Chapter Outline

Control Performance Evaluation and Tuning

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

5.4.1 Control Performance Metrics
Textbook: Sec. 5.1.4, items 1–2

5.4.2 PID Parameters and Tuning Procedures
Textbook: Sec. 5.1.4, items 3–4; proportional–integral–derivative (PID)

5.4.3 Model-Based Parameter Design
Textbook: Sec. 5.1.4, item 5

5.4.4 Whole-Vehicle Performance Tuning
Textbook: Sec. 5.1.4, item 6; plus frequency response and delay

5.4.1 Control Performance Metrics: Stability, Speed, and Accuracy

Responses Can Differ Despite the Same Target

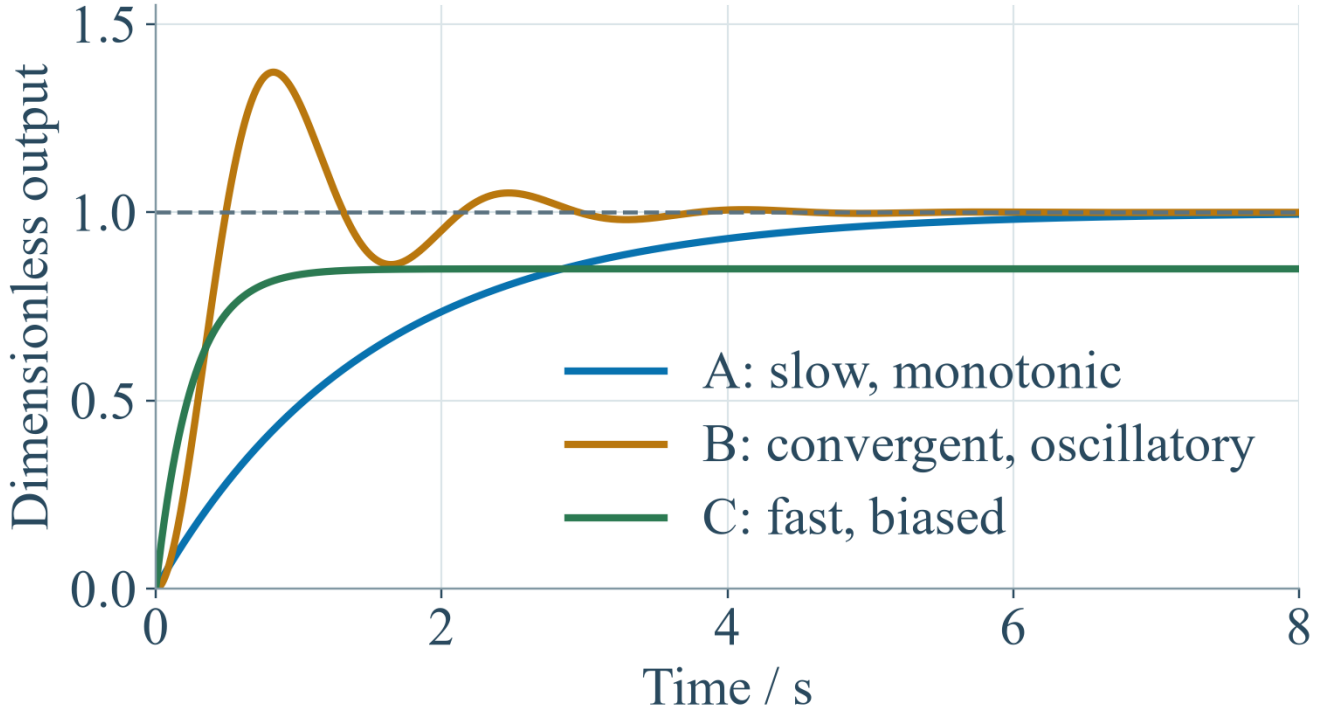


Figure: Three instructional responses, all with a target of 1

Stability: Does the response converge after a disturbance?

Damping: Are amplitude and cycle count acceptable?

Response speed: How long to respond to a target change?

Accuracy: How much error remains at steady state?

