INTERACTIVE CONTROL ENGINEERING
Principles and Applications

Volume 7 Nonlinear Dynamics, Oscillations, and Stabilization

CHAPTER 3

Aeropendulum Simulator in Which the Setpoint Chooses the Plant

Pages 53–76

Edited by John D. Hedengren

How to cite this chapter. "Aeropendulum Simulator in Which the Setpoint Chooses the Plant," in *Nonlinear Dynamics, Oscillations, and Stabilization*, John D. Hedengren, Ed., vol. 7 of *Interactive Control Engineering: Principles and Applications*. Advanced Process Solutions, LLC, 2026, ch. 3, pp. 53–76.

The page numbers in this file are those of the printed volume.

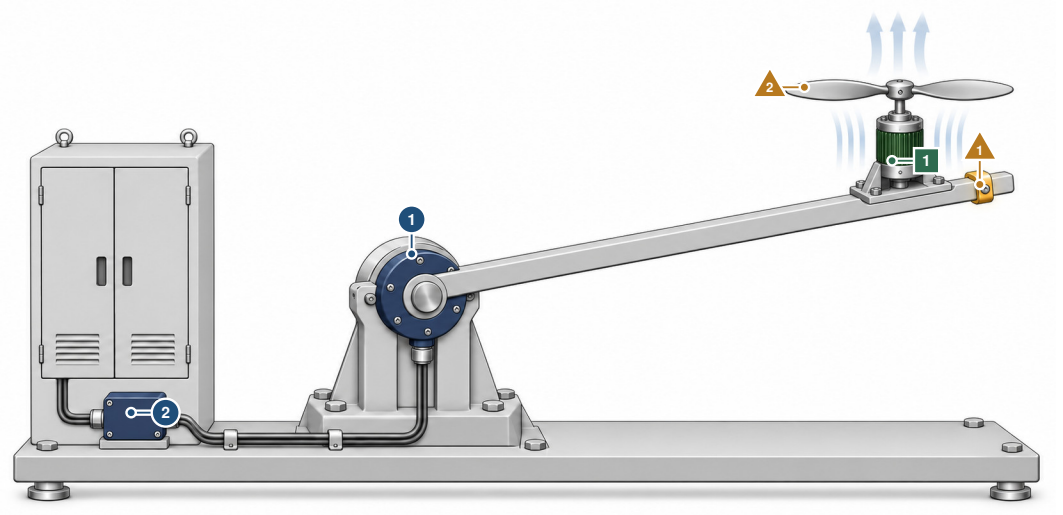
The app for this chapter
https://apmonitor.com/apps/aeropendulum/

The volume online: contents, other chapters and citation records
https://apmonitor.com/apps/books/v07-nonlinear-stabilization/#aeropendulum

Copyright © 2026 Advanced Process Solutions, LLC. All rights reserved. Produced by Advanced Process Solutions, LLC.

CHAPTER 3

Aeropendulum Simulator in Which the Setpoint Chooses the Plant



The controller moves 1 motor and propeller, throttle sets their speed

The controller reads 1 encoder at the pivot, arm angle 2 battery voltage

What disturbs it 1 clip added at the tip 2 chipped propeller blade

Notice that the propeller sits at the far end of the arm from the encoder that reads its angle, and that the arm here is above level, where gravity pulls it away from the setpoint. Illustration, not to scale.

IN PLAIN TERMS **A Propeller Arm That Changes Its Nature**

The main idea. A small propeller at the end of a light arm lifts it about a pivot to the angle the controller asks for. Below level gravity pulls the arm back like a swing, but above level it pulls the arm farther away.

The challenge. So a controller set up at one angle can fail at another, even while its usual stability check looks fine. The propeller can only push, and once its push runs out no controller can hold the arm.

Why it matters. Aircraft, engines and many plants change their behavior with the point where they run in the same way.

THREE WAYS TO READ THIS CHAPTER

Explore	12 min	the first page, the fact on the abort vehicle (Section 3.1), the history in Section 3.2, the two try-its and the fact on the setpoint program in Section 3.6, and the key points
Study	45 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	15 min	the first page, the results (Section 3.6), Technology in Practice (Section 3.7), Limitations (Section 3.8) and the open challenge

The control question Gravity gives a propeller arm a stiffness $G\cos\theta$ that is positive below horizontal, zero at it and negative above it, so the setpoint decides whether the plant is a pendulum, an integrator or an unstable equilibrium. What can a PID tuned at one angle do at another, and which remedy repairs which failure?

Learning objectives

- Derive the arm's stiffness, static map and actuator slope at an operating angle, and the stall angle when full thrust runs short.
- Apply the Routh condition of a PID on the linearized arm to find the angle at which a fixed tuning goes unstable.
- Explain why a healthy phase margin misses a slow mode that loses its damping.
- Compute the gains of a schedule that holds the closed-loop polynomial, and compare it with feedback linearization.
- Distinguish windup at an actuator limit from windup with nothing at a limit, and choose between back-calculation and an integral zone.

Prerequisites Sections 1.4.2, 1.4.3 and 1.4.7 (fold, schedule, feedback linearization, limits of an unstable pole), and Sections 1.6.1 and 1.6.2 (engine and velocity-form PID). Chapter 2 for a plant whose gain changes sign.

The app https://apmonitor.com/apps/aeropendulum/

See also Volume 1, Chapter 5 for an air-driven actuator that holds a weight on a feedforward, and Volume 1, Chapter 4 for a microcontroller kit with its own sample time. Volume 2, Chapter 2 for a step test that depends on the operating point, Volume 2, Chapter 6 for gain scheduling and Volume 2, Chapter 3 for back-calculation at an actuator bound.

3.1 Introduction

Small electric multirotor vehicles hold themselves up on fixed-pitch rotors, and the usual way to command their thrust is to command the rotor's speed [1]. A brushless hobby motor driving such a rotor is a plant in its own right, identified from step excitation and bounded by its supply voltage [2]. The smallest machine built on this actuator is the aeropendulum, a light rod on a pivot with a motor and propeller at its free end, which has served as a low-cost experiment for classical and modern control [3] and as an open-source shield for microcontroller boards [4]. Its thrust pushes the arm one way only, and gravity decides what each push is worth.

The setpoint chooses the plant. Linearized at an angle θ^* from hanging, the weight torque on the arm changes by $G\cos\theta^*$ per radian, so in the simulator of this chapter the arm has a natural frequency of 5.95 rad/s hanging and 2.48 rad/s at 80°, behaves as an integrator at horizontal, where holding it takes 72.97 % of the throttle, and runs away from its equilibrium at 120° at +4.06 s^{-1}. A proportional-integral-derivative (PID) loop tuned at 30° shows what that does to a tuning. It settles a 3° step there with 5 % overshoot and carries the arm to 80°, and its linearized loop keeps a phase margin between 49 and 55° from 15 to 105°. The same loop is unstable from 105.8°, and a step to 120° puts the arm on its top stop 1.40 s later. The mode that escapes is a slow one near 1.6 rad/s, far below the crossover near 10 rad/s where the margin is read, and its damping ratio falls from 0.53 at 30° to 0.10 at 100°. The plant corrects a common misreading, that a healthy margin at the crossover certifies a tuning. A gain schedule reverses the picture, holding the damping between 0.38 and 0.53 while its phase margin falls to 19° at 150°.

The difficulty belongs to a general class, plants whose gain, time constants or stability move with the operating point. Gain scheduling, which builds a family of linear designs and indexes them by a measured variable, had its early applications in flight control and automotive engine control before research turned to linearization-based scheduling and linear parameter-varying designs [5], and it has been applied in fields from aerospace to process control [6]. The neutralizer of Volume 2, Chapter 6 schedules its controller on a gain that moves by orders of magnitude across one valve stroke. An

airframe flown across its envelope and an engine run across its speed range share the arm's shape, a nonlinear plant whose linearization has to be redone wherever the loop goes. A model identified at one setpoint describes that setpoint only, which case 2 below measures.

> ✦ DID YOU KNOW? **Unstable for Most of an Abort**
>
> A crew capsule's launch abort vehicle must abort safely at any time during the atmospheric ascent, and its aerodynamics leave it unstable for most of the abort flight. Its controller gains are designed on linear models and scheduled in time, altitude and Mach number, then tested in nonlinear simulation [7].

A study of one designed law on one rig, such as an H_∞ controller about an identified operating point [8] or model reference control of a linear parameter-varying model [9], leaves open the angle at which a simpler tuning stops working and the mode through which it fails, and on hardware finding them ends on the stop. This chapter works out, on a simulated desktop kit whose every equation and constant is given here, how the sign of one stiffness decides what a tuning made at one angle can do at another, and which standard remedy repairs which failure. The model, the PID, the feedforward, the schedule and feedback linearization are all standard methods. Section 3.2 traces the rig and its ideas. Section 3.3 gives the kit as modeled, with its quantized encoder and throttle and a one-sided motor lag. Section 3.4 linearizes the arm at every angle. It replaces the one-sided lag by its mean within 1.3 % in gain and 1.8° in phase, and it derives a closed form, the Routh condition of the PID on the lag-free linearized arm, whose edge lies within 0.3° of the sampled loop's at four proportional gains and which places the escaping mode near $1/\sqrt{\tau_I \tau_D}$. Section 3.5 gives the controller as coded, with a loop view computed from the eigenvalues of the sampled closed loop and checked against the running arm. Section 3.6 reports the nine cases and the graded program as measured, one remedy at a time, with one case that no controller can meet. Section 3.7 describes the sensors, drives and controllers of real rigs, Section 3.8 states what the model leaves out, and Section 3.9 sums up.

3.2 Propeller Arms on the Control Bench

Control courses have used the aeropendulum as a hands-on experiment. A small dc motor at one end of a light rod drove a 2 in propeller, a potentiometer at the pivot measured the angle, and a custom board connected the rig through a virtual serial port to a MATLAB/Simulink module sampling at up to 200 Hz [3]. On it senior mechanical engineering students designed PID and lead-lag controllers, and state-space controllers combined with feedback linearization, in a semester of assignments that needed neither a specialized laboratory nor teaching assistants [3].

The aeropendulum then became an object of dynamics research. Digitization and time delay modify the stability of feedback-controlled systems, and an analytical, numerical and experimental study of the aeropendulum estimated its parameters from a nonlinear model, defining damping coefficients of order zero, one and two, and compared the stability charts of several controllers, finding the larger stable region for an act-and-wait controller [10]. The pivot of the simulator keeps the first two orders of damping and leaves the third out, as Section 3.3.5 states.

Home-made and open-source versions followed. Recommendations for building a home-made prototype from chosen structural materials and electronics, with ideas for identifying a linear model and observing stability, were offered so that a working system could be kept at home [11]. An open-source reference design then packaged the aeropendulum as a swappable shield for Arduino boards, built from off-the-shelf electronics and 3D-printed parts, with an interface for the Arduino IDE, MATLAB and Simulink and classroom examples of both its use and its limitations [4]. The Temperature Control Lab of Volume 1, Chapter 4 is the same idea in heat transfer, a microcontroller benchmark whose cycle time, discrete sampling, communication overhead and model mismatch are its point [12]. The kit simulated here keeps a microcontroller's sample time, its 8-bit throttle and its 12-bit encoder.

