INTERACTIVE CONTROL ENGINEERING
Principles and Applications

Volume 20 Motion Control, Robotics, and Autonomous Systems

CHAPTER 11

A Table Tennis Robot That Must Swing Before It Knows Where

Pages 258–279

Edited by John D. Hedengren

How to cite this chapter. "A Table Tennis Robot That Must Swing Before It Knows Where," in *Motion Control, Robotics, and Autonomous Systems*, John D. Hedengren, Ed., vol. 20 of *Interactive Control Engineering: Principles and Applications*. Advanced Process Solutions, LLC, 2026, ch. 11, pp. 258–279.

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

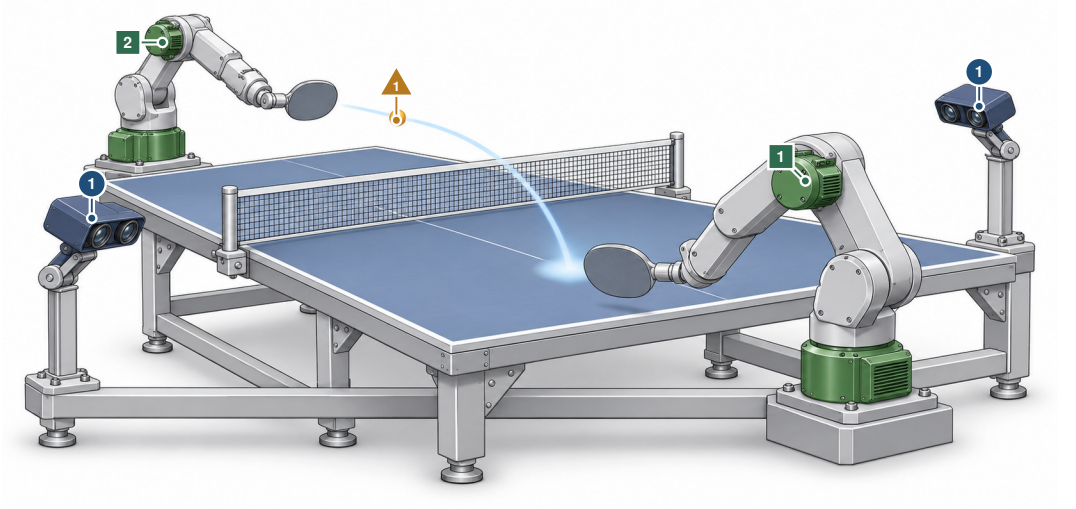
The app for this chapter
https://apmonitor.com/apps/table-tennis-robot/

The volume online: contents, other chapters and citation records
https://apmonitor.com/apps/books/v20-motion-robotics/#table-tennis-robot

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

CHAPTER 11

A Table Tennis Robot That Must Swing Before It Knows Where



The controller moves 1 joint motors of the near arm, which swing the paddle 2 joint motors of the opponent arm

The controller reads 1 camera pairs, which report the ball a little late

What disturbs it 1 spin and air, which bend the ball's path

Notice that the near arm must start its swing toward the end of the faint arc, a prediction, before the ball has arrived there. Illustration, not to scale.

IN PLAIN TERMS A Robot That Must Swing Before It Knows

The main idea. A robot arm plays table tennis, and the ball crosses the table in about half a second. Cameras watch the ball, and a computer works out where it will meet the arm's paddle.

The challenge. Every extra look makes that guess better but leaves the arm less time to swing. The arm must stop watching at the right moment, and its guess must include the air and the spin that bend the ball's path.

Why it matters. An arm that waits too long, or forgets the air, swings at an empty spot and loses the point.

THREE WAYS TO READ THIS CHAPTER

Explore	11 min	the first page, the fact on ball speed in the Introduction, the history segment (Section 11.2), the try-it segments of cases 7 and 1 (Sections 11.4 and 11.6), the fact on the commit time, 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 measured cases and the game (Section 11.6), Technology in Practice (Section 11.7), Limitations (Section 11.8) and the open challenge

The control question A table-tennis ball crosses the table in about half a second, and every camera frame sharpens the prediction of where it will cross the hitting plane while leaving the arm less time to get there. When should the arm stop watching and commit, and what must its flight model, stroke and trajectory get right for the swing to arrive in time?

Learning objectives

- Compute the drag factor of the ball and explain why a predictor without drag commits close in place but wrong in time.
- Explain why the commit time has an optimum, and locate it from the returns and the time left in a commit sweep.
- Derive the paddle's normal speed from the impact law, and the rolling exit of a gripping bounce.
- Compare stepped, trapezoidal and minimum-jerk swings by their peak acceleration and by the lag of a drive tuned away from the shaping.
- Decide when a camera's rate and latency make an interception infeasible for any commit rule.

Prerequisites Sections 1.4.2 (arm kinematics), 1.4.4 (quintic and bang-bang profiles), 1.4.6 (the Kalman filter and the age of a report), 1.3 (loop action) and 1.6.4 (scenarios and the scorecard).

The app https://apmonitor.com/apps/table-tennis-robot/

See also Volume 5, Chapter 13 for an extended Kalman filter on late fixes, Volume 5, Chapter 12 for a state that is never measured and Volume 5, Chapter 3 for a measurement too slow for the loop. Volume 8, Chapter 15 uses the same six-axis arm to keep a ball bouncing.

11.1 Introduction

A ball served at 10 m/s and 3° from 0.30 m behind the far end of a table-tennis table reaches a hitting plane 0.30 m behind the near end line 0.504 s later in the simulator of this chapter, after a bounce 0.62 m from that end line. At that speed the air slows the ball by 12.6 m/s^2, more than gravity pulls it down, and a predictor that leaves the air out expects the ball after 0.373 s and puts the bounce at 0.28 m. A robot at the near end has to choose a hitting point inside that half second from camera frames that arrive late, and then bring a paddle there at the right speed within the limits of six joint drives. A 2014 promotional duel between a table-tennis professional and a six-axis industrial arm brought this problem to a general audience, and the problem is older and wider than the game.

DID YOU KNOW? Nine Meters a Second, and Faster

An amateur hits a table-tennis ball at up to 9 m/s, and a professional at three times that. A ball at amateur speed crosses the table in 0.4 s. It cannot carry a marker or paint, because either changes its weight or texture, so robots track it with passive cameras [1].

Two beliefs about control fail at this table. The first is that a controller should act once it knows. Every camera frame improves the prediction, and every frame spent watching is time the arm no longer has for its swing. Over twenty random serves to the full pipeline, a commit 50 ms after the serve returns 16 balls onto the table, a commit 200 to 275 ms after it returns 19, and a commit at 350 ms returns 10, with 243 ms left for a swing that needs a quarter of a second or more. The second belief concerns opponents. The tuned arm on the far side was designed with a declared blind spot, a commit fixed 150 ms after the student's strike and a spin it assumes rather than estimates, in the expectation that spin and wide placement would beat it. Measured over ten games against the suggested settings, its spin assumption decides none of them, and the points it loses come from its clock-frozen plan and from the ends of its reach.

The control problem belongs to a class, interception under a sensing delay with a decision deadline. A robot that catches a thrown ball predicts a free flight and closes its hand where the ball will be [2], and a robot table-tennis player chooses where and when to strike from a predicted trajectory before

it plans the stroke [3, 4]. In each case the estimate improves with time while the time left to act shrinks, so the moment of commitment is a design variable with an optimum, and the actuator's speed and acceleration limits set how much time the action needs. The picking arm of Chapter 10, which meets a product on a moving belt from camera reports that arrive late, a flying shear that must meet a moving strip and a batch cut at a predicted end point have the same shape. The flight itself is the other half of the problem, because a light ball's drag and Magnus lift move its bounce by decimeters [5, 6], drag alone by 34 cm for the serve above, and its spin changes how it leaves the table [7, 8].

The published robot players are each built around one platform, and how much the flight model, the commit time and the joint limits contribute is seen, when it is seen, inside a complete system [3, 4, 9–11]. A student who watches a robot miss cannot tell whether the prediction, the timing or the arm was at fault. This chapter takes those causes apart on a simulated table with a six-axis arm at each end, switching one on at a time, and counts the returns each one costs. The flight model, the gripping bounce, the extended Kalman filter [12], damped least-squares inverse kinematics [13, 14] and minimum-jerk trajectories [15] are standard methods. The measurements are of one model at one game seed. Among them are a commit-time optimum 319 to 390 ms before the hit, an estimator with a spin state that reads 33.5 rev/s of a 40 rev/s backspin before the bounce while every predictor without one misses by 31 to 37 cm, and the finding that the tuned opponent's declared blind spot is not where it loses. The simulator is a 33-state model with every impact located inside its integration step, the shared six-axis arm of the series and a measurement boundary that keeps both arms' controllers on their own camera frames, and the chapter states it completely enough to build again. Each of the nine cases and the game has a completion rule, and no setting of the pipeline wins case 9.

Section 11.2 follows robot table tennis from its first players. Section 11.3 gives the table, the ball, the two arms and the cameras. Section 11.4 spends the half second twice, on knowing where the ball will be and on moving there, and Section 11.5 describes the controllers as coded. Section 11.6 reports the nine cases and the game as measured, Sections 11.8 and 11.9 state the limitations and the conclusions, Section 11.7 names the devices of real robot players, and Section 11.10 turns the same cases into exercises.

11.2 Robots at the Table

Table tennis became a robotics benchmark because it packs perception, prediction and motion into a fraction of a second on a table of fixed size. A robot ping-pong player documented in a 1988 monograph framed the game as an experiment in real-time intelligent control, in which vision, the prediction of the ball and the planning of each return had to be finished in the time the ball takes to cross the table [3]. A low-cost player built around a personal computer was reported in 2003 [9], and a learning approach to robotic table tennis appeared in 2005 [10]. Striking movements that a robot learns to select and generalize were reported in 2013 [4], and in 2022 a muscular robot learned to play from scratch [11]. Across these decades the decomposition stayed recognizable, in which the ball is predicted, a hitting point and time are chosen, and the stroke is planned and executed.

