Chapter Outline

Exercises and Classroom Discussion

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.9.1 Concepts and Methods
Textbook: Sec. 8.4, Questions 1–5

8.9.2 Geometric Calculations
Textbook: Sec. 8.4, Question 6

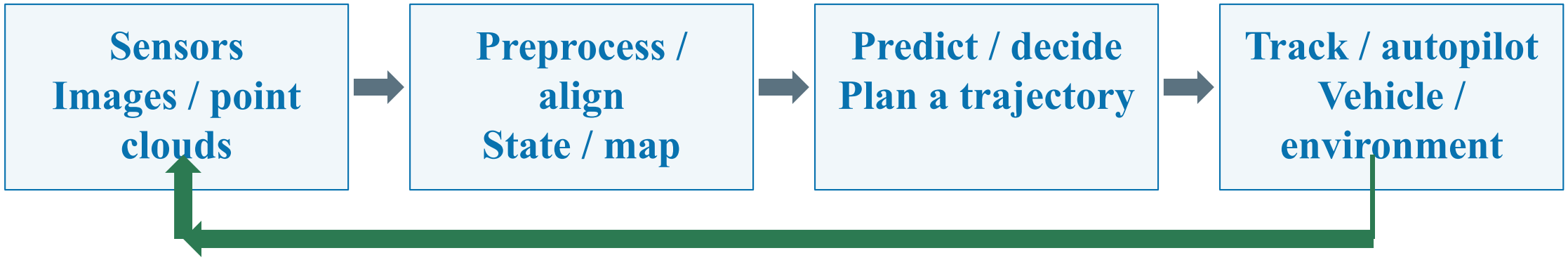
8.9.3 Perception and Decision-Making Design
Textbook: Sec. 8.4, Questions 7 and 9

8.9.4 System Development and Transfer
Textbook: Sec. 8.4, Questions 8 and 10

8.9.1 Concepts and Methods: Q1 · Closed-Loop Modules

From Observations to Estimates, Plans, and Actions

Textbook Sec. 8.4 · Q1: Identify the key modules and data flow in a multimodal perception and intelligent decision-making loop.



Observation feedback carries acquisition timestamps. Pose, maps, targets, and uncertainty inform decisions; trajectory references enter tracking control.

Actions change motion and observations; new observations correct estimates and subsequent actions.

8.9.1 Concepts and Methods: Q1 · Delay and Action

How Perception Delay Changes Stopping Distance

Static obstacle; straight-line speed 2 m/s; perception + actuation delay 0.15 s. Deceleration 2 m/s²; extra margin 0.30 m.

$$d_{\text{req}} = v\tau + \frac{v^2}{2a} + m = 2 \times 0.15 + \frac{2^2}{2 \times 2} + 0.30 = 1.60\,\text{m}$$

Delay adds reaction distance; braking adds the speed-squared term. The margin covers other errors.

$$d_{\text{obs}} - d_{\text{req}} = 3.00 - 1.60 = 1.40\,\text{m}$$

Brake before clearance drops below 1.60 m. Recheck clearance, speed, and delay after each observation.

8.9.1 Concepts and Methods: Q1 · Evidence of Feedback

Observable Checks for a Working Closed Loop

Data	Observable result	Check
Measurements / time	Images, point clouds, and inertial data use acquisition time	Buffer new data; mark stale data invalid
State / map	Observations update pose, targets, maps, and uncertainty	Frames/times agree; innovations are interpretable
Trajectory / execution	Log timed references separately from actual motion	Tracking error converges; timing and dynamic limits are met
Feedback / degradation	Reobserve and correct actions after disturbances	Sensor loss triggers timeout, braking, or degraded operation

Ground truth evaluates the estimator. Using it as estimator input changes the perception test.

8.9.1 Concepts and Methods: Q2 · Dense and Sparse Flow

Choose Flow Coverage to Match the Task

Textbook Sec. 8.4 · Q2: Compare dense and sparse optical flow and their applications.

Type	Output / mechanism	Uses / conditions
Dense flow	Per-pixel displacement. Horn–Schunck: global smoothness. Farnebäck: local polynomial approximation.	Scene-wide motion and dynamic regions. Weak texture/occlusion need priors; cost grows with image size.
Sparse flow	Selected feature displacements. Lucas–Kanade estimates local motion, often with image pyramids.	Tracking, odometry, real-time control. Requires stable features; no direct output in unsampled regions.

Compare coverage, output, assumptions, computation, and application.

8.9.1 Concepts and Methods: Q2 · Metric Velocity

Convert Pixel Motion to Metric Motion

Example: a static scene; pure lateral camera translation, no rotation. Image motion: 3 px/frame to the right at 30 Hz. Axial depth: 5 m. Pixel focal length: 320 px.

$$\dot{u} = \frac{\Delta u}{\Delta t} = 3 \times 30 = 90\ \text{px/s}$$

$$\dot{u} = -\frac{f_x}{Z} v_{c,x} \quad \Rightarrow \quad v_{c,x} = -\frac{5}{320} \times 90 = -1.40625\ \text{m/s}$$

Here $v_{c,x}$ is camera translation, positive to the right; Z is axial depth. Rightward image motion means leftward camera motion.

Without depth, report pixels/s. Account separately for camera rotation and target motion.

8.9.1 Concepts and Methods: Q3 · Map Semantics

Occupancy, Surfaces, and Distances Answer Different Queries

Textbook Sec. 8.4 · Q3: Compare ESDF, TSDF, and occupancy grids for obstacle avoidance.

Map	Stored quantity / query	Role in avoidance
Occupancy grid	Cell occupancy state/probability; unknown cells are separate.	Collision/grid search; distance transform for nearest obstacles.
TSDF Surface	Truncated signed distance function: fuse near-surface data, often along rays.	Reconstruct surfaces; fuse noise. Truncated values are not full clearance.
ESDF Euclidean	Euclidean signed distance field: nearest-boundary distance; free space is positive here.	Distance/gradients support clearance checks and continuous trajectory optimization.

Unknown space is blocked here; exploration needs sensing-range and braking limits.

8.9.1 Concepts and Methods: Q3 · Collision Clearance

Vehicle Radius Changes the Clearance Test

Example: approximate the vehicle by a bounding sphere of radius 0.25 m; add a 0.15 m error margin. The ESDF is positive in free space; the center is 0.35 m from the obstacle boundary.

$$\rho = r_{\text{robot}} + m = 0.25 + 0.15 = 0.40\ \text{m}$$

Inside obstacle	Insufficient clearance	Center clearance met

0.35 m < 0.40 m: short by 0.05 m. At 0.50 m, clearance exceeds the threshold by 0.10 m.

Include map discretization, localization error, and tracking error in the safety radius.

