Chapter Outline

Exercises, Case Analysis, and Course Projects

Chapter 4
Vehicle Motion Modeling and Simulation

4.1 Modeling Fundamentals and Kinematics
4.2 Rigid-Body Dynamics and Solvers
4.3 Propulsion, Forces, and Observations
4.4 Trim, Linearization, and Dynamics
4.5 Multicopter Modeling and Simulation
4.6 Measurement and Static Identification
4.7 Dynamic Identification and Refinement
4.8 Typical Vehicles and Extensions
4.9 Modeling Tools and Deployment
4.10 API and SDK Modeling Interfaces
4.11 Examples and Integrated Practice
4.12 Exercises and Course Projects

4 Units in This Section

4.12.1 Concept and Principle Review

Textbook: Sec. 4.4, Exercises 1, 2, 5

4.12.2 Calculations and Approximation Error

Textbook: Sec. 4.4, Exercises 3, 4, 6, 8

4.12.3 Design Tasks and Course Projects

Textbook: Sec. 4.4, Exercises 7, 9, 10

4.12.4 Result Discussion and Chapter Summary

Textbook: Sec. 4.1.2.2, Fig. 4.6; chapter review

4.12.1 Concept and Principle Review: Evidence for Exercises and Staged Experiments

Predict Changed Conditions Before Choosing Validation

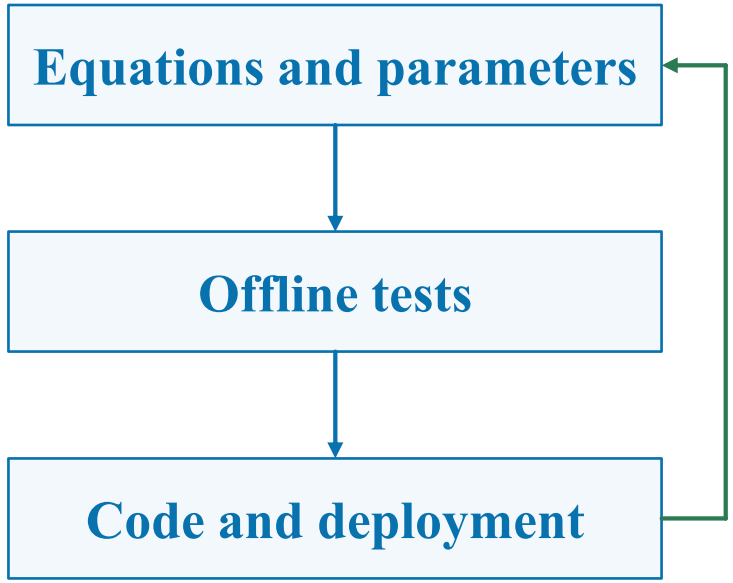
- **Connection:** Slide deck 4.11 provided reproducible procedures. Here, change the conditions, predict the results, then independently choose models, interfaces, and validation methods.
- **Exercise map:** Of the ten exercises in textbook Sec. 4.4, 4.12.1 covers 1, 2, 5; 4.12.2 covers 3, 4, 6, 8; 4.12.3 covers 7, 9, 10. Calculations retain conditions and units; designs state inputs/outputs, implementation steps, and supported conclusions.
- **Staged experiment:** A4 is Chapter 4's fourth staged experiment. In 4.12.3, make one model change with a clear goal. For the A1–A4 sequence, see "Staged Experiment A1: Actuators and Synthesis" in slide deck 4.3.

Modeling is complete only when inputs, conditions, results, and evidence can be explained.

4.12.1 Concept and Principle Review: Q1—MBD Workflow

Textbook Q1: How Does MBD Connect Development Stages?

- **Textbook Q1:** Describe the model-based design (MBD) process and advantages of the RflySim Toolchain (slide deck 4.1; textbook Sec. 4.1.1).
- **Six stages:** Modeling and identification → rapid control prototyping → digital simulation validation → automatic code generation → hardware-in-the-loop (HITL; textbook HIL) → system integration and testing (textbook Fig. 4.1).
- **Advantages/conditions:** One model's generated code reduces hand-coding differences; parameters and tests are reusable and traceable. Reuse can propagate errors: verify deployment with matched initial state, input, and simulation time.



Every development stage should leave a model or result that can be checked.

4.12.1 Concept and Principle Review: Q2—Model Hierarchy

Textbook Q2: How Are the Three Core Subsystems Connected?

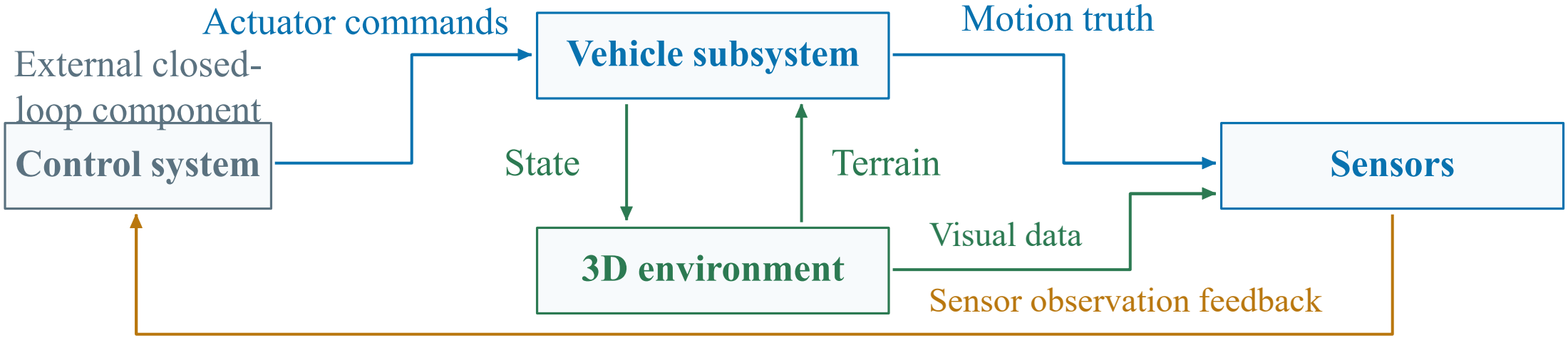


Figure: Motion truth, environment, and observations close the loop

- **Textbook Q2:** Explain the roles and connections of the three core subsystems and four vehicle subnodes in the unified framework (slide deck 4.1; textbook Sec. 4.1.3).
- **Core:** vehicle, sensors, and 3D environment. The controller is external (slides 4.1, 4.3).

Identify the hierarchy before connecting signals to modules.

4.12.1 Concept and Principle Review: Q2—Vehicle Subsystem Outputs

Q2 Diagram: What Do the Four Vehicle Subsystems Output?

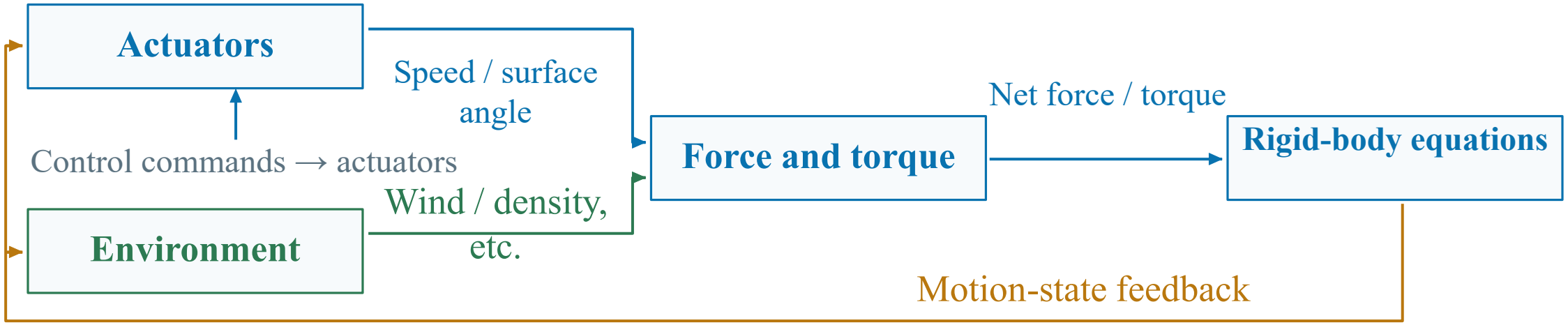


Figure: Combined component effects advance states, which feed back to components

- Four subnodes, outputs labeled: actuators, environment, force/torque synthesis, rigid body.
- Label signal dimensions, frames, units, and update times. Acc may mean specific force (slides 4.3).

Sum forces and torques from all sources, then advance states with the rigid-body equations.