> ⧖ HISTORY **From a Challenge of 1983 to Learned Play**
>
> Early work on robot table tennis is traced to a challenge of 1983: a competition whose simplified rules allowed for the technology of the time [16]. For most of the years since, robot players have predicted the ball from a dynamics model, either fitted or learned, chosen a virtual hitting point for the paddle, and then generated the joint trajectories to reach it. A newer line of research drops the hitting point and the ball model and controls the joints at a high rate with policies learned by trial or from demonstrations [1]. A player built that way, trained in simulation and moved to the real arm without further training, has since played full competitive matches against people it had never met [16].

The physics of the ball was worked out alongside. A review of sports-ball aerodynamics described how drag and the lift of a spinning ball depend on the flow around a sphere [5]. Fitted to recorded

Table 11.1 Signals by Role. Ranges Are Those the Interface Offers

Symbol	Role	Default	Range
$\mathbf{p}$, $\mathbf{v}$	states, ball position, velocity	per serve	—
ω	state, ball spin, constant in flight	per serve	—
q_i, $\dot{q}_i$	states, six joints and rates per arm	ready pose	ranges
$\hat{\mathbf{p}}_h$, t_h	CV, predicted hitting point and time	—	—
$q_{c,i}$	MV, joint commands	per plan	ranges
y_p, z_p	MV in Manual, paddle across, height	0, 20 cm	± 70, -5 to 50 cm
$\mathbf{z}_k$	measured, camera frame of $\mathbf{p}$	200 Hz	50 to 200 Hz
σ	estimate, $\hat{\mathbf{p}}_h$ uncertainty	—	—
$\mathbf{w}$	DV, gust across the hall	0	$\pm 3\,\mathrm{m/s}$
—	DV, worn ball, change in drag	0 %	± 30 %
—	DV, base drive derated to	100 %	40 to 100 %
—	DV, hidden-spin serve, 45 rev/s	—	event
—	DV, 50 ms camera dropout, net cord	—	events
—	outputs, score, returns, rallies	0	—

flights of the table-tennis ball, a drag coefficient of 0.54 ± 0.074 and a lift coefficient of 0.069 ± 0.029 were identified [6], and a rebound model with friction fitted to the same group's measurements gave a table a normal restitution of 0.93 and a sliding friction coefficient of 0.25 [8]. Measurements of bouncing balls show the contact gripping and slipping in turn, which a rigid grip law idealizes [7]. The laws of table tennis fix the ball at 40 mm and 2.7 g and ask a table to return a ball dropped from 30 cm to about 23 cm (Law 2.1.3).

The arm draws on manipulator kinematics and trajectory planning [17], which Sections 1.4.2 and 1.4.4 summarize. Damped least squares made inverse kinematics robust near singular configurations in 1986 [13, 14], a minimum-jerk profile was proposed in 1985 as a model of smooth reaching [15], and the Kalman filter of 1960 [12], in its extended form, is the usual estimator for a flight whose model is nonlinear. A robot catching thrown balls showed the same predict-and-intercept structure outside the game [2]. The simulator combines these pieces in a form small enough to teach with, and each one can be switched on or off.

11.3 The Table and Two Arms as Modeled

The world frame has x along the table from the near end line, where the student's arm plays, to the far end line at $x = L = 2.74\,\mathrm{m}$, y across it, positive to the student's left, with the center line at $y = 0$ and the side lines at $\pm 0.7625\,\mathrm{m}$, and z up from the playing surface, as shown in Figure 11.1. The net stands over $x = 1.37\,\mathrm{m}$, 0.1525 m high, its posts 0.1525 m outside the side lines. Each arm stands on a fixed base 0.85 m behind its end line, 0.05 m below the playing surface, and strikes on a hitting plane $x = x_h$, 0.30 m behind its end line. Two cameras watch the ball for each arm. The variables appear with their roles in Table 11.1. The constants of the ball, the arms and the cameras and every threshold were chosen for teaching, as the model dialog and the Instructions page state.

11.3.1 Flight

The ball is a sphere of radius $r = 20\,\mathrm{mm}$ and mass $m = 2.7\,\mathrm{g}$. Its position, velocity and spin obey Equation (11.1).

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v}, \qquad \dot{\omega} = \mathbf{0} \\ \dot{\mathbf{v}} &= -g\,\mathbf{e}_z - k_D\,|\mathbf{v}-\mathbf{w}|\,(\mathbf{v}-\mathbf{w}) + k_M\,\omega \times (\mathbf{v}-\mathbf{w}) \\ k_D &= \frac{\rho\, C_D\, \pi r^2}{2m} \end{aligned} \tag{11.1}$$

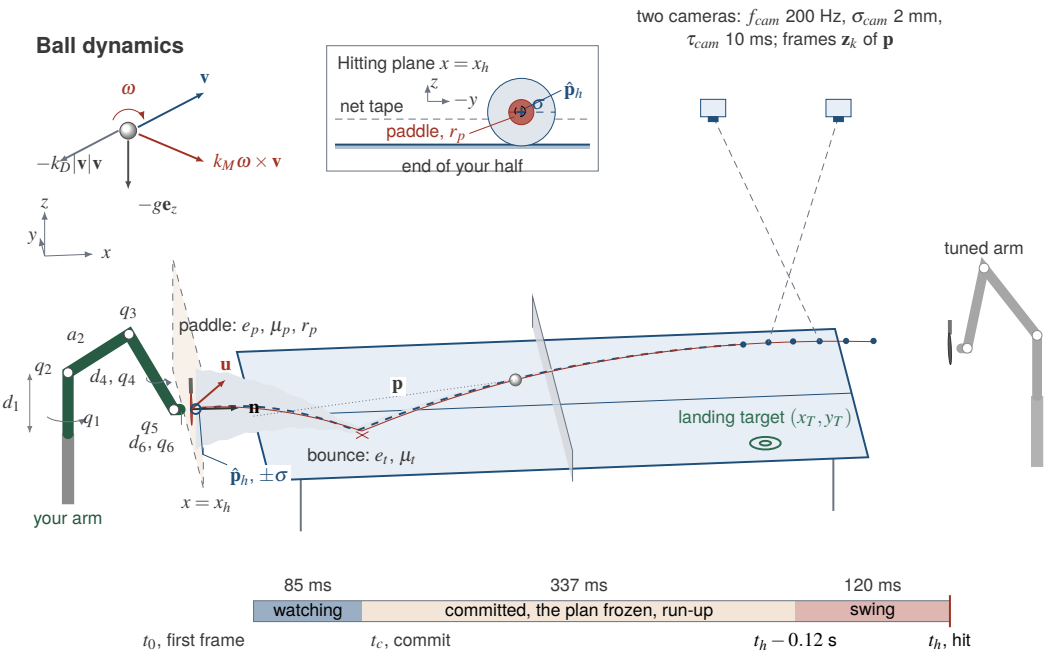


Figure 11.1 The table, the two six-axis arms and the student's pipeline at the commit on the first serve of the challenge with the suggested settings, drawn in the Table view's orthographic projection from the engine's state. The ball at $\mathbf{p}$ with velocity $\mathbf{v}$ and spin $\boldsymbol{\omega}$ feels gravity $-g\,\mathbf{e}_z$, drag $-k_D|\mathbf{v}|\mathbf{v}$ and Magnus lift $k_M\,\boldsymbol{\omega}\times\mathbf{v}$, and bounces with restitution e_t and grip μ_t. The cameras deliver frames $\mathbf{z}_k$ (dots) at f_{cam}, late by τ_{cam} with noise σ_{cam}. The dashed curve is the prediction at the commit and the shaded band its $\pm\sigma$, which reaches 202 mm at the predicted hitting point $\hat{\mathbf{p}}_h$ on the plane $x = x_h$, seen nearly edge on. The near arm, with joints q_1 to q_6 and lengths d_1, a_2, d_4, d_6, holds its paddle (restitution e_p, grip μ_p, radius r_p) at the planned hitting pose with normal $\mathbf{n}$ and velocity $\mathbf{u}$ that send the ball to the landing target. The upper-left inset separates the ball-force terms (schematic directions). The upper-middle inset draws the plane $x = x_h$ from behind the arm, with the paddle on $\hat{\mathbf{p}}_h$ inside its σ circle and the true crossing as a dashed ring. The bar is the Plan view's time budget for this ball, from the first frame t_0 to the commit t_c, the start of the swing $t_h - 0.12$ s and the hit t_h. Not to scale across the panels.

where $\mathbf{w} = [0,\ w_y,\ 0]^{\mathsf{T}}$ is a gust across the hall, $\rho = 1.2\,\mathrm{kg/m^3}$ and $C_D = 0.45$, which give $k_D = 0.1257\,\mathrm{m^{-1}}$, and $k_M = 4.5\times10^{-4}\,\mathrm{m/rad}$. A worn ball scales k_D by the disturbance slider, and neither arm knows. At 5 m/s, 10 and 15 the drag deceleration is 3.1, 12.6 and 28.3 $\mathrm{m/s^2}$, against 3.8, 15.1 and 33.9 $\mathrm{m/s^2}$ for the fitted $C_D = 0.54$ [6], and the terminal speed is 8.83 m/s. The fitted lift coefficient corresponds to a Magnus factor of about 1.0×10^{-3} m/rad, so the teaching value is less than half the measured one. Spin keeps its value through a flight.

