## Chapter Outline

# Exercises and Classroom Discussion

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

**5.8.1 Concepts and Methods**

Textbook: Sec. 5.4, Exercises 1–3

**5.8.2 Filtering and Control Calculations**

Textbook: Sec. 5.4, Exercises 4, 5, 7, 8

**5.8.3 System Integration and Validation**

Textbook: Sec. 5.4, Exercises 6, 9, 10

**5.8.4 Integrated Design and Classroom Discussion**

Baseline, one change, and report review

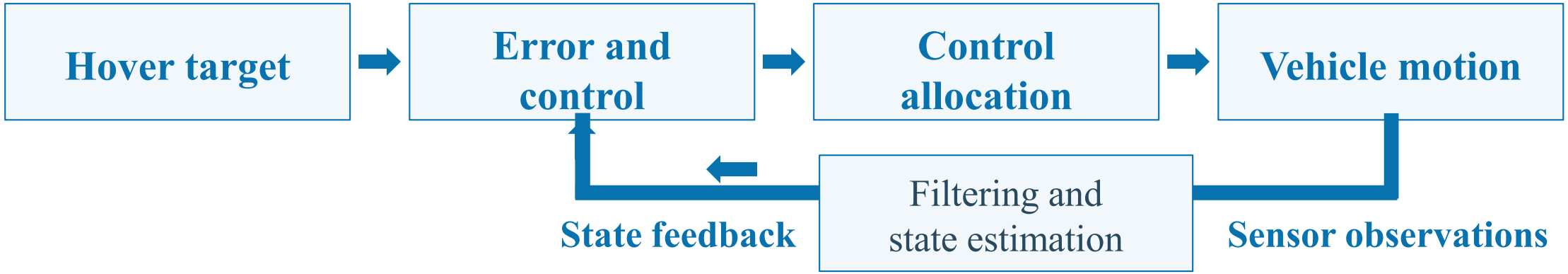
### 5.8.1 Concepts and Methods: Original Questions and Review

## Exercises 1–3: Read the Questions, Then Review

| Ex. | Questions: Textbook, Sec. 5.4 | Review Slides |
|---|---|---|
| **Ex. 1** | Briefly describe the roles of state estimation and low-level control in unmanned aerial vehicle systems. | Slide Section 5.1 (5.1.2–5.1.4) |
| **Ex. 2** | Describe the characteristics and suitable applications of complementary filters, Kalman filters, and extended Kalman filters. | Slide Section 5.2 (5.2.2–5.2.5) |
| **Ex. 3** | Describe the physical meaning of linearization and trimming, and the typical operating points in unmanned aerial vehicle control design. | Slide Section 5.1 (5.1.4)<br>Slide Section 4.4 |

### 5.8.1 Concepts and Methods: Module Roles

# Ex. 1: What Do Estimation and Control Do in Hover?




- **Filtering and estimation:** fuse model and observations to estimate state and confidence.
- **Low-level control:** use targets and feedback to correct motion within actuator limits.
- **Reference chain:** distinguish targets, observations, estimates, errors, requests and achieved values; close the loop.

### 5.8.1 Concepts and Methods: Filter Selection

# Ex. 2: Match the Method to the Model and Data

| Method | Features and assumptions | Applications and limitations |
|---|---|---|
| **Complementary filter** | Short-term gyro information + long-term gravity reference; usually fixed fusion parameters | Lightweight attitude estimation; dynamic acceleration contaminates the gravity reference |
| **Kalman filter (KF)** | Linear model; prediction and observation covariances determine update weights | Linear position/velocity subsystems; check model and noise assumptions |
| **Extended Kalman filter (EKF)** | Nonlinear state and observation models; Jacobians approximate covariance propagation locally | Attitude/navigation fusion; initialization and linearization error matter |

Reference: near-level motion and limited compute → complementary filter; linear model with known noise → KF; nonlinear → EKF.

### 5.8.1 Concepts and Methods: Trim and Linearization

# Ex. 3: State the Operating Condition for Small Perturbations

- **Trim:** in a selected steady condition, velocity and attitude remain constant; position may change during constant-speed motion.
- **Hover:** no external disturbance, near-zero velocity and angular rate; thrust balances weight.
- **Straight, level flight:** nonzero airspeed and constant altitude, attitude, and control inputs.

$$\Delta\dot{\mathbf{x}} = A\Delta\mathbf{x} + B\Delta\mathbf{u}$$

$$A = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_0, \quad B = \left.\frac{\partial f}{\partial \mathbf{u}}\right|_0$$

Take partial derivatives of $\dot{x} = f(x, u)$ at trim (subscript 0); the local model is valid only nearby.