4.12.1 Concept and Principle Review: Q5—White, Gray, and Black Boxes

Textbook Q5: Choose a White-, Gray-, or Black-Box Model

- Compare white-, gray-, and black-box models and uses (slides 4.1; textbook Sec. 4.2.2.1).

Dimension	White box	Gray box	Black box
Known physics	Complete equations and structure	First-order lag; thrust $\propto$ speed^2	No assumed internal structure
Parameters	Separate tests or manuals	Data-fit gain/time constant	Fit input/output data
Interpretability	All parameters have physical meaning	Parameters describe components	Coefficients may not map to components
Data needs	Mainly validation data	Excite the target regime	Extensive data cover use
Applications	Explain physics; design new components	Known structure; parameters to calibrate	Unknown structure or complex physics

- All three routes must state operating conditions and use independent validation data.

Model type assigns roles to physics and data; it is not an inherent accuracy ranking.

4.12.1 Concept and Principle Review: Complexity and Operating Conditions

Model Selection: Is a More Complex Model Always Better?

- For small motor steps near hover, a first-order gray-box model may explain the main dynamics. Higher-order terms add parameters to estimate.
- Large voltage changes or actuator saturation may invalidate a low-order model near the original operating point. Expand data coverage or add the relevant physics.
- On the same independent validation set, compare errors, residual patterns, and computational cost. Choose a model that meets the task with credible parameters, beyond fit percentage alone.

Model adequacy depends on target conditions and independent validation.

Chapter Outline

Exercises, Case Analysis, and Course Projects

Chapter 4
Vehicle Motion Modeling and Simulation

4.1 Modeling Fundamentals and Kinematics
4.2 Rigid-Body Dynamics and Solvers
4.3 Propulsion, Forces, and Observations
4.4 Trim, Linearization, and Dynamics
4.5 Multicopter Modeling and Simulation
4.6 Measurement and Static Identification
4.7 Dynamic Identification and Refinement
4.8 Typical Vehicles and Extensions
4.9 Modeling Tools and Deployment
4.10 API and SDK Modeling Interfaces
4.11 Examples and Integrated Practice
4.12 Exercises and Course Projects

4 Units in This Section

4.12.1 Concept and Principle Review

Textbook: Sec. 4.4, Exercises 1, 2, 5

4.12.2 Calculations and Approximation Error

Textbook: Sec. 4.4, Exercises 3, 4, 6, 8

4.12.3 Design Tasks and Course Projects

Textbook: Sec. 4.4, Exercises 7, 9, 10

4.12.4 Result Discussion and Chapter Summary

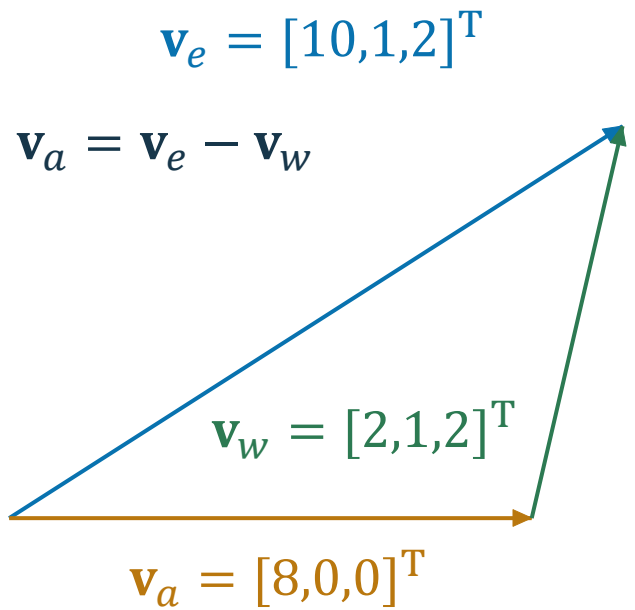
Textbook: Sec. 4.1.2.2, Fig. 4.6; chapter review

4.12.2 Calculations and Approximation Error: Q3—Ground, Wind, and Air Velocity

Textbook Q3: Express Wind and Ground Velocity in One Frame

- **Q3:** Relate air/ground/wind speeds; define angle of attack and sideslip (slides 4.3; textbook Sec. 4.1.4.4).
- **Body aligned north–east–down (NED), no wind:** Ground velocity $[10,1,2]^T$ m/s gives airspeed ≈10.25 m/s.
- Angle of attack is 11.31° and sideslip 5.60°. With wind $[2,1,2]^T$, air-relative velocity becomes $[8,0,0]^T$ m/s and both angles are zero.
- **Definitions:** u_a, v_a, w_a: body-frame air-relative components; V_a: airspeed magnitude (slides 4.3). Angles are undefined at zero airspeed.

$$\alpha = \text{atan2}(w_a, u_a), \quad \beta = \arcsin(v_a/V_a)$$



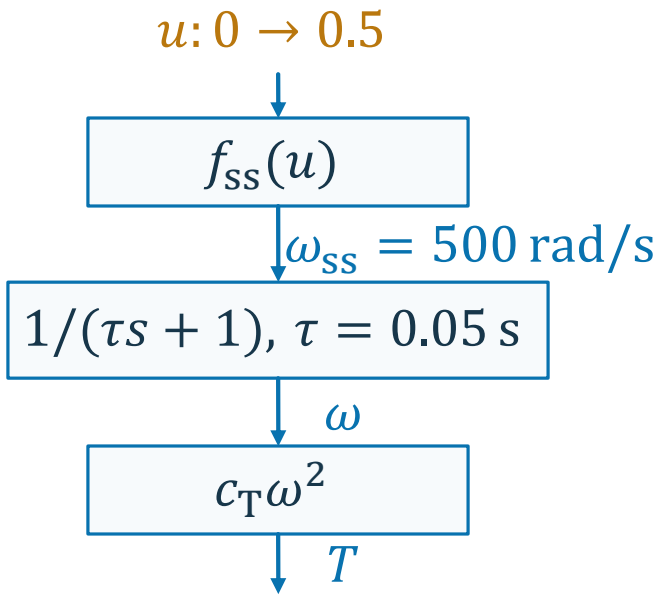
Air-relative velocity is a same-frame vector difference; airflow angles follow from that vector.

4.12.2 Calculations and Approximation Error: Q4—Steady Mapping and Dynamics

Textbook Q4: Cascade a Steady Mapping and a Dynamic Block

- **Textbook Q4:** Explain the idealized actuator cascade "steady-state function + dynamic transfer function" (slide deck 4.3; textbook Sec. 4.1.4.3).
- **Example:** Normalized input 0→0.5, steady speed 500 rad/s, time constant 0.05 s, initial speed zero.
- At 0.05 s, speed is 316.1 rad/s. With thrust coefficient $c_T = 2 \times 10^{-5}$, thrust is only 2.00 N, versus a final 5 N.
- The target may jump while speed remains continuous. After shutdown, speed and thrust decay; they do not vanish instantly unless braking or resetting is added.

$$\omega = 500\left(1 - e^{-t/0.05}\right), \quad T = c_T \omega^2$$



General form (textbook Eq. (4.12)):

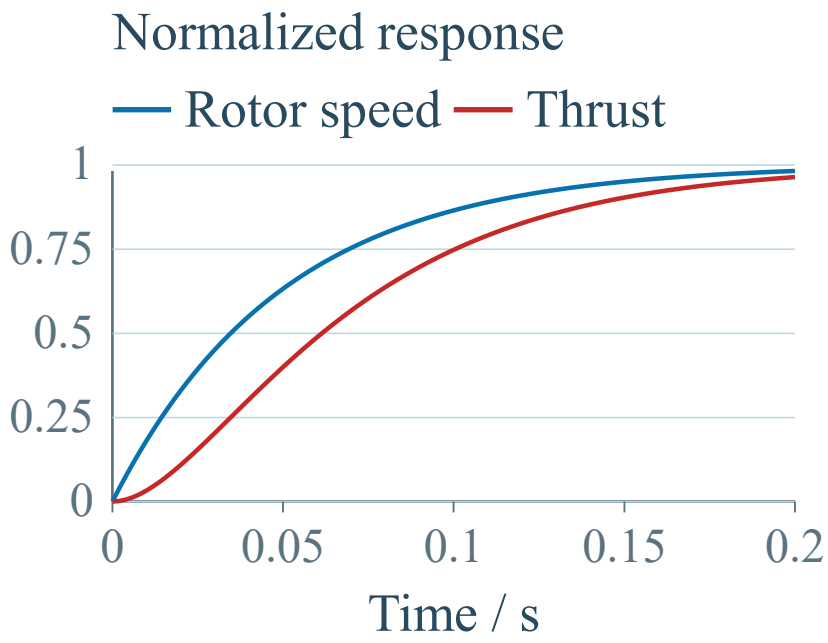
$$\delta_i(s) = G_{ss,i}(s)\, f_{ss,i}(\sigma_i(s))$$

Target speed may jump; actual speed and thrust still take time to develop.

