INTERACTIVE CONTROL ENGINEERING
Principles and Applications

Volume 4 Override, Split-Range, and Constraint Control

CHAPTER 13

Quadruped Balance Simulator for Sharing Ground Forces Inside Friction Cones

Pages 320–345

Edited by John D. Hedengren

How to cite this chapter. "Quadruped Balance Simulator for Sharing Ground Forces Inside Friction Cones," in *Override, Split-Range, and Constraint Control*, John D. Hedengren, Ed., vol. 4 of *Interactive Control Engineering: Principles and Applications*. Advanced Process Solutions, LLC, 2026, ch. 13, pp. 320–345.

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

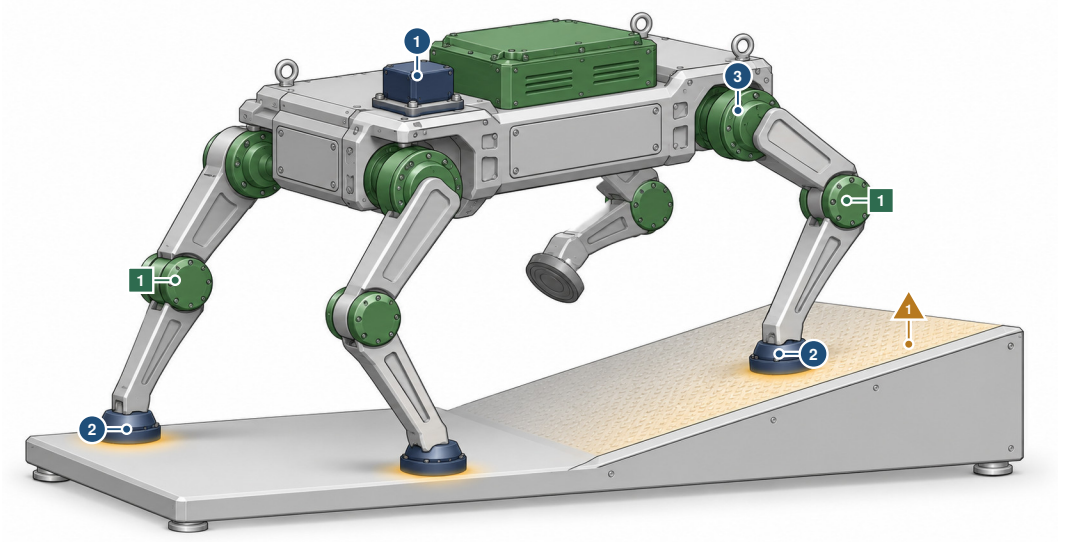
The app for this chapter
https://apmonitor.com/apps/quadruped-balance/

The volume online: contents, other chapters and citation records
https://apmonitor.com/apps/books/v04-override-constraint/#quadruped-balance

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

CHAPTER 13

Quadruped Balance Simulator for Sharing Ground Forces Inside Friction Cones



The controller moves 1 hip and knee motors that push each foot

The controller reads 1 tilt and turn sensor on the body 2 contact switches in the feet 3 joint angle encoders

What disturbs it 1 a ramp and a slippery patch

A robot balances only through the feet that are down, and each one can push but not pull, and only inside the grip of the ground under it. Illustration, not to scale.

IN PLAIN TERMS A Robot Dog That Can Only Push

The main idea. A four-legged robot keeps its body steady only by pushing on the ground with the feet that are down. Its computer decides how hard each foot pushes, and a foot can never pull or push sideways harder than the ground's grip allows.

The challenge. The grip, the feet on the ground and the strength of each leg change at every step. A computer that guesses wrong, say that a slippery floor grips well, asks for a push the ground cannot give, and the body tips.

Why it matters. Any machine that shares one job among several limited helpers must know what each can really give.

THREE WAYS TO READ THIS CHAPTER

Explore	12 min	the first page, the history segment (Section 13.2), the fact on leg odometry (Section 13.3), the fact and the two try-it segments in the results (Section 13.6) and the key points
Study	65 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 13.6), Technology in Practice (Section 13.8), Limitations (Section 13.9) and the two open challenges

The control question A walking robot corrects its height, pitch and speed only by pushing on the ground, and each foot pushes but cannot pull, only inside its friction cone and only while it touches. Which limit binds in each situation, and what does a controller gain by knowing the true cone, the true contacts and the true leg limits, or by planning the forces ahead?

Learning objectives

- Reduce the feasible closed loop to three double integrators and place their roots from the gains.
- Derive the tipping and friction bounds on braking and tell which one caps a push.
- Explain how planning with the wrong friction or the scheduled feet turns a force request into an unintended moment.
- Compare a reactive allocation with a predictive force planner on a leg whose force limit switches with the gait.
- Configure friction and footholds so that the robot passes a mixed course, and judge the margins it leaves.

Prerequisites Sections 1.4.8 (control allocation), 1.4.5 (saturation and lost actuator effect), 1.6.2 (velocity form) and 1.6.4 (scenarios). Section 1.6.3 is needed only for the predictive planner.

The app https://apmonitor.com/apps/quadruped-balance/

See also Volume 20, Chapter 4 for an allocator whose effectors gain and lose authority, Volume 20, Chapter 6 for thrust allocation on a double-integrator hull and Volume 20, Chapter 13 for constraints that change at contact. Volume 9, Chapter 4 for offset-free predictive control with a disturbance estimate. In this volume, Chapter 12 for allocation by fixed priority.

13.1 Introduction

Legged robots move their bodies only by pushing on the ground. Lifting a sagging body, raising a dipping nose, braking after a shove and holding a posture on a slope all arrive as ground reaction forces at whichever feet are planted at that instant, and each of those forces is bounded three ways. A foot can push but cannot pull, its leg can press only as hard as its motors allow, and its component along the ground can be no larger than the friction coefficient times its normal component [1, 2]. A foot in the air carries nothing.

Control courses usually treat an actuator as a signal that a loop drives anywhere between two limits, so a loop that computes the right command is taken to have delivered it. A legged robot separates the command from the delivery. In the simulator of this chapter, a robot trotting across a patch of floor with a friction coefficient of 0.15 is shoved from behind, and an allocation that assumes a coefficient of 0.6 hands the feet 68.2 N of braking that it believes is feasible. The ground returns 14.8 N. The vertical forces had been arranged to cancel the nose-down moment of the braking that never arrived, so the body pitches up by 13.9° instead of braking level. What the loops request, what the allocation believes and what the ground returns are three different quantities, and the cases of the application are built around the gaps between them.

The problem is one member of a wider family. Several actuators share one requested total, each has hard limits of its own that change with the operating state, and an allocation must decide how much of the total each supplies and report what none of them can. The internal forces of a multifingered grasp chosen under friction and joint-torque limits [1], the deflections of redundant aircraft control surfaces chosen to produce a demanded moment [3], and the ground reaction forces that hold up a humanoid or a quadruped [2, 4] are one problem. Legs add a feature that the other members of the class rarely face, because the set of available actuators changes with the gait, as feet land, lift, slide and miss a foothold several times a second. Volume 20, Chapter 4 meets a milder form of it, effectors whose authority grows and fades with airspeed, and Volume 20, Chapter 13 a constraint set that changes at the instant of contact.

A block diagram of the system hides what matters. While every request is feasible, the allocation is transparent and the torso behaves as three decoupled double integrators, one per loop, which is the case that the linear analysis of Section 13.5 treats exactly. The cases of the application are the

moments when a limit binds. Tipping over the front feet caps braking before friction does, leg reach caps posture on a ramp, the true friction cone caps braking on a slippery floor, a foot over a hole contributes nothing, and a weak leg caps what its diagonal pair can hold. Reviews of control education place laboratory and software exercises at the center of good practice [5] and describe a long-standing distance between control theory and industrial practice [6]. A plant whose set of actuators changes at every step is a direct way to expose that distance.

This chapter determines, case by case, which limit binds in a quadruped balance simulator and what each controller choice costs, and it sets the simulator down in enough detail that it can be rebuilt without its source. The single rigid body, force allocation by quadratic programming, Raibert foot placement and predictive control of ground reaction forces each have an established literature [2, 7, 8]. The model is planar, with unilateral, friction-limited and force-limited contact, sliding feet, toe stubs, missed footholds and odometry that slip corrupts. It is controlled by an allocation that is feasible at every iteration or by a receding-horizon force planner. Closed-form checks against the running engine cover the three loops as double integrators while the allocation is feasible, a braking bound in which tipping over the front feet binds before friction does, and a copper-loss expression that accounts for the energy figure. Each case changes one quantity and reads the consequence from a scorecard that keeps the worst pitch, any fall and the worst push recovery.

Section 13.2 traces balance control on legs. Section 13.3 gives the torso, the legs, the contacts and the sensors, and Section 13.4 the three loops, the allocation, the trot and the planner. The allocation is the weighted least squares of Section 1.4.8 with three demands and a set of actuators that changes at every step. Section 13.5 derives what the feet can deliver, and Section 13.6 reports the nine measured cases. Section 13.7 lists the checks of the engine, Section 13.8 sets the robot's devices beside real hardware, and Section 13.9 states the limitations.

13.2 Balance on Legs, From Creeping Gaits to Force Planning

The first quantitative stability criterion for walking machines was static. A quadruped that lifts one leg at a time remains upright while the vertical projection of its center of mass lies inside the polygon of its supporting feet, and the shortest distance from that projection to the edge of the polygon was proposed as the stability margin of creeping gaits in 1968 [9]. The criterion suits slow machines. It survives in this application in two forms, the support interval drawn under the robot on its scene and the tipping limit on braking derived in Section 13.5.

Dynamic balance came from running machines. Simple algorithms that balance a machine on one leg were generalized to several legs, and the concept of a virtual leg mapped the trot, the pace and the bound, gaits that use the legs in pairs, onto one virtual leg in support at a time, with a quadruped controlled in trotting experiments [7]. The fixed-timing trot and the foot placement rule of this application descend from that work. Hydraulic machines later carried legged balance onto rough ground [10], and quadrupeds with torque-controlled joints, such as HyQ, which stands 1 m tall, weighs about 90 kg and carries 12 torque-controlled joints [11], made the force at each foot something a controller could command directly.

Force-based balance was worked out on humanoids. One method is told in the history segment after this paragraph. Another computes full-body joint torques from a desired motion of the center of mass and the contact forces that produce it [12]. On a 75 kg quadruped the same distribution, posed as a quadratic program with unilateral and no-slip constraints and minimum actuator effort, carried the robot walking between two walls inclined at 50° [2]. The mathematics had appeared earlier in grasping, where the internal forces of a multifingered hand were chosen by linear programming under friction and joint-torque limits [1]. Aircraft control allocation solves the same demanded-moment problem over redundant surfaces, with optimization methods that trade the error in the achieved moment against the control effort [3].

> ⧖ HISTORY **2007: Sharing a Humanoid's Weight Among Its Feet**
>
> Force-based balance was worked out on a hydraulic humanoid of 93.7 kg. Its controller set the ground force and the center of pressure that the body needed and turned them into joint torques for any number of contact points, without measuring the force at those points, and it kept the robot balanced against unknown external forces. The normal forces were shared by a pseudo-inverse that minimizes the sum of their squares, which avoids needless internal forces between the contacts. The horizontal forces followed in proportion to the normal ones, so the contacts that pressed hardest received the most friction. A weighting matrix could be added to respect the friction cone or to unload a limb with a broken joint, and every control process finished within 1 ms [4].

Prediction followed. Reducing the robot to a single rigid body driven by its ground reaction forces made force planning convex, and horizons of up to 0.5 s were solved to optimality in under 1 ms at 20 to 30 Hz on a quadruped that stood, trotted, bounded and galloped [8]. That formulation is the reference design for the predictive planner of this application, which keeps the single rigid body in the plane and solves it approximately with a fixed number of iterations.