5.4.1 Control Performance Metrics: Step-Response Metrics

Four Metrics Can Be Read from One Step Response

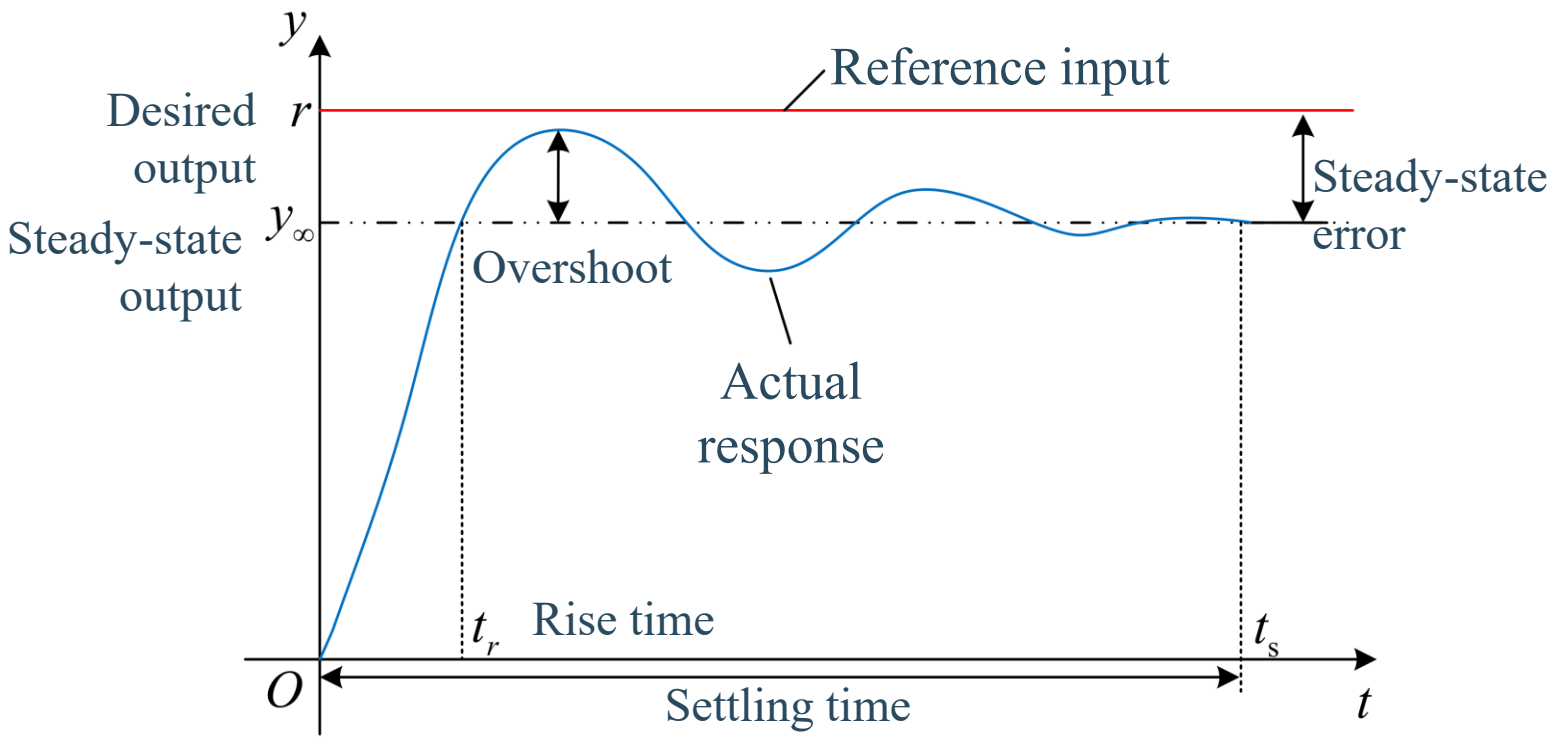


Figure 5.3 Step-response metrics

Rise time: First 0–100% crossing.

Overshoot: Peak excess as a percentage of the final value.

Settling time: Enter and stay within a 5% band.

Steady-state error: Absolute final-value error relative to the reference.

Tables: 10–90% rise; 2% settling band.

5.4.1 Control Performance Metrics: Worked Example of Metric Calculation

Readings, Denominator, and Thresholds Define the Metrics

Example readings: $r = 1.00$, $y_\infty = 0.98$, $y_{\max} = 1.10$; the initial value is 0.

$$e_{ss} = |r - y_\infty| = |1.00 - 0.98| = 0.02$$

$$\sigma\% = \frac{y_{\max} - y_\infty}{y_\infty} \times 100\% = \frac{1.10 - 0.98}{0.98} \times 100\% = 12.24\%$$

Output is 0.098 at 0.20 s and 0.882 at 1.10 s.
Use final value 0.98 for 10%–90% thresholds; settling also requires staying in the band.

$$t_{r,10\%-90\%} = 1.10 - 0.20 = 0.90\ \text{s}$$

$$0.98(1 \pm 5\%) = [0.931,\ 1.029]$$

5.4.1 Control Performance Metrics: Test Inputs and Acceptance Criteria

Different Inputs Test Different Control Capabilities

Input or condition	Main observations	Instructional acceptance criteria
Unit step	Rise, overshoot, settling, steady-state error	Overshoot ≤15%; 2% settling time <5 s
Constant disturbance of 0.2	Maximum deviation and recovery	Deviation ≤0.05; final value returns to 1
Continuous trajectory	Time alignment of target and output	Report both mean squared error and lag
Change in load or delay	Response and remaining actuator headroom	Keep constraints fixed; repeat the same input

Fair comparison: fix initial conditions, plant, input, sampling, filtering, and output limits.
Record control input and time fraction at the limit; do not select only the best output trace.

Chapter Outline

Control Performance Evaluation and Tuning

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

5.4.1 Control Performance Metrics

Textbook: Sec. 5.1.4, items 1–2

5.4.2 PID Parameters and Tuning Procedures

Textbook: Sec. 5.1.4, items 3–4; proportional–integral–derivative (PID)

5.4.3 Model-Based Parameter Design

Textbook: Sec. 5.1.4, item 5

5.4.4 Whole-Vehicle Performance Tuning

Textbook: Sec. 5.1.4, item 6; plus frequency response and delay

5.4.2 PID Parameters and Tuning Procedures: How PID Parameters Affect Performance

PID Parameter Effects Show Typical Trends

Gain increase	Rise time	Overshoot	Settling time	Steady-state error	Stability
Increase K_p	Decreases	Increases	Slightly increases	Decreases	Decreases
Increase K_i	Slightly decreases	Increases	Increases	Greatly decreases	Decreases
Increase K_d	Slightly increases	Decreases	Decreases	Almost unchanged	Improves

Table 5.1 Qualitative Effects of PID Gains on Performance (Typical Trends)

Keep the plant and other gains fixed; check noise, saturation, and stability margins.

Proportional–integral–derivative (PID): P uses current error, I sums past error, and D uses the error rate.