4.12.2 Calculations and Approximation Error: Q4—Speed Squared and Thrust

Check the Square Law with Two Normalized Curves

- Blue is speed divided by 500; red is thrust divided by 5. Both use the Q4 teaching step from zero initial conditions.
- Conclusion: At 0.05 s, speed reaches 63.2% of its final value, while thrust reaches only 40.0% (0.632^2). Use this to check component logs.



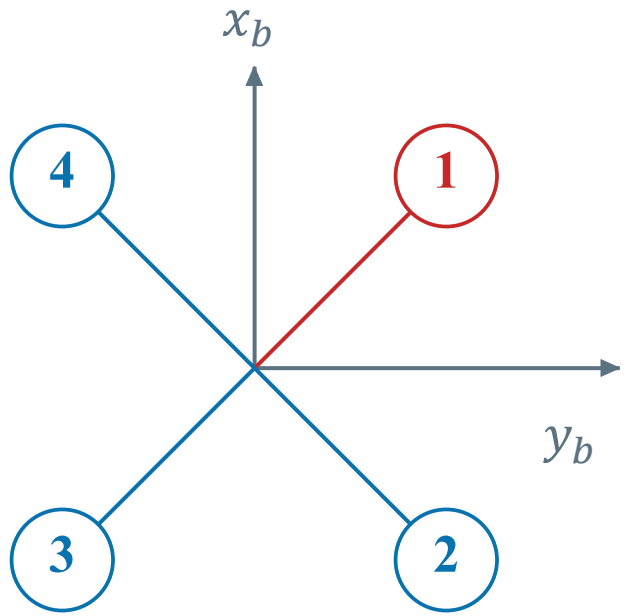
First-order speed; speed-squared thrust

Thrust scales with speed squared; its transient fraction differs from the speed fraction.

4.12.2 Calculations and Approximation Error: Q6—X-Layout Rotor Numbering

Textbook Q6: Specify X-Layout Numbering and Signs First

- **Q6:** Explain X-quadcopter allocation/inverse allocation (slides 4.3; textbook Sec. 4.1.4.4). Forward-right-down (FRD) example: rotors 1–4 clockwise from front right; arm $d = 0.25$ m; yaw signs $[+, -, +, -]$; $a = d/\sqrt{2}$.
  - Rotor 1: Position $[a, a, 0]^T$, upward thrust $[0,0, -T_1]^T$. Their cross product gives negative roll and positive pitch torque.
- With $\eta = c_{\mathrm{M}}/c_{\mathrm{T}} = 0.02$ m, adding 1 N to rotor 1 gives increment $[1, -0.1768, 0.1768, 0.02]^{\mathrm{T}}$: first entry in N, the others in N·m.



FRD top view: Thrust out of page

Every synthesis-matrix column must correspond to a clearly identified rotor.

4.12.2 Calculations and Approximation Error: Q6—Synthesis and Inverse Allocation

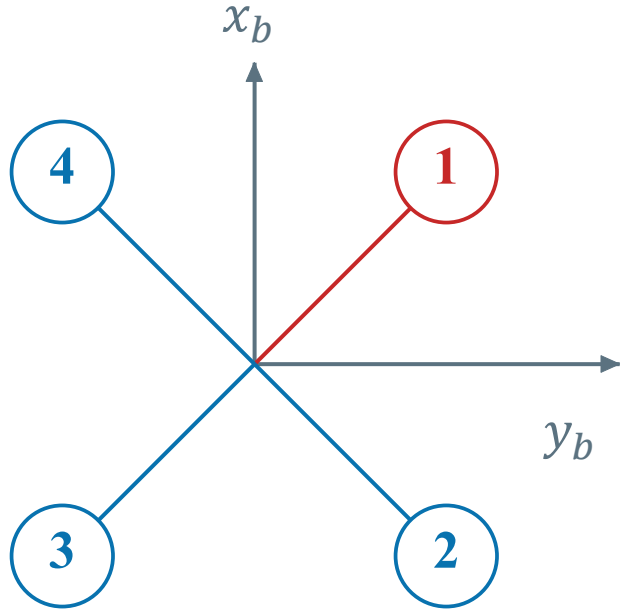
Q6 Derivation: Build the Synthesis Matrix, Then Invert It

- Textbook Eq. (4.35) uses ϖ_i^2. Substitute $T_i = c_{\mathrm{T}}\varpi_i^2$ and $c_{\mathrm{M}}\varpi_i^2 = \eta T_i$, with $a = d/\sqrt{2}$:

$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \mathbf{M}_4 \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}, \quad \mathbf{M}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -a & -a & a & a \\ a & -a & -a & a \\ \eta & -\eta & \eta & -\eta \end{bmatrix}$$

- **Inverse:** Four orthogonal rows give the inverse directly.

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \mathbf{M}_4^{-1} \begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}, \quad \mathbf{M}_4^{-1} = \frac{1}{4}\begin{bmatrix} 1 & -1/a & 1/a & 1/\eta \\ 1 & -1/a & -1/a & -1/\eta \\ 1 & 1/a & -1/a & 1/\eta \\ 1 & 1/a & 1/a & -1/\eta \end{bmatrix}$$



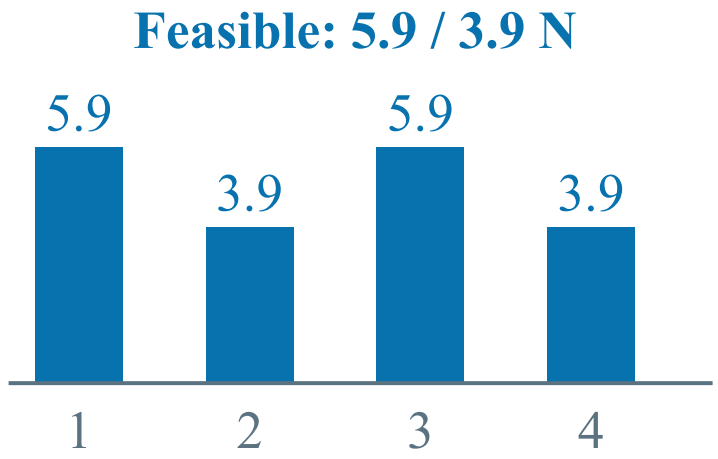
FRD top view: Thrust out of page

4.12.2 Calculations and Approximation Error: Q6—Inverse Allocation and Thrust Limits

Q6 Discussion: A Mathematical Inverse Must Also Be Feasible

- Keep 19.6 N total (4.9 N each): Add ΔT to rotors 1 and 3; subtract it from 2 and 4. Roll/pitch torque changes cancel.
- Yaw target 0.08 N·m requires $\Delta T = 1$ N, giving 5.9, 3.9, 5.9, 3.9 N. Total thrust stays unchanged.
- At a 0.48 N·m target, the required increment is 6 N, making the other pair −1.1 N. Conventional unidirectional rotors cannot deliver this; change the target or allocation strategy.

$$\Delta T = \frac{\tau_z}{4\eta}$$

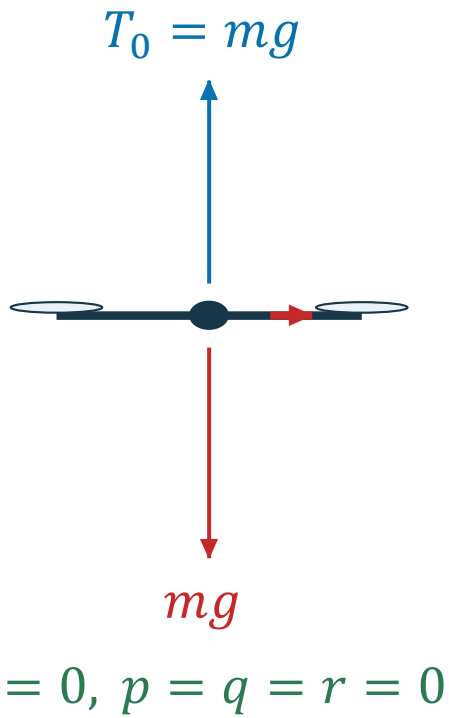


Negative or excessive thrust means the target exceeds the actuators' feasible range.

4.12.2 Calculations and Approximation Error: Q8—Hover Trim

Textbook Q8: Hover Linearization Starts at Trim

- **Textbook Q8:** Describe hover linearization and discretization (slide deck 4.4; textbook Sec. 4.1.4.5).
- State conditions first: Level, stationary, no wind or contact. Solve trim satisfying $f(x_0, u_0) = 0$; motors must also be at matching steady states.
- Define perturbations $\delta x = x - x_0$ and $\delta u = u - u_0$; compute local coefficients, choose a sampling period, and compare small-perturbation responses with the nonlinear model.