11.3.2 Bounce and Strike

Both impacts follow one law, written relative to a surface of unit normal $\mathbf{n}$, toward the ball, moving at $\mathbf{u}$, as shown in Equation (11.2).

$$\begin{aligned} v_n &= (\mathbf{v}-\mathbf{u})\cdot\mathbf{n}, \qquad J_n = -(1+e)\,v_n \\ \mathbf{s} &= (\mathbf{v}-\mathbf{u}) - v_n\mathbf{n} - r\,\boldsymbol{\omega}\times\mathbf{n} \\ \mathbf{J}_t &= \begin{cases} -\frac{\alpha}{1+\alpha}\,\mathbf{s} & \frac{\alpha}{1+\alpha}|\mathbf{s}| \le \mu J_n \\ -\mu J_n\,\mathbf{s}/|\mathbf{s}| & \text{otherwise} \end{cases} \\ \mathbf{v}^+ &= \mathbf{v} + J_n\mathbf{n} + \mathbf{J}_t, \qquad \boldsymbol{\omega}^+ = \boldsymbol{\omega} - \frac{\mathbf{n}\times\mathbf{J}_t}{\alpha r} \end{aligned} \tag{11.2}$$

where $\mathbf{s}$ is the slip of the contact point, $\alpha = I/(mr^2) = 2/3$ for a thin hollow sphere, so the gripping impulse is $-\frac{2}{5}\mathbf{s}$, and the impulses are per unit mass. A gripping ball leaves rolling on the surface, and a ball whose grip would need more than μ times the normal impulse slides. The spin about the normal is kept, and angular momentum about the contact point is conserved either way, which one of the application's checks confirms to 10^{-12}. The table has $\mathbf{n} = \mathbf{e}_z$, $\mathbf{u} = \mathbf{0}$, $e_t = 0.905$ and $\mu_t = 1.0$. With this drag, $e_t = 0.905$ returns a ball dropped from 30 cm to the 23 cm the laws ask. The grip is a teaching value chosen so that bounces grip, since the 10 m/s serve at 3° needs $\mu_t \ge 0.76$ to grip, while on the measured 0.25 [8] it would slide and reach the hitting plane 68 ms sooner. The paddle has the normal and velocity of the arm's tool at the contact, $e_p = 0.7$ and $\mu_p = 1.2$ for the rubber, both teaching values, and it strikes only a ball whose center passes within $r_p + 0.3r$ of the paddle's center, $r_p = 75$ mm. The normal part of Equation (11.2) gives $v_n^+ = -e_p(v_n - u_n) + u_n$, so turning a 6 m/s ball into a 10 m/s one takes a paddle moving at 3.41 m/s along its normal, and a paddle brushing across the ball puts spin on it through $\mathbf{J}_t$.

11.3.3 The Six-Axis Arms

Each arm is the shared servo-level six-axis arm of the simulator framework, `shared/src/framework/core/arm.js`, the same arm that holds the paddle in Volume 8, Chapter 15. Its chain is an anthropomorphic arm with a spherical wrist on a base turned by ψ ($\psi = 0$ for the near arm and π for the far one), as shown in Equation (11.3).

$$\begin{aligned} R &= R_z(\psi+q_1)\,R_y(q_2+q_3)\,R_z(q_4)\,R_y(q_5)\,R_z(q_6) \\ \mathbf{p}_t &= \mathbf{b} + d_1\mathbf{e}_z + a_2\,R_z(\psi+q_1)R_y(q_2)\,\mathbf{e}_z \\ &\quad + d_4\,R_z(\psi+q_1)R_y(q_2+q_3)\,\mathbf{e}_z + d_6\,R\,\mathbf{e}_z \\ \mathbf{n} &= R\,\mathbf{e}_z \end{aligned} \tag{11.3}$$

where $\mathbf{b}$ is the base, $\mathbf{p}_t$ the paddle's center and $\mathbf{n}$ its normal, the flange's z axis, with $d_1 = 0.30$ m, $a_2 = 0.42$ m, $d_4 = 0.40$ m and $d_6 = 0.08$ m, a reach of 0.90 m. The paddle's center and normal make a five-number task, and the roll q_6 about the normal is left free. On its hitting plane at 20 cm above the table the arm reaches 0.68 m across, and 0.62 to 0.68 m over the heights and face tilts it plays.

Each joint is a drive at the servo level, with no inertia matrix, Coriolis terms or torque limits, as shown in Equation (11.4).

$$\begin{aligned} \ddot{q}_i &= \mathrm{sat}_{\ddot{q}_{max,i}}\left(\omega_j^2(q_{c,i}(t) - q_i) - 2\zeta_j\omega_j\dot{q}_i\right) \\ |\dot{q}_i| &\le \dot{q}_{max,i} \end{aligned} \tag{11.4}$$

where $\omega_j = 2\pi f_j$ and ζ_j are the drives' bandwidth and damping, 15 Hz and 1 by default, and the command $q_{c,i}(t)$ is the controller's latest value carried along its rate between 5 ms updates, as a drive interpolates its setpoint. A drive at its speed limit gets no further acceleration in that direction. The base, shoulder and elbow are limited to 360 °/s and 3000 °/s^2, and the three wrist axes to twice that, a

departure from a single limit for all six that the wrist needs to turn the paddle within a quarter-second swing. At a lever of 0.5 m the base limits correspond to 3.14 m/s and 26.2 m/s^2 at the paddle. A joint's range is a soft limit. Moving toward an end at $\dot{q}_i$ with d_{end} of range left, the drive brakes at $\min(1.25\,\dot{q}_i^2/(2d_{end}),\ \ddot{q}_{max,i})$ once $\dot{q}_i^2/(2d_{end})$ reaches $0.8\,\ddot{q}_{max,i}$, whatever the command, and the plant keeps a hard clamp at the range as a backstop. Over every case and both challenge settings no joint arrives at a range end faster than 6.9 °/s.

11.3.4 Cameras and Events

Each arm has its own cameras, a stereo pair modeled as one measurement of the ball's center, as shown in Equation (11.5).

$$\mathbf{z}_k = \mathbf{p}(t_k) + \sigma_{cam}\,\mathbf{n}_k, \qquad t_k = k/f_{cam} \tag{11.5}$$

where frame k is delivered at $t_k + \tau_{cam}$ and $\mathbf{n}_k$ holds three unit normal deviates drawn from that arm's own seeded stream, three per frame whatever the noise level, $f_{cam} = 200$ Hz, $\tau_{cam} = 10$ ms and $\sigma_{cam} = 2$ mm. The student's cameras are adjustable down to 50 Hz and up to 150 ms, while the tuned arm's stay at the defaults. Spin is never measured. A dropout loses every frame exposed while it lasts, and no frame is taken while the ball is not in play.

The table, the paddles and the net act between integration steps. When the ball's surface crosses the table or a paddle face inside a step, the ball and the paddle are carried linearly back to the contact instant, the law of Equation (11.2) is applied there, and the ball flies the rest of the step on the true model. A ball through the net's plane below the tape is dead, and one that passes within one radius over it clips the tape, keeps 45 % of its speed along the table and pops up. The referee scores a strike or a serve as a shot that must bounce once on the receiver's half. A second bounce there or a ball that passes the receiver is a missed return, and a first bounce on the striker's own half, a ball past the receiver's end line without a bounce or a ball off the side is out. A game goes to eleven points, won by two, with the serve changing every two points and every point from ten all. Serves are launched rather than struck, from 0.10 m behind the server's end line, 0.25 m up (the ball machine's from 0.30 m behind the far end line), aimed on the true model at a seeded point 0.30 to 0.75 m from the receiver's end line and within 0.40 m of the center line, at a seeded speed of mean 8.5 m/s and standard deviation 0.6 m/s, with up to 20 rev/s of topspin or backspin. The serve counts as the server's shot, and the rule that a serve bounces on both halves is left out.

11.3.5 Parameters and Numerics

Table 11.2 gives every parameter. The 33 states, the ball's nine and each arm's twelve, are advanced by fourth-order Runge-Kutta steps of 0.5 ms, in which a 10 m/s ball moves 5 mm. Both arms' controllers and cameras run every 5 ms and the historian logs at 100 Hz, on the engine of Section 1.6.1. Within a step the controller runs first, then the integration, then the speed and range clamps, the impacts, the referee and the serve clock. At 0.5 ms the integrated step overshoots a contact by at most 1.8 mm at the table and 2.9 mm at a paddle before the back-interpolation, halving at 0.25 ms, and the planning serve's bounce lands at the same 0.61954 m at 0.5, 0.25, 0.1 and 0.01 ms. The 66.3 s game of the suggested settings plays in about 13 s at five times real time. Every serve, camera frame and opponent target comes from seeded streams, so a case replays identically after Reset, from the list after another case, and when opened first.

11.3.6 Assumptions

A rigid ball with constant aerodynamics. The drag and Magnus factors do not change with speed or spin, and the spin does not decay in flight. The fitted factors are larger [6], so real air matters more than it does here.