## Chapter Outline

# Exercises and Classroom Discussion

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

**5.8.1 Concepts and Methods**

Textbook: Sec. 5.4, Exercises 1–3

**5.8.2 Filtering and Control Calculations**

Textbook: Sec. 5.4, Exercises 4, 5, 7, 8

**5.8.3 System Integration and Validation**

Textbook: Sec. 5.4, Exercises 6, 9, 10

**5.8.4 Integrated Design and Classroom Discussion**

Baseline, one change, and report review

### 5.8.2 Filtering and Control Calculations: Original Questions and Review

# Exercises 4, 5, 7, 8: Read the Questions, Then Review

| Ex. | Questions: Textbook, Sec. 5.4 | Review Slides |
|---|---|---|
| Ex. 4 | Describe the engineering significance of saturated proportional-integral-derivative (PID) control and its anti-windup mechanism. | Slide Section 5.3 (5.3.2) |
| Ex. 5 | Analyze the typical effects of proportional (P), integral (I), and derivative (D) gains on stability, response speed, and accuracy. | Slide Section 5.4 (5.4.1–5.4.2) |
| Ex. 7 | Implement a complementary filter for roll and pitch estimation, and discuss parameter selection and its response to measurement noise and disturbances. | Slide Section 5.2 (5.2.2) |
| Ex. 8 | Starting from a six-degree-of-freedom nonlinear model, explain how to linearize it and derive an initial control law using a linear quadratic regulator (LQR). | Slide Section 5.1 (5.1.4)<br>Slide Section 5.4 (5.4.3) |

### 5.8.2 Filtering and Control Calculations: Anti-Windup

# Ex. 4: How Should Integration Update Above the Limit?

Normalized teaching quantities: $K_p = 2,\ K_i = 1\ \mathrm{s}^{-1},\ e = 1,\ I = 0.5,\ \Delta t = 0.1\ \mathrm{s}$

$I$ is the integral output contribution including $K_i$; limits $[-1,1]$; back-calculation gain $K_b = 2\ \mathrm{s}^{-1}$.

$$u_{\mathrm{req}} = K_p e + I = 2.5, \quad u_{\mathrm{act}} = 1, \quad I^+ = I + \Delta t\,[K_i e + K_b(u_{\mathrm{act}} - u_{\mathrm{req}})]$$

| Mechanism | Update this step | Result |
|---|---|---|
| **Integrate normally** | $I^+ = 0.5 + 1 \times 1 \times 0.1$ | 0.60 |
| **Conditional integration** | At upper saturation with positive error: freeze this step. | 0.50 |
| **Back-calculation** | $I^+ = 0.5 + 0.1[1 + 2(1 - 2.5)]$ | 0.30 |

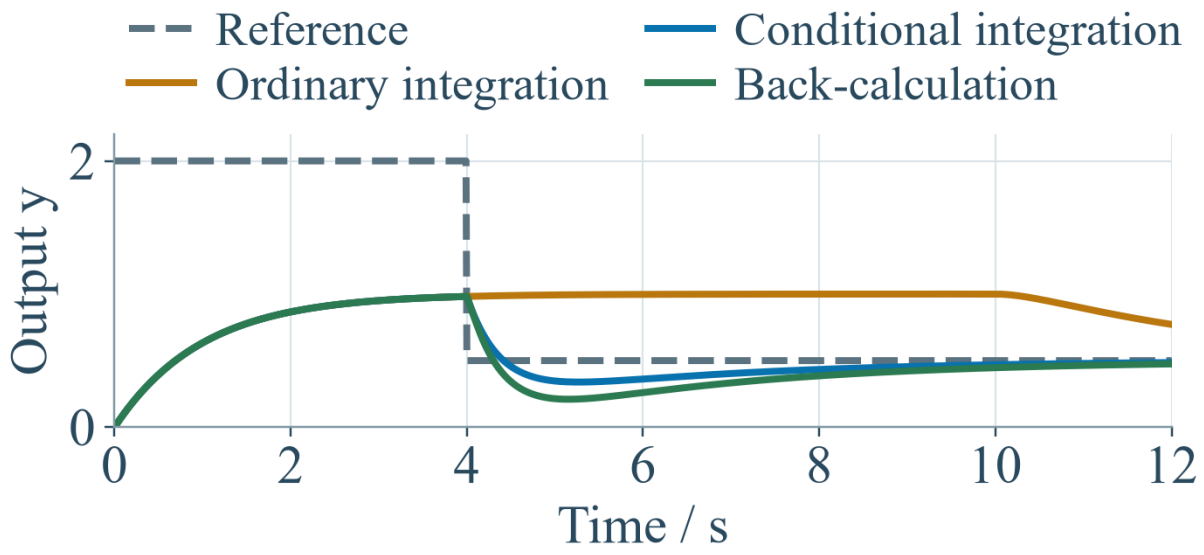
Purpose: avoid integral buildup and shorten recovery after saturation or a target decrease.

### 5.8.2 Filtering and Control Calculations: Recovery from Saturation

# Ex. 4 Answer: Recovery After the Target Decreases

Initial conditions: $y(0) = I(0) = 0$; $I$ is the integral output contribution (including $K_i$); target drops from 2 to 0.5 at 4 s.

$$\dot{y} = -y + u, \quad |u| \le 1, \quad K_p = 2,\ K_i = 1\ \mathrm{s}^{-1},\ K_b = 2\ \mathrm{s}^{-1}$$




| Method | $I(4\,\mathrm{s})$ | $y(6\,\mathrm{s})$ |
|---|---|---|
| **Plain** | 4.982 | 0.998 |
| **Conditional I** | 0.000 | 0.362 |
| **Back-calc.** | −0.553 | 0.260 |