Find an operating point with no state change before examining nearby perturbations.

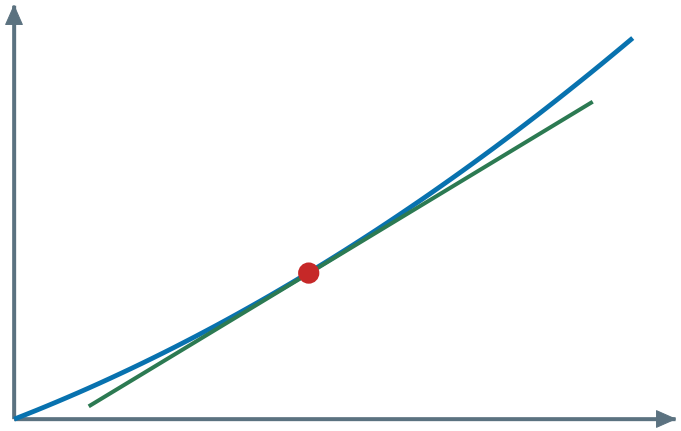
4.12.2 Calculations and Approximation Error: Q8—Perturbation Model

Q8 Discussion: A Linear Model Describes Perturbations

- At trim, differentiate to get A, B: sensitivities of state derivatives to small changes in states or inputs.
- Use a local 3D attitude error. Quaternion perturbations must respect unit norm; four components cannot be varied arbitrarily and independently.
- Compare nonlinear and linear responses to small steps, then increase amplitude gradually. Their differences define the approximation's useful range.

$$\delta\dot{x} = A\delta x + B\delta u, \quad A = \left.\frac{\partial f}{\partial x}\right|_0, \quad B = \left.\frac{\partial f}{\partial u}\right|_0$$

Nonlinear curve and local tangent



Take δx near trim

Matrix coefficients are sensitivities near the current operating point, not laws for all conditions.

4.12.2 Calculations and Approximation Error: Hover Height and Velocity States

Hover Example: Retain Two Upward-Motion States

- **To expose discretization, use an independent ideal 2 kg model:** Upward height h, upward velocity v_h, level attitude, total-thrust perturbation δT.
- Near $T_0 = mg$, neglecting drag and motor dynamics, height and velocity perturbations satisfy:
- For M0's command-to-height response (minimal template e0_MinModelTemp, used since slide deck 4.3), also cascade motor dynamics and input mapping. Here the input is already actual total-thrust perturbation.

$$\delta\dot{h} = \delta v_h, \qquad \delta\dot{v}_h = \frac{\delta T}{m}$$

Net thrust changes velocity first, then height through integration.

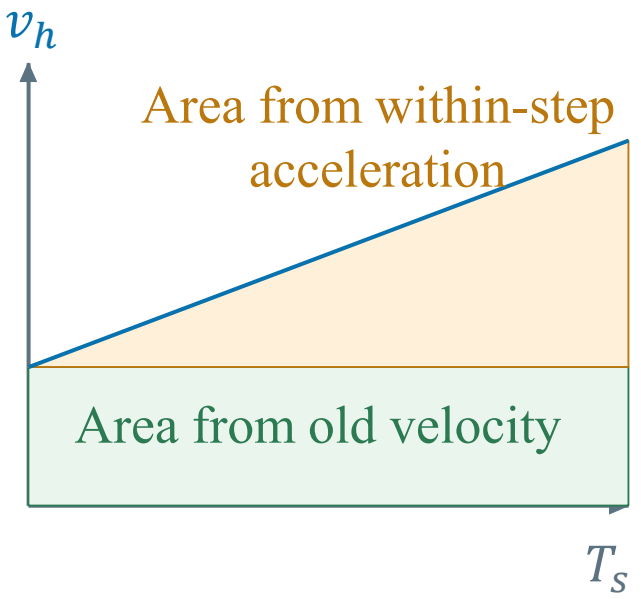
4.12.2 Calculations and Approximation Error: Zero-Order-Hold Calculation

Hold an Input for One Sample: What Is the Next State?

- Zero initial perturbations: Hold $\delta T = 2$ N for $T_s = 0.1$ s in the 2 kg model. First-step velocity and height increments are 0.1 m/s and 0.005 m.
- **After 10 exact-hold steps:** Velocity increment 1 m/s, height increment 0.5 m. Euler accumulates only old velocity, giving 0.45 m; half the step gives 0.475 m.

$$\delta v_{k+1} = \delta v_k + T_s \delta T_k / m$$

$$\delta h_{k+1} = \delta h_k + T_s \delta v_k + T_s^2 \delta T_k / (2m)$$



See "Exact Discretization of a First-Order Model" (slides 4.4).

A constant input under zero-order hold still changes state throughout the interval.

4.12.2 Calculations and Approximation Error: Small-Angle Approximation Error

Use Small-Angle Approximations Within an Error Limit

- At total thrust mg, north acceleration is $a_N = -g\sin\theta$. Near hover, approximate it by $-g\theta$, with θ in rad.
- For $g = 9.8$ m/s²: At 10°, exact/linear values are about −1.702/−1.710 m/s²; at 30°, −4.900/−5.131 m/s². Relative errors are about 0.5%/4.7%. For a 1% limit, keep tilt within about 14°.
- For derivation and comparison, see "How Small-Angle Error Grows with Angle" in slide deck 4.4.

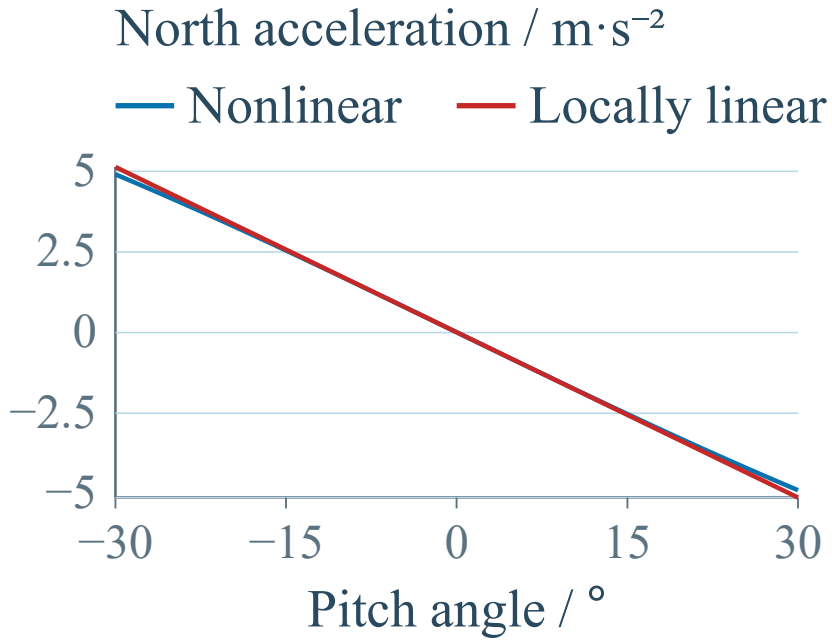


Figure: Exact projection and small-angle approximation

As tilt grows, the local line increasingly departs from the original geometry.

Chapter Outline

Exercises, Case Analysis, and Course Projects

Chapter 4
Vehicle Motion Modeling and Simulation

4.1 Modeling Fundamentals and Kinematics
4.2 Rigid-Body Dynamics and Solvers
4.3 Propulsion, Forces, and Observations
4.4 Trim, Linearization, and Dynamics
4.5 Multicopter Modeling and Simulation
4.6 Measurement and Static Identification
4.7 Dynamic Identification and Refinement
4.8 Typical Vehicles and Extensions
4.9 Modeling Tools and Deployment
4.10 API and SDK Modeling Interfaces
4.11 Examples and Integrated Practice
4.12 Exercises and Course Projects

4 Units in This Section

4.12.1 Concept and Principle Review

Textbook: Sec. 4.4, Exercises 1, 2, 5

4.12.2 Calculations and Approximation Error

Textbook: Sec. 4.4, Exercises 3, 4, 6, 8

4.12.3 Design Tasks and Course Projects

Textbook: Sec. 4.4, Exercises 7, 9, 10

4.12.4 Result Discussion and Chapter Summary

Textbook: Sec. 4.1.2.2, Fig. 4.6; chapter review

4.12.3 Design Tasks and Course Projects: Q7—Wind Generation and Connections

Textbook Q7: Define the Wind Model and Its Purpose First

- **Textbook Q7:** Design wind injection using MIL-F-8785C or a similar specification (slide deck 4.3; textbook Sec. 4.1.4.2).
  - $\mathbf{v}_{\text{wind}} = \mathbf{v}_{\text{turb}} + \mathbf{v}_{\text{cons}} + \mathbf{v}_{\text{shear}} + \mathbf{v}_{\text{gust}} + \cdots$
- Discrete Dryden turbulence (shaping filters; slides 4.3) takes height, speed, and ground-to-body R^T. Its output is already body-frame wind; convert to NED before M0 WindDist.
- The four terms are turbulence, constant wind, wind shear, and gusts, all defined in the ground frame. Frozen turbulence assumes mean wind and turbulence intensity are small relative to ground speed. Log intensity, scale, sample period, and seed.

