Chapter Outline

Chapter 1 Review and Exercises

Chapter 1 Introduction

1.1 Concepts and Applications

1.2 History and Trends

1.3 Components and Architecture

1.4 Coordinate Systems and Control Principles

1.5 Assembly and Debugging

1.6 RflySim Toolchain

1.7 Learning Path and Experiments

1.8 Exercises and Discussion

3 Units in This Section

1.8.1 Chapter Review

Review of Concepts, Principles, Assembly, and Platforms

1.8.2 Exercise Key Points

Textbook: Sec. 1.7

1.8.3 Classroom Discussion

Transferring and Validating Autonomous Obstacle Avoidance Across Vehicles

1.8.1 Chapter Review: Shared Thread

From Understanding Tasks to Development and Validation

- **Concepts and development:** Distinguish systems by physical interaction, autonomy, and vehicle type; relate capability development to the history of robots, unmanned aerial vehicles (UAVs), and swarms. (Slides 1.1–1.2; textbook Sec. 1.1)
- **Architecture and principles:** Six layers organize perception, decision-making, and execution into a closed loop. Establish common coordinates and attitude representations before analyzing resultant forces and moments. (Slides 1.3–1.4; textbook Sec. 1.2)
- **Assembly and debugging:** Connect the three line types correctly; match autopilot orientation, motor numbers, and rotation to configuration. Validate with propellers removed before fitting them for flight. (Slides 1.5; textbook Sec. 1.3)
- **Platforms and learning:** RflySim connects assembly testing, algorithm development, hardware-in-the-loop simulation (HITL; HIL in the textbook), and real-vehicle deployment. Reproduce and modify examples through the unified entry point. (Slides 1.6–1.7; textbook Secs. 1.4–1.6)

Establish the task and system relationships first, then test the approach through models, experiments, and real-vehicle results.

Chapter Outline

Chapter 1 Review and Exercises

Chapter 1 Introduction

1.1 Concepts and Applications

1.2 History and Trends

1.3 Components and Architecture

1.4 Coordinate Systems and Control Principles

1.5 Assembly and Debugging

1.6 RflySim Toolchain

1.7 Learning Path and Experiments

1.8 Exercises and Discussion

3 Units in This Section

1.8.1 Chapter Review

Review of Concepts, Principles, Assembly, and Platforms

1.8.2 Exercise Key Points

Textbook: Sec. 1.7

1.8.3 Classroom Discussion

Transferring and Validating Autonomous Obstacle Avoidance Across Vehicles

1.8.2 Exercise Key Points: Concepts and Architecture

Questions 1–2: Concept Relationships and System Roles

Q1: Relationships and differences among four concepts (Slides 1.1; textbook Sec. 1.1.1)

Embodied intelligence uses a body in an environment feedback loop; disembodied intelligence processes information without directly closing a physical interaction loop.

Intelligent unmanned systems perceive, decide, and act autonomously; those centered on a vehicle are intelligent unmanned vehicle systems.

Interaction, autonomy, and vehicle type are separate, intersecting dimensions; they do not form one ranking.

Q2: Six architectural layers, responsibilities, and data/control flows (Slides 1.3; textbook Sec. 1.2.1)

Body structure supports loads; perception and localization estimate state; control and decision-making generate commands; actuation and execution produce forces.

Environmental interaction: Wind, light, and terrain affect observations and forces. Swarm collaboration: Communication and cooperation among vehicles.

Closed loop: Measurement→estimation→decision is data flow; command→actuation→motion is control flow; motion is measured again.

1.8.2 Exercise Key Points: Coordinate Systems

Question 3: Relating Two Coordinate Conventions

Compare axes, applications, and the right-hand rule. What must change across interfaces? (Slides 1.4.2; textbook Sec. 1.2.3.2)

Comparison	NED / FRD	ENU / FLU
Ground Axes	North, East, Down	East, North, Up
Body-Frame Axes	Forward, Right, Down	Forward, Left, Up
Common Uses	Aviation Autopilots	Robotics and ROS Interfaces

Conversion: Ground frame NED(x, y, z)→ENU$(y, x, -z)$; body frame FRD(x, y, z)→FLU$(x, -y, -z)$.