The simplifications that made these controllers tractable concern the contacts. A rigid schedule assumes that a foot touches down when the gait says it will, while a contact estimate fused from kinematics, dynamics and the schedule detected touchdown with 99.3 % accuracy and a delay of 4 to 5 ms and let a controller respond leg by leg to early and late contacts [13]. The usual no-slip assumption corrupts state estimation and control alike on slippery ground, which led to separate treatment of slip detection and slip recovery [14]. Terrain templates and learned foothold selection carried the LittleDog quadruped over rough terrain [15], and a push larger than the standing feet can absorb has to be caught with a step, at a capture point [16]. The application reproduces each of these problems in its simplest form and deliberately keeps the fixed schedule, the no-slip odometry and the absence of stepping, so that their consequences can be measured.

13.3 Planar Torso, Legs and Contacts

The simulated robot is a 12 kg rigid torso moving in the sagittal plane on four massless legs, each made of two 0.20 m segments, with hips 0.19 m ahead of and behind the torso center. In the plane the two front legs share one hip and the two rear legs another, while every leg keeps its own foot, contact state and force, and the application declares this approximation on its scene and in its model panel. Figure 13.1 draws the robot with each symbol the equations use, and Table 13.1 gives the role of every signal.

13.3.1 Torso and Mass Properties

The torso is a rigid body driven only by the ground reactions of its planted feet, because a massless leg transmits exactly the force at its foot. The state describes the combined body of robot and backpack, whose translation and rotation obey Equation (13.1).

$$\begin{aligned} m_t \ddot{x} &= \sum_{i \in P} F_{x,i}, \qquad m_t \ddot{z} = \sum_{i \in P} F_{z,i} - m_t g \\ I_t \ddot{\varphi} &= \sum_{i \in P} \left(r_{x,i} F_{z,i} - r_{z,i} F_{x,i} \right), \qquad \mathbf{r}_i = \mathbf{p}_i - (x, z) \end{aligned} \tag{13.1}$$

where P is the set of feet in contact, $\mathbf{p}_i$ is the position of foot i in m, the forces are in N, and φ is in rad, positive nose-up. A backpack of mass m_b centered at the body-frame point $\mathbf{b} = (-0.10, 0.085)$ m, on the rear of the back, gives the mass properties of Equation (13.2).

$$m_t = m + m_b, \quad \mathbf{c} = \frac{m_b \mathbf{b}}{m_t}, \quad I_t = I + m \|\mathbf{c}\|^2 + m_b \|\mathbf{b} - \mathbf{c}\|^2 \tag{13.2}$$

where $\mathbf{c}$ is the offset of the combined center of mass from the torso center in the body frame. A 4 kg pack moves the center of mass 2.50 cm back and 2.13 cm up and raises I_t from 0.30 to 0.352 kg m^2.

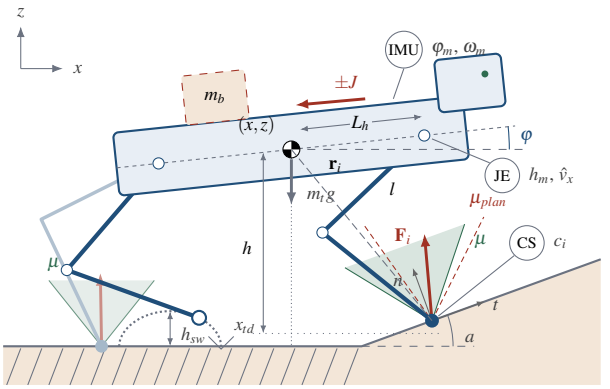


Figure 13.1 Planar robot dog on a floor that turns into a ramp of slope a. The torso, pitched by φ with its center of mass at (x, z) and weight $m_t g$, carries hips L_h ahead of and behind its center, and each leg of two segments of length l reaches a foot at moment arm $\mathbf{r}_i$. A planted foot receives the ground reaction $\mathbf{F}_i$, resolved along the ground t and its normal n, inside the true friction cone of half-angle $\arctan \mu$ (green), while the allocation plans inside the dashed cone of half-angle $\arctan \mu_{plan}$ about the normal it believes in. The near rear foot swings toward the touchdown point x_{td} with clearance h_{sw}, and the far rear leg, drawn lighter, stands on its own foot. The body height h runs from the planted feet to the hip line, the push $\pm J$ lands on the top of the back and a backpack m_b sits on its rear. The IMU, the joint encoders JE and the foot contact switches CS supply φ_m, ω_m, h_m, $\hat{v}_x$ and c_i.

Table 13.1 What Each Signal Is, With the Value It Holds While Standing

Symbol	Role	Nominal	Range
h	CV, body height	28 cm	SP 18 to 34
φ	CV, pitch, + nose-up	0°	SP ±15
v_x	CV, forward speed	0 m/s	SP −0.4 to 0.8
F_z	MV, total vertical force	117.7 N	0 to $\sum n_{max,i}$
M_y	MV, pitch moment	0 N m	±60
F_x	MV, total horizontal force	0 N	$\lvert F_x \rvert \leq 150$
t_i, n_i	allocated force per foot	0, 29.4 N	cone, F_{max}
x, z, φ	states, torso	0, 0.28 m, 0	—
v_x, v_z, ω	states, torso rates	0	—
$p_{x,i}$, $p_{z,i}$	states, four feet	±0.19 m, 0	reach 0.10 to 0.38
c_i	states, contact modes	stuck	air, stuck, sliding
—	state, fall flag	0	0 or 1
J	DV, push on the back	10 N s	2 to 16, events
m_b	DV, backpack	0 kg	0 to 6
—	DV, rear-left leg capacity	100 %	20 to 100
—	DV, ground ahead	flat	six courses
φ_m, ω_m	measured, IMU	—	noise, Table 13.2
h_m, $\dot{h}_m$, $\hat{v}_x$	measured, leg kinematics	—	noise, Table 13.2
c_i	measured, contact switches	—	ideal, every 2 ms

When the pack goes on or comes off during a run the torso does not move, and the state is relabeled to the new combined body with the velocity of the torso carried over.

13.3.2 Unilateral, Friction-Limited Contact

The ground is a profile with height $z_g(x)$, slope angle $a(x)$ and static friction coefficient $\mu_s(x)$, and a force at a foot is resolved into a component F_t along the ground and F_n along its normal, as shown in Equation (13.3).

$$F_x = F_t \cos a - F_n \sin a, \qquad F_z = F_t \sin a + F_n \cos a \tag{13.3}$$

A planted foot delivers the force (f_t, f_n) its leg applies whenever that force is physically possible, and the contact law of Equation (13.4) decides what reaches the body.

$$\begin{aligned} F_n &= \text{clamp}\,(f_n, 0, F_{max,i}) \\ (F_t, v_s) &= (f_t,\ 0) \qquad \text{if } |f_t| \le \mu F_n \\ (F_t, v_s) &= \left(\mu_k F_n \,\text{sgn} f_t,\ -\text{sgn}(f_t)\,\frac{|f_t| - \mu_k F_n}{b}\right) \end{aligned} \tag{13.4}$$

where the last line applies when the push leaves the cone, μ is μ_s for a stuck foot and $\mu_k = 0.85\,\mu_s$ for a sliding one, $F_{max,i}$ is the normal-force rating of leg i, v_s is the speed of the foot along the ground in m/s, and $b = 150\,\text{N}\,\text{s}/\text{m}$ is the back-drive damping of the leg, the one number a massless, force-controlled leg needs to say how fast it gives way. The tangential command is clipped to $\pm F_{max,i}$ first. A sliding foot moves opposite to the push it failed to hold and sticks again once $|f_t|$ falls back inside $\mu_k F_n$. The stick or slip decision is algebraic within a step, with the threshold chosen by the mode that the previous step left, so the right-hand side stays smooth inside each Runge-Kutta step. A foot in the air carries no force whatever its leg commands and moves with the velocity its swing controller gives it.

13.3.3 Legs, Swing, Terrain and Hard Events

Each leg reaches between $l_{min} = 0.5\,l$ and $l_{max} = 1.9\,l$ from its hip, 0.10 m to 0.38 m, and a swing foot moves no faster than $v_{sw} = 2.5\,\text{m}/\text{s}$. The knee of each two-segment leg points backward and is computed from the hip and the foot. Contact modes change only at hard events applied after each integration step. A swing foot touches down when it is at or below solid ground within its leg's reach and is not being driven upward, and a planted foot lifts off when the gait releases it. A planted foot farther than l_{max} from its hip loses the ground, which is counted as a reach loss, while an airborne foot at full reach is carried along by its hip. A swing foot moving forward below the top of an upward face is stopped at the face, a toe stub, and lands short. The robot falls when its pitch passes 40°, when a corner of the torso or a knee touches the ground, or when a planted leg folds below l_{min}, and the simulation then freezes.

Every course starts with at least 0.6 m of flat, dry floor with $\mu_s = 0.8$. The ramp runs from 0.7 to 2.7 m at an adjustable angle, the slippery patch from 0.75 to 1.95 m with $\mu_{patch} = 0.15$, the step rises 6 cm at 0.95 m, and the hole is 12 cm wide at 0.85 m, with a floor 1 m down, beyond any leg. The mixed course combines an 8° ramp from 0.6 to 1.35 m, a patch from 1.75 to 2.45 m, a 6 cm step at 2.95 m and a finish line at 3.8 m.

13.3.4 What the Controller Measures

The controller never reads the true state, as Equation (13.5) sets out.

$$\begin{aligned} \varphi_m &= \varphi + \sigma_\varphi \varepsilon_1, \qquad \omega_m = \omega + \sigma_\omega \varepsilon_2 \\ h_m &= \frac{1}{|P|} \sum_{i \in P} - \left(r'_{x,i} \sin\delta + r'_{z,i} \cos\delta\right) + \sigma_h \varepsilon_3 \\ \hat{v}_x &\leftarrow \text{lag}\left(\frac{1}{|P|} \sum_{i \in P} \left(v_{T,x} - \dot{p}_{x,i}\right) + \sigma_v \varepsilon_4,\ \tau_v\right) \end{aligned} \tag{13.5}$$

where ε_k are independent unit normal deviates drawn each control period from the seeded generator, $\delta = \varphi_m - \varphi$ is the IMU error, $\mathbf{r}'_i$ runs from the torso center to foot i, $v_{T,x}$ is the horizontal velocity

of the torso center and lag is the exact discrete first-order lag. The same odometry gives the vertical rate $\dot{h}_m$ used by the height loop, and the integral of $\hat{v}_x$ is the position estimate $\hat{x}$. The noise levels are $\sigma_\varphi = 0.1°$, $\sigma_\omega = 0.02\,\mathrm{rad/s}$, $\sigma_h = 1.5\,\mathrm{mm}$ and $\sigma_v = 0.02\,\mathrm{m/s}$, and the odometry is smoothed with $\tau_v = 15\,\mathrm{ms}$. Contact switches report the true contact of each foot, sampled every 2 ms. The odometry assumes that a planted foot does not move, so a foot sliding at v_s biases the estimate by the slip speed and the position estimate drifts.

> **DID YOU KNOW? Leg Odometry Drifts by About a Tenth**
>
> In a 60 s static walk of about a meter, a filter that fused leg kinematics with an inertial unit drifted in position by up to roughly 10 % of the distance traveled, from inaccurate leg kinematics and fault-prone contact detection [17]. The odometry of this chapter has no such filter.

13.3.5 Disturbances and Numerics

A push is a horizontal impulse J on the top of the back, 0.06 m above the hip line, applied as the velocity jumps of Equation (13.6).

$$\Delta v_x = \frac{J}{m_t}, \qquad \Delta\omega = -\frac{r_{z,J}\,J}{I_t} \tag{13.6}$$

in which $r_{z,J}$ is the height of the push point above the combined center of mass in m. The calibrated 10 N s push starts the unloaded robot at 0.833 m/s with a nose-down rate of 2.0 rad/s. The backpack is a step in m_b, and the rear-left leg can be derated to a fraction of F_{max}.