8.9.1 Concepts and Methods: Q4 · Visual Servoing

Which Error Does Each Method Control?

Textbook Sec. 8.4 · Q4: Compare position-based (PBVS), image-based (IBVS), and end-to-end visual servoing and their uses.

Method	Feedback / properties	Uses / key conditions
PBVS	Vision estimates relative pose; control reduces 3D pose error.	Docking/guidance/positioning: known geometry; accurate calibration and pose.
IBVS	Feature error + image Jacobian → commands; shorter geometric chain.	Alignment/tracking/approach: visible target; valid depth/Jacobian approximations.
End-to-end	Learned policy: images/tasks → actions or motion references.	Training/task coverage; data distribution/compute; independent closed-loop tests.

All can use planning/safeguards; distinguish the quantity used for control.

8.9.1 Concepts and Methods: Q4 · Control Paths

Three Paths from Vision to Control Commands

PBVS	Images Relative pose	Pose error Control law	Velocity command Autopilot execution
IBVS	Images Feature error	Image Jacobian Control law	Velocity command Autopilot execution
End-to-end	Images Task conditions	Learned policy Action output	Limits / interface Autopilot execution

Motion produces new images for feedback. Add self-localization, maps, or global planning to PBVS when the task needs them.

8.9.1 Concepts and Methods: Q4 · One Control Step

How One-Axis Image Error Decreases

Example: a static target; lateral camera translation only; fixed attitude; axial depth 5 m. The target is 32 px right of the desired position; focal length 320 px; gain 1/s; time step 0.10 s.

$$e_x = \frac{u - u^*}{f_x} = \frac{32}{320} = 0.10, \qquad L_x = -\frac{1}{Z} = -0.20\ \mathrm{m}^{-1}$$

$$v_{c,x} = -\lambda L_x^{-1} e_x = -1 \times (-5) \times 0.10 = 0.50\ \mathrm{m/s}$$

$$e_{x,k+1} \approx e_{x,k} + L_x v_{c,x} \Delta t = 0.10 - 0.20 \times 0.50 \times 0.10 = 0.09$$

Rightward camera motion moves the projection left. Reobserve; check command limits and visibility.

8.9.1 Concepts and Methods: Q5 · Vision and LiDAR

How Vision and LiDAR Constrain Pose and Maps

Textbook Sec. 8.4 · Q5: Compare visual, LiDAR, and visual/LiDAR-inertial SLAM and their uses.

Type	Measurements / properties	Uses / limitations
Visual SLAM	Feature/pixel matches; texture/color, light hardware. Monocular metric scale needs extra constraints.	Light platforms with texture/stable exposure. Weak texture, blur, and lighting changes weaken constraints.
LiDAR SLAM	Ranging point clouds and scan matching provide metric geometry, with less dependence on visible light.	Rich-geometry navigation/mapping. Corridors, repeated planes, reflections, occlusion cause degeneracy.

SLAM = simultaneous localization and mapping; LiDAR = light detection and ranging. Compare scale, degeneracy, and task needs.

8.9.1 Concepts and Methods: Q5 · Inertial Fusion

Use Fast Inertial Updates Between Slower Observations

Combination	Added constraints / properties	Conditions / limitations
Visual-inertial odometry (VIO)	Fast IMU propagation; vision corrects pose/bias. Sufficient excitation can recover monocular metric scale.	Fast maneuvers/brief visual loss. Calibrate timing/extrinsics; long-term drift needs added constraints.
LiDAR-inertial odometry (LIO)	IMU propagation; compensate scan motion. Point-cloud matching corrects inertial drift.	Low-light mapping with rich geometry. Degeneracy, timing error, and IMU bias still matter.
Vision / LiDAR / inertial fusion	Combine complementary texture, ranging geometry, and inertial-motion constraints.	Complex dynamic tasks: joint models avoid double-counting correlated data.

IMU: inertial measurement unit; GNSS: global navigation satellite system. Full SLAM needs globally consistent maps; GNSS can constrain outdoor absolute position.

8.9.1 Concepts and Methods: Q5 · Selection Criteria

Let Scene Conditions Guide the Fusion Design

Task conditions	Suggested choice	Observable checks
Stable texture **Limited payload**	Visual-inertial; add stereo if needed for scale	Reprojection residuals, scale, and recovery after brief frame loss
Low light **Rich geometry**	LiDAR-inertial; compensate scan motion at each point's time	Scan-matching residuals, motion distortion, and bias stability
Repeated corridors **Little texture**	Add distinguishable observations or absolute-position constraints	Degenerate directions, uncertainty, and map consistency after loop closure

Choose complementary constraints; check scale, space/time calibration, and localization error.

Practice and Discussion

Choose a Method Under the Same Task Conditions

Scenario / conditions	Answer / reasoning
Need a scene-wide motion field; sufficient texture	Use dense flow for per-pixel output; still check occlusion, weak texture, priors, and errors
Static target; fixed attitude; lateral translation Z = 5 m; 32 px right; focal length 320 px; gain 1/s	Normalized error 0.10; interaction entry −0.20/m. IBVS moves the camera right at 0.50 m/s
Low light; rich geometry; no loop closure configured	LiDAR-inertial constraints are an option. Odometry alone does not ensure full SLAM global consistency

8.9.1 Concepts and Methods: Unit Summary

Measurements and Task Constraints Guide Method Choice

Feedback: distinguish measurements, states, references, and actions. New observations correct subsequent actions as the environment changes.

Optical flow: dense and sparse differ in coverage. Converting pixel motion to metric motion also needs depth and a motion model.

Maps: occupancy, surfaces, and distances support different queries. Clearance thresholds include vehicle radius and error margins.

Servoing / localization: choose errors and available constraints; check calibration, visibility, scale, degeneracy, and global consistency.

Chapter Outline

Exercises and Classroom Discussion

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.9.1 Concepts and Methods
Textbook: Sec. 8.4, Questions 1–5

8.9.2 Geometric Calculations
Textbook: Sec. 8.4, Question 6

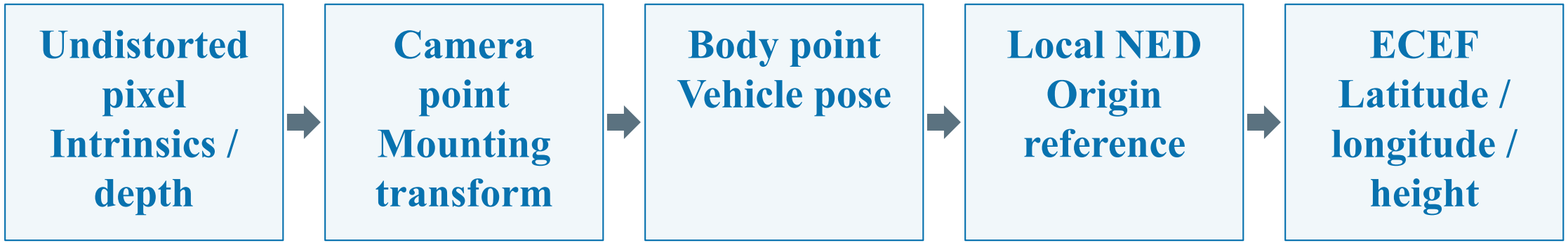
8.9.3 Perception and Decision-Making Design
Textbook: Sec. 8.4, Questions 7 and 9

8.9.4 System Development and Transfer
Textbook: Sec. 8.4, Questions 8 and 10

8.9.2 Geometric Calculations: Q6 · Complete Chain

What Is Needed to Convert Pixels to Latitude, Longitude, and Height?