Designs for the rig grew past the single PID. A prototype identified with the identification tools of MATLAB was stabilized about an operating point by an H_∞ controller and made to track a square wave of angle [8], and model reference controllers for linear parameter-varying systems, designed with a parameter-dependent Lyapunov function, were evaluated on an aeropendulum in real time [9].

Behind these designs stand two older ideas. Gain scheduling had proven to be a successful methodology before an analysis could guarantee its robustness, performance or even nominal stability, and the analysis published in 1990 treated scheduling on a reference trajectory and scheduling on the plant output, confirming that the scheduling variable should vary slowly [13]. The arm's schedule can follow either the setpoint or the angle the encoder reads, and case 5 measures the difference. Feedback linearization grew from conditions under which a nonlinear system is equivalent to a controllable linear one [14], and within a decade it was a textbook method [15]. On the aeropendulum it was taught on the hands-on rig of 2012 [3], and an integrator against constant disturbances has since been added to it, with the integrator's initial condition chosen so that the closed loop keeps its intended response [16].

> ⧖ HISTORY **Inverting the Airframe, Mid-1980s to Late 1990s**
>
> In aircraft flight control, feedback linearization goes by the name dynamic inversion [17]. It joins the vehicle's nonlinear equations of motion with onboard models of its aerodynamics, mass properties and engine to find the control combination that gives a commanded motion, while outer loops shape the response and give robustness to model errors. In the mid-1980s it was a new theoretical approach, proposed for a research aircraft flying at high angle of attack in the nonlinear, uncertain region past the stall. Fitting its laws into the very limited computers of the time was a significant challenge, and that design never flew. In the late 1990s a crew rescue vehicle program chose it because a change in the vehicle's shape needed only an update of the onboard aerodynamic model. A production aircraft now flies it from conventional flight to hover [18].

3.3 The Kit as Modeled

The simulated kit is a uniform carbon rod of mass m_r and length L hanging from a pivot on a desk stand, with a brushless motor and a three-inch propeller, of mass m_m together, at l_p from the pivot, as shown in Figure 3.1. The propeller blows sideways in the plane of the swing, so its thrust acts perpendicular to the arm and only toward larger angles. The angle θ is measured from hanging, 0 down, 90° horizontal and 180° up, and the three integrated states are θ, the rate $\dot{\theta}$ and the propeller speed W. The rig and every threshold are teaching values sized like the desktop kits that courses build around a microcontroller, not the data sheet of a product, and the application states this in its model panel and on its Instructions page. Table 3.1 sorts the signals by role.

3.3.1 The Arm About Its Pivot

The rod is uniform and the motor and propeller are a point mass, so the moment balance about the pivot is Equation (3.1).

$$J\,\ddot{\theta} = l_p T - G\sin\theta - b_f\,\dot{\theta} - \tau_{c,f}\,\operatorname{sgn}\dot{\theta} + \tau_{fan}(t) \tag{3.1}$$

where T is the thrust and the inertia and the weight torque at horizontal are given by Equation (3.2).

$$\begin{aligned} J &= \tfrac{1}{3}m_r L^2 + m_m l_p^2 + m_c L^2 \\ G &= g\left(\tfrac{1}{2}m_r L + m_m l_p + m_c L\right) \end{aligned} \tag{3.2}$$

Here m_c is a clip at the tip, $b_f = b\,(1 + f/100)$ and $\tau_{c,f} = \tau_c(1 + f/100)$ are the pivot's viscous and Coulomb friction raised by the friction disturbance f, and $\tau_{fan}(t) = \tau_0 + \tau_1 \sin(2\pi t/p_{fan})$ is the desk fan, a steady push and a slow swing of period p_{fan}, positive toward larger angles. Every term is in N m. The defaults give $J = 0.003\,383\,\mathrm{kg\,m^2}$ and $G = 0.1197\,\mathrm{N\,m}$, and the hanging swing has a damping ratio of $b/2\sqrt{GJ} = 0.025$. Within one integration step the Coulomb torque keeps the

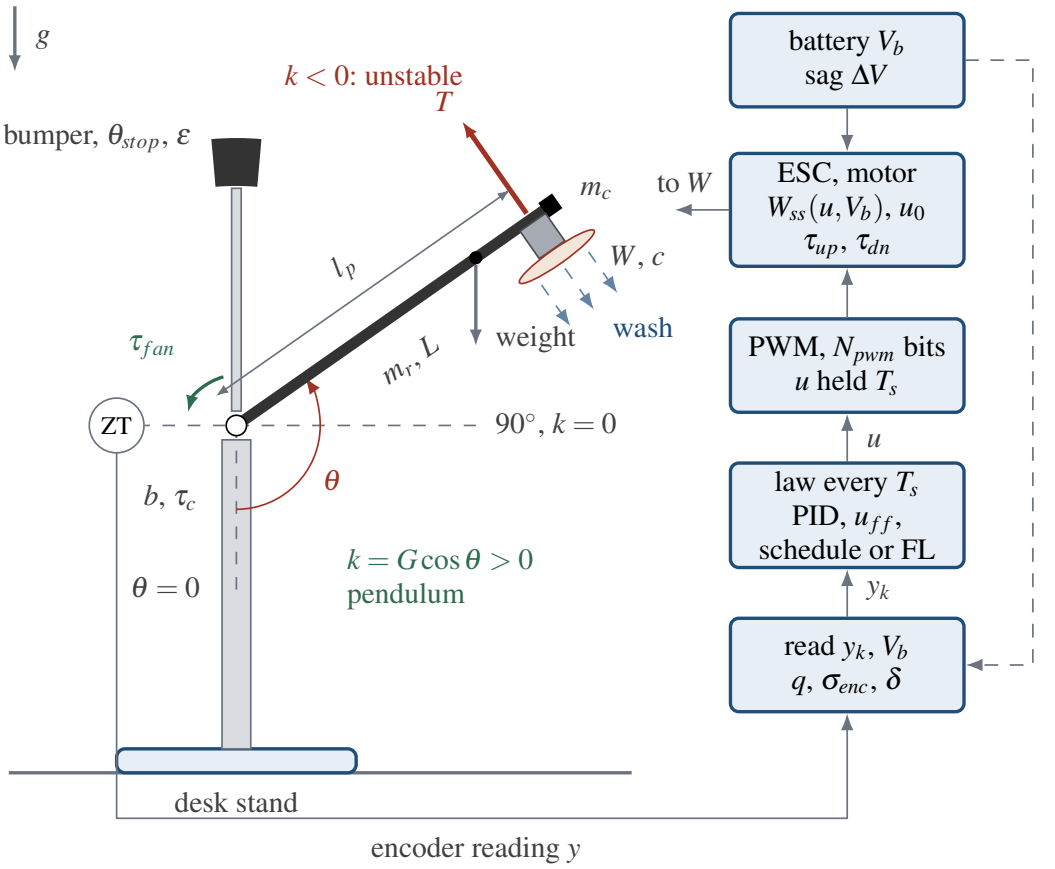


Figure 3.1 The simulated desktop aeropendulum and the kit's chain from the encoder to the propeller, not to scale. The arm is drawn at $\theta = 125°$ from hanging, past horizontal, where the stiffness $k = G\cos\theta$ is negative. A uniform rod (m_r, L) carries the motor and a pusher propeller at l_p, whose thrust T acts perpendicular to the arm toward larger θ while its wash streams the other way. The weight acts at the center of mass, a clip m_c can be added at the tip, and a chip removes the share c of the thrust. The pivot has viscous and Coulomb friction b and τ_c, a desk fan adds the torque τ_{fan}, and a rubber bumper on a bracket stops the arm at θ_{stop} with restitution ε. Every T_s the controller reads the encoder ZT (count q, noise σ_{enc}, offset δ) and the battery V_b and writes an N_{pwm}-bit duty u, which the speed controller and motor turn into the propeller speed W through the dead band u_0 and the lags τ_{up} and τ_{dn}.

sign it had when the step began, which keeps the right-hand side smooth. Between steps a rate that has changed sign is set to zero, and the arm then stays put until the net torque on it exceeds $\tau_{c,f}$. A rubber bumper above the pivot blocks the top 10° of the circle. An arm found at $|\theta| \geq \theta_{stop} = 175°$ is returned to the stop, and an arrival faster than $\omega_{rest} = 0.05\,\mathrm{rad/s}$ is reversed with restitution $\varepsilon = 0.3$ and counted as a hit, while a slower one is held.

Table 3.1 Signals of the Kit by Role

Symbol	Role	Nominal	Range
θ	CV and state, arm angle	per case	SP 0 to 170°
$\dot{\theta}$	state, arm rate	0	—
W	state, propeller speed	holds θ	0 to 20 000 rpm
u	MV, throttle duty	holds θ	0 to 100 %, 255 counts
y	measured, encoder angle	θ	4096 counts a turn
V_b	measured, battery	7.4 V	6.0 to 8.4 V
τ_{fan}	DV, desk fan, push and swing	0	±15, 0 to 15 mN m
m_c	DV, clip at the tip	0	0 to 10 g
ΔV	DV, battery sag	0	0 to 1.4 V
c	DV, thrust lost to a chip	0	0 to 50 %
$\Delta\dot{\theta}$	DV, tap on the arm	—	+30 °/s event
δ	DV, encoder offset	0	±10°
f	DV, pivot friction rise	0	0 to 500 %
T, T^*, T_f	outputs, thrust, held, full	—	—
k	output, stiffness $G\cos\theta$	—	±0.120 N m/rad
ζ, PM	outputs, loop at the setpoint	—	—

3.3.2 A Propeller That Only Pushes

The electronic speed controller drives the motor up and does not brake it, so the propeller speed follows its steady value through the one-sided lag of Equation (3.3).

$$\tau\dot{W} = W_{ss} - W, \qquad \tau = \begin{cases} \tau_{up}, & W_{ss} > W \\ \tau_{dn}, & \text{otherwise} \end{cases}$$
$$W_{ss} = W_{max}\,\frac{V_b}{V_0}\,\frac{\max(0,\ u - u_0)}{100 - u_0} \tag{3.3}$$

where u is the duty in percent, $u_0 = 8\,\%$ the dead band, $\tau_{up} = 30\,\text{ms}$ the spin-up lag, $\tau_{dn} = 75\,\text{ms}$ the coast-down lag, and $W_{max} = 20\,000\,\text{rpm}$ the full-throttle speed at the rated $V_0 = 7.4\,\text{V}$. The battery voltage is $V_b = V_{bat} - \Delta V$. The thrust is the square of the speed, as shown in Equation (3.4).

$$T = k_T(1-c)\,W^2, \qquad k_T = \frac{T_{max}}{W_{max}^2}$$
$$T_f = T_{max}\left(\frac{V_b}{V_0}\right)^2(1-c) \tag{3.4}$$

where c is the share of thrust a chipped propeller has lost, $T_{max} = 0.80\,\text{N}$ is full thrust on a fresh pack, $k_T = 1.824 \times 10^{-7}\,\text{N s}^2/\text{rad}^2$, and T_f is full thrust as the battery and the propeller now allow. The thrust is never negative, and below the dead band the propeller's steady speed is zero.

