Chapter Outline

Safety Decision-Making and System Improvement

Chapter 7 Health Management and Safety Assessment

7.1 Health Management Fundamentals

7.2 Unified Fault Modeling

7.3 Safety Reinforcement and Fault Data

7.4 Fault Hazards and Safety Assessment

7.5 Diagnosis and Prognostics

7.6 Safety Decisions and Improvement

7.7 Fault Injection Interfaces

7.8 Experiments and Validation

7.9 Exercises and Classroom Discussion

3 Units in This Section

7.6.1 Response and Control Tradeoffs

Textbook: Sec. 7.1.5.5

7.6.2 Hierarchical Decision-Making and Training

Textbook: Sec. 7.1.5.5

7.6.3 System Regression Testing and Improvement

Textbook: Sec. 7.1.5.6

7.6.1 Response and Control Tradeoffs: Decision Inputs

Use Diagnosis with the Mission and Remaining Capabilities

Diagnosis and state
Fault, position, battery
→
Mission and constraints
Distance, site
→
Executable actions
Wait, abort, land

Input	Inspection: Motor Effectiveness Loss	Decision Question
Fault evidence	Affected motor, effectiveness change, diagnostic confidence	Change the mission or switch control?
Current state	Height, velocity, positioning, battery, link	Is holding, moving, or descending feasible?
Mission scenario	Return path, alternate landing site, people nearby	Which feasible action has more acceptable consequences?

7.6.1 Response and Control Tradeoffs: PX4 Failsafe Actions

Warning, Hold, Return, and Land Produce Different Behaviors

PX4 v1.12.3 multicopter. Versions: Section 5.5, “Control Outputs Across Versions.”

Action	Mechanism	Preconditions and Consequences
Warning	Issue an alert; continue the current mission mode	Warning-level conditions that permit continued flight
Hold	Enter Hold mode; control the current position	Requires valid positioning and sufficient control capability
Return	Enter Return mode; follow the return configuration	Requires usable navigation and a return path/destination
Land	Enter Land mode; descend and land	Uses available estimates; the landing area affects the consequences

7.6.1 Response and Control Tradeoffs: Battery Triggers

Battery Conditions Select Failsafe Levels through Configuration

Teaching thresholds: 15%, 10%, 5%; COM_LOW_BAT_ACT=3 (critical: Return; emergency: Land).

Threshold Parameter	Battery Range in This Example	Failsafe Result
BAT_LOW_THR	12%: below 15%, still above 10%	Warning; continue the current mode
BAT_CRIT_THR	8%: below 10%, still above 5%	Return; Land if navigation is invalid
BAT_EMERGEN_THR	4%: below 5%	Emergency Land; no Hold delay

Remaining capacity (%); v1.12.3 defaults 15%/7%/5%; teaching values above.

7.6.1 Response and Control Tradeoffs: Link-Loss Timing

Loss Detection and Action Delay Are Separate Waits

v1.12.3 example: COM_RC_LOSS_T=0.5 s; COM_RCL_ACT_T=3 s; Return configured.

6.0 s
Last remote input

➡

6.5 s
Loss confirmed; Hold

➡

9.5 s
Request Return

Two intervals: 0.5 s confirms input loss; 3 s holds position while waiting for the signal to recover.

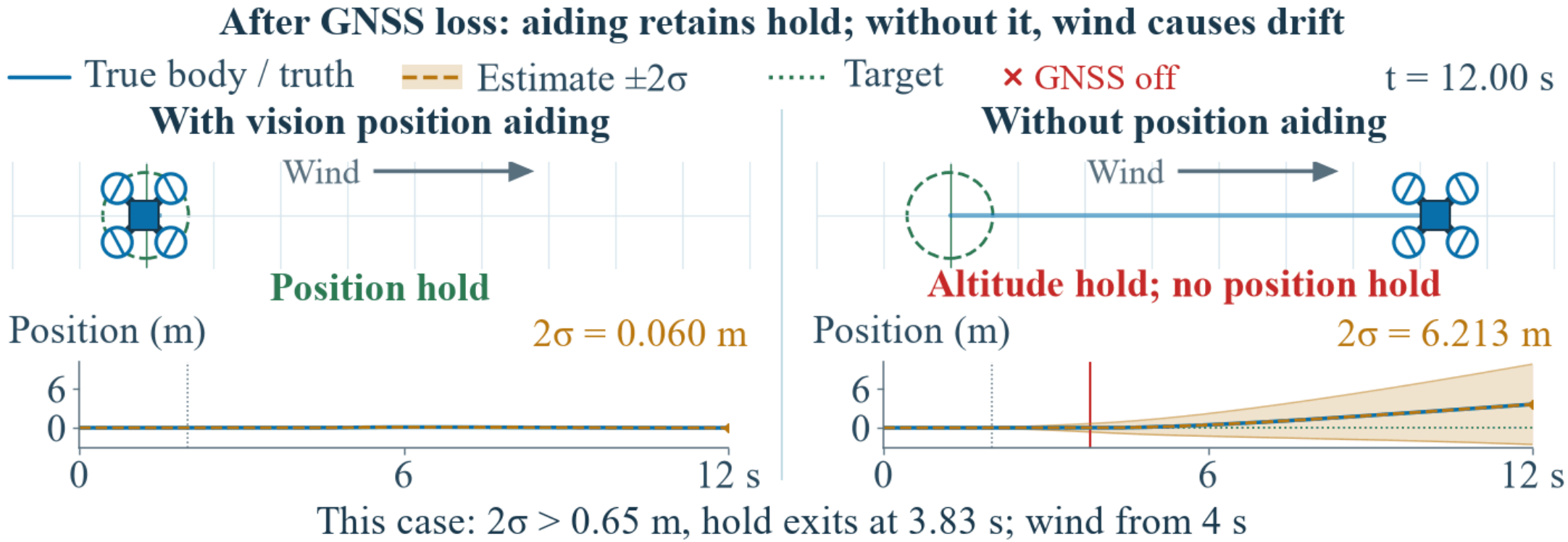
Competing actions: emergency battery landing during Hold takes priority as the more severe action.

PX4 v1.12 failsafe conditions: https://docs.px4.io/v1.12/en/config/safety.html

7.6 Safety Decisions and Improvement

7.6.1 Response and Control Tradeoffs: Remaining Positioning Capability

Can Position Still Be Held after GNSS Loss?