Textbook Sec. 8.4 · Q6: How is the complete pixel-to-geographic-coordinate chain implemented?



Scale: a pixel gives a ray; depth, slant range, stereo, or a known plane sets scale.

Agree on transform direction, axes, observation-time pose, local origin, and height datum.

NED = north−east−down (down is positive); ECEF = Earth-centered, Earth-fixed. See Sec. 8.1.5, Eqs. 8.7–8.12.

8.9.2 Geometric Calculations: Q6 · Back-Projection

Recover a Camera Point from Axial Depth

Example: ideal undistorted 640 × 480 camera; horizontal field of view 90°; square pixels; principal point (320, 240) px. Target pixel: (384, 272) px. Axial depth: 5 m.

$$\mathbf{K} = \begin{bmatrix} 320 & 0 & 320 \\ 0 & 320 & 240 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{q} = \mathbf{K}^{-1} \begin{bmatrix} 384 \\ 272 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.1 \\ 1 \end{bmatrix}$$

$$^{c}\mathbf{p} = Z\mathbf{q} = 5 \begin{bmatrix} 0.2 \\ 0.1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.5 \\ 5.0 \end{bmatrix} \text{ m} \qquad \text{(Eq. 8.7)}$$

Camera point: 1.0 m right, 0.5 m down, 5.0 m forward. The measurement is axial depth.

8.9.2 Geometric Calculations: Q6 · Mounting and Pose

Apply a Nonzero Offset and Yaw to the Target

Camera right/down/forward → body forward/right/down (FRD); physical directions align.
Body-frame offset (0.2, 0, 0.1) m; roll = pitch = 0.
Yaw 90°; vehicle NED position (10, 20, −10) m.

$$^{b}\mathbf{p} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0.5 \\ 5 \end{bmatrix} + \begin{bmatrix} 0.2 \\ 0 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 5.2 \\ 1 \\ 0.6 \end{bmatrix} \text{m}$$

$$^{e}\mathbf{p} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5.2 \\ 1 \\ 0.6 \end{bmatrix} + \begin{bmatrix} 10 \\ 20 \\ -10 \end{bmatrix} = \begin{bmatrix} 9 \\ 25.2 \\ -9.4 \end{bmatrix} \text{m}$$

Target: 9 m north, 25.2 m east, 9.4 m above the local origin. Compare Eqs. 8.8 and 8.10.

8.9.2 Geometric Calculations: Q6 · Geodetic Coordinates

Convert Local NED through ECEF to Geodetic Coordinates

WGS 84 origin: latitude 31°, longitude 121°, ellipsoidal height 100 m. ECEF uses Earth's center of mass as origin and meters as units.

$$\mathbf{R}_e^f = \begin{bmatrix} -\sin\phi_0\cos\lambda_0 & -\sin\lambda_0 & -\cos\phi_0\cos\lambda_0 \\ -\sin\phi_0\sin\lambda_0 & \cos\lambda_0 & -\cos\phi_0\sin\lambda_0 \\ \cos\phi_0 & 0 & -\sin\phi_0 \end{bmatrix}$$

$$\mathbf{r}_f = \mathbf{r}_{f,0} + \mathbf{R}_e^f \ {}^e\mathbf{p}, \qquad (\phi, \lambda, h) = ECEF2LLA(\mathbf{r}_f)$$

ϕ_0, λ_0 are the origin latitude and longitude; $\mathbf{r}_{f,0}$ is its ECEF position. h is target ellipsoidal height. Convert angles to radians for trigonometric functions.

8.9.2 Geometric Calculations: Q6 · End-to-End Check

Check the Numerical Result by Reverse Projection

Output	Full-chain result	Interpretation
Latitude / longitude	31.000081175° 121.000263858°	Mainly 25.2 m east and 9.0 m north of the origin
Ellipsoidal height	109.400056 m	Negative down coordinate: about 9.4 m above the origin
Reverse projection	(384.000, 272.000) px	Inverse transforms and camera intrinsics recover the original pixel

Do not add meters to degrees. Check camera/body axes, the NED height sign, observation time, and height datum.

Practice and Discussion

Does the 5 m Measurement Mean Slant Range or Axial Depth?

Keep K and pixel (384, 272) px, but now measure a 5 m slant range. Find the camera point. Can 25.2 m east be added directly to the origin's longitude of 121°?

$$r = 5\,\mathrm{m} \quad \Rightarrow \quad {}^{c}\mathbf{p} = 5\frac{[0.2,\ 0.1,\ 1]^{\top}}{\sqrt{1.05}} \approx [0.976,\ 0.488,\ 4.880]^{\top}\,\mathrm{m}$$

The axial depth is now about 4.880 m. A 5 m slant range and a 5 m axial depth give different points; use the correct point in subsequent transformations.

Convert 25.2 m east through NED → ECEF before recovering longitude. Latitude/longitude are angles, so meters cannot be added directly.

8.9.2 Geometric Calculations: Unit Summary

Scale, Transformation Direction, and Reference Define Coordinates

Ray / scale: undistort and back-project pixels; axial depth, slant range, or geometric constraints supply 3D scale.

Transformation chain: camera → body → local NED → ECEF → latitude/longitude/height. Agree on rotation direction, translation frame, and observation time.

Height datum: NED is positive downward. Ellipsoidal height and elevation above sea level use different references; check GNSS/map heights.

Reverse check: asymmetric attitudes and nonzero offsets test axes. Inverse transforms and projection should recover the original pixel.

Chapter Outline

Exercises and Classroom Discussion

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.9.1 Concepts and Methods
Textbook: Sec. 8.4, Questions 1–5

8.9.2 Geometric Calculations
Textbook: Sec. 8.4, Question 6

8.9.3 Perception and Decision-Making Design
Textbook: Sec. 8.4, Questions 7 and 9

8.9.4 System Development and Transfer
Textbook: Sec. 8.4, Questions 8 and 10

8.9.3 Perception and Decision-Making Design: Q7 · Rates

Sampling Periods and Roles of Four Sensors

Textbook Sec. 8.4 · Q7: Design time synchronization and fusion in the NeighborhoodPark scene: camera 30 Hz, LiDAR 10 Hz, IMU 200 Hz, and GNSS 5 Hz.

