Chapter 2: Mathematical Models of Systems

Two units – From Physical Laws to Validated Design Models

Gang Chen

South China University of Technology

From Chapter 1 signals to a physical plant

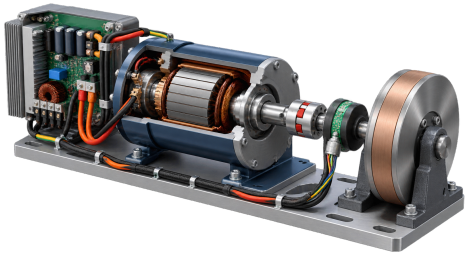
Chapter 1 fixed the course convention:

$$e = r - y_m, \qquad y_m = y + n,$$

with controller output u, plant output y, plant-input disturbance d, and sensor noise n.

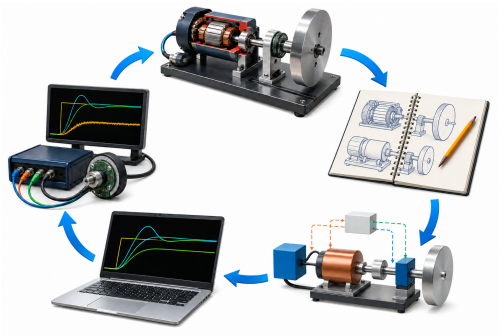
Question for this unit

If u is armature voltage and y is shaft angle, what equations must the plant obey before we design feedback?



Live signal map: $u = v$ (V) $\rightarrow$ DC motor plant $\rightarrow y = \theta$ (rad).

Why a trustworthy model matters



Physical plant → model → executable evidence

A neat equation can still be wrong

A reversed sign, mixed unit, hidden operating point, or omitted electrical dynamic can make a controller design confidently fail.

- Physics constrains signs and units.
- Assumptions constrain validity.
- Simulation tests consequences.

From physics to equations

physical boundary → governing law → ODE → linear approximation → transform → executable comparison

The model is accepted only when the chain is internally consistent and its intended use is stated.

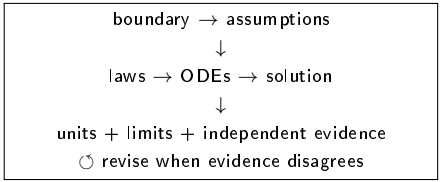
Learning outcomes and Unit A route

Learning outcomes

- Derive translational, rotational, electrical, and coupled motor balances.
- Linearize a nonlinear relation at a declared operating point and quantify error.
- Transform linear ODEs while retaining nonzero initial-condition terms.
- Interpret poles, zeros, and partial fractions as response information.
- Build and verify an ODE-level DC motor model in MATLAB and Simulink.

Physical laws $\rightarrow$ linearization $\rightarrow$ Laplace algebra $\rightarrow$ validated motor ODE

2.1 Introduction: a six-step modelling loop



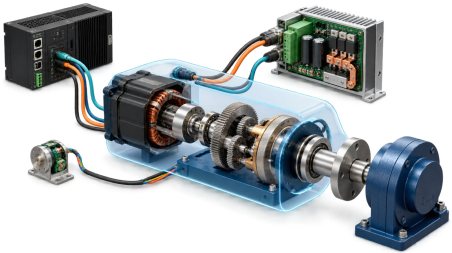
The loop is procedural and testable.

1. Define the system boundary, components, input, output, and disturbances.
2. State assumptions and the intended operating region.
3. Apply governing laws with a declared sign convention.
4. Rearrange the resulting differential equations.
5. Solve or simulate the desired variables.
6. Compare predictions with units, limiting cases, and independent evidence.

Design check

A model is provisional: disagreement sends us back to the boundary, assumptions, parameters, or equations.

A model begins with a boundary and assumptions



Translucent envelope: expanded servo-plant boundary.

Expanded servo-plant boundary

Inside: motor, shaft, gear train, and load.
Outside: controller, power amplifier, and position sensor.

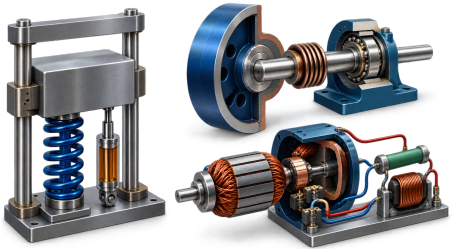
Unit A evidence model: motor only

Gear-train and load dynamics are omitted; nominal load torque is $\tau_L = 0$. The authoritative J and b are motor parameters; no claim is made that the authoritative J and b include reflected load.

- rigid motor shaft; lumped winding
- constant motor parameters near room temperature
- linear magnetic circuit; K_t and K_e constant
- viscous motor friction only; saturation and gear backlash are excluded

Failure mode: a large current can saturate the magnetic circuit, invalidating constant K_t.

2.2 Through-variables and across-variables



Different hardware, the same conservation logic.

through-variable

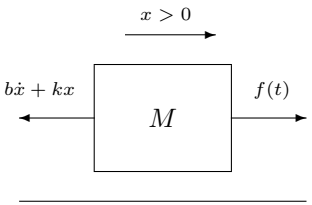
A conserved flow transmitted through an element: current i (A), force F (N), or torque τ (N.m).

across-variable

A difference measured across an element: voltage v (V), velocity difference (m/s), or angular-speed difference (rad/s).

Kirchhoff-type balances conserve through-variables at a junction; compatibility relations connect across-variables around a loop or between connected bodies.

Translational elements: force balance and constitutive laws



Live free-body diagram; right is positive.

For displacement x measured positive to the right,

$$F_M = M\ddot{x}, \qquad F_b = b\dot{x}, \qquad F_k = kx.$$

Newton's law for an applied force $f(t)$ gives

$$\boxed{M\ddot{x} + b\dot{x} + kx = f(t)}.$$

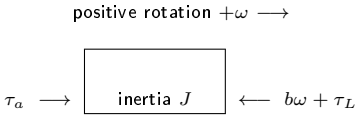
M: kg; b: N.s/m; k: N/m; every term: N.

Valid for rigid mass, linear spring, and viscous damping.

Failure mode

Coulomb friction is discontinuous at zero velocity and is not represented by $b\dot{x}$.

Rotational elements are the same pattern



One positive direction fixes every torque sign.

For shaft angle θ and angular speed $\omega = \dot{\theta}$,

$$\tau_J = J\dot{\omega}, \qquad \tau_b = b\omega, \qquad \tau_k = k_\theta \theta.$$

Summing torque in the positive rotation direction,

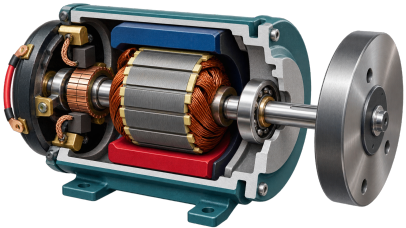
$$\boxed{J\ddot{\theta} + b\dot{\theta} + k_\theta \theta = \tau_a - \tau_L}.$$

J: kg.m^2; b: N.m.s/rad; k_θ: N.m/rad; torque terms: N.m.

Procedure

Draw one free body, label positive rotation, then assign every torque a sign before algebra.

Electrical elements: voltage balance and stored energy



Copper stores magnetic energy; resistance dissipates it.