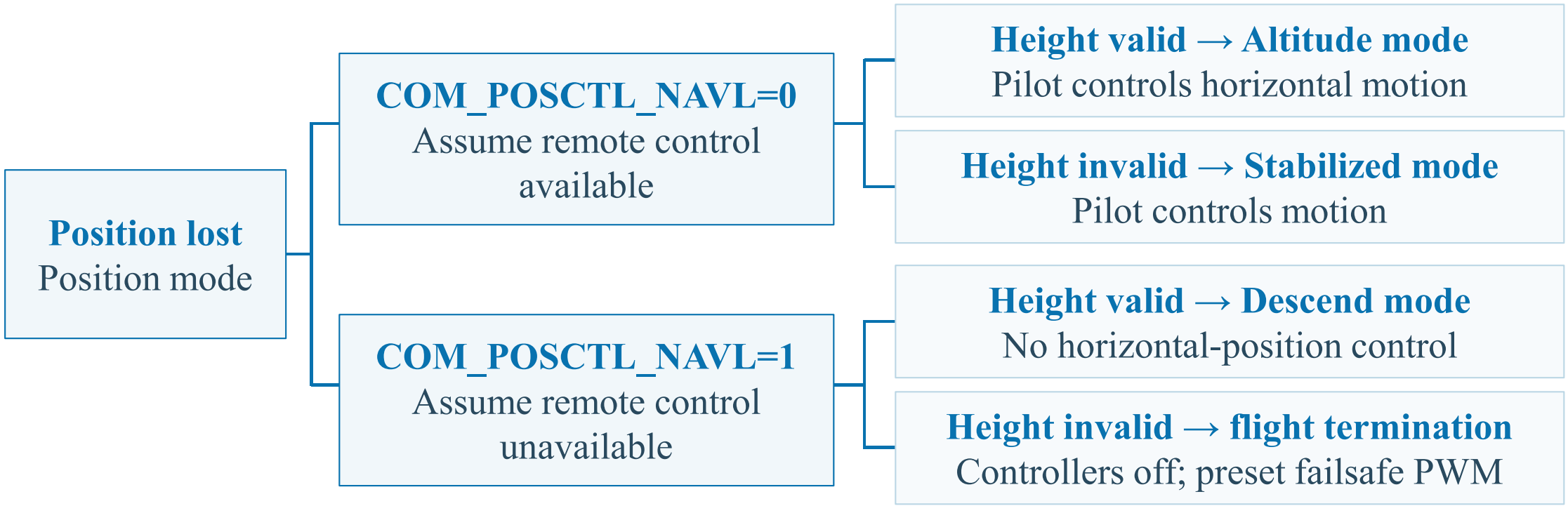


GNSS = global navigation satellite system. Same outage and crosswind; MuJoCo horizontal motion.
Right: remote control/height valid, no manual correction; position loss → Altitude mode.

7.6.1 Response and Control Tradeoffs: Mode Preconditions

Position Loss May Remove the Basis for Hold and Return

v1.12.3 Position mode: COM_POSCTL_NAVL selects fallback, then check height. Leaves are requests; verify motion. PWM = pulse-width modulation.



7.6 Safety Decisions and Improvement

7.6.1 Response and Control Tradeoffs: Decision Methods

Rules Organize Conditions; Learning Selects; Hybrids Assign Authority

Method	How It Works	Main Tradeoffs
Rules	State machine: conditional modes; behavior tree: feasible branches by priority.	Inspectable; design-limited coverage
Learning	Complex states: choose from observations/mission; update from outcomes.	Poor sample efficiency and interpretability
Hybrid	Rules limit actions; a learned policy improves performance or mission results.	Explicit constraints; align interfaces and fallback

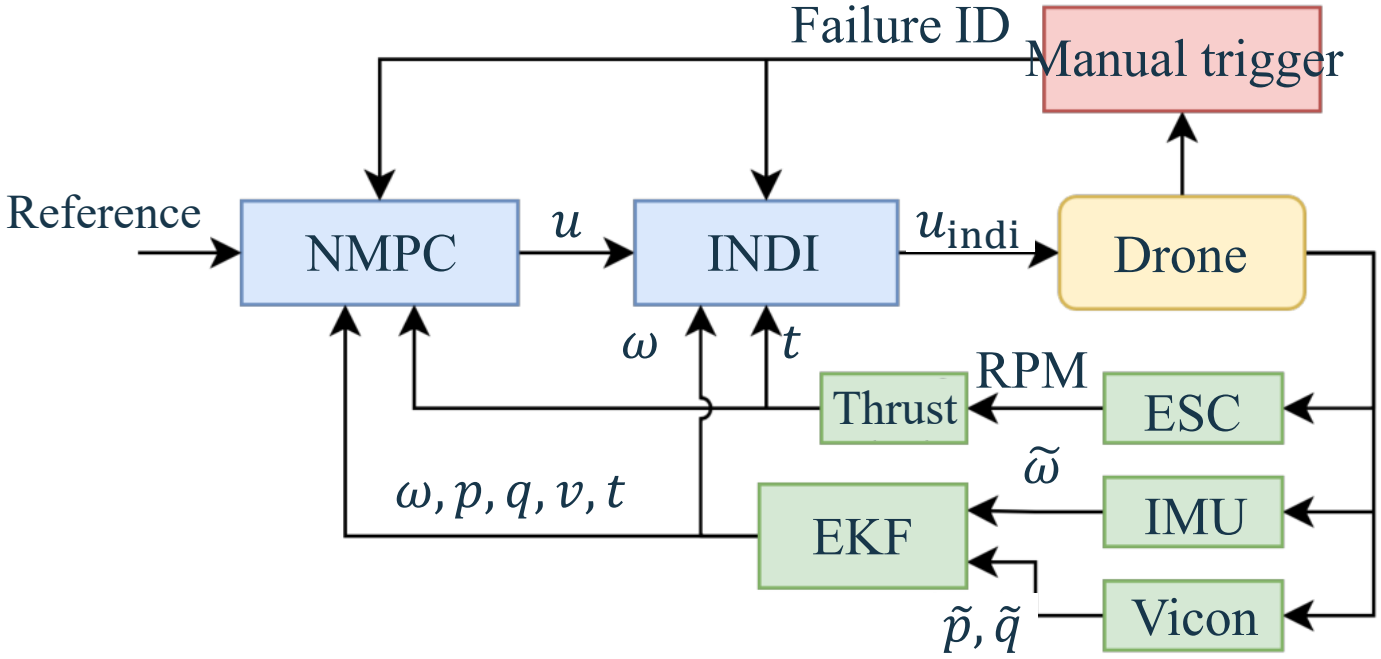
Define triggers and execution prerequisites, then optimize behavior for each scenario.

7.6 Safety Decisions and Improvement

7.6.1 Response and Control Tradeoffs: Known-Fault Tolerant Control

After One Rotor Fails, Prioritize Thrust Direction and Position

Three thrust inputs cannot control total thrust and all three moments at once.