Holding the arm still at θ^* takes the thrust and the throttle of Equation (3.5), which is the static map of the rig.

$$T^* = \frac{G\sin\theta^*}{l_p}, \qquad u^* = u_0 + (100 - u_0)\sqrt{T^*/T_f}$$
$$\theta_{stall} = \arcsin\frac{T_f\,l_p}{G} \tag{3.5}$$

The held thrust peaks at horizontal, 0.3989 N or 72.97 % of the throttle, so the same throttle holds θ^* and $180° - \theta^*$, 68.46 % for both 60° and 120°. When full thrust falls below the thrust that holds the arm horizontal, the angles around 90° are not equilibria at all, and the highest angle that full throttle holds is the stall angle. Horizontal is the fold of Section 1.4.2, the angle where the holding input is largest, and each throttle below the peak holds two equilibria, one stable and one not.

Table 3.2 Model Constants. Class V Is a Slider in the Model Dialog and H a Hidden Constant

Symbol	Meaning	Value	Range	
m_r	rod mass	0.020 kg	0.005 to 0.08	V
L	rod length	0.32 m	0.2 to 0.5	V
m_m	motor and propeller	0.030 kg	0.01 to 0.08	V
l_p	motor radius	0.30 m	0.15 to 0.5	V
b	pivot viscous friction	1.0 mN m s/rad	0 to 10	V
τ_c	pivot Coulomb friction	0.05 mN m	0 to 1	V
g	gravity	9.81 m/s^2	1.6 to 24.8	V
T_{max}	full thrust at V_0	0.80 N	0.3 to 2	V
u_0	throttle dead band	8 %	0 to 20	V
τ_{up}	spin-up lag	30 ms	5 to 200	V
τ_{dn}	coast-down lag	75 ms	5 to 400	V
W_{max}	full speed at V_0	20 000 rpm	fixed	H
V_0	rated voltage, two cells	7.4 V	fixed	H
V_{bat}	battery voltage	7.4 V	6.0 to 8.4	V
θ_{stop}	top stop	175°	fixed	H
ε	bumper restitution	0.3	0 to 0.9	V
ω_{rest}	resting-contact rate	0.05 rad/s	fixed	H
N_{enc}	encoder resolution	12 bit	8 to 16	V
σ_{enc}	encoder noise	0.5 counts	0 to 3	V
N_{pwm}	throttle resolution	8 bit	6 to 12	V
J, G	inertia, weight torque	0.003383, 0.1197	derived	
T_{90}^*	thrust to hold 90°	0.3989 N	derived	

3.3.3 What the Microcontroller Reads and Writes

The controller never reads the true angle, the propeller speed, the thrust, a clip, a fan or a chip. Every sample interval T_s it reads the magnetic encoder at the pivot and the battery, as shown in Equation (3.6).

$$y_k = q\,\mathrm{rnd}\Big(\frac{\theta(t_k)+\delta}{q} + \sigma_{enc}\,n_k\Big), \qquad q = \frac{2\pi}{2^{N_{enc}}} \tag{3.6}$$

where rnd rounds to the nearest integer, n_k is a standard normal deviate from a generator seeded at 20261001, $\sigma_{enc} = 0.5$ counts, δ is the offset of a slipped magnet, and $N_{enc} = 12$, so one count is 0.0879°. The battery reading is exact. The duty the controller writes is rounded to the throttle's $N_{pwm} = 8$ bits and held until the next sample, as shown in Equation (3.7).

$$u_k = \frac{100}{N}\,\mathrm{rnd}\Big(\frac{N u_{cmd}}{100}\Big), \qquad N = 2^{N_{pwm}} - 1 \tag{3.7}$$

One count is 0.392 % of the throttle. Near horizontal that is a large step, because count 185 holds 80.8°, count 186 holds 87.7° and count 187 holds no angle at all.

3.3.4 Constants, Initial States and Time Stepping

Table 3.2 lists the constants. The framework's engine (Section 1.6.1) integrates the three states with fourth-order Runge–Kutta steps of 1 ms, calls the controller on that clock, which runs its law once T_s has elapsed, and logs at 50 Hz. The step specific to this app is the friction latch and the top stop, applied after each integration and before the outputs are computed, and a run plays at 0.25 to 5 times real time. A case starts with the arm hanging, at rest on its setpoint with the propeller at the speed that holds it, or at rest where the manual throttle, rounded to the throttle's counts, holds it. The encoder noise is seeded, so every run replays exactly, and Reset and the reload button act as in every framework app. The application passes its 40 verification checks, among them energy

conserved to 1.8×10^{-11} of G over 10 s without thrust or friction, swing frequencies within 0.04 % of $\sqrt{G\cos\theta^*/J}$, and, with the encoder noise off, halving the step moving the overshoot of case 3 by 0.0072°.

3.3.5 Assumptions

One rigid arm in a vertical plane. The rod does not bend, the stand does not move, and the motor and propeller are a point mass at l_p.

A pivot with two orders of damping. The pivot's friction is viscous plus Coulomb, with no static level above the sliding one. The quadratic, second-order damping that model-based estimation found on a physical aeropendulum [10] is absent, and a real bearing adds stiction that makes an arm stick and slip near its setpoint.

Thrust from speed alone. The thrust is $k_T W^2$ with a constant coefficient, with no correction for the air that the arm's own motion moves through the rotor, no wash on the arm and no wall effect. On multirotors, thrust control has been proposed in place of speed control because a rotor held at one speed keeps its thrust poorly in updrafts and downdrafts [1]. On a swinging arm the same inflow would add damping beyond the pivot's damping ratio of 0.025.

A first-order motor on an ideal battery. The steady speed is linear in the throttle above the dead band and in the voltage, the speed controller has no current limit or dynamics of its own, the motor does not heat, and the sag is imposed as a disturbance instead of following from the current drawn. The lags are 30 and 75 ms, half the 60 and 150 ms that the application's plan proposed. With the plan's values the best of 360 PID tunings on the application's own loop analysis damps the arm at 30° to only $\zeta = 0.29$, and halved the same search reaches 0.57, so that the cases show a damping that changes with the angle rather than one that is light everywhere.

A fan that pushes the same way at every angle. The fan is a torque about the pivot independent of where the arm points, while a real fan's push depends on the angle and is gusty.

A kit that computes instantly. The encoder adds Gaussian noise of half a count before rounding and has no eccentricity, the throttle only rounds, and the controller computes at the sampling instant with no jitter.

3.4 One Arm Linearized at Every Angle

The linearizations of this section are the analysis model behind the application's Loop view. They are built on the true rig with the controller exactly as it runs, while Equations (3.1) to (3.7) remain the simulation model and the declared rig of Section 3.5 remains the controller's own model, which knows the measured battery but no clip, chip or added friction.

3.4.1 Stiffness, Static Map and Actuator Slope

About an equilibrium θ^* held by T^*, small deviations of the throttle reach the angle through the transfer function of Equation (3.8).

$$P(s) = \frac{\Delta\theta(s)}{\Delta u(s)} = \frac{K_u}{(\bar{\tau}s+1)(Js^2+bs+k)} \tag{3.8}$$

where $\bar{\tau}$ is the motor's lag for small swings (Section 3.4.2), $k = G\cos\theta^*$ in N m/rad is the arm's stiffness and $K_u = 2l_p\sqrt{T^*T_f}/(100-u_0)$ in N m per percent is the slope of the static map, the throttle's grip on the thrust torque. Below horizontal $k > 0$ and the arm is a lightly damped pendulum with natural frequency $\sqrt{k/J}$, 5.95 rad/s hanging, 5.54 rad/s at 30° and 2.48 rad/s at 80°. At horizontal $k = 0$ and the arm is an integrator with a slow pole at $-b/J = -0.30\,\mathrm{s}^{-1}$. Above horizontal $k < 0$ and the equilibrium has a root at $-b/2J + \sqrt{b^2/4J^2 - k/J}$, $+4.06\,\mathrm{s}^{-1}$ at 120° and $+5.39\,\mathrm{s}^{-1}$ at 150° (4.21 and $5.54\,\mathrm{s}^{-1}$ without the pivot's friction), so it is unstable by the linearization alone [19]. The slope K_u grows with the square root of the thrust held, from 2.61 mN m per percent at 30° to 3.68 mN m at 90°, so the throttle has the least grip near hanging. The static gain K_u/k grows without bound as the

arm approaches horizontal, which is why a single count of the throttle moves the arm 6.9° between counts 185 and 186. The application's Map view draws these poles against the setpoint.

3.4.2 The Motor's Two Lags as One

The one-sided lag of Equation (3.3) is positively homogeneous in its input, so its describing function depends on frequency alone. Computed by integrating the lag through twelve periods of a sinusoid, it lies within 1.27 % in gain and 1.79° in phase of a single lag at the mean $\bar{\tau} = (\tau_{up} + \tau_{dn})/2 = 52.5\,\text{ms}$ from 1 to 30 rad/s, the worst at 30 rad/s, which is the lag the analysis uses. The one-sided lag still matters for large moves, where the propeller coasts from 13 065 rpm toward 1244 rpm in case 6 while gravity, not the motor, brings the arm down.

3.4.3 The Edge of a Fixed Tuning

Without the motor lag and the sampling, a PID with gains K_c, $K_i = K_c/\tau_I$ and $K_d = K_c\tau_D$ on the linearized arm closes the loop of Equation (3.9), whose Routh condition is Equation (3.10).

$$J s^3 + (b + K_d K_u)\, s^2 + (k + K_c K_u)\, s + K_i K_u = 0 \tag{3.9}$$

$$(b + K_d K_u)(k + K_c K_u) > J K_i K_u \tag{3.10}$$

Every term is in N m/s per radian. The derivative adds damping and the proportional gain adds stiffness to what gravity supplies, while the integral asks for both. At the edge the cubic has roots at $\pm j\omega_0$ with $\omega_0^2 = (k + K_c K_u)/J$, and because b is small against $K_d K_u$ the condition reads approximately $k + K_c K_u > J/(\tau_I \tau_D)$ with $\omega_0 \approx 1/\sqrt{\tau_I \tau_D}$. The closed loop needs a total stiffness above the inertia divided by the product of the two times.

