Intelligent Unmanned System Control Practice

Chapter 5 Filtering, State Estimation, and Low-Level Control

Author: Xunhua Dai, Central South University

Website: https://faculty.csu.edu.cn/daixunhua

Email: dai.xh@csu.edu.cn

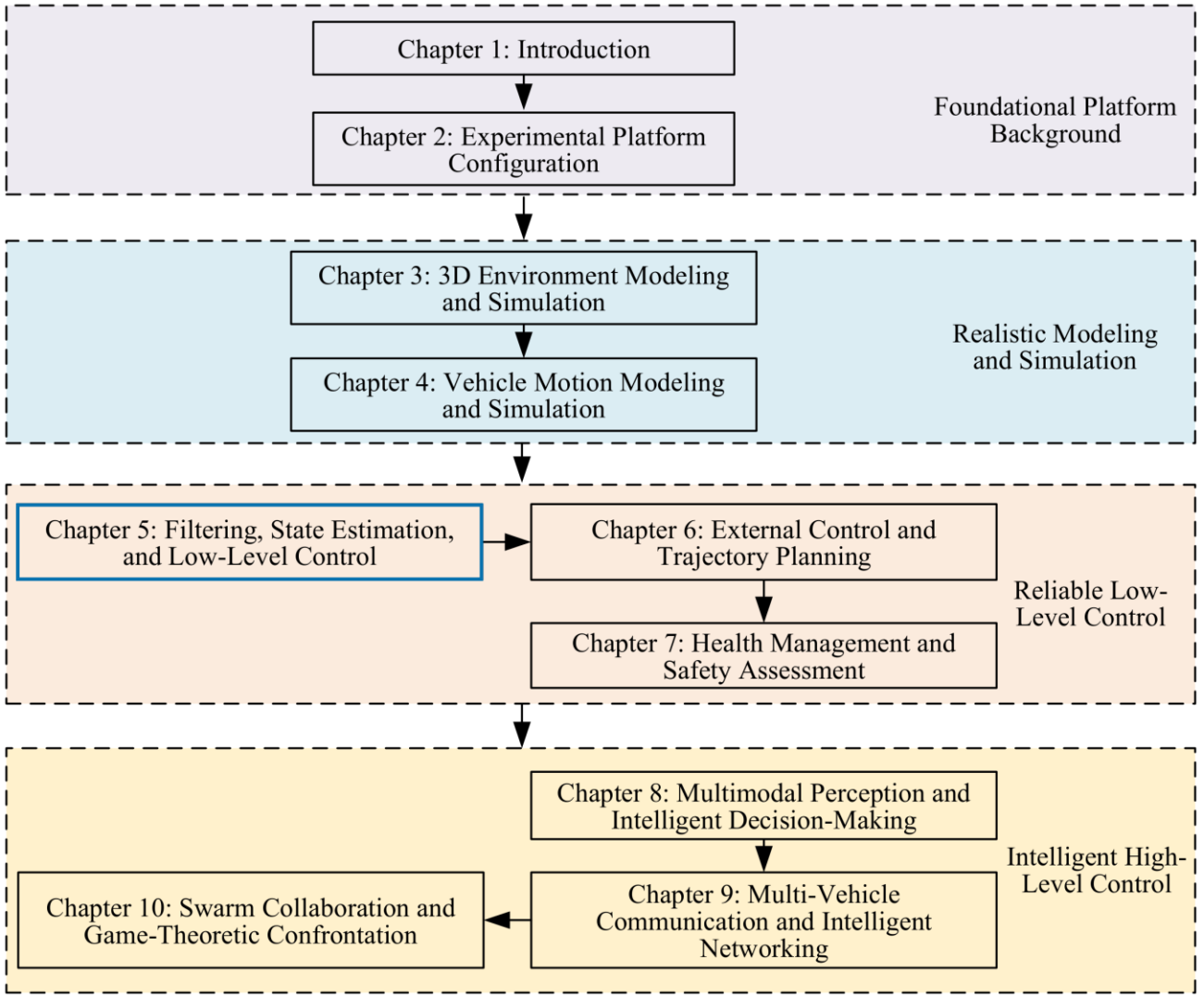
RflySim: http://rflysim.com

Companion Paper



Dai, Xunhua, Jinhu Tu, and Quan Quan. "RflySim ToolChain: a rapid development and validation toolchain for intelligent unmanned swarm systems." Journal of Systems Engineering and Electronics 36.4 (2025): 1077-1093.

Textbook is coming soon...



Chapter Learning Objectives

What You Can Do After Chapter 5

1. Explain the roles of estimation and control in a feedback loop
2. Perform a filter update, a control calculation, and control allocation
3. Evaluate and tune control performance using responses and control inputs
4. Explain staged validation with models, firmware, hardware, and real vehicles

Chapter Outline

Filtering, State Estimation, and Low-Level Control: Chapter Roadmap

Chapter 5 Filtering, State Estimation, and Low-Level Control

5.1 Filtering and Low-Level Control

5.2 Sensor Calibration and Estimation

5.3 Controller Design and Allocation

5.4 Control Performance and Tuning

5.5 PX4 Architecture and Interfaces

5.6 Control Development and Testing

5.7 Other Vehicles and Advanced Cases

5.8 Exercises and Classroom Discussion

4 Units in This Section

5.1.1 Control Principles of Unmanned Systems

Textbook: Sec. 5.1.1.1

5.1.2 Importance of Low-Level Control

Textbook: Sec. 5.1.1.2

5.1.3 Importance of Filtering and State Estimation

Textbook: Sec. 5.1.1.3

5.1.4 Low-Level Control Development Process

Textbook: Secs. 5.1.1.4 and 5.1.2

5.1.1 Control Principles of Unmanned Systems: Modeling and Control

Modeling and Control Answer Different Questions

Motion model	**Input: thrust and torque**	→	**Model**	→	**Output: motion state**
Control design	**Target and state feedback**	→	**Controller**	→	**Output: actuator commands**

- **Plant:** the physical system being controlled; here, the UAV body.
- **Controlled variable:** the variable to regulate; altitude in an altitude-hold task.
- **Controller:** computes corrective commands from the target and feedback; actuators apply them to the vehicle.

5.1.1 Control Principles of Unmanned Systems: Altitude Hold

Why Can the Same Throttle Fail to Hold Altitude?

$$m\ddot{h} = T - mg + d$$

1D model: altitude is positive upward; north-east-down (NED) position is positive downward.

Conditions: level attitude and zero drag.

Case	Thrust versus weight	Change in motion
Original mass, stationary hover	Total thrust equals weight	Zero vertical acceleration
More payload, unchanged thrust	Total thrust is below the new weight	Begins accelerating downward
Brief downward external force	Net vertical force is downward	Altitude and velocity change

h: altitude; T: total thrust; m: mass; g: gravitational acceleration; d: vertical disturbance force (positive upward).

5.1.1 Control Principles of Unmanned Systems: System Components

How Do Target, Controller, and Vehicle Close the Loop?