Right-hand rule: Thumb along x, index finger along y, middle finger along z. With the thumb along a positive axis, curled fingers give positive rotation.

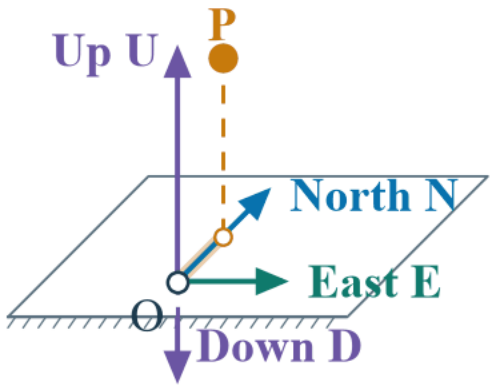
Both conventions are right-handed; interface conversion must align axis directions and rotation signs together.

1.8.2 Exercise Key Points: Coordinate Systems

Question 3: Same Position, Different Coordinate Components

NED→ENU: swap N/E, invert z

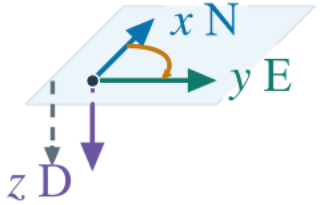
P: N 1 m, Up 2 m



P (m)	x	y	z
NED	1.0	0.0	−2.0
ENU	0.0	1.0	2.0

Thumb x · index y · middle $x \times y = z$

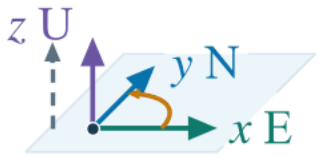
NED



✓ **Right-handed**

CW from top

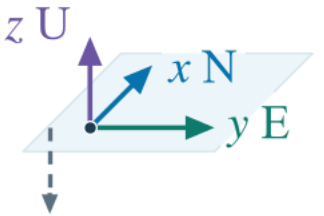
ENU



✓ **Right-handed**

CCW from top

Flip z only



✕ **Left-handed**

Middle ↓, z ↑

Figure: Two Component Sets for One Point and Right-Handedness Checks (Schematic)

Swap north/east components and reverse the vertical component to preserve right-handedness between NED and ENU.

1.8.2 Exercise Key Points: Attitude and Underactuation

Questions 4–5: Representation and Control Constraints

Q4: Gimbal-lock example and attitude-representation trade-offs (Slides 1.4.3; textbook Secs. 1.2.3.3–1.2.3.4)

- **Singularity example: At** $\theta = 90°$, $[\phi\ \theta\ \psi] = [0°\ 90°\ 0°]$ and $[90°\ 90°\ 90°]$ represent the same attitude.
- Trade-offs: Quaternions avoid singularities and support smooth interpolation, but must have unit length; q and $-q$ represent the same attitude.
  Rotation matrices simplify composition and vector transforms, but have 9 elements and must remain orthogonal.

Q5: Underactuation and external-command combinations (Slides 1.4.5; textbook Sec. 1.2.3.6)

- Underactuation: Fewer independent inputs than degrees of freedom. A quadcopter has 6 degrees of freedom and 4 inputs: total thrust and three moments. Horizontal motion requires tilt.
- Command combinations: Four targets—attitude + throttle; velocity + yaw rate; or position + yaw angle.

Attitude singularities arise from parameterization; underactuation limits arise from insufficient independent control actions.

1.8.2 Exercise Key Points: Forward-Flight Forces

Question 6: Conditions for Level, Constant-Speed Forward Flight

Mass m, total thrust f; nose-down pitch θ is negative. Derive level, constant-speed conditions, speed coupling, and risks.