For the shipped tuning, $K_c = 0.2\,\%/°$, $\tau_I = 0.4\,\text{s}$ and $\tau_D = 1.0\,\text{s}$, the right side of Equation (3.10) divided by $b + K_d K_u$ is 0.0082 to 0.0083 N m/rad from 30 to 120°, close to $J/(\tau_I \tau_D) = 0.0085\,\text{N m/rad}$, while $K_c K_u$ is 0.0299 N m/rad at 30° and 0.0422 N m/rad at 90°. Below horizontal gravity's stiffness, up to 0.120 N m/rad hanging, supplies most of the left side. Past horizontal it subtracts, and the condition fails at 106.07°, against 105.79° for the sampled loop with the motor lag and the hold, with the escaping pair at 1.56 rad/s against 1.61 rad/s, and $1/\sqrt{\tau_I \tau_D} = 1.58\,\text{rad/s}$. More proportional gain with the same times moves the edge, to 124.74, 140.62 and 160.70° for 0.4, 0.6 and 1.0 %/°, against 124.48, 140.40 and 160.57° for the sampled loop, which costs damping at 30° (0.36, 0.27 and 0.16 against 0.53). The lag-free condition places the sampled edge within 0.3° at all four gains, so at the shipped lag and sample time the edge is a property of the PID's structure against a negative stiffness rather than of the lag or the sampling.

3.4.4 Margin and Damping Part Ways

The Loop view decides stability from the eigenvalues of the sampled closed loop, the plant of Equation (3.8) held over each sample with the controller's states, and maps them by $s = \ln z/T_s$ to give each mode's damping ratio. It reports the least damping ratio ζ among the oscillating modes beside the phase margin at the last frequency where $|\mathscr{L}|$ falls through one, for $\mathscr{L}(j\omega) = C(\mathrm{e}^{j\omega T_s})H(j\omega)P(j\omega)$ with the controller as it runs and the hold H. Figure 3.2 draws both against the setpoint. The two answer different questions. Near the crossover, 9.3 to 11.4 rad/s from 15 to 105°, the arm's inertia term $J\omega_c^2$, 0.29 to 0.44 N m/rad, outweighs any stiffness, so the phase there barely depends on the angle, and the fixed tuning's margin stays between 49.2 and 55.4°. The slow mode lives where the stiffness counts. Up to 74° the least damped mode is the fast pair, at 7.6 to 12.6 rad/s with ζ between 0.48 and 0.53 from 15°, and from 75° it is the slow pair at 1.4 to 1.6 rad/s, whose ζ falls almost linearly to zero at 105.8°. From 100° on, the open loop crosses the negative real axis near 1.6 rad/s with a gain of 1.42 at 100°, 1.04 at 105° and 0.99 at 106°, so the loop is stable past horizontal only while it has gain to spare at that low frequency, a margin that no reading at the crossover reports.

🕮 **Pause.** Try Exercise 3.1 before reading on.

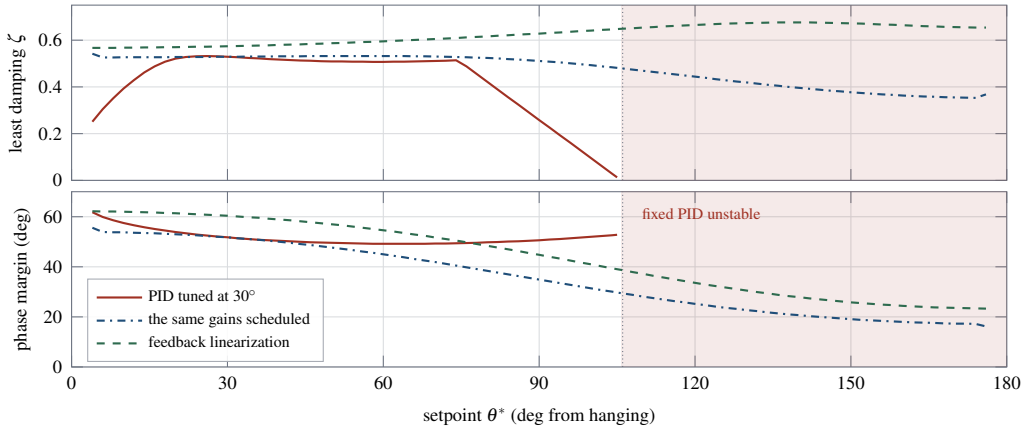


Figure 3.2 Least damping ratio and phase margin of the linearized sampled loop against the setpoint, from the application's own loop analysis at 10 ms samples. The PID tuned at 30° (solid) keeps its margin while its slow mode loses its damping and goes unstable at 105.8° (shaded), near the lag-free Routh edge at 106.07° (dotted vertical line). Scheduled (dash-dotted), the same gains keep ζ between 0.38 and 0.53 while the margin falls to 19.1° at 150°. Feedback linearization is dashed.

The schedule of Section 3.5 moves the curves the other way. It holds the three lower coefficients of Equation (3.9) at their values at the design angle, so the lag-free loop has the same poles at every angle and the sampled loop keeps ζ between 0.377 and 0.532 from 15 to 150°. To do so it raises the proportional gain to supply the stiffness that gravity takes away, and the phase margin at the crossover falls from 53.4 to 19.1°, a smaller delay margin that case 8 meets as a shorter allowable sample time. Feedback linearization keeps ζ between 0.569 and 0.676 with a margin falling from 61.7 to 25.8°.

3.5 The Kit's Controller as Coded

Table 3.3 lists the settings. The controller works from the encoder reading and the battery voltage alone, and its own model is the declared rig, the constants of Table 3.2 at the measured battery, with no clip, no chip and the nominal friction.

3.5.1 Sampling, Throttle and Saturation

The law executes when a whole T_s has passed on the 1 ms clock, reads Equation (3.6) and the battery, and writes a command that is clamped to 0 to 100 % and rounded by Equation (3.7). The saturation flag compares the law's unclamped output with those limits and is reset each sample, which makes a law that asks the propeller to pull read as saturated. The scorecard's throttle chatter is the root mean square change of the applied duty from one sample to the next in automatic mode, and the throttle has no rate limit beyond the motor's own lag. All laws act in the same direction, more throttle when the arm is below its setpoint, which the application's sign checks confirm for every structure.

3.5.2 The Shared PID on a Bias or a Feedforward

The PID is the framework's velocity-form algorithm of Section 1.6.2, written in this chapter's symbols in Equation (3.11), with T_s for the control period and the action positive.

$$
\begin{aligned}
e_k &= \theta_{SP} - y_k, \qquad P_k = K_c e_k \\
D_k &= -K_c \tau_D \left(y_{f,k} - y_{f,k-1}\right)/T_s \\
S_k &= S_{k-1} + \left(P_k - P_{k-1}\right) + \left(D_k - D_{k-1}\right) \\
\hat{u}_k &= u_b + S_k + I_{k-1}, \qquad u_{cmd} = \mathrm{clamp}(\hat{u}_k,\ 0,\ 100) \\
I_k &= I_{k-1} + K_i e_k T_s + \left(u_{cmd} - \hat{u}_k\right) T_s/T_t
\end{aligned}
\tag{3.11}
$$

Table 3.3 Controller Settings, Defaults and Ranges

Symbol	Setting	Default	Range
—	structure	PID	sched., FL
u_b	fixed bias, feedforward off	54 %	0 to 100
—	gravity feedforward	off	on, off
K_c	proportional gain	0.2 %/°	0 to 3
τ_I	integral time, 0 is off	0.4 s	0 to 20
τ_D	derivative time	1.0 s	0 to 2
τ_f	derivative filter	0 ms	0 to 200
T_s	sample time	10 ms	2 to 100
—	anti-windup	back-calc.	zone, off
e_I	integral zone	10°	1 to 30
θ_d	schedule design angle	30°	10 to 160
—	schedule variable	measured angle	setpoint
Δu	step-test size	2 %	0.4 to 10
ω_{FL}	outer frequency of FL	5 rad/s	1 to 15
ζ_{FL}	outer damping of FL	0.9	0.2 to 1.5
$\tau_{I,FL}$	outer integral time	2 s	0 to 20
θ_s	state feedback angle scale	3°	0.5 to 20
u_s	state feedback throttle scale	10 %	1 to 50
σ_q	observer speed	1	0.1 to 10

where y_f is the reading through an exact first-order lag, $y_{f,k} = a\,y_{f,k-1} + (1-a)\,y_k$ with $a = \mathrm{e}^{-T_s/\tau_f}$ ($a = 0$ for $\tau_f = 0$), $K_i = K_c/\tau_I$, and $T_t = \tau_I$ is the tracking time of the back-calculation. The shared PID's optional slew limit is not used, so the clamped value is the output. The running sum, the limit and the back-calculation behave as Section 1.6.2 describes. The gains act on errors in radians, so 0.2 %/° is 11.46 % per radian, and the integral is updated only while K_i is not zero. The bias u_b is either the fixed 54 % that holds 30°, or the gravity feedforward of Equation (3.12), the throttle that holds the setpoint on the inverted static map of the declared rig.

$$u_{ff} = u_0 + (100 - u_0)\sqrt{\frac{G \sin\theta_{SP}}{l_p T_{max} (\hat{V}_b/V_0)^2}} \tag{3.12}$$

where $\hat{V}_b$ is the measured battery, so the feedforward is 53.94 % at 30° and 72.47 % at 80°, and 100 % where the setpoint cannot be held.

3.5.3 Back-Calculation and the Integral Zone

Back-calculation acts only while the output is clamped, and on this plant a feedforward makes every large error a transient during which nothing need be at a limit. The integral zone also stops the integral while $|e_k| > e_I$, by setting K_i to zero for that sample, which clamps nothing, suspends the back-calculation with it and leaves the accumulated I as it was. Switching the integral by a condition on the error is itself a switching defense against windup, and a variable-structure PID method of that kind has been compared with several existing ones [20]. With anti-windup off, T_t is infinite and the integral integrates whatever the throttle does.

3.5.4 Schedule, Feedback Linearization and State Feedback

The schedule rescales the three gains tuned at the design angle θ_d so that the lag-free closed loop of Equation (3.9) keeps the polynomial it had there, as shown in Equation (3.13).

$$\begin{aligned} K_c' &= \max\left(0,\ \frac{K_c K_u(\theta_d) + k(\theta_d) - k}{K_u}\right) \\ K_i' &= K_i \frac{K_u(\theta_d)}{K_u}, \qquad K_d' = K_d \frac{K_u(\theta_d)}{K_u} \end{aligned} \tag{3.13}$$

where k and K_u are evaluated on the declared rig at the angle the encoder reads, or at the setpoint if the reader chooses, held between 5 and 175°, and $k(\theta_d)$ and $K_u(\theta_d)$ are their values at θ_d. The integral and derivative gains scale as $1/K_u$, and the proportional gain also makes up the stiffness that gravity loses, $k(\theta_d) - k$, which past horizontal is an anti-spring. It is the schedule of Section 1.4.3, which holds the closed-loop polynomial of the design point, applied to a PID on the lag-free arm. From the 30° tuning the proportional gain is 0.167, 0.375, 0.632, 0.984 and 1.589 %/° at 15, 60, 90, 120 and 150°. A schedule that only scales all three gains to hold the open loop's magnitude at one frequency cannot supply that stiffness. Applied to the same tuning with the frequency at 6, 8, 10 or 12 rad/s, it is unstable from 146, 124, 110 or 106°.