Dryden: Body-frame wind

↓

$$v_{w,e} = R v_{w,b}$$

↓

M0 WindDist: NED

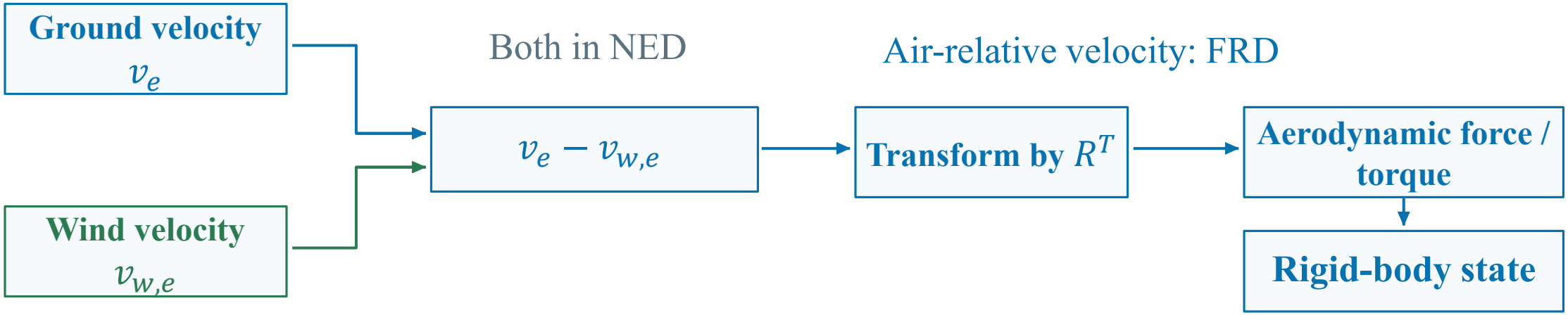
↓

$$v_{a,b} = R^T (v_e - v_{w,e})$$

A wind-field design needs generation rules, frame connections, and load validation.

4.12.3 Design Tasks and Course Projects: Q7—Relative Airflow and Loads

Q7 Diagram: Wind Must Affect Relative Airflow and Loads



Fix ground velocity and attitude to check direction first

Figure: Wind enters the load chain

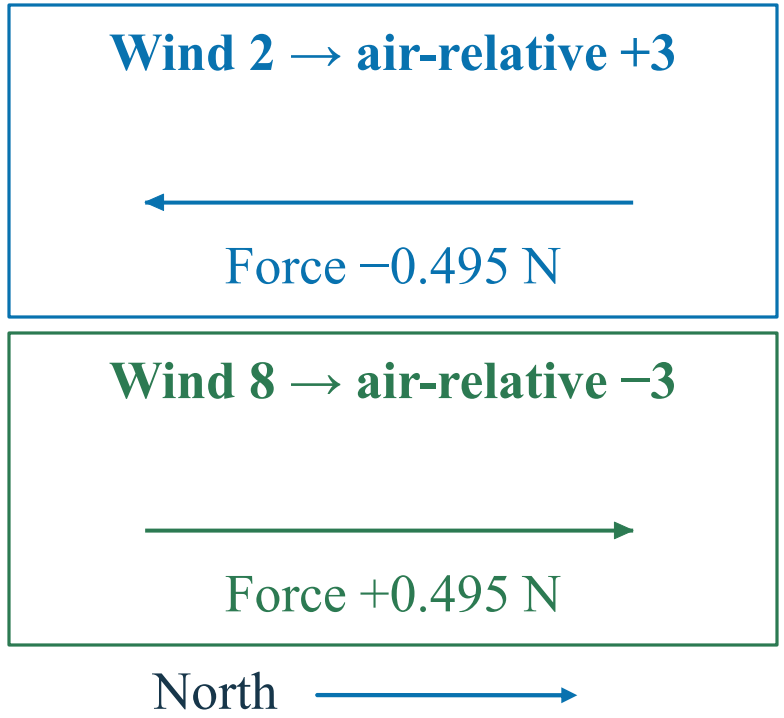
- Connect wind through the air-relative velocity using the specified frame and units, then trace changes in loads and states.

A displayed wind arrow does not prove that wind acts on the dynamics.

4.12.3 Design Tasks and Course Projects: Q7—Deterministic Direction Check

Q7 Acceptance: Check Deterministic Cases Before Random Wind

- Align body and NED; set ground velocity to 5 m/s north and northward wind to 2 or 8 m/s. Vehicle air-relative north velocity $v_{a,N}$ is +3 or −3 m/s, respectively.
- Using $F_N = -0.055 v_{a,N}|v_{a,N}|$ (0.055 is a teaching coefficient) gives −0.495 or +0.495 N. Log wind, air-relative velocity, aerodynamic force, and simulation time.
- Add documented random wind. Reproduce it with identical parameters and seed. Check sample mean and variance using record-length tolerances; exact equality to theory is not required.

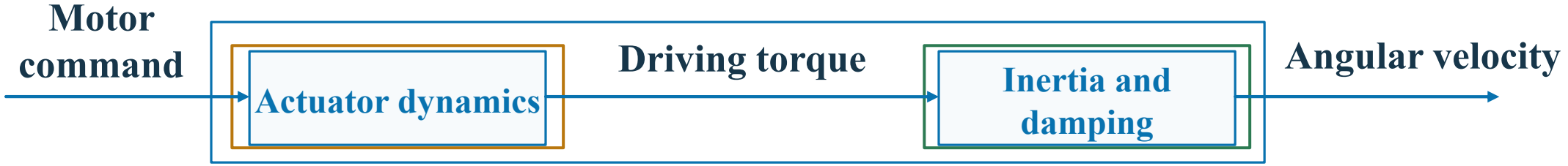


Compare random responses only after direction and dimensional checks pass.

4.12.3 Design Tasks and Course Projects: Q9—Roll Identification Boundary

Textbook Q9: State Where the Roll Input Enters

- **Textbook Q9:** Refine actuator and damping parameters with system identification, using roll as the example. Review slide decks 4.6, 4.7 and textbook Sec. 4.2.2.4.



① Bench: Command → torque ③ Overall: includes actuator dynamics ② Body: Torque → angular velocity

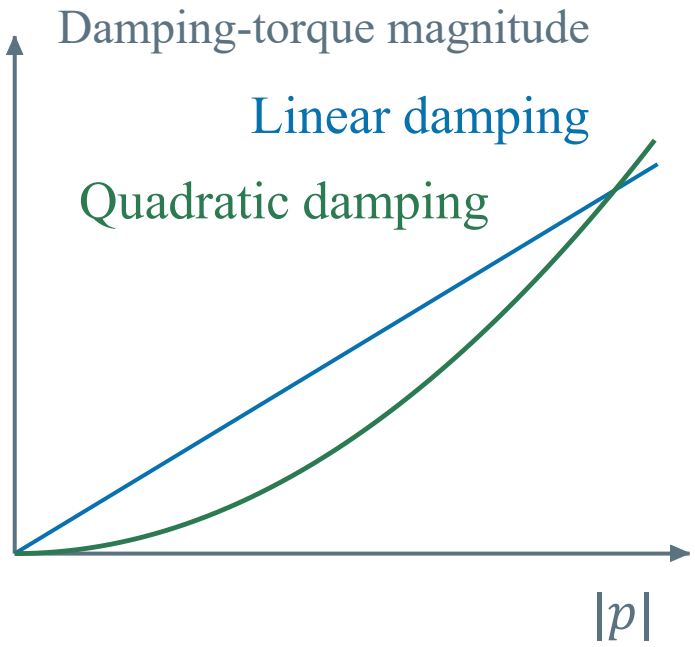
- With inertia, initial state, and conditions known, choose identifiable parameters and split estimation/validation data. Identify actuator time constant separately from bench steps; match motor, propeller, and voltage.
- Synthetic data test recovery of known parameters. Real-vehicle accuracy requires independent measurements from a matching source and setup.