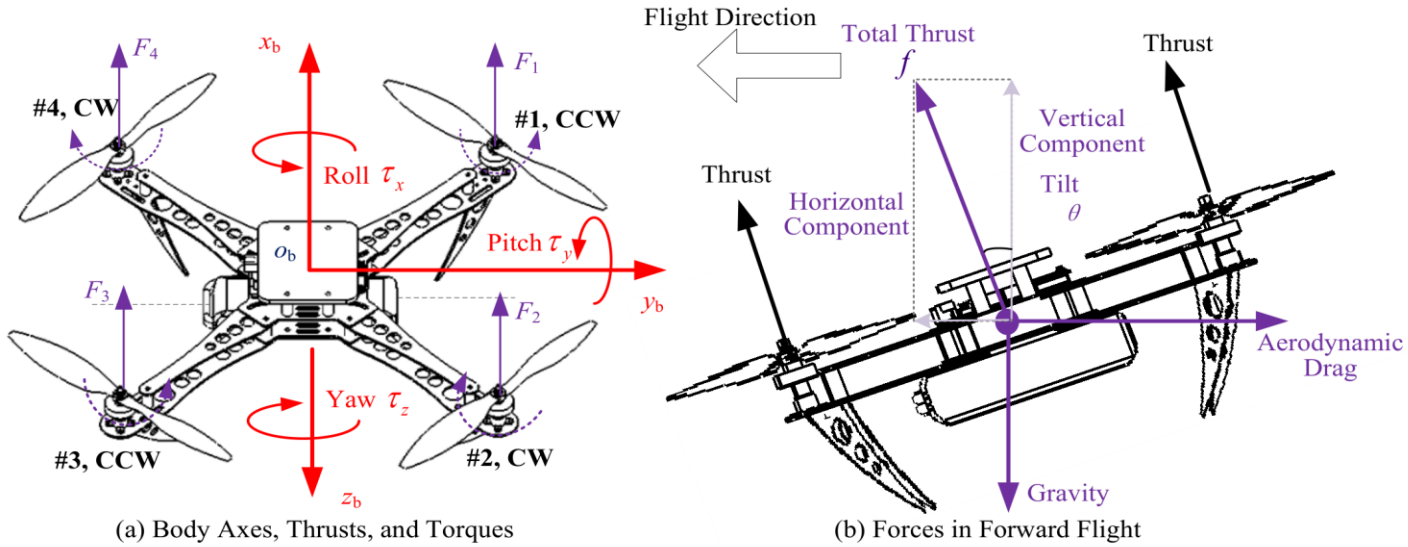


Fig. 1.12 Multicopter Force Model

Stable attitude, zero roll; v is airspeed

$$G = mg, \qquad D(v) = kv^2$$

Let α be the geometric tilt magnitude

$$\alpha = -\theta, \quad 0 \le \alpha < 90^\circ$$

$$f\cos\alpha = mg$$

$$f\sin\alpha = kv^2$$

$$v^2 = \frac{mg}{k}\tan\alpha$$

$$f = \frac{mg}{\cos\alpha}$$

Forward tilt increases required thrust; propulsion saturation may cause altitude loss and raise the risk of attitude instability.

1.8.2 Exercise Key Points: Forward-Flight Forces

Question 6: How Tilt Changes Speed and Thrust

Substitute into the two Q6 results, $v^2 = (mg/k)\tan\alpha$ and $f = mg/\cos\alpha$:
thrust is in units of weight mg; steady speed at 45° is 1. (Slides 1.4.4; textbook Sec. 1.2.3.5)

Tilt α	Total Thrust (mg)	Relative Speed
15°	1.04	0.52
30°	1.15	0.76
45°	1.41	1.00
60°	2.00	1.32

- Speed coupling: Greater tilt gives higher speed.
  From 45°→60°, speed rises about 32%, but thrust about 41%.
- Propulsion margin: With a thrust limit of $2mg$, maximum α is 60°.