Feedback linearization asks an outer PID for an angular acceleration, adds the weight torque at the measured angle and inverts the thrust square on the declared rig, as shown in Equation (3.14).

$$\begin{aligned} \alpha &= \omega_{FL}^2 e + \frac{\omega_{FL}^2}{\tau_{I,FL}} \int e\,dt - 2\zeta_{FL}\omega_{FL}\dot{y}_f \\ T_{cmd} &= \frac{J\alpha + G\sin y}{l_p}, \qquad u = u_0 + (100 - u_0)\sqrt{T_{cmd}/T_f} \end{aligned} \tag{3.14}$$

The outer loop is the shared PID of Equation (3.11) with $K_c = \omega_{FL}^2$ and $\tau_D = 2\zeta_{FL}/\omega_{FL}$, and its limits are the accelerations that zero and full thrust give at the measured angle, so its anti-windup sees the throttle's limits. An advanced option replaces the chosen structure by a linear quadratic regulator with integral action and a steady-state Kalman estimator designed at the setpoint from the declared model, a teaching comparison computed by the framework's Riccati module and designed as in Section 1.4.5.

3.5.5 Step-Test Fit

In manual mode a step test moves the throttle by Δu, rounded to the counts, and fits the readings from 2 s before the step to 8 s after it with Equation (3.15) by output error.

$$y(t) = y_0 + K\Delta u\,\phi(t - t_d) + \mathrm{e}^{-\zeta\omega_n t}(A\cos\omega_d t + B\sin\omega_d t) \tag{3.15}$$

where ϕ is the unit step response of $\omega_n^2/(s^2 + 2\zeta\omega_n s + \omega_n^2)$, zero before the step, $\omega_d = \omega_n\sqrt{1-\zeta^2}$, and the last term is the free swing the arm already had. For each (ω_n, ζ, t_d) the four linear parameters come from a least-squares solve, and the three others from a grid with a coordinate search, t_d limited to 150 ms. A fit whose dead time ends on that limit says so in its note, which then reports the dead time as a bound rather than an estimate.

3.5.6 Manual Mode and the Models Kept Apart

Manual mode hands the throttle to the operator in 0.1 % steps, rounded to the counts, with a helper that sets the throttle the static map gives for the setpoint and two step-test buttons. In automatic mode the manual station shows u_{cmd} smoothed over 0.3 s, a switch to manual hands over the command in force, and every law tracks the throttle in force, so a change of mode or structure is bumpless. The simulation model runs the true rig, the controller's laws run the declared one, the Loop view analyzes the true rig linearized with the declared controller, and the step test identifies a fourth model, second order about

Table 3.4 Scenario Sequence

#	Change from the defaults	Teaching point
1	manual, the throttle ramped to counts 138, 175, toward 186, back to 175	no throttle holds past horizontal
2	manual, step tests at 20, 45 and 80°, feedforward, $\tau_f = 30$ ms	a model of the angle
3	steps 30 to 80 to 120°	a tuning carried too far
4	feedforward on, 30 to 80 to 30°	the integral freed, and loose
5	schedule and feedforward, 15, 60, 105, 150, 15°	one tuning everywhere
6	as 5, 120 to 20 to 120°	windup with no limit
7	as 5 with the zone at 100°, fan, clip, sag	what the model misses
8	as 7 at 140°, $\tau_f = 20$ ms, taps at 10, 40, 80 ms	sampling past horizontal
9	as 5, 6.6 V, chip 40 %, 60 to 90 to 60°	an unreachable setpoint
C	program 20 to 120° with gust, clip and fan	everything at once

Table 3.5 Outcomes Read From the Running Engine

Case and condition	Result
1, counts 138 and 175	29.2 to 31.5°, 59.3 to 62.1°
1, toward count 186	past 90° at 30.94 s, top stop at 32.83 s
2, fits at 20, 45, 80°	1.14, 2.31, 7.96 °/%; 5.72, 4.78, 2.77 rad/s
3, 30 to 80°	4.25° over, 3.57 s, integral 17.94 %
3, 80 to 120°	top stop 1.40 s after the step
4, feedforward on / off	integral −0.48 / 18.01 %, over 16.79 / 4.25°
5, schedule, four steps	3.40, 2.655, 1.01, 3.46 s, no stop
5, with the zone / on the setpoint	0.515 to 1.25 s / top stop at 14.39 s
6, back-calculation / zone	drop 2.94 s, I −8.13 / 0.96 s, −1.81
7, fan, clip, sag	102.7, 93.2, 99.5 to 100.1°
8, 10 / 40 / 80 ms	0.70, 1.42 s outside ±1° / lost
8, filter 0 / 20 ms, first 5 s	chatter 2.35 / 0.74 %
9, any structure	68.2 to 79.8°, stall 73.16°
9, return, aw on / off	0.80 s / not settled after 15 s
C, shipped / suggested	lost at 140° / worst 2.26 s, 9.2°

one angle. The loop view's stability agrees with the running arm on both sides of the edge, stable at 100° and lost at 110° after a 3° step, as the application's own check confirms. The eigenvalues come from the framework's balanced Hessenberg QR routine, `shared/src/framework/core/eig.js`.

3.6 Nine Cases and a Setpoint Program

Table 3.4 sets out the cases, each a departure from the settings of Table 3.3 together with any change the text names, and Table 3.5 holds what the engine measured. The engine was stepped outside the browser at the default seed, one 1 ms step at a time as the page steps it, and times count from the start of the run. A change the text names gives one run whether it is made before Run or carried through Reset. A settling time from the 1 ms record that falls on a half-hundredth of a second is given to the millisecond.

3.6.1 By Hand Past Horizontal

Case 1 hands the operator the throttle and moves it as a hand on the slider would. Ramped to count 138 of 255, 54.12 %, the arm lifts and settles between 29.2 and 31.5° about the 30.26° that count holds, swinging with a period of 1.1375 s, 5.52 rad/s, against the 5.53 rad/s that $\sqrt{G\cos\theta^*/J}$ predicts at that angle. Count 175 settles it between 59.3 and 62.1°. From 26 s the throttle creeps a count at a time toward count 186, the last whose thrust is below the 0.3989 N that holds the arm horizontal. The arm passes 80° at 30.08 s and 90° at 30.94 s with the duty at count 186, and nothing turns it back, because past horizontal the weight torque falls as the arm rises, and it reaches the top stop at 32.83 s.

Brought back to count 175 at 38 s, the throttle that held 60°, it stays on the bumper, where 175° needs only 0.035 N, and only a duty below 27.18 % would let it fall off.

> **TRY IT IN THE APP** **1 · By Hand: 30°, 60°, Then Past Horizontal**
>
> Load the scenario and press Run without touching the throttle. The arm settles near 30° and then 60°. From 26 s the throttle creeps up a count at a time, the arm passes 90° at 30.94 s, and nothing turns it back. It reaches the top stop at 32.83 s (Table 3.5) and stays there when the throttle that held 60° returns.

3.6.2 Step Tests That Find a Different Arm

Case 2 rests the arm at 19.9°, steps the throttle five counts, 1.961 %, and fits Equation (3.15), then lets the PID carry the arm to 45° and back to manual for the same test, and at 80° steps one count down. The fits find 1.14, 2.31 and 7.96 °/% with natural frequencies of 5.72, 4.78 and 2.77 rad/s, against 1.13, 2.24 and 9.01 °/% and 5.75, 4.89 and 2.62 rad/s for the arm linearized at the middle of each step. The gain grows 7.0 times from 20° to 80°, while the stiffness the fit implies, $J\omega_n^2$, falls 4.3 times against 4.8 for $G\cos\theta$ at the middle angles, because at 80° even one count moves the arm 3.12° and the fit averages a stiffness that changes by a third across the step. The 80° fit also runs its dead time to the 150 ms limit of the search, as its note reports, and a search bounded at 300 ms finds 170 ms while moving the gain by 0.08 % and the frequency by 0.5 %. Every fitted damping ratio lies between 0.041 and 0.082, and every model is a model of its angle. The cruise controller of Volume 2, Chapter 2 finds the same with road speed in place of angle.

3.6.3 A Tuning Carried Too Far

Case 3 holds 30° under the shipped PID and steps to 80° at 2 s and to 120° at 14 s. The 50-degree step overshoots 4.25° and settles within ±2° in 3.57 s, with the integral carrying 17.94 % of the throttle that the fixed bias leaves out. The step to 120°, inside the unstable band of Figure 3.2, puts the arm on the stop at 15.40 s, 1.40 s after the step (Table 3.5), and the trend of Figure 3.3 shows it on the bumper. By 30 s it has struck the bumper 12 times. On the running loop a 3° step overshoots 5 % at 30°, 24 % at 80° and 80 % at 100°, where the arm never settles within ±1°, and at 120° it is lost.

The Loop view says where and why. The damping ratio is 0.53 at 30°, 0.42 at 80°, 0.26 at 90° and 0.10 at 100°, and the loop is unstable from 105.8°, while the phase margin reads between 49 and 55° from 15 to 105°. The proportional gain is small because gravity's spring did most of its work at 30°, 0.104 N m/rad, and Equation (3.10) shows that past horizontal the proportional gain alone must outweigh both the negative stiffness and $J/(\tau_I\tau_D)$. Raising it to 0.4, 0.6 or 1.0 %/° moves the edge to 124.5, 140.4 or 160.6° and costs the design point its damping. Over the first 30 s the unfiltered derivative on a 12-bit encoder makes the duty chatter by 2.76 %.

> **TRY IT IN THE APP** **3 · A PID Tuned at 30°, Carried to 80° and 120°**
>
> Load the scenario, open the Loop view and press Run. The step to 80° overshoots 4.25° and settles. The right panel shades the band past 106° where the same tuning is unstable, and the step to 120° lands inside it: the arm reaches the top stop 1.40 s after the step (Table 3.5).

3.6.4 Feedforward, and the Integral It Leaves Loose

Case 4 runs the same PID on the feedforward of Equation (3.12). Just before the step back from 80° the integral holds −0.48 % of the throttle against 18.01 % on the fixed bias, so the bias now follows the setpoint and the integral does not have to. The step itself is worse, 16.79° of overshoot on the way up against 4.25° and 6.88° on the way back against 0.05°, because the integral still integrates the 50-degree error, to 7.94 % at its largest. With the bias already right, that is windup with nothing at a limit.

Pause. Try Exercise 3.2 before reading on.



Figure 3.3 The application in case 3, *A PID tuned at 30°, carried to 80° and 120°*, at $t = 20.00$ s on the Loop view with a 20 s trend window and no control changed, a composite of the Loop view (top) and the arm-angle trend (bottom) cut from one screenshot. (1) The badges read θ 126.6°, setpoint 120°, throttle 62.7 % and thrust 0.333 N, with loop and plant unstable. (2) The open loop at the 120° setpoint, marked unstable, with its crossover at 8.24 rad/s and a distance from −1 of 0.43. (3) The same tuning at every setpoint, the least damping ratio solid and the distance from −1 dashed, and the unstable band shaded and labeled from 106°. (4) The step to 80° settling and the step to 120° at 14 s, with the trend header reading an encoder angle of 125.9°. (5) The arm on the bumper until 16.8 s, then swinging down to 28.1° at 18.2 s and back up, values the screen does not print.