For a series armature circuit,

$$v_R = Ri, \qquad v_L = L_a \frac{\mathrm{d}i}{\mathrm{d}t}, \qquad v_e = K_e \omega.$$

Kirchhoff's voltage law gives

$$\boxed{v = v_L + v_R + v_e}.$$

R: ohm; L_a: H = V.s/A; K_e: V.s/rad; all terms: V.

Sign check

Back-emf opposes applied voltage for motoring operation; using $-K_e\omega$ on the loss side would inject energy instead of opposing motion.

One dynamic pattern crosses physical domains

storage rate

$=$

applied flow

$-$ dissipation

one state per independent storage element

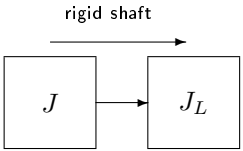
Domain	storage	dissipation	balance variable
Translational	$M\dot{v}$	bv	force (N)
Rotational	$J\dot{\omega}$	$b\omega$	torque (N.m)
Electrical	$L\dot{i}$	Ri	voltage (V)

Each first-order storage element contributes a state: velocity for mass/inertia, current for inductance. Across-domain analogy helps organize equations, but it does not make unlike units interchangeable.

Design check

Add only terms with the same physical dimension; coupling constants convert between domains.

Worked example: rigid-load acceleration at startup



$K_t i \to (J + J_L)\dot{\omega}$

Rigid inertias share one angular acceleration.

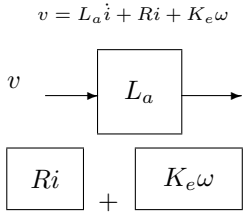
Given: $J = 0.010$ kg.m^2, rigid $J_L = 0.004$ kg.m^2, $K_t = 0.010$ N.m/A, $i(0^+) = 1.00$ A, $\omega(0^+) = 0$, $\tau_L = 0$.
Model: $(J + J_L)\dot{\omega} = K_t i - \tau_L$ at startup.
Substitution:

$$(0.010 + 0.004)\dot{\omega} = (0.010)(1.00) - 0 = 0.010 \text{ N.m},$$

$$\dot{\omega} = \frac{0.010 \text{ N.m}}{0.014 \text{ kg.m}^2} = 0.714 \text{ rad/s}^2.$$

Result: $\boxed{\dot{\omega}(0^+) = 0.714 \text{ rad/s}^2}$.
Check: positive current gives positive acceleration; adding rigid load inertia reduces acceleration from the motor-only value 1.00 rad/s^2.

Worked example: armature current derivative



Live KVL allocation; positive i follows v.

Given: $v = 1.00$ V, $i = 0.400$ A, $\omega = 5.00$ rad/s, $R = 1.00$ ohm, $K_e = 0.010$ V.s/rad, $L_a = 0.50$ H.
Model: $L_a\dot{i} + Ri + K_e\omega = v$.
Substitution:

$$Ri = (1.00)(0.400) = 0.400 \text{ V}, \qquad K_e\omega = (0.010)(5.00) = 0.050 \text{ V},$$

$$\dot{i} = \frac{1.00 - 0.400 - 0.050}{0.50} = \frac{0.550 \text{ V}}{0.50 \text{ V.s/A}} = 1.10 \text{ A/s}.$$

Result: $\boxed{\dot{i} = 1.10 \text{ A/s}}$.
Check: $Ri + K_e\omega = 0.45$ V is below the applied voltage, so current must still be increasing.

Interaction: Construct a physical balance

Predict – 4 minutes. Before constructing the balance, predict the signs and structure of the inertia, viscous-loss, motor-torque, and opposing-load terms for a motor driving a rigid load with inertia J_L and viscous loss b_L.

Hint

Reflect all quantities to the same shaft, then sum torque with one sign convention.

Interaction: Construct a physical balance

Predict – 4 minutes. Before constructing the balance, predict the signs and structure of the inertia, viscous-loss, motor-torque, and opposing-load terms for a motor driving a rigid load with inertia J_L and viscous loss b_L.

Hint

Reflect all quantities to the same shaft, then sum torque with one sign convention.

Answer

$$(J + J_L)\dot{\omega} + (b + b_L)\omega = K_t i - \tau_L.$$

The combination is invalid if a compliant gear or shaft allows different angles.

Worked example: rotor mechanical balance

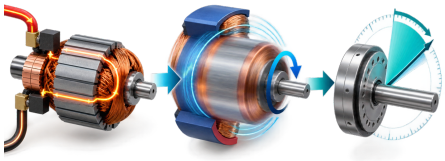
Given: $i = 0.600$ A, $\omega = 0.030$ rad/s, $K_t = 0.010$ N.m/A, $b = 0.100$ N.m.s/rad, $J = 0.010$ kg.m^2, and $\tau_L = 0$.
Model: $J\dot{\omega} = K_t i - b\omega - \tau_L$.
Substitution:

$$K_t i = (0.010)(0.600) = 0.0060 \text{ N.m}, \qquad b\omega = (0.100)(0.030) = 0.0030 \text{ N.m},$$

$$\dot{\omega} = \frac{0.0060 - 0.0030 - 0}{0.010} = 0.300 \text{ rad/s}^2.$$

Result: $\boxed{\dot{\omega} = 0.300 \text{ rad/s}^2}$.
Check: motor torque exceeds viscous loss by 0.0030 N.m, so positive acceleration is physically consistent.

Worked example: coupled first-order ODEs



i creates torque, ω creates back-emf, and θ accumulates ω.

Given: $R = 1$, $L_a = 0.50$, $J = 0.010$, $b = 0.10$, $K_t = K_e = 0.010$ in SI units; $x = [i\ \omega\ \theta]^T$, $u = v$, $y = \theta$.

Model:

$$A = \begin{bmatrix} -R/L_a & -K_e/L_a & 0 \\ K_t/J & -b/J & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1/L_a \\ 0 \\ 0 \end{bmatrix}.$$

Substitution:

$$A = \begin{bmatrix} -1/.5 & -.01/.5 & 0 \\ .01/.01 & -.1/.01 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -2 & -.02 & 0 \\ 1 & -10 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}.$$

Result: $\boxed{\dot{x} = Ax + Bv, \quad y = Cx + Dv}$, $C = [0\ 0\ 1]$, $D = 0$.

Check: A is 3×3, B is 3×1; its third row reproduces $\dot{\theta} = \omega$.

Authoritative motor parameters and units

Symbol	value	SI unit	physical meaning
J	0.01	kg.m^2	rotor inertia
b	0.10	`N.m.s/rad`	viscous coefficient
K_t	0.01	`N.m/A`	torque constant
K_e	0.01	`V.s/rad`	back-emf constant
R	1.00	ohm	armature resistance
L_a	0.50	H	armature inductance

Scripts load these values from `revised/common/servo_parameters.m`; the deck does not own a second parameter set.