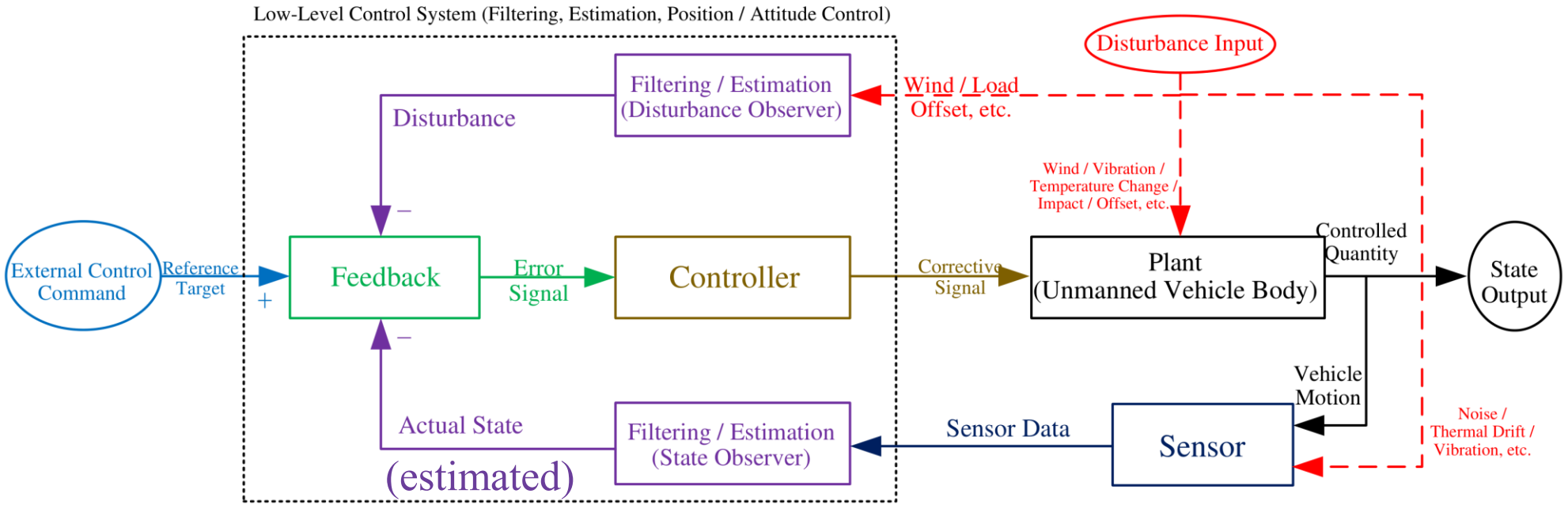
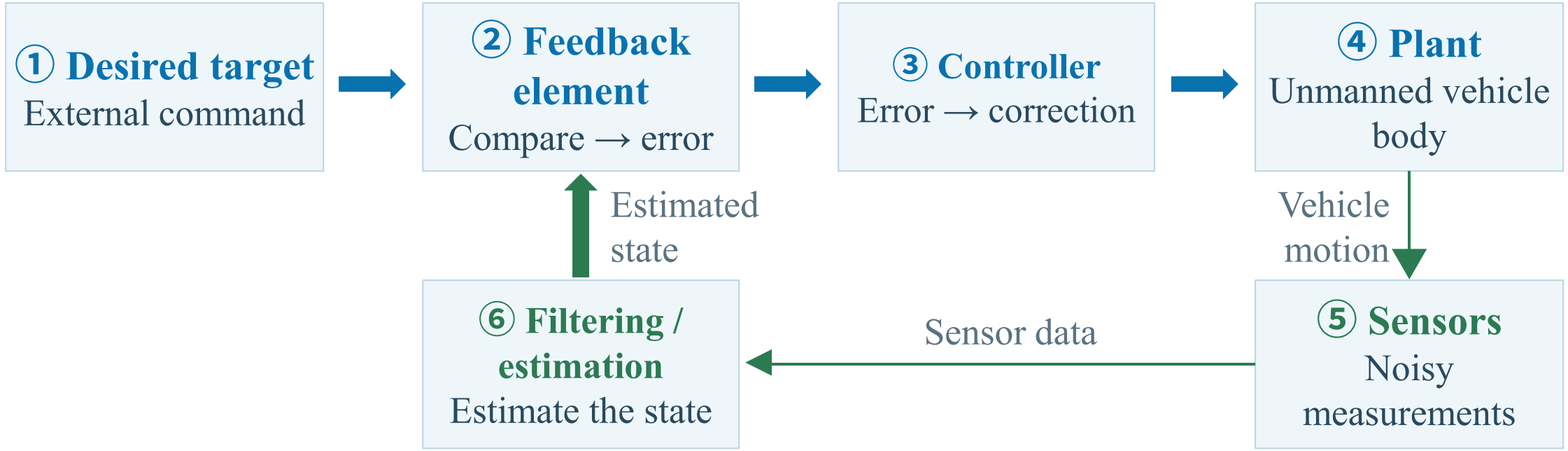


Fig. 5.1 Control Principles of Unmanned Systems

Infer disturbance from inputs and observations, then convert it to compensation; these connections are omitted.

5.1.1 Control Principles of Unmanned Systems: Reading the Main Loop

Fig. 5.1: Compare, Correct, Measure, and Estimate



- **Feedback is estimated:** "Actual State" in Fig. 5.1 is the filtering and estimation output.
- **Entry points differ:** external forces act on the vehicle; measurement noise affects sensors first.

5.1.1 Control Principles of Unmanned Systems: Open and Closed Loops

One Disturbance: How Does Feedback Change Motion?

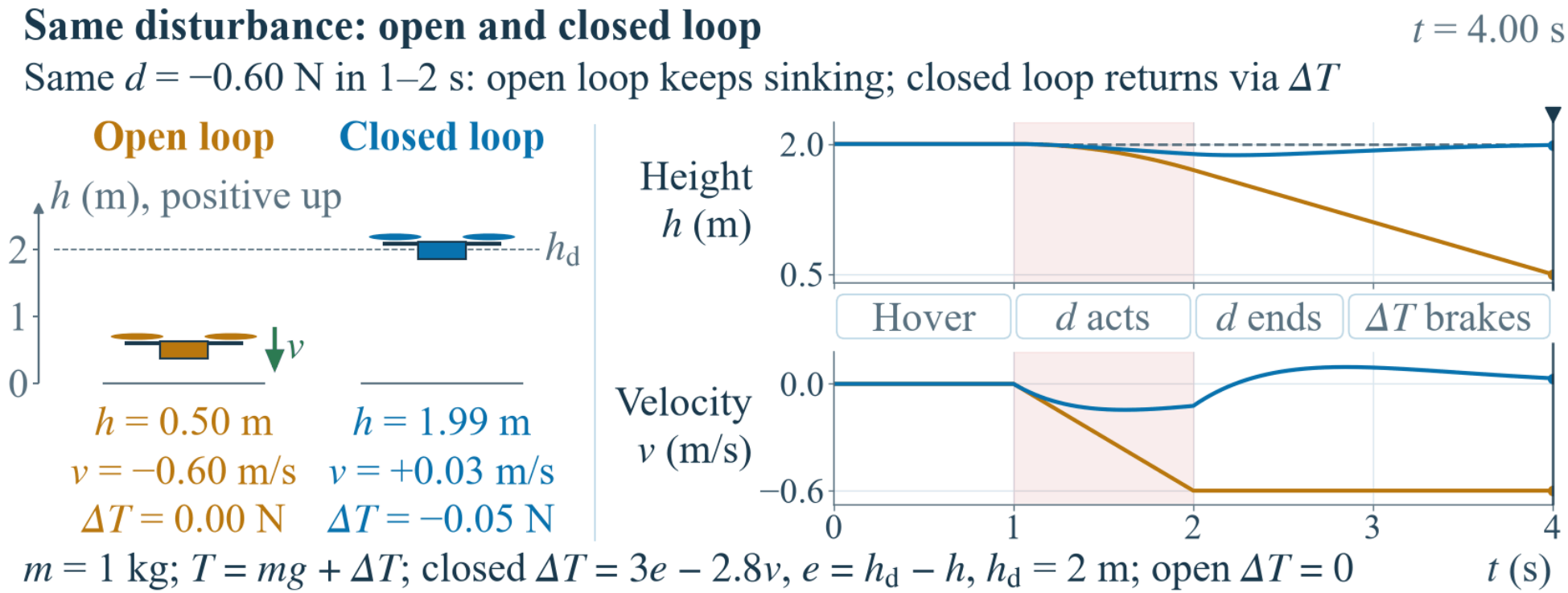


Figure: open- and closed-loop altitude and velocity under the same disturbance (1D numerical simulation)

5.1.1 Control Principles of Unmanned Systems: Error and Action

From Altitude Error to Thrust Correction

Given: target 2.0 m, estimate 1.8 m, mass 1 kg, proportional gain 3 N/m.

Condition: $v = 0$ m/s; $\hat{h}$ is estimated altitude and h_d is target altitude (subscript d means desired).

$$e = h_d - \hat{h} = 2.0 - 1.8 = 0.2\,\text{m}$$

$$\Delta T = K_p e = 3\ \text{N/m} \times 0.2\,\text{m} = 0.6\,\text{N}$$

$$T = mg + \Delta T = 9.81 + 0.6 = 10.41\,\text{N}$$

- **Feedforward and feedback:** gravity feedforward gives hover thrust; feedback corrects the error.
- Continuous motion: $-2.8v$ in the animation is velocity feedback for damping; assess stability from the full response.