3.6.5 One Tuning From 15 to 150 Degrees

Case 5 schedules the 30° gains on the measured angle and walks the setpoint from 15° to 60, 105 and 150° and back to 15°, 6 s apart, on the feedforward. The four steps settle within ±2° in 3.40, 2.655, 1.01 and 3.46 s with overshoots of 9.0, 9.0, 17.2 and 8.8°, and the arm never touches the stop. The same feedforward under the fixed PID settles the first step in 2.45 s and no other, with 8 hits on the stop. The overshoots are the integral's, as in case 4, and with the integral zone the schedule settles the four steps in 0.65, 0.515, 1.25 and 1.09 s. Scheduled on the setpoint instead, the gains jump to the destination's at the step while the arm is still far from it, and the step to 150° overshoots onto the stop at 14.39 s, 0.39 s after the step, the hazard that scheduling on a variable which jumps invites [13]. Feedback linearization settles the steps in 3.72, 3.18, 2.85 and 5.32 s, and with the zone in 0.955, 0.79, 0.515 and 1.49 s.

3.6.6 A Propeller That Can Only Stop

Case 6 meets the windup of case 4 on the one-sided propeller. The scheduled PID holds 120° on the feedforward, the propeller at 13 065 rpm, until the setpoint drops to 20° at 2 s. The law asks for less than nothing for 0.160 s, the propeller coasts toward 1244 rpm, and gravity takes the arm down, which released with the motor off would pass 20° in 0.369 s. Back-calculation can act only during those 0.160 s, yet the integral reaches −8.13 %, the arm undershoots to 14.4° and settles in 2.94 s, and the rise back to 120° at 10 s overshoots 32.71° and settles in 4.63 s. Without anti-windup the drop takes 3.24 s, little worse, because this windup is not the limit's. With the integral zone the integral

stays within 1.81 %, and the drop and the rise settle in 0.96 and 0.91 s with 0.90 and 1.44° past their setpoints.

3.6.7 What the Feedforward's Model Misses

Case 7 holds 100° with the schedule, the feedforward and the zone. The battery sag of 0.6 V at 21 s moves the arm least, between 99.5 and 100.1°, because the kit measures the battery and the feedforward rises from 72.47 to 78.16 % by itself. The fan's 6 mN m from 3 to 9 s lifts the arm to 102.7° and it is back within ±2° in 0.76 s, the integral settling at −2.04 %. The 5 g clip at 13 s, which adds 0.0157 N m of weight torque at horizontal, drops it to 93.2°, back in 2.405 s, because the feedforward is 4.10 % short of the 76.57 % that now holds 100°, and a 15 % chip would leave it 5.46 % short. Without the feedforward the integral carries 28.19 % by the end and the sag drops the arm to 90.2°, and without the integral the arm stays at 103.21° under the fan and at 93.79° with the clip.

3.6.8 Sample Time and Encoder Past Horizontal

Case 8 holds 140° with a 20 ms derivative filter and taps the arm at 2 s with 10 ms samples, at 8 s with 40 ms and at 16 s with 80 ms. After the tap the arm is outside ±1° for 0.70 s at 10 ms and 1.42 s at 40 ms, the least damping falling from 0.308 to 0.134, and at 80 ms it reaches the stop 2.68 s after the tap. The linearized loop at 140° is unstable from 63.0 ms, and the limit depends on the angle, 91.8, 77.2 and 60.5 ms at 60, 100 and 150°, because the schedule spends phase margin past horizontal, 36.5, 25.3 and 16.7° at 60, 100 and 140° with this filter at 10 ms. An unstable pole leaves a window of delay that narrows as the pole grows (Section 1.4.7), and the hold's half sample comes out of it. One encoder count is 0.0879°, a rate step of 8.79 °/s differenced over 10 ms. Over the first 5 s, the tap at 2 s included, the unfiltered duty changes by 2.35 % root mean square from one sample to the next, six counts, against 0.74 % with the filter and 0.34 % unfiltered on a 16-bit encoder.

3.6.9 Out of Thrust

Case 9 sags the battery to 6.6 V and chips 40 % of the thrust away, which leaves 0.3818 N of full thrust against the 0.3989 N that holds horizontal, 4.3 % short, so the stall angle of Equation (3.5) is 73.16°. On the battery alone full thrust would still be 59.5 % above the thrust that holds horizontal. Asked for 90° at 3 s, the arm swings between 68.2 and 79.8° at full throttle, a pendulum again with no loop left to damp it, and the scheduled PID, feedback linearization and the fixed PID spend 11.3 to 12.0 of the 12 s there. Anti-windup decides only the return. With back-calculation the law asks for 109.1 % just before the setpoint returns at 15 s, and the arm is back within ±2° of 60° 0.80 s later, while without it the integral winds to 89.4 %, the law asks for 179.5 %, and 15 s later the arm still swings between 64.2 and 73.2°. The blending tank of Volume 2, Chapter 3 meets a setpoint beyond its actuator in the same way, with back-calculation holding the stored output at the bound.

🕮 **Pause.** Try Exercise 3.3 before reading on.

3.6.10 The Setpoint Program

The graded case starts at rest at 20° under the shipped tuning and runs the program 20, 60, 100, 140, 45 and 120°, a step every 8 s, through a 6 mN m gust from 11 to 14 s, a 5 g clip at 20 s and a fan oscillating by 1.5 mN m from 48 s. Every step and disturbance must be back within ±2° within 3.0 s and stay there, no more than 12° past any new setpoint, with no touch of the stop, the throttle saturated for at most 2 % of the 72 s and a chatter of at most 2 %, at the kit's 10 ms, 8-bit throttle and 12-bit encoder. Table 3.6 compares configurations, and Figure 3.4 draws three of its columns against their limits. As shipped the arm swings between 78.5 and 125.5° at 100°, reaches the stop at 25.06 s and strikes it 34 times, and it never settles at 100, 140 or 120°. The suggested configuration keeps the 30° gains and adds the schedule, the feedforward, the zone and a 20 ms filter. Its slowest segment, the clip, takes 2.26 s, its largest overshoot is 9.2° on the step to 140°, and over seven other seeds of the encoder noise the slowest segment takes 2.17 to 2.30 s. Each part matters, and feedback linearization

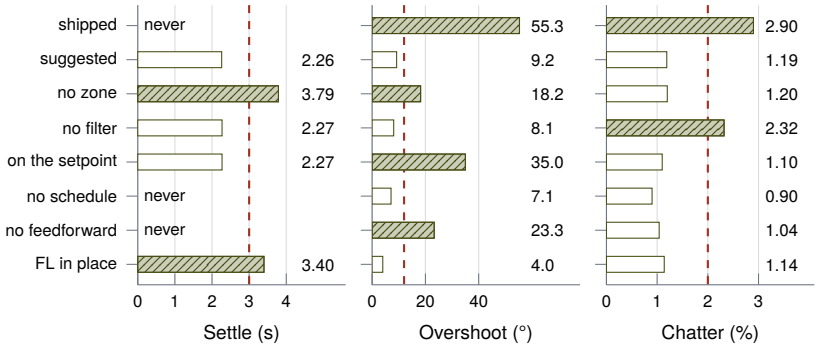


Figure 3.4 The setpoint program in the eight configurations of Table 3.6: the slowest return to ±2° (left), the largest overshoot (middle) and the throttle chatter (right), each with its limit as a dashed line. Hatched bars break that limit and white bars meet it, and three configurations never settle. Notice that the suggested configuration is the only row that is white in all three panels, while the run without the schedule has the smallest overshoot but one and never settles.

Table 3.6 The Setpoint Program in Eight Configurations

Configuration	Settle (s)	Over (°)	Sat. (%)	Chat. (%)	Stop
shipped	never	55.3	0.00	2.90	34
suggested	2.26	9.2	0.43	1.19	0
no zone	3.79	18.2	0.42	1.20	0
no filter	2.27	8.1	0.36	2.32	0
on the setpoint	2.27	35.0	0.42	1.10	1
no schedule	never	7.1	0.00	0.90	0
no feedforward	never	23.3	0.00	1.04	0
FL in place	3.40	4.0	0.17	1.14	0

Settle is the slowest segment's return to ±2°, rounded as the application reports it, Over the largest overshoot past a new setpoint, Sat. the share of the run saturated and Chat. the throttle chatter. The limits are 3.0 s, 12°, 2 %, 2 % and no stop, and only the suggested configuration meets all five.

with the zone and the filter misses only because the clip takes 3.40 s. The case is judged as given, so a change to a model constant or a disturbance voids an attempt.

> ✦ **DID YOU KNOW? A Smaller Overshoot That Never Settles**
>
> From the simulator: in the setpoint program, the suggested configuration without its schedule overshoots only 7.1°, against 9.2° with it, and never saturates the throttle. Yet it never settles, while the full configuration settles in 2.26 s (Table 3.6).

3.7 Technology in Practice

What Measures It

The one controlled variable is the arm's angle, read at the pivot. Laboratory aeropendulums read it with incremental encoders of 600 pulses per revolution [8, 16]. The chapter's kit reads a 12-bit magnetic encoder, 4096 counts a turn, with half a count of noise and an offset when its magnet slips (Section 3.3). One count is 0.0879°, a rate step of 8.79 °/s when differenced over 10 ms, which is why case 8 needs a derivative filter to keep the throttle from chattering.

> **TECHNOLOGY Sensor: A Rotary Encoder at the Pivot**
>
> A laboratory aeropendulum reads its rod angle with an incremental encoder of 600 pulses per revolution, read in quadrature to four times that resolution, and runs its law at a sampling period of 0.02 s [16]. Quadrature means two pulse trains a quarter period apart, so the one that leads tells a counter which way the shaft turns. A Hall-effect sensor that reads a magnet on the moving part removes the backlash and wear of a mechanical linkage [21]. Every encoder still rounds the angle to a count. Quantization acts roughly as noise whose variance is the square of one step over twelve, it can cause small oscillations, and a controller passes it on to the actuator unless it filters it [17].

What Moves It

The final element is a motor and propeller: a brushless motor behind an electronic speed controller on one laboratory rig, and a direct-current motor fed a pulse-width-modulated signal through an H-bridge on another [8, 16]. Because the drive is much faster than the swing of the arm, the second rig left its dynamics out and measured the throttle's map instead: it applied a series of throttles, read the angle at each equilibrium, computed the torque from the static balance, and fitted a cubic that converts the controller's torque back to a throttle [16]. This is the static map of Equation (3.5) measured on hardware. The chapter's motor carries that map with its dead band of 8 %, adds the 30 ms spin-up and 75 ms coast-down lags that a static map leaves out, and lets the battery set full thrust.

