Chapter Outline

Key Swarm Technologies and Challenges

Chapter 10 Swarm Collaboration and Game-Theoretic Confrontation

10.1 Swarm Fundamentals and Architectures

10.2 Consensus and Formation Control

10.3 Game Theory and RL Fundamentals

10.4 Single-Agent RL Algorithms

10.5 Multi-Agent RL and Scaling

10.6 Technologies and Challenges

10.7 RflySim Framework and Interfaces

10.8 Experiments and Validation

10.9 Exercises and Classroom Discussion

4 Units in This Section

RL = reinforcement learning

10.6.1 Localization Fusion and a Common Coordinate Frame

Textbook: Sec. 10.1.4 · Localization and a Common Frame

10.6.2 Clock Synchronization and Motion Prediction

Textbook: Sec. 10.1.4 · Synchronization and Motion Prediction

10.6.3 Collision Detection and Avoidance

Textbook: Sec. 10.1.4 · Collision Detection and Avoidance

10.6.4 Fault Reconfiguration and Interaction Performance

Textbook: Sec. 10.1.4 · Fault Reconfiguration and Scaling

10.6.1 Localization Fusion and a Common Coordinate Frame: Engineering Overview

Information Errors Become Swarm Control Errors



State bias: Wrong frames, stale data, or drift corrupt relative position and velocity errors.

Control effects: Incorrect corrections can enlarge formation errors, consume control authority, and reduce clearance for avoidance.

System constraints: Health in Chapter 7, perception quality in Chapter 8, and message freshness in Chapter 9 determine whether inputs are usable.

10.6.1 Localization Fusion and a Common Coordinate Frame: Localization Sources

Localization Sources Have Different Failure Modes

Localization Source	Information Provided	Conditions and Failure Modes
Motion capture	Pose in an external frame	Calibration, line of sight, rigid-body ID. Occlusion breaks tracking.
GNSS / RTK / DGPS	Geographic position with differential corrections	Satellite geometry, correction link, solution status
Ultra-wideband (UWB) ranging	Anchor ranges and estimated position	Anchor geometry, non-line-of-sight bias, latency
VIO / visual SLAM	Local trajectory and map localization	Texture, lighting, inertial calibration, drift
Relative localization	Relative position, attitude, or range	Markers, depth, or point clouds. Check observability.

VIO: visual-inertial odometry. SLAM: simultaneous localization and mapping. See Chapter 8. GNSS: global navigation satellite system. RTK: real-time kinematic. DGPS: differential GPS.

10.6.1 Localization Fusion and a Common Coordinate Frame: Frame Alignment

A Common Frame Requires Rotation and Translation

$$\mathbf{p}^W = R_{WB}\mathbf{p}^B + \mathbf{t}_{WB}$$

$$R_{WB} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{p}^B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{t}_{WB} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

$$\mathbf{p}^W = \begin{bmatrix} 10 \\ 1 \end{bmatrix} \text{m}$$

Teaching assumption: A 2D rigid transform, with common frame W and local frame B. All nodes share axis directions, origin, units, and transform version.

10.6.1 Localization Fusion and a Common Coordinate Frame: Two-Source Fusion Example

Fusion Reduces Random Error but Retains Shared Bias

$$\hat{p} = \frac{z_1/\sigma_1^2 + z_2/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2}, \quad \sigma_f^2 = \frac{1}{1/\sigma_1^2 + 1/\sigma_2^2}$$

Teaching Input	Position / m	Std. Deviation / m	Norm. Weight
Source 1	10.0	0.2	0.2
Source 2	10.3	0.1	0.8
Fused output	10.24	0.0894	1.0

Conditions: Same frame and time, independent unbiased estimates. Extended Kalman filters, factor graphs, or sliding windows can fuse continuous data from multiple sources.