The equations are integrated by fourth-order Runge-Kutta at $\Delta t = 0.5\,\mathrm{ms}$, the controller runs every $\Delta t_c = 2\,\mathrm{ms}$ with the leg forces held between executions, and the historian logs at 50 Hz. The engine steps as Section 1.6.1 describes, with the hard events applied after each integration step. Halving the step to 0.25 ms moves the robot of case 7 with sensed contacts from 1.7123 to 1.7116 m after 4 s and its worst pitch from −13.28 to −13.42°, with the same 42 touchdowns in both runs. The noise comes from the seeded generator of the framework, so a run replays exactly. Every numerical value of the model is listed in Table 13.2.

13.3.6 Assumptions

The model is planar. There is no roll, no yaw and no sideways motion, and the left and right legs at each end share a hip, so the robot cannot balance itself sideways and the roll sway of a real trot is absent. The assumption holds for straight walking over ground that is the same across the width of the robot, and it fails for a ramp taken at an angle or a hole under one side only.

The legs are massless. Swinging a leg costs nothing and pushes nothing back on the body, and a touchdown has no impact. This is fair at a walk or a slow trot, where the legs carry a small fraction of the mass, and wrong for running.

Each leg delivers the force it is commanded after a 4 ms lag, up to a normal-force rating that does not depend on the configuration of the leg. Real limits are joint torques whose force equivalent changes with the knee angle.

The ground is rigid, with Coulomb friction and an assumed back-drive damping that sets how fast a foot slides. Real feet roll and deform, and real ground gives way.

The trot has fixed timing and no recovery step, so late contacts do not shift the schedule and a push beyond what standing feet can stop cannot be caught [13, 16].

The sensors carry white noise only, the contact switches are ideal, the terrain map is perfect when it is selected, and no estimator fuses the measurements, so a slip appears directly as position error [14].

13.4 Loops, Allocation, Gait and Planner

The controller runs every 2 ms. Three feedback loops turn the measured height, pitch and speed into a requested body wrench $\mathbf{w} = (F_x, F_z, M_y)$, an allocation turns the wrench into one force per planted

Table 13.2 Parameters of the Robot, Ground and Sensors. Class a Can Be Changed in the Parameter Panel, h Is Hidden From It, and c Is Set in the Source. The Controllers Keep the Values of m and I Listed Here

Symbol	Meaning	Value	Unit	Cl.
m	robot mass	12	kg	a
I	pitch inertia	0.30	kg m^2	a
L_h	hip to torso center	0.19	m	a
l	thigh and shank length	0.20	m	h
l_{min}, l_{max}	leg reach	0.10, 0.38	m	c
—	torso length, top, bottom	0.50, 0.05, −0.04	m	c
F_{max}	leg normal-force rating	150	N	a
v_{sw}	swing speed limit	2.5	m/s	a
τ_a	leg force lag	4	ms	a
b	back-drive damping	150	N s/m	h
μ_s, μ_{patch}	floor, patch friction	0.8, 0.15	—	a
μ_k/μ_s	kinetic to static ratio	0.85	—	a
—	ramp angle in case 4	25	°	a
—	step height, hole width	6, 12	cm	a
σ_φ, σ_ω	IMU noise	0.1, 0.02	°, rad/s	a
σ_h, σ_v	height, odometry noise	1.5, 0.02	mm, m/s	a
τ_v	odometry filter	15	ms	a
J	calibrated push	10	N s	a
—	push height above hip line	0.06	m	c
b	backpack position	−0.10, 0.085	m	c
—	pitch at which it falls	40	°	h
g	gravity	9.81	m/s^2	c
Δt, Δt_c	physics, control step	0.5, 2	ms	c
—	historian rate	50	Hz	c

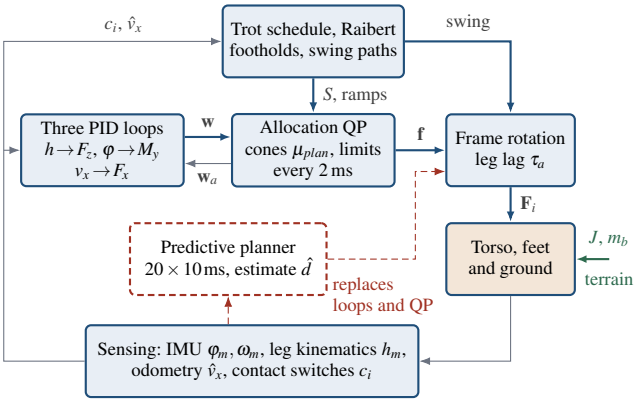


Figure 13.2 Control structure. Three loops request the wrench **w**, the allocation shares it among the feet in the set S that the trot schedule plans and the contact switches confirm, and the delivered wrench $\mathbf{w}_a$ returns to the loops for anti-windup. The forces **f** are rotated from the believed ground frame into the true one and reach the feet through the leg lag. The dashed path is the predictive planner, which replaces the loops and the allocation together when it is selected.

foot, and a fixed-timing trot decides which feet are planted. The structure is shown in Figure 13.2, and its settings are listed in Table 13.3.

13.4.1 Three Loops as Implemented

Each loop runs the velocity-form PID of Section 1.6.2, Equations (1.12) to (1.15), with $P_k = K_c e_k$ and one correction added to the integral, as shown in Equation (13.7).

$$\begin{aligned} u_k^u &= u_{b,k} + \sum_{j \le k} \left(P_j - P_{j-1} \right) + I_{k-1} \\ u_k &= \text{clamp}\left(u_k^u, u_{min}, u_{max} \right) \\ I_k &= I_{k-1} + \left[\frac{K_c}{\tau_I} e_k + \frac{u_k - u_k^u}{T_t} + \frac{w_{a,k} - u_k}{T_t} \right] \Delta t_c \end{aligned} \tag{13.7}$$

Here $u_{b,k}$ is a bias recomputed every period, u_k^u is the output the algorithm demands, built on the integral accumulated through scan $k-1$, $T_t = \tau_I$ is the tracking time and $w_{a,k}$ is the component of the wrench that the allocation actually delivered. The first correction is the back-calculation of the framework against the clamp. The second is added by the application so that an integral stops winding up against a force the feet could not supply, the case of an actuator that loses its effect without reaching its clamp in Section 1.4.5. Both act on the integral term only and only when the loop has integral action, and an integral time at its 5 s slider maximum removes that action. The framework PID can differentiate a filtered measurement, but all three loops set that derivative to zero and take their rate terms instead through the bias from measured rates, $u_b = mg - K_{c,h}\tau_{D,h}\dot{h}_m$ for the height loop and $u_b = -K_{c,\varphi}\tau_{D,\varphi}\,\omega_m$ for the pitch loop, so no derivative filter is needed and a setpoint step produces no derivative kick. Away from the limits, where neither correction acts, the recursion reduces to the positional laws of Equation (13.8).

$$\begin{aligned} F_z &= mg + K_{c,h}\left(e_h + \frac{1}{\tau_{I,h}} \int e_h \, dt \right) - K_{c,h}\tau_{D,h}\,\dot{h}_m \\ M_y &= K_{c,\varphi}\left(e_\varphi + \frac{1}{\tau_{I,\varphi}} \int e_\varphi \, dt \right) - K_{c,\varphi}\tau_{D,\varphi}\,\omega_m \\ F_x &= K_{c,v}\left(e_v + \frac{1}{\tau_{I,v}} \int e_v \, dt \right) \end{aligned} \tag{13.8}$$

with $e_h = h_{SP} - h_m$, $e_\varphi = \varphi_{SP} - \varphi_m$ and $e_v = v_{SP} - \hat{v}_x$. The bias mg is the nominal weight of the unloaded robot, 117.72 N, whatever mass the Parameter panel sets, so a backpack or a heavier robot has to be found by the integral. The speed setpoint is ramped at 0.8 m/s^2, and the integral of the speed error is the distance between a position reference that advances at the setpoint and the estimate $\hat{x}$, so a standing robot with $v_{SP} = 0$ holds its place. All three loops form the error as setpoint minus measurement with the same sign, so a body below its height setpoint receives more vertical force, a nose-down error a nose-up moment and a speed deficit more forward force, which makes each loop reverse acting in the ISA convention. The limits are $0 \le F_z \le \sum n_{max,i}$ over the planted feet, $|M_y| \le 60$ N m and $|F_x| \le \min(150,\ \mu_{plan} \sum n_{max,i})$ N, and no rate limit applies to any output. With the pitch measured from the ground, the setpoint becomes $\varphi_{SP} + \hat{a}$, where $\hat{a}$ is the estimated ground slope smoothed by a 0.25 s lag.

In manual mode the operator sets F_z, M_y and F_x directly, on sliders of 0 to 400 N, ±40 N m and ±120 N clamped to the same limits, and the allocation shares them among the planted feet. Each loop keeps running and tracks the delivered component by setting I_k so that u_k equals it, which makes the return to automatic bumpless. The same tracking runs while the planner is in charge.

Table 13.3 Controller Settings at Their Defaults, With Ranges. Settings Marked adv. Are Shown in the Advanced View, and the Final Block Is Set in the Source

Symbol	Meaning	Default (range)
h_{SP}	height setpoint	28 cm (18 to 34)
$K_{c,h}$, $\tau_{I,h}$, $\tau_{D,h}$	height loop	1500 N/m, 0.5 s, 0.12 s
φ_{SP}	pitch setpoint	0° (±15)
$K_{c,\varphi}$, $\tau_{I,\varphi}$, $\tau_{D,\varphi}$	pitch loop	60 N m/rad, 0.6 s, 0.10 s
v_{SP}	speed setpoint	0 or 0.4 m/s (−0.4 to 0.8)
$K_{c,v}$, $\tau_{I,v}$	speed loop	120 N s/m, 0.8 s
—	integral removed at	$\tau_I = 5$ s
μ_{plan}	friction assumed	0.6 (0.05 to 1.2)
—	forces go to	sensed contacts, or scheduled feet
—	leg limits planned with	nominal, or actual
—	force planning	reactive, or predictive
N	horizon, 10 ms blocks	20 (4 to 40)
—	planner iterations	40 (1 to 150), adv.
h_{sw}	swing clearance	5 cm (1 to 12)
k_v	foot placement gain	0.08 s (0 to 0.3)
—	footholds	blind, or terrain map
T, D	gait period, duty factor	0.36 s, 0.55, adv.
W	wrench weights	1, 1, 30 m^{-2}
λ_t, λ_n, β	allocation regularization	0.02, 0.002, 0.01
—	allocation iterations	40
R_t, R_n	planner force weights	2×10^{-3}, 2×10^{-4} N^{-2}
—	estimate limits, $\hat{\mathbf{d}}$	3 m/s^2, 15 rad/s^2

13.4.2 Allocation Inside the Cones

The allocation turns $\mathbf{w}$ into ground-frame forces $\mathbf{f} = (t_j, n_j)$ for the planted feet by the quadratic program of Equation (13.9).

$$\begin{aligned} \min_{\mathbf{f}} \; & (A\mathbf{f} - \mathbf{w})^\top W (A\mathbf{f} - \mathbf{w}) + \beta \|\mathbf{f} - \mathbf{f}^-\|^2 \\ & + \lambda_t \textstyle\sum_j t_j^2 + \lambda_n \sum_j n_j^2 \\ \text{s.t. } & 0 \le n_j \le n_{max,j}, \qquad |t_j| \le \mu_{plan}\, n_j \end{aligned} \tag{13.9}$$

where the column pair of A for foot j maps its force through the believed ground angle $\hat{a}_j$ and its moment arm to (F_x, F_z, M_y), the weight $W = \mathrm{diag}(1, 1, 30)$, with the last entry in m^{-2}, ranks the moment above the horizontal and vertical forces, which carry equal weight, $\lambda_t = 0.02$ keeps tangential pushes toward the middle of the cone, $\lambda_n = 0.002$ regularizes the normal forces, and $\beta = 0.01$ ties the answer to the previous period's $\mathbf{f}^-$. Each feasible set is a friction cone cut off by the leg's rating, a triangle in the (t, n) plane, and $n_{max,j}$ is the rating the controller plans with, nominal or actual, scaled by the loading ramp at the start and end of each stance. The program is solved by 40 iterations of accelerated projected gradient [18], with the step set from a power-iteration estimate of the largest eigenvalue of the Hessian and every foot's force projected back onto its triangle after each step. The answer is therefore feasible however few iterations run, and what it cannot deliver appears as a shortfall $\mathbf{w}_a \neq \mathbf{w}$ that the loops treat as saturation. Chapter 12 shares the force on one car body among tires, fans and thrusters by fixed priority instead of quadratic weights.