  Further tilt leaves the vertical component below mg, causing altitude loss.
- Control margin: Near the thrust limit,
  less moment-adjustment capacity remains to reject gusts and other disturbances.

Greater tilt requires more thrust; altitude falls only when maximum thrust's vertical component is below weight.

1.8.2 Exercise Key Points: Forward-Flight Forces

Question 6: At What Forward Tilt Does Altitude Fall?

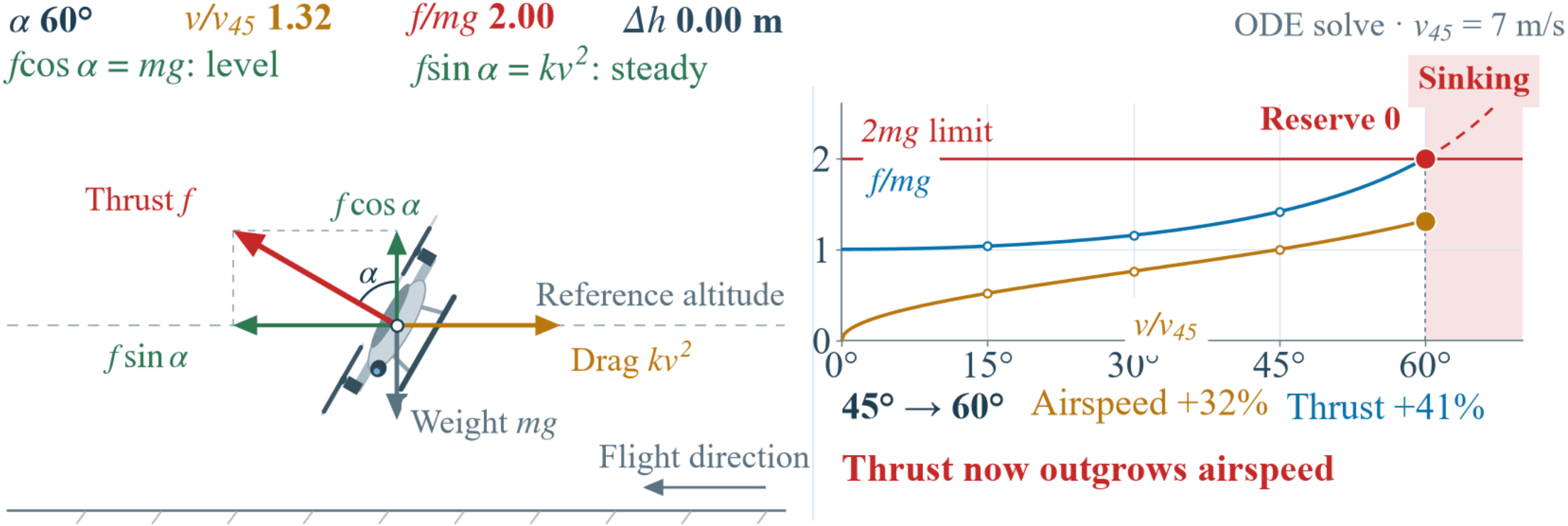


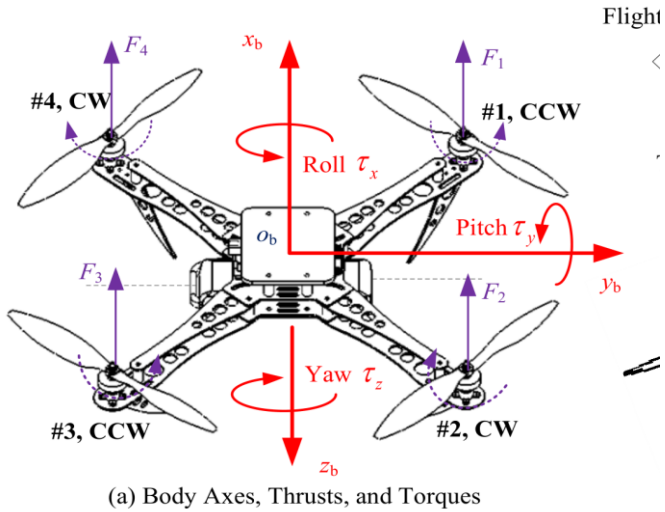
Figure: Forward Tilt, Steady Airspeed, and Thrust Limit (Offline Simulation of a Simplified Model)

After thrust reaches its limit, further tilt leaves insufficient vertical force to balance weight, and altitude begins to fall.