5.4.2 PID Parameters and Tuning Procedures: Overview of PID Tuning Effects

P Speeds Up, I Removes Offset, D Reduces Overshoot: Balance All Three

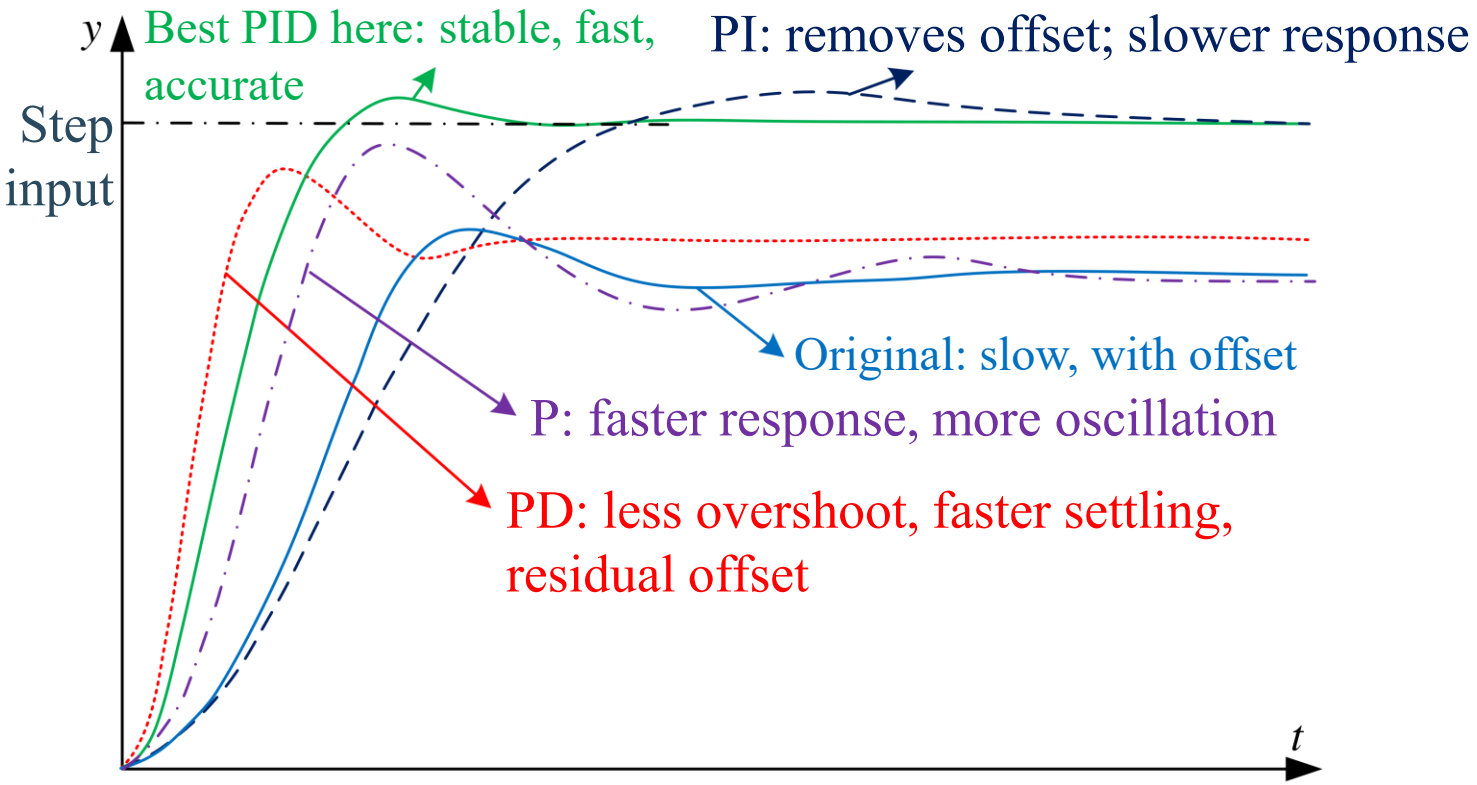


Figure 5.4 Effect of PID Tuning on the Step Response

Original: slow, with steady-state error.

P: faster response; oscillation may increase.

PD: less overshoot; offset may remain.

PI: removes offset; slows the response.

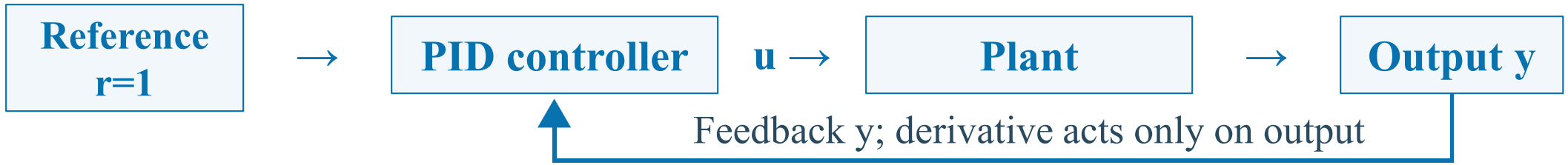
PID: one compromise among speed, accuracy, and stability.

5.4.2 PID Parameters and Tuning Procedures: Common Instructional Model

Fix the Model and Control Law Before Comparing Gains

$$\ddot{y} + 2\dot{y} + y = u, \quad y(0) = \dot{y}(0) = I(0) = 0$$

$$u = K_{\mathrm{p}}(1 - y) + K_{\mathrm{i}} I - K_{\mathrm{d}}\dot{y}, \qquad \dot{I} = 1 - y$$



I accumulates error over time; zero initial conditions, unit step, time in s.
Output and input are dimensionless; integral gain is in 1/s and derivative gain in s.
Derivative acts on the output without filtering; no saturation is applied. Peak input ≤4.2, below the ±5 comparison boundary.

$$[K_{\mathrm{p}}, K_{\mathrm{i}}, K_{\mathrm{d}}]\text{: P} = [4,0,0],\ \text{PD} = [4,0,2],\ \text{PI} = [4,2,0],\ \text{PID} = [4,2,2]$$

5.4.2 PID Parameters and Tuning Procedures: P and PD: Numerical Comparison

Derivative Action Reduces Overshoot but Slows the Rise

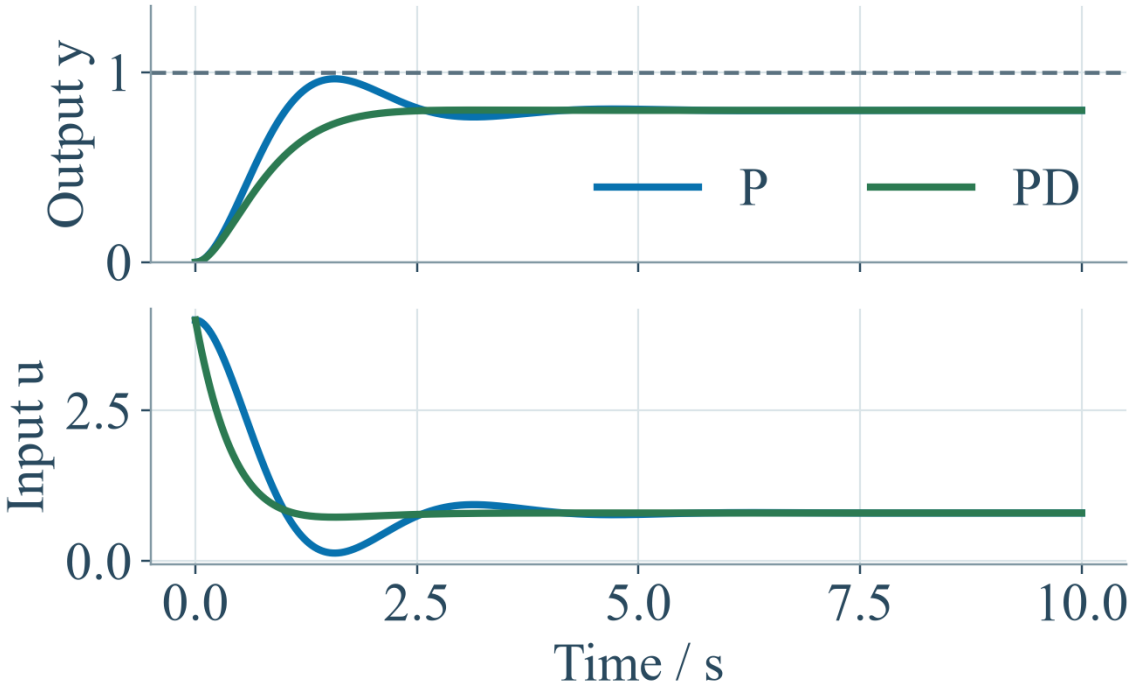


Figure: P and PD output y (top) and input u (bottom), unit step

Metric	P	PD
Final value	0.800	0.800
Overshoot /%	20.79	0.19
Rise time/s	0.689	1.278
2% settling/s	3.736	2.075

$$\ddot{y} + 2\dot{y} + y = u, \quad K_p = 4, \quad K_i = 0, \quad K_d = 0 \text{ (P) or } 2 \text{ (PD)}$$