The allocation works in the frame the controller believes. Its moment arms run from the torso center rather than from the true center of mass and are rotated by the IMU error, and its ground angle is $\hat{a}_j$, which for blind footholds is the angle of the line through the planted front and rear feet and for the terrain map is the mapped slope at each foot. The command sent to leg i is converted into the true

ground frame at its foot by Equation (13.10).

$$\begin{bmatrix} f_t \\ f_n \end{bmatrix}_i = R(-a_i)\,R(-\delta)\,R(\hat{a}_i) \begin{bmatrix} t \\ n \end{bmatrix}_i \tag{13.10}$$

where $R(\cdot)$ is a planar rotation. This is where a wrong idea of the slope or of the pitch turns into a push the controller did not intend. Each leg then applies its command through the first-order lag τ_a, with zero force on a foot in the air, and the wrench is shared either among every foot the schedule calls planted or, by default, only among those whose contact switch also reports ground.

13.4.3 Trot and Foot Placement

The trot moves the diagonal pairs, front right with rear left and front left with rear right, half a period apart. With period $T = 0.36\,\mathrm{s}$ and duty factor $D = 0.55$ each pair stands for 0.198 s and swings for 0.162 s, so both pairs are down together for 18 ms at each hand-over. The timing is fixed, and a foot that lands late or early does not move the schedule. The capacity the allocation may count on ramps up from 30 % over the first 20 ms of a stance and down over its last 30 ms. Each swing foot aims at the touchdown point of Equation (13.11).

$$x_{td} = x_{hip}(t_{td}) + \frac{\hat{v}_x T_{st}}{2} + k_v\,(\hat{v}_x - v_{SP}) \tag{13.11}$$

where $x_{hip}(t_{td})$ is the hip position predicted at touchdown, $T_{st} = DT$ is the stance time and $k_v = 0.08\,\mathrm{s}$ moves the foot forward to brake when the body is too fast [7], with the offset from the hip limited to ± 0.12 m. The path from lift-off to x_{td} follows a smoothstep in both coordinates plus a lift of $h_{sw} \sin \pi s$ at swing fraction s, at no more than v_{sw} and inside the reach of the leg. A blind foot aims at the ground height the planted feet stand on, extrapolated along the line through them, and a mapped foot aims at the true ground height with half of any rise added to its lift. A swing that ends in the air keeps reaching down at 0.4 m/s, and a leg that reaches 97.5 % of l_{max} without finding ground records a missed foothold and waits for its next swing. Standing, a foot that has lost the ground is put back under its hip within 0.25 s.

13.4.4 Predictive Force Planner

The planner replaces the loops and the allocation together. It is a receding-horizon controller of the kind outlined in Section 1.6.3, written for this plant. At every 10 ms mark of the clock, and again whenever a foot lands, lifts or misses its foothold, it chooses the ground-frame forces of every foot the schedule expects to be down over $N = 20$ intervals of 10 ms, each held constant within its interval, on the planar single rigid body the controller believes in, as shown in Equation (13.12).

$$\begin{aligned} \min_{\mathbf{u}_0 \dots \mathbf{u}_{N-1}} \; & \sum_{k=1}^{N} (\mathbf{s}_k - \mathbf{r}_k)^{\top} Q (\mathbf{s}_k - \mathbf{r}_k) \\ & + \sum_{k=0}^{N-1} \sum_{j} \left[R_t t_{jk}^2 + R_n (n_{jk} - \bar{n}_k)^2 \right] \\ \text{s.t. } & \mathbf{s}_{k+1} = A_d \mathbf{s}_k + B_k \mathbf{u}_k + \mathbf{d}_g + \hat{\mathbf{d}}, \quad \mathbf{u}_k \in \mathscr{P}_k \end{aligned} \tag{13.12}$$

where $\mathbf{s} = (x, z, \varphi, v_x, v_z, \omega)$ starts from the measurements, A_d is an exact double integrator per axis over one interval, $\mathbf{d}_g$ is gravity, B_k maps the forces of the feet planned to be down in interval k, at their present positions or at their planned footholds, through the believed normals and torso-centered moment arms of the nominal mass and inertia, and $\bar{n}_k$ is an equal share of the weight among those feet. The diagonal of Q is 4×10^3, 3×10^5, 3×10^4, 1.5×10^3, 3×10^3 and 2×10^2 in SI units, with R_t and R_n in Table 13.3. The reference advances the position at the speed setpoint from the present estimated error, limited to ± 0.3 m, and holds the height and pitch setpoints with zero vertical and

angular rates. Each $\mathscr{P}_k$ is the same set of triangles as the allocation uses, including the loading ramps. The problem has up to 160 force variables when all four feet stand and 89 on average in the trot of case 8, and it is solved by 40 iterations of the same accelerated projected gradient, with the gradient from one forward rollout and one adjoint sweep and the plan warm-started from the previous one shifted by an interval. Only the first interval is applied.

The integral action of the planner is the estimate $\hat{\mathbf{d}}$, a constant vertical and angular acceleration that its model fails to explain, updated at each solve from the error between the velocities it predicted and those it measures, the prediction being interpolated along the interval when a contact change brings the solve forward, with a time constant of 0.25 s and limits of ±3 m/s^2 and ±15 rad/s^2. The horizontal channel is left out because slip corrupts the odometry, and an estimator there would integrate a sliding foot's speed into more push. Volume 9, Chapter 4 obtains the same offset-free action in a linear MPC from a fuel bias carried by its Kalman estimator. The estimate is frozen whenever the support differs from the one the last plan assumed.

13.4.5 Simulation, Analysis and Controller Models

Equations (13.1) to (13.6) form the simulation model, which carries the true mass properties, the true ground at each foot and contact modes that switch at hard events. The analysis model of Section 13.5 is its reduction about standing with every request feasible, in which each loop acts on a double integrator. The internal model of the controller is smaller than either. The allocation knows the nominal geometry, the believed slope and the friction it assumes, and the planner knows a single rigid body of the nominal mass and inertia with a disturbance estimate, while neither knows the backpack, a mass or inertia set in the Parameter panel, the true friction of the patch, a leg's derating unless told, or the slip of a foot.

13.5 What the Feet Can Deliver

The statements of this section follow from Equations (13.1), (13.4), (13.8) and (13.9) by closed-form reasoning, and each is checked against the running engine.

13.5.1 Three Double Integrators While Every Request Is Feasible

When the allocation meets the wrench exactly and the leg lag is neglected, Equation (13.1) closed by Equation (13.8) splits into three independent loops around double integrators. The characteristic polynomial of the height loop is Equation (13.13), the pitch loop has the same form with I and the pitch gains, and the speed loop acting on the position error gives Equation (13.14).

$$ms^3 + K_{c,h}\tau_{D,h}s^2 + K_{c,h}s + K_{c,h}/\tau_{I,h} = 0 \tag{13.13}$$

$$ms^2 + K_{c,v}s + K_{c,v}/\tau_{I,v} = 0 \tag{13.14}$$

At the defaults the height loop has roots at -2.73 and $-6.13 \pm 7.34j$ 1/s, and without its integral it would be an oscillator of 11.2 rad/s with a damping ratio of 0.67. The pitch loop has roots at -2.04 and $-8.98 \pm 9.10j$ 1/s, or 14.1 rad/s with a damping ratio of 0.71 without its integral, and the speed loop has real roots at -1.46 and −8.54/s.

A crouch checks the reduction. Stepping the height setpoint from 28 to 24 cm at $t = 1$ s drives the allocated vertical force from 117.7 N down to 55 N and up to 138.3 N, and the body undershoots to 23.03 cm 0.35 s after the step, where Equation (13.13) with its setpoint zero predicts 22.87 cm at 0.38 s. The difference is the 4 ms leg lag and the noise that the linear model leaves out. The zero at $-1/\tau_{I,h} = -2/\text{s}$, slower than every closed-loop root, produces the undershoot, since with the proportional term acting on the measurement the same poles give none, and the body stays within 2 mm of the new setpoint only from 0.87 s after the step.

The regularization of Equation (13.9) leaves a small shortfall that manual mode exposes. With four feet sharing a vertical request F, the vertical part of Equation (13.9) gives $\sum n_j = F/(1 + \lambda_n/4)$, which for the 117.72 N weight is 0.059 N short. In manual mode nothing corrects it, and the engine

Table 13.4 The Calibrated Push at Three Speed Gains, Against the Linear Speed Loop of Equation (13.14), Which Assumes Any Braking Is Available

$K_{c,v}$ (N s/m)	60	120	300
roots (1/s)	−2.50, −2.50	−1.46, −8.54	−1.32, −23.7
initial braking request (N)	50	100	250
linear slide (cm)	12.3	6.8	3.0
engine slide (cm)	13.1	10.1	7.5
peak braking (N)	43.4	56.6	86.4
nose-down dip (°)	3.95	5.11	9.30
recovery (s)	1.02	0.25	0.33
largest push survived (N s)	11.0	11.5	12.0

measures a mean shortfall of 0.059 N and a sink of 6.15 cm in 5 s, against the 6.13 cm that a uniform 0.059 N on 12 kg gives. A wrench that balances the weight does not hold a pose.

13.5.2 Two Limits on Braking

A standing robot brakes a push with horizontal forces at its feet, a height h below the center of mass, and those forces pitch the body nose-down. The vertical forces must supply the balancing moment, and the rear feet, which cannot pull, reach zero first. With the rear feet unloaded, the moment balance about the center of mass gives the bound of Equation (13.15).

$$F_b \le \min\left(\mu_{plan} F_z,\ \frac{x_f F_z - M_y}{h}\right) \tag{13.15}$$

where F_b is the braking force, x_f is the distance the front feet stand ahead of the center of mass, F_z is the vertical force and M_y is the nose-up moment the pitch loop still requests. Standing still with no moment request, the tipping term is $L_h mg/h = 79.9$ N against a friction term of 70.6 N, so the planned cone would bind first. A 10 N s push changes that, because the body moves over its feet and the pitch loop starts to ask for a nose-up moment. At the braking peak, 30 ms after the push, the front feet stand only 16.6 cm ahead of the center of mass and the rear feet carry nothing. The loops ask for 82.6 N of braking, 120.6 N of vertical force and 9.82 N m nose-up, and the allocation, ranking the moment first, returns 56.6 N, 135.6 N and 6.78 N m, pressing the front feet harder than asked to buy moment. With those values Equation (13.15) gives 56.1 N against a friction term of 81.4 N. Tipping, not friction, caps the braking. Braking at a constant 56.6 N from the first instant would stop the body in $mv_0^2/(2F_b) = 7.37$ cm.