Different input boundaries identify different transfer relationships.

4.12.3 Design Tasks and Course Projects: Q9—Damping Structure

Q9 Discussion: Damping Structure Defines the Fitted Parameter

- Neglect cross-coupling locally on one axis; τ_x excludes damping to be estimated. M0's net-torque log already includes retained damping: do not subtract it twice. The analytical copy in slide deck 4.11 has zero damping and cannot identify it.
- Known J_x permits estimation of d_p. M0 uses $-C_p p|p|$, with no first-order term at zero angular velocity; estimate C_p with the quadratic structure.
- Normalized input also includes actuator dynamics. Without independent inertia or torque information, several parameters may not be uniquely separable.



$$J_x\dot{p} = \tau_x - d_p p \quad \text{or} \quad J_x\dot{p} = \tau_x - C_p p|p|$$

Linear viscous and quadratic damping differ; their coefficients are not interchangeable.

4.12.3 Design Tasks and Course Projects: Q9—Estimation and Independent Validation

Q9 Implementation: Preprocess, Estimate, Then Validate Independently

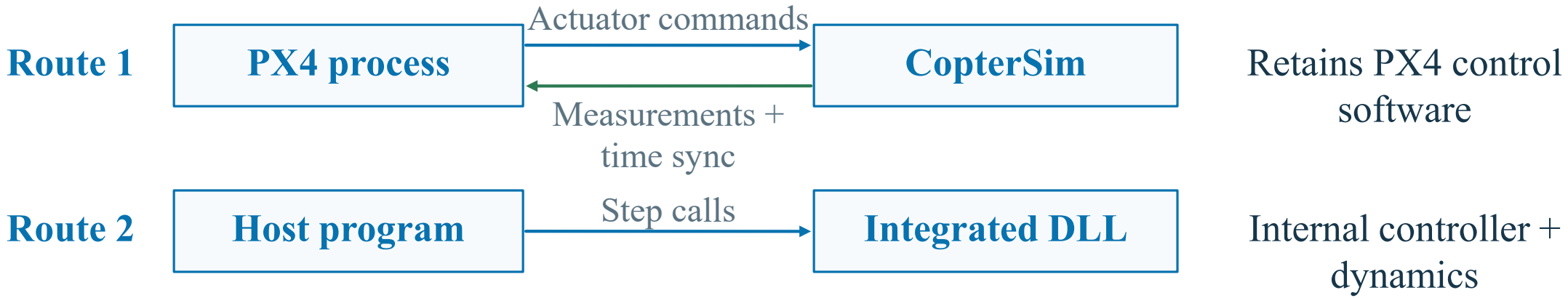
- Check timestamps, units, bias, and saturation. Align inputs and outputs by when they take effect. Differentiating angular velocity amplifies noise; document the treatment.
- Fit the estimation segment and record structure, objective, and parameter constraints. Simulation-based fitting must also match the experiment segment's initial state.
- Freeze the model, then compare peaks, steady values, delays, and residuals on an independent case. If it guides retuning, validate again on another untouched segment.

Independent validation reveals parameters that fit only the original data.

4.12.3 Design Tasks and Course Projects: Q10—Integrated Models and Acceleration

Textbook Q10: How Can Integrated Models Accelerate Experiments?

- **Textbook Q10:** Build an RflySim integrated model for accelerated software-in-the-loop simulation and training (SITL; textbook SIL). See slide deck 4.11 and textbook Sec. 4.2.4.



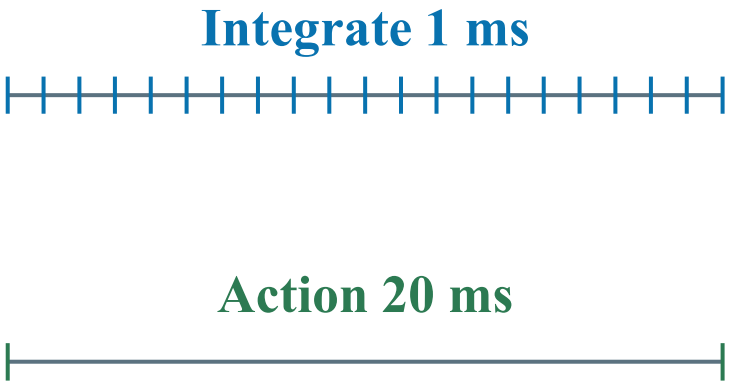
- **Same-task comparison:** Route 2's dynamic-link library (DLL) has a controller different from PX4. Compare this training surrogate with the PX4 chain. A training environment needs at least reset, action, observation, and termination.

Accelerated training must preserve simulation time and validate against the target system.

4.12.3 Design Tasks and Course Projects: Simulation Time and Wall Time

Acceleration Depends on Scheduling Simulation Time

- Fix initial state, seed, and action sequence. Hold each action for integer base steps. ModelUpdate advances DLL model time and discards sub-step remainders (slides 4.10, “How ModelUpdate Converts Time to Steps”).
- Compare trajectories and termination events at matching simulation times, then measure wall time. Record hardware, parallel count, display settings, and step size separately.
- Match action limits, observations, and collision termination to the task. A collision fails even with small endpoint error.



$$\rho = \frac{\Delta t_{\text{sim}}}{\Delta t_{\text{wall}}}, \qquad N = \text{int}(\Delta t / h)$$

Advance simulation time by model steps; wall time measures performance.

4.12.3 Design Tasks and Course Projects: Q10—Training Episode Interfaces

Q10 Diagram: Which Interfaces Does One Episode Need?

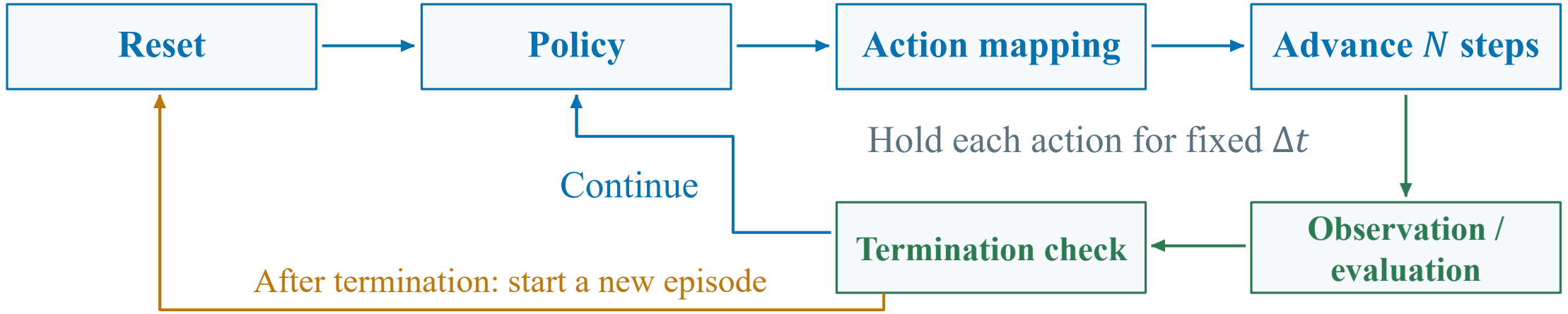


Figure: Actions, states, and episode boundaries share one time base

- Fix initial state and random conditions for each episode; advance the same simulation duration per action.

Fixed-duration steps make action effects comparable; reset gives each new episode a defined start.

4.12.3 Design Tasks and Course Projects: A4—Choose a Model Change

A4 Topic: One Model Change You Can Explain

Complete any one route. Start in subdirectories 01, 03, or 04 (read Readme.pdf; slide deck 4.11):
[installation directory]/RflySimAPIs/4.RflySimModel/1.BasicExps/e6_ModelingLearningLab/

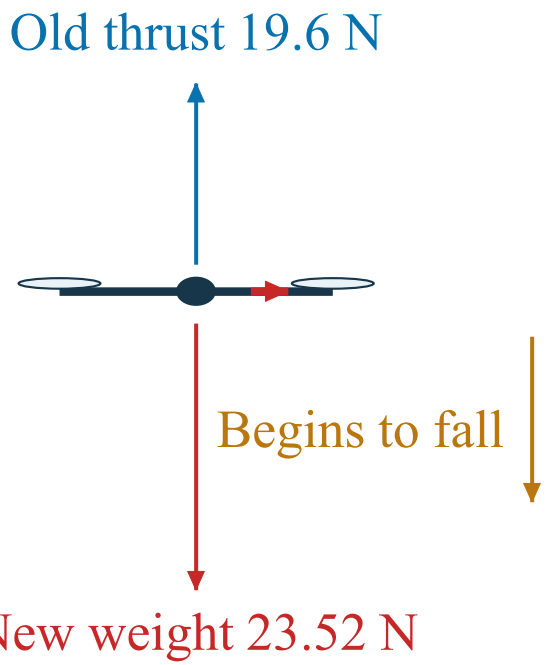
Route	Change	Prediction	Deliverables
Basic	Same multicopter model: mass +20%	Hover thrust; instantaneous acceleration	Update parameter source; compare matched conditions
Advanced	Update one actuator submodel using matching independent propulsion data	Error change before/after update	Retain data unused in tuning; explain before/after errors
Transfer	Ground-vehicle baseline (4.8/4.11): change geometry or drag parameters	Motion response after the change	Restate states, inputs, and constraints; do not reuse multicopter thrust formulas