Worked example: steady current and speed checkpoints

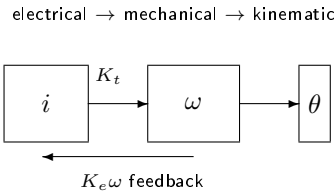
Given: $v = 1.00$ V, $R = 1.00$ ohm, $K_e = K_t = 0.010$, $b = 0.100$ N.m.s/rad, $\tau_L = 0$, and constant speed.
Model: $Ri + K_e\omega = v$, $b\omega = K_t i$ with $\dot{i} = \dot{\omega} = 0$.
Substitution:

$$i_{ss} = \frac{1.00}{1.00 + (0.010)(0.010)/0.100} = \frac{1.00}{1.001} = 0.9990 \text{ A},$$

$$\omega_{ss} = \frac{0.010}{0.100}(0.9990) = 0.09990 \text{ rad/s}.$$

Result: $\boxed{i_{ss} = 0.9990 \text{ A}, \quad \omega_{ss} = 0.09990 \text{ rad/s}}$.
Check: $Ri + K_e\omega = 0.9990 + 0.0010 = 1.0000$ V and $b\omega = K_t i = 0.00999$ N.m. Position still ramps because there is no restoring torque.

MATLAB: encode the ODE without changing the physics



Code follows the physical state order.

1. Load the shared parameter structure `p=servo_parameters()`.
2. Define $x = [i; \omega; \theta]$ and a 1 V step.
3. Implement the three right-hand sides exactly.
4. Call `ode45` on $0 \le t \le 4$ s from $x(0) = 0$.
5. Assert finite states and agreement of $i(4)$ and $\omega(4)$ with steady checkpoints.

Engineering question

Which state reacts first, and why does angle continue growing after current and speed settle?

2.3 Linearity means two properties

If input x_1 produces y_1 and x_2 produces y_2, a linear system satisfies

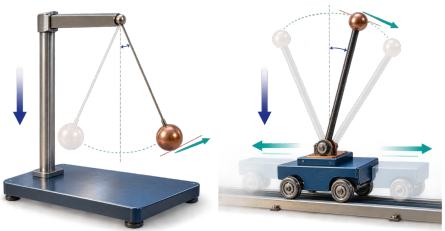
$$\mathcal{H}\{ax_1 + bx_2\} = ay_1 + by_2.$$

This combines:

- superposition: responses to summed inputs add;
- homogeneity: scaling the input scales the output by the same factor.

$y = x^2$ fails superposition; $y = mx + c$ fails homogeneity globally, but its deviation relation $\Delta y = m\Delta x$ is linear about any fixed operating point.

Operating points separate bias from dynamics



Hanging and upright equilibria have different local slopes.

Let $x = x_0 + \Delta x$ and $y = y_0 + \Delta y$, where (x_0, y_0) is a steady operating point.

Why deviation variables?

They remove constant bias terms and express how small changes propagate:

$$\Delta y \approx m\,\Delta x.$$

An operating point must satisfy the equilibrium equations under its corresponding constant input. A convenient point that is not an equilibrium produces an affine drift term, not a valid equilibrium small-signal model.

Taylor linearization is a local first-order model

$$g(x)$$
$$\downarrow \text{ expand at } x_0$$
$$g(x_0) + g'(x_0)\Delta x$$
retain first order only when the remainder is bounded

The tangent is local; the physics remains nonlinear.

For $y = g(x)$ near x_0,

$$g(x) = g(x_0) + g'(x_0)(x - x_0) + \frac{g''(x_0)}{2}(x - x_0)^2 + \cdots .$$

Neglect higher-order terms and define deviations:

$$\boxed{\Delta y \approx g'(x_0)\Delta x}.$$

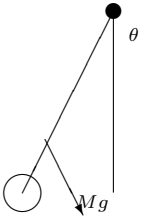
$g'(x_0)$ carries output-unit/input-unit.

Validity ends when neglected terms are no longer small.

Failure mode

If $g'(x_0) = 0$, the first nonzero higher-order term may dominate.

Pendulum: linearize restoring torque at equilibrium



equilibrium reference

Choose the equilibrium before choosing the tangent.

For a pendulum, $\tau_g = Mg\ell \sin\theta$. About $\theta_0 = 0$,

$$\tau_g - \tau_{g0} \approx \left.\frac{\partial(Mg\ell\sin\theta)}{\partial\theta}\right|_0 (\theta - 0)$$
$$= Mg\ell\theta.$$

The small-angle dynamics become

$$J\ddot{\theta} + b\dot{\theta} + Mg\ell\theta = \tau_a.$$

Angle must be in radians: only then is $\sin\theta \approx \theta$ dimensionally meaningful.

MATLAB: compute a defensible validity boundary

Define error relative to the nonlinear restoring term:

$$\varepsilon_r(\theta) = \frac{|\theta - \sin\theta|}{|\sin\theta|}, \qquad \varepsilon_r(0) = 0.$$

The script solves

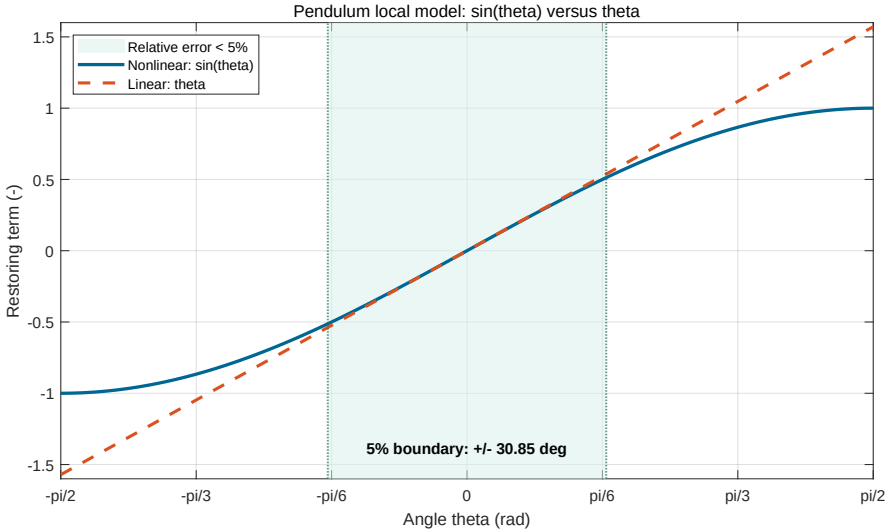
$$\frac{\theta_b - \sin\theta_b}{\sin\theta_b} = 0.05$$

with `fzero`, then asserts $\theta_b = 0.5384$ rad $= 30.85^\circ$ within tolerance.

Design check

State the denominator of a relative error. Using $|\theta|$ instead gives a different boundary.

Evidence: the pendulum approximation is local



Shaded region: $|\theta - \sin\theta| / |\sin\theta| < 5\%$; generated and asserted by `generate_ch02_figures.m`.

Interaction: Choose a valid linearization point

hanging: $\theta_0 = 0$
$\cos\theta_0 = +1$

upright: $\theta_0 = \pi$
$\cos\theta_0 = -1$

Compare local slopes before calculating.

Interpret – 3 minutes. A pendulum normally hangs near 0 rad, but a controller must also balance it upright near π rad. Interpret the equilibria and local slopes of $\sin\theta$ to select the appropriate linearization point for each task.

Hint

The slope is $\cos\theta_0$; the input required for equilibrium must also be included.

Interaction: Choose a valid linearization point

hanging: $\theta_0 = 0$
$\cos\theta_0 = +1$

upright: $\theta_0 = \pi$
$\cos\theta_0 = -1$

Compare local slopes before calculating.