Instantaneous impacts with a grip law. A real contact has compliance, grips and slips in turn [7], and loses restitution with speed, and a real paddle's rubber, sponge and blade bend. The table's grip

Table 11.2 Model Parameters. A Is Adjustable in the Model Dialog, H Hidden, C Fixed in the Code

Symbol	Meaning	Value	
Ball and table			
r, m	ball radius, mass	20 mm, 2.7 g	C
k_D	drag factor, C_D 0.45	0.1257 m^{-1}	A
k_M	Magnus factor	4.5×10^{-4} m/rad	A
e_t, μ_t	table restitution, grip	0.905, 1.0	A
e_p, μ_p	rubber restitution, grip	0.7, 1.2	A
r_p	paddle radius	75 mm	H
L, W, h_n	table length, width, net height	2.74, 1.525, 0.1525 m	C
Arms			
d_1, a_2, d_4, d_6	chain lengths	0.30, 0.42, 0.40, 0.08 m	C
b	base, behind the end line, below the top	0.85, 0.05 m	C
x_h	hitting plane behind the end line	0.30 m	C
$\dot{q}_{max}$	speed limit, axes 1 to 3 (wrist ×2)	360 °/s	A
$\ddot{q}_{max}$	acceleration limit, axes 1 to 3 (wrist ×2)	3000 °/s^2	A
—	ranges, axes 1 to 6 (±°)	170, 100, 150, 200, 125, 200	C
f_0, ζ_0	drive tuning the commands assume	15 Hz, 1	C
Cameras			
f_{cam}, τ_{cam}	frame rate, latency	200 Hz, 10 ms	A
σ_{cam}	noise per axis	2 mm	A
Tuned opponent and serves			
$t_{c,opp}$	fixed commit after the strike	150 ms	A
b_{spin}	spin assumed until its bounce	1 (yes)	A
y_{opp}	target width, 0.65 of it used	0.55 m	H
v_s, σ_v	serve speed, spread	8.5, 0.6 m/s	A
ω_s, y_s	largest serve spin, widest target	20 rev/s, 0.40 m	A
—	game seed	1	A
Numerics			
Δt, T_c	integration step, control period	0.5, 5 ms	C
—	historian	100 Hz	C

is four times the measured value, which keeps a gripping bounce for spinning balls but makes fast, low bounces grip that would slide.

Servo-level arms. The kinematics are exact, but there is no inertia, torque limit or flexibility, and a real arm's limits change with its pose and payload. The paddle's strike uses the velocity of its center and leaves out its rotation at the contact.

Ideal cameras apart from noise, rate and latency. Calibration error, motion blur and occlusion are absent, and every frame of a ball in play is taken.

Fixed bases and a simplified game. Neither arm moves its base, so a ball short enough to bounce twice before the hitting plane is out of play. Serves are launched by a machine at the server's paddle, and lets and edge balls are left out.

11.4 Half a Second, Spent Twice

11.4.1 Where the Ball Will Be

The flight model decides the prediction more than the frames do. Launched at 10 m/s and 3° from 0.30 m behind the far end line with no spin and the net removed, the ball bounces 0.62 m from the near end line after 0.283 s and reaches the hitting plane at 0.504 s. Without drag it would bounce at 0.28 m and arrive at 0.373 s, a difference of 34 cm along the table and 131 ms in time. Spin moves the bounce as well. With 50 rev/s of backspin the lift holds the ball up and the bounce moves to 0.43 m, and with 50 rev/s of topspin it moves to 0.78 m. Through the grip of the bounce, 40 rev/s of topspin brings the ball to the plane at 0.439 s and 40 rev/s of backspin at 0.532 s, 93 ms apart.

A fit or a filter anchored to the frames can carry the ball forward only with the forces its model has, as shown in Figure 11.2a. On the planning serve, after the 30 frames of the first 150 ms, the median distance from the predicted hitting point to the ball at the predicted instant is 296 to 329 mm without drag and 10.5 mm to 10.9 mm with it. The no-drag error is mostly timing. Its predicted point lies

4.4 to 4.7 cm from the true crossing, but it expects the ball 73 to 81 ms early, and the paddle swings through an empty plane. Adding a spin state to a spinless ball costs accuracy at first, because the extended Kalman filter (EKF) must also explain the frames with a spin it does not yet know, and its miss after 30 frames is 40 mm.

Spin reverses the ranking, as shown in Figure 11.2b. On case 7's serve, 8.5 m/s with 40 rev/s of backspin, the drag-only fit and EKF stay 30 to 32 cm off until the bounce, the no-drag models are 37 cm off by then, and only the EKF on the drag-and-Magnus model sees the lift bend the flight. At frame 68, where case 7 commits, 14 ms before the bounce, it misses by 22 mm. The fit restarts its window after the bounce and catches up, to 23 mm by frame 80, but by then the plan is frozen.

> **TRY IT IN THE APP** **7 · The Hidden Spin**
>
> Run the case as loaded, with the Fit estimator. Every serve carries 40 rev/s of backspin, and the fit returns none of the ten. Press Reset, set Estimator to EKF and run again. The filter reads the spin from the bend of the flight before the bounce, and all ten come back.

11.4.2 When to Stop Watching

Information arrives at 200 Hz and the deadline does not move. An arm whose swing needs a quarter of a second or more inside its joint limits must commit at least that long before the hit, and waiting buys frames that narrow the prediction it commits to. Case 3 measures the trade over 20 random serves with the full pipeline, as shown in Figure 11.2c. Committing 50 to 150 ms after the serve, 539 to 444 ms before the hit, returns 16 or 17 and leaves the hit time uncertain by 31 to 39 ms rms. Committing 200 to 275 ms after it, 390 to 319 ms before the hit, returns 19. From 350 ms, with 243 ms left, the returns fall to 10 to 12 although the hit time is known to about 11 to 23 ms rms, because the swing no longer fits. The in-plane error at the commit falls only from 38 to 14 mm across the sweep, so the cost of committing early is mostly in the timing.

Pause. Try Exercise 11.2 before reading on.

11.4.3 What the Arm Can Do in the Time Left

The stroke must be delivered at an instant, with a position, an orientation and a velocity, by drives with limits. A minimum-jerk move of 0.3 m in 0.25 s needs 27.7 m/s^2 at its peak, 1.44 times the 19.2 m/s^2 of a bang-bang move (Equation (1.5)), and on the arm it asks 68 % of a speed limit and 100 % of the base's acceleration limit and ends 0.1 mm from its pose. The same move with the joint commands stepped to the pose ends 150 mm off. A move of 0.5 m in 0.25 s needs 32 m/s^2 and 4.0 m/s even on a bang-bang profile, and on the arm its minimum-jerk version asks 144 % of an acceleration limit and ends 116 mm off. The forward stroke is limited by the elbow to a paddle speed of about 2 m/s, so a drive planned for 7.5 m/s leaves at 6.35 m/s.

11.5 The Pipeline as Coded

Every 5 ms each arm takes the frame due, updates its tracker, predicts where the ball will cross its hitting plane, re-plans the stroke and the arm's motion until it commits, and writes six joint commands. The controllers read only their own camera frames and their own joint angles, with three declared exceptions. The referee's dead-ball signal ends a track, the wide placement aims away from the far paddle's true position, and the tuned arm takes its spin assumption after its own serve from the serve it launched. The truth, where the ball really crossed, is recorded for the display and the score and logged in columns prefixed `truth_`. No loop in this application is a PID loop. The velocity-form routine of Section 1.6.2 plays no part here, nothing integrates an error, so there is no windup to guard against, and the only limits are the drives' own.

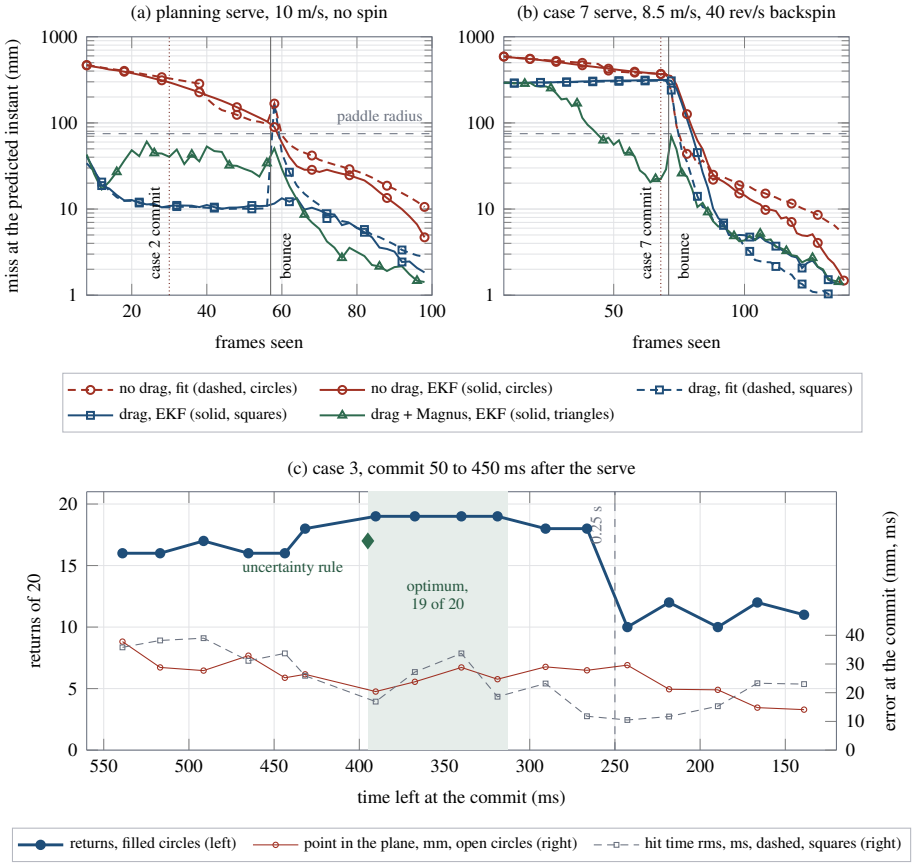


Figure 11.2 Prediction and the commit, from the application's own predictor and engine. (a) and (b) The distance from the predicted hitting point to the true ball at the predicted instant against the camera frames seen, the median of 20 noise seeds, for the least-squares fit (dashed) and the EKF (solid) on the no-drag (circles), drag (squares) and drag-and-Magnus (triangles) models. The fit has no spin state, so the fit on the drag-and-Magnus model equals the fit on drag and is not drawn. The horizontal line is the paddle's radius. (c) Case 3 over 20 random serves with the commit fixed 50 to 450 ms after the serve in 25 ms steps, with the returns that land (left axis, filled circles) and, at the commit, the in-plane error of the committed point (open circles) and the rms error of the hit time (dashed, squares) (right axis), against the time left before the hit. The band marks the optimum, 19 of 20, and the diamond the uncertainty rule.