1.8.2 Exercise Key Points: Differential Yaw Control

Question 7: Adjusting Rotor Speeds for Right Yaw

Top view: Rotors 1 and 3 turn counterclockwise; 2 and 4 clockwise. Give a right-yaw differential strategy with total thrust unchanged.



(a) Body Axes, Thrusts, and Torques

Fig. 1.12(a) Attitude Angles and Forces

Speed Up 1 and 3; Slow Down 2 and 4

Counterclockwise rotors exert clockwise reaction torque $c_M\omega^2$ on the body.

$$f_i = c_T\omega_i^2, \quad f = \sum_{i=1}^{4} f_i$$

$$\omega_1^2 = \omega_3^2 = \omega_0^2 + \delta$$

$$\omega_2^2 = \omega_4^2 = \omega_0^2 - \delta$$

$$\Delta f = 0,\ \tau_z = c_M(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) = 4c_M\delta > 0$$

ω_0 is the equal-speed baseline; $\delta > 0$, with neither speed limit reached.

Increase squared speeds of rotors 1 and 3 and equally decrease 2 and 4, preserving total thrust while producing right-yaw moment.

1.8.2 Exercise Key Points: Rotation Conventions

Question 8: Transforming Body to Ground Coordinates

Given Z-Y-X Euler angles, write the rotation matrix from body b to ground e and use the

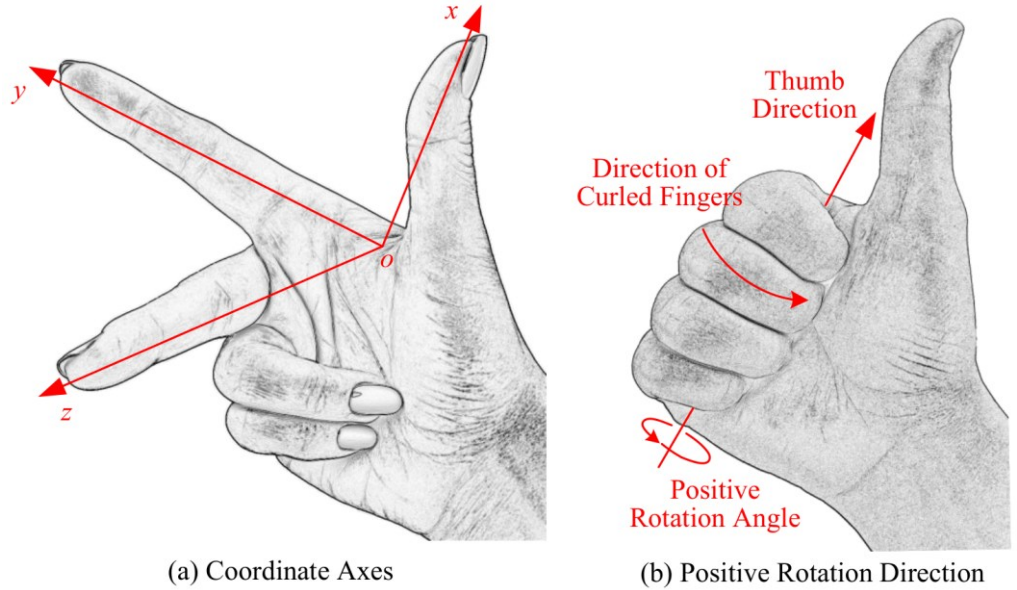


Fig. 1.10 Coordinate Axes and Positive Rotations Under the Right-Hand Rule

- **Vector convention:** $\mathbf{v}_e$ and $\mathbf{v}_b$ are components of the same vector.
- Intrinsic order: Yaw ψ about z, then pitch θ about the new y, then roll ϕ about the new x.
- Positive rotation: Thumb along the positive axis; curled fingers give positive rotation.

$$\mathbf{v}_e = R_b^e \mathbf{v}_b$$

$$R_b^e = R_z(\psi) R_y(\theta) R_x(\phi)$$

Rotation matrices have a definite meaning only after specifying transform direction, axis order, and positive angles.