Interpret – 3 minutes. A pendulum normally hangs near 0 rad, but a controller must also balance it upright near π rad. Interpret the equilibria and local slopes of $\sin\theta$ to select the appropriate linearization point for each task.

Hint

The slope is $\cos\theta_0$; the input required for equilibrium must also be included.

Answer

Use $\theta_0 = 0$ for hanging operation: slope $+1$. Use $\theta_0 = \pi$ for upright balance: slope -1. A model about 0 predicts the wrong restoring direction near π.

Validate the local model before using it

$$\varepsilon_r(\theta)$$
$$\leq 5\% \text{ inside}$$
$$|\theta| \leq 0.5384 \text{ rad}$$
test one point just outside the accepted interval

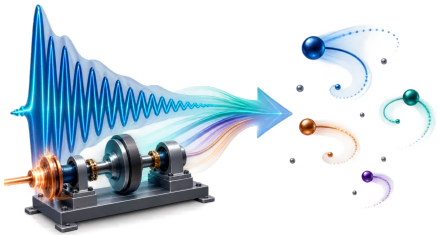
A numeric boundary makes the local claim testable.

1. Declare the operating point and deviation variables.
2. Define an error metric and acceptable bound.
3. Compare nonlinear and linear predictions over the expected range.
4. Test one point just outside the accepted range.
5. Retain the nonlinear model when excursions exceed the bound.

Intentional failure case

At $\theta = \pi/2$, the linear term is 1.571 while $\sin\theta = 1$; the relative error is 57.1%. The local model is not globally valid.

2.4 The unilateral Laplace transform



A causal transient is represented by algebraic modes in the s-domain.

For a causal time function $f(t)$,

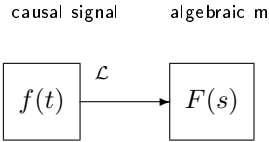
$$\boxed{F(s) = \mathcal{L}\{f(t)\} = \int_{0^-}^{\infty} f(t)e^{-st}\,\mathrm{d}t}, \qquad s = \sigma + j\omega.$$

The 0^- lower limit includes impulses or jumps at the origin. A sufficient existence condition is that f is piecewise continuous and of exponential order.

Physical meaning

The transform replaces time differentiation by algebra in complex frequency while preserving initial-condition information.

Transform pairs and useful operations



Time signal	Transform	use
$u(t)$	$1/s$	step input
$e^{-at}u(t)$	$1/(s+a)$	real mode
$\sin(\omega t)u(t)$	$\omega/(s^2+\omega^2)$	oscillation
$t^n u(t)$	$n!/s^{n+1}$	polynomial input

Start from a declared causal time signal.

Linearity gives $\mathcal{L}\{af+bg\} = aF+bG$; integration gives $\mathcal{L}\{\int_0^t f(\tau)\,\mathrm{d}\tau\} = F(s)/s$; time delay gives $\mathcal{L}\{f(t-a)u(t-a)\} = e^{-as}F(s)$. Failure mode: omitting the unit-step factor can hide the causal convention.

Derivatives carry initial-condition terms

derivative

$$\mathcal{L}\{\dot{f}\} = sF(s) - f(0^-)$$

stored initial energy is an additive response source

Integration by parts yields

$$\mathcal{L}\{\dot{f}\} = sF(s) - f(0^-),$$
$$\mathcal{L}\{\ddot{f}\} = s^2F(s) - sf(0^-) - \dot{f}(0^-).$$

For $a\dot{y} + y = r(t)$,

$$(as + 1)Y(s) = R(s) + ay(0^-).$$

Design check

Keep initial-condition terms when solving a response. Use zero initial conditions only when isolating the input-driven dynamic relation that Unit B will call a transfer function.

Interaction: Transform an ODE with a nonzero initial condition

Calculate – 4 minutes. Transform

$$\dot{y} + 2y = 3u(t), \qquad y(0^-) = 1,$$

solve for $Y(s)$, and identify the term caused only by stored initial energy.

Interaction: Transform an ODE with a nonzero initial condition

Calculate – 4 minutes. Transform

$$\dot{y} + 2y = 3u(t), \qquad y(0^-) = 1,$$

solve for $Y(s)$, and identify the term caused only by stored initial energy.

Answer

$$(s+2)Y - 1 = \frac{3}{s}, \qquad Y = \frac{3}{s(s+2)} + \frac{1}{s+2}.$$

The second term is the initial-condition contribution; setting it to zero changes the response.

Poles, zeros, and partial fractions expose response modes

$$Y(s) = \sum_k \frac{A_k}{s - p_k}$$

p_k sets rate
A_k sets weight

algebraic modes reconstruct the time response

For $Y(s) = N(s)/D(s)$, roots of D are poles; roots of N are zeros. With distinct poles p_k,

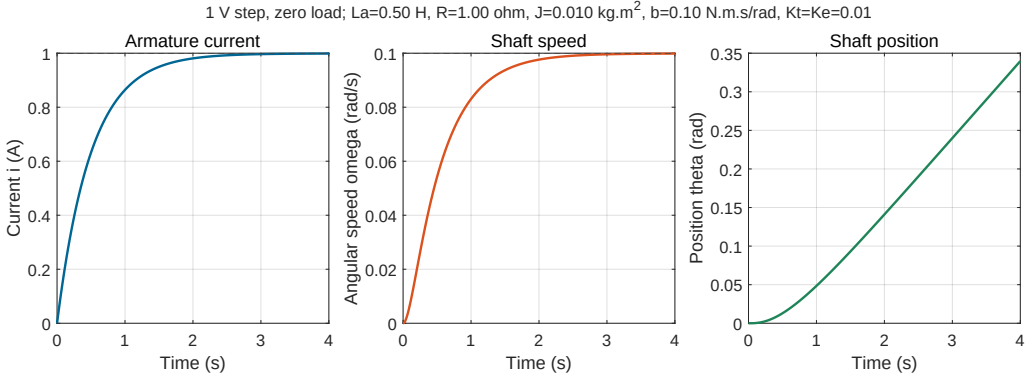
$$Y(s) = \sum_k \frac{A_k}{s - p_k}, \qquad A_k = \lim_{s \to p_k} (s - p_k) Y(s).$$

Example:

$$\frac{s+3}{(s+1)(s+2)} = \frac{2}{s+1} - \frac{1}{s+2} \Longrightarrow y(t) = 2e^{-t} - e^{-2t}.$$

Poles identify modal rates; residues set modal weights. A cancelled or tiny residue can hide a mode in one output.

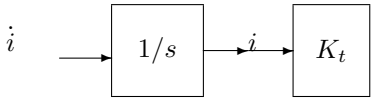
Evidence: current and speed settle; position accumulates



The 1 V, zero-load ODE simulation approaches $i_{ss} = 0.9990$ A and $\omega_{ss} = 0.09990$ rad/s; θ continues to integrate speed.

Simulink: build the exact integrator chains

each independent storage element contributes one



Three integrators realize current, speed, and angle.

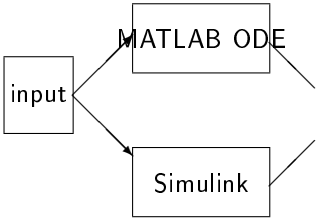
Live signal definitions

$$\dot{i} = \frac{v - Ri - K_e\omega}{L_a}, \qquad \dot{\omega} = \frac{K_t i - b\omega}{J}, \qquad \dot{\theta} = \omega.$$

The builder creates Sum, Gain, and Integrator blocks programmatically, logs i, ω, and θ, saves `ch02_dc_motor_ode.slx`, and never opens the editor.