Forward Euler, $\Delta t = 0.01\,\mathrm{s}$

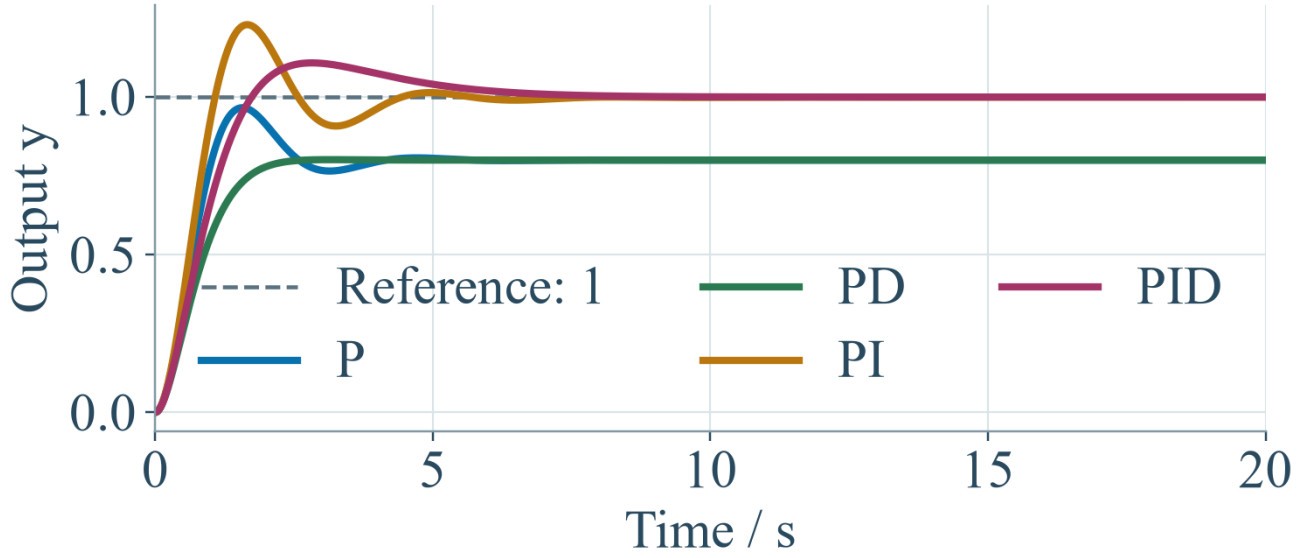
Result: plain I reaches 4.982 and leaves upper saturation near 10 s; none settles within 0.49–0.51.

### 5.8.2 Filtering and Control Calculations: Reading Parameter Effects

# Ex. 5: How Do I and D Affect Stability, Speed, and Accuracy?

$$\ddot{y} + 2\dot{y} + y = u, \quad u = K_p(1 - y) + K_i \int (1 - y)dt - K_d\dot{y}$$

Unit step from t=0; $K_p = 4$; proportional-integral (PI): $K_i = 2$, proportional-derivative (PD): $K_d = 2$, PID: both; zero initial states.




Teaching example

Here, P/PD settle at 0.8 (error 0.2); PI/PID errors approach zero, and D reduces overshoot.

### 5.8.2 Filtering and Control Calculations: Common-Data Metrics

# Ex. 5 Answer: Agree on Final Values and Thresholds

Use each response's own final value for 10%→90% rise, the 2% band, and the overshoot denominator.

| Gains (P,I,D) | Rise / s | Overshoot (%) | 2% settling / s | Steady-state error |
|---|---|---|---|---|
| **P (4,0,0)** | 0.689 | 20.79 | 3.736 | 0.20 |
| **PD (4,0,2)** | 1.278 | 0.19 | 2.075 | 0.20 |
| **PI (4,2,0)** | 0.727 | 22.96 | 4.172 | 0.00 |
| **PID (4,2,2)** | 1.173 | 10.84 | 5.990 | 0.00 |

Final values: P/PD 0.8; PI/PID 1.
All input peaks stay below the common threshold 5 (Slide Section 5.4).

### 5.8.2 Filtering and Control Calculations: PID Performance

# Ex. 5 Answer: State the Costs of Parameter Trends

| Increase in | Typical effect | Conditions and costs |
|---|---|---|
| **Proportional gain** | Stronger correction, often faster response;<br>constant error may decrease | Overshoot, oscillation, and input demand<br>may increase |
| **Integral gain** | Accumulates error, removes constant offset | More prone to saturation; recovery time<br>and stability may worsen |
| **Derivative action** | Adds damping and reduces overshoot | Amplifies noise; needs filtering; account for added delay |

Stability: stability and decay; speed: rise/settling time; accuracy: steady-state/tracking error.

### 5.8.2 Filtering and Control Calculations: Complementary Filter Design

# Ex. 7: Roll and Pitch from Six-Axis Measurements

Forward-right-down (FRD) body frame; angles rad, angular rates rad/s, specific force m/s².

$$\theta_a = \arcsin\frac{f_x}{\|\mathbf{f}\|}, \quad \phi_a = -\arcsin\frac{f_y}{\|\mathbf{f}\|\cos\theta_a}$$

$$\hat{\theta}_k = \alpha(\hat{\theta}_{k-1} + q_m\Delta t) + (1-\alpha)\theta_a$$

$$\hat{\phi}_k = \alpha(\hat{\phi}_{k-1} + p_m\Delta t) + (1-\alpha)\phi_a, \quad \alpha = \frac{\tau_c}{\tau_c + \Delta t}$$

Applies near level when gravity dominates; body angular rates generally differ from Euler-angle derivatives.