1.8.2 Exercise Key Points: Rotation Matrices

Question 8: Start with Three Elementary Rotation Matrices

$$R_z(\psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}, \quad R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}$$

$c_\beta = \cos\beta, \quad s_\beta = \sin\beta, \quad \beta \in \{\phi, \theta, \psi\}$ **Composition:** $R_{\mathrm{b}}^{\mathrm{e}} = R_z(\psi)R_y(\theta)R_x(\phi)$

Reading the matrices: In the rotation-axis row and column, only the diagonal element is 1; others are 0. The other axes mix through cos and sin.
Each column gives a rotated axis in the original frame. The composite acts on column vectors from right to left.

Single-Axis Example: Pitch Up 30° About y Only

$\theta = 30°, \quad \psi = \phi = 0°$

Column 1 is the nose direction: Still north; negative z means upward.

$$R_y(30°)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_{\mathrm{b}} = \begin{bmatrix} \sqrt{3}/2 \\ 0 \\ -1/2 \end{bmatrix}_{\mathrm{e}}$$

Multiply the elementary matrices in Z-Y-X order to obtain the body-to-ground rotation matrix.

1.8.2 Exercise Key Points: Rotation Matrices

Question 8: Full Expression and Numerical Check

$$R_{\mathrm{b}}^{\mathrm{e}} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}$$

$$c_\beta = \cos\beta, \quad s_\beta = \sin\beta, \quad \beta \in \{\phi, \theta, \psi\} \qquad R_{\mathrm{e}}^{\mathrm{b}} = (R_{\mathrm{b}}^{\mathrm{e}})^{\mathsf{T}}$$

Two-Angle Check: Nose East and 30° Up (NED)

$$\psi = 90^\circ, \quad \theta = 30^\circ, \quad \phi = 0^\circ$$

Column 1 is the nose direction: Eastward; negative z means upward.

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_{\mathrm{b}} \longmapsto \begin{bmatrix} 0 \\ \sqrt{3}/2 \\ -1/2 \end{bmatrix}_{\mathrm{e}}$$

A two-angle check tests rotation order; the reverse transform uses the original matrix transpose.

1.8.2 Exercise Key Points: Rotation Order

Question 8: Swapping Rotation Order Changes Attitude

Z-Y-X: Yaw 90°, Then Pitch Up 30° About the New y

$$R_z(90°)R_y(30°)\begin{bmatrix}1\\0\\0\end{bmatrix}_{\mathrm{b}} = \begin{bmatrix}0\\\sqrt{3}/2\\-1/2\end{bmatrix}_{\mathrm{e}}$$

Nose east and 30° upward.

Reversed: Pitch Up 30°, Then Turn 90° About the New z

$$R_y(30°)R_z(90°)\begin{bmatrix}1\\0\\0\end{bmatrix}_{\mathrm{b}} = \begin{bmatrix}0\\1\\0\end{bmatrix}_{\mathrm{e}}$$

Nose level east; body banked 30° right.

- **Component order:** Column vectors are north, east, down; negative z means upward.
- **Checking method:** Use two nonzero angles and see whether column 1 captures heading and nose-up tilt together.
  With only one nonzero angle, both orders give the same result.

The same angles in different orders give different attitudes; check with two nonzero angles.