Chapter Outline

Filtering, State Estimation, and Low-Level Control: Chapter Roadmap

Chapter 5 Filtering, State Estimation, and Low-Level Control

5.1 Filtering and Low-Level Control

5.2 Sensor Calibration and Estimation

5.3 Controller Design and Allocation

5.4 Control Performance and Tuning

5.5 PX4 Architecture and Interfaces

5.6 Control Development and Testing

5.7 Other Vehicles and Advanced Cases

5.8 Exercises and Classroom Discussion

4 Units in This Section

5.1.1 Control Principles of Unmanned Systems

Textbook: Sec. 5.1.1.1

5.1.2 Importance of Low-Level Control

Textbook: Sec. 5.1.1.2

5.1.3 Importance of Filtering and State Estimation

Textbook: Sec. 5.1.1.3

5.1.4 Low-Level Control Development Process

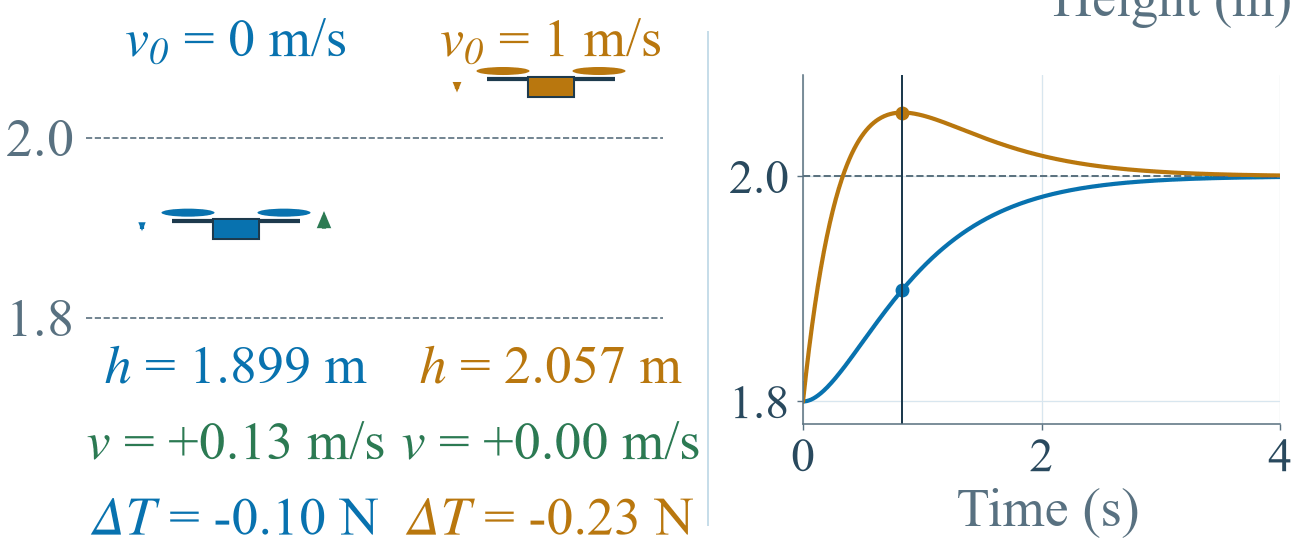
Textbook: Secs. 5.1.1.4 and 5.1.2

5.1.2 Importance of Low-Level Control: Inertia and Response

Why Overshoot When Already Near the Target?

Less thrust does not stop ascent instantly

Same initial height; different initial speeds



- **Same initial position:** both are 0.2 m from the target.
- **Different velocities:** faster ascent requires earlier thrust reduction.
- **Coordinated feedback:** position brings it closer; velocity helps it slow down.

Figure: altitude responses with two initial velocities under the same law $\Delta T = 4e - 4v$ (teaching example)

5.1.2 Importance of Low-Level Control: Low-Level and High-Level Control

Motion Targets Need Stabilizing Control

Mission level — **Inspect a location or follow a given route**

↓ Generate motion targets by level

Motion targets — **Desired position, velocity, attitude, or angular rate**

↓ Send targets to each control loop

Low-level control — **Regulate pose using state feedback and command the actuators**

Lower-level external targets, such as attitude or angular rate, make the external program responsible for more control layers.

Reliable autonomous tasks depend first on the vehicle's low-level motion capabilities.

5.1.2 Importance of Low-Level Control: Disturbances and Uncertainty

Different Problems Enter at Different Points in the Loop

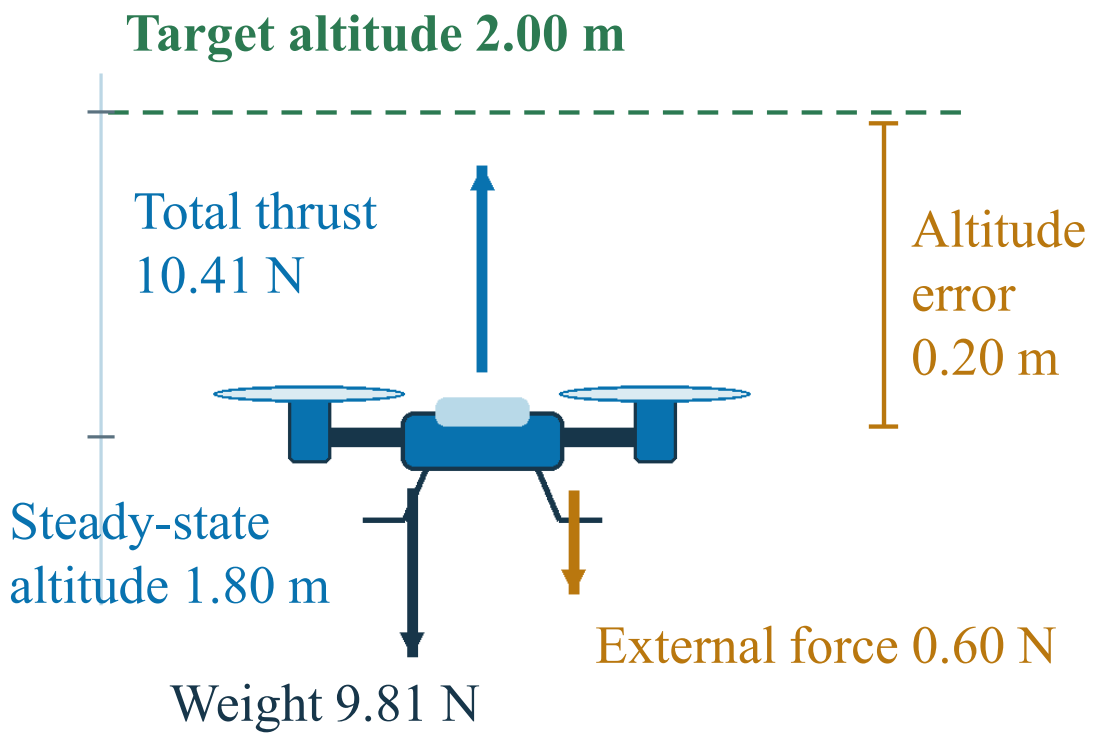
Factor	What it affects first	Evidence to check first
External disturbances	Vehicle forces and motion	Velocity changes and attitude deviations
Measurement noise and drift	Sensor readings	Raw measurements, bias, and residuals
Actuator limits	Actual forces and torques	Requested and actual values, saturation
Model uncertainty	Predicted versus actual response	Mass, inertia, and response differences
Faults and packet loss	Data or component availability	Timestamps, validity flags, and state transitions

Zero bias: zero-input output offset; residual: measurement minus prediction; validity flag: whether data is usable.

5.1.2 Importance of Low-Level Control: Persistent External Force

Persistent Force: Why Settle Below the Target?

Same 1 kg vehicle, 2 m target, and velocity damping; a persistent downward force of 0.60 N.



Stable and unsaturated: velocity returns to zero.
The proportional term balances the added force.

$$K_p e = 0.60\,\mathrm{N}$$

$$e = 0.60\,\mathrm{N} \div (3\ \mathrm{N/m}) = 0.20\,\mathrm{m}$$

$$h = 2.00 - 0.20 = 1.80\,\mathrm{m}$$

Proportional compensation requires nonzero error.
How can we keep compensation and return to the target?

5.1.2 Importance of Low-Level Control: Integral Compensation

How Does Integral Action Build Compensation?