11.5.1 Tracking and Prediction

A flight starts when a frame lands more than 3 cm from where the track expects it, at a strike or a serve, and the next frame initializes position and velocity from the two. The fit then solves for the position and velocity at the latest frame by one Gauss-Newton pass per frame over at most 80 frames since the flight began or since the bounce its model predicts, with the model's own curvature carried back from the estimate, and it has no spin. The EKF [12], the nonlinear form of the variance-chosen filter of Section 1.4.6, carries $\hat{\mathbf{x}} = [\hat{\mathbf{p}},\ \hat{\mathbf{v}},\ \hat{\omega}]$ on the chosen model of Equation (11.1) with $\mathbf{w} = \mathbf{0}$, its

own bounce, and the update of Equation (11.6).

$$\begin{aligned} P^- &= FPF^\mathsf{T} + Q, \qquad F = I + A\Delta + \tfrac{1}{2}(A\Delta)^2_{\mathbf{p}} \\ K &= P^- H^\mathsf{T}(HP^- H^\mathsf{T} + \sigma_{cam}^2 I)^{-1}, \qquad H = [I \;\; 0 \;\; 0] \\ \hat{\mathbf{x}} &= \hat{\mathbf{x}}^- + K(\mathbf{z}_k - \hat{\mathbf{p}}^-), \qquad P = P^- - KHP^- \end{aligned} \tag{11.6}$$

where A is the flight's Jacobian, the squared term is kept only in the position rows, the propagation runs in steps of at most 5 ms, and Q adds $q_v\Delta$ to each velocity variance, $q_v = 0.01\,\mathrm{m}^2/\mathrm{s}^3$, and $100\,\mathrm{rad}^2/\mathrm{s}^3$ times Δ to each spin variance. When the model estimates spin, it starts at zero with a standard deviation of 120 rad/s and enters the estimate only through the Magnus curvature and the bounce, and otherwise it is held at zero. At a bounce the covariance passes through the bounce map's Jacobian, taken by finite differences at the contact state, and each velocity variance gains $(0.08\,\mathrm{m/s})^2$ for the bounce model's own error. The prediction rolls the estimate forward in steps of 4 ms, carrying the covariance every 20 ms, through the bounce on the arm's half, to the hitting plane x_h, or to planes 0.10 and 0.20 m nearer the table when the ball would pass x_h less than 3 cm above the surface. The uncertainty σ is the larger of the across and height standard deviations at the plane.

11.5.2 The Stroke, Inverted From the Landing

The stroke is a boundary-value problem. A landing target on the far half, the middle or 0.55 m off it away from the far paddle, fixes the outgoing velocity, found by Newton's method on the elevation and heading of rollouts of the arm's own flight model to 5 mm, with the ball passing at least 5 cm over the tape. The normal and velocity of the paddle then follow from Equation (11.2), as shown in Equation (11.7).

$$\begin{aligned} u_n &= \frac{v_n^{out} + e_p\, v_n^{in}}{1 + e_p}, \qquad \mathbf{u} = u_n\mathbf{n} + b\,\hat{\mathbf{t}} \\ \mathbf{n} &\parallel \mathbf{v}^{out} - \mathbf{v}^{in} - \mathbf{J}_t \end{aligned} \tag{11.7}$$

where $\hat{\mathbf{t}}$ is the face's upward direction and b the brush, iterated until the normal moves less than 10^{-7}, with the outgoing spin fed back into the aim over two or three passes. A target out of range raises the speed by 12 % at most twice, and a ball that would not clear the tape lowers it by 12 % at most four times. The push leaves at 6.0 m/s aimed 0.55 m from the far end line, the drive at 7.5 m/s aimed 0.40 m from it and the spin stroke at 7.0 m/s with $b = 2.4\,\mathrm{m/s}$, aimed 0.45 m from it. The paddle's center goes to $\hat{\mathbf{p}}_h - r\mathbf{n}$.

11.5.3 Inverse Kinematics and the Three Trajectories

Damped least squares [13, 14] solves the five-number task, as shown in Equation (11.8).

$$\Delta\mathbf{q} = J_t^\mathsf{T}\left(J_t J_t^\mathsf{T} + \lambda^2 I\right)^{-1} \begin{bmatrix} \mathbf{p}^* - \mathbf{p}_t \\ L(\mathbf{n} \times \mathbf{n}^*) \end{bmatrix} \tag{11.8}$$

where J_t is the geometric Jacobian with its angular rows projected across the normal and scaled by $L = 0.3\,\mathrm{m/rad}$, $\lambda = 0.02\,\mathrm{m}$, the step is capped at 0.5 rad and the joints are clamped into range after each one. Warm-started every 5 ms it takes about one iteration and 1 µs, and a pose counts as reached within 1 cm and 0.05 rad. The joint rates at the hit come from a regularized least squares that gives the paddle $\mathbf{u}$ with its normal held. The *PD step* sends the hitting pose to the drives unshaped. The *trapezoid* waits at the ready pose, then moves the paddle on a straight line at $25\,\mathrm{m/s}^2$ and arrives at rest. *Min-jerk* plans a quintic per joint, Equation (11.9), first to a run-up pose 0.12 m behind the hitting pose reached 0.22 s before the hit, then at the commit from the present reference to the hitting pose and its joint rates, then a 0.16 s follow-through and a 0.6 s recovery.

$$q_i(\tau) = \sum_{k=0}^{5} c_{ik}\tau^k, \qquad (q_i, \dot{q}_i, \ddot{q}_i)\big|_{\tau=T} = (q_{h,i},\ \dot{q}_{h,i},\ 0) \tag{11.9}$$

where the six coefficients also match the reference's position, rate and acceleration at $\tau = 0$, and T is the time left to the hit. A swing asking more than a joint limit at any of 25 samples is slowed, the stroke's speed scaled by 0.85 up to seven times. The reference r reaches the drives through the inverse of the servo tuned as commissioned, as shown in Equation (11.10).

$$q_c = r + \frac{2\zeta_0}{\omega_0}\dot{r} + \frac{\ddot{r}}{\omega_0^2}, \qquad \omega_0 = 2\pi \cdot 15\,\text{Hz}, \quad \zeta_0 = 1 \tag{11.10}$$

so a drive tuned as assumed follows the reference exactly in its linear range, and a drive detuned from it lags the plan. Each drive of Equation (11.4) is reverse acting in the convention of Section 1.3 ($a = +1$), because its acceleration rises when the angle falls short of its command. A ball predicted 30 cm to the student's left turns the base joint left, in the positive direction.

11.5.4 Commit Rules

Until the commit the plan is re-made every 5 ms from the latest prediction, once at least eight frames have arrived. *Time* commits a fixed time after the flight's first frame, *Frames* after a number of frames and *Uncertainty* when $\sigma \le \sigma_c$, 10 mm by default, or when the time left falls to $\max(0.12\,\text{s},\ T_{sw})$. For the minimum-jerk trajectory T_{sw} is the shortest swing, found by bisection between 0.06 and 0.5 s, whose quintic stays within 95 % of every limit, timed at the speed the stroke will get. For the drive and the spin stroke a long swing winds back before it goes forward, so the bound often sits at its 0.5 s cap. At the commit the plan freezes and the swing is planned in the time left, so a late commit leaves too little time and the swing is slowed.

11.5.5 Manual Mode and the Opponents

In Manual the paddle stays 0.45 m behind the end line, facing the table, where the two sliders or the arrow keys put it, through the same inverse kinematics, and on a phone a tap on the table moves it. The steady server returns every ball it reaches with a push within 0.15 m of the middle of the student's half, 0.40 to 0.60 m from the end line. The tuned arm has the student's best pipeline, the drag-and-Magnus EKF, minimum-jerk swings and 15 Hz drives, plays a drive stroke on every return at a seeded target 0.25 to 0.55 m from the student's end line and within 0.36 m of the center line, and returns to the center between strokes. Its blind spot, written in the model dialog, has two parts. It commits a fixed 150 ms after the student's strike whatever its uncertainty, and until the ball has bounced on its half it holds the spin that a flat return of its own shot would carry, 40 % of the incoming spin, without estimating it.

11.6 Nine Cases and a Game, Measured

Table 11.3 lists the cases and Table 11.4 what they returned. The numbers below come from the application's engine stepped outside the browser exactly as the page steps it, at game seed 1, and any setting a sentence changes was set before the run began. A return counts when the ball lands on the far half. The application passes its 36 verification checks. Figure 11.3 draws the returns of Table 11.4 as bars.

11.6.1 By Hand, and the Flight Model

Case 1 hands the paddle to two sliders against twelve machine serves at 7.0 to 8.0 m/s, which take 0.62 to 0.75 s to reach the hitting plane. Left where it starts, the paddle returns none and the first point is lost 1.85 s into the run, and the status line says how far, and to which side, the ball passed the paddle. Switched to Auto before Run, the arm returns all twelve. Case 2 serves the planning ball ten times to a no-drag fit that commits 150 ms after the serve. At the commit its hitting point is 3.6 cm from the true crossing, but it expects the ball 71 ms early, so each swing passes before the ball arrives and all ten returns go out. With drag the point is 0.9 cm off and the time 1 ms, and all ten land 0.6 cm from their targets. Drag and spin also returns all ten, 3.2 cm from the targets, because the fit has no spin state and only the stroke's aim allows for spin.