Feedback terms Ri, $K_e\omega$, and $b\omega$ retain the negative signs established by the physical balances.

Independent numerical check: MATLAB ODE versus Simulink



Two independent numerical paths test one model.

1. Use the same shared parameters, 1 V input, zero initial state, and 4 s stop time.
2. Simulate the integrator-chain model with an ODE45 solver.
3. Independently integrate the three equations with MATLAB `ode45`.
4. Compare final i, ω, and θ component by component.
5. Require every relative error to be below 1%; otherwise fail the builder.

Expected checkpoint

Both paths finish near $[i, \omega, \theta] = [0.9987,\ 0.09986,\ 0.3397]$ in SI units.

Interaction: Diagnose a dimensional inconsistency

Diagnose – 3 minutes. A student writes

$$J\dot{\omega} + b\omega = K_t v.$$

What is dimensionally wrong, and what physical relation is missing?

Interaction: Diagnose a dimensional inconsistency

Question recap. A student writes

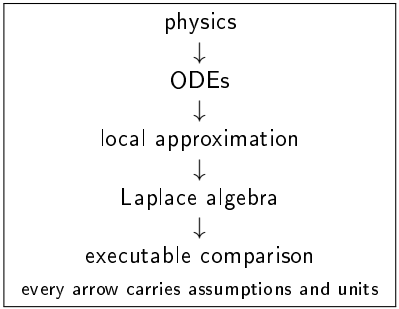
$$J\dot{\omega} + b\omega = K_t v.$$

What is dimensionally wrong, and what physical relation is missing?

Answer

K_t has units N.m/A, so $K_t v$ has units N.m.V/A, not torque. The torque law is $\tau_m = K_t i$; voltage affects current through $L_a\dot{i} + Ri + K_e\omega = v$. The electrical balance cannot be bypassed without deriving a justified reduced model.

Unit A procedure: from physics to executable evidence



The model, calculation, and measured response must agree.

1. Draw the boundary and name input, output, internal variables, and disturbances.
2. State sign conventions, assumptions, parameters, and units.
3. Apply conservation laws and element relations to derive ODEs.
4. Linearize only about a valid operating point and quantify the range.
5. Transform derivatives with initial-condition terms intact.
6. Implement the ODE and compare independent numerical realizations.

Design check

Do not accept a model because it runs; accept it because its physics, units, scope, and evidence agree.

Artifact for Unit B: a verified motor ODE

Produced by Unit A

$$L_a\dot{i} + Ri + K_e\omega = v, \qquad J\dot{\omega} + b\omega = K_t i, \qquad \dot{\theta} = \omega,$$

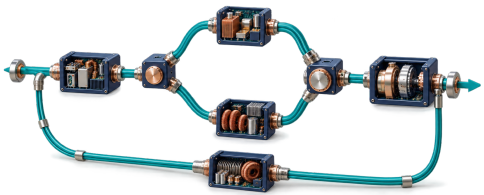
with declared zero initial conditions for the nominal input-driven experiment and all nonzero initial-condition terms retained when solving general responses.

Next: Unit B

Convert this verified ODE to transfer functions, connect block diagrams and signal-flow graphs, and prove that equivalent realizations give the same zero-initial-condition response.

Chapter 2 status: in progress; Unit A complete only after all gates pass.

Unit B: from differential equations to design models



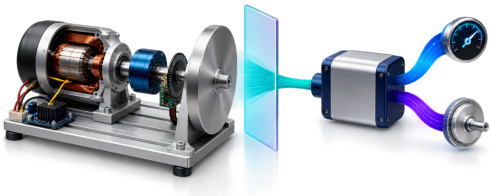
Consumed from Unit A

Verified motor ODEs, one sign convention, SI units, and unilateral Laplace transforms that retain initial-condition terms.

Unit B question

Which compact input–output and interconnection models remain equivalent, and what numerical evidence supports that claim?

One plant, four complementary representations

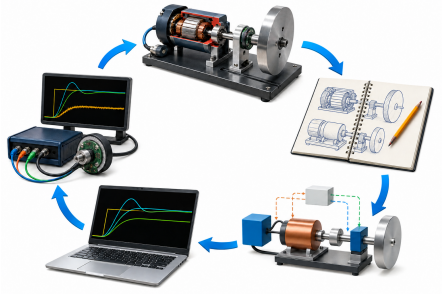


1. ODEs expose states, energy, and initial conditions.
2. A transfer function exposes one zero-state input–output map.
3. Block diagrams expose executable signal algebra.
4. Signal-flow graphs expose paths, loops, and touching relations.

Design check

Equivalent models answer the same declared question; they need not expose the same internal information.

Equivalence is a quantitative claim



Compare the physical ODE, $G_\theta(s)$, and a Simulink Transfer Fcn block under the same 1 V input, zero initial conditions, parameters, and sample times.

$$\varepsilon_{\max} = \frac{\max_t |\theta_{\mathrm{TF}}(t) - \theta_{\mathrm{ODE}}(t)|}{\max_t |\theta_{\mathrm{ODE}}(t)|} < 1\%.$$

A trace overlay is supporting evidence; signs, units, scope, and assumptions are part of the equivalence claim.

Unit B learning outcomes and route

Learning outcomes

- Derive and calculate speed and position transfer functions.
- Reduce cascade, parallel, and negative feedback models.
- Apply Mason's gain formula with explicit paths and nontouching loops.
- Use `tf`, `series`, `parallel`, `feedback`, and `minreal`.
- Validate ODE and Transfer Fcn realizations and diagnose mismatch.

Design check

Every computational concept is followed by a complete numerical worked example.

2.5 The transfer function is a zero-state ratio

$$U(s) \xrightarrow{G(s)} Y(s), \qquad \boxed{G(s) = \left.\frac{Y(s)}{U(s)}\right|_{x(0^-)=0}}.$$

- It belongs to the named input–output pair of an LTI model.
- It excludes the natural response caused by stored initial energy.
- Its units are output units divided by input units.

Design check

A transfer function is not a picture of the hardware and does not specify a unique internal realization.

Scope checks before accepting a proposed ratio

A useful $G(s)$ requires a linear (or declared local-linear) time-invariant model, named input and output, compatible units, and zero initial conditions.

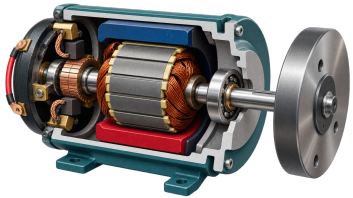
Causality

A rational physical plant is normally proper: denominator degree is at least numerator degree.

Failure mode

Dividing a response that includes initial-energy terms by $U(s)$ produces an experiment-dependent ratio, not the plant transfer function.