5.4.2 PID Parameters and Tuning Procedures: PI and PID: Numerical Comparison

Integral Action Removes Offset; Transients Still Need a Trade-Off

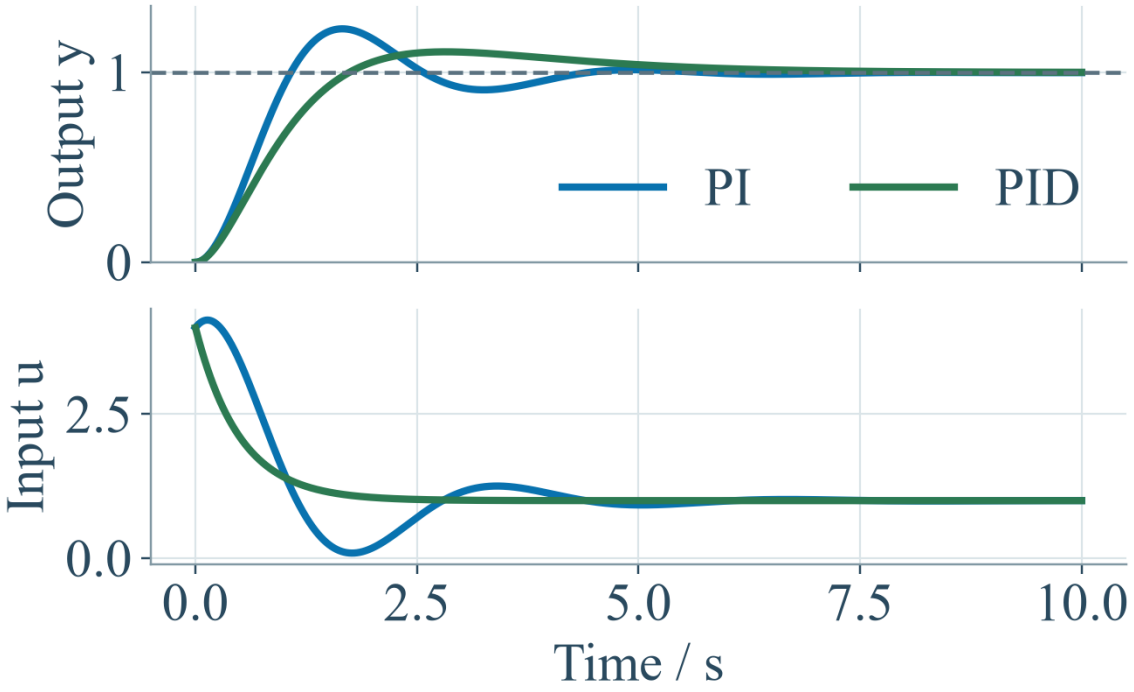


Figure: PI and PID output y (top) and input u (bottom), unit step

Metric	PI	PID
Final value	1.000	1.000
Overshoot /%	22.96	10.84
Rise time/s	0.727	1.173
2% settling/s	4.172	5.990

$$\ddot{y} + 2\dot{y} + y = u, \quad K_p = 4, \quad K_i = 2, \quad K_d = 0 \text{ (PI) or } 2 \text{ (PID)}$$

5.4.2 PID Parameters and Tuning Procedures: Combined Validation Results

One Candidate Must Pass the Same Set of Conditions

Same instructional model; baseline PI $[K_p, K_i, K_d] = [4,2,0]$, candidate PID $[4,2,2]$.

① Step test
Reference 1; no disturbance or noise.
Read overshoot and 2% settling time.

② Disturbance test (separate)
Reference 1; add +0.2 at the plant input at 8 s; no measurement noise.

③ Noise test (separate)
Reference 0; no disturbance;
Measurement noise $n(t) = 0.02\sin(30t)$.
Root mean square (RMS) input, 15–20 s.

Metric / criterion	PI	PID	Assessment
Step overshoot ≤15%	22.96%	10.84%	Candidate passes
2% settling time ≤5 s	4.172 s	5.990 s	Candidate fails
Peak disturbance error ≤0.05	0.04147	0.03343	Both pass
RMS control input (noise only)	0.05693	0.84610	High noise cost

5.4.2 PID Parameters and Tuning Procedures: Parameters and Log Evidence

Each Gain Change Needs a Complete Set of Results

Record	Baseline PI	Candidate PID
Gain vector	$[K_p, K_i, K_d] = [4,2,0]$	$[K_p, K_i, K_d] = [4,2,2]$
Conditions	Same plant: step, disturbance, noise	Same plant: step, disturbance, noise
Step assessment	Overshoot fails; settling passes	Overshoot passes; settling fails
Disturbance and noise	Disturbance passes; low noise cost	Disturbance passes; high noise cost

Logging example: [安装目录]/RflySimAPIs/5.RflySimFlyCtrl/0.ApiExps/5.Log-Write-Read/Readme.pdf

Log targets, states, requested/actual outputs, and timestamps; verify the new parameters took effect.

5.4.2 PID Parameters and Tuning Procedures: Multicopter Tuning Procedure

Start with a Stable Inner Loop and Test One Change at a Time

Stage	Main action	Save evidence
Establish baseline	Check directions and limits; set I=D=0; start with small P	Version and initial values
Tune proportional gain	Increase P to slight oscillation, then reduce it a little	Step response and input
Add integral action	Add enough I to remove offset; prevent windup	Offset and integral state
Add derivative action	Add D with low-pass filtering; compare damping and noise	Response and noise
Retest both loops	Once the inner loop is stable, tune the outer loop; change one gain at a time	Identical conditions
Staged validation	Simulation → hardware-in-the-loop (HITL; HIL in the textbook) → real vehicle; first flight: limit tilt and speed	Passed checks at each stage

See “e5.2 Attitude Tuning Experiment” for its directory, steps, and criteria.