10.6.1 Localization Fusion and a Common Coordinate Frame: Fusion and Source Switching

Source Switching Must Preserve the Common Reference

Stage	State Processing	Swarm Control Action
Both sources available	Estimate the frame transform. Check timing and bias.	Increase the new source weight after relative position is continuous.
Old source degrades	Reduce its weight and retain uncertainty.	Increase safety margins and limit aggressive maneuvers.
New source jumps / loses tracking	Reject outliers. Fall back to prediction or a valid source.	Hold, slow down, or leave formation as capabilities permit.

Shared frames align meaning. Fusion improves estimates. Both are essential.

Chapter Outline

Key Swarm Technologies and Challenges

Chapter 10 Swarm Collaboration and Game-Theoretic Confrontation

10.1 Swarm Fundamentals and Architectures
10.2 Consensus and Formation Control
10.3 Game Theory and RL Fundamentals
10.4 Single-Agent RL Algorithms
10.5 Multi-Agent RL and Scaling
10.6 Technologies and Challenges
10.7 RflySim Framework and Interfaces
10.8 Experiments and Validation
10.9 Exercises and Classroom Discussion

4 Units in This Section

10.6.1 Localization Fusion and a Common Coordinate Frame
Textbook: Sec. 10.1.4 · Localization and a Common Frame

10.6.2 Clock Synchronization and Motion Prediction
Textbook: Sec. 10.1.4 · Synchronization and Motion Prediction

10.6.3 Collision Detection and Avoidance
Textbook: Sec. 10.1.4 · Collision Detection and Avoidance

10.6.4 Fault Reconfiguration and Interaction Performance
Textbook: Sec. 10.1.4 · Fault Reconfiguration and Scaling

10.6.2 Clock Synchronization and Motion Prediction: Time Alignment

State Age Includes Update Intervals and Network Waiting

$$\Delta t = t_{\text{now}} - t_{\text{sample}}, \quad |e_p| \approx |v|\,|e_t|$$

Error Source	Effect on Neighbor States
Sampling and publication intervals	Low-rate messages grow stale between updates.
Queueing, transmission, loss	State age keeps increasing beyond the nominal send period.
Clock offset and drift	Biased age estimates cause overprediction or underprediction.

Example: $5\ \text{m/s} \times 0.020\ \text{s} = 0.10\ \text{m}$. Timestamps and link metrics: Chapter 9.

10.6.2 Clock Synchronization and Motion Prediction: Synchronization Methods

Clock Synchronization Must Meet the Timing Budget

Method	Mechanism and Requirements	Errors to Observe
GNSS time synchronization	Common satellite time, pulse per second (PPS), hardware capture	Occlusion, receiver error, local time transfer
Network Time Protocol (NTP)	Network timestamp exchange, software clock adjustment	Path asymmetry, scheduling jitter, drift
Precision Time Protocol (PTP)	Precise timestamp exchange, optional hardware timestamping	Network card and switch support, path and clock configuration

Derive the timing budget from speed and allowable position error. Measure synchronization and holdover after signal loss. A protocol name does not guarantee accuracy.

10.6.2 Clock Synchronization and Motion Prediction: Motion Prediction

Predicting the Current Position from an Old State

$$\hat{\mathbf{p}}(t+\Delta t) \approx \mathbf{p}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2 \tag{10.46}$$