Derive the motor speed transfer function



With $\tau_L = 0$ and zero initial conditions,

$$V \xrightarrow{L_a s + R} I, \quad K_e\Omega \text{ opposes } V, \qquad K_t I = (Js + b)\Omega$$

$$(L_a s + R)I + K_e\Omega = V, \qquad I = \frac{Js + b}{K_t}\Omega.$$

Eliminating I gives

$$\boxed{G_\omega(s) = \frac{\Omega}{V} = \frac{K_t}{(L_a s + R)(Js + b) + K_t K_e}}.$$

G_ω: (rad/s)/V; $K_t K_e$ is the electromechanical coupling term.

Position adds one kinematic integrator

$$V(s) \xrightarrow{G_\omega(s)} \Omega(s) \xrightarrow{1/s} \Theta(s)$$

because $\dot{\theta} = \omega$ and therefore $\Theta = \Omega/s$ for zero initial angle.

$$\boxed{G_\theta(s) = \frac{\Theta}{V} = \frac{K_t}{s[(L_a s + R)(Js + b) + K_t K_e]}}.$$

Design check

The pole at the origin is physical: without a restoring torque, sustained nonzero speed accumulates angle, so open-loop position has no finite DC gain.

Worked example 1: numerical motor transfer functions

Given: $L_a = 0.50$ H, $J = 0.010$ kg.m^2, $R = 1.00$ ohm, $b = 0.100$ N.m.s/rad, $K_t = 0.010$ N.m/A, $K_e = 0.010$ V.s/rad.
Model: $G_\omega = K_t/[L_a J s^2 + (L_a b + RJ)s + (Rb + K_t K_e)]$, $G_\theta = G_\omega / s$.
Substitution:

$$L_a J = (.50)(.010) = .005, \quad L_a b + RJ = (.50)(.10) + (1)(.010) = .060,$$
$$Rb + K_t K_e = (1)(.10) + (.010)(.010) = .1001.$$

Result:

$$\boxed{G_\omega = \frac{.010}{.005s^2 + .060s + .1001} \ (\text{rad/s})/\text{V}}, \quad G_\theta = \frac{.010}{.005s^3 + .060s^2 + .1001s} \ \text{rad/V}.$$

Check: $G_\theta = G_\omega / s$ adds exactly one pole at the origin and preserves every finite coefficient.

Worked example 2: poles, time constants, and DC gain

Given: $G_\omega = .010/(.005s^2 + .060s + .1001)$.
Model: poles solve $as^2 + bs + c = 0$; $\tau_i = 1/|p_i|$; $K_{\mathrm{dc}} = G_\omega(0)$.
Substitution:

$$p_{1,2} = \frac{-.060 \pm \sqrt{(.060)^2 - 4(.005)(.1001)}}{2(.005)} = \frac{-.060 \pm .03995}{.010},$$

$$p_1 = -2.005,\ p_2 = -9.995 \text{ rad/s}, \quad \tau_1 = .499 \text{ s},\ \tau_2 = .100 \text{ s}, \quad K_{\mathrm{dc}} = .010/.1001 = .09990 \text{ (rad/s)/V}.$$

Result: $\boxed{p = \{-2.005, -9.995\} \text{ rad/s}, \quad K_{\mathrm{dc}} = .09990 \text{ (rad/s)/V}}$.
Check: both poles are negative and the slower 0.499 s mode governs settling.

Interaction: Decide whether a proposed transfer function is valid

Predict – 3 minutes. For $\dot{y} + 2y = r(t)$ with $y(0^-) = 3$, a student divides the complete response by $R(s)$ and calls the result the plant transfer function. Is the claim valid?

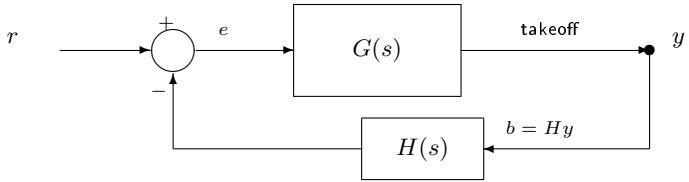
Interaction: Decide whether a proposed transfer function is valid

Predict – 3 minutes. For $\dot{y} + 2y = r(t)$ with $y(0^-) = 3$, a student divides the complete response by $R(s)$ and calls the result the plant transfer function. Is the claim valid?

Answer

No. The initial-energy term depends on the experiment. The plant transfer function is $1/(s+2)$, defined with zero initial conditions; the separate natural response is $3/(s+2)$.

2.6 A block diagram is executable signal algebra



A block maps e to $y = Ge$; the filled takeoff copies y without changing it; the signed summing junction forms $e = r - b$. Arrows declare causal direction.

Design check

Exact topology, labels, and signs remain live in LaTeX; reduction starts from signal equations rather than visual proximity.

Cascade blocks multiply when loading is negligible

$$x_1 \xrightarrow{G_1} x_2 \xrightarrow{G_2} y, \qquad x_2 = G_1 x_1, \quad y = G_2 x_2$$

$$\boxed{G_{\text{series}} = G_2 G_1}.$$

Physical caveat

Connecting real components may load the first and change its model. The product rule assumes loading is negligible or already represented.

Worked example 3: cascade and parallel reduction

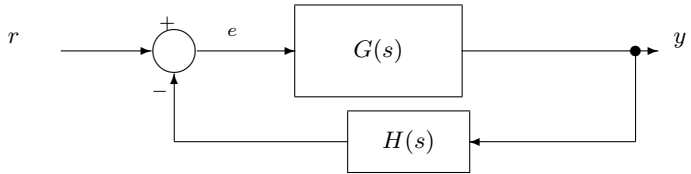
Given: $G_1 = 2/(s+2)$, $G_2 = 3/(s+4)$ in cascade; a positive parallel path $G_3 = 1/(s+4)$.
Model: $u \xrightarrow{G_1} x \xrightarrow{G_2} y_1$, $u \xrightarrow{G_3} y_2$, $y = y_1 + y_2$; hence $G = G_1G_2 + G_3$.
Substitution:

$$G_1G_2 = \frac{6}{(s+2)(s+4)}, \quad G = \frac{6}{(s+2)(s+4)} + \frac{1}{s+4},$$

$$G = \frac{6 + (s+2)}{(s+2)(s+4)} = \frac{s+8}{s^2+6s+8}.$$

Result: $\boxed{G(s) = (s+8)/(s^2+6s+8)}$.
Check: $G(0) = 1$, matching $(2/2)(3/4) + 1/4 = .75 + .25 = 1$.

Negative feedback changes the denominator



The live topology gives $e = r - Hy$ and $y = Ge$, so $(1 + GH)y = Gr$ and

$$\boxed{\frac{Y}{R} = \frac{G}{1 + GH}}.$$

Positive feedback changes the lower sign and produces denominator $1 - GH$.

Failure mode

Write the summing equation first; memorizing a denominator sign hides a change in physical interconnection.

Worked example 4: closed-loop poles and DC gain

Given: $G_c = 2$, $G_p = 1/(s+1)$, $H = 3$, negative feedback.
Model: $T = G_c G_p/(1 + G_c G_p H)$.
Substitution:

$$G_c G_p = \frac{2}{s+1}, \qquad T = \frac{2/(s+1)}{1 + 6/(s+1)} = \frac{2}{s+1+6} = \frac{2}{s+7}.$$

The pole is $s = -7$ rad/s, $\tau = 1/7 = .143$ s, and $T(0) = 2/7 = .2857$.

Result: $\boxed{T(s) = 2/(s+7), \quad p = -7 \text{ rad/s}, \quad K_{\text{dc}} = .2857}$.