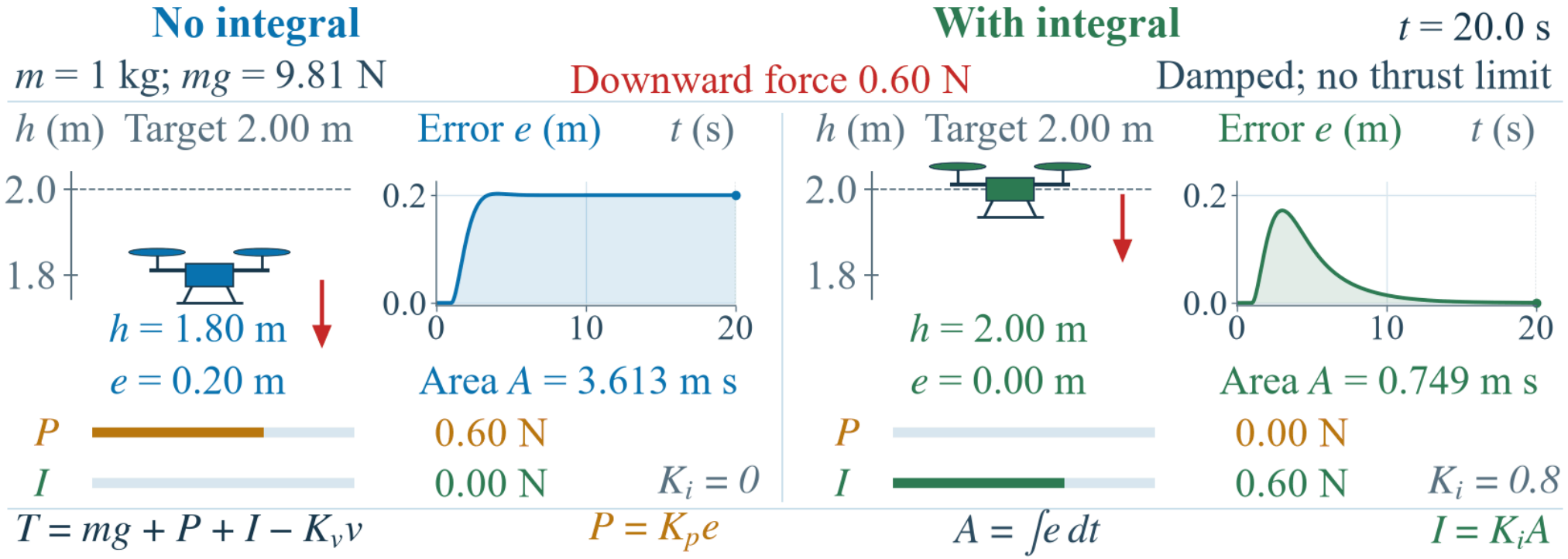
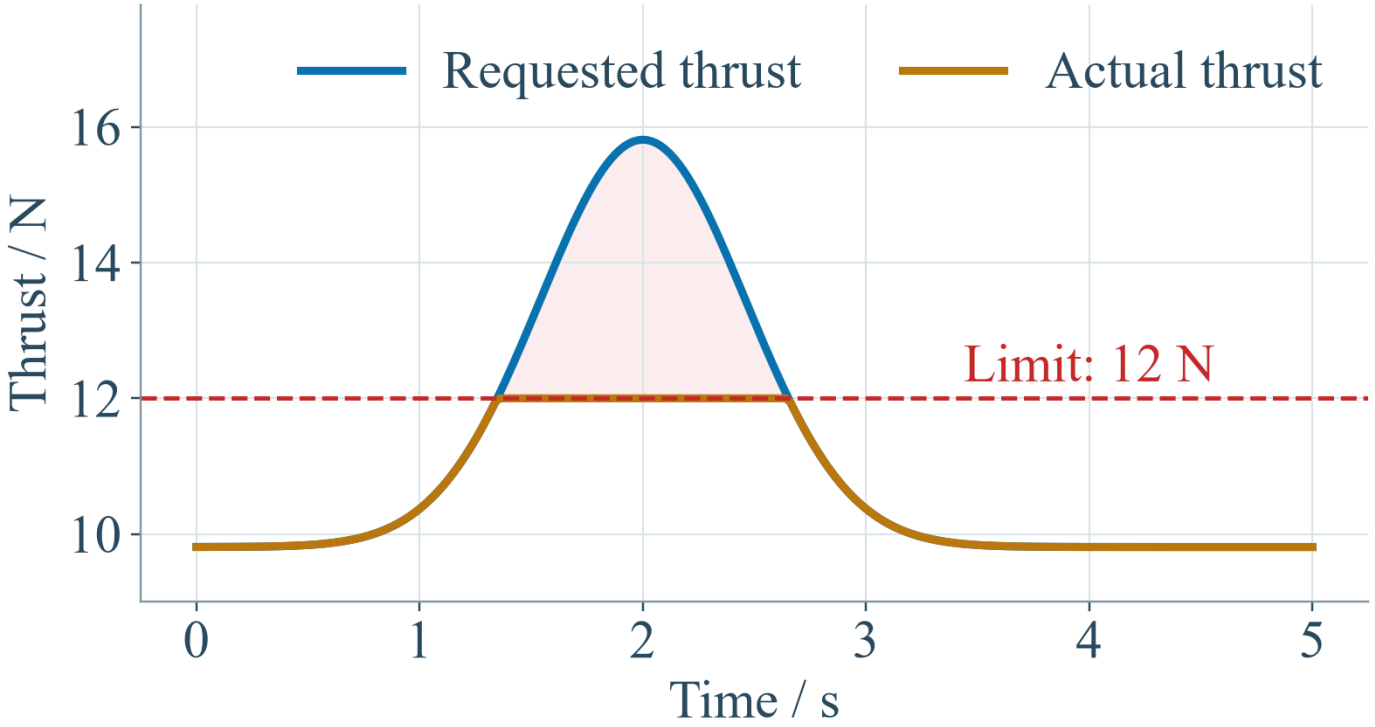


Figure: as error approaches zero, the integral stops growing and retains compensation (persistent-force numerical example)

5.1.2 Importance of Low-Level Control: Actuator Limits

Actuators Cannot Always Deliver the Request



- **Requested value:** the action the control algorithm seeks.
- **Actual value:** constrained by current, speed, travel, and other limits.
- **Design implication:** anti-windup is needed if integration drives further into saturation.

Figure: a thrust shortfall appears when the request exceeds the limit

Chapter Outline

Filtering, State Estimation, and Low-Level Control: Chapter Roadmap

Chapter 5 Filtering, State Estimation, and Low-Level Control

5.1 Filtering and Low-Level Control

5.2 Sensor Calibration and Estimation

5.3 Controller Design and Allocation

5.4 Control Performance and Tuning

5.5 PX4 Architecture and Interfaces

5.6 Control Development and Testing

5.7 Other Vehicles and Advanced Cases

5.8 Exercises and Classroom Discussion

4 Units in This Section

5.1.1 Control Principles of Unmanned Systems

Textbook: Sec. 5.1.1.1

5.1.2 Importance of Low-Level Control

Textbook: Sec. 5.1.1.2

5.1.3 Importance of Filtering and State Estimation

Textbook: Sec. 5.1.1.3

5.1.4 Low-Level Control Development Process

Textbook: Secs. 5.1.1.4 and 5.1.2

5.1.3 Importance of Filtering and State Estimation: States and Observations

True States, Measurements, and Estimates Differ

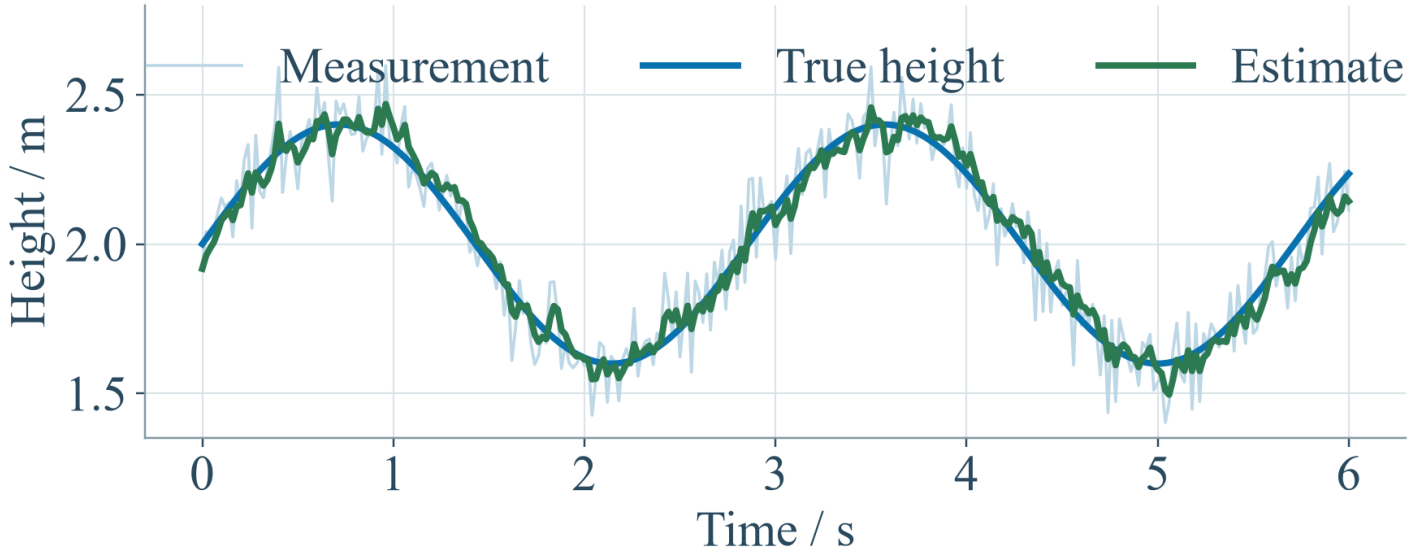


Figure: true altitude, noisy measurements, first-order recursive estimate (teaching model supplies truth).