1.8.2 Exercise Key Points: Rotation Order

Question 8: Check Rotation Order with Nose and Wing Directions

- **Common start:** Nose north, wings level. Each step rotates about the current body axis.
- **Yaw then pitch:** Face east, then pitch up 30° about the new y; the nose gains an upward component.
- **Pitch then yaw:** Rotate about the tilted new z. The final nose is level east, with 30° right bank.
- **Vector check:** The red nose vector is matrix column 1. Components are north, east, down.

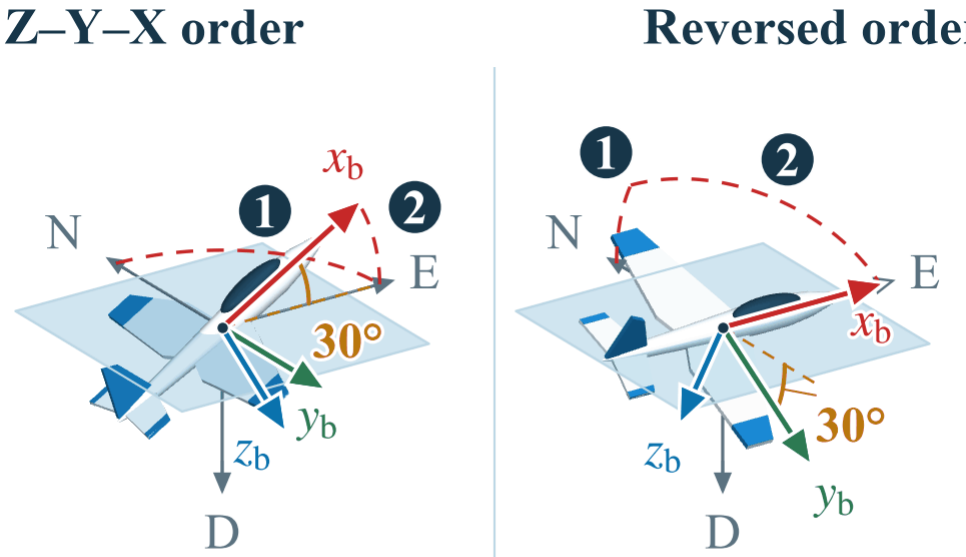


Figure: Two Intrinsic Rotation Orders (Schematic)

Intrinsic rotations use body axes updated by each motion; swapping order therefore changes final attitude.

1.8.2 Exercise Key Points: Vehicle Selection

Question 9: Aircraft Trade-Offs for Valley Mapping

For vertical takeoff and landing (VTOL) in a narrow valley plus long-range cruise mapping, compare three aircraft types and recommend one with trade-offs.

Aircraft Type	Takeoff/Landing Needs	Cruise and Mapping	Main Cost
Multicopter	Vertical Takeoff and Landing in Place	Convenient hover mapping Rotors support weight; high energy use	Usually tens of minutes endurance Long range requires an energy budget
Fixed Wing	Usually a runway, catapult, or parachute recovery	Wings support weight Endurance can reach hours	Takeoff/landing site needs restrict deployment
Hybrid VTOL	Vertical Takeoff and Landing	Wings support weight in cruise Lower energy use, longer endurance	Additional rotor propulsion Reduced payload, complex transition

The task favors hybrid VTOL, but final selection still depends on range, payload, and transition-flight conditions.

1.8.2 Exercise Key Points: Wiring Assembly

Question 10: Three Line Types—Key Points and Risks

Using a multicopter, list assembly points and common risks. (Slides 1.5.1–1.5.2; textbook Sec. 1.3.1)

Line Type	Assembly Points	Common Risks
Propulsion Power	Check polarity, wire gauge, and connectors Sound solder joints, secured and insulated	Reversed polarity or shorts Poor joints overheat; excessive voltage drop
Component Power	Check rated voltage and regulation Provide supply margin and ground connections	Overvoltage damage, undervoltage restarts Supply overload or poor common ground
Control Signals	Match protocol, port, and signal ground Keep clear of power wires; secure harnesses	Wrong ports or protocol mismatch Interference, packet loss, abnormal outputs

Wire according to voltage, current, and signal function; secure mounting, insulation, and common ground make connections reliable.