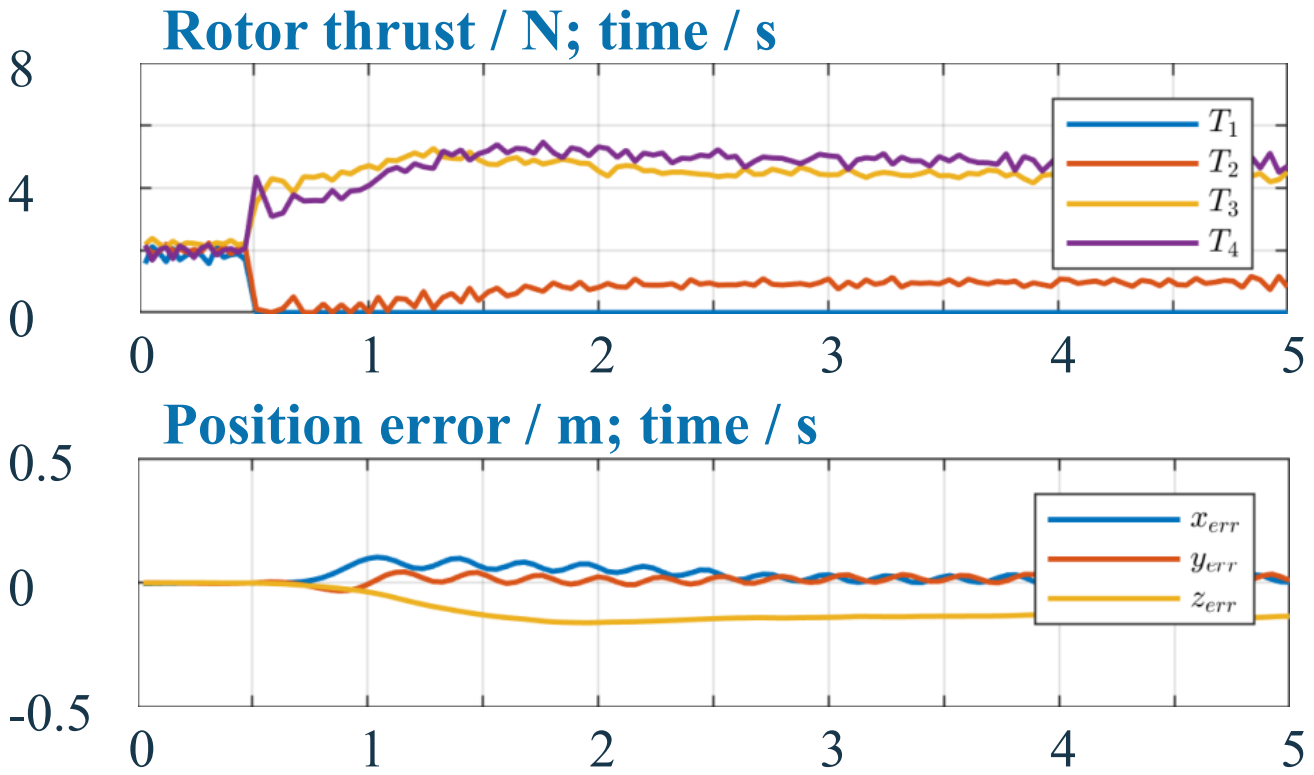
NMPC: nonlinear model predictive control
INDI: incremental nonlinear dynamic inversion
EKF: extended Kalman filter
IMU: inertial measurement unit
VICON: motion-capture system
ESC: electronic speed controller
RPM: revolutions per minute

u, u_{indi}: NMPC/INDI rotor thrust commands; t: speed-based actual thrust estimate (N).
p, v: inertial position/velocity (m, m/s); q: attitude quaternion; ω: body angular rate (rad/s).
Nan et al., IEEE RA-L, 2022, Fig. 7; DOI: https://doi.org/10.1109/LRA.2022.3154033

7.6 Safety Decisions and Improvement

7.6.1 Response and Control Tradeoffs: Experimental Results

Failed-Rotor Thrust Falls to Zero while Position Is Controlled



T_1 T_2 T_3 T_4

After 0.5 s, rotor 1 thrust is zero; the remaining rotors redistribute thrust.

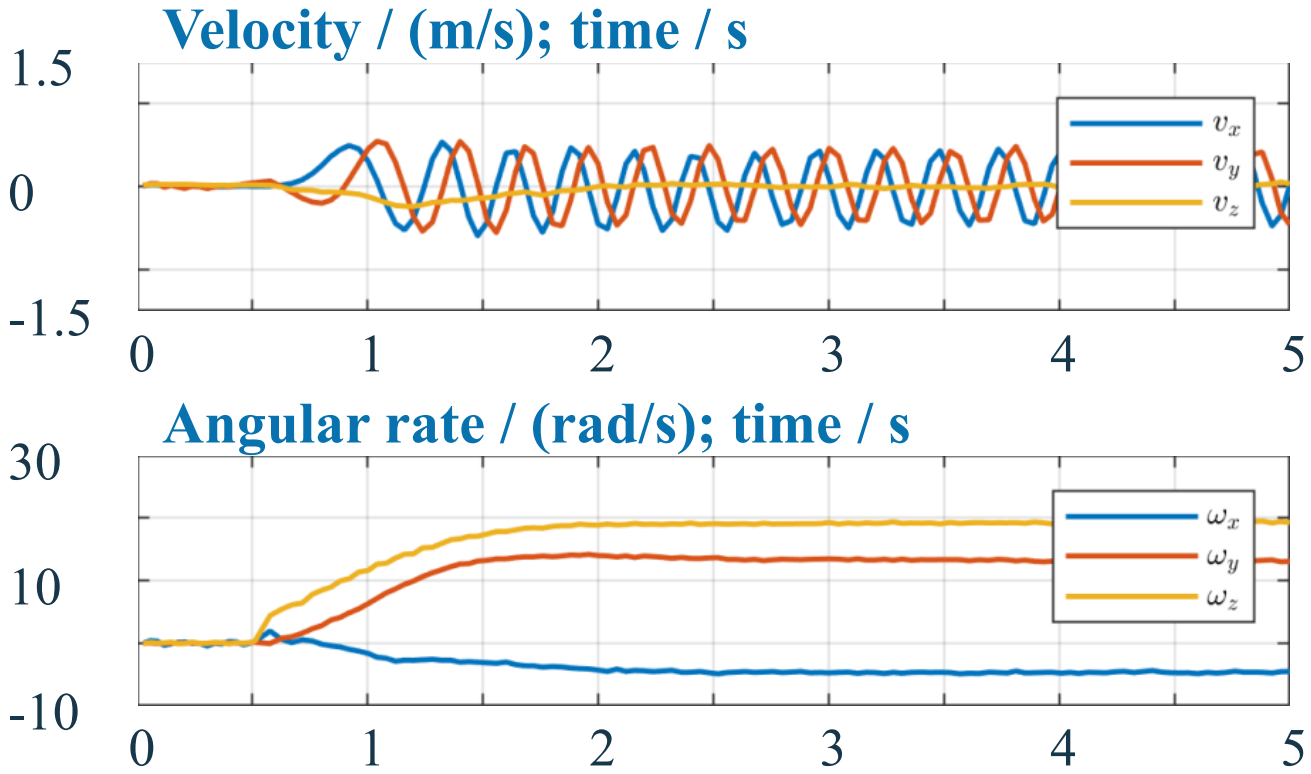
x_{err} y_{err} z_{err}

Horizontal errors approach zero; vertical error retains an offset.