5.4.2 PID Parameters and Tuning Procedures: e5.2 Attitude Tuning Experiment

Tune Inner-Loop Rate, Then Outer-Loop Angle, Then Sweep Frequency

Directory: [安装目录]/RflySimAPIs/5.RflySimFlyCtrl/1.BasicExps/e5-AttitudeCtrl/e5.2/

Start with Readme.pdf; run Init_control.m to open AttitudeControl_tune.slx automatically. Software simulation only; MATLAB 2022b or later required.

① **Inner-loop P:** set I and D to 0. In AttitudeControl, replace desired angular rate with a step, log the step and q signals, and increase Kp_PITCH_AngleRate gradually.

② **After each change:** rerun the edited Init_control.m, click Run, and view the step response in Simulation Data Inspector.

③ **I, D, and outer loop:** tune Ki_PITCH_AngleRate and Kd_PITCH_AngleRate next. Once the inner loop is satisfactory, use a desired pitch-angle step and increase outer-loop Kp_PITCH_ANGLE from a small value.

④ **Frequency sweep:** set desired pitch angle as Open-loop Input and actual pitch angle as Open-loop Output; sweep frequency to obtain a Bode plot.

5.4.2 PID Parameters and Tuning Procedures: Criteria, Stopping, and Staged Validation

Stop at the Preset Targets, Then Validate in Stages

Expected results and criteria: meet preset step-response overshoot and settling-time targets, with no sustained oscillation. Read phase margin at the gain crossover and gain margin at the phase crossover.

Stop tuning: repeat the same hover, turns, and forward flight at fixed altitude. Stop when responses are fast, stable, and have no obvious overshoot; proceed to the next stage.

Common issue: if Init_control.m is not rerun after a change, the model keeps using old parameters. See Readme FAQ Q1 and Q2 for tuning order and margin readings.

Validate in stages:

Software simulation	→	Hardware-in-the-loop (HITL)	→	Real-vehicle experiments
Modeling, identification, and virtual tuning find the right parameter magnitudes		Run the same parameters on flight-controller hardware; repeat the fixed maneuvers		Limit maximum tilt and speed; fly low in an open area

Firmware version: for HITL and real-vehicle tests, see slide deck 5.5, “Control Outputs Across Versions”.

Chapter Outline

Control Performance Evaluation and Tuning

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

5.4.1 Control Performance Metrics

Textbook: Sec. 5.1.4, items 1–2

5.4.2 PID Parameters and Tuning Procedures

Textbook: Sec. 5.1.4, items 3–4; proportional–integral–derivative (PID)

5.4.3 Model-Based Parameter Design

Textbook: Sec. 5.1.4, item 5

5.4.4 Whole-Vehicle Performance Tuning

Textbook: Sec. 5.1.4, item 6; plus frequency response and delay

5.4.3 Model-Based Parameter Design: Reference Model Sources

Define the States, Input, and Model Source First

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u, \qquad \mathbf{x} = \begin{bmatrix} \theta_{\text{err}} \\ \omega_{\text{err}} \end{bmatrix}$$

States (actual minus target): θ_{err} is angle error (rad); ω_{err} is angular-rate error (rad/s). The target is fixed and target angular rate is 0; input is normalized torque.

Roll reference model

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -5.8 & -0.59 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 19.27 \end{bmatrix}$$

Pitch reference model

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1.36 & -6.81 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 20.54 \end{bmatrix}$$

Use the bundled axis reference model if its file, A, or B is missing, values are nonfinite, or fit <60%. Identified from FEISI-J150 frequency sweeps; fit measures agreement between model output and logged data.

Start here: 5.RflySimFlyCtrl/2.AdvExps/e5_LQR-CtrlExp/2.ModelID2LQRSimple/Readme.pdf

5.4.3 Model-Based Parameter Design: Two State-Feedback Design Methods

Pole Placement and LQR Use Different Design Objectives

Pole placement: specify desired poles

$$\mathrm{eig}(\mathbf{A}-\mathbf{BK})=\{p_1,p_2,\dots\}$$

```
K = place(A,B,p);
```

Linear quadratic regulator (LQR): specify a cost

$$J=\int_0^\infty (\mathbf{x}^\top\mathbf{Q}\mathbf{x}+u^\top\mathbf{R}u)\,\mathrm{d}t$$

```
K = lqr(A,B,Q,R);
```

The example uses pole placement; LQR additionally specifies Q (state weights) and R (input weights).

Pole placement function https://www.mathworks.com/help/control/ref/place.html

LQR solver function https://www.mathworks.com/help/control/ref/lti.lqr.html

5.4.3 Model-Based Parameter Design: Pole Placement for the Reference Models

Match Coefficients to Obtain Gains for the Desired Poles

Fixed target; θ denotes angle error; a, d, b are entries of the reference matrices A=[0,1;−a,−d], B=[0;b].

$$\ddot{\theta} = -a\theta - d\dot{\theta} + bu, \quad u = -k_1\theta - k_2\dot{\theta}$$

$$(s + 6.7)(s + 5.3) = s^2 + 12s + 35.51$$

$$s^2 + (d + bk_2)s + (a + bk_1) \Rightarrow d + bk_2 = 12, \quad a + bk_1 = 35.51$$

$$\mathbf{K}_{\text{roll}} = \left[\frac{35.51 - 5.8}{19.27}, \frac{12 - 0.59}{19.27}\right] = [1.5418,\ 0.5921]$$

$$\mathbf{K}_{\text{pitch}} = [1.6626,\ 0.2527], \quad p_1 = -6.7,\ p_2 = -5.3$$

5.4.3 Model-Based Parameter Design: Complete Reference-Model Results

Here, the Two Angle-Error Curves Coincide for Equal Initial States

Initial conditions: $\theta_{\mathrm{err}} = 20^\circ$, $\omega_{\mathrm{err}} = 0$ rad/s.