1.8.2 Exercise Key Points: Preflight Checks

Question 10: From Propeller-Off Validation to First Flight

Give the essential debugging and safety checklist before the first flight. (Slides 1.5.3–1.5.4; textbook Sec. 1.3.3)

- **Firmware and airframe:** Flash hardware-compatible firmware and select the preset and motor layout matching the aircraft.
- **Before propeller-off power-on:** Check structural fasteners, center of gravity, wiring polarity, and insulation; remove loose objects.
- **State and inputs:** Check mounting orientation, attitude display, radio directions, and modes; calibrate sensors and radio control.
- **Propeller-off outputs:** Check each motor's number and rotation, plus arming, link-loss, and low-battery protection responses.
- **Fit propellers with power off:** Confirm checks passed, match propeller type and rotation, and secure them. Clear the area and recheck.
- **First flight:** Set geofence and return conditions, agree abort conditions, and check battery, positioning, site/airspace, and weather. Start with brief low-altitude takeoff and landing, then inspect aircraft and logs.

Validate with propellers removed before fitting them; confirm the environment and stop conditions before the first real flight.

Chapter Outline

Chapter 1 Review and Exercises

Chapter 1 Introduction

1.1 Concepts and Applications

1.2 History and Trends

1.3 Components and Architecture

1.4 Coordinate Systems and Control Principles

1.5 Assembly and Debugging

1.6 RflySim Toolchain

1.7 Learning Path and Experiments

1.8 Exercises and Discussion

3 Units in This Section

1.8.1 Chapter Review

Review of Concepts, Principles, Assembly, and Platforms

1.8.2 Exercise Key Points

Textbook: Sec. 1.7

1.8.3 Classroom Discussion

Transferring and Validating Autonomous Obstacle Avoidance Across Vehicles

1.8.3 Classroom Discussion: Transfer Across Vehicles

Transfer UAV Obstacle Avoidance to a Front-Wheel-Steered Car

Which modules can be reused? Which models or interfaces must change? How can successful transfer be validated?

- **Reusable:** Obstacle-perception, mapping, and path-planning approaches, plus message formats and experiment metrics.
- **Adapt:** Motion model, feasible paths, turning and braking constraints, sensor mounting, and actuator interfaces.
- **Constraint differences:** A quadcopter has 6 degrees of freedom and 4 independent inputs; it can hover, move sideways, or climb over obstacles. A conventional front-wheel-steered car has 3 planar degrees of freedom, with drive and steering inputs. It cannot move directly sideways and must route around obstacles within its turning radius.
- **Validation:** Reproduce a baseline using the same task metrics, then compare simulation, HITL, and real-vehicle responses in sequence.

Example: Avoid the same obstacle and compare collision rate, minimum clearance, and task completion time.

Task methods transfer across vehicles; motion constraints, control interfaces, and validation conditions must suit the new vehicle.

Chapter Summary

Closing Chapter 1 and Preparing for Chapter 2

- **Chapter method:** Start from tasks and concepts, organize modules with architecture, and explain behavior through motion models and experiments.
- **Common errors:** Changing only z in frame conversion; confusing Euler-angle order with transform direction; fitting propellers before propeller-off validation; balancing differential yaw through speeds rather than squared speeds.
- **Homework:** Complete the ten textbook questions. Explain relationships in conceptual answers, assumptions and checks in calculations, and trade-offs in applications.
- **Review locations:** Slides 1.4 (textbook Secs. 1.2.2–1.2.3) cover coordinates and control; Slides 1.5 (textbook Sec. 1.3) cover assembly/debugging; Slides 1.6–1.7 (textbook Secs. 1.4–1.6) cover platforms and examples.
- **Chapter 2 preparation:** Organize the computer environment and introductory examples. Use the unified entry point to find tools, resources, and environment checks.

Understand system relationships and reason through exercises, then test understanding through platform setup and example reproduction.