Table 11.3 The Nine Cases and the Challenge

#	Configuration	Teaching point
1	Manual, 12 machine serves near 7.5 m/s	a hand is too late
2	no-drag fit, commit 150 ms, 10 planning serves	the flight model
3	full pipeline, commit 50 ms, 20 random serves	the commit time
4	PD step, serves 0.30 m either side	the trajectory
5	full pipeline, drives at 3 Hz	the drive tuning
6	drag model, push, 12 planning serves	the stroke
7	fit, 10 serves with 40 rev/s backspin	a hidden state
8	push against the steady server	a rally in a loop
9	one 50 Hz camera, 100 ms late, 13 m/s drives	honestly infeasible
C	a game to eleven against the tuned arm	everything at once

Table 11.4 Outcomes Read From the Running Engine

Case and condition	Result
1, sliders untouched / Auto	0 of 12, first point lost at 1.85 s / 12 of 12
2, no drag / drag / drag + spin	0, 10, 10 of 10; at the commit 3.6, 0.9, 0.9 cm off, hit time 71, 1, 1 ms rms
3, commit 50 / 200 / 450 ms	16, 19, 11 of 20; 539, 390, 139 ms left
3, uncertainty rule	17 of 20; commits 191 ms in, 395 ms before the hit
4, PD step / trapezoid / min-jerk	0, 0, 10 of 10; 10, 6, 10 struck
5, drives 3 / 5 / 8 / 15 Hz	5, 9, 12, 10 of 12 land; 7 net at 3 Hz
6, push / drive / spin	12 of 12 each; 1.0, 0.5, 4.7 cm from target
7, fit / EKF / EKF on drag	0, 10, 0 of 10; EKF reads 33.5 rev/s of 40
8, push / drive, 120 s	40, 43 land; 21.1, 22.5 cm; longest rally 12, 14
9, slow camera, 144 designs	0 of 10 in every one
9, 200 Hz, 10 ms / 100 Hz, 50 ms	10 of 10 at 2.7 cm / 1 of 10
C, shipped / suggested	0–11 in 28.2 s / 11–9 in 66.3 s

Pause. Try Exercise 11.1 before reading on.

TRY IT IN THE APP **1 · By Hand**

Load the case, press Run and leave Paddle across and Paddle height where they start. Watch each ball pass the paddle: none of the twelve serves comes back, the first point is lost 1.85 s into the run, and the status line says how far, and to which side, the ball passed. Press Reset, switch to Auto and run again. The arm returns all twelve.

11.6.2 The Commit, the Trajectory and the Drives

Case 3 is the sweep of Figure 11.2c. The uncertainty rule returns 17 of 20, committing 191 ms after the serve on average and 395 ms before the hit, and all 20 of its commits come from the latest-swing bound, because σ never falls to 10 mm before the bounce. The spin about the line of flight leaves no trace in the frames until the bounce, so σ stays near 20 cm, and the rule is in practice a deadline. In case 4, with serves 0.30 m to either side, stepped commands strike all ten balls and return none, because the drives saturate, overshoot and arrive with no stroke speed. Moved 0.3 m across in 0.25 s on the arm alone, stepped commands hold the base, elbow and forearm-roll drives on their acceleration limits for 240, 239 and 243 ms of the 250 ms. The trapezoid strikes six, arrives at rest and returns none, while the minimum-jerk swing returns all ten, its drives at a limit for 7 % of the committed time against 52 % for the step. Case 5 detunes the drives to 3 Hz while the commands are still shaped for 15 Hz. The drives lag by about $2\zeta/\omega = 0.106$ s, every ball is struck late with the face tilted, seven of twelve go into the net and five land 54.5 cm from their targets on average. At 5, 8 and 15 Hz, 9, 12 and 10 land.

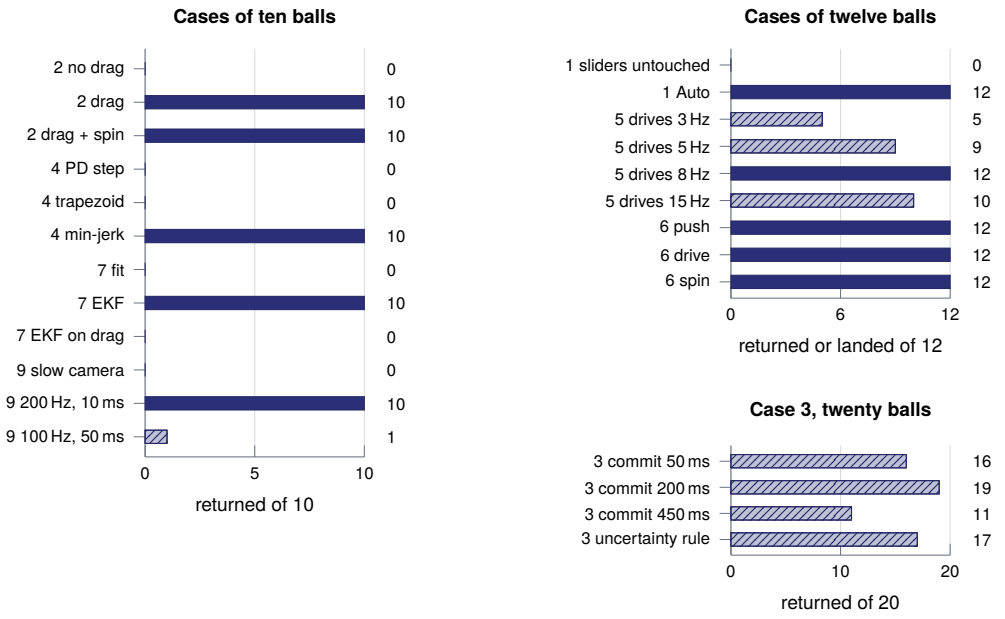


Figure 11.3 The returns of Table 11.4 drawn as bars, by case and setting, in three panels by the number of balls each case plays. Solid bars return every ball, hatched bars some of them, and a setting with no bar returns none; case 5 counts the balls that land. Notice that each case turns on one part of the pipeline, the flight model in case 2, the trajectory in case 4 and the spin state in case 7, and that in case 3 neither the earliest nor the latest commit returns the most.

> ✦ **DID YOU KNOW? Knowing More, Returning Less**
>
> From the simulator: in case 3, a commit 200 ms after the serve returns 19 of 20 balls with 390 ms left before the hit. Waiting until 450 ms sharpens the prediction but returns only 11, because 139 ms is too little time for the swing (Table 11.4).

11.6.3 The Stroke and a Hidden Spin

Case 6 returns the planning serve with the drag model. The push leaves at 5.79 m/s and lands 1.0 cm from its target on average, the drive at 6.35 m/s and 0.5 cm, and the drive placed wide 1.0 cm, so the planner's model is the flight's. The ball arrives with 35.1 rev/s of topspin picked up rolling off the table, and both flat strokes send it back with 14 to 15 rev/s of backspin. The spin stroke moves the paddle at 2.87 m/s, brushing up across the ball, leaves 1.7 rev/s of backspin and lands 4.7 cm off, because the drag model leaves out the lift of the spin it puts on. Case 7 serves 40 rev/s of backspin at 8.5 m/s. The fit returns none of ten and the EKF on the drag model none, while the EKF on the drag-and-Magnus model returns all ten, having read 33.5 rev/s of backspin from the curvature of the flight at the first serve's commit, 356 ms before the hit. The bounce makes the backspin visible to every model, because the contact slides and the ball leaves the table with 2.18 m/s of horizontal speed instead of the 3.66 m/s of the same serve without spin, but it comes too late.

🕮 **Pause.** Try Exercise 11.3 before reading on.

11.6.4 A Rally, and a Camera Too Slow

Case 8 plays the steady server for 120 s. With the push 40 of the student's returns land, 21.1 cm from their targets on average, and the longest rally is 12 strikes. The server fails to return 18 balls and the student 3, and 10 of the server's failures are pushes that bounce twice before its hitting plane,

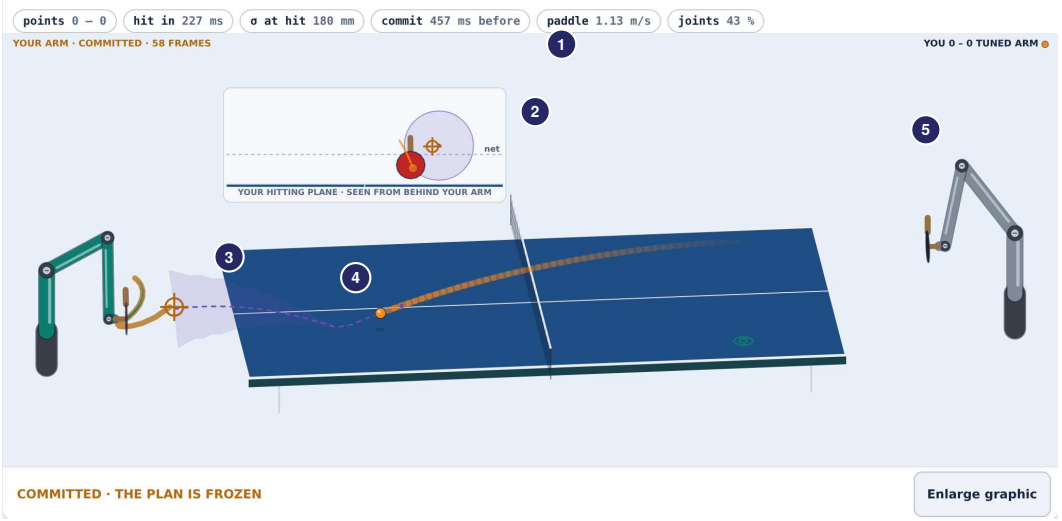


Figure 11.4 The student's arm committed on the tuned arm's first serve, 1.20 s into the challenge played with the suggested settings, on the Table view, cropped to the plant view. The tuned arm (5) served at 0.90 s, 8.50 m/s with 17.1 rev/s of topspin, and the ball (4) is about to bounce. The student's arm committed at 0.985 s, 85 ms after the first frame and 457 ms before the planned hit, and has 58 frames. The badges (1) read the score 0–0, the hit in 227 ms, σ at the hit 180 mm, the commit 457 ms before, the paddle at 1.13 m/s and the joints at 43 % of their limits. The dashed predicted path (3) widens inside its band toward the hitting-point marker (amber in the app), and the panel (2) above the near half draws the hitting plane from behind the arm with the 180 mm circle around the prediction.