Folder: [安装目录]/RflySimAPIs/5.RflySimFlyCtrl/1.BasicExps/e4-FilterDesign/e4.1/
Script: Attitude_estimator0.m reads e4_A.bin and runs the filter in its loop.
Function form: Attitude_cf.m in the same folder (interface: Slide Section 5.2.2).
Run and interpret: e4.1/e4.2 examples in Slide Section 5.2.

### 5.8.2 Filtering and Control Calculations: Complementary Filter Calculation

# Ex. 7 Answer: Include Time Step and Fusion Weight

$$\Delta t = 0.1\ \text{s},\ \tau_c = 0.9\ \text{s},\ \alpha = 0.9$$
$$\hat{\theta}_{k-1} = 10^\circ,\ q_m = 2^\circ/\text{s},\ \theta_a = 9^\circ$$
$$\hat{\phi}_{k-1} = 5^\circ,\ p_m = 1^\circ/\text{s},\ \phi_a = 4^\circ$$

$$\hat{\theta}_k = 0.9(10 + 2 \times 0.1) + 0.1 \times 9 = 10.08^\circ$$

$$\hat{\phi}_k = 0.9(5 + 1 \times 0.1) + 0.1 \times 4 = 4.99^\circ$$

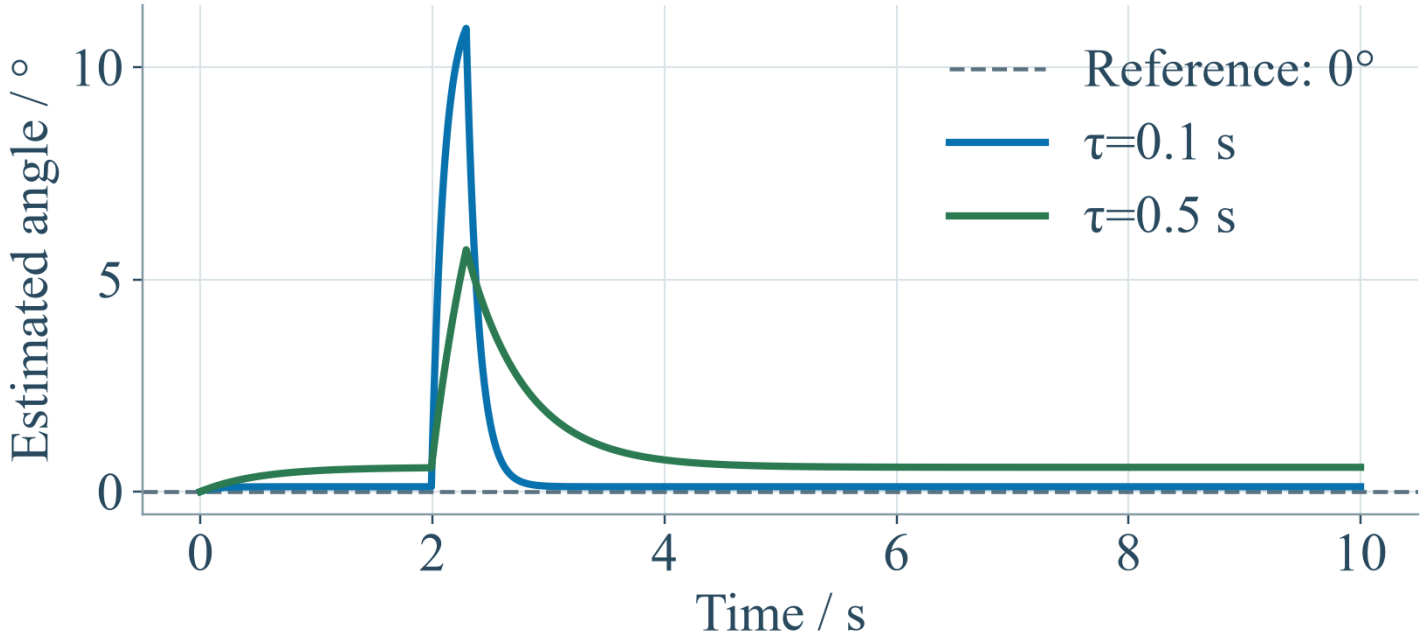
Code interface: Attitude_cf(dt, z, phi_cf_k, theta_cf_k, tao); dt=$\Delta t$, tao=$\tau_c$ (s).
$\mathbf{z} = [p_\text{m}, q_\text{m}, r_\text{m}, f_x, f_y, f_z]^\text{T}$ (rad/s, m/s²); angle inputs and outputs are in rad.

Result: pitch 10.08°, roll 4.99°; code uses rad, with 10.08°=0.175929 rad.