Check: negative feedback moves the open-loop pole from -1 to -7 and reduces DC gain from 2 to $.2857$.

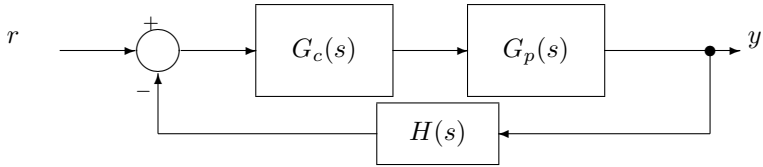
Moving a junction requires compensating algebra

Move	Compensation	Equality check
sum behind G	multiply moved input by G	$G(x+d) = Gx + Gd$
sum ahead of G	divide moved input by G	$G(x + d/G) = Gx + d$
takeoff behind G	branch carries Gx	downstream signals remain Gx
takeoff ahead of G	divide moved branch by G	branch value is unchanged

Design check

A move is valid only when every downstream signal equation is unchanged.

Interaction: Reduce a three-block feedback loop

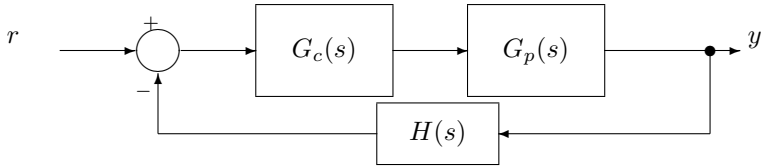


Calculate – 4 minutes. For the negative-feedback topology above,

$$G_c = \frac{s+2}{s+4}, \qquad G_p = \frac{5}{s+1}, \qquad H = 2.$$

Derive Y/R without skipping the unsimplified denominator.

Interaction: Reduce a three-block feedback loop



Calculate – 4 minutes. For the negative-feedback topology above,

$$G_c = \frac{s+2}{s+4}, \qquad G_p = \frac{5}{s+1}, \qquad H = 2.$$

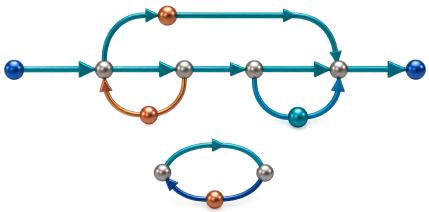
Derive Y/R without skipping the unsimplified denominator.

Answer

$$\frac{Y}{R} = \frac{5(s+2)}{(s+4)(s+1)+10(s+2)} = \frac{5s+10}{s^2+15s+24}.$$

The plus term follows from negative feedback.

2.7 Signal-flow graphs expose path structure



A node stores one signal. A directed branch multiplies its source by a transmittance. A forward path repeats no node; a loop returns to its start without repeating another node. Nontouching loops share no node.

Design check

Every node equals the sum of all incoming branch contributions.

Mason's gain formula organizes paths and loops

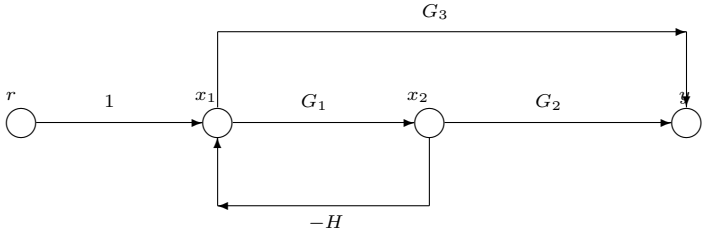
$$\boxed{T(s) = \frac{\sum_k P_k \Delta_k}{\Delta}}, \qquad \Delta = 1 - \sum_i L_i + \sum_{\substack{i<j \\ \text{nontouching}}} L_i L_j - \cdots .$$

P_k is a forward-path gain; L_i is a signed loop gain; Δ_k is formed after removing loops that touch path P_k.

Design check

List paths, loops, and all nontouching loop products before substituting numbers.

An exact graph specified by live branches



$$P_1 = G_1G_2, \quad P_2 = G_3, \quad L_1 = -G_1H.$$

The loop touches both forward paths, so $\Delta_1 = \Delta_2 = 1$ and $\Delta = 1 - L_1 = 1 + G_1H$.

Design check

The equations carry the exact topology, directions, and gains.

Worked example 5: Mason paths, loop, and result

Given: preceding graph with $G_1 = 2$, $G_2 = 3$, $G_3 = 4$, $H = .5$.
Model: $T = (P_1\Delta_1 + P_2\Delta_2)/\Delta$.
Substitution:

$$P_1 = (2)(3) = 6, \quad P_2 = 4, \quad L_1 = -(2)(.5) = -1,$$

$$\Delta = 1 - L_1 = 1 - (-1) = 2, \quad \Delta_1 = \Delta_2 = 1, \quad T = \frac{(6)(1) + (4)(1)}{2} = 5.$$

Result: $\boxed{Y/R = 5}$.
Check: $x_1 = r - .5x_2 = r - x_1$, so $x_1 = .5r$ and $y = (6 + 4)x_1 = 5r$.

Interaction: Apply Mason's rule to a small graph

Interpret – 4 minutes. Explain why $L_1 = -G_1H$, why $\Delta = 1 + G_1H$, and why neither path receives a nontrivial cofactor.

Interaction: Apply Mason's rule to a small graph

Interpret – 4 minutes. Explain why $L_1 = -G_1H$, why $\Delta = 1 + G_1H$, and why neither path receives a nontrivial cofactor.

Answer

The return branch is negative. Mason subtracts the signed loop gain, giving $1 - (-G_1H)$. The loop shares x_1 with both forward paths, so $\Delta_1 = \Delta_2 = 1$.

2.8 A reusable design-model workflow

declare boundary and variables

$\Downarrow$

derive balances and linearize if required

$\Downarrow$

form transfer functions and interconnections

$\Downarrow$

check units, limiting cases, and executable agreement

Each computational concept receives a complete worked example; each example ends with a physical or numerical sanity check.

Different plants reuse the same modeling grammar

Example	storage	loss/flow	natural output
photovoltaic actuator	charge/field approximation	electrical resistance	voltage/current
liquid-level tank	liquid volume	outlet resistance	height
traction drive	mass and rotation	drag/friction	speed/position
electrical filter	capacitor/inductor energy	resistance	voltage

Design check

Block algebra is reusable only after each plant's boundary, units, and constitutive laws are declared.

A design model remains scoped, even when validated

Retain

Dominant storage, dissipative terms, input/output units, signs, and modes that influence design bandwidth.

Declare as approximations

Rigid couplings, negligible loading, neglected fast poles, local linearization, saturation exclusion, and zero initial conditions.

Validation inside one experiment does not authorize use outside the declared scope.

2.9 MATLAB operations mirror the block algebra

Operation	purpose
`tf`, `ss`	construct declared dynamic models
`series`, `parallel`	combine cascade and parallel paths
`feedback`	close the declared return loop
`minreal`	cancel algebraic pole–zero pairs with explicit tolerance
`pole`, `zero`, `step`, `lsim`	inspect dynamics and responses

Evidence rule

Software output verifies an encoded model; it does not repair incorrect signs, topology, units, or assumptions.