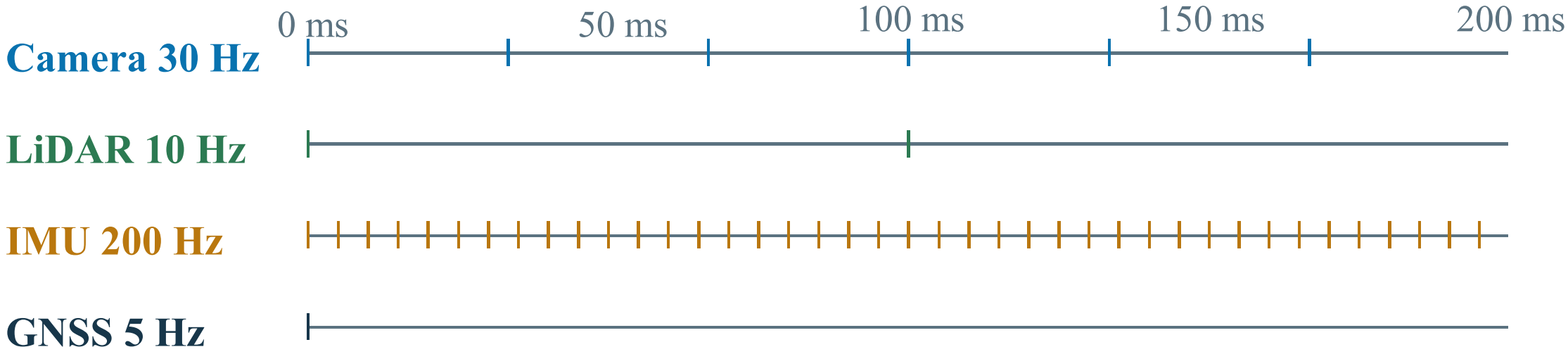
Sensor	Period	Samples / 200 ms	Fusion role
Camera, 30 Hz	33.33 ms	6 images	Visual motion, texture, targets
LiDAR, 10 Hz	100 ms	2 point clouds	Range geometry; scan matching
IMU, 200 Hz	5 ms	40 samples	Angular velocity/specific force; fast propagation
GNSS, 5 Hz	200 ms	1 sample	Outdoor absolute position

Counts assume all streams start at t = 0; window [0, 200) ms; no dropped data.

8.9.3 Perception and Decision-Making Design: Q7 · Timing

Keep Four Sampling Rhythms on One Time Base

Acquisition time identifies the measured state. Reception minus acquisition time gives delay. Keep one simulation clock and original acquisition timestamps.

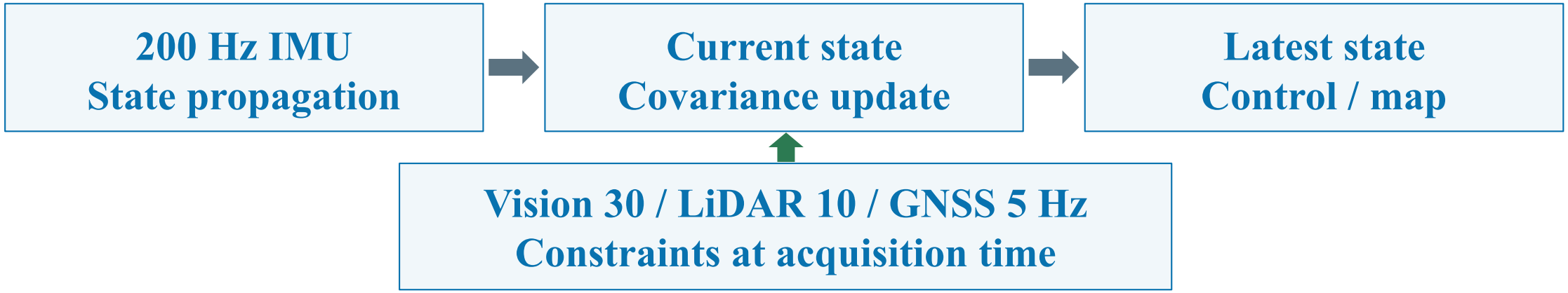


Use hardware-triggered acquisition where available. In software, buffer by acquisition time, interpolate or propagate, and deskew each point in a scan.

8.9.3 Perception and Decision-Making Design: Q7 · Fusion

Inertial Propagation with Asynchronous Observation Updates

Suggested design: an error-state filter or sliding window. Estimate position, velocity, attitude, and IMU bias; output a common frame, timestamp, and covariance.



$\mathbf{r}_i = \mathbf{z}_i - h_i(\hat{\mathbf{x}}(t_i))$, Delayed data: update past state; repropagate to now.

8.9.3 Perception and Decision-Making Design: Q7 · Gating

How Much Position Error Comes from a 20 ms Offset?

Example: straight-line speed 3 m/s; sensor timestamps differ by 20 ms. A 10 Hz scan spanning 100 ms also needs compensation for motion within the scan.

$$\Delta p \approx v\Delta t = 3 \times 0.020 = 0.060\ \mathrm{m}$$

Motion during the scan can reach about 0.30 m. Deskew points using their acquisition times and the IMU/state trajectory.

$$\frac{|r|}{\sigma_r} = \frac{0.40}{0.10} = 4 > 3$$

Example gate: residual 0.40 m; innovation standard deviation 0.10 m; threshold 3 standard deviations. Reject or downweight the observation and retain the reason.