### 5.8.2 Filtering and Control Calculations: Filter Parameters and Disturbances

# Ex. 7 Answer: Less Pulse Response, More Bias Error

Synthetic pitch: true angle 0; step 0.01 s; gyro bias 0.02 rad/s; reference disturbed by 0.2 rad during 2–2.3 s.




| $\tau_c$ / s | Residual / ° | Peak / ° |
|---|---|---|
| **0.1** | 0.115 | 10.917 |
| **0.5** | 0.573 | 5.700 |

Teaching example

At 10 s, residual ≈ bias × $\tau_c$ (0.115°/0.573°); larger $\tau_c$ reduces the pulse peak.

### 5.8.2 Filtering and Control Calculations: Six Degrees of Freedom to LQR (Undisturbed Hover)

# Ex. 8: Choose the Operating Point Before Local Feedback

Six-degree-of-freedom model → Linearize at trim → Normalize scales → Initial LQR law

$$\Delta\dot{\mathbf{x}} = A\Delta\mathbf{x} + B\Delta\mathbf{u}, \quad \Delta\mathbf{u} = -K\Delta\mathbf{x}$$

$$J = \int_0^\infty (\Delta\mathbf{x}^\mathsf{T} Q \Delta\mathbf{x} + \Delta\mathbf{u}^\mathsf{T} R \Delta\mathbf{u})\, dt$$

Hover trim: $T = mg,\ \phi = \theta = 0,\ \boldsymbol{\omega} = \mathbf{0}$; at small angles, $\Delta\ddot{p}_\mathrm{N} \approx -g\,\Delta\theta$ (yaw angle 0).

States available for feedback; $(A, B)$ stabilizable, $(A, Q^{1/2})$ detectable; $Q \succeq 0,\ R \succ 0$.
Detectable: modes unpenalized by the cost must be stable.
Check coupling and constraints for the full vehicle.

### 5.8.2 Filtering and Control Calculations: Scalar LQR Baseline (Slide Section 5.4)

# Ex. 8 Answer: Larger Weights Have Tradeoffs

$$\dot{x} = u, \quad x(0) = 1, \quad Q = 1, \quad -P^2/R + 1 = 0$$

$$P = \sqrt{R}, \quad K = P/R = 1/\sqrt{R}, \quad u = -Kx$$

Evaluate unsaturated responses with common weights $Q_e = R_e = 1$; input limit 0.6 checks only the initial request.

| Input weight | Gain and initial input | Common-weight cost | Initial check |
|---|---|---|---|
| $R = 1$ | $K = 1,\ u(0) = -1$ | 1.00 | Exceeds limit initially |
| $R = 4$ | $K = 0.5,\ u(0) = -0.5$ | 1.25 | Within limit |

$P$: Riccati solution; $R$: design input weight; larger weights reduce input and slow convergence.

### 5.8.2 Filtering and Control Calculations: One Second-Order LQR Step

# Ex. 8 Extension: From Weights to Gain to Input

$\Delta\mathbf{x} = [\Delta p_{\mathrm{N}},\ \Delta v_{\mathrm{N}}]^{\mathrm{T}}$; $\Delta u$ is the northward acceleration increment; assume fast inner loops and negligible inter-axis coupling.

Normalization scales: 1 m (position), 1 m/s (velocity), 1 m/s² (input).

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad Q = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1$$

$$A^{\mathrm{T}}P + PA - PBR^{-1}B^{\mathrm{T}}P + Q = 0$$

$$P = \begin{bmatrix} 2\sqrt{5} & 2 \\ 2 & \sqrt{5} \end{bmatrix}, \quad K = R^{-1}B^{\mathrm{T}}P = [2 \quad \sqrt{5}]$$

$$\Delta\mathbf{x}_0 = \begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}, \quad \Delta u_0 = -2(0.2) - \sqrt{5}(-0.1) = -0.176393\ \mathrm{m/s^2}$$

### 5.8.2 Filtering and Control Calculations: Full LQR Response

# Ex. 8 Answer: Check Input Limits as Well as Stability

$$\lambda(A - BK) = -1.1180 \pm 0.8660j, \quad |\Delta u| \le 1\ \mathrm{m/s^2}$$

| Time / s | Position deviation / m | Velocity deviation / (m/s) | Acceleration / (m/s²) |
|---|---|---|---|
| 0 | 0.200000 | −0.100000 | −0.176393 |
| 1 | 0.077905 | −0.104054 | 0.076864 |
| 2 | 0.011625 | −0.033390 | 0.051412 |
| 4 | −0.002684 | 0.002288 | 0.000250 |
| 8 | 0.000032 | −0.000037 | 0.000018 |

$$J = \Delta\mathbf{x}_0^{\mathrm{T}} P \Delta\mathbf{x}_0 = 0.121246$$

Counterexample: $\Delta p_N = 2$ m, $\Delta v_N = 0$ requests $\Delta u = -4\ \mathrm{m/s^2}$, exceeding the same limit.