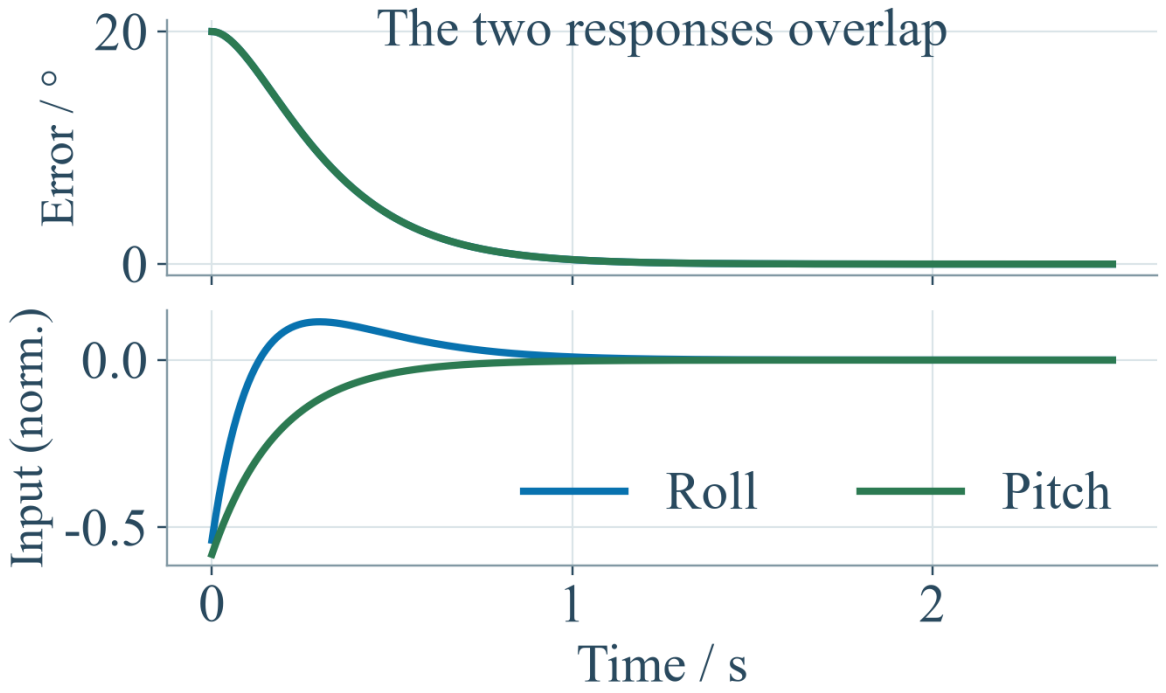


Figure: Angle errors (top) and normalized inputs (bottom), no saturation

Reading	Roll	Pitch
Initial input	−0.5382	−0.5804
Angle error 0.5 s	4.1059°	4.1059°
Angle error 1.0 s	0.3846°	0.3846°
Angle error 2.5 s	Both about 0.000164°	

Same closed-loop polynomial and initial states;
normalized inputs differ between axes.

5.4.3 Model-Based Parameter Design: Weights and the Riccati Equation

LQR Combines State Error and Input in One Cost

Continuous time, infinite horizon, no hard constraints; normalize states and inputs by allowable scales. Q weights states, R weights inputs, and P is the stabilizing Riccati solution.

$$J = \int_0^\infty (\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u})\, \mathrm{d}t$$

$$\mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P} + \mathbf{Q} = 0$$

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}, \qquad \mathbf{u} = -\mathbf{K}\mathbf{x}$$

Common sufficient conditions: $Q \succeq 0,\ R \succ 0$; (A, B) stabilizable, $(Q^{1/2}, A)$ detectable. $\succeq$ means positive semidefinite; $\succ$ means positive definite. Nondecaying modes must be controllable and included in the state cost.

5.4.3 Model-Based Parameter Design: One-Dimensional LQR Example

Scalar LQR by Hand: P=√R Is the Stabilizing Root

$$\dot{x} = u, \quad x(0) = 1, \quad A = 0,\ B = 1,\ Q = 1$$

$$-\frac{P^2}{R} + 1 = 0 \quad \Rightarrow \quad P = \sqrt{R}, \quad K = \frac{P}{R} = \frac{1}{\sqrt{R}}$$

Input weight	Stabilizing solution	Feedback gain	Closed-loop pole
$R = 1$	$P = 1$	$K = 1$	-1
$R = 4$	$P = 2$	$K = 0.5$	-0.5

$$x(t) = e^{-Kt}, \qquad u(t) = -Ke^{-Kt}$$

5.4.3 Model-Based Parameter Design: Complete LQR Results and a Common Cost

Lower Input Demand Means Slower Convergence

2% initial-value band: $|x(t)| \le 0.02|x(0)| = 0.02$

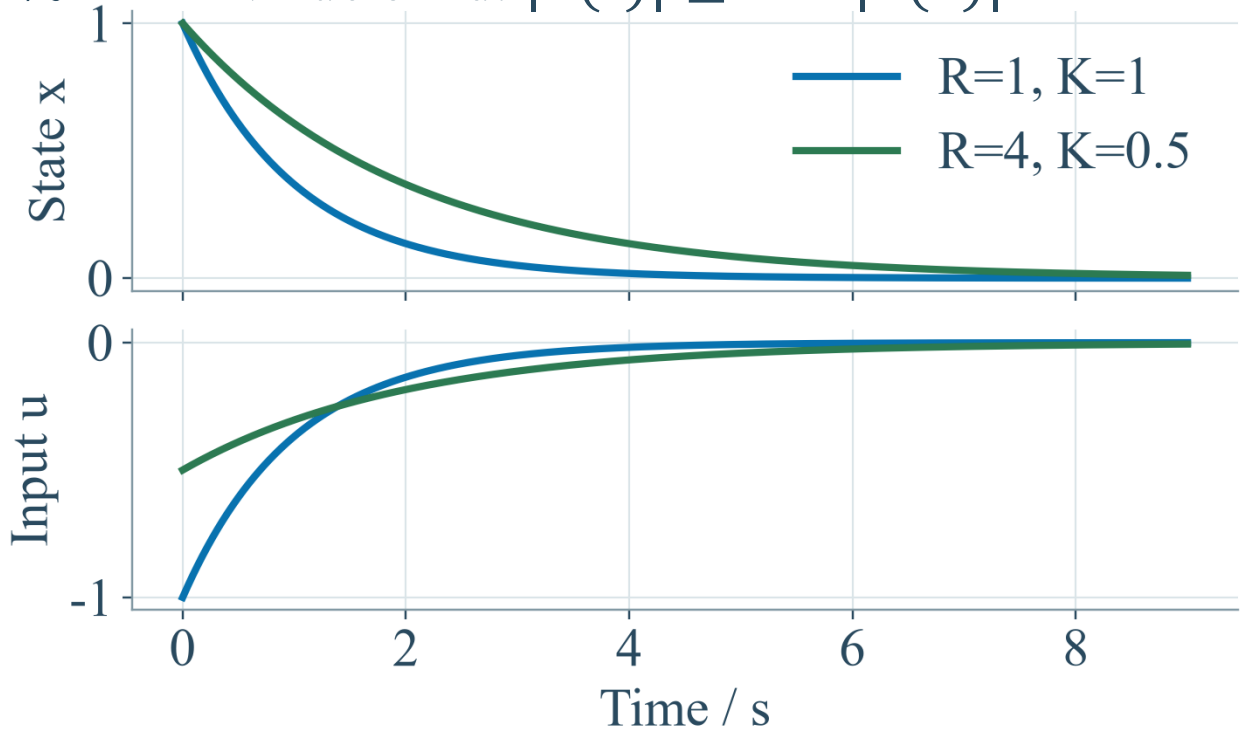


Figure: State x (top) and input u (bottom) for R=1 and R=4

Evaluate both: $Q = R = 1$.