8.9.3 Perception and Decision-Making Design: Q7 · Logs

Log Fusion Outputs and Degradation Criteria

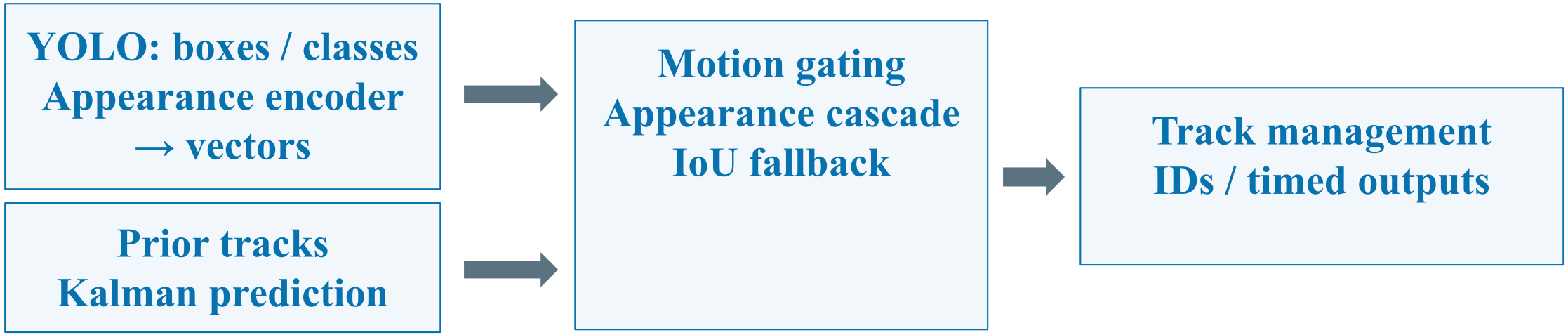
Log / event	Required output	Check
Spatial / temporal setup	Rates, acquisition/reception times, intrinsics/extrinsics, origin	Timestamps follow sensor rates; axes and units agree
Fusion output	Position, velocity, attitude, covariance, valid sources, age	New data update estimates; uncertainty reflects degradation
Loss / recovery	Log GNSS occlusion, black frames, and point-cloud loss	Suspend invalid constraints; explain timeouts and recovery
Independent evaluation	Aligned estimate/ground-truth errors, delay, task outcome	Separate ground truth; use the same error reference

VisionCaptureApi receives timestamps and IMU data; verify rates/time base against acquisition logs.

8.9.3 Perception and Decision-Making Design: Q9 · Integration

Turn Detection Boxes into Persistent Identities

Textbook Sec. 8.4 · Q9: Design integrated object detection and multi-object tracking using YOLO + Deep SORT, or CenterTrack.



Detections also carry confidence. Match the appearance model to the target class. IoU = intersection over union; it supports remaining associations.

8.9.3 Perception and Decision-Making Design: Q9 · Association

Combine Motion Gating with Appearance Costs

Gate candidates using predicted covariance before comparing appearance. This design may also gate on class.

$$d^2_{\text{motion}} = (\mathbf{z} - \hat{\mathbf{z}})^\top \mathbf{S}^{-1} (\mathbf{z} - \hat{\mathbf{z}}), \qquad d_{\text{app}} = 1 - \mathbf{r}_i^\top \mathbf{r}_j$$

$\mathbf{S}$ is innovation covariance; $\mathbf{r}_i, \mathbf{r}_j$ are unit appearance vectors. Rows below are tracks and columns are detections; all candidates pass motion gating.

$$\mathbf{C} = \begin{bmatrix} 0.10 & 0.70 \\ 0.80 & 0.15 \end{bmatrix}, \qquad J_{\text{diag}} = 0.25 < J_{\text{cross}} = 1.50$$

Choose diagonal pairs. Reject pairs that fail motion gating, even when appearance is similar.

8.9.3 Perception and Decision-Making Design: Q9 · Track Life

Maintain or End an Identity When Observations Are Lost

Example: confirm after 3 consecutive matches; retain an unmatched confirmed track for 0.50 s. Use acquisition times; distinguish predictions from observations.

Status	Update rule	Output rule
Tentative	Accumulate consecutive matches; delete after an early miss	Unconfirmed tracks cannot drive closed-loop control
Confirmed	New detections correct predictions and update appearance	ID, class, box, and observation time
Temporarily lost	Predict only; check elapsed time from timestamps	ID, prediction flag, and observation age
Deleted	Delete after 0.50 s; stop normal track output	A returning object needs a new track or reidentification

8.9.3 Perception and Decision-Making Design: Q9 · CenterTrack

Predict Centers and Associations from Consecutive Frames

Alternative: CenterTrack takes current and previous images plus a previous-detection heatmap. It predicts current centers, boxes, and backward displacements, then updates identities.

$$\tilde{\mathbf{c}}_{t-1} = \mathbf{c}_t + \Delta\mathbf{c} = \begin{bmatrix} 130 \\ 75 \end{bmatrix} + \begin{bmatrix} -8 \\ 2 \end{bmatrix} = \begin{bmatrix} 122 \\ 77 \end{bmatrix} \text{ px}$$

Previous track	Previous center / distance	Association
ID17	(121, 77) px; distance 1 px	Inside the 5 px gate: associate ID17
Another ID	(141, 77) px; distance 19 px	Outside the gate: reject

CenterTrack jointly detects/tracks. State frame intervals and long-occlusion handling.

8.9.3 Perception and Decision-Making Design: Q9 · Platform

Design Outputs and Task-Specific Training Data

Stage	Implementation	Check
Acquire / preprocess	New frame/time; convert BGR if needed; resize and pad	No duplicate update; map boxes back to the original image
Detection → tracking	Class, box, confidence, appearance vector, frame interval	Reject invalid boxes; use the actual motion-model interval
Training / output	Task-specific data splits, detection/appearance models, IDs	Test crossings, occlusion, reappearance; show predictions/age
Geometry / control	Add depth, extrinsics, estimated pose for metric targets	2D centers stay in pixels; degrade the task after target loss

VisionCaptureApi supplies images; the tracker maintains IDs. Validate model weights for this task.

8.9.3 Perception and Decision-Making Design: Q9 · Evaluation

Detection Accuracy and Identity Accuracy Answer Different Questions

Example: $N_{\mathrm{GT}} = 20$ ground-truth object-frames; 2 false negatives (FN), 1 false positive (FP), and 1 identity switch (IDSW). MOTA is multiple object tracking accuracy.

$$\mathrm{MOTA} = 1 - \frac{\mathrm{FN} + \mathrm{FP} + \mathrm{IDSW}}{N_{\mathrm{GT}}} = 1 - \frac{4}{20} = 80\%$$

Throughput example: 30 Hz input gives 33.33 ms between frames. Serial processing at 40 ms/frame supports at most 25 Hz.

$$f_{\mathrm{serial}} = \frac{1000}{40} = 25\ \mathrm{Hz} < 30\ \mathrm{Hz}$$

Log identity errors, frame age, and update rate. Test crossings, occlusion, loss, and reappearance.

Practice and Discussion

Use Observation Time and Track Status in Decisions

Scenario / conditions	Action / reasoning
Speed 3 m/s; camera timestamp is 20 ms earlier than the IMU reference	Offset is about 0.06 m. Align by acquisition time, propagate, or update history; do not change spatial rotation to compensate for time
Confirmed track has no match for 0.60 s; retention threshold 0.50 s	Threshold exceeded: stop normal track output and degrade the task. A predicted box is not a new observation
30 Hz images; serial processing takes 40 ms/frame	At most 25 Hz. Use bounded/latest-frame buffering, actual prediction intervals, and frame-age statistics

Outputs specify common coordinates, time, validity, identity, and observation age.