## Chapter Outline

# Exercises and Classroom Discussion

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

**5.8.1 Concepts and Methods**

Textbook: Sec. 5.4, Exercises 1–3

**5.8.2 Filtering and Control Calculations**

Textbook: Sec. 5.4, Exercises 4, 5, 7, 8

**5.8.3 System Integration and Validation**

Textbook: Sec. 5.4, Exercises 6, 9, 10

**5.8.4 Integrated Design and Classroom Discussion**

Baseline, one change, and report review

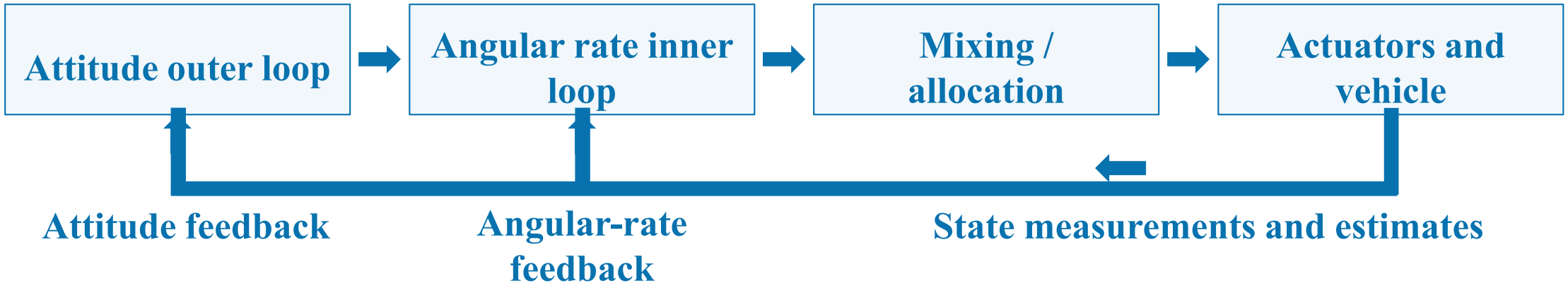
### 5.8.3 System Integration and Validation: Original Questions and Review

# Exercises 6, 9, 10: Read the Questions, Then Review

| Ex. | Questions: Textbook, Sec. 5.4 | Review Slides |
|---|---|---|
| **Ex. 6** | Design a quadcopter's inner/outer attitude loops and mixing structure, and explain integration with PX4/uORB. | 5.3: Units 3–4<br>5.5: Unit 2 |
| **Ex. 9** | Complete closed-loop development in the RflySim+PX4+Simulink toolchain, from simulation to real-vehicle deployment, and explain how to avoid control-module conflicts. | 5.5: model and firmware<br>5.6: Units 3–5 |
| **Ex. 10** | Describe failsafe implementation for communication loss, inertial measurement unit (IMU) anomalies, and other faults. | This section: pp. 27, 35<br>Chapter 7: further study |

### 5.8.3 System Integration and Validation: Cascaded Attitude Control and Mixing

# Ex. 6: Connect Attitude Loops to Four Motor Outputs




$$p_{\mathrm{r}} = K_{\mathrm{att}}(\phi_{\mathrm{r}} - \hat{\phi}), \quad \tau_{x,\mathrm{req}} = K_p(p_{\mathrm{r}} - p_{\mathrm{m}}) + I - K_d\dot{p}_{\mathrm{m}}$$

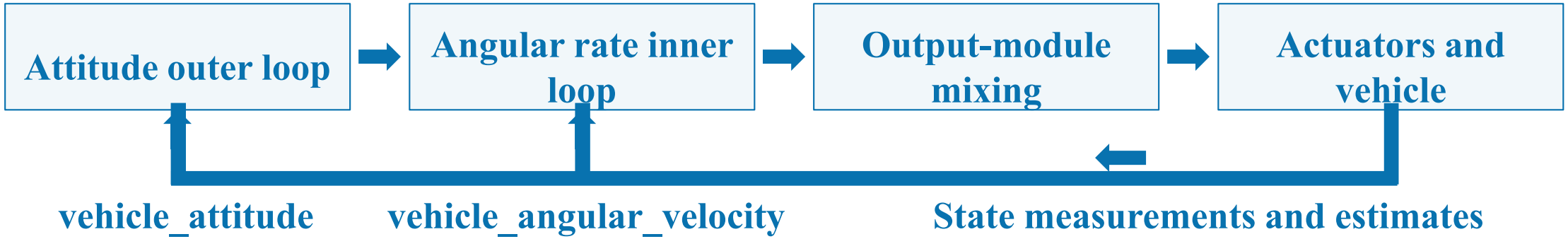
$p_{\mathrm{r}}/p_{\mathrm{m}}$: roll-rate target/measurement; $\hat{\phi}$: estimated roll angle; $\tau_{x,\mathrm{req}}$: requested roll torque.

$$\boldsymbol{\delta} = [0.5,0.1,0,0]^{\mathsf{T}} \Rightarrow \boldsymbol{\sigma} = [0.4,0.4,0.6,0.6]^{\mathsf{T}}$$

$\delta$: normalized total thrust and 3-axis torques; $\sigma$: four throttle channels
(Eq. 5.29, Slide Section 5.3; coefficients 1).

### 5.8.3 System Integration and Validation: Attitude Loops and Mixing

# Ex. 6 Answer: Integrate the Control Loops with PX4/uORB




Inner loop publishes actuator_controls_0 (normalized torques and throttle) → mixing → actuator_outputs → PWM (pulse-width modulation)

Check FRD body axes, normalized units, timestamps, saturation, and one output source. For v1.14 and later, control_allocator handles allocation; see “Control Outputs Across Versions” in Slide Section 5.5.