> **TECHNOLOGY Final Element: A Speed-Controlled Rotor**
>
> On small multirotor vehicles the lowest level of control is the motor and rotor, and the usual practice regulates rotor speed, turning a thrust demand into a speed through a static model of a rotor in free air. That model shows significant errors in gusts or when the rotor moves, most of all in updrafts and downdrafts. The electronic speed controller runs its own loop, typically at 1 to 2 kHz, and limits its voltage to prevent overcurrent, while its link to the vehicle's central inertial unit is far slower, typically 50 Hz [1]. The chapter's thrust, a fixed coefficient times the speed squared, is such a static model, and Section 3.3.5 names the inflow it leaves out.

Where the Controller Runs

A teaching rig may run its law on a desktop computer through a data acquisition board, as one rig does at 0.02 s, with an emergency stop button for unexpected behavior [16]. On a multirotor the speed loop runs on the electronic speed controller at 1 to 2 kHz, and the link to the central inertial unit is typically 50 Hz [1]. The chapter's kit computes on a microcontroller every 10 ms of a 1 ms clock, reads the encoder and the battery and writes an 8-bit duty. The velocity-form PID, with its schedule, feedforward and anti-windup, or the outer loop of feedback linearization, is the one function that holds the chapter's structure. Figure 3.5 draws the loop on kit hardware.

What Is Modernizing It

The first development is thrust control on the speed controller itself. A power-based thrust model, computed in real time from the measured current, bus voltage and rotor speed, with an inner current loop and an outer PI loop on thrust, held the thrust setpoint better than speed control in downdrafts and updrafts and reduced a quadrotor's tracking error in a gust [1]. It addresses the inflow that the chapter's thrust law ignores. The second is the parameter-varying design. Newer formulations, linear parameter-varying ones among them, relax classical scheduling's restriction to near-equilibrium operation with a good degree of success [6], and the model reference design cited in Section 3.2 is one tried on an aeropendulum. The third is the generated inverse. Some modeling tools generate the code for an inverse model automatically, though feedback linearization can cancel beneficial terms of the natural dynamics and must be used with care [17].

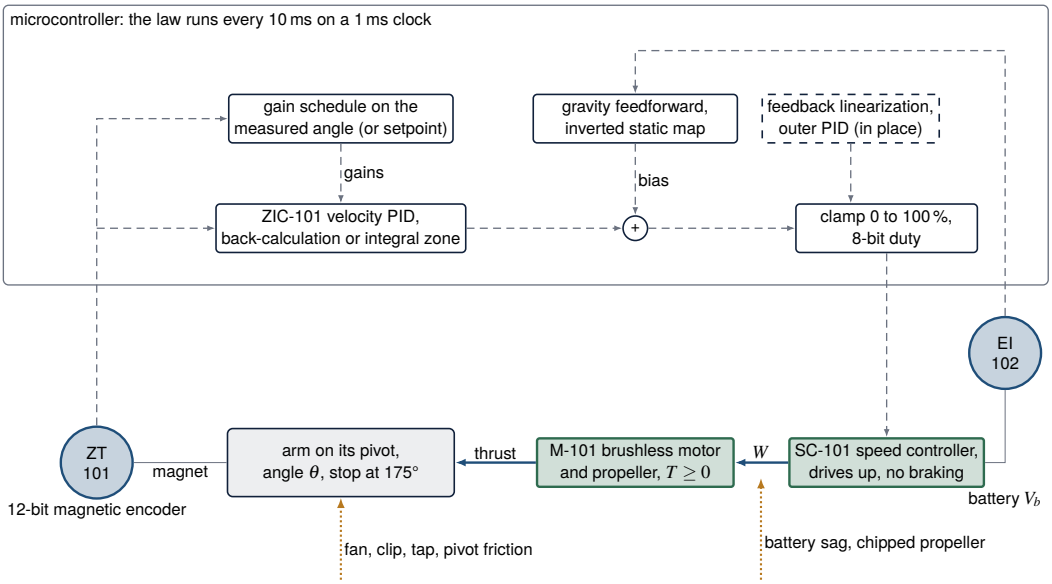


Figure 3.5 The angle loop on kit hardware. The magnetic encoder ZT-101 at the pivot feeds the velocity-form PID ZIC-101, whose gains the schedule sets and whose bias the gravity feedforward supplies from the measured battery (EI-102). Feedback linearization (dashed) takes the PID's place when it is selected. The clamped 8-bit duty goes to the speed controller SC-101, which spins the motor and propeller M-101 up but does not brake it. Notice that the thrust can only push the arm toward larger angles, so every law meets a hard floor at zero thrust as well as a ceiling at full thrust.

> ⚠ **IN PRACTICE** **An Inverse Still Needs Feedback**
>
> A design guide for reentry flight control by dynamic inversion warns that the inversion alone does not guarantee robustness, so a robust outer loop is required. Although advocates assert that no gain scheduling is needed, gain adjustments are still required whenever commands saturate the actuators, because only the vehicle's dynamics are inverted and the actuators are left out [22]. On a laboratory aeropendulum, feedback linearization without an integrator left a steady-state error from disturbances and model uncertainty, which an integrator removed [16]. The chapter's outer PID, with its integral and its limits at zero and full thrust, answers both warnings.

3.8 Limitations

The arm is rigid, the pivot has two orders of damping and no stiction, the thrust ignores the air's motion through the rotor, the motor is a first-order lag on an ideal battery, and the encoder and the throttle only round, so a real kit would damp its swing more and stick near its setpoint, and its fan would push differently at every angle. The lags and every threshold, the 3.0 s, 12°, 2 % and 2 % of the graded case among them, are teaching values rather than any kit's.

The Loop view is linear. It leaves out the rounding, the Coulomb friction and every large-signal effect, so it describes small motions about the setpoint and not a 50-degree step, and the Routh condition of Equation (3.10) is lag-free, a close estimate here because the slow mode lies far below the lags. The schedule and the feedforward trust the declared rig at every instant, and no estimator learns a clip or a chip. The step-test fit assumes a second-order mode about one angle, which fails near horizontal, where the smallest step the throttle can make crosses a third of the stiffness.

OPEN CHALLENGE **Scheduling Faster Than the Theory Allows**

Classical gain scheduling rests on two restrictions: operation near equilibrium, and a plant that moves between operating regions slowly enough. A survey found newer formulations easing the first with a good degree of success, while relaxing the slow variation restriction remained very much an open question. It judged the robustness analysis of gain-scheduled controllers very immature, noted that they often work even when the conditions are violated, and named the choice of scheduling variable as an open need [6]. In case 5 a schedule on the measured angle never touches the stop, and one on the jumping setpoint overshoots onto it. Progress would be a test that tells, before the arm finds out, which scheduling variable is safe.

3.9 Conclusion

The simulator of this chapter carries one tuning across the three plants of a single arm. Gravity's stiffness, $G\cos\theta$, makes the arm a pendulum at 5.95 rad/s hanging, an integrator at horizontal and an equilibrium that diverges at $+4.06\,\mathrm{s}^{-1}$ at 120°. A PID tuned at 30° keeps a phase margin of 49 to 55° up to 105°, yet its slow mode goes unstable at 105.8°, within 0.3° of the lag-free Routh edge. Gravity feedforward takes the integral's share at 80° from 18.01 to −0.48 % but exposes windup with nothing at a limit, which the integral zone cuts from −8.13 % to −1.81 % on the drop of case 6. The schedule holds the damping between 0.38 and 0.53 from 15 to 150° while its phase margin falls to 19.1°, and the graded program passes at 2.26 s only with the schedule, the feedforward, the zone and the filter together. Its arm is rigid, its thrust is deaf to the moving air and its kit computes without delay, and the natural extensions are inflow and stiction, a battery that sags with its current, an estimator for the clip and the chip, and a schedule identified from the reader's own step tests. The kit is a browser laboratory in which the sign of one stiffness decides what a tuning can do.

KEY POINTS

- Gravity makes the arm a pendulum below level, neutral at level and unstable above it, so the setpoint chooses the plant.
- A PID tuned at 30° keeps a phase margin of 49 to 55° up to 105°, yet its slow mode goes unstable at 105.8°, within 0.3° of the lag-free Routh edge.
- Gravity feedforward empties the integral but exposes windup with nothing at a limit, which the integral zone cuts.
- The schedule holds the damping between 0.38 and 0.53 from 15 to 150° while its phase margin falls to 19.1°.

Where else this applies. Volume 2, Chapter 6 schedules its controller on a gain that moves by orders of magnitude across one valve stroke, and Chapter 2 holds an operating point whose steady-state gain changes sign.

References

[1] M. Bangura and R. Mahony, "Thrust control for multirotor aerial vehicles," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 390–405, 2017.

[2] P. Pounds, R. Mahony, and P. Corke, "System identification and control of an aerobot drive system," in *2007 Information, Decision and Control*, 2007, pp. 154–159.

[3] E. T. Enikov and G. Campa, "Mechatronic aeropendulum: Demonstration of linear and nonlinear feedback control principles with MATLAB/Simulink real-time windows target," *IEEE Transactions on Education*, vol. 55, no. 4, pp. 538–545, 2012.

[4] A. Vargová, J. Boldocký, M. Gulan, P. Tibenský, E. Mikuláš, and G. Takács, "AeroShield: An open-source propeller-driven pendulum device for control engineering education," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 9600–9605, 2023.

[5] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[6] D. J. Leith and W. E. Leithead, "Survey of gain-scheduling analysis and design," *International Journal of Control*, vol. 73, no. 11, pp. 1001–1025, 2000.

[7] S. J. McNamara, C. I. Restrepo, J. M. Madsen, E. A. Medina, R. W. Proud, and R. J. Whitley, "Gain scheduling for the Orion launch abort vehicle controller," in *AIAA Guidance, Navigation, and Control Conference*, Portland, OR, 2011.

[8] R. Breganon, U. N. L. T. Alves, J. P. L. S. De Almeida, F. S. F. Ribeiro, M. Mendonça, R. H. C. Palácios, and M. A. F. Montezuma, "Model identification and H_∞ control of an aeropendulum," *Journal of Applied Research and Technology*, vol. 21, no. 4, pp. 526–534, 2023.

[9] A. Abdullah and N. Borham, "Mixed H_2/H_∞ model reference control of linear parameter-varying discrete-time systems with real-time application to aeropendulum system," *International Journal of Robust and Nonlinear Control*, vol. 32, no. 18, pp. 9920–9938, 2022.

[10] G. Habib, A. Miklos, E. T. Enikov, G. Stepan, and G. Rega, "Nonlinear model-based parameter estimation and stability analysis of an aero-pendulum subject to digital delayed control," *International Journal of Dynamics and Control*, vol. 5, no. 3, pp. 629–643, 2017.

[11] D. M. Ovalle M. and L. F. Combita A., "On the implementation of low-cost home-made aeropendulum prototypes for improving understanding of dynamical systems concepts," in *2021 IEEE 5th Colombian Conference on Automatic Control (CCAC)*, 2021, pp. 297–302.