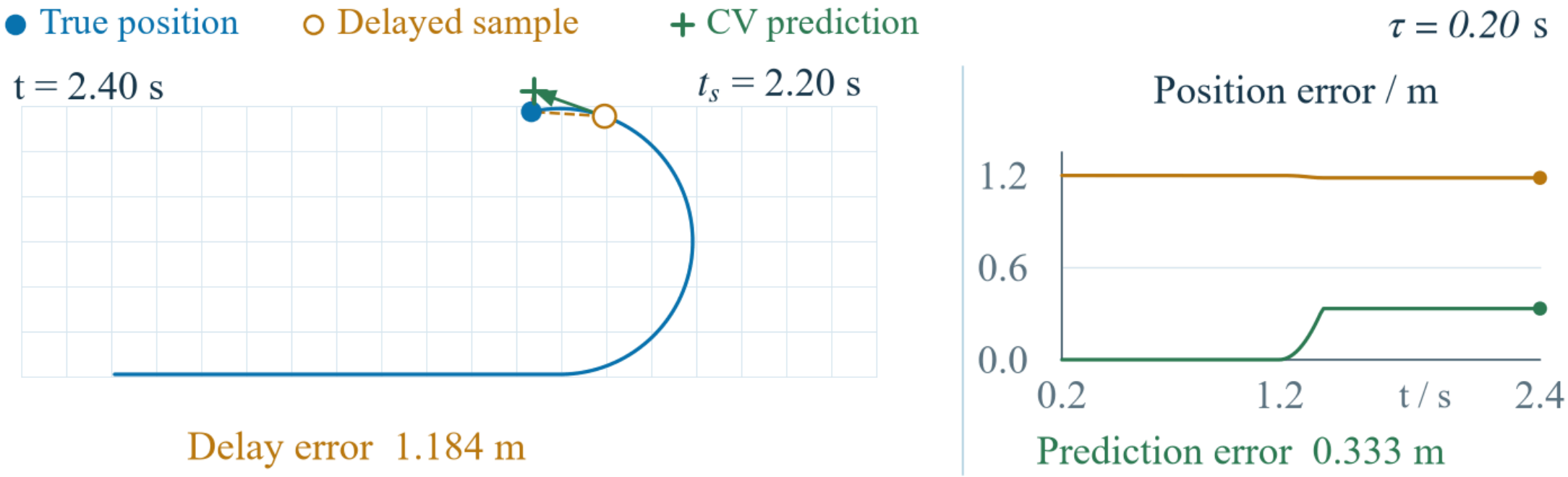
Units: Position in m, velocity in m/s, acceleration in m/s², time in s. All vectors use the same common frame.

$$p = 10, \quad v = 2, \quad a = 1, \quad \Delta t = 0.2 \quad \Rightarrow \quad \hat{p} = 10.42\ \text{m}$$

Conditions: Short-term motion admits a constant-velocity or constant-acceleration approximation, and timestamps are trustworthy. Prediction compensates for stale data but introduces model error.

10.6.2 Clock Synchronization and Motion Prediction: Reproducible Prediction Example

Old and Predicted Positions on the Same Trajectory



Eq. 10.46, prediction acceleration = 0. Turning leaves a residual.

Animation: 6 m/s, delay 0.2 s, constant velocity. Previous: 2 m/s, constant acceleration.

10.6.2 Clock Synchronization and Motion Prediction: Error Bounds and Buffering

Uncertainty Limits the Prediction Horizon

Constant-velocity extrapolation assumes exact initial position, velocity, and state age, with $\|\mathbf{a}(t)\| \le a_{\max}$ throughout the interval.

$$|e_{\text{model}}| \le \frac{1}{2} a_{\max} \Delta t^2, \quad a_{\max} = 2, \quad \Delta t = 0.5 \quad \Rightarrow \quad |e_{\text{model}}| \le 0.25 \text{ m}$$

Buffering and interpolation: Interpolate between two samples to align past states. The current time beyond the latest message still requires extrapolation.

Error scope: e_{model} is model truncation error only. Total error also includes initial estimation and clock errors.

Fallback: Beyond the age or error threshold, use local perception and slow down, increase spacing, or leave the cooperative task.

Richer models: Filters or learned spatiotemporal models can use history and neighbor interactions. Test generalization on unseen maneuvers.