True state	Measurement	Estimate
The object's actual motion state	Sensor information containing errors	An inference from the model and observations

5.1.3 Importance of Filtering and State Estimation: Complementary Information

Different Sensors Reveal Different Motion Clues

Information source	What it provides	Main limitations
Inertial measurement unit (IMU)	Gyroscope: angular rate; accelerometer: specific force	Integration accumulates error; acceleration affects the gravity reference
Global navigation satellite system (GNSS)	Position, velocity, and a long-term reference	Obstruction and interference can degrade information
Cameras and visual estimation	Image feature changes aid localization	Depend on lighting, texture, and motion conditions
Lidar	Range and geometry aid localization	Rain, snow, fog, and geometric degeneracy affect results

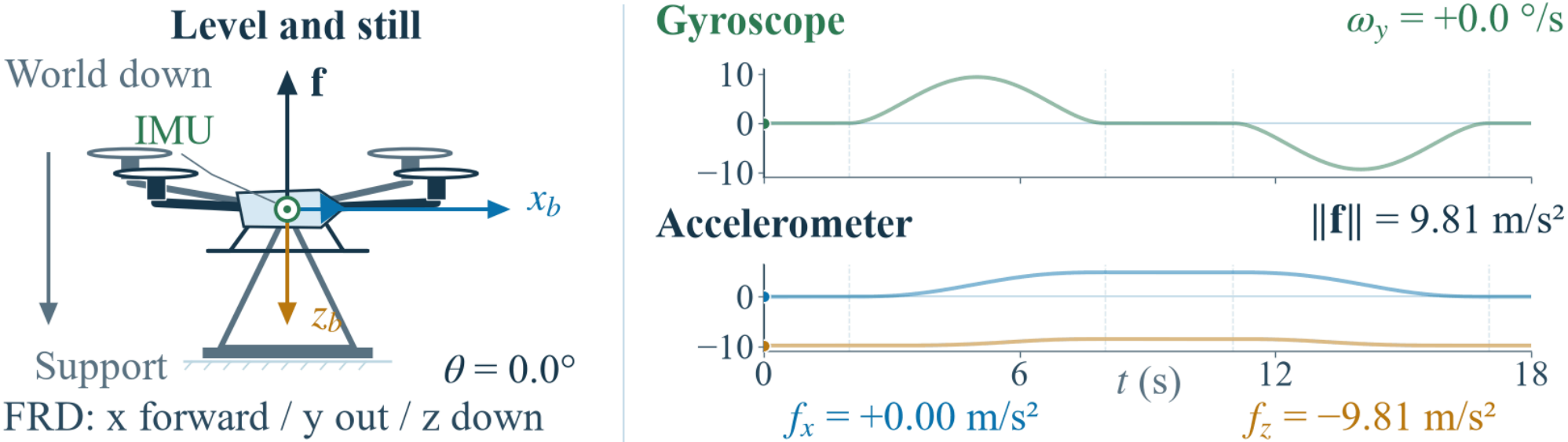
Specific force is nongravitational force per unit mass: $\boldsymbol{f} = \boldsymbol{a} - \boldsymbol{g}$ (in the same coordinate frame).

Align time and coordinates before fusion; gravity direction alone cannot determine yaw.

5.1.3 Importance of Filtering and State Estimation: IMU Readings

IMU Readings: Rest, Rotation, and Stopping

After stopping: zero angular rate, nonzero specific force



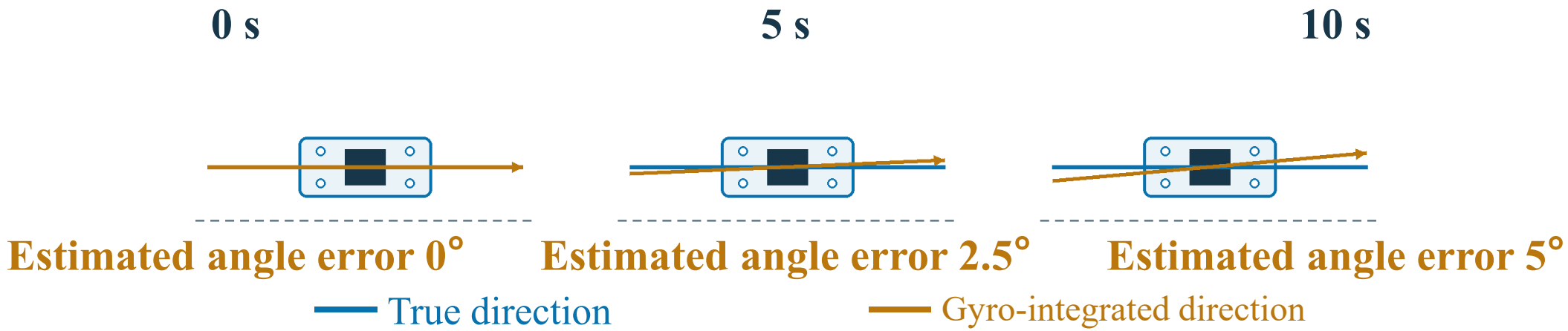
Supported; slow pitch about IMU; translation and lever-arm effects negligible.

Figure: during slow supported rotation, angular rate returns to zero after stopping; specific-force components still reflect tilt

5.1.3 Importance of Filtering and State Estimation: Drift and Complementarity

Gyro Drift: How Can Other Observations Correct It?

The vehicle stays level; a residual gyro bias of 0.5°/s makes the integrated angle drift continuously.



$$\Delta\theta = b_\omega \Delta t = 0.5^\circ/\mathrm{s} \times 10\,\mathrm{s} = 5^\circ$$

Gyro integration tracks short-term changes; a quasi-static gravity reference corrects slow roll and pitch drift.

Gravity cannot determine yaw. With acceleration, specific force is not gravity alone.

5.1.3 Importance of Filtering and State Estimation: Smoothing and Delay

Smoother Feedback Is Not Always Better for Control

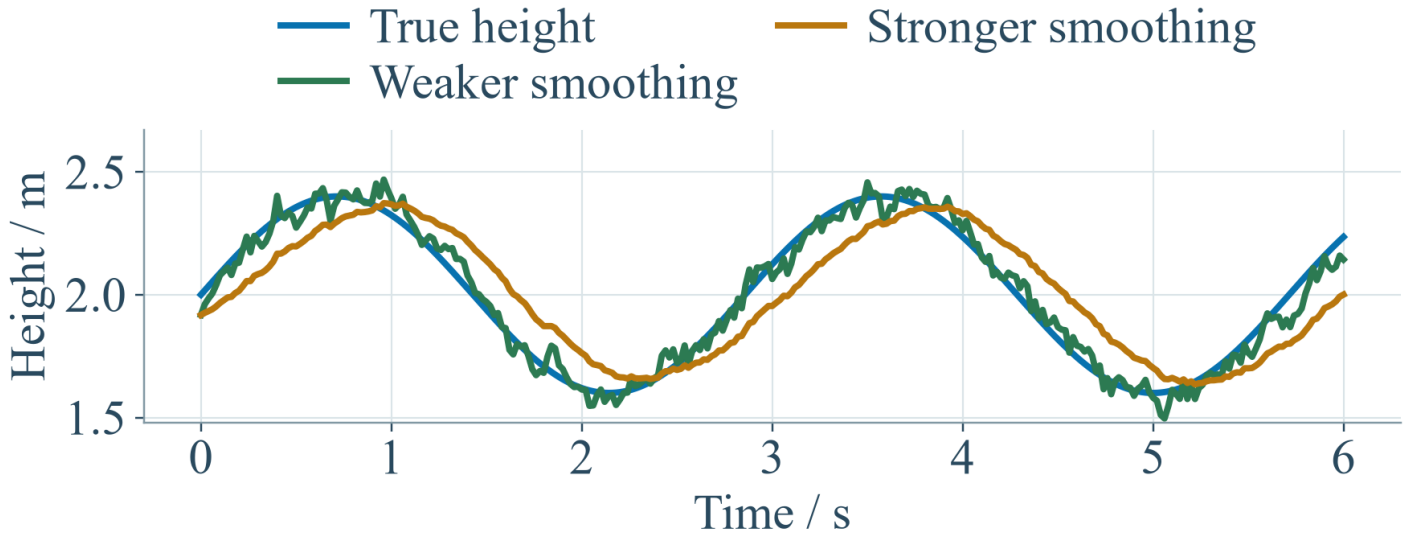


Figure: same noisy data; only the new-measurement weight changes (teaching example).