which an arm on a fixed base cannot reach. With the drive 43 land, 22.5 cm from their targets, and the server fails 10 times. Over ten game seeds the push's longest rally in 150 s runs from 8 to 25 strikes, so a rally's length is a matter of chance and the case is scored on landings. Case 9 gives the arm one camera at 50 Hz with 100 ms of latency against 13 m/s drives. The drive reaches the plane 351 ms after it leaves, the eighth frame is exposed 140 ms into the flight and delivered 100 ms later, so the first plan comes 110 ms before the ball arrives. Every one of 144 designs, 16 commit settings by three strokes by three trajectories, returns none of ten. The 200 Hz, 10 ms camera returns all ten, 200 Hz with 50 ms of latency all ten at 17.0 cm, 100 Hz with 50 ms one, and 50 Hz with 20 ms none.

11.6.5 The Game, and Where the Tuned Arm Loses

The challenge is a game to eleven, won by two, against the tuned arm at the default game seed, with the model and the opponent as given, and the score is the points. The configurations are compared in Table 11.5. As shipped, with the no-drag fit, a commit 100 ms after the opponent's strike, the PD step and the push, the student loses 0–11 in 28.2 s and wins none of ten seeds, and the status line names the first loss as the ball 71 ms later than predicted with a joint at its limit. The drag model and the minimum-jerk swing alone win all ten seeds. The suggested settings, the drag-and-Magnus EKF, the uncertainty commit, minimum-jerk swings and the spin stroke placed wide, win 11–9 in 66.3 s and all ten seeds, and 20 of their 21 commits come from the latest-swing bound, 316 to 499 ms before the hit. The state of that game 1.20 s in is shown in Figure 11.4. The status line names the cause of every point the student's arm loses, a prediction off and by how much, a ball late or early, a late paddle, a drive at its limit or a pose out of reach. Play ten seeds runs the student's settings over game seeds 1 to 10 in the background and reports the win rate beside the measured benchmark of the shipped

Table 11.5 The Challenge Against the Tuned Arm, Game Seeds 1 to 10

Configuration	Seed 1	Won
shipped (no-drag fit, commit 100 ms, PD step, push)	0–11	0 of 10
shipped with drag and min-jerk	11–5	10 of 10
suggested (drag + Magnus EKF, uncertainty, min-jerk, spin wide)	11–9	10 of 10
suggested, placed in the middle	11–3	9 of 10
suggested, commit fixed at 150 ms	4–11	5 of 10
suggested, opponent estimates spin all the way	11–5	10 of 10
suggested, opponent commits at 200 ms / 300 ms	11–7 / 11–7	10 / 9 of 10

and suggested settings, and the challenge stops crediting a game as soon as any model parameter or disturbance moves from its published value.

The tuned arm was designed to be beaten at its declared blind spot, and the expectation was that a heavy spin stroke or a wide placement would do it. Measured, the two parts of the blind spot are not equal. With its spin assumption removed in the model dialog, the suggested settings still win all ten seeds, so the assumed spin decides nothing. Of the 110 points the tuned arm loses in the suggested settings' ten games, 52 come from its plan, frozen 150 ms after the student's strike, putting its paddle more than a paddle's radius from where the ball crossed, 34 from wide balls whose hitting pose sits at the end of its reach or a joint range, 13 from its own returns going out, 6 from swings beyond its joint limits, 3 from balls that bounce twice before its plane, and one each from a missing commit and a joint at its range end. Against pushes to the middle, the shipped settings with drag and minimum-jerk swings, 40 of the 113 points it loses are balls that bounce twice before its hitting plane. Moving its commit to 200 ms or 300 ms still loses 10 and 9 seeds of ten, so the fixed clock is not the only weakness. The arm that loses is weak where its geometry is, a reach that ends about 0.65 m off the middle and a hitting plane behind its end line, as much as where its model is. The gap between the design intent and the measurement is itself the finding. A good controller is beaten where it actually fails, which a match statistic finds and a design note only suggests. The student's own commit matters as much. The suggested pipeline made to commit at a fixed 150 ms wins only five of ten, and of the 47 points the student's arm loses in the ten suggested games, 15 are committed points more than 7.5 cm off and 10 are short balls.

The game's secondary counts depend on the step. At 0.25 ms the shipped settings still lose 0–11 and the suggested settings win, 11–5 instead of 11–9, with the same 0 and 10 of ten seeds, while the case-3 returns move by one or two serves at a setting and case 5's count at 5 Hz moves from 9 to 11. The verdicts and the optimum survive the halving, and the counts are quoted at the shipped step.

11.7 Technology in Practice

Each arm of Section 11.3 reads a camera pair and its own joint angles, and it moves six joint drives. Figure 11.5 draws the student's loop with real devices in the place of the schematic's symbols.

What Measures It

A robot player measures the ball with high-speed cameras alone, because the ball itself cannot be instrumented. The two cameras of a pair are synchronized so that each detection can be triangulated, and a neural detector can work on the raw sensor image, so that no time is spent converting it to color [1]. A player driven by artificial muscles read the ball's position from a color-based camera system at 180 Hz [11]. The stereo pair of Equation (11.5), with 200 Hz frames 10 ms late and 2 mm of noise, stands for such a system, and its dropout event for the frames a real one loses. One research player kept its returns until its cameras were slowed to 50 frames a second or delayed by 150 ms, when its swings came too late and its balls went into the net [1], a limit of the kind that case 9 measures.

TECHNOLOGY **Sensor: High-Speed Stereo Cameras**

A research player tracked the ball with two synchronized cameras with global shutters, a 4 ms exposure and 125 frames a second, mounted about 2 m above the play area on opposite sides of the table, and it triangulated each pair of detections into a position. The wide lenses that let both cameras see the whole play area made calibration harder and added uncertainty to the triangulation. When processing falls behind the frames, latency grows until frames are dropped [1], and a ball that climbs out of the field of view is lost to the robot [16]. On a trend either fault shows as a gap in the camera track, while the prediction runs on its model alone.

What Moves It

The final elements are the joint drives of a motor-driven industrial arm, which follow the references of the arm's own controller [1]. The drives of Equation (11.4), 15 Hz servos with speed and acceleration limits and soft range ends, stand for such drives. The inverse servo of Equation (11.10) plays the part of the identification a real arm needs before its motion matches the plan, and a drive detuned from that model lags, as case 5 shows.

TECHNOLOGY **Final Element: Joint Servo Drives**

Each joint of an industrial arm is a servo drive that follows a reference from the arm's own controller. In one table-tennis system the arm took a position and a speed target for each joint and returned joint feedback at 248 Hz, and its joints turned at up to 420 or 600 °/s. The controller passes the reference through a low-pass filter and gains of its own, so the arm was identified before its motion matched the simulator. Its safety layer stops the arm when it detects a collision, predicts a joint past its range, or finds a joint commanded so fast that it reaches a torque limit, and the stop freezes the robot [1]. On a trend a safety stop is a joint that halts in mid-swing and stays there.

Where the Controller Runs

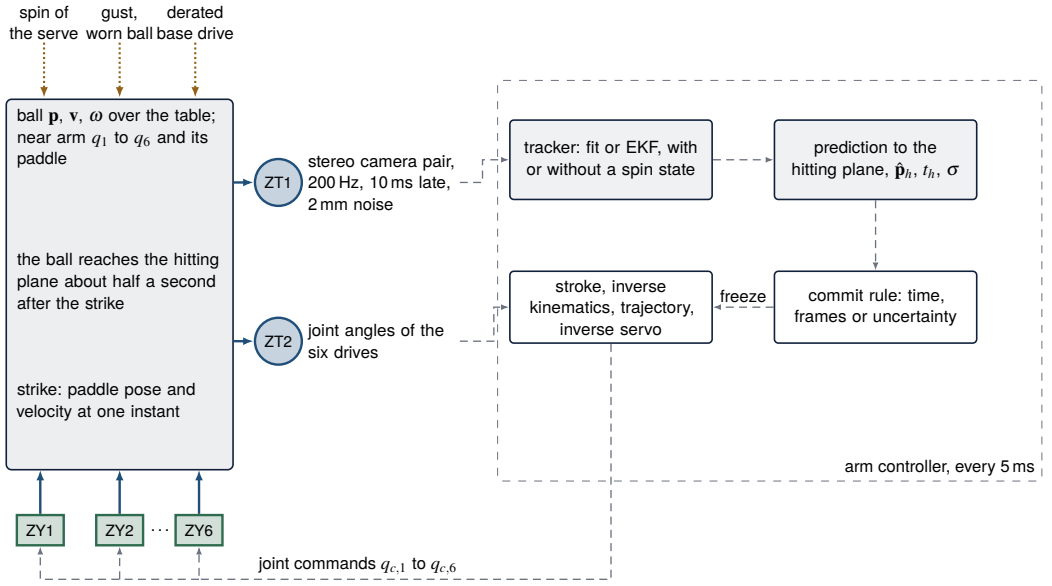
The controllers of this chapter run every 5 ms on drives integrated every 0.5 ms. A research player ran its learned policies at 100 Hz in most experiments, while the arm's controller exchanged commands and feedback at 248 Hz through low-level control threads kept apart from the learning code [1]. A later player split the work in two layers: skill policies sent joint velocity commands at 50 Hz, and a high-level controller chose a skill once for every stroke of the opponent [16]. Here the tracker, the predictor, the commit rule and the stroke planner form the upper layer, and the shaped commands and the drives the lower one.