Table 13.4 compares the linear speed loop with the engine. At the softest gain the loop asks for 50 N at the start, below the tipping cap, and the slide of 13.1 cm is close to the linear 12.3 cm. At the default and the stiffest gains the requests of 100 and 250 N are cut to what tipping allows, and the slides are 10.1 and 7.5 cm instead of 6.8 and 3.0 cm. The stiffest loop brakes harder, up to 86.4 N, because at that instant the allocation lifts the vertical force to 159.8 N against the 121.5 N asked and settles for 4.56 N m of the 11.7 N m of moment asked. What a stiffer loop buys is a larger recoverable push. The largest push the standing robot survives, found on the 0.5 N s steps of the push setting, rises from 11.0 to 11.5 to 12.0 N s as the gain rises from 60 to 120 to 300, and a finer 0.25 N s grid finds the same edges. The slides at the edge are 16 to 18 cm, close to the 19 cm between the center of mass and the front feet. The price is the pitch dip, which at the calibrated push grows from 3.95 to 9.30°. A push beyond the edge cannot be stopped inside the support at all. At the default gain 12 N s carries the body 40.6 cm until its chest touches the ground at 1.79 s, and only a step could have caught it [16].

13.5.3 Offsets From an Unmodeled Load

The bias of the height loop is the nominal weight, so with its integral off the 39.24 N of a 4 kg pack must come from the proportional term, a sag of $m_b g / K_{c,h} = 2.62$ cm. The pack also sits 0.10 m behind the torso center about which the controller takes moments, and its weight tilts the body nose-up until the proportional pitch term balances it, as shown in Equation (13.16).

$$K_{c,\varphi}\,\varphi_{ss} = m_b g\,(0.10\cos\varphi_{ss} + 0.085\sin\varphi_{ss}) \tag{13.16}$$

Its solution is 3.96°, the second term being the extra lever that the tilt itself gives the raised pack. With both integrals off the engine settles at 25.37 cm, a sag of 2.63 cm, and at 3.96°. With integral times of 0.5 and 0.6 s both offsets vanish, the body is back within 2 mm and 0.2° of its setpoints 1.58 s after the pack goes on, and the feet settle at 44.5 N behind and 34.0 N ahead, against the 44.4 and 34.1 N that a zero moment about the shifted center of mass requires.

13.5.4 Braking on a Slippery Patch

On the patch, friction replaces tipping as the binding term, and no controller can beat it. A body moving 0.833 m/s faster than its reference and braked by the full kinetic friction $\mu_k m g$, with $\mu_k = 0.1275$, needs 0.67 s and 27.8 cm to return to the reference speed, or 23.6 cm at the static coefficient. The robot runs 28.2 cm ahead of its position reference with the allocation planning at 0.6 and 29.0 cm at 0.15, so the friction assumption changes almost nothing about the surge. What it changes is the moment. At 3.36 s, 60 ms after the push, the allocation planning at 0.6 commands 68.2 N of braking and the ground returns 14.8 N. The vertical forces were set to cancel the 19.1 N m nose-down moment that the planned braking would produce at the feet, only 4.2 N m of it arrives, and the body receives 12.3 N m nose-up where the controller commanded 2.95 N m nose-down. Planning at 0.15, the commanded braking of 17 to 21 N is close to the 15 to 18 N the ground returns, and once the 4 ms leg lag has passed the moment the body receives stays within about 1 N m of the moment commanded.

🕮 **Pause.** Try Exercise 13.2 before reading on.

13.5.5 What the Energy Figure Measures

The application reports an energy proxy per meter walked, the integral over the planted legs of the absolute mechanical power $|\mathbf{F}_i \cdot (\mathbf{v}_{hip,i} - \mathbf{v}_{foot,i})|$ plus a copper-loss term $k_{cu}\|\mathbf{F}_i\|^2$ with an assumed $k_{cu} = 0.01\,\mathrm{W/N^2}$ per leg, standing for the heat the motors dissipate while they hold a force, one of the principal losses of a legged robot [19]. With massless legs and feet that stay put the mechanical term is small, and in a trot two feet share the weight for most of the cycle, so the proxy reduces to Equation (13.17).

$$\frac{E}{d} \approx \frac{2k_{cu}(mg/2)^2}{v} = \frac{k_{cu}(mg)^2}{2v}, \qquad \mathrm{CoT} \approx \frac{k_{cu} m g}{2v} \tag{13.17}$$

where E is the energy in J, d is the distance walked in m and $\mathrm{CoT} = E/(mgd)$ is the cost of transport. At 0.4 m/s it gives 173 J/m and a cost of transport of 1.47, against 178.3 J/m and 1.51 measured over 8 s of trotting in case 8 with the leg at full capacity, of which the copper term is 99 %. The figure therefore measures how evenly and for how long force is held rather than mechanical work, and swing clearance, which costs nothing when the legs are massless, leaves it at 178.31 J/m for every clearance from 3 to 12 cm.

13.6 Results Across the Cases

The nine cases are listed in Table 13.5. Their outcomes, measured at the default seed with the stated change and nothing else, are gathered in Table 13.6. Each change is made after the case is loaded and before the run starts, which gives the same run as the change carried through a reset. Runs last 4 to 17 s. Most results are peaks, times, distances or counts, and the few averages, the rms height and the energy per meter, are quoted with their windows. Figure 13.3 redraws the pitch ranges of cases 5, 8 and 9 from the table.

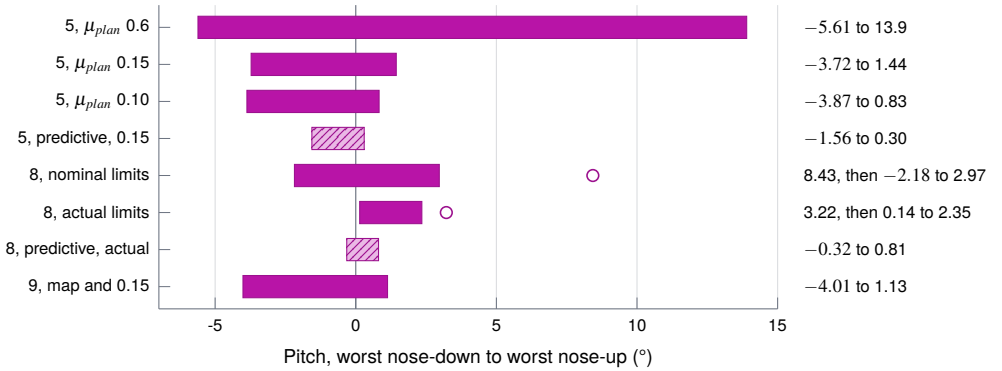


Figure 13.3 Pitch ranges of Table 13.6 for the slippery patch (case 5), the weak leg (case 8) and the graded course (case 9). Solid bars are the reactive allocation and hatched bars the predictive planner. In case 8 the open circle marks the first nose-up step and the bar the range that follows it. Notice that the widest range comes from planning at 0.6 on a patch of 0.15, and that the planner gives the narrowest range in both cases where it runs.

Table 13.5 The Nine Cases and What Each Isolates

#	Scenario and setup	Teaching point
1	Stand quietly	feedback holds a pose
2	Push recovery, 10 N s at 1 s	friction and tipping cap braking
3	Added backpack, 4 kg, integrals off	offsets and integral action
4	Ramp, trot at 0.4 m/s up 25°	leg reach and body posture
5	Slippery patch, push at 3.3 s	plan inside the true cone
6	Raised step, blind, $h_{sw} = 5$ cm	clearance or terrain knowledge
7	Missed contact, forces to scheduled feet	give force only to contacts
8	Force-limited leg at 30 %	reactive against predictive
9	Challenge, 3.8 m course, push at 5.5 s	everything at once

13.6.1 Standing, a Push and a Load

Standing on flat ground between 1 and 5 s, the four feet carry 29.42 to 29.43 N each, 117.71 N in all against a 117.72 N weight, averaged at the physics step, and the true height stays within 0.25 mm rms of 28 cm through 1.5 mm of height noise. The push of case 2 and the pack of case 3 follow the analysis of Section 13.5. The predictive planner answers the calibrated push differently from the reactive loops. It brakes harder, up to 91.2 N, while applying almost no net moment (–0.04 N m), which is about the full tipping capacity of Equation (13.15) with M_y near zero (90.5 N at that instant), and it holds the nose-down dip to 1.36° against 5.11°, at the cost of a slightly longer slide, 10.35 cm against 10.10 cm, and a recovery of 0.36 s.

13.6.2 Posture on a Ramp

Trotting at 0.4 m/s onto the 25° ramp with the pitch measured from level, the robot keeps its back horizontal, so the ground under the front hip is 17.7 cm higher than under the rear hip, and the rear legs must stretch while the front legs fold. The rear legs reach their full 38 cm. The rear left foot loses the ground at 3.05 s and, searching at full extension, cannot reach the ramp again, the rear right foot does the same at 3.21 and 3.51 s, and the front right knee touches the ramp at 3.82 s. The fall is certain but its timing is not, and over the default seed and seven others it comes between 3.54 and 3.84 s, by a knee or a leg folded to its limit, after two or three reach losses. Measured from the ground, the pitch setpoint follows the estimated slope, the body tilts up to 25.6°, the longest planted

Table 13.6 Measured Results at the Default Seed

Case and condition	Result
1, automatic, 1 to 5 s	29.42 to 29.43 N per foot, h 0.25 mm rms
1, crouch to 24 cm	low point 23.03 cm, within 2 mm after 0.87 s
1, manual, wrench equal to weight	0.059 N short, sinks 6.15 cm in 5 s
2, default	slide 10.1 cm, dip 5.11°, recovery 0.25 s
2, $J = 12$ N s	slides 40.6 cm, falls at 1.79 s
2, predictive planner	slide 10.3 cm, dip 1.36°, recovery 0.36 s
3, integrals off	25.37 cm, 3.96° nose-up
3, integrals on	within 2 mm and 0.2° after 1.58 s
4, level body	leg at its 38 cm reach, knee down at 3.82 s
4, pitch from the ground	longest leg 31.12 cm, climbs
4, level body, $h_{SP} = 25$ cm	longest leg 35.86 cm, climbs
4, level body, predictive	climbs to 2.80 m, knee down at 7.17 s
5, $\mu_{plan} = 0.6$	slip 15.6 cm, −5.61 to 13.9°, 1.42 s
5, $\mu_{plan} = 0.15$	slip 1.07 cm, −3.72 to 1.44°, 0.59 s
5, $\mu_{plan} = 0.10$	no slip, −3.87 to 0.83°, 0.77 s
5, predictive, $\mu_{plan} = 0.15$	slip 0.99 cm, −1.56 to 0.30°, 1.16 s
6, $h_{sw} = 5$ cm	six toe stubs, falls at 2.81 s
6, $h_{sw} = 7$, 8 cm	two stubs, none, both climb
6, terrain map, 5 cm	no stub, −0.15 to 0.05°
6, predictive, blind 5 cm	five stubs, falls at 2.81 s
7, scheduled feet	falls at 2.13 s, nose 27.9° down
7, sensed contacts	worst −13.28°, walks on
7, sensed, predictive	worst −3.13°, four misses, past at 6.22 s
7, scheduled, predictive	falls at 2.49 s
8, nominal limits	8.43°, then −2.18 to 2.97°, 0.40 m/s
8, actual limits	3.22°, then 0.14 to 2.35°, 0.317 m/s
8, predictive, actual limits	−0.32 to 0.81°, 0.40 m/s
9, as shipped	seven stubs, falls at 7.63 s
9, terrain map and $\mu_{plan} = 0.15$	finish 9.25 s, −4.01 to 1.13°, 0.55 s
9, $h_{sw} = 8$ cm, $\mu_{plan} = 0.15$	finish 9.57 s, recovery 2.54 s
9, terrain map only	finish 9.68 s, recovery 0.77 s, slip 4.58 cm

leg reaches only 31.12 cm, and the robot climbs to 4.66 m in 12 s. A level back at a 25 cm height also climbs, with its longest leg at 35.86 cm. Friction does not bind on this ramp, because the 0.8 of the floor and the 0.6 of the planning cone both exceed the $\tan 25° = 0.47$ that holding the weight requires. With a level body the predictive planner avoids the early reach losses and climbs to 2.80 m, but its front left knee touches the ground at 7.17 s. The outcome is marginal, since at two of the eight seeds the planner climbs the whole ramp to 4.70 m.

