 G_3 (1 + G_2)}{1 + G_2 G_3 H_1 + G_1 G_3 (1 + G_2) H_2}}$$

Special-value check. $G_2 = 0$ kills the middle path and the H_1 loop, leaving one loop: $G_1 G_3/(1 + G_1 G_3 H_2)$. The boxed result agrees. ✓

Handbook § 2.4.6

From block diagram to signal flow graph

A signal flow graph keeps only what the algebra needs.

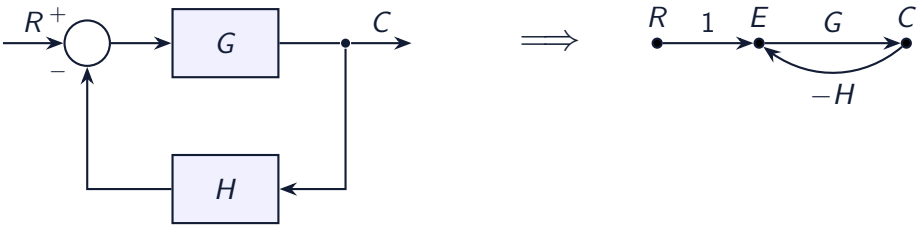
Block diagram	**Signal flow graph**
a signal (wire)	a **node**
a block G	a **branch** of gain G
a summer	a node; its signs move *onto the incoming branches* $(-H)$
a pickoff	a node
a plain wire between nodes	a branch of gain 1

No algebra, no moves

The conversion is mechanical. The labels E_i, V_i you wrote in § 2.4 are already the nodes.

Handbook § 2.5.2, 2.5.5

One loop, drawn both ways



Where Mason's formula comes from

One loop. Loop gain $L = -GH$ (the sign lives in L). Rule 3 reads

$$\frac{C}{R} = \frac{P}{1-L} \qquad \text{so} \qquad \Delta = 1 - L = 1 + GH$$

Two loops that share no node act independently, like two closed loops in series:

$$\Delta = (1 - L_1)(1 - L_2) = 1 - L_1 - L_2 + L_1 L_2$$

Two loops that share a node cannot be separated: no product term, $\Delta = 1 - L_1 - L_2$.

The general pattern

$$\frac{C}{R} = \frac{1}{\Delta} \sum_k P_k \Delta_k, \qquad \Delta = 1 - \sum L_i + \sum_{\text{non-touching pairs}} L_i L_j - \sum_{\text{triples}} L_i L_j L_k + \cdots$$

Δ_k is Δ with every loop that touches path k deleted.

Handbook § 2.5.4

Paths, loops, and what touching means

- **Forward path** — source to sink, no node visited twice. Gain P_k: the product of its branch gains.
- **Loop** — back to where it started, no node visited twice. Gain L_i: the product of its branch gains, *signs included*.
- **Touching** — two loops (or a loop and a path) that share *at least one node*. Sharing a node without sharing a branch still counts.

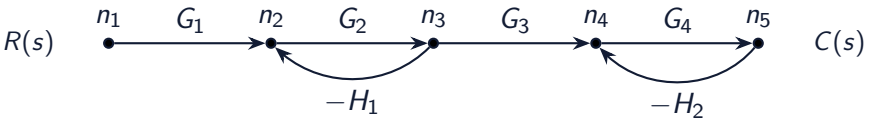
Decide by listing nodes, never by looking

Write each loop's node set and intersect them. One common node means touching, and touching means no product term.

The workflow: paths $\rightarrow$ loops $\rightarrow$ non-touching sets $\rightarrow \Delta \rightarrow$ each $\Delta_k \rightarrow$ assemble.

Handbook § 2.5.3

SFG example 1 — the graph



SFG example 1 — paths and loops

Five nodes $n_1 \dots n_5$ in a chain with gains $G_1 \dots G_4$; feedback $-H_1$ from n_3 to n_2, and $-H_2$ from n_5 to n_4.

Step 1 — forward paths. Only one: $P_1 = G_1 G_2 G_3 G_4$

Step 2 — loops.

$$L_1 : \; n_2 \to n_3 \to n_2, \;\; L_1 = -G_2 H_1 \qquad L_2 : \; n_4 \to n_5 \to n_4, \;\; L_2 = -G_4 H_2$$

Step 3 — touching? L_1 visits $\{n_2, n_3\}$, L_2 visits $\{n_4, n_5\}$. No common node: a **non-touching pair**.