4.12.3 Design Tasks and Course Projects: A4—Added Payload Example

A4 Basic: A Reproducible Reference for 20% More Mass

- **Independent ideal case:** 2 kg, $g = 9.8$ m/s², original thrust 19.6 N. Increase mass to 2.4 kg; new equilibrium thrust is 23.52 N.
- With old thrust, level attitude, airborne, and no other forces, down acceleration is 1.633 m/s². Ground support would mask this motion.
- New trim thrust: 5.88 N/rotor. With the same thrust coefficient, speed scales by $\sqrt{1.2} = 1.0954$, rather than by +20%.

$$a_D = 9.8 - \frac{19.6}{2.4} \approx 1.633 \text{ m/s}^2$$



With thrust unchanged, added mass changes acceleration first.

4.12.3 Design Tasks and Course Projects: A4—Fix Comparison Conditions

Fix the Comparison Conditions Before A4

- Fix the object, parameters, initial state, input, environment, time step, and stop conditions; change only one item. Predict analytically first, then define error measures and the comparison interval.
- Deliver the model/scripts, parameter source, raw logs, processing code, and entry point. Distinguish the distributed DLL from a DLL you generated.
- Include at least one repeat and an independent case. Link the question, assumptions, prediction, results, and limits so another person can reproduce the experiment from files.

Conditions

Parameters + initial state + input

Evidence

Raw logs + processing code

Reproduction

Command + output + criteria

Define the question and criteria before running to avoid selecting favorable results afterward.

4.12.3 Design Tasks and Course Projects: A4—Evaluation Evidence

A4 Evaluation: Physics, Parameters, and Implementation

Grade evidence supporting the project goal, rather than model complexity or screenshot count.

Criterion	Weight	Checkable evidence
Physical explanation	25%	States, loads, units, and assumptions match implementation
Parameters and data	25%	Clear parameter/data sources and operating conditions
Interface implementation	20%	Checkable initial state, input, advancement, and output
Validation	20%	Analytical, numerical, or independent data support conclusions
Reproduction files	10%	Model/scripts, parameter source, logs, processing code, entry point

Hardware-accuracy claims need matching independent measurements; judge sensitivity-only work by its stated goal.

Chapter Outline

Exercises, Case Analysis, and Course Projects

Chapter 4
Vehicle Motion Modeling and Simulation
4.1 Modeling Fundamentals and Kinematics
4.2 Rigid-Body Dynamics and Solvers
4.3 Propulsion, Forces, and Observations
4.4 Trim, Linearization, and Dynamics
4.5 Multicopter Modeling and Simulation
4.6 Measurement and Static Identification
4.7 Dynamic Identification and Refinement
4.8 Typical Vehicles and Extensions
4.9 Modeling Tools and Deployment
4.10 API and SDK Modeling Interfaces
4.11 Examples and Integrated Practice
4.12 Exercises and Course Projects

4 Units in This Section

4.12.1 Concept and Principle Review

Textbook: Sec. 4.4, Exercises 1, 2, 5

4.12.2 Calculations and Approximation Error

Textbook: Sec. 4.4, Exercises 3, 4, 6, 8

4.12.3 Design Tasks and Course Projects

Textbook: Sec. 4.4, Exercises 7, 9, 10

4.12.4 Result Discussion and Chapter Summary

Textbook: Sec. 4.1.2.2, Fig. 4.6; chapter review

4.12.4 Result Discussion and Chapter Summary: Case Discussion—Layered Decisions

Case Discussion: Each Piece of Evidence Answers One Question

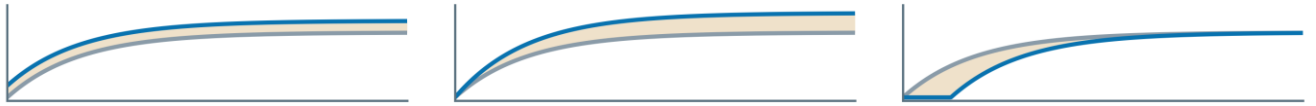
Evidence	What it shows	Still needed	Decision
DLL loads; time advances by the step size	Interfaces and stepping work	Same-initial-state comparison	**Pass: Interfaces**
Synthetic validation: RMSE 2.148 rad/s (4.7 teaching data)	Recovers known parameters	Independent measurements	**Pass: Calculation chain**
Fit 82.24%; positive real pole (4.7 historical record)	Similar short-term curves	Check poles and physics	Investigate: Do not deploy
Independent measured validation error far exceeds estimation error	Parameter or structure mismatch	Diagnose from residuals	Investigate: Parameters/structure

Residuals: gray = measurements, blue = model. RMSE = root mean square error.

Left—offset: check bias/operating point

Center—scale: check gain/saturation

Right—delay: check time bases/latency



Validate each claim; no result substitutes for another.

4.12.4 Result Discussion and Chapter Summary: Fig. 4.6—Model and Hardware Evidence

Textbook Fig. 4.6: Connect Model and Hardware Evidence

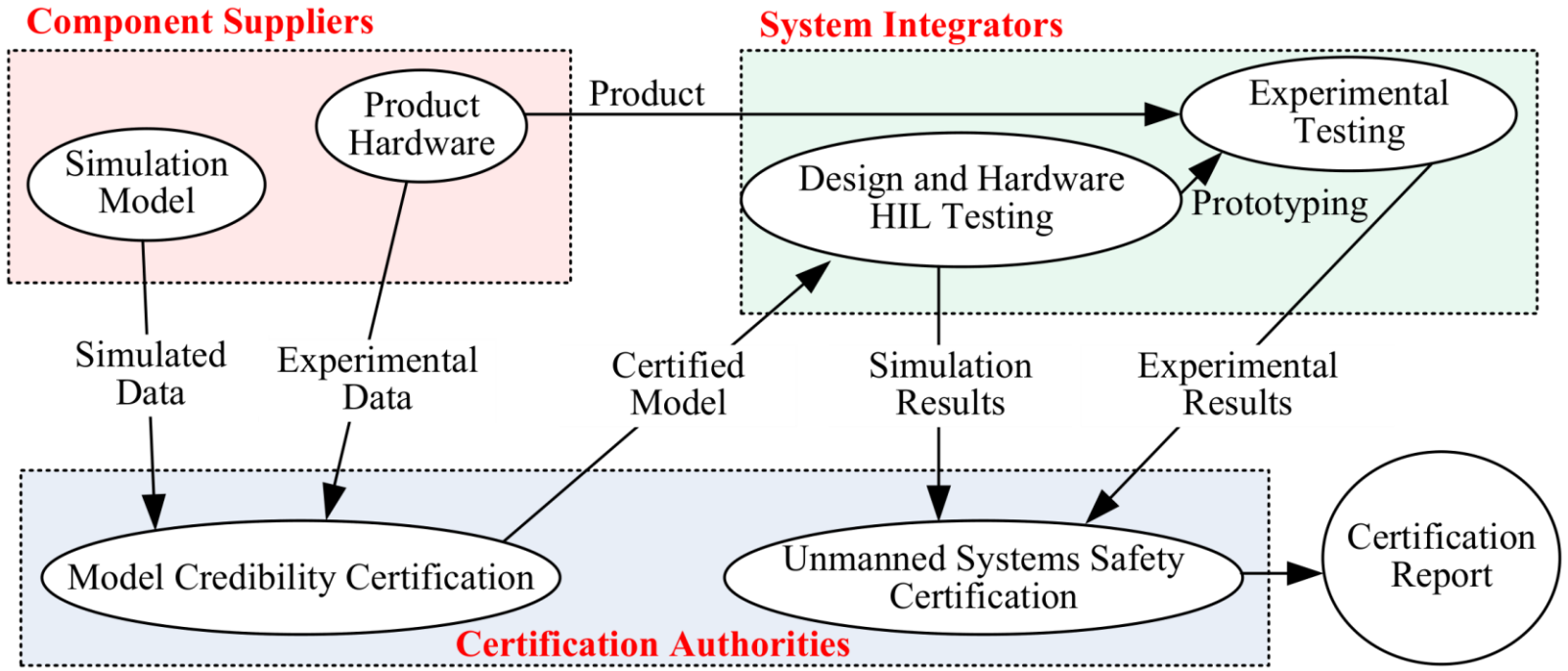


Fig. 4.6 Certification System Architecture under a Unified Modeling Framework (textbook Sec. 4.1.2.2)

Red—Component suppliers: Hardware + digital models

Green—System developers: In-the-loop simulation and experiments

Blue—Certification authorities: Model credibility and system safety

Shared digital models must retain sources, validation conditions, and scope of use.