Same paper, Fig. 8: first two groups of curves, rotor thrust and position error

7.6.1 Response and Control Tradeoffs: Remaining Control Limits

Position Control Comes with Continuous Spin and Velocity Oscillation



v_x v_y v_z

Near-zero position error still accompanies oscillating velocity components.

ω_x ω_y ω_z

Body z-axis angular rate is about 20 rad/s; the vehicle keeps spinning.

Same paper, Fig. 8: last two groups, velocity and rotation in the same failure test

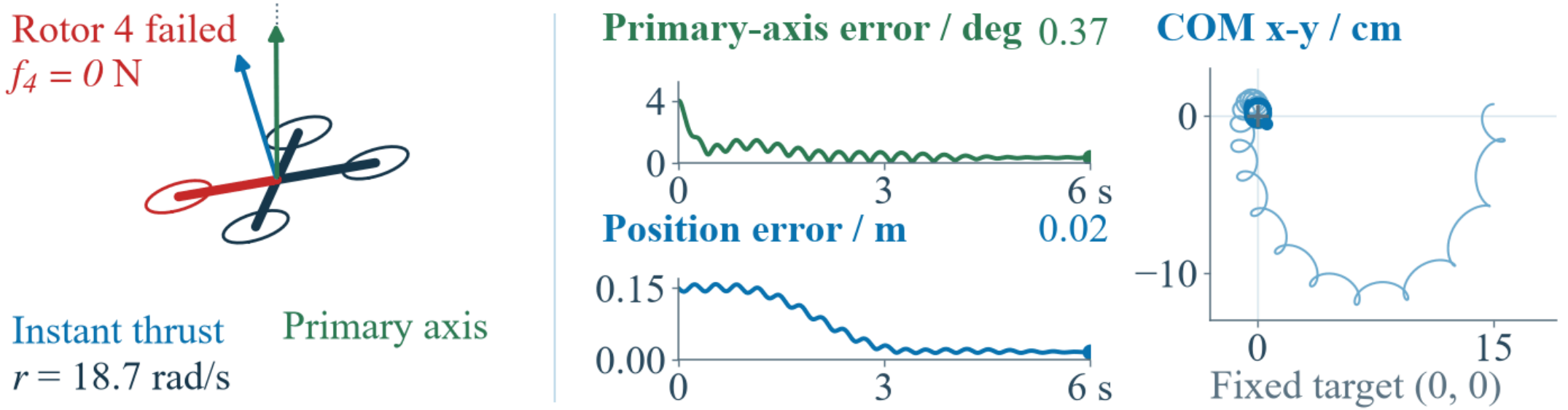
7.6.1 Response and Control Tradeoffs: Controlled Spin

Relaxed Hover: Continuous Spin and Primary-Axis Convergence

Numerical example near spinning equilibrium: rotor 4 failed initially; no switching transient.

Spin continues while the primary axis and position recover

$t = 6.00$ s



Small disturbance near spinning hover; time slowed by 2

Mueller and D' Andrea numerical example; separate from the preceding Nan et al. experiment.