8.9.3 Perception and Decision-Making Design: Unit Summary

Time, Identity, and Uncertainty Have Distinct Meanings

Time: share an acquisition clock while keeping the 30/10/200/5 Hz rates. Reception times separately measure transport and processing delay.

Fusion: IMU propagation connects asynchronous observations; late data update history. Retain covariance and valid sources in the common state.

Identity: motion gating, appearance, or backward-center prediction supports association. Define outputs for confirmed, predicted, lost, and deleted tracks.

Validation: report estimation errors, identity errors, and frame age separately. Define observable checks for occlusion, stream loss, crossings, and reappearance.

Chapter Outline

Exercises and Classroom Discussion

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.9.1 Concepts and Methods
Textbook: Sec. 8.4, Questions 1–5

8.9.2 Geometric Calculations
Textbook: Sec. 8.4, Question 6

8.9.3 Perception and Decision-Making Design
Textbook: Sec. 8.4, Questions 7 and 9

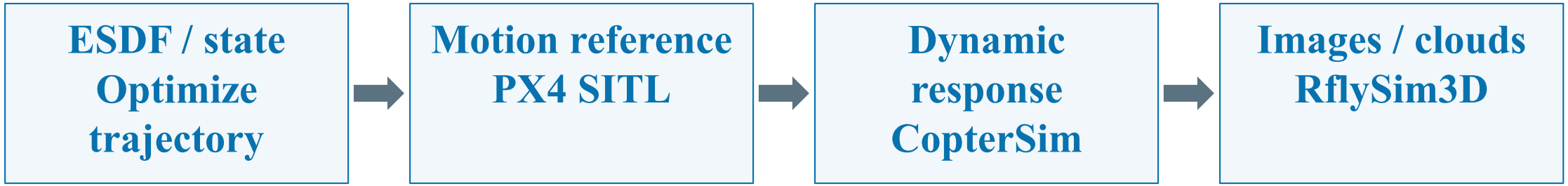
8.9.4 System Development and Transfer
Textbook: Sec. 8.4, Questions 8 and 10

8.9.4 System Development and Transfer: Q8 · Objective

Connect ESDF Planning to a Complete SITL Loop

Textbook Sec. 8.4 · Q8: Develop optimization-based obstacle-avoidance trajectory planning using an ESDF, then implement SITL simulation with the RflySim toolchain.

Software-in-the-loop (SITL; SIL in the textbook) connects PX4 software to simulated vehicles, sensors, and environments. Planning turns states/goals into timed references.



New measurements update maps/states for replanning. Log ground truth separately to evaluate clearance and tracking errors.

8.9.4 System Development and Transfer: Q8 · Model

Smoothness and Obstacle Costs for a Cubic B-Spline

Suggested model: a 3D trajectory through an observed static map. Optimize cubic B-spline control points and time allocation. The ESDF is positive in free space.

$$\mathbf{p}(t) = \sum_i N_{i,3}(t)\mathbf{q}_i\,, \qquad 0 \le t \le T$$

$$J = w_s \int_0^T \|\ddot{\mathbf{p}}(t)\|^2\, dt + w_c \sum_j [\max(0, \rho - d(\mathbf{p}(t_j)))]^2 \Delta t + w_T T$$

$\mathbf{q}_i$: control points (m); $N_{i,3}$: cubic basis functions; T: duration (s); d: ESDF distance (m); ρ: safety radius (m).

Normalize weights; require independent hard clearance and dynamic-feasibility checks.

8.9.4 System Development and Transfer: Q8 · Constraints

Constrain Endpoints, Dynamics, and Space

Specify endpoint positions/velocities and continuous position, velocity, and acceleration. Set speed/acceleration limits from vehicle capabilities. This design treats unknown regions as blocked.

$$\mathbf{p}(0) = \mathbf{p}_0, \quad \mathbf{p}(T) = \mathbf{p}_g, \quad \|\dot{\mathbf{p}}(t)\| \le v_{\max}, \quad \|\ddot{\mathbf{p}}(t)\| \le a_{\max}$$

$$d(\mathbf{p}(t)) \ge \rho, \qquad \mathbf{p}(t) \in \Omega_{\text{observed}}$$

$$\mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} - \alpha \nabla_{\mathbf{q}} J, \qquad \alpha > 0$$

Start from a searched path. Check clearance/dynamic feasibility after optimization; replan or brake if infeasible.

8.9.4 System Development and Transfer: Q8 · Between Samples

Extend Sampled Clearance Checks to the Whole Segment

Example: an accurate static ESDF; speed bounded by 2 m/s throughout; sample interval 0.10 s; safety radius 0.40 m. The nearest sample is at most half an interval away.

$$d(\mathbf{p}(t)) \geq d(\mathbf{p}(t_j)) - v_{\max}\frac{\Delta t}{2}$$

$$d_{\text{lower}} = 0.60 - 2 \times \frac{0.10}{2} = 0.50\,\text{m} > 0.40\,\text{m}$$

$$d(t_j) = 0.45\,\text{m} \quad \Rightarrow \quad d_{\text{lower}} = 0.35\,\text{m} < 0.40\,\text{m}$$

The second case needs finer sampling or continuous checks. Include map discretization error in margins; dynamic obstacles require time-varying distances.

8.9.4 System Development and Transfer: Q8 · Replanning

Run Planning Updates and Control Sending Independently

Planning proposes candidates; control independently sends the active valid trajectory. This pseudocode still needs timestamps and timeouts.

1. state, map = latest_timestamped_inputs()
2. seed = search_initial_path(state, map, goal)
3. candidate = optimize_esdf_spline(seed, deadline)
4. if validate_whole_trajectory(candidate): atomic_swap(candidate)
5. else: keep_valid_safe_plan_or_brake()
6. control_thread: sample_active_plan(clock); send_setpoint()

Optimization timeouts must not interrupt the command stream. Stale inputs must not drive new plans. Define stop and lost-link behavior in advance.

8.9.4 System Development and Transfer: Q8 · Map and Path

Generate the Current Run's Path with the Installed e16 Baseline

Installed example: RflySimAPIs/8.RflySimVision/2.AdvExps/ e16_ESDFPathPlan

Step	Actual entry / output	Check
1. Map setup	maps/map_cropped.yaml and its PGM image	Traceable resolution/origin; example: 0.05 m/px
2. Generate path	Open maps/WinWSL.bat; run python3 OGM2ESDF.py	Select one start and two goals on the map
3. Save result	maps/planned_path.npy	Nonempty current output; verify endpoints and minimum distance

Extend this 2D baseline: add time allocation, 3D maps, dynamic limits, and whole-trajectory validation.