Metric	$R = 1$	$R = 4$
Max. |u|	1.000	0.500
2% settling/s	3.912	7.824
Integral of squared input	0.500	0.250
Common cost	1.000	1.250

$$J_{\text{ref}} = \int_0^{\infty} (x^2 + u^2)\,\mathrm{d}t$$

5.4.3 Model-Based Parameter Design: Feasibility Check

Compute the Requested Input, Then Check Feasibility

One-step example: $K = [4,3]$ is a separate instructional gain choice.

$$\mathbf{x} = \begin{bmatrix} 0.1\ \text{rad} \\ 0.2\ \text{rad/s} \end{bmatrix}, \qquad \mathbf{K} = [4,\ 3]$$

$$u = -\mathbf{K}\mathbf{x} = -(4 \times 0.1 + 3 \times 0.2) = -1.0$$

Quantity	Meaning and units
$k_1 = 4$	Angle-error gain: 1/rad
$k_2 = 3$	Angular-rate-error gain: s/rad

Actuator limits ± 0.6: request -1 is clipped to -0.6; resimulate the constrained system to reassess the pole-based conclusions.

$$u_{\text{actual}} = \text{clip}(u, -0.6, 0.6) = -0.6$$

5.4.3 Model-Based Parameter Design: Design and PID Implementation

Mapping State Feedback Depends on the Control Structure

$$u = -k_\theta(\theta - \theta_{\mathrm{sp}}) - k_\omega(\omega - \omega_{\mathrm{sp}})$$

Subscript sp means setpoint; each state is actual value minus setpoint.

Equivalent structure	Mapping conditions
Two-state PD	Fixed target: k_θ maps to K_{p}, and k_ω maps to K_{d} acting on the measurement
Feedback with integral action	Add an integral-of-error state, rebuild the augmented model, and redesign the gains
Cascaded PID	Check each loop's inputs, units, sampling, filtering, and control allocation

Model-based design provides a justified starting point; saturation, estimation, delay, and the implemented structure determine usability.

Chapter Outline

Control Performance Evaluation and Tuning

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

5.4.1 Control Performance Metrics

Textbook: Sec. 5.1.4, items 1–2

5.4.2 PID Parameters and Tuning Procedures

Textbook: Sec. 5.1.4, items 3–4; proportional–integral–derivative (PID)

5.4.3 Model-Based Parameter Design

Textbook: Sec. 5.1.4, item 5

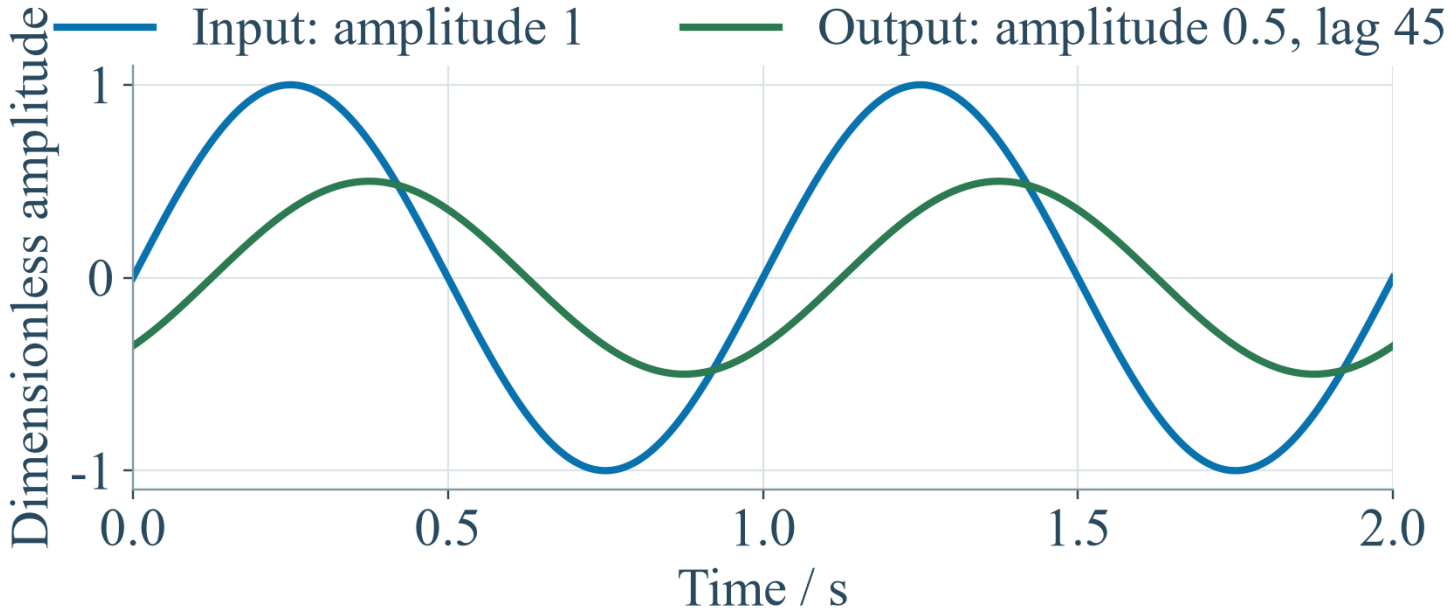
5.4.4 Whole-Vehicle Performance Tuning

Textbook: Sec. 5.1.4, item 6; plus frequency response and delay

5.4.4 Whole-Vehicle Performance Tuning: Introduction to Frequency Response

At One Frequency, Read Amplitude Ratio and Time Lag

Delay near crossover adds phase lag and reduces stability margins.



Sinusoid; see 4.4, “Convert Hz, rad/s, and Phase-Time Differences”.