Chapter Outline

Safety Decision-Making and System Improvement

Chapter 7 Health Management and Safety Assessment

7.1 Health Management Fundamentals

7.2 Unified Fault Modeling

7.3 Safety Reinforcement and Fault Data

7.4 Fault Hazards and Safety Assessment

7.5 Diagnosis and Prognostics

7.6 Safety Decisions and Improvement

7.7 Fault Injection Interfaces

7.8 Experiments and Validation

7.9 Exercises and Classroom Discussion

3 Units in This Section

7.6.1 Response and Control Tradeoffs

Textbook: Sec. 7.1.5.5

7.6.2 Hierarchical Decision-Making and Training

Textbook: Sec. 7.1.5.5

7.6.3 System Regression Testing and Improvement

Textbook: Sec. 7.1.5.6

7.6.2 Hierarchical Decision-Making and Training: Two Layers

High-Level Behavior, Low-Level Execution and Urgent Protection

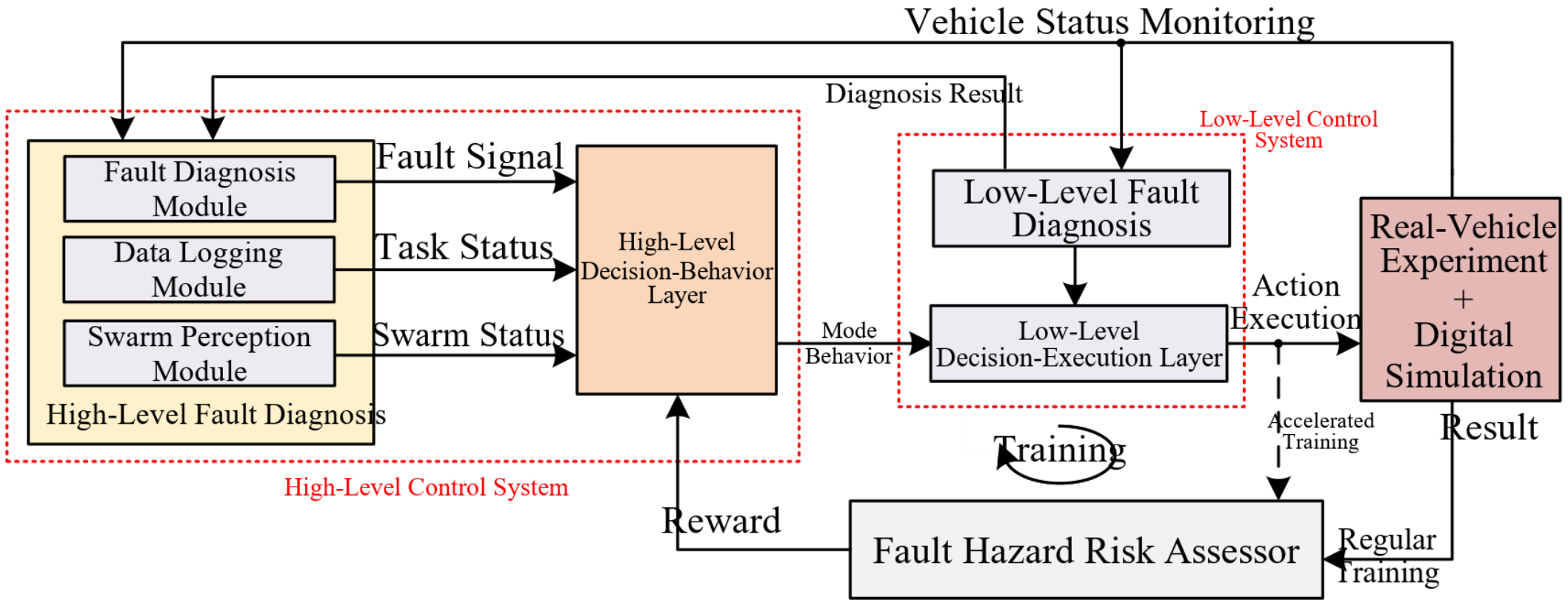
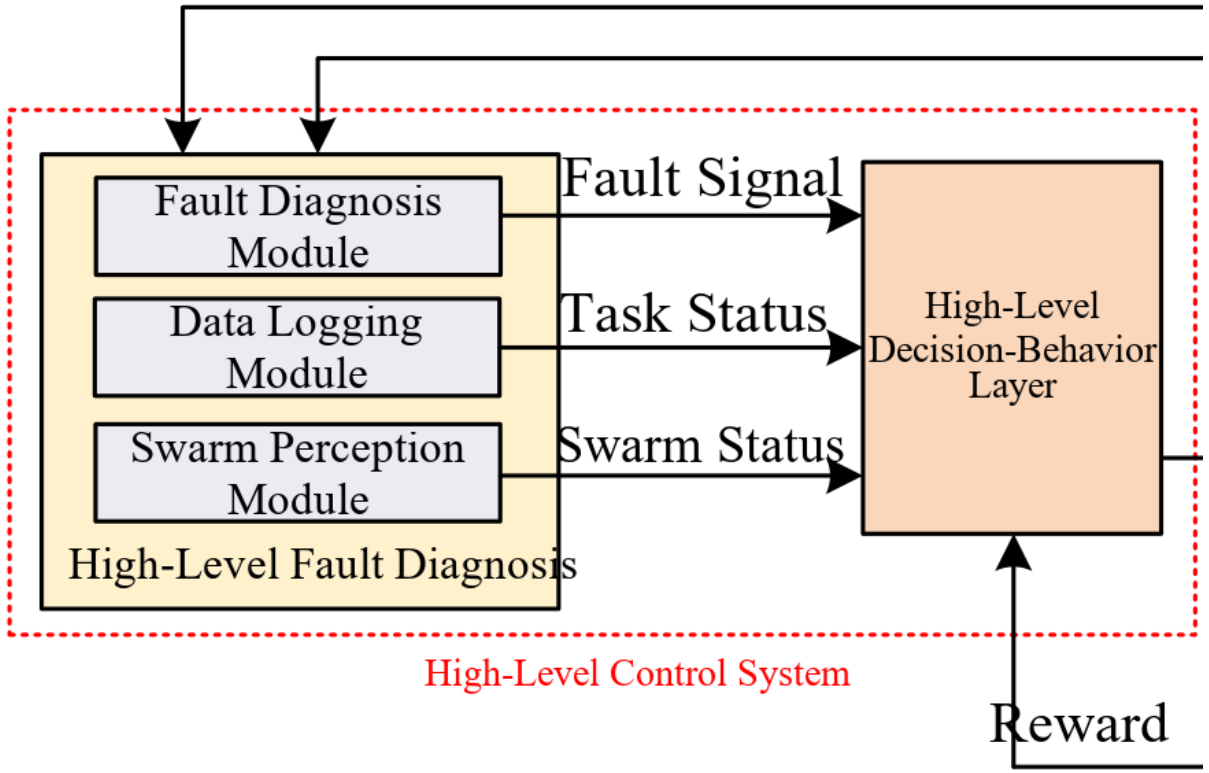


Fig. 7.11 Training Framework for the Safety Decision-Execution Layer

7.6.2 Hierarchical Decision-Making and Training: High-Level Decisions

Fuse Fault, Mission, and Swarm States to Select Behavior



Inputs: anomalies from diagnosis, remaining mission goals, and swarm coordination.

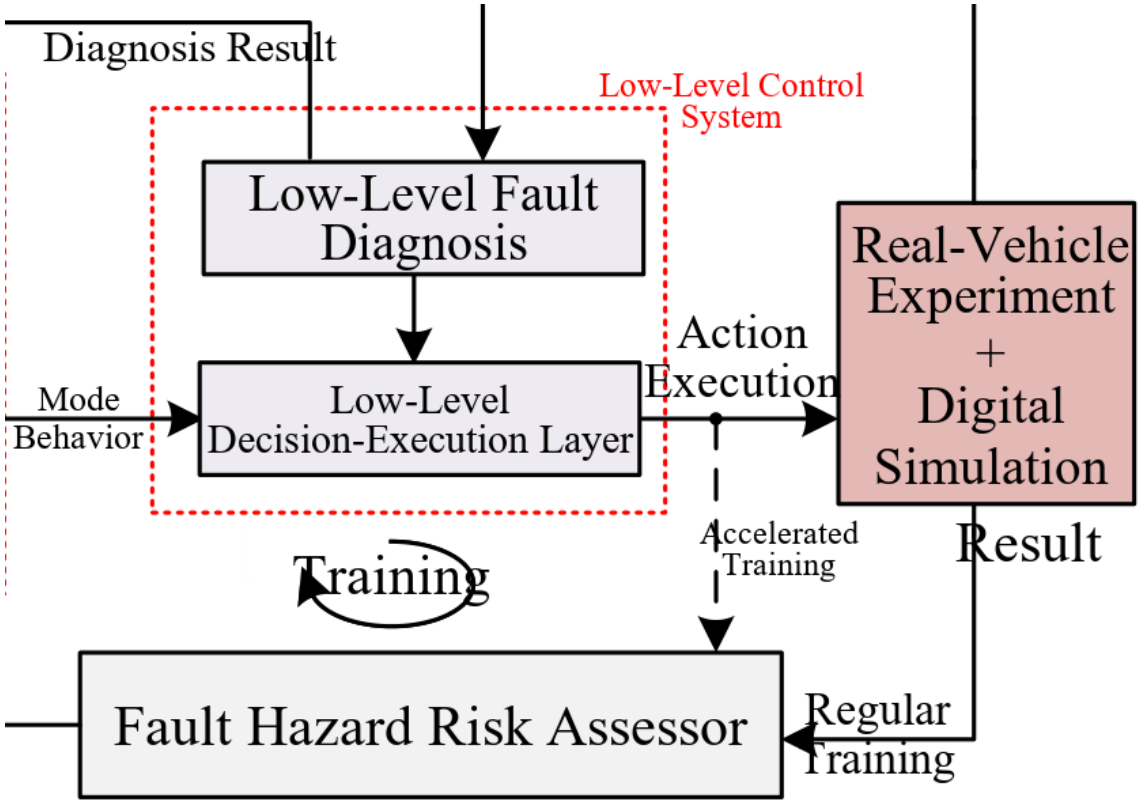
Output behavior: Return, hover, abort the mission, wait for takeover, or Land.