Result: stronger smoothing peaks about 0.3 s later and 0.03 m lower; weaker smoothing responds sooner but retains more jitter.

Estimation must provide usable states promptly, as well as smooth them.

5.1.3 Importance of Filtering and State Estimation: Prediction and Correction

Predict Position Before the Next Measurement

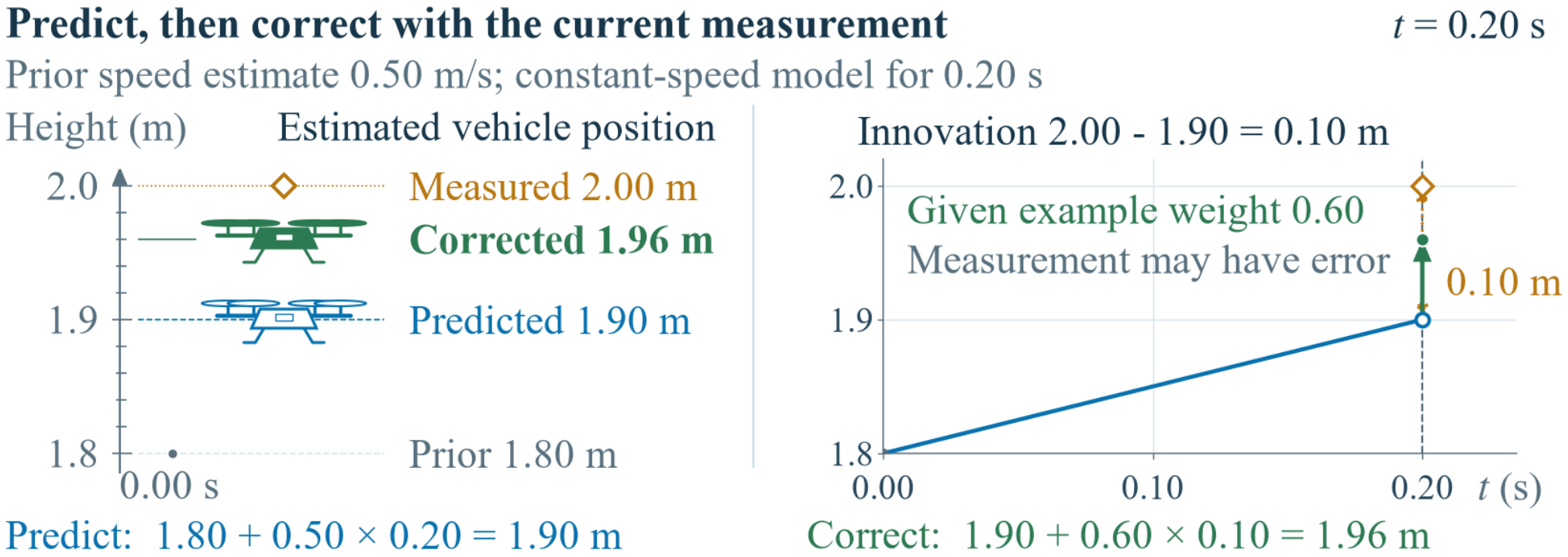


Figure: predict the previous estimate to the current time, then correct it using the new observation's innovation

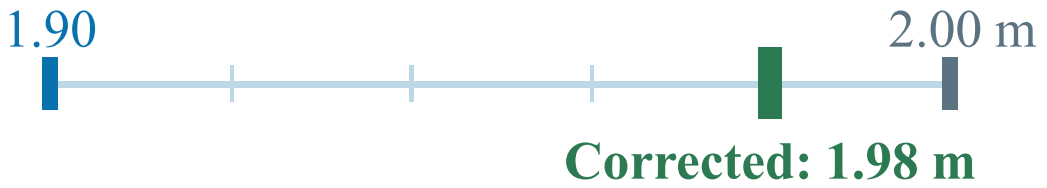
5.1.3 Importance of Filtering and State Estimation: Observation Reliability

How Should Estimates Adapt to Observation Quality?

At the same time: predicted altitude 1.90 m, new observation 2.00 m; difference 0.10 m.

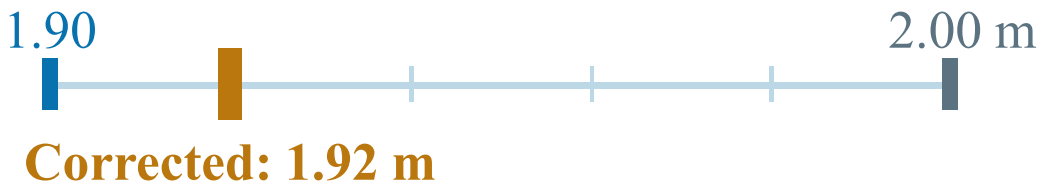
More reliable observation

$$1.90 + 0.8 \times 0.10 = 1.98\,\text{m}$$



Noisier observation

$$1.90 + 0.2 \times 0.10 = 1.92\,\text{m}$$



0.8 and 0.2 are illustrative weights; a Kalman filter computes weights from prediction and observation uncertainty.

Keep predicting without new observations; do not reuse the previous measurement as new information.