Pause. Try Exercise 13.1 before reading on.

13.6.3 The Slippery Patch

Walking steadily across the patch before the push, no planted foot carrying more than 1 N uses more than 67.8 % of the real cone, so the wrong assumption costs nothing until the push arrives at 3.3 s. A front foot touching down at 3.05 s with 0.6 N of load slides for an instant, no measurable distance, and the application counts and marks a slip only on a foot carrying more than 1 N, so the first slip marked in Figure 13.5 comes after the push. After the push, planning at 0.6 slides the feet 15.6 cm, holds the pitch between −5.61° and 13.9°, and takes 1.42 s to recover, and the odometry, fooled by the sliding feet, ends 8.3 cm behind the true position. Planning at the true 0.15, the feet slide 1.07 cm, the pitch stays between −3.72° and 1.44° and recovery takes 0.59 s. The predictive planner with the true cone narrows the pitch to −1.56° to 0.30° but takes 1.16 s to recover.

Sweeping the assumed coefficient shows that the choice is a trade, as shown in Figure 13.4. Slip and the nose-up peak rise with the excess of μ_{plan} over the true value up to about 0.6 and change little beyond it, reaching 16.3 cm and 14.5° at 0.8, while recovery slows from 0.59 to 1.44 s. Planning exactly at the true coefficient is not the best choice, because it puts every braking command on the edge of the real cone, where pitch noise and the leg lag tip it over. The setting moves in steps of 0.05,

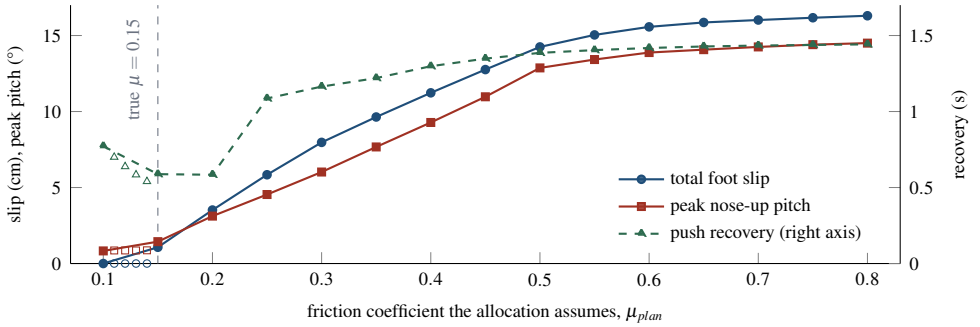


Figure 13.4 Case 5 with only the friction coefficient the allocation assumes changed. Foot slip and the nose-up peak grow with the excess over the true coefficient of 0.15. Filled markers lie on the 0.05 steps of the setting, and hollow markers from 0.11 to 0.14 are counterfactual runs between them, where the fastest recovery, 0.540 s with no slip, comes at 0.14.

and the next step down, 0.10, never slides a foot but takes 0.77 s to recover against 0.59 s at 0.15. Counterfactual runs between the two steps, which a learner cannot set, locate the optimum. At 0.14 the slip is exactly zero and recovery is fastest, 0.540 s, and further conservatism costs time without any gain, 0.64 s at 0.12.

> ✦ DID YOU KNOW? **The Steadier Planner Recovers Later**
>
> From the simulator: in Table 13.6 the predictive planner holds the pitch nearest to level in cases 2, 5 and 8, a dip of 1.36° against 5.11° after the push, yet it recovers later, in 0.36 s against 0.25 s after the push and in 1.16 s against 0.59 s on the patch at the true friction.

13.6.4 A Step and a Hole

Blind with 5 cm of clearance, the feet pass the 6 cm step low in their arc, strike its face six times, and the robot pitches onto its chest at 2.81 s. At 7 cm it strikes twice and climbs, and at 8 cm it climbs with no strike and a pitch between −0.25° and 0.11°. With the terrain map each foot aims at the top of the step and its path rises to meet it, so even 5 cm climbs without a strike, between −0.15° and 0.05°. The predictive planner does not rescue the blind 5 cm swing, which falls at 2.81 s after five strikes, because no choice of forces moves a foot that a face has stopped. Over the default seed and seven others the blind 5 cm swing strikes six or seven times and falls between 2.81 and 2.83 s, and 8 cm never strikes.

In case 7 the allocation first shares the weight among the feet that the schedule says are down. A front foot drops into the 12 cm hole, its share of the support goes nowhere, and the robot falls at 2.13 s, when its front left knee touches the ground with the nose 27.9° down at that physics step (the worst pitch on the scorecard, sampled at the control call before it, reads 27.1°). With force sent only to feet whose contact switch reports ground, the front left foot is stopped by the far wall of the hole at 2.06 s, reaches full extension without finding ground at 2.29 s, and the remaining feet carry the body through a worst nose-down pitch of 13.28°, after which the robot walks on. The predictive planner, which plans the landing of the next pair into the forces it applies now, holds the drop to 3.13°, although with forces sent to the scheduled feet it too falls, at 2.49 s. With sensed contacts it also brakes the lurch over the hole back to its position reference, so the rear feet land short, at the edge of the hole, and miss in turn. The robot rocks in place, its nose up to 8.21°, strikes the far wall six times in all and misses four footholds before all four feet are past the hole at 6.22 s. The position weight is the cause. In a counterfactual run with a quarter of the weight Q_x, the planner misses only the first foothold and is past the hole at 2.94 s, against 2.75 s for the reactive allocation.

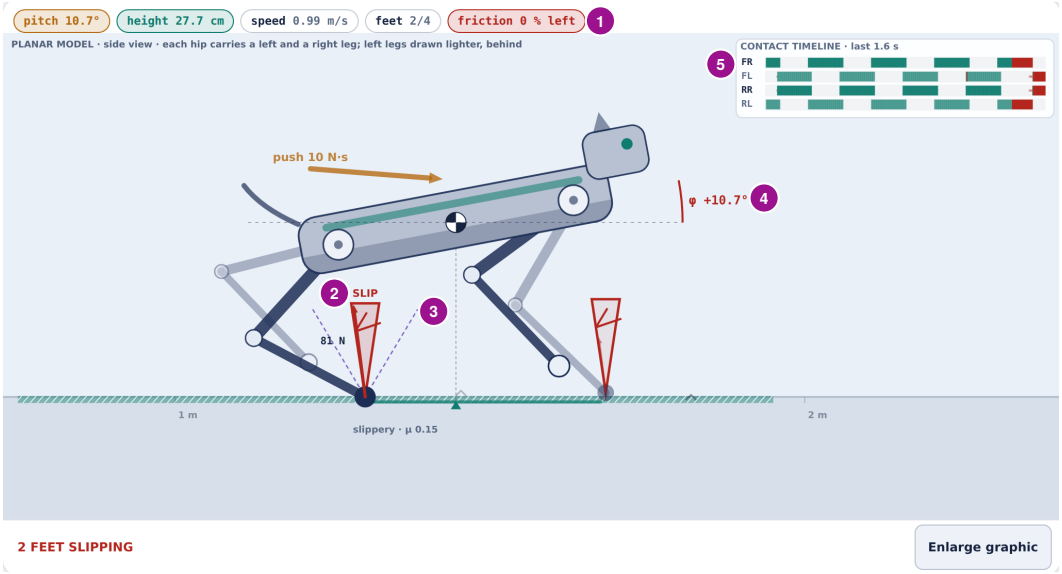


Figure 13.5 The scene panel of the quadruped application in scenario 5, Slippery patch, captured at $t = 3.50$ s, 0.20 s after the 10 N s push. The robot trots on the $\mu = 0.15$ patch with the allocation assuming 0.6, and both feet of the planted diagonal pair slide. (1) The friction badge reads 0 % left, since a sliding foot has no reserve, and the status line below the scene reads 2 feet slipping. (2) The near rear foot, carrying 81 N and labeled SLIP, has its true cone outlined in red, as has the far front foot. (3) The dashed planning cone at the near foot is visibly wider than the true one. (4) The body is pitched 10.7° nose-up under the fading push arrow, and the speed badge reads 0.99 m/s. (5) The contact timeline shows the pair that took the push sliding in red before it lifted and the pair now down sliding since it landed. At that instant the far front foot carried 49 N, the speed setpoint was 0.40 m/s and the odometry read 0.84 m/s, the allocated horizontal force was −65 N against −66 N asked and the pitch moment −18.1 N m, and the first loaded foot had slid 4.5 ms after the push.

> **TRY IT IN THE APP** **7 · Missed Contact**
>
> Load the scenario and press Run. Watch the front foot that steps into the hole and the contact timeline: with the force shared among the scheduled feet, the robot falls at 2.13 s. Press Reset, set Forces go to to Sensed contacts and run again. The remaining feet now carry the body through a worst nose-down pitch of 13.28°, and the robot walks on.

Pause. Try Exercise 13.3 before reading on.

13.6.5 A Weak Leg, Reactive and Predictive

With the rear-left leg at 30 % of its rating, 45 N, and the allocation planning with nominal limits, the allocation asks that leg for 138.8 N in the first weak stance and about 90 N in every later one, 82.8 to 93.1 N after the first second, and the leg delivers 45 N. The body pitches up 8.43° at 0.20 s and then rocks between −2.18° and 2.97° at 0.40 m/s, with the mean pitch 2.2° apart between the stance of the weak pair and that of the strong pair. The imbalance is structural. With only the weak pair down, a front foot and a capped rear foot must carry 117.7 N, the front foot must take 72.7 N, the vertical forces leave a nose-up moment of 5.3 N m, and canceling it with horizontal force at the feet would take 18.8 N of braking. Planning with the actual limit, the allocation stays feasible by giving up horizontal force rather than moment, which carries the largest weight in Equation (13.9). After a

first step of 3.22° the pitch stays between 0.14° and 2.35° while the speed falls to 0.317 m/s, averaged from 2 to 10 s. The predictive planner with the actual limit sees each weak stance coming inside its 0.2 s horizon and spreads the correction across the cycle. The pitch stays between −0.32° and 0.81° at 0.40 m/s, and the mean pitch of the two stances differs by 0.2°. Over 8 s the energy proxy charges the three configurations 213, 259 and 204 J/m against 178 J/m with a healthy leg, the slower walk and the uneven sharing of the second both adding held force per meter, as Section 13.5.5 explains.

> **TRY IT IN THE APP** **8 · Force-Limited Leg**
>
> Load the scenario and press Run. With the rear-left leg at 30 % of its rating, the body pitches up 8.43° and then rocks between −2.18° and 2.97°. Press Reset, set Leg force limits it plans with to Actual and run again. After a first step of 3.22° the pitch stays between 0.14 and 2.35°, but the walk slows to 0.317 m/s.

13.6.6 The Graded Course

The course runs over flat floor, an 8° ramp, a patch with $\mu = 0.15$ and a 6 cm step to a finish at 3.8 m, with the calibrated push at 5.5 s on the patch. The target is to finish without a fall, keep the pitch within ±10°, recover from the push within 2 s and let the feet slide less than 2 cm in total. The attempt is judged when the run stops itself, 3 s after the finish line or 0.35 s after a fall, on the case as given, so changing a model parameter or a disturbance voids it. As shipped, the push slides the feet 4.60 cm and is recovered in 0.77 s, but the blind 5 cm swing strikes the step seven times from 6.93 s on and the robot falls onto its chest at 7.63 s, with a worst pitch of 25.6°. With the terrain map and the allocation planning at 0.15 the robot crosses the finish at 9.25 s, the pitch never leaves −4.01° to 1.13°, the feet slide 0.46 cm and the push is recovered in 0.55 s, at all eight seeds tried. Against the target these are margins of 6.0° of pitch, 1.45 s of recovery and 1.54 cm of slip.