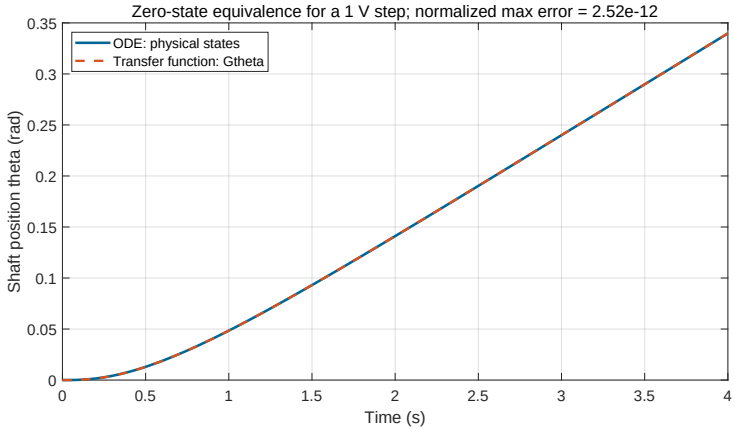
MATLAB evidence: construct both motor transfer functions

Separate code evidence

```
Gomega = tf(p.Kt,[p.La*p.J, p.La*p.b+p.R*p.J, p.R*p.b+p.Kt*p.Ke]);
Gtheta = Gomega/tf([1 0],1);
thetaTf = lsim(Gtheta,ones(size(tGrid)),tGrid);
```

Coefficient order is descending powers of s. The position denominator gains a trailing zero because division by s adds the physical integrator.

ODE and transfer-function responses coincide



Vector evidence uses one time grid, one 1 V input, zero initial conditions, and the shared motor parameter source.

Worked example 6: normalized equivalence error

Given: the verified run gives $\max|\theta_{\mathrm{TF}}-\theta_{\mathrm{ODE}}| = 8.56\times10^{-13}$ rad and $\max|\theta_{\mathrm{ODE}}| = 0.339741$ rad.
Model: $\varepsilon_{\max} = \max|\theta_{\mathrm{TF}}-\theta_{\mathrm{ODE}}| / \max|\theta_{\mathrm{ODE}}|$.
Substitution:

$$\varepsilon_{\max} = \frac{8.56\times10^{-13}\ \mathrm{rad}}{0.339741\ \mathrm{rad}} = 2.52\times10^{-12}, \qquad 100\varepsilon_{\max} = 2.52\times10^{-10}\% < 1.00\%.$$

Result: $\boxed{\varepsilon_{\max} = 2.52\times10^{-10}\% < 1\%}$.
Check: radians cancel and the metric is dimensionless. The 1% limit is $1/(2.52\times10^{-10}) = 3.97\times10^{9}$ times larger, so the error is about 9.60 orders of magnitude below the acceptance limit.

Headless Simulink Transfer Fcn realization

$$1\ \mathrm{V} \xrightarrow{G_\theta(s)} \theta_{\mathrm{TF}}(t) \xrightarrow{\text{logged samples}} \varepsilon_{\max}$$

The generated model stores numerator K_t and denominator $[L_aJ,\ L_ab + RJ,\ Rb + K_tK_e,\ 0]$, saves `ch02_motor_transfer_function.slx`, and never opens an editor. The same build script simulates the ODE model, interpolates both logged outputs onto a common time base, and asserts the 1% threshold.

Simulink evidence remains separate from diagnosis

Builder evidence

```
add_block('simulink/Continuous/Transfer Fcn',...)
set_param(block,'Numerator','p.Kt','Denominator',denTheta)
assert(normalizedMaxError < 0.01,...)
```

Design check

The code frame documents realization and assertion. The following interaction contains only the engineering question, hint, and answer.

Interaction: Diagnose an ODE–transfer-function mismatch

Diagnose – 5 minutes. The ODE begins at $\theta(0) = 0.20$ rad while the transfer-function simulation begins from rest. Both receive the same 1 V input, and their traces differ by nearly a constant offset. Is the transfer-function derivation necessarily wrong?

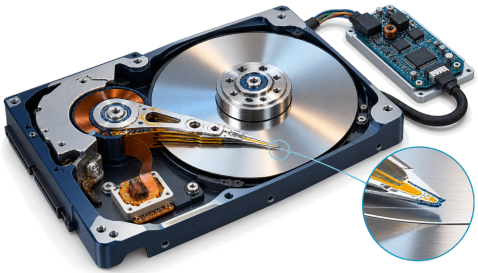
Interaction: Diagnose an ODE–transfer-function mismatch

Diagnose – 5 minutes. The ODE begins at $\theta(0) = 0.20$ rad while the transfer-function simulation begins from rest. Both receive the same 1 V input, and their traces differ by nearly a constant offset. Is the transfer-function derivation necessarily wrong?

Answer

No. A transfer function describes the zero-state input–output response. Match initial conditions or add the ODE natural response separately, then compare again. Only a remaining mismatch implicates coefficients, signs, input, or sampling.

2.10 Disk drive: model only what the design needs



The actuator rotates a read head over a disk track. The rigid-flexure approximation, linear amplifier/motor behavior, negligible back-emf, and no saturation define the first design model. The electrical pole is much faster than the mechanical pole.

Scope

Flexure resonances and nonlinear limits must return before final hardware qualification; they are intentionally absent from this first model.

Worked example 7: disk-drive approximation and feedback

Given: $J = 1$ N.m.s^2/rad, $b = 20$ N.m.s/rad, $L = 1$ mH, $R = 1$ ohm, $K_m = 5$ N.m/A, amplifier $K_a = 40$.
Model: $G = K_m/[s(Js + b)(Ls + R)]$; neglect the fast electrical pole when $\tau_e \ll \tau_m$; close unity feedback with K_a.
Substitution:

$$\tau_e = L/R = .001 \text{ s}, \qquad \tau_m = J/b = .050 \text{ s}, \qquad \tau_e/\tau_m = .02,$$

$$G(s) = \frac{5000}{s(s+20)(s+1000)} \approx \frac{5}{s(s+20)},$$

$$T = \frac{40[5/(s(s+20))]}{1 + 40[5/(s(s+20))]} = \frac{200}{s^2 + 20s + 200}.$$

Result: $\boxed{T(s) = 200/(s^2 + 20s + 200)}$.
Check: τ_e is fifty times faster than τ_m and $T(0) = 1$, consistent with unity-feedback tracking of a 0.1 rad command.

Representation checklist: choose, connect, verify

Representation	best question	required check
ODE/state model	energy, states, initial conditions	units and balances
transfer function	zero-state input–output dynamics	poles, zeros, properness
block diagram	interconnection algebra	signs and loading
signal-flow graph	path/loop structure	touching relations and cofactors
executable model	numerical response	common inputs, parameters, time base

Design check

Use the simplest representation that preserves the question being asked.

Chapter 2 closes by checking model agreement

physical laws $\Rightarrow$ linear ODEs $\Rightarrow$ Laplace model $\Rightarrow$ transfer functions $\Rightarrow$ interconnections $\Rightarrow$ validated simulation.

What you can now defend

Each coefficient traces to a parameter and unit; each reduction traces to live signal equations; each numerical result shows substitution and a check; and ODE, transfer-function, and headless Simulink realizations satisfy the same normalized-error contract.

Chapter 2 status: complete only when all automated and visual gates pass.