1 Hz: 1 cycle per second.
Peak times differ by 0.125 s.

$$\frac{A_{\text{out}}}{A_{\text{in}}} = 0.5$$

$$20\log_{10}(0.5) = -6.02\ \text{dB}$$

$$\phi_{\text{lag}} = 360^\circ f\,\Delta t = 360^\circ \times 1 \times 0.125 = 45^\circ$$

Unity negative feedback: open loop $L(s)$, closed loop $T(s) = L(s)/(1 + L(s))$. Read stability margins on the open loop.

5.4.4 Whole-Vehicle Performance Tuning: Reading Phase Margin

A 20 ms Delay Costs 11.46° of Phase at Crossover

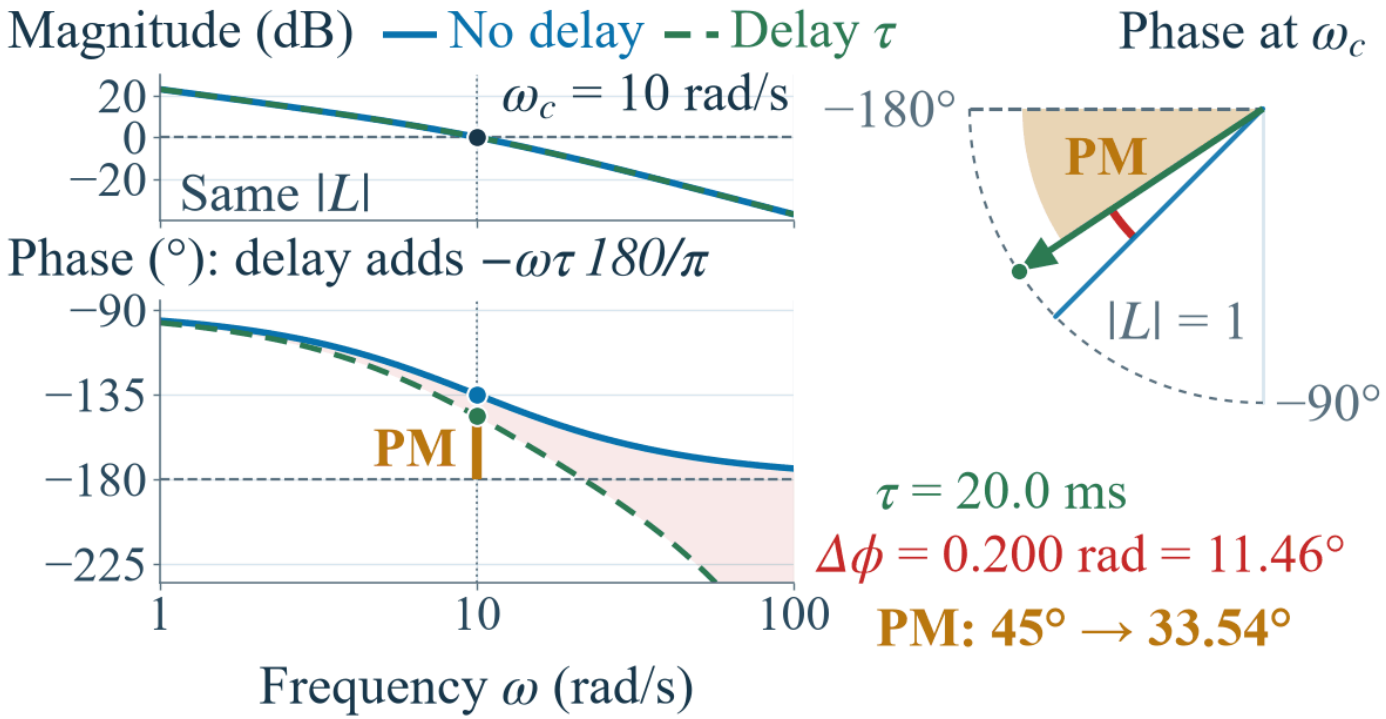


Figure: Unity negative feedback; open-loop example, delay 0→20 ms. Smaller phase margin (PM): closer to instability, usually more overshoot and oscillation.

e5.2: frequency sweep → Bode plot; read phase margin at the gain crossover

$$L(s) = \frac{100\sqrt{2}}{s(s+10)}$$

$$\omega_c = 10 \text{ rad/s}$$

$$\text{PM} = 180° - 135° = 45°$$

$$\Delta\phi = 10 \times 0.02 \times \frac{180°}{\pi}$$

$$= 11.46°$$

$$\text{PM}_{\text{delay}} = 33.54°$$

5.4.4 Whole-Vehicle Performance Tuning: Checking Structure, Sensing, and Actuators

Check Structure, Vibration Isolation, and Filtering Before Gains

① Structural coupling
Evidence: fixed-frequency oscillation varies by airframe
Check: rigidity, fasteners, and resonance

→

② Vibration isolation / thermal drift
Evidence: IMU (inertial measurement unit) noise/bias varies by environment
Check: mounting, vibration, and temperature

→

③ Filtering and state estimation
Evidence: smooth state estimate, but clear lag behind target
Check: timestamps of raw and estimated data

④ Scheduling / communication
Evidence: uneven timestamp intervals or large data age
Check: cycle/execution times and dropped samples

→

⑤ Actuators
Evidence: requests change, but actual output stays at its limit
Check: allocation headroom and output mapping

→

⑥ Model / operating point
Evidence: tracking degrades markedly after a load change
Check: parameters, trim, and identification

Classroom Exercise

When the Response Slows, Should You Increase Gain First?

Task: Hide the last two columns; use the logs to propose a cause and one distinguishing test.

Given log evidence	Initial diagnosis	Suggested diagnostic check
A: feedback lags by 60 ms	Excessive filtering or stale data	Align raw, estimated, and target timestamps
B: output stays at its limit	Actuator authority or saturation bottleneck	Compare requested, allocated, and actual outputs
C: small input, timely feedback	Insufficient control action is a possible cause	Increase P in small steps under the same conditions and retest

Key points: check feedback reliability, then actuator feasibility, then the direction of gain changes. Support each diagnosis with an observation or comparison that narrows down the cause.

Section Summary

Conditions and Evidence Determine the Tuning Direction

Metric conventions: specify the rise interval, overshoot denominator, and tolerance band.

Gain comparisons: fix the plant, conditions, and constraints; report output and input together. Stop at the targets, then validate in stages.

Model methods: pole placement specifies poles; LQR specifies a cost. Check the control structure before mapping gains to PID.

Whole-vehicle diagnosis: check structure, vibration isolation, and filtering first; delay reduces phase margin at crossover.

Common errors: treating one improvement as a full pass, directly comparing costs with different weights, and applying linear pole conclusions after saturation.