8.9.4 System Development and Transfer: Q8 · SITL Integration

Connect the Path File to Autopilot and Dynamics

Step	Actual entry / signal	Check
1. Initialize	Client/VisionCapAPIDemo.bat	Wait for positioning/filter initialization before external control
2. Same result	Copy maps/planned_path.npy to Server	Use the same current file; no silent fallback points
3. Control entry	Open Server/WinWSL.bat; run python3 esdf_planner.py	Test axes/modes with an asymmetric path; keep sending SendPosNED
4. Complete the loop	Estimated-position feedback, logs, stop/lost-link handling	Use estimated state for progress; define termination behavior

Initialization log: GPS 3D fixed & EKF initialization finished

8.9.4 System Development and Transfer: Q8 · Acceptance

Check Clearance, Dynamic Limits, and Tracking Error Together

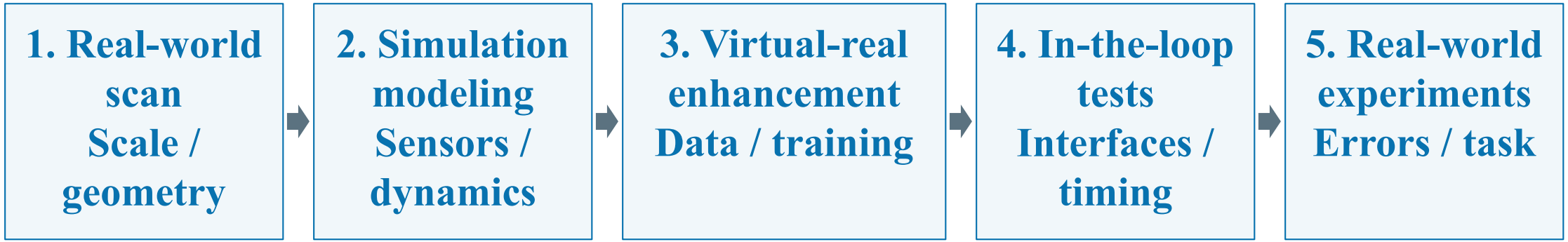
Condition	Deliverables	Accept / reject criterion
Reachable goal	Reference/actual trajectories, minimum clearance, tracking error	Reach goal; satisfy whole-trajectory clearance/tracking margins
Dynamic limits	Speed, acceleration, control limits, replanning time	Respect vehicle limits; keep trajectory switches continuous
Too narrow / no path	Infeasibility reason, rejected candidates, braking trajectory	Reject candidates that violate radius + margin
Lost / stale input	Input age, state uncertainty, lost-link action	Degrade on timeout; never treat stale maps as fresh indefinitely

Deliver model/interface/configuration logs. Keep reference, estimated actual trajectories, and evaluation ground truth separate.

8.9.4 System Development and Transfer: Q10 · Transfer

Five Textbook Stages with Feedback for Transfer

Textbook Sec. 8.4 · Q10: Design a simulation-to-real (Sim2Real) transfer process for multimodal perception and intelligent decision-making using RflySim.



Keep each stage's outputs: scene/calibration, simulation settings, model/data versions, in-the-loop logs, and real-world task records.

Feed failed conditions back to geometry, sensor models, data coverage, synchronization, or control implementation, then retest.

8.9.4 System Development and Transfer: Q10 · Setups

Compare Virtual and Physical Setups for the Same Ring Task

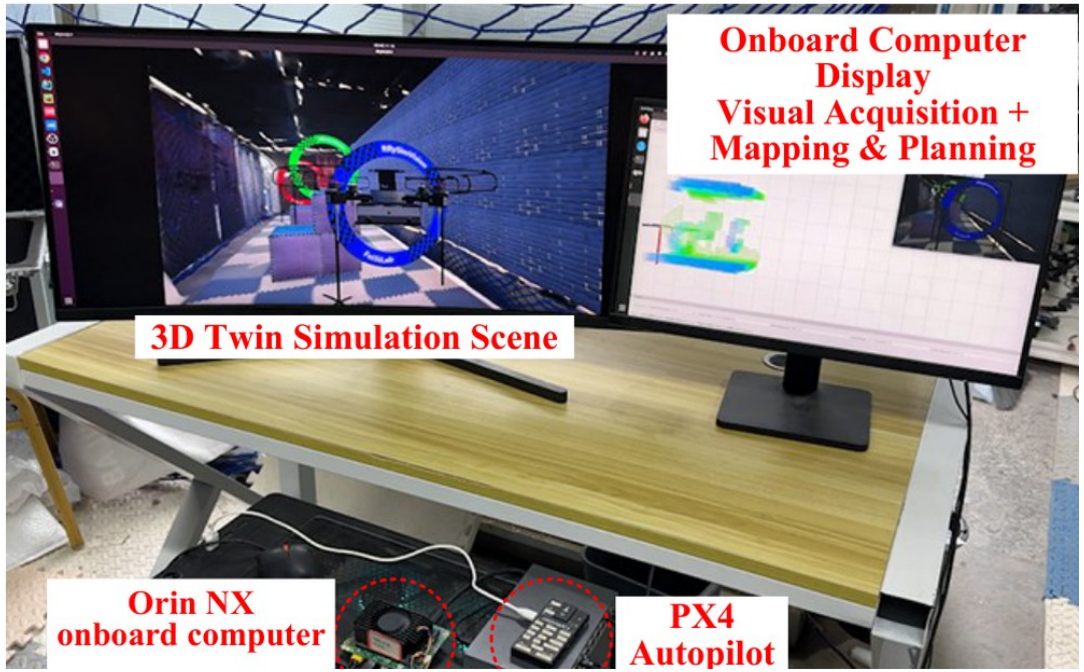


Fig. 8.11(a) Hardware-in-the-loop twin (HITL; HIL in the textbook)