Chapter Outline

Key Swarm Technologies and Challenges

Chapter 10 Swarm Collaboration and Game-Theoretic Confrontation

10.1 Swarm Fundamentals and Architectures

10.2 Consensus and Formation Control

10.3 Game Theory and RL Fundamentals

10.4 Single-Agent RL Algorithms

10.5 Multi-Agent RL and Scaling

10.6 Technologies and Challenges

10.7 RflySim Framework and Interfaces

10.8 Experiments and Validation

10.9 Exercises and Classroom Discussion

4 Units in This Section

10.6.1 Localization Fusion and a Common Coordinate Frame

Textbook: Sec. 10.1.4 · Localization and a Common Frame

10.6.2 Clock Synchronization and Motion Prediction

Textbook: Sec. 10.1.4 · Synchronization and Motion Prediction

10.6.3 Collision Detection and Avoidance

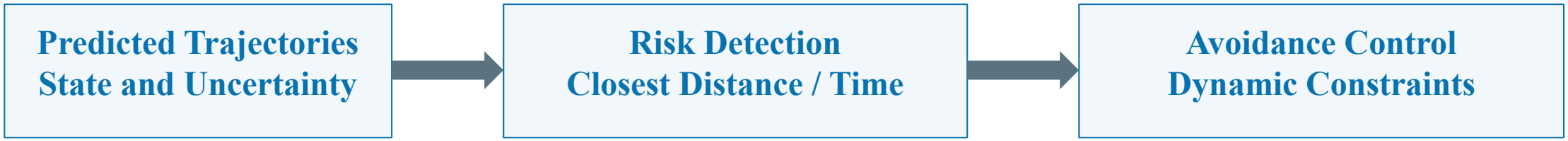
Textbook: Sec. 10.1.4 · Collision Detection and Avoidance

10.6.4 Fault Reconfiguration and Interaction Performance

Textbook: Sec. 10.1.4 · Fault Reconfiguration and Scaling

10.6.3 Collision Detection and Avoidance: Detection and Control

Detection and Avoidance Have Different Roles



Contact detection: Records collisions that have occurred, including object IDs and positions, for simulation fault analysis.

Predictive detection: Estimates future closest separation to flag risk early. It depends on state freshness and model assumptions.

Avoidance loop: After changing commands, monitor actual separation, saturation, and formation recovery. Target-point spacing alone is insufficient.

10.6.3 Collision Detection and Avoidance: Velocity Obstacle Intuition

Relative Motion Determines Closest Approach

$$\mathbf{r}(\tau) = \mathbf{r}_0 + \mathbf{v}_{\text{rel}}\tau, \quad \tau_* = \text{clip}\left(-\frac{\mathbf{r}_0^T \mathbf{v}_{\text{rel}}}{\|\mathbf{v}_{\text{rel}}\|^2}, 0, T\right)$$

$$\mathbf{r}_0 = (4,1)\ \text{m}, \quad \mathbf{v}_{\text{rel}} = (-2,0)\ \text{m/s}, \quad T = 3\ \text{s}$$

$$\tau_* = 2\ \text{s}, \quad d_{\min} = \|(0,1)\| = 1\ \text{m} < 1.5\ \text{m}$$

clip restricts time to the horizon. Example safety threshold: 1.5 m.

Conditions: Constant velocity over a short horizon, with aligned positions and velocities. At zero relative velocity, distance is constant. Do not use the zero-denominator formula.

10.6.3 Collision Detection and Avoidance: Execution Limits

Safe Separation Includes Reaction and Braking

Model: One-dimensional head-on approach. Closing speed during reaction is at most v_c.
Then relative deceleration along the approach is at least $a_c > 0$ until closing speed reaches 0.

$$d_{\text{req}} = R + \epsilon + v_c T_r + \frac{v_c^2}{2a_c}$$

$$R = 1,\ \epsilon = 0.4,\ v_c = 4,\ T_r = 0.15,\ a_c = 4$$

$$d_{\text{req}} = 1 + 0.4 + 4 \cdot 0.15 + \frac{4^2}{2 \cdot 4} = 4.0\ \text{m}$$

R and ϵ are the combined protection radius and associated state margin, in m. Time is in s.
The 4 m result is specific to this example, not a universal real-vehicle safety distance.