Predict before you assemble

This is the "two independent loops" case from the derivation, so Δ should factorise.

Handbook § 2.5.6

Step 4 — the determinant.

$$\Delta = 1 - (L_1 + L_2) + L_1 L_2 = 1 + G_2 H_1 + G_4 H_2 + G_2 G_4 H_1 H_2 = (1 + G_2 H_1)(1 + G_4 H_2)$$

It factorises, as predicted. ✓

Step 5 — the co-factor. P_1 visits every node, so both loops touch it: $\Delta_1 = 1$.

Step 6 — assemble.

$$\boxed{\frac{C}{R} = \frac{P_1 \Delta_1}{\Delta} = \frac{G_1 G_2 G_3 G_4}{1 + G_2 H_1 + G_4 H_2 + G_2 G_4 H_1 H_2}}$$

The $L_1 L_2$ term is the one that goes missing

Forget it and the answer still "looks" reasonable. Only the node-set check in Step 3 tells you it belongs.

Handbook § 2.5.6

SFG example 2 — the feedforward diagram, by Mason

Take Worked example 2 of § 2.4 and convert it with the recipe:

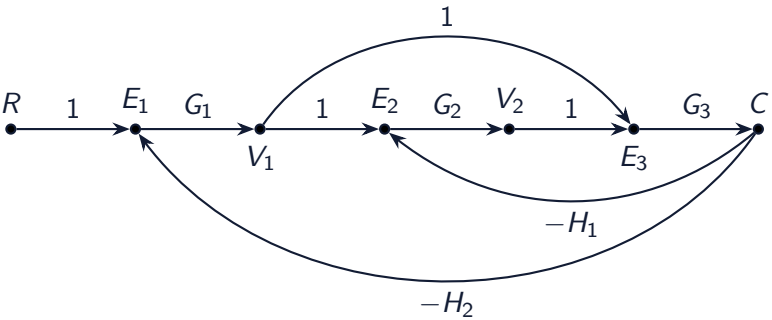
- nodes $R, E_1, V_1, E_2, V_2, E_3, C$;
- feedforward: a branch of gain 1 from V_1 to E_3;
- feedback branches $-H_1$ ($C \to E_2$) and $-H_2$ ($C \to E_1$).

Predict before you count

The feedforward branch opens a **second forward path**. And because H_2 returns to E_1, the signal can come back round *through* that branch too — expect an extra loop.

Handbook § 2.5.8

SFG example 2 — the graph



SFG example 2 — paths and loops

Forward paths.

$$P_1 : R\,E_1 V_1 E_2 V_2 E_3\,C = G_1 G_2 G_3, \qquad P_2 : R\,E_1 V_1 E_3\,C = G_1 G_3$$

Loops.

$$\begin{aligned} L_1 &: E_2 V_2 E_3 C, & L_1 &= -G_2 G_3 H_1 \\ L_2 &: E_1 V_1 E_2 V_2 E_3 C, & L_2 &= -G_1 G_2 G_3 H_2 \\ L_3 &: E_1 V_1 E_3 C, & L_3 &= -G_1 G_3 H_2 \end{aligned}$$

Touching. All three loops contain C: no non-touching pairs.

L_3 is the loop that gets missed

The H_2 feedback closes around the feedforward branch as well as the main path. Three loops, not two.

Handbook § 2.5.8

SFG example 2 — assemble

Determinant.

$$\Delta = 1 - (L_1 + L_2 + L_3) = 1 + G_2G_3H_1 + G_1G_3(1 + G_2)H_2$$

Co-factors. P_1 visits every node: $\Delta_1 = 1$. P_2 visits $\{R, E_1, V_1, E_3, C\}$ and every loop contains C: $\Delta_2 = 1$.