An 8 cm clearance with $\mu_{plan} = 0.15$ also finishes, at 9.57 s, but recovers only after 2.54 s and fails. Near the step the blind slope estimate tilts by 8.9 to 9.6°, more than the 8.5° half-angle of a cone planned at 0.15, so every force the allocation may use points slightly backward, the allocated forward force stays near −1 N against requests of up to 18.0 N, and the speed sags to 0.24 m/s before the robot is over the step. The failure is marginal. At the default seed the recovery from the push has not been held for its 0.3 s before the robot reaches the step, while at the seven other seeds it has, after 0.53 to 0.79 s, and the same settings pass. A 7.5 cm clearance with 0.15, or 8 cm with $\mu_{plan} = 0.20$, also passes at the default seed. The terrain map alone finishes at 9.68 s with a recovery of 0.77 s and a worst pitch of 5.24°, and an 8 cm clearance alone finishes at 9.61 s, but the push slides their feet 4.58 cm and 3.66 cm, past the 2 cm the target allows, at every seed tried. The step lesson of case 6 and the friction lesson of case 5 are both needed, and the target does not single out the terrain map among the step fixes.

13.7 Verification of the Engine

All 35 checks that the application carries pass on the version measured here. A standing robot with a quarter of the weight on each foot has derivatives below 10^{-12}, the closed-loop vertical forces at rest match the weight within 0.2 %, and no foot in the air carries force at any physics step of a 3 s trot. Neither the allocation nor the planner exceeds a friction or force limit at any control step of four whole runs or anywhere over the planner's horizon, the planner's iterations never raise its cost, and reloading the same scenario reproduces a run exactly. The results of this chapter use the default seed, and the orderings of the reactive cases survive other noise realizations. Over the default seed and seven others, planning at 0.10 on the patch never slides a foot and recovers in 0.772 to 0.780 s (0.540 to 0.546 s at the counterfactual 0.14), planning at 0.15 slides 0.93 to 1.09 cm, planning at 0.6 slides 15.52 to 15.78 cm, and the calibrated push of case 2 slides 10.10 to 10.28 cm. Two kinds of result are less robust. The 8 cm clearance with $\mu_{plan} = 0.15$ fails the challenge only at the default seed, as Section 13.6 reports. The predictive planner solves on the 10 ms marks of the clock and at every contact change, so the instant at which it is selected no longer decides its outcome. Selected at

0, 2, 4, 6, 8, 10 or 20 ms, it gives the same push response, the same recovery on the patch and the same crossing of the hole at the default seed. Its marginal results are the level-body ramp, which it climbs at two of eight seeds and falls on at six, and the hole with sensed contacts, which it crosses at seven of eight seeds and falls in at the eighth, at 5.64 s.

13.8 Technology in Practice

A real legged robot reads and moves what the loops of Section 13.4 read and move. Figure 13.6 draws the three loops, the allocation and the planner with field hardware in place of the schematic's symbols.

What Measures It

The controlled variables are the body's height, pitch and forward speed. The model takes the pitch and its rate from an inertial unit with 0.1° and 0.02 rad/s of noise, and the height and speed from leg odometry built on the joint angles of the planted legs, which a contact switch on each foot names. Real robots typically do not use such raw readings directly, but turn them into the pose and velocity of the body with a state estimator [20].

> **TECHNOLOGY Sensor: Inertial Unit and Leg Kinematics**
>
> An inertial measurement unit on the torso measures the body's acceleration and turn rate, and encoders on the joints give the joint angles, from which the known leg geometry places each foot relative to the body. Both carry noise, and the inertial readings also carry biases that wander as random walks. An extended Kalman filter can fuse the two, taking each planted foot as fixed and adding noise to its position to allow a little slip. With one foot or more on the ground, pitch, roll and velocity are observable, while position and heading are not, and faulty contact detection shows as a slow drift of the position estimate [17]. The model's unit and odometry carry white noise only.

What Moves It

Each foot force is made by the motors of its leg. A balance controller of this kind sets the ground force that the body needs and transforms it into joint torques [4], so each leg's own joint loops turn the force the allocation hands it into torques at the hip and knee. Real limits are those joint torques, whose force equivalent changes with the knee angle, as Section 13.3.6 notes, while the model rates every leg at one normal force.

> **TECHNOLOGY Final Element: Proprioceptive Leg Actuators**
>
> Hydraulic legs offer a large power-to-weight ratio but are expensive and need specialized upkeep. An electric motor behind a high-ratio gearbox gains torque, but the gearbox's static friction keeps the joint from being driven backward, so the torque cannot be set through the motor current, and the springs or torque sensors added after it limit bandwidth or suffer under impacts. A motor of high torque density with a low gear ratio sets the joint torque through its current alone, can be back-driven to soften an impact, and controls force at high bandwidth. Such proprioceptive actuators are now the dominant design for legged robots [20]. The model's leg reaches its commanded force after a 4 ms lag, up to a fixed 150 N rating that case 8 cuts to 30 %.

Where the Controller Runs

Reactive controllers of this kind take the state as known and constant over the control period, typically between 1 and 10 ms, and the set of feet in contact as given and fixed for each period. Posed as a quadratic program, with the friction cones approximated as pyramids, the problem is solved in hundreds of microseconds to a few milliseconds. A predictive layer on a simplified model commonly runs above it at a lower rate [21]. Learned controllers show the same split, with a policy updated at about 50 Hz and the joint torque still produced at 1 kHz [20]. In this chapter the loops and the

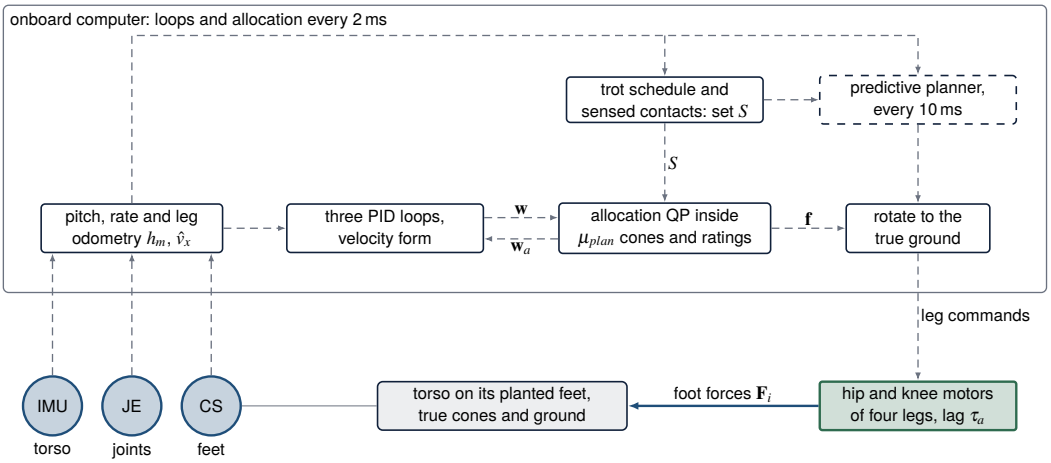


Figure 13.6 The balance loops on field hardware. An inertial unit (IMU) on the torso, encoders on the joints (JE) and a contact switch on each foot (CS) feed the pitch, the leg odometry and the set of planted feet. The three loops ask for a wrench $\mathbf{w}$, the allocation shares it among the feet that the trot schedule plans and the switches confirm, and the delivered wrench $\mathbf{w}_a$ returns to the loops. The forces are rotated into the true ground frame and sent to the hip and knee motors. The dashed planner replaces loops and allocation when selected. Notice that every limit the allocation respects is one it believes, set by μ_{plan} and the ratings it plans with, while the ground applies the true ones.

allocation form the reactive layer and run every 2 ms, inside that range, with each cone already a triangle in the plane, and the predictive planner, when selected, replaces both and solves every 10 ms and at each contact change.

What Is Modernizing It

Three developments bear on this robot. The first is the actuator. Proprioceptive actuators have become the dominant design, and open-source designs from universities and inexpensive hardware from industry have sped up experiments [20]. The second is learning. A controller trained in simulation with the robot's mass, the friction coefficient, the motor strength and the latency randomized transfers better to the real robot, much as a robust controller is designed for a set of parameters, and the approach is now widely adopted [20]. The third is the model inside predictive control. Recent work has used more complex models, some of them the full robot, made possible by better gradient-based methods and fast open-source dynamics libraries, and with Moore's law ending further gains may come from parallel computing [21]. The planner of this chapter keeps the single rigid body in the plane.

IN PRACTICE Soft or Strict Priorities When Tasks Conflict

The tasks given to a legged robot's balance controller conflict in practice, so the designer has to say how they trade. In the common quadratic-program approach, the friction cones and other inequalities stay hard constraints, and a weighted sum of the squared errors of the other tasks is minimized, a soft priority. Regularization terms keep the answer from jumping between consecutive control periods. Weights grow hard to tune as tasks multiply, and a strict hierarchy can rank the tasks instead, though the cost of its solvers has limited its use [21]. The allocation of this chapter is the soft kind, with the pitch moment weighted above the two forces, $W = \mathrm{diag}(1, 1, 30)$.

13.9 Limitations

Where each modeling assumption breaks down is stated in Section 13.3.6. Three of them bear on the results. Because the legs are massless, swing clearance and swing speed carry no dynamic or energetic cost, and a result that depends on either is optimistic. Leg ratings are constant forces rather than joint torques, and an assumed back-drive damping sets the speed of every slip.

The gait never adapts its timing, estimates its contacts beyond ideal switches, or steps to catch a push. The standing robot survives at most 11.0 to 12.0 N s, depending on the speed gain, and a larger push is a fall by construction.

The predictive planner is approximate, with 40 iterations per solve, and it does not rescue every case. It falls on the ramp with a level body, at the low step, and in the hole with forces sent to scheduled feet, and with sensed contacts it holds its position reference so tightly that it rocks at the edge of the hole for 3.5 s. Its disturbance estimate is limited to 3 m/s^2 vertically, which a 4 kg pack on the 12 kg model already exceeds, since it needs 3.27 m/s^2, and the planner then leaves a sag of 1.6 mm. At the 6 kg slider limit both limits are reached, leaving 6.7 mm and 0.35° where the reactive integrals leave 0.1 mm.

OPEN CHALLENGE A Safety Layer Over a Learned Walker

Controllers learned by reinforcement walk robustly outdoors yet struggle in safety-critical places such as stepping stones and confined spaces. One route to safety trains the policy under constraints. Another, for deployment, is a safety filter built from reachability analysis, control barrier functions or predictive methods. Such filters are well studied in control, but their use in locomotion has been limited, and more research on them could help certify legged robots for safety-critical service [20]. In the language of this volume, a safety filter is an override placed over a learned controller.

The sensors carry white noise only, the terrain map is perfect, and no estimator fuses the measurements. The energy figure is almost entirely an assumed copper loss. The numerical targets of the scenarios are design choices of a teaching simulator, not validated performance of a robot.

OPEN CHALLENGE Knowing the Friction and the Load

The friction under the feet and the payload on the back are the quantities this chapter's controllers lack. A recent review of learned locomotion calls measuring or estimating such factors accurately nearly impossible, yet needed for a controller to be robust or adaptive. Current methods build a belief from a history of measurements, or train a student policy, which sees only that history, to imitate a teacher policy that was given the friction and the terrain in simulation [20]. In this chapter the assumed friction is a setting, and case 5 measures what a wrong choice costs.

13.10 Conclusion

A legged robot balances only through the forces its feet can deliver, and the simulator of this chapter makes each limit on those forces measurable. A planar torso on four massless legs with unilateral, friction-limited and force-limited contact is held by three loops whose wrench an allocation shares

inside the friction cones, with a fixed-timing trot and an optional receding-horizon force planner. While every request is feasible the loops are three double integrators, and the cases are the moments when that stops being true. After a 10 N s push, tipping over the front feet caps the braking at 56.6 N while the planned cones would allow 81.4 N. A level body on a 25° ramp stretches its rear legs to their full 38 cm reach and falls, while a body laid along the ground climbs. On a slippery patch an allocation that assumes 0.6 slides the feet 15.6 cm and pitches the body up 13.9°, and one that assumes 0.10 slides them not at all. Forces sent to scheduled feet drop the robot into a hole, and only the predictive planner keeps both pitch and speed on a leg derated to 30 %. The model is planar, its legs are massless, its gait neither adapts its timing nor steps, and its energy figure is almost entirely an assumed copper loss. A fuller model would add roll, leg dynamics, contact estimation with event-based timing, recovery steps and friction estimation. The quadruped is an accessible plant, with nothing to install, for learning to allocate force against limits that change with every step.