10.6.3 Collision Detection and Avoidance: Algorithm Mechanisms

Avoidance Methods Address Different Constraints

Method	Mechanism	Conditions and Limits
Artificial potential field	Combine attraction to the goal with repulsion from neighbors.	May stall. Saturation and discrete execution affect the result.
Velocity obstacle	Exclude relative velocities that enter the unsafe region.	Depends on motion prediction and assumed neighbor responses.
Model predictive control (MPC)	Optimize future trajectories subject to constraints.	Model quality, feasibility, real-time solver budget
Perception fusion / end-to-end policy	Improve obstacle information / map inputs directly to actions.	Perception errors, training distribution, action feasibility

10.6.3 Collision Detection and Avoidance: System Integration

Safety Arbitration and Task Recovery

State	Command Authority and Action	Observable Exit Condition
Normal task	Main controller commands. Safety monitor remains active.	Predicted distance or error margin triggers risk.
Risk response	Safety module modifies commands within actuator limits.	Separation recovers and risk stays below the exit threshold.
Task recovery	Replan from the current state. Avoid jumping to the old target.	Tracking and formation errors return to allowed ranges.

Over the full run, check minimum distance, feasible actions, and task recovery.

Chapter Outline

Key Swarm Technologies and Challenges

Chapter 10 Swarm Collaboration and Game-Theoretic Confrontation

10.1 Swarm Fundamentals and Architectures
10.2 Consensus and Formation Control
10.3 Game Theory and RL Fundamentals
10.4 Single-Agent RL Algorithms
10.5 Multi-Agent RL and Scaling
10.6 Technologies and Challenges
10.7 RflySim Framework and Interfaces
10.8 Experiments and Validation
10.9 Exercises and Classroom Discussion

4 Units in This Section

10.6.1 Localization Fusion and a Common Coordinate Frame
Textbook: Sec. 10.1.4 · Localization and a Common Frame

10.6.2 Clock Synchronization and Motion Prediction
Textbook: Sec. 10.1.4 · Synchronization and Motion Prediction

10.6.3 Collision Detection and Avoidance
Textbook: Sec. 10.1.4 · Collision Detection and Avoidance

10.6.4 Fault Reconfiguration and Interaction Performance
Textbook: Sec. 10.1.4 · Fault Reconfiguration and Scaling

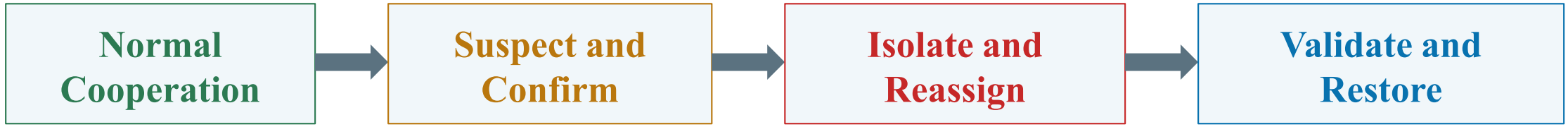
10.6.4 Fault Reconfiguration and Interaction Performance: Fault Diagnosis

Node Faults, Link Loss, and Task Failure Differ

Anomaly Scope	Evidence	Swarm Effect and Response
Node health anomaly	Battery, sensor residuals, flight control state	Reduced capability. Limit the task and arrange takeover.
Communication loss	Message age, heartbeats, neighbor perception	Reject stale states first. Distinguish node failure from link loss.
Unreliable localization	Jumps, covariance, source-switch status	Increase spacing or degrade operation. Reject wrong relative positions.
Task cannot continue	Remaining resources, routes, changes to targets	Reassign roles, regions, or formation topology.

10.6.4 Fault Reconfiguration and Interaction Performance: States and Recovery

Diagnosis Triggers Isolation and Task Reconfiguration



Decision conditions: Combine duration, independent evidence, and task risk. A single lost packet should not trigger swarm-wide reconfiguration.

Reconfiguration: Change leaders, communication topology, task regions, or roles. Reduce task objectives if remaining resources are insufficient.