Feedback: check mission end state, hazard changes, and the mode actually executed below.

Fig. 7.11 (high level) Inputs to Behavior Decisions

7.6.2 Hierarchical Decision-Making and Training: Execution and Feedback

Local Protection Acts Directly; Assessment Feeds Back Results



Low level: rapid state-machine protection; hardware-in-the-loop (HITL; textbook HIL) or real-vehicle training is hard to accelerate.

Assess actual state, mode, and mission end state after execution.

Offline: a historical risk assessor screens actions; interaction tests check model errors.

Fig. 7.11 (right) Execution, Results, and Pre-Assessment

7.6.2 Hierarchical Decision-Making and Training: Response-Time Example

Urgent Faults Can Bypass the High-Level Round Trip

Teaching delays: detection 0.15 s; uplink 0.04 s; high-level computation 0.05 s; downlink 0.06 s; execution 0.01 s.

$$T_{\text{upper}} = 0.15 + 0.04 + 0.05 + 0.06 + 0.01 = 0.31\,\text{s}$$

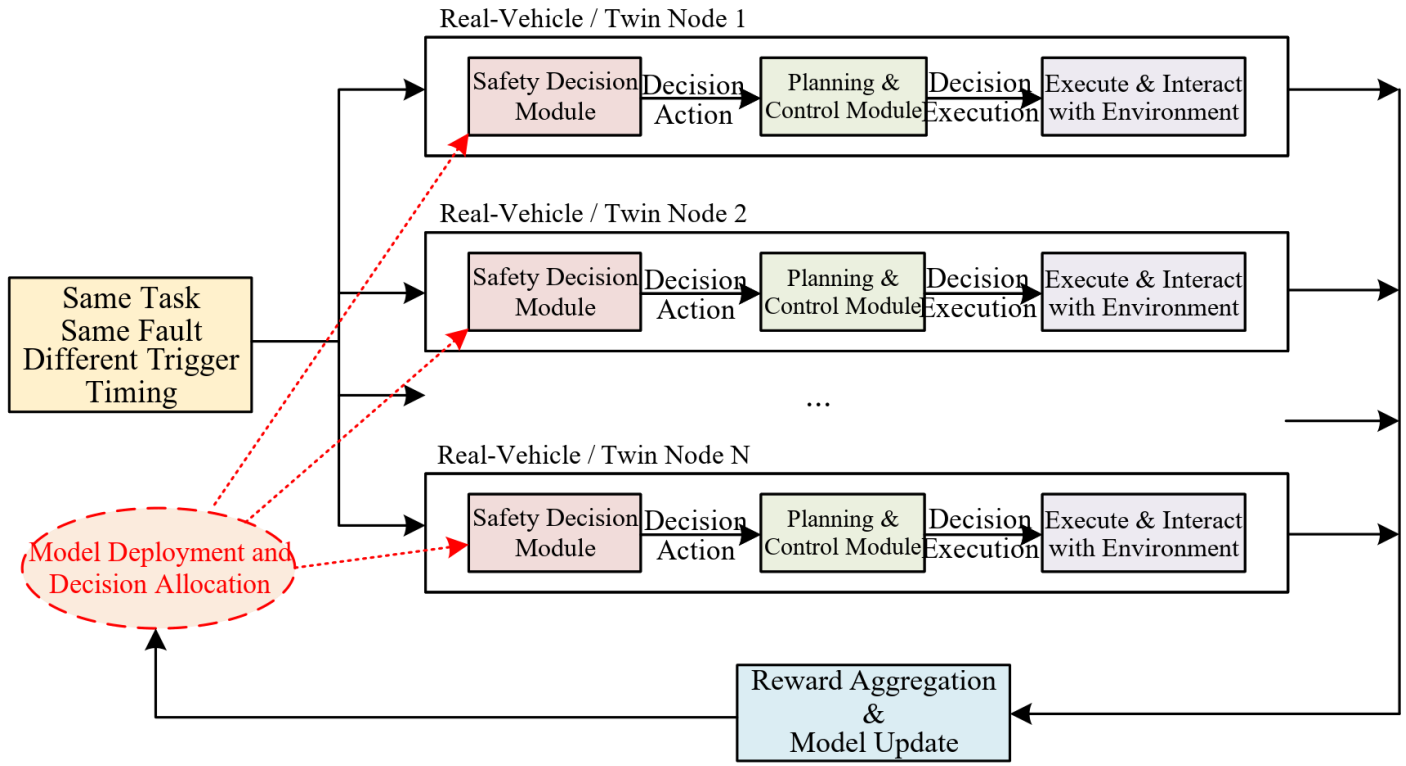
$$T_{\text{local}} = 0.15 + 0.01 = 0.16\,\text{s}$$

Both paths need detectable faults and executable protection; check actual delays in event logs.

Local emergency response reduces communication and high-level waiting.

7.6.2 Hierarchical Decision-Making and Training: Parallel Training

Parallel Nodes Broaden Conditions; Shared Feedback Updates the Model



Control variables: same mission and fault; vary trigger time or state.

Each node: select behavior, execute control, observe environmental feedback.

Shared update: aggregate results, update the model, distribute the same version.

Fig. 7.12 Parallel Safety Decision-Making Training Framework

7.6.2 Hierarchical Decision-Making and Training: Inside a Node

Decision, Control, and Interaction Are Separate Steps

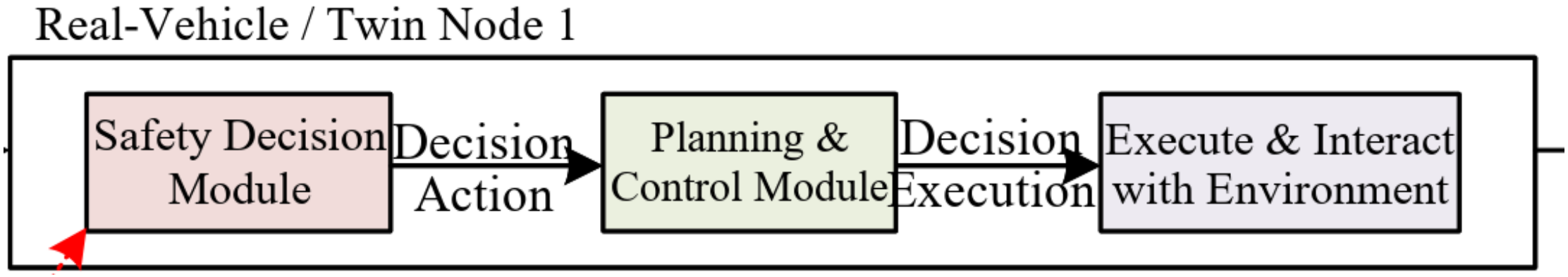


Fig. 7.12 (node 1) Behavior Reaches Motion through Control and Execution

Behavior: abort inspection and head for an alternate landing area.