### 5.8.3 System Integration and Validation: Development and Release Evidence

# Ex. 9: Validate Each Stage and Avoid Module Conflicts

| Stage | Evidence to proceed | Fallback on failure |
|---|---|---|
| **Simulink modeling** | Frames, units, directions; simulated response and limits | Correct model, parameters, or signals |
| **Code generation** | Compile C code into firmware as px4_simulink_app | Restore build and configuration |
| **Software-in-the-loop SITL (textbook: SIL)** | Firmware closed loop, message rates, one publisher | Stop the new source; return to baseline |
| **Hardware-in-the-loop HITL (textbook: HIL)** | Real flight-controller timing; simulated state and feedback | Stop simulation; restore configuration |
| **Real-vehicle prep.** | Calibration, channels, native baseline, fallback, logs | Abort as planned and review |

Full takeover: mask native motor outputs. Partial replacement: one publisher for that layer's topic.

### 5.8.3 System Integration and Validation: Implementation Workflow

# Ex. 9 Answer: RflySim+PX4+Simulink Workflow

| | | |
|---|---|---|
| **① Simulink modeling**<br>Closed-loop model simulation | **② Code generation**<br>Simulink Coder → C code | **③ Compile into firmware**<br>px4_simulink_app |
| **④ SITL**<br>Software-in-the-loop | **⑤ HITL**<br>Hardware-in-the-loop (HITLRun) | **⑥ Real vehicle**<br>Calibration, fallback, and logs |

**Review:** "Model to Firmware" and "Validation Levels" in Slide Section 5.6; "Native Output Masking Options" in Section 5.5

**Full motor-output takeover**
Installer item 10: "是否屏蔽PX4自身控制器输出" (mask native PX4 output) → "是" (Yes);
only px4_simulink_app drives the motors;
textbook: mask the PX4 motor-control code.

**Replace only one loop**
Mask native publication to the same uORB topic;
keep one publisher at that layer;
e9 masks native actuator_controls_0 publication.

### 5.8.3 System Integration and Validation: Module Replacement Example

# Ex. 9 Example: e9-ReplacePX4Module (Firmware 1.12.3)

Folder: [安装目录]/RflySimAPIs/5.RflySimFlyCtrl/1.BasicExps/e9-ReplacePX4Module/
Start with Readme.pdf and Init_control.m; requires Pixhawk 6X, a transmitter, and a receiver

**Steps:** mask native publication → build and flash → verify native publication stops in HITL → upload model → take off in Stabilized mode

**Full procedure and layered checks:** "Module Replacement Example" and "Partial Replacement Acceptance Checks" in Slide Section 5.5

**Criteria:** baseline works and native publication stops; new source alone is active; takeoff and attitude control work, with poorer yaw control than native

**Diagnose:** monitor actuator_controls_0 with uORB; use Inspector to assist with telemetry inspection

**Restore:** the experiment changes firmware source; rerun the installer afterward to restore firmware

### 5.8.3 System Integration and Validation: Failsafe Mechanisms

# Ex. 10: Fault Actions Depend on Remaining State Reliability

| Fault | Detection and state dependencies | Reference response |
|---|---|---|
| **Communication loss** | Command timeout/loss flag; check localization, attitude, and energy | If states remain usable, hold, return, or land under the preset policy |
| **IMU anomaly** | Timestamps, health, and consistency; attitude/angular rate become less reliable | Switch only to a valid backup; otherwise use the degraded strategy |
| **After the response** | Monitor fault duration, action execution, and state validity | Abort under the plan when conditions are met; retain event logs |

Return needs usable localization and attitude control; select actions by fault type and remaining state.

### 5.8.3 System Integration and Validation: Diagnosing Anomalies

# Maximum Request, No Position Change: What Next?

Synthetic scenario: north-east-down (NED), down positive; target altitude 1 m means vertical position −1 m.

| Time / s | Position / m | Total thrust request (norm.) | State timestamp / s |
|---|---|---|---|
| **0.00** | 0.00 | 1.00 | 0.00 |
| **0.01** | 0.00 | 1.00 | 0.00 |
| **0.02** | 0.00 | 1.00 | 0.00 |

- **No takeover or actuator limits:** check the active publisher and achieved actuator outputs.
- **Stale state or model not advancing:** check update rate, timestamps and simulation time.
- **Direction/units:** check NED signs and thrust direction before changing parameters.

## Chapter Outline

# Exercises and Classroom Discussion

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

**5.8.1 Concepts and Methods**

Textbook: Sec. 5.4, Exercises 1–3

**5.8.2 Filtering and Control Calculations**

Textbook: Sec. 5.4, Exercises 4, 5, 7, 8

**5.8.3 System Integration and Validation**

Textbook: Sec. 5.4, Exercises 6, 9, 10

**5.8.4 Integrated Design and Classroom Discussion**

Baseline, one change, and report review

### 5.8.4 Integrated Design and Classroom Discussion: Common Baseline

# State the Plant, Closed Loop, and Evaluation Conditions

Task: choose one branch and improve one part of a common baseline; submit using the reproducible report example.