Recovery: Verify frames, clocks, health, and task versions before rejoining. Prevent duplicate assignments.

10.6.4 Fault Reconfiguration and Interaction Performance: Reassignment Example

Task Takeover Must Satisfy Time and Capacity Limits

Teaching assumptions: Two vehicles each take one inspection point, depart together, and follow nonconflicting routes. Cost is estimated flight time.

Candidate Vehicle	Inspection Point P	Inspection Point Q
A	3 s	7 s
B	4 s	4 s

Assignment	Time to Finish Both	Total Flight Time
A to P, B to Q	$\max(3,4) = 4$ s	$3 + 4 = 7$ s
A to Q, B to P	$\max(7,4) = 7$ s	$7 + 4 = 11$ s

10.6.4 Fault Reconfiguration and Interaction Performance: Computation and Communication

Interaction Structure and Deadlines Determine Load

$$E_{\text{all}} = N(N-1), \quad E_{\text{local}} = Nk, \quad T_{\text{calc}} = 5 + 0.01E \text{ ms}$$

50-Node Teaching Setup	Directed Pairs	Serial Time	20 ms Budget
All-to-all	2450	29.5 ms	Deadline missed
6 neighbors per node	300	8.0 ms	Meets this compute budget

Assume constant cost per interaction. Also measure message age, perception, and execution time. Sparsity, grouping, and parallelism must retain task-relevant information.

10.6.4 Fault Reconfiguration and Interaction Performance: Interaction and Structure

Grouping and Roles Reduce Operator Workload

Managed Object	Displayed and Controlled Information	Conflict Reduced
Task group	Coverage, completion, remaining group resources	Repeatedly sending the same target to individual vehicles
Anomalous node	Health, state age, localization quality, takeover progress	Confusing root causes with consequences
Heterogeneous role	Capabilities, payloads, command authority, task version	Invalid takeover, duplicate tasks, unauthorized commands

Teaching workload: $50 \times 2 = 100$ s. For 10 groups and 3 anomalies: $10 \times 2 + 3 \times 5 = 35$ s. Summaries must retain access to detailed anomaly evidence.

10.6.4 Fault Reconfiguration and Interaction Performance: Integrated Example

State Processing Across Indoor and Outdoor Tasks

Scenario Change	System Action	Assessment Metric
Outdoor to indoor	Align GNSS with the local map. Gradually change fusion weights.	Common-frame position jumps, relative error
5 Hz neighbor messages	Predict to control time using actual state age.	Prediction error, maximum message age
Crossing in a narrow area	Adjust trajectories using error margins and execution limits.	Minimum distance, fraction of saturated actions
Persistent loss of localization	Degrade, isolate, reassign, and update the task version.	Recovery time, task completion rate

Exercises and Discussion

What Remains Necessary After Prediction?

Question	Reference Answer
Old position 5 m, velocity 3 m/s, acceleration −1 m/s², age 0.2 s. What is the predicted position?	$5 + 3 \cdot 0.2 - 0.5 \cdot 1 \cdot 0.2^2 = 5.58$ m
If neighbor packets keep being lost after prediction, is flying at the same speed reasonable?	Check age, error bounds, and local perception. Then slow down, increase spacing, or withdraw as capabilities permit.
Is assigning a failed node's task to the nearest vehicle always optimal?	Capacity, payload, route conflicts, and existing tasks also constrain the choice. Compare feasible assignments.

Section Summary

Trustworthy States and Feasible Cooperation

Space: Keep transforms, calibration, and map versions consistent. Fusion needs appropriate noise and correlation assumptions.

Time: Synchronization makes timestamps comparable. Prediction reduces state lag, while model error limits extrapolation.

Safety and faults: Detection triggers feasible avoidance. Diagnosis, isolation, and task takeover jointly support recovery.

Common errors: Detection does not guarantee avoidance. Prediction does not remove delay. Node count alone does not determine compute or communication load.