Assemble.

$$\frac{C}{R} = \frac{P_1 + P_2}{\Delta} = \boxed{\frac{G_1G_3(1 + G_2)}{1 + G_2G_3H_1 + G_1G_3(1 + G_2)H_2}}$$

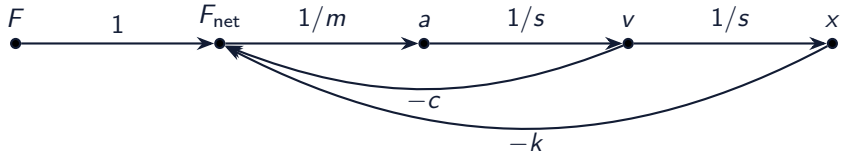
Same as node labelling

The feedforward cost one extra numerator term and one extra loop. No junction had to be moved.

Handbook § 2.5.8

Back to the print head: the graph is the state model

Newton as a graph: net force → acceleration → velocity → position; spring and damper forces fed back.



One path, $P_1 = 1/(ms^2)$. Loops $-c/(ms)$ and $-k/(ms^2)$ both pass through F_{net}: touching.

$$\frac{X}{F} = \frac{1/(ms^2)}{1 + \dfrac{c}{ms} + \dfrac{k}{ms^2}} = \frac{1}{ms^2 + cs + k}$$

The two $1/s$ branches *are* the two states $z_1 = x$, $z_2 = v$: one system, four dialects.

Handbook § 2.5.4, 2.6.4

What the four animations warn you about

1. **Linearising and then using it far away.**
   $\sin\theta \approx \theta$ is 2 % out at 20° and 17.3 % at 60°. State the operating point and the range you trust the model over.
2. **A "state" that cannot predict.**
   Choosing $z = x$ alone for a mass–spring–damper. Count energy stores: two stores, two states, A is 2×2.
3. **Reading the wrong wire.**
   In the servo the motor turns 300° while the camera turns 30°. Say which signal C is, and put every block — gearbox and sensor included — into G and H.
4. **A move without its factor.**
   Moving the gate's tacho pickoff past $1/s$ without the $\times s$ turns DC gain 1 into 0.83. Every move keeps every signal.

animations: Decks/ch2-anim/

Three ways to lose marks here

1. **Copying $1 + GH$ without reading the summer.**
   Read the sign off the diagram first. A $+$ feedback summer gives $1 - GH$; in a signal flow graph the minus lives on the branch, $-H$.

2. **Deciding "non-touching" by eye.**
   List each loop's nodes and intersect. One shared node means no product term. And count loops that run through a feedforward branch.

3. **Moving a junction without compensating.**
   Summer forward $\times G$, pickoff forward $\times 1/G$. Skip it and every later line is wrong, however neat.

And check every symbolic answer

Set one gain to zero and see whether what is left is the simple diagram you expect. Before quoting $G(0)$ as a steady state, check the poles.

What the assessment will ask

Assessment	Week	Weight	What it asks from this chapter
Quiz 1	Wk 4	5% CLO1	Transfer functions from given ODEs: poles, zeros, DC gain, stability.
Quiz 2	Wk 7	5% CLO1	Block-diagram reduction and state space.
Test 1	Wk 9	30% CLO1+CLO2	With Chapters 1 and 3: definitions (MCQ); conversions and closed-loop transfer functions (analytical).

What a full-mark answer looks like

- every wire labelled, or every path and loop listed with its nodes;
- each summer's sign written down before any algebra;
- the answer as *one* fraction, then one line of check (a special value, or the other method).

What to take away

1. ODE, $G(s)$, state space and diagram are one system in four dialects. Order = highest derivative = number of poles = number of states.
2. $G(s)$ is the Laplace ratio with zero initial conditions. Stability is pole location — find the poles before trusting $G(0)$.
3. $G(s) = C(sI - A)^{-1}B + D$. The eigenvalues of A are the poles, and can reveal one that $G(s)$ hides.
4. Feedback: forward gain over $1+$ loop gain. In doubt, write the loop equations and solve them.
5. Mason: Δ from loops and non-touching sets, decided by node; Δ_k deletes every loop that touches path k.

Handbook § 2.0–2.5. Worked examples: § 2.2.5, 2.3.3, 2.4.5–2.4.6, 2.5.6–2.5.8.

Tutorial: § 2.6, Problems 1–10 and Tutorial 2.3 Q(a1)–Q(a9) — attempt it before the tutorial class.