KEY POINTS

- A walking robot moves its body only by pushing on the ground, and each foot pushes only inside its friction cone and only while it touches.
- While every request is feasible the loops act on three double integrators, and the cases are the moments when a limit binds.
- After a 10 N s push, tipping over the front feet caps the braking at 56.6 N, below the 81.4 N the planned cones allow.
- Assuming 0.6 on the patch slides the feet 15.6 cm, and forces sent to scheduled feet drop the robot into a hole.
- Only the predictive planner keeps both pitch and speed on a leg derated to 30 %.

Where else this applies. Volume 20, Chapter 6 shares a demanded force among a ship's thrusters, and Volume 20, Chapter 4 among effectors whose authority changes with the airspeed.

References

[1] J. Kerr and B. Roth, "Analysis of multifingered hands," *The International Journal of Robotics Research*, vol. 4, no. 4, pp. 3–17, 1986.

[2] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Autonomous Robots*, vol. 41, no. 1, pp. 259–272, 2017.

[3] M. Bodson, "Evaluation of optimization methods for control allocation," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 703–711, 2002.

[4] S. Hyon, J. G. Hale, and G. Cheng, "Full-body compliant human–humanoid interaction: Balancing in the presence of unknown external forces," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 884–898, 2007.

[5] J. A. Rossiter, B. Pasik-Duncan, S. Dormido, L. Vlacic, B. Jones, and R. Murray, "A survey of good practice in control education," *European Journal of Engineering Education*, vol. 43, no. 6, pp. 801–823, 2018.

[6] B. W. Bequette, "Process control practice and education: Past, present and future," *Computers & Chemical Engineering*, vol. 128, pp. 538–556, 2019.

[7] M. Raibert, M. Chepponis, and H. Brown, "Running on four legs as though they were one," *IEEE Journal on Robotics and Automation*, vol. 2, no. 2, pp. 70–82, 1986.

[8] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.

[9] R. B. McGhee and A. A. Frank, "On the stability properties of quadruped creeping gaits," *Mathematical Biosciences*, vol. 3, pp. 331–351, 1968.

[10] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "BigDog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10 822–10 825, 2008.

[11] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of HyQ – a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.

[12] B. J. Stephens and C. G. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1248–1255.

[13] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4399–4406.

[14] F. Jenelten, J. Hwangbo, F. Tresoldi, C. D. Bellicoso, and M. Hutter, "Dynamic locomotion on slippery ground," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4170–4176, 2019.

[15] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Fast, robust quadruped locomotion over challenging terrain," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2665–2670.

[16] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.

[17] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots: Consistent fusion of leg kinematics and IMU," in *Robotics: Science and Systems VIII*, 2012.

[18] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[19] S. Seok, A. Wang, M. Y. Chuah, D. J. Hyun, J. Lee, D. M. Otten, J. H. Lang, and S. Kim, "Design principles for energy-efficient legged locomotion and implementation on the MIT Cheetah robot," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, pp. 1117–1129, 2015.

[20] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, "Learning-based legged locomotion: State of the art and future perspectives," *The International Journal of Robotics Research*, vol. 44, no. 8, pp. 1396–1427, 2025.

[21] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 40, pp. 43–63, 2024.

13.11 Exercises

13.1 Concept check · 2 min

In scenario 4, Ramp, the robot trots up the 25° ramp with its pitch measured from level and falls when a knee touches the ramp. Which limit ends the run?

(a) The friction cone, because $\tan 25°$ exceeds the 0.6 the allocation plans with
(b) The leg rating, because the uphill legs must carry the weight at 150 N each
(c) The leg reach, because the rear legs stretch to their full 38 cm and lose the ground
(d) The pitch moment clamp, because holding the back level needs more than 60 N m

13.2 Concept check · 2 min

In scenario 5, Slippery patch, the allocation plans with a friction of 0.6 on a patch whose true coefficient is 0.15. After the push at 3.3 s the body pitches 13.9° nose-up instead of braking level. Why?

(a) The push lands above the center of mass and rotates the body nose-up
(b) The pitch integral winds up against the 60 N m moment clamp
(c) The IMU reads the slip as pitch, and the pitch loop follows the false reading
(d) The vertical forces cancel a nose-down braking moment that the ground never returns

13.3 Concept check · 2 min

In scenario 7, Missed contact, with Forces go to set to Scheduled feet, a front foot steps into the hole while the schedule says it is planted. What happens to the force the allocation assigns to that foot?

(a) It never reaches the body, and the allocation still believes it was delivered
(b) The contact law passes it to the diagonal partner, which then carries both shares
(c) The allocation reports it as a shortfall, and the loops treat it as saturation
(d) It reaches the body after the 4 ms leg lag, once the foot lands

13.4 Theory and calculation · 6 min

Use Equations (13.13) and (13.14) with the defaults of Table 13.3 and $m = 12\,\text{kg}$.

(a) Remove the integral from the height loop. Find the natural frequency and the damping ratio of $ms^2 + K_{c,h}\tau_{D,h}s + K_{c,h} = 0$. *Checkpoint:* 11.2 rad/s and 0.67.

(b) With the integral, $s = -2.73$ is a root of the cubic. Use the sum and the product of the roots to find the other two. *Checkpoint:* $-6.13 \pm 7.34\,j$ 1/s.

(c) Find the roots of the speed loop. *Checkpoint:* -1.46 and −8.54/s.

13.5 Theory and calculation · 6 min

Use Equation (13.15) with $L_h = 0.19\,\text{m}$, $h = 0.28\,\text{m}$, $mg = 117.72\,\text{N}$ and $\mu_{plan} = 0.6$.

(a) For the robot standing still with no moment request, $x_f = L_h$ and $F_z = mg$. Find the tipping term and the friction term. Which binds? *Checkpoint:* 79.9 N and 70.6 N, friction.

(b) At the braking peak after the 10 N s push, $x_f = 0.166\,\text{m}$, $F_z = 135.6\,\text{N}$ and $M_y = 6.78\,\text{N m}$. Find both terms again. *Checkpoint:* About 56 N and 81.4 N, tipping.

(c) The push starts the body at J/m. Braking at a constant 56.6 N from the first instant, how far does it travel before it stops? Compare with the engine slide of Table 13.4. *Checkpoint:* About 7.4 cm against 10.1 cm.

13.6 Theory and calculation · 5 min

In scenario 3, Added backpack, a 4 kg pack centered at $\mathbf{b} = (-0.10, 0.085)$ m goes on a 12 kg robot whose height and pitch integrals are off. Take $g = 9.81\,\text{m/s}^2$.

(a) Find the extra weight and the height sag it causes, given that the bias of the height loop stays at the nominal weight. *Checkpoint:* 39.24 N and 2.62 cm.

(b) Solve Equation (13.16) for the steady pitch. Start from the first-order value $m_b g\,0.10/K_{c,\varphi}$ and iterate once. *Checkpoint:* 3.75°, then 3.95°, and the solution is 3.96°.

(c) Find the offset $\mathbf{c}$ of the combined center of mass from Equation (13.2). *Checkpoint:* 2.50 cm back and 2.13 cm up.

13.7 App challenge · 8 min

Load scenario 2, Push recovery, at the default seed, and run it to $t = 4\,\text{s}$. Read the Worst pitch and Push recovery tiles of the scorecard.

(a) Report the nose-down dip and the recovery time at the default Speed gain of 120 N s/m. *Checkpoint:* 5.1° and 0.25 s, within 0.1° and 0.01 s.

(b) Press Reset, set Speed gain to 300 N s/m, and run to 4 s again. Then Reset with 60 N s/m. Report both pairs. *Checkpoint:* 9.3° and 0.33 s, then 3.9° and 1.02 s.

(c) Explain why the stiffest gain does not recover fastest, and why its dip is the largest. *Checkpoint:* Tipping caps braking, so a stiffer loop buys a harder pitch disturbance, not a proportional cut in the slide.

13.8 App challenge · 8 min

Load scenario 5, Slippery patch, at the default seed, and run it to $t = 8\,\text{s}$. Read the Foot slip, Worst pitch and Push recovery tiles.

(a) Report the three tiles with Friction the allocation assumes at its default of 0.6. *Checkpoint:* 15.6 cm, 13.9° (from −5.6) and 1.42 s.

(b) Press Reset, set the friction to 0.15 before pressing Run, and run to 8 s. Repeat with 0.10. Report the tiles for both. *Checkpoint:* 0.15: 1.1 cm, 3.7°, 0.59 s. 0.10: 0.0 cm, 3.9°, 0.77 s.

(c) The true coefficient is 0.15. Which setting would you ship for this patch, and why is the most conservative one not the best? *Checkpoint:* 0.15 recovers fastest of the three. Below it the allowed braking is smaller and recovery slows.

13.9 App challenge · 10 min

Load the challenge scenario, Challenge · short mixed course, at the default seed. Each run stops itself 3 s after the finish line or 0.35 s after a fall. Read the Robot, Course, Worst pitch, Push recovery and Foot slip tiles.

(a) Run it as shipped. When does it fall, and which part of the target had it already failed before the fall? *Checkpoint:* Falls at 7.63 s, after the push had slid the feet 4.6 cm.

(b) Press Reset, set Friction the allocation assumes to 0.15 and Footholds to Terrain map, and run again. Report the finish time, the worst pitch, the recovery and the slip, and the margin of each against the target. *Checkpoint:* 9.25 s, 4.0°, 0.55 s, 0.5 cm. Margins 6.0°, 1.45 s, 1.5 cm.

(c) Press the reload button, which restores the scenario's settings, and set only Footholds to Terrain map. Does the run pass? Explain with cases 5 and 6. *Checkpoint:* It finishes at 9.68 s but slides 4.6 cm and fails.

13.10 Application area · 4 min

The energy figure of Section 13.5.5 is a proxy with an assumed copper loss $k_{cu} = 0.01\,\mathrm{W/N^2}$ per leg.

(a) Evaluate Equation (13.17) for the trot at 0.4 m/s and compare with the measured 178.3 J/m. *Checkpoint:* 173 J/m and a cost of transport of 1.47, against 1.51 measured.

(b) The proxy reads the same 178.31 J/m for every swing clearance from 3 to 12 cm. Should a design team use it to choose the clearance of a real robot? *Checkpoint:* No. The legs are massless, so clearance costs nothing in the model.

13.11 Application area · 4 min

Friction cones in practice.

(a) Find the half-angle $\arctan\mu$ of the cone for the floor (0.8), the default plan (0.6) and the patch (0.15). *Checkpoint:* 38.7°, 31.0° and 8.5°.

(b) In the challenge with a blind 8 cm swing and $\mu_{plan} = 0.15$, the slope estimate near the step tilts by 8.9 to 9.6°. What does that do to the forward force the allocation may use, and what removes the problem? *Checkpoint:* Every allowed force points slightly backward. The terrain map supplies the true ground normal.

13.12 Review the agent · 5 min

A trainee asks an AI assistant how to make the standing robot of scenario 2 stop shorter after the 10 N s push. The assistant replies: *"The speed loop is a second-order system with roots set by Equation* (13.14)*. Raising the Speed gain from 120 to 300 N s/m moves the slow root and cuts the predicted slide from 6.8 to 3.0 cm, so the robot will stop in less than half the distance with no other effect."* Find the flaw and say how you would check it in the app.