IN PRACTICE **Measuring the Delays Before Tuning**

A fast vision loop is commissioned by measuring its delays first. One team logged time stamps through its whole software stack, measured the mean and the spread of each component's latency, and gave the same delays to its simulator. Its cameras went on opposite sides of the table, which cut the bias of the triangulation [1]. The arm and its cell fall under ISO 10218, whose Part 1 covers the industrial robot and Part 2 the robot system and its integration [18]. The camera of Equation (11.5) has one fixed latency, so the model skips this step.

What Is Modernizing It

Policies trained in simulation now move to the real arm without further training, once the simulator carries the measured delays of each component and the balls that real play produces [1, 16]. A parallel simulation that runs beside the real arm serves as a safety layer and keeps it out of states defined as undesirable [1]. Arms driven by pneumatic artificial muscles are a second route. Each muscle contracts as its air pressure rises and can only pull, so two act against each other on one joint. They



ZY1 to ZY6: joint servo drives, 15 Hz with damping 1, base, shoulder and elbow limited to 360 °/s and 3000 °/s^2, the three wrist axes to twice that.

Figure 11.5 The student's loop with real devices in place of the schematic's symbols. ZT1 is the stereo camera pair and ZT2 the joint angles. In the arm controller (dashed frame) the tracker feeds the prediction to the hitting plane, and the commit rule freezes the plan that the stroke, the inverse kinematics and the trajectory turn into six joint commands for the drives ZY1 to ZY6. Notice that the camera reaches the drives only through the commit: after it, new frames no longer change the swing.

give high accelerations, and their softness keeps the arm inside a safe region while it learns, but they are nonlinear, change with temperature and wear, and show hysteresis [11].

11.8 Limitations

The ball's drag and Magnus factors are teaching values smaller than fitted ones, its spin does not decay, and its impacts are instantaneous, with a grip law in place of a compliant contact and a table grip four times a measured one. The arms are servo-level, with exact kinematics and limited drives but no dynamics, fixed bases and hitting planes behind their end lines, and their forward stroke is limited by the elbow to about 2 m/s of paddle speed, so its drives leave at about 6.4 m/s. The cameras have no calibration error, blur or occlusion, serves are launched rather than struck, and the serve rule, lets and edge balls are left out.

> **OPEN CHALLENGE** **Fast Balls and Spin Nobody Sees**
>
> A learned robot player that beat every beginner it met still lost every match against the most advanced players. Its latency of about 100 ms limited the decisions it could make on a fast ball, and it could not read extreme spin, which opponents exploited [16]. Spin is hard because the cameras see only the ball's position, and spin shows only in a bend of the flight and in the bounce, as case 7 of this chapter shows. Progress would be a robot that reads heavy spin and answers fast balls well enough to beat advanced players. Its builders name predictive models of the ball and faster links between sensors and actuators as routes [16].

The tuned opponent is a teaching opponent, the student's pipeline with a declared blind spot and safe targets, and not a model of any player or machine. Its losses are counted by a set of rules applied to its own flight record, and a point with several faults is charged to the first that applies. The

uncertainty rule's latest-swing bound is conservative, so it commits up to 0.5 s before the hit for the drive and the spin stroke. A game's counts are sensitive to the integration step and the seed, which is why the challenge reports ten seeds beside the default one.

11.9 Conclusion

A table-tennis robot has about half a second to watch, decide and swing, and this simulator makes each part of that budget measurable on one table with two six-axis arms. Drag moves the planning serve's arrival by 131 ms and its bounce by 34 cm, so a no-drag predictor commits 3.6 cm from the true crossing and 71 ms early and returns nothing, while a drag model returns all ten serves within a centimeter of their targets. Committing 200 to 275 ms after the serve returns 19 of 20, earlier commits return 16 or 17 and later ones as few as 10. A minimum-jerk swing reaches a pose 0.3 m away in a quarter second to 0.1 mm, where stepped commands end 150 mm off, and only an estimator with a spin state reads a 40 rev/s backspin before the bounce. A slow camera defeats all 144 designs tried. The tuned opponent loses where its reach ends and its clock freezes its plan, not where its assumed spin was expected to fail it. The model is idealized in its ball, its contacts, its arms and its cameras. A fuller model would add spin decay and a compliant bounce, rigid-body arm dynamics, a moving base, struck serves and a second opponent whose blind spot lies in its model rather than its geometry. The simulator is a game in which knowing more always costs time to move.

KEY POINTS

- A robot player has about half a second to watch, decide and swing, and every look at the ball costs time to move.
- Drag moves the planning serve's arrival by 131 ms, so a no-drag predictor commits 3.6 cm from the true crossing but 71 ms early and returns nothing.
- Committing 200 to 275 ms after the serve returns 19 of 20, while earlier commits return 16 or 17 and later ones as few as 10.
- Only an estimator with a spin state reads a 40 rev/s backspin before the bounce.

Where else this applies. The same choice of when to act on a late measurement is met in Chapter 10, and the same six-axis arm keeps a ball bouncing in Volume 8, Chapter 15.

References

[1] D. D'Ambrosio, J. Abelian, S. Abeyruwan, M. Ahn, A. Bewley, J. Boyd, K. Choromanski, O. Cortes, E. Coumans, T. Ding *et al.*, "Robotic table tennis: A case study into a high speed learning system," in *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, 2023.

[2] B. Hove and J.-J. E. Slotine, "Experiments in robotic catching," in *Proceedings of the 1991 American Control Conference*. IEEE, 1991, pp. 380–386.

[3] R. L. Andersson, *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control*, ser. The MIT Press Series in Artificial Intelligence. Cambridge, MA: MIT Press, 1988.

[4] K. Mülling, J. Kober, O. Kroemer, and J. Peters, "Learning to select and generalize striking movements in robot table tennis," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 263–279, 2013.

[5] R. D. Mehta, "Aerodynamics of sports balls," *Annual Review of Fluid Mechanics*, vol. 17, no. 1, pp. 151–189, 1985.

[6] J. Nonomura, A. Nakashima, and Y. Hayakawa, "Analysis of effects of rebounds and aerodynamics for trajectory of table tennis ball," in *Proceedings of SICE Annual Conference 2010*. IEEE, 2010, pp. 1567–1572.

[7] R. Cross, "Grip-slip behavior of a bouncing ball," *American Journal of Physics*, vol. 70, no. 11, pp. 1093–1102, 2002.

[8] A. Nakashima, Y. Ogawa, Y. Kobayashi, and Y. Hayakawa, "Modeling of rebound phenomenon of a rigid ball with friction and elastic effects," in *Proceedings of the 2010 American Control Conference*. IEEE, 2010, pp. 1410–1415.

[9] L. Acosta, J. J. Rodrigo, J. A. Mendez, G. N. Marichal, and M. Sigut, "Ping-Pong player prototype – A PC-based, low-cost, ping-pong robot," *IEEE Robotics & Automation Magazine*, vol. 10, no. 4, pp. 44–52, 2003.

[10] M. Matsushima, T. Hashimoto, M. Takeuchi, and F. Miyazaki, "A learning approach to robotic table tennis," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 767–771, 2005.

[11] D. Büchler, S. Guist, R. Calandra, V. Berenz, B. Schölkopf, and J. Peters, "Learning to play table tennis from scratch using muscular robots," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3850–3860, 2022.

[12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[13] C. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.

[14] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 163–171, 1986.

[15] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *The Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[16] D. B. D'Ambrosio, S. Abeyruwan, L. Graesser, A. Iscen, H. Ben Amor, A. Bewley, B. J. Reed, K. Reymann, L. Takayama, Y. Tassa *et al.*, "Achieving human level competitive robot table tennis," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 74–82.

[17] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, ser. Advanced Textbooks in Control and Signal Processing. London: Springer, 2009.

[18] R. Bostelman and J. Falco, "Survey of industrial manipulation technologies for autonomous assembly applications," National Institute of Standards and Technology, Gaithersburg, MD, NIST Interagency/Internal Report 7844, 2012.

11.10 Exercises

11.1 ○ Concept check · 2 min

In scenario 2, The predictor, the no-drag fit commits 150 ms after each serve. At the commit its hitting point lies 3.6 cm from where the ball really crosses the hitting plane, and all ten returns go out. Which explanation is right?

(a) The point is too far off across the plane for a paddle of 75 mm radius to meet the ball

(b) The commit leaves too little time for the swing, so the paddle reaches the plane after the ball

(c) The point is close, but the fit expects the ball 71 ms early and the swing passes before it

(d) The camera noise of 2 mm per axis scatters the fit, so the aim of each stroke is random

11.2 ○ Concept check · 2 min

In scenario 3, The commit time, a commit 350 ms after the serve returns 10 of 20 serves, against 19 of 20 for a commit at 200 ms, although the later commit knows the hit time to about 11 ms rms. What is the main cause?

(a) The frames after the bounce confuse the estimator, so the later prediction is worse

(b) The serves are faster at the later setting, because the serve speeds depend on the commit

(c) The tracker stops taking frames once it commits, so the paddle aims at a stale estimate

(d) About 243 ms remain, and the swing no longer fits inside the joint limits

11.3 ○ Concept check · 2 min

In scenario 7, The hidden spin, every serve carries 40 rev/s of backspin. The EKF on the drag-and-Magnus model returns all ten serves, while the same EKF on the drag model returns none. Why?

(a) Only the drag-