4.12.4 Result Discussion and Chapter Summary: Fig. 4.6—Evidence Responsibilities

Fig. 4.6 Discussion: Different Parties Need Different Evidence

- Suppliers state interfaces, parameter sources, and conditions. Developers validate integrated interfaces and tasks and recheck component assumptions.
- Independent evaluators assess agreement with hardware and suitability for the intended use.

- State a checkable scope, e.g., independent step tests show that the model predicts this component's speed transient within the listed voltage and input ranges.

Component supplier

Interfaces, parameters, conditions

System developer

Integration and task validation

Independent evaluator

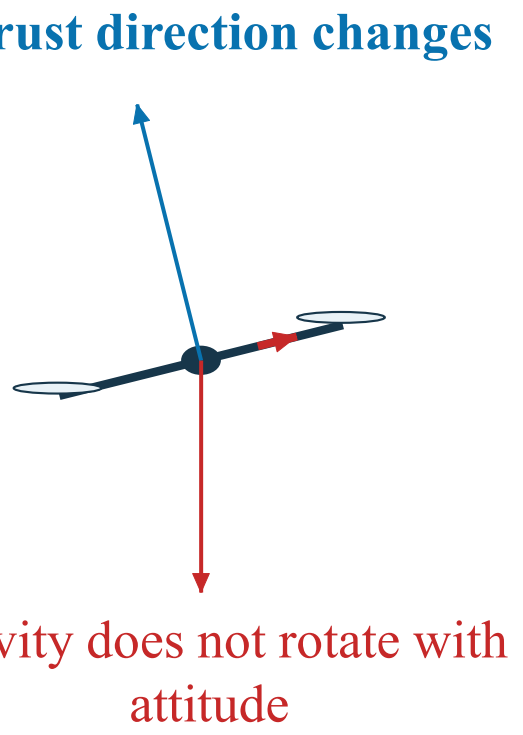
Intended use and hardware agreement

Model documentation, system tests, and independent evaluation must trace to the same object.

4.12.4 Result Discussion and Chapter Summary: Motion Models and Validation

Chapter Review: From Motion Equations to a Usable Model

- Turn equations/components into runnable models with defined inputs, time, and outputs. Refine parameters from data; validate independently on new vehicles and conditions.
- Zero-drag beginner case (4.1, 4.2): Balanced thrust after a climb retains upward velocity; braking needs opposite acceleration. Equal normalized inputs can yield different paths if initial states, parameters, or control boundaries differ.
- A4 links prediction, execution, and evidence into complete modeling. Plausible curves still need units, time bases, comparable conditions, and a stated conclusion scope.

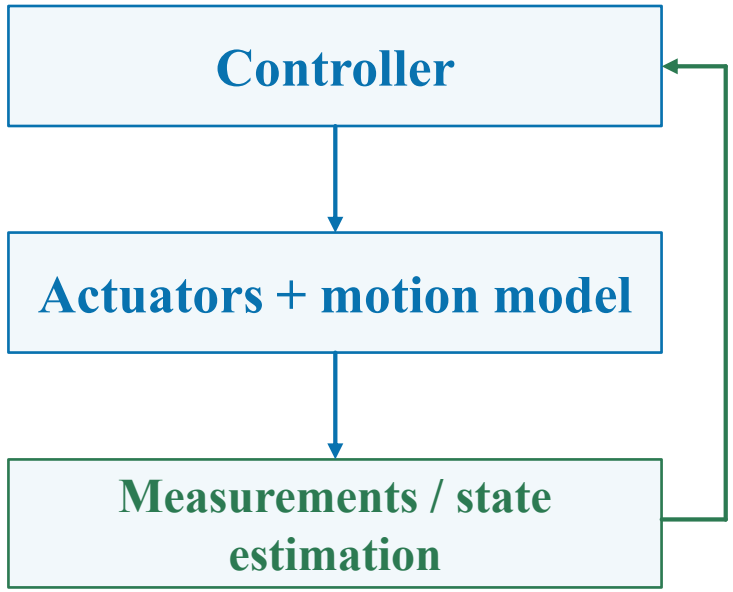


Modeling connects physics, implementation, and validation evidence into one chain.

4.12.4 Result Discussion and Chapter Summary: Model Boundaries for Chapter 5

Enter Chapter 5 with a Clearly Defined Model

- **Carry into Chapter 5:** Validated states, input ranges, actuator dynamics, measurements, and their delay/error characteristics for stabilization and tracking control.
- Retain the same model baseline. Compare controller changes under common conditions to distinguish control effects from parameter changes.
- Trace every motion curve to inputs, speed, loads, and states. Preserve sources and independent validation so others can reproduce results.



Next chapter: Design feedback

Know what the model describes before designing its controller.

Chapter Summary

Chapter Summary: Core Concepts for Describing Motion

- **Frames and attitude (4.1):** NED ground and FRD body frames. Euler angles and rotation matrix R describe attitude; R maps body vectors to ground. Quaternions store attitude and update from body angular velocity.
- **Rigid body and integration (4.2):** Newton–Euler equations describe translation and rotation. Numerical integration advances states by time steps, affecting accuracy and stability.
- **Components and environment (4.3):** An actuator cascades a steady-state function and first-order dynamics; thrust is $T = c_T \varpi^2$. Wind, atmosphere, and specific force enter loads and observations.
- **Dynamic systems (4.4):** State space and transfer functions express the same local model. Step response reveals time constants; frequency response reveals magnitude ratio and phase.

Chapter Summary

Chapter Summary: Modeling, Identification, and Reproduction

- **Implementation and interfaces (4.5, 4.9, 4.10):** Progressive programming, code generation/deployment, and application programming interface (API) / software development kit (SDK) interfaces turn equations into callable models.
- **Parameter measurement (4.6):** Measure bench parameters; identify steady-state relationships from static data.
- **Linearization and discretization (4.4):** Trim and linearize for a local model; discretize continuous dynamics under zero-order hold for sampled updates.
- **Dynamic identification (4.7):** Identify dynamic parameters; validate with free-run simulation and independent data.
- **Transfer and reproduction (4.8, 4.11):** Restate states, inputs, and constraints for each new vehicle; reproduce example results under matched conditions.

Chapter Summary

Chapter Summary: Where Model Conclusions Apply

- **Local models (4.4; "Small-Angle Approximation Error" here):** Linearization and small-angle approximations apply near an operating point. Set the range by allowed error, e.g., about 14° tilt for 1%.
- **Discretization (4.4):** Forward Euler for a first-order model converges when $0 < T_s < 2\tau$. At the upper limit, error oscillates at constant amplitude; beyond it, error diverges.
- **Data sources (4.6, 4.7):** Synthetic teaching data test the calculation chain. Hardware-accuracy claims need matching independent measurements.
- **Input boundaries (4.7):** Command→torque, torque→angular velocity, and command→angular velocity are different objects. Conclusions apply to the chosen boundary.
- **Operating conditions (4.6, 4.8):** Parameters vary with voltage, propeller, and load. State conditions, input range, and validation method.
- **Time advancement (4.9, 4.10):** Simulation time advances by base steps. Acceleration reduces wall time without changing simulation-time meaning.

Chapter Summary

Chapter Summary: Common Errors

- **NED signs:** z_D is positive down; height is $H = -z_D$. Upward thrust has a negative z component (4.1, 4.3).
- **R and R^T:** R maps body vectors to ground; use R^T for the reverse (4.1).
- **Count gravity once:** Total force includes gravity once. Specific force is net nongravitational force divided by mass (4.2, 4.3).
- **Degrees and radians:** Use rad for trigonometry, small-angle approximations, and attitude integration (4.1).
- **rpm and rad/s:** $\varpi = 2\pi n/60$, with n in rpm; c_T uses rad/s (4.3, 4.6).
- **One-step prediction and free-run simulation:** One-step prediction uses measured states every step; validate with free-run simulation (4.7).
- **Closed-loop data and the plant:** Reference-to-output is closed-loop. Actual plant input/output data and suitable methods can still identify the plant (4.7).
- **Training error and independent validation:** Training error shows fit only; validation data must be unused in estimation (4.7).