Chapter Outline

Filtering, State Estimation, and Low-Level Control: Chapter Roadmap

Chapter 5 Filtering, State Estimation, and Low-Level Control

5.1 Filtering and Low-Level Control

5.2 Sensor Calibration and Estimation

5.3 Controller Design and Allocation

5.4 Control Performance and Tuning

5.5 PX4 Architecture and Interfaces

5.6 Control Development and Testing

5.7 Other Vehicles and Advanced Cases

5.8 Exercises and Classroom Discussion

4 Units in This Section

5.1.1 Control Principles of Unmanned Systems

Textbook: Sec. 5.1.1.1

5.1.2 Importance of Low-Level Control

Textbook: Sec. 5.1.1.2

5.1.3 Importance of Filtering and State Estimation

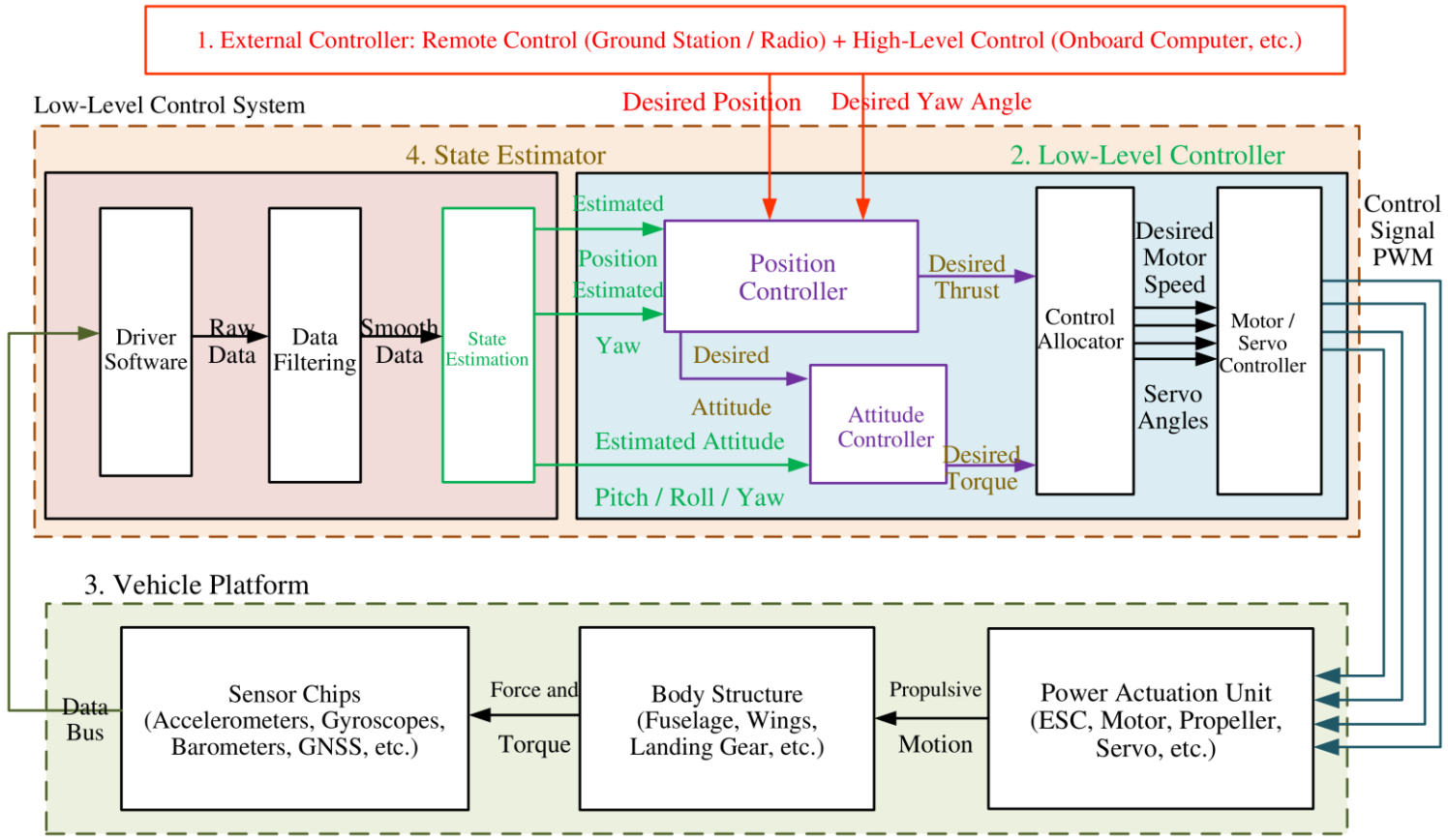
Textbook: Sec. 5.1.1.3

5.1.4 Low-Level Control Development Process

Textbook: Secs. 5.1.1.4 and 5.1.2

5.1.4 Low-Level Control Development Process: Low-Level Control Framework

Fig. 5.2: Four Groups of Control System Modules



1. External controller
Position and yaw targets

2. Low-level controller
Pose control and allocation

3. Vehicle body system
Actuation units, frame, sensors

4. Filtering and state estimation
Estimates state from sensors

Fig. 5.2 Low-Level Control System Framework for Unmanned Systems

5.1.4 Low-Level Control Development Process: Observation and Action Chains

Fig. 5.2: Observation and Action Chains

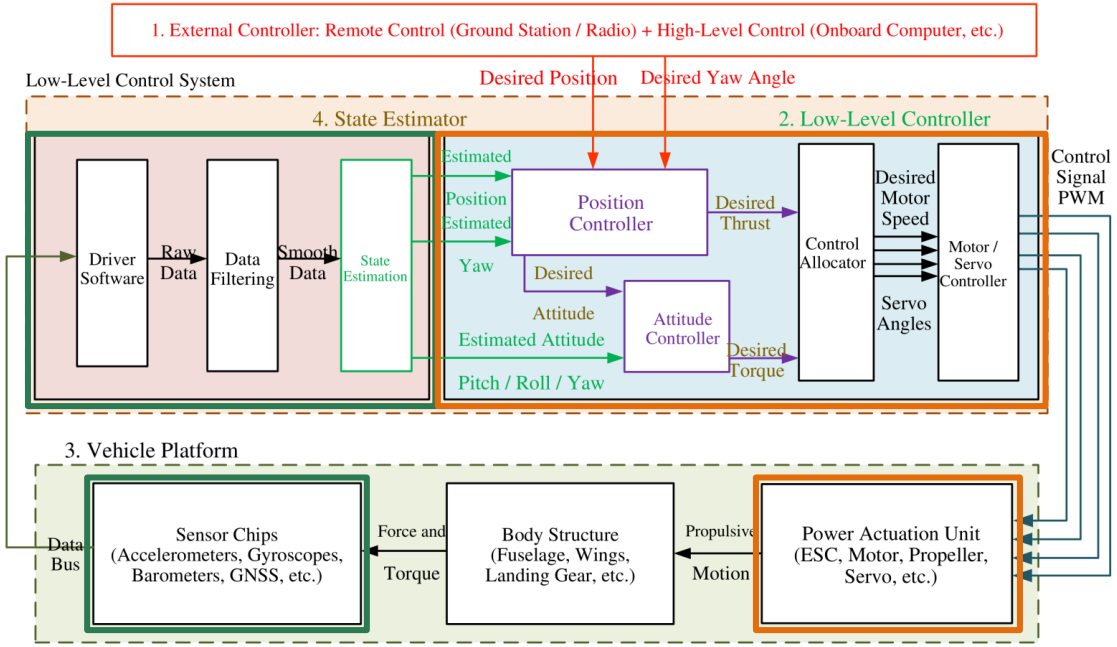


Fig. 5.2: green observation chain; orange action chain

PWM (pulse-width modulation) is one actuator signal format in the figure.

Observation chain (green): motion to state feedback

Vehicle motion → sensors → drivers and filtering → state estimation → position and attitude feedback

Action chain (orange): targets to physical action

Targets and feedback → position and attitude control → total thrust and torque demands → allocation and drivers

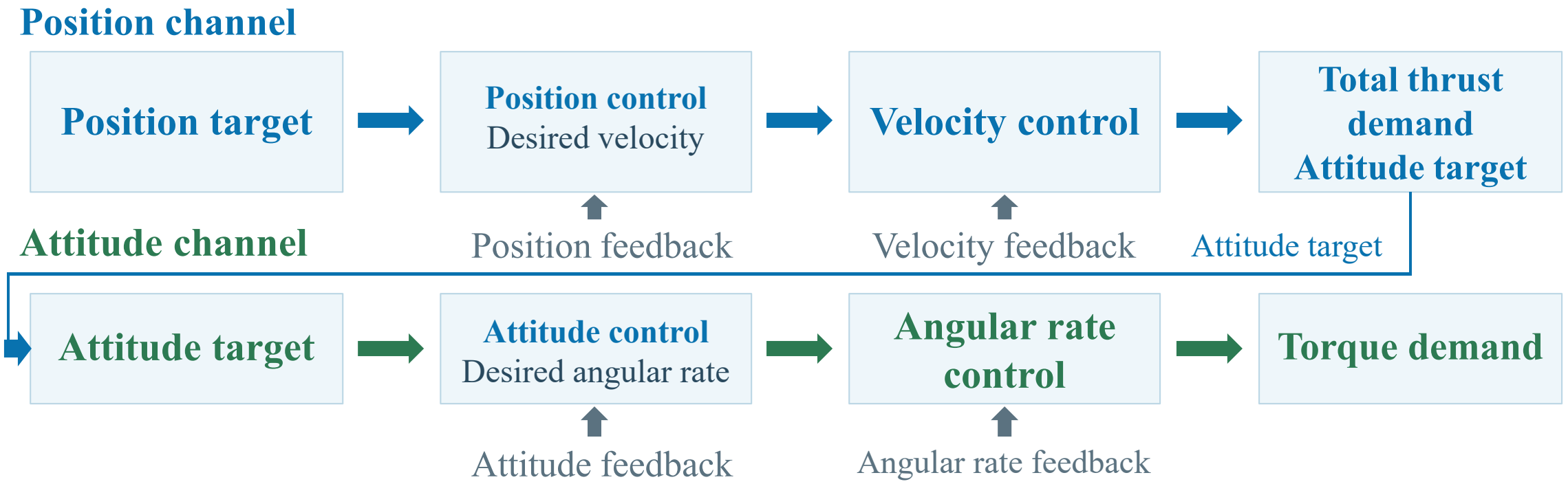
→ motors, servos, and other actuators

→ actual forces and torques → new vehicle motion

Modules run continuously in parallel, each at its own rate.

5.1.4 Low-Level Control Development Process: Cascaded Loops

Motion Targets Pass Through Cascaded Loops



- **Inner and outer loops:** a much faster inner loop lets the outer loop treat its command as tracked.
- **Control allocation:** jointly map total thrust and torque demands to individual actuator commands.

5.1.4 Low-Level Control Development Process: Control Allocation

How Do the Same Motors Lift and Rotate?