[12] J. Park, R. A. Martin, J. D. Kelly, and J. D. Hedengren, "Benchmark temperature microcontroller for process dynamics and control," *Computers & Chemical Engineering*, vol. 135, p. 106736, 2020.

[13] J. S. Shamma and M. Athans, "Analysis of gain scheduled control for nonlinear plants," *IEEE Transactions on Automatic Control*, vol. 35, no. 8, pp. 898–907, 1990.

[14] L. Hunt, R. Su, and G. Meyer, "Global transformations of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 28, no. 1, pp. 24–31, 1983.

[15] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice Hall, 1991.

[16] U. N. L. T. Alves, R. Breganon, M. Mendonça, and R. H. C. Palácios, "An integral feedback linearization applied to an aeropendulum," *Control and Cybernetics*, vol. 54, no. 1, pp. 49–77, 2025.

[17] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd ed. Princeton, NJ: Princeton University Press, 2021.

[18] J. T. Bosworth, "Success stories in control: Nonlinear dynamic inversion control," NASA Armstrong Flight Research Center, report DFRC-E-DAA-TN1713, preprint, NASA Technical Reports Server 20110008197, https://ntrs.nasa.gov/citations/20110008197, 2010.

[19] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[20] A. Scottedward Hodel and C. E. Hall, "Variable-structure PID control to prevent integrator windup," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 2, pp. 442–451, 2001.

[21] T. R. Kuphaldt, *Lessons in Industrial Instrumentation*, version 2.33 ed. Open textbook, Creative Commons Attribution 4.0 license, ibiblio.org/kuphaldt/socratic/sinst, 2024.

[22] D. Ito, J. Georgie, J. Valasek, and D. T. Ward, "Reentry vehicle flight controls design guidelines: Dynamic inversion," NASA Johnson Space Center, Houston, TX, Technical Publication NASA/TP-2002-210771, 2002, nASA Technical Reports Server 20020039166.

3.10 Exercises

3.1 ○ Concept check · 2 min

Under the PID tuned at 30°, the Loop view reports a phase margin between 49 and 55° for every setpoint from 15 to 105°, yet the loop is unstable from 105.8°. Why does the margin give no warning?

(a) The crossover moves past the Nyquist frequency of the 10 ms sampling near horizontal

(b) The coast-down lag of the motor adds phase lag only past horizontal, where the thrust falls

(c) The encoder rounds away the error near horizontal, so the loop is open there

(d) The escaping mode is a slow pair near 1.6 rad/s, far below the crossover where the margin is read

3.2 ○ Concept check · 1 min

Case 4 runs the PID tuned at 30° on the gravity feedforward instead of the fixed bias of 54 %, and steps the setpoint from 30 to 80°. What does the feedforward change?

(a) The integral's share at 80° falls to near zero, but the step overshoots more
(b) The integral's share at 80° falls to near zero, and the overshoot falls with it
(c) The integral's share at 80° is unchanged, since a feedforward acts only in transients
(d) The throttle saturates on the step, and back-calculation then winds the integral down

3.3 Concept check · 2 min

Case 9 sags the battery to 6.6 V and chips 40 % of the thrust away, and then asks for 90°. Which statement holds?

(a) Feedback linearization reaches 90°, because it inverts the thrust square on the measured battery
(b) No structure reaches 90°, and anti-windup decides only how the arm returns to 60°
(c) The scheduled PID reaches 90° once its integral has wound up far enough
(d) Every structure reaches 90° slowly, because the static map still has an equilibrium there

3.4 Theory and calculation · 7 min

Take the default constants of Table 3.2: a rod of 0.020 kg and 0.32 m, a motor and propeller of 0.030 kg at $l_p = 0.30$ m, pivot friction $b = 1.0$ mN m s/rad and $g = 9.81\,\mathrm{m/s^2}$, with no clip.

(a) Compute J and G from Equation (3.2). *Checkpoint:* $J = 0.003\,383\,\mathrm{kg\,m^2}$ and $G = 0.1197$ N m.

(b) Compute the natural frequency $\sqrt{G\cos\theta^*/J}$ hanging and at 80°, and the damping ratio $b/2\sqrt{GJ}$ of the hanging swing. *Checkpoint:* 5.95 rad/s, 2.48 rad/s and 0.025.

(c) Compute the root of the equilibrium at 120°, with the pivot's friction and without it. *Checkpoint:* $+4.06\,\mathrm{s^{-1}}$ and $+4.21\,\mathrm{s^{-1}}$.

3.5 Theory and calculation · 6 min

The static map of Equation (3.5) gives the thrust and the throttle that hold the arm still. Use $G = 0.1197$ N m, $l_p = 0.30$ m, $T_{max} = 0.80$ N and $u_0 = 8$ %.

(a) Find the thrust and the throttle that hold the arm horizontal on a fresh battery, and the throttle that holds 60°. Which other angle does that throttle hold? *Checkpoint:* 0.3989 N and 72.97 %; 68.46 %, which also holds 120°.

(b) Case 9 runs at $V_b = 6.6$ V with a chip of $c = 0.40$. Find full thrust, how far it falls short of the horizontal thrust, and the stall angle. *Checkpoint:* 0.3818 N, 4.3 % short, 73.16°.

3.6 Theory and calculation · 10 min

The shipped PID has $K_c = 0.2\,\%/^\circ$ (11.46 % per radian), $\tau_I = 0.4$ s and $\tau_D = 1.0$ s. Use $J = 0.003\,383\,\mathrm{kg\,m^2}$, $G = 0.1197$ N m, $b = 0.001$ N m s/rad, $l_p = 0.30$ m, $T_f = 0.80$ N and $u_0 = 8$ %.

(a) Compute the actuator slope $K_u = 2l_p\sqrt{T^*T_f}/(100 - u_0)$ at 30 and 120°. *Checkpoint:* 2.61 and 3.43 mN m per percent.

(b) Evaluate the coefficients of Equation (3.9) at 120° for the shipped gains, and decide stability with Equation (3.10). *Checkpoint:* The s coefficient is −0.0206 N m/rad, so the loop is unstable.

(c) Compute the scheduled K_c', $\tau_I' = K_c'/K_i'$ and $\tau_D' = K_d'/K_c'$ at 120° from Equation (3.13) with the design angle at 30°. *Checkpoint:* 0.984 %/°, 2.59 s and 0.154 s.

3.7 App challenge · 8 min

Load scenario 3, A PID tuned at 30°, carried to 80° and 120°, and change no setting. The setpoint steps to 80° at 2 s and to 120° at 14 s. Use the default seed.

(a) Run to 20 s and export the CSV file. Find the overshoot past 80° from the largest `th_deg` between 2 and 14 s, and read `i_pct` at 13.90 s. *Checkpoint:* 4.25° within 0.05°, and 17.94 % within 0.1 %.

(b) In the same file, find the first row at which `stop_hits` is above zero. *Checkpoint:* 15.40 s, 1.40 s after the step.

(c) Press Reset, set Proportional gain to 0.60 %/°, and run to 30 s. Read the Top stop and the Loop at the setpoint tiles. *Checkpoint:* 0 hits, and ζ 0.14 with a phase margin of 27°.

Explain the result of (c) with the edge that Equation (3.10) gives for this gain.

3.8 App challenge · 7 min

Load scenario 6, A propeller that can only stop. The scheduled PID holds 120° on the feedforward, and the setpoint drops to 20° at 2 s. Use the default seed.

(a) Run as shipped and pause at about 9.9 s, before the setpoint returns at 10 s. Read the Throttle saturated, Since the last step and Overshoot tiles, and export the file to find the lowest `i_pct`. *Checkpoint:* 0.16 s, settled in 2.94 s, 5.6°; −8.13 %.

(b) Press Reset, set Anti-windup to + zone, and repeat (a). *Checkpoint:* 0.15 s, settled in 0.96 s, 0.9°; −1.81 %.

Explain why back-calculation leaves the integral so large when the throttle is pinned for so short a time.

3.9 App challenge · 8 min

Load the challenge scenario, Challenge, the setpoint program, and press Reveal suggested tuning. Use the default seed and run each case for the full 72 s at speed 5. After each run read the challenge result in the Learn panel: the slowest segment, the overshoot, the share of the run saturated and the chatter.

(a) Run with the suggested settings. *Checkpoint:* 2.26 s, 9.2°, 0.43 %, 1.19 %; the target is met.

(b) Press Reset, set Derivative filter to 0 ms, and run again. *Checkpoint:* 2.27 s, 8.1°, 0.36 %, 2.32 %; not met.

(c) Press Reset, set Derivative filter back to 20 ms and Anti-windup to Back-calc., and run again. *Checkpoint:* 3.79 s, 18.2°, 0.42 %, 1.20 %; not met.

Name the limit each failing run breaks and the cause.

3.10 Application area · 4 min

Case 7 holds 100° on the feedforward of Equation (3.12), which reads the battery but knows nothing of a clip. Use $G = 0.1197\,\mathrm{N\,m}$, $L = 0.32\,\mathrm{m}$, $l_p = 0.30\,\mathrm{m}$, $T_{max} = 0.80\,\mathrm{N}$ and $u_0 = 8\,\%$.

(a) The battery sags from 7.4 to 6.8 V. Find the feedforward before and after the sag. *Checkpoint:* 72.47 % and 78.16 %.

(b) A 5 g clip goes on the tip of a fresh battery's arm. Find the added weight torque at horizontal, the throttle that now holds 100°, and the shortfall of the feedforward. *Checkpoint:* 0.0157 N m, 76.57 %, 4.10 %.

Say which part of the controller must supply the shortfall, and name one measurement or estimate a production design could add.

3.11 Application area · 4 min

The kit's 8-bit throttle moves in counts of 0.392 %. Use the static map of Equation (3.5) with $G = 0.1197\,\mathrm{N\,m}$, $l_p = 0.30\,\mathrm{m}$, $T_f = 0.80\,\mathrm{N}$ and $u_0 = 8\,\%$, and invert it for the angle, $\sin\theta^* = T_f\, l_p\, f^2/G$ with $f = (u - u_0)/(100 - u_0)$.

(a) Find the angles that counts 185 and 186 of 255 hold, and say what count 187 does. *Checkpoint:* 80.8° and 87.7°; count 187 holds no angle.

(b) A 10-bit throttle has 1023 counts. Find the last count that holds an angle and the angle it holds, and the angle the count below it holds. *Checkpoint:* Count 746 holds 87.0°, count 745 holds 84.6°.

3.12 Review the agent · 5 min

An AI assistant is asked to extend the PID tuned at 30° to 120° and replies as follows. “The throttle’s grip K_u is 2.61 mN m per percent at 30° and 3.43 mN m per percent at 120°, so the loop gain rises by 32 %. Scale all three gains by $K_u(30^\circ)/K_u(\theta^*)$, which is 0.76 at 120°. The loop gain is then the same at every angle, and the arm will hold 120° as it holds 30°.” Find the flaw and say how you would check it.