Control and result: record trajectory targets, thrust/attitude outputs, actual position, and landing completion.

7.6.2 Hierarchical Decision-Making and Training: Feedback and Objectives

Rewards Follow Outcomes; Updating and Acceptance Are Separate

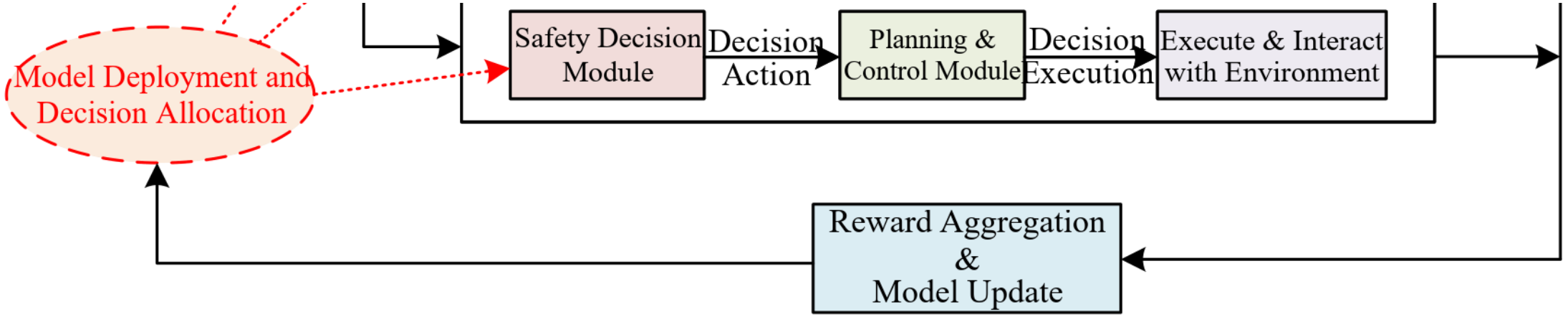


Fig. 7.12 (bottom) Result Aggregation, Model Update, and Redistribution

Check feasibility first: navigation, thrust, and available landing areas define the action set.

Then calculate feedback: mission end state, hazard change, execution constraints.

Reserve acceptance conditions: missions, faults, and trigger states excluded from updating.

7.6.2 Hierarchical Decision-Making and Training: Foundation Models and Trees

Generate Behavior Trees Offline; Revise Branches from Test Feedback

Inputs: mission description, fault cases, test data, and intended objectives.

A large language model (LLM) generates a behavior tree (BT) within mission boundaries and allowed actions; text states nodes, triggers, and outputs.

1 Generate Offline	→	2 Review and Replay	→	3 Validate / Release
Tree and nodes Candidate conditions and action logic		Conditions, outputs, branches; end states and protection		Validate independently; run regression tests; release only if criteria pass

Failure → LLM: failed scenario, triggered node, hazard consequences. Revise, review, replay, validate; release and retain the version only after criteria pass.

Only tested conditions are covered; neither generation nor one replay guarantees safety.

7.6.2 Hierarchical Decision-Making and Training: Runtime Supervision

Optional: Keep Advanced Policies within a Recoverable Region

Runtime assurance (RTA): an advanced policy proposes actions; a monitor permits execution or transfers control to a backup.



Define the backup controller's operating region and switching conditions first; then check sensing, delays, and execution margins.

NASA (National Aeronautics and Space Administration): formal runtime supervision and backup control framework.

https://ntrs.nasa.gov/citations/20240006522

Chapter Outline

Safety Decision-Making and System Improvement

Chapter 7 Health Management and Safety Assessment

7.1 Health Management Fundamentals

7.2 Unified Fault Modeling

7.3 Safety Reinforcement and Fault Data

7.4 Fault Hazards and Safety Assessment

7.5 Diagnosis and Prognostics

7.6 Safety Decisions and Improvement

7.7 Fault Injection Interfaces

7.8 Experiments and Validation

7.9 Exercises and Classroom Discussion

3 Units in This Section

7.6.1 Response and Control Tradeoffs

Textbook: Sec. 7.1.5.5

7.6.2 Hierarchical Decision-Making and Training

Textbook: Sec. 7.1.5.5

7.6.3 System Regression Testing and Improvement

Textbook: Sec. 7.1.5.6

7.6 Safety Decisions and Improvement

7.6.3 System Regression Testing and Improvement: Regression Matrix

After Policy Updates, Retest Normal, Fault, and Combined Conditions

Condition	Record for Both Policies	Main Criteria
Normal inspection	Same mission/initial state; trajectory and control outputs	Mission complete; no marked performance loss
Reduced motor effectiveness	Same fault parameters/events; diagnosis and mode timing	Response active; thrust/attitude constraints met
Fault + impaired positioning	Position validity, fallback, horizontal drift	Modes have valid prerequisites; acceptable end state
Fault + low battery	Protection level, destination, energy and landing state	Correct protection priority; avoid further battery depletion

Old/new: same initial state and fault parameters; mission-defined thresholds (teaching design).

7.6.3 System Regression Testing and Improvement: Observable Evidence

Trace Each Case to Action Effect and Actual Motion End State

Fault takes effect → **Diagnostic output** → **Mode/target changes** → **Motion and end state**

Log Evidence	Problems It Can Locate
Fault/diagnosis events	Injection active? False alarms, misses, detection delay
Requested/actual modes	Behavior accepted? Did local protection override it?
Outputs and actual state	Saturation, low thrust, estimate/model errors
Final mission state	Abort/landing complete? Unacceptable consequences?