**Plant/model**

**Check:** operating point, states, inputs and frames

**Evidence:** model, parameters and validity

**Feedback source**

**Check:** measurement/estimate/truth and sampling

**Evidence:** data, timestamps and health

**Control/allocation**

**Check:** loops, requests, achieved values and limits

**Evidence:** equations, signal diagram and outputs

**Conditions/metrics**

**Check:** same inputs, initial state, noise and constraints

**Evidence:** responses, inputs and failure conditions

### 5.8.4 Integrated Design and Classroom Discussion: Improvement Branches

# Change One Main Part and Explain Benefits and Costs

| Topic branch | Main change | Comparison and acceptance |
|---|---|---|
| **Filter parameters** | Time constant; observation weights | Same data segment; compare bias, noise, and lag |
| **Control method** | Anti-windup; model-based feedback | Same model, initial states, limits; compare response and input |
| **Output constraints** | Allocation limits; failed actuators | Same control request; compare achieved values and allocation residuals |
| **Vehicle transfer** | Plant model; control inputs | Same validation framework; check new operating points and coupling |

### 5.8.4 Integrated Design and Classroom Discussion: Reproducible Report Example

# Neither Setting Meets Both Requirements

Analysis: synthetic pitch data; initial angle 0; step 0.01 s; gyro bias 0.02 rad/s.

Accelerometer reference contaminated by 0.2 rad during 2–2.3 s; only the time constant changes, with all other inputs and equations identical.

| Acceptance criterion | Time constant 0.1 s | Time constant 0.5 s |
|---|---|---|
| **Residual at 10 s < 0.2°** | 0.115°: pass | 0.573°: fail |
| **Pulse peak < 6°** | 10.917°: fail | 5.700°: pass |

- **Conclusion:** neither setting meets both criteria; add observation-validity checks and bias handling.
- **Deliver:** data, equations, parameters, versions, curves, failure conditions and reproducible calculations.

### 5.8.4 Integrated Design and Classroom Discussion: Engineering Tradeoffs

# When Is a More Complex Algorithm Worth Using?

| Scenario | Reference judgment | Evidence needed |
|---|---|---|
| **Less noise, more delay** | Tracking and stability requirements determine acceptable delay | Error, delay, and closed-loop response using the same data |
| **Faster tracking, stronger input** | Actuator and energy limits determine achievable action first | Input peaks, rates, saturation, and response metrics |
| **Better model, heavier computation** | Computation must finish within the task's allowed delay | Model-mismatch tests, worst-case compute delay, and update rate |

After-Class Practice

# Examples for Exercises 6 and 7

**Ex. 6 · e5-AttitudeCtrl/e5.1 (tuning: e5.2 in Slide Section 5.4)**
**Start:** Sim/Init_control.m opens AttitudeControl_Sim.slx automatically;
HITL requires Pixhawk 6X
**Criteria:** simulated attitude follows transmitter channels; linear analysis gives a Bode plot and stability margins

**Ex. 7 · e4-FilterDesign/e4.1 (run: "e4.1 and e4.2 Examples" in Slide Section 5.2)**
**Start:** Attitude_estimator0.m reads bundled e4_A.bin; MATLAB only, no flight controller
**Criteria:** four pitch curves gyro, acc, cf, px4; normally cf has no drift and is smoother than acc

All folders are under [安装目录]/RflySimAPIs/5.RflySimFlyCtrl/1.BasicExps/ and contain Readme.pdf.

After-Class Practice

# Examples for Exercises 9 and 10

**Ex. 9 · e9-ReplacePX4Module (firmware 1.12.3)**
Requires Pixhawk 6X and a transmitter; changes firmware source; steps: “Module Replacement Example” in Slide Section 5.5
**Criteria:** baseline works, native publication stops; new source alone is active, Stabilized takeoff works, yaw is poorer than native

**Ex. 10 · e8-FailsafeLogic/e8.1 (e8.4: automatic return after transmitter loss)**
**Start:** Sim/Init_control.m opens e8_1_sim.slx with RflySim3D; HITL requires Pixhawk 6X
**Criteria:** manual-to-return switching gives a return curve; change ch5 from 300 to 500 for a landing curve

All folders are under [安装目录]/RflySimAPIs/5.RflySimFlyCtrl/1.BasicExps/ and contain Readme.pdf.

## Chapter Summary

# Reliable Low-Level Control Needs Estimation and Validation

- **Concepts:** estimation provides state and confidence. Low-level control corrects motion within actuator limits and realizes requests through allocation.
- **Methods:** set the operating point and scales, then compute gains and one-step outputs. Compare fixed conditions and common metrics; stop when targets are met. Progress through model simulation, SITL, HITL and real-vehicle validation, with one control source per output layer.
- **Limits:** linear models are local to trim; saturation requires reconsidering unconstrained optima. New vehicle types or airspeeds require gains and validation to be revisited (gain scheduling). Topics depend on firmware.
- **Pitfalls:** treating estimates as truth; ignoring saturation; inconsistent units, frames or timestamps; calling one calculation a full simulation or a successful build a successful run; mixing v1.12.3/v1.14+ topics; comparing different conditions; treating one passing case as a full-envelope guarantee.

**Reliable low-level control gives high-level decisions a dependable basis for execution.**