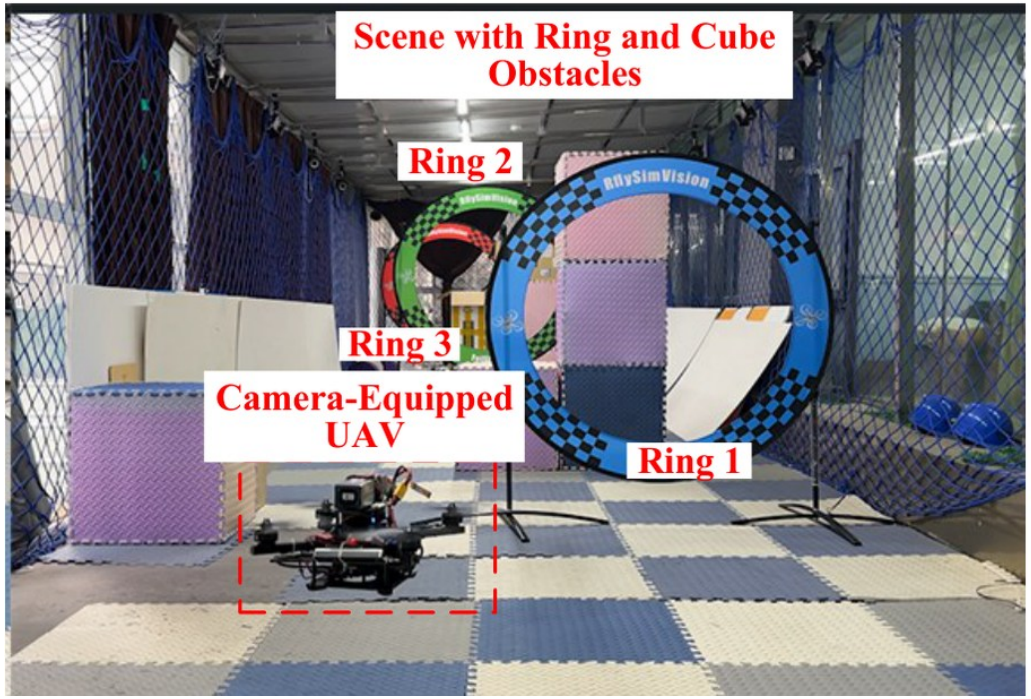


Fig. 8.11(b) Real-World Experiment

Match interfaces; compare errors, clearance, and delay under matched conditions.

8.9.4 System Development and Transfer: Q10 · Scan and Model

Match Sensors, Geometry, and Dynamics

Match	Stages 1–2: concrete work	Comparison
Scene / reference	Scanned geometry, scale, obstacles, common origin	Known lengths, asymmetric positions, consistent coordinates
Vision / point clouds	Camera intrinsics/extrinsics; scan model, sampling, delay	Projection, depth, point-cloud geometry at the same viewpoint
Inertial / noise	Bias, noise, timestamps, and observation-loss cases	Noise statistics/time alignment for static and moving data
Vehicle / control	Mass, inertia, thrust response, and control bandwidth	Tracking error/dynamic response under the same reference

Use real calibration/identification; cover task conditions and isolate each virtual–real difference.

8.9.4 System Development and Transfer: Q10 · Data

Data Coverage and the Independent Test Gap

Data strategy: randomize task-relevant light, materials, noise, and delay; use consistent labels and split training/test data by scene or batch.

Independent test: hold out real-world data not used for tuning. Compare the same task, metric, and conditions; prevent adjacent-frame leakage.

Illustrative: 80/100 simulated tasks and 70/100 real-world tasks succeed.

$$S_{\text{sim}} = 80\%, \quad S_{\text{real}} = 70\%, \quad \Delta S = 10 \ \text{ percentage points}$$

Group the gap by lighting, occlusion, speed, and target class. Return failures to the corresponding development stage.

8.9.4 System Development and Transfer: Q10 · In-the-Loop

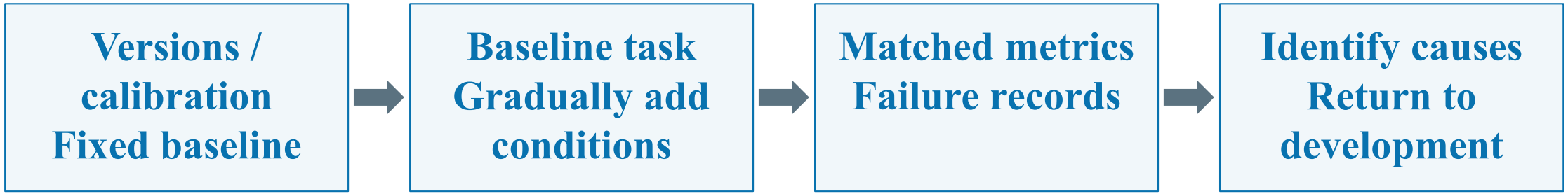
Move Validation Toward the Onboard System

Validation level	Connected components	Acceptance focus
Software loop (SITL)	Algorithm + software PX4; simulated vehicle/environment	Data semantics, feedback logic, frames, failure cases
Onboard compute	Algorithm on target onboard computer; replayed inputs allowed	Inference/fusion/planning time, frame age, resource use
Hardware loop (HITL)	Real autopilot with simulated dynamics and sensors	Transport, mode switches, continuous sending, lost-link action

Use simulation time for alignment; wall clock for performance. Log mean/tail latency and maximum input age. Textbook SIL/HIL = SITL/HITL.

8.9.4 System Development and Transfer: Q10 · Physical Tests

Close the Loop from Real-World Tasks Back to Development



Align detections/IDs, pose/uncertainty, reference/actual trajectories, clearance, processing delay, and takeover causes by time.

Compare the same metrics by condition. Route held-out real-task failures to modeling, data, calibration, or control.

Deliver fixed versions/configurations, calibration, in-the-loop logs, task statistics, and failure-feedback records as transfer evidence.

Practice and Discussion

Check an Integrated Design Item by Item

Discussion scenario	Key answer
30 Hz input; tracking reaches only 25 Hz	Bounded/latest-frame buffering; predict with actual intervals; record frame age
Two sensors are offset by 20 ms	Align or propagate by acquisition time; at 3 m/s this means 0.06 m of position error
ESDF distance 0.35 m; safety threshold 0.40 m	Short by 0.05 m: reject, increase clearance, or replan
Only a pixel box, but a geographic target position is requested	Add depth/geometry, intrinsics/extrinsics, observation-time pose, and a common origin

Self-check: conditions, units, frames/times, model limits, results, and failure criteria.

8.9.4 System Development and Transfer: Unit Summary

System Development Needs Reproducible Validation Records

Model: trajectories include time. Soft costs guide search; endpoint, clearance, and dynamic constraints need independent hard checks.

Integration: trace the current map/path, verify NED transforms, and run planning updates independently of continuous control sending.

Acceptance: define reachable, no-path, and stream-loss criteria. Keep planned references, actual estimated trajectories, and evaluation ground truth separate.

Transfer: scan → model → virtual-real enhancement → in-the-loop → physical task. Compare matched metrics and feed failures back to the relevant stage.

8.9 Exercises and Classroom Discussion

Chapter Summary

Perception and Decisions Need Interpretable Spatial and Temporal States

Perception: pixels, features, targets, and identities are different levels. Optical flow describes image motion; metric motion also needs depth and a motion model.

States: transformations align axes, scale, and origins; fusion handles timing, noise, and correlation and outputs states with uncertainty.

Planning / control: maps define usable space; trajectories satisfy clearance and dynamic limits; feedback corrects motion. Define actions for unknown space, stale inputs, and failures.

Common errors: pixels are not meters; positive center distance does not ensure vehicle clearance; predictions are not new observations. Ground truth evaluates; illustrative calculations teach.