7.6.3 System Regression Testing and Improvement: System Feedback Loop

If System Criteria Fail, Follow the Feedback Branch to Redesign

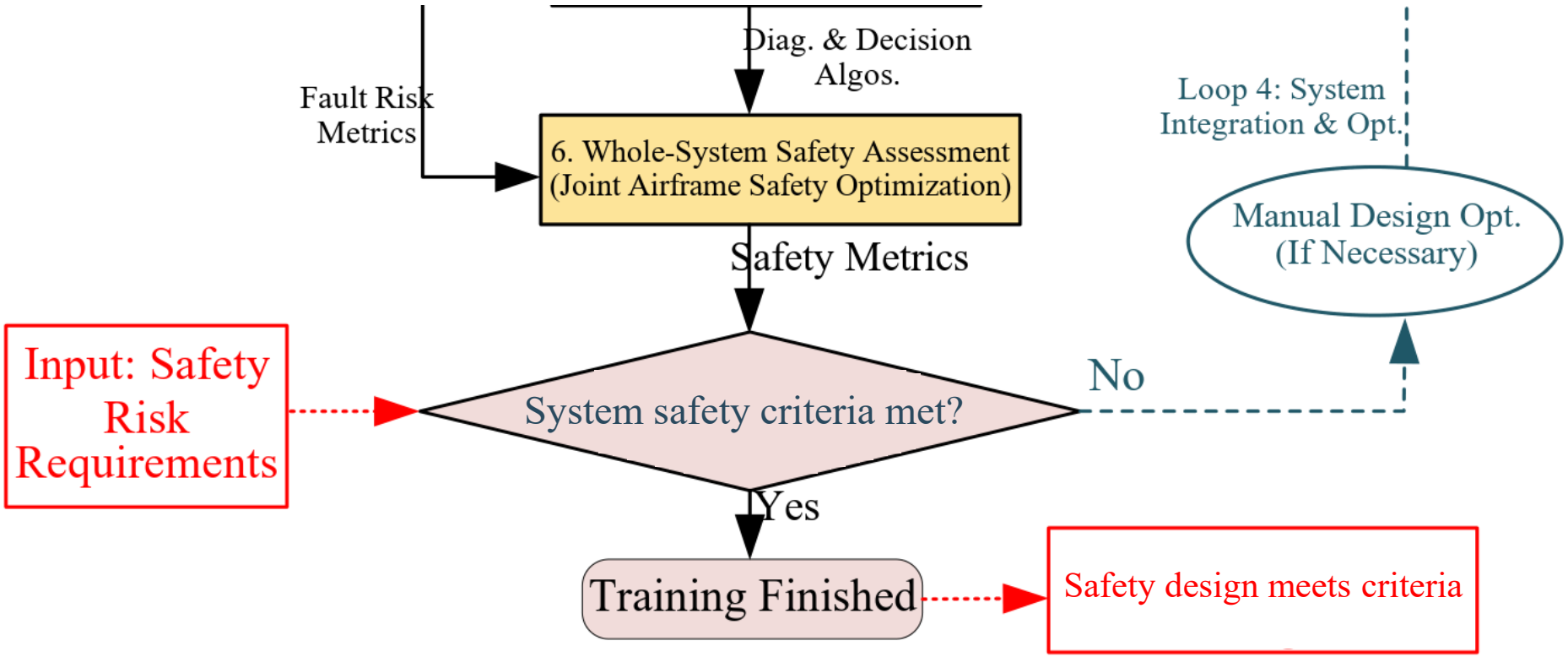


Fig. 7.6: textbook “optimal or meets criteria”; here, system safety criteria must pass.

7.6.3 System Regression Testing and Improvement: Software and Hardware

Reduce Fault Occurrence; Improve Detection and Remaining Control

Target	Inspection-Case Measures	Validation and Cost
Fault source	Component selection, connections, environmental protection	Changed fault characteristics; maintenance and cost
Detection chain	Dedicated detection and sensor cross-checks	Fault visibility, false alarms, detection delay
Remaining capability	Actuation/power/perception redundancy	Controllability; weight and common-cause faults
Software policy	Diagnostic thresholds, protection priority, control reconfiguration	Normal/fault regression; consistent modes and versions

Software improves actions; hardware changes fault occurrence and recovery limits.

Exercises and Discussion

Predict Failsafe Actions and Explain Their Preconditions

v1.12.3: COM_RC_LOSS_T=1.0 s; COM_RCL_ACT_T=5 s; remote-control loss → Return. Battery 20%/12%/6% → Warning/Return/Land; navigation, return, control initially valid. Last remote-control input: 12.0 s. Paper exercise.

Event	Question
13.5 s: remote control still lost	Which protection phase is active? When was Return due?
14.0 s: battery falls to 5%	Continue waiting in Hold or change action? Why?
Separate case: positioning lost	COM_POSCTL_NAVL=0; height also invalid. Which mode?

Reference Points

Separate Timeout, Priority, and Navigation Fallback

Timing: loss confirmed at 13.0 s; Hold at 13.5 s; Return requested at 18.0 s if no new condition intervenes.

Battery priority: 5% at 14.0 s is below the 6% emergency threshold. Land overrides the pending Return wait.

Position fallback: COM_POSCTL_NAVL=0 and invalid height select Stabilized mode, controlled by the pilot.

Check actual mode, descent response, and an acceptable landing area; acceptance of the request is only the first step.

Exercises and Discussion

Estimate Response Time and Place Protection at the Right Layer

Teaching delays: detection 0.15 s; high-level compute 0.05 s; downlink 0.06 s; execution 0.01 s.
Uplink: 0.04 → 0.2 s; protection must take effect within 0.4 s. Estimate on paper.

Topic	Question
Response time	With 0.2 s uplink, how long does each path take?
Deadline	Which path meets 0.4 s? What about the original example?
Layer assignment	Which faults act locally first, and which behaviors go to the high level?

Reference Points

Slower Communication Makes Local Emergency Protection More Important

High-level path: $T_{\text{upper}} = 0.15 + 0.2 + 0.05 + 0.06 + 0.01 = 0.47\,\text{s}$

Local path: $T_{\text{local}} = 0.15 + 0.01 = 0.16\,\text{s}$

Deadline: 0.47 s exceeds 0.4 s; only the local path qualifies. The original high-level path was 0.31 s, within the limit.

Keep local: highly urgent faults with tight timing act before reporting, e.g., emergency battery landing.

Send upward: behaviors balancing mission progress, site, and swarm state, e.g., mission abort or Return.

Section Summary

Improve the System with Executable Actions and Traceable Evidence

Diagnosis and conditions ➡ **Remaining capability** ➡ **Action and execution** ➡ **Results and improvement**

Response: triggers start protection; navigation, thrust, and site constrain actions.

Hierarchy: high level chooses; low level protects; feedback improves different targets.

Validation: same inputs/variables, traceable versions; error and tolerance limit conclusions.

Regression: normal, fault, and combined tests; improve software and hardware together.

Pitfalls: requested mode ≠ controlled motion; model credibility ≠ safety probability; wind rejection ≠ fault tolerance.