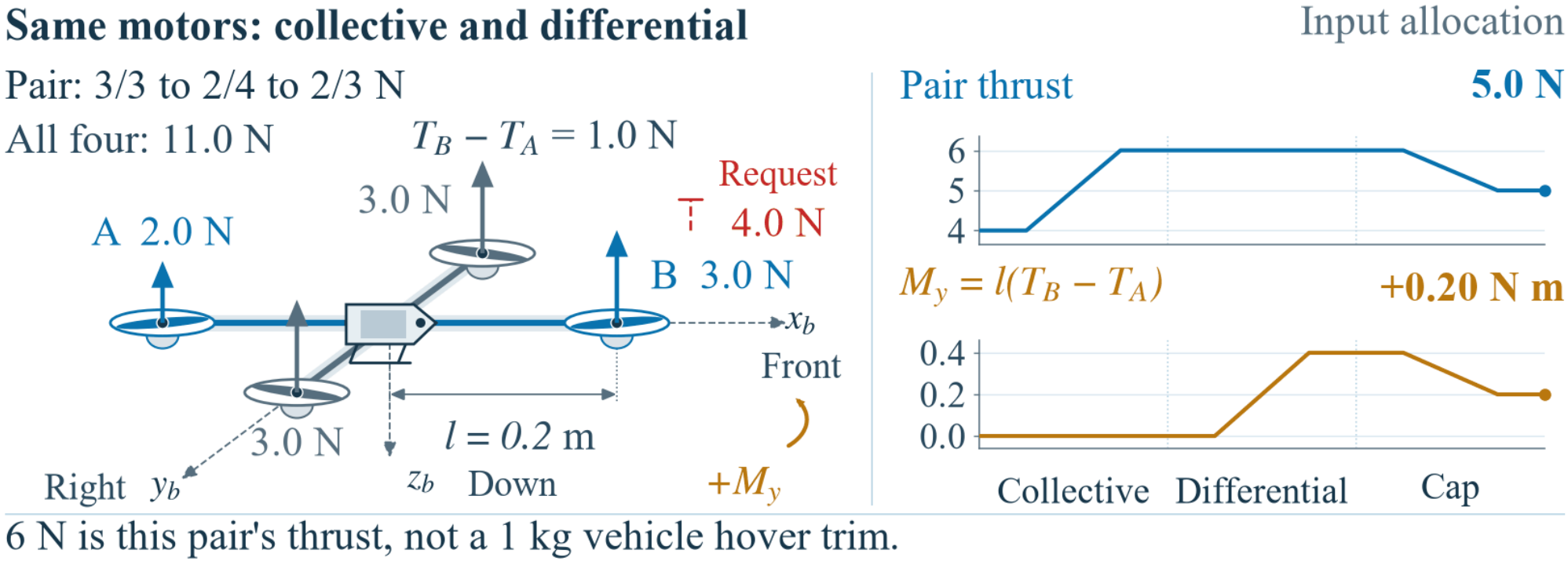


Figure: effects of collective thrust, opposite-arm differential thrust, and one-sided saturation on force and torque

5.1.4 Low-Level Control Development Process: Operating Points and Linearization

Design and Validation Need Different Models

Stage	Model choice	Main question
Understand the behavior	1D or single-channel model	How do force, velocity, and error interact?
Local design	Linear model near operating point (superposition and homogeneity)	How should gains be chosen, and is the local response stable?
Broader validation	Full nonlinear model with disturbances/saturation	Does it still work with large commands and model differences?

- Operating point: level hover at rest, no external disturbance, zero net force/torque, $T_0 = mg$.
- **Local linearization:** first-order approximation about an operating point (derivation: 4.4.2).
- **Common baseline:** match mass, inertia, units, coordinates, and initial conditions.
- **Scope:** validate local small-perturbation conclusions with large commands and saturation.

5.1.4 Low-Level Control Development Process: Staged Validation

Which Real Components Does Each Stage Validate?

Model Simulation	Software-in-the-loop SITL; textbook SIL	Hardware-in-the-loop HITL; textbook HIL	Real-Vehicle Experiments
Added component Control algorithm computation	**Added component** Firmware process and software interfaces	**Added component** Target autopilot hardware	**Added components** Vehicle, actuators, and real environment
Main checks Control direction, response, and saturation	**Main checks** Message links, scheduling, and logic	**Main checks** Hardware links, real-time behavior, and resources	**Main checks** Unmodeled differences and constrained operation

Model simulation, SITL, and HITL: vehicle motion comes from the simulation model

Real vehicle and environment

At each step, retain the preceding baseline to attribute new differences to the added components.

5.1.4 Low-Level Control Development Process: RflySim Development Support

Choose an Example for the Current Problem

Common parent directory: [安装目录]/RflySimAPIs/5.RflySimFlyCtrl/

Learning task	Actual directory	Focus (slide section)
Sensor calibration	1.BasicExps/e3-SensorCalib	Bias, scale, and calibration residuals (5.2)
Filtering and state estimation	1.BasicExps/e4-FilterDesign	Attitude estimation, noise/delay (5.2)
Attitude control	1.BasicExps/e5-AttitudeCtrl	Inner-loop structure and response (5.4)
Development and code generation	0.ApiExps/3.DesignExps	Model integration, simulation, and code generation (5.5–5.6)

Read each Readme.pdf first; 5.5–5.6 cover partial replacement, module switching, data interfaces, and logs.

5.1.4 Low-Level Control Development Process: Data and Results

Compare Attitude Estimates from the Same Log

Continue in the earlier e4-FilterDesign directory:
e4.1/Attitude_estimator0_fly.m reads logdata.mat in the same directory

Input: timestamped IMU records

timestamp; gx, gy; ax, ay, az
Intervals: convert microseconds to seconds.

Processing: same data, three estimates

gyro: gyroscope integration
acc: gravity reference
cf: complementary combination

See Section 5.2 for execution and parameter comparisons.

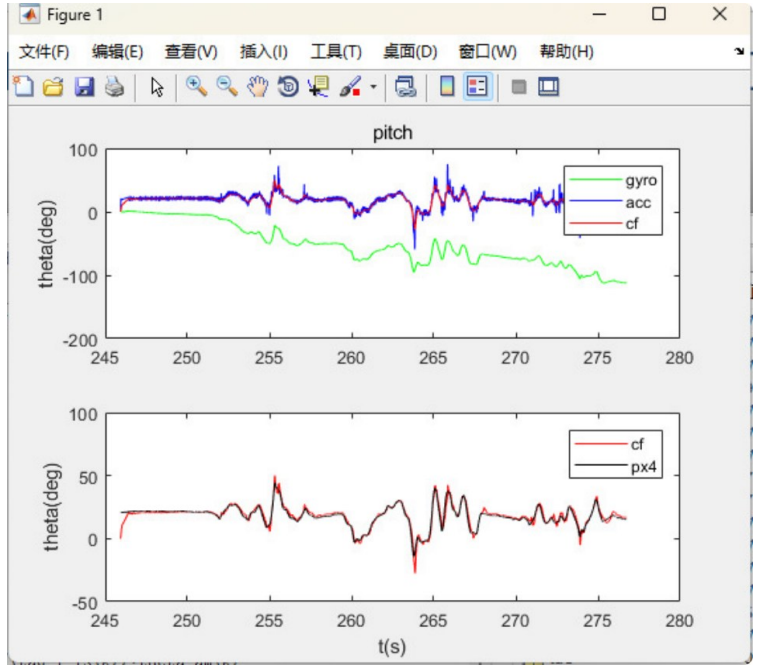


Figure: example pitch results; PX4 is also an estimate.

Classroom Discussion

Which Loop Holds Evidence of a Trajectory Error?

Observed symptom	Check these quantities first	Interpretation
Altitude stays below target	Target, estimated altitude, thrust request	Distinguish bias, payload, and output limits
Measurements show strong jitter	Raw data, estimates, timestamps	Check noise, filtering, and sampling timing
Control output stays at its limit	Requests, actuator outputs, saturation status	Check actuator headroom before gains

Follow-up: what additional evidence would distinguish two possible causes?

- Reference points: an external altitude reference helps assess estimation bias; outputs and saturation help assess headroom; verify actual forces with simulation truth or independent measurements.

Classroom Exercise

Calculate an Altitude Correction with New Data

Given: mass 1.5 kg, altitude target 3.0 m, estimate 2.85 m, velocity 0, proportional gain 4 N/m.

(1) Find the error e, correction ΔT, and total thrust T.
(2) With a total thrust limit of 15.0 N, can the actuators deliver the full correction?
(3) If the estimate is always 0.10 m too low and finally settles at 3.0 m, what is the true altitude?

Reference Points

(1) $e = 0.15$ m; $\Delta T = 4 \times 0.15 = 0.60$ N; $T = 14.715 + 0.60 \approx 15.32$ N.
(2) No: only 15.0 N is available, giving about 0.29 N of correction and a slower climb back.
(3) When the estimate settles at 3.0 m, true altitude is about 3.10 m: estimation bias becomes position error.

Section Summary

Estimate the State, Then Deliver the Action

- **Control:** adjust inputs to change motion; equal positions with different velocities require different actions.
- **Estimation:** combine models and observations into state feedback; consider both error and timely response.
- **Engineering limits:** actuators, sampling timing, models, and the environment jointly determine achievable performance.
- **Development:** implement cascaded layers, design with local linearization, then validate progressively through real-vehicle tests.
- **Common pitfalls:** measurements are not truth, feedback alone does not ensure stability, and requests may exceed actuator capability.

Reliable state feedback is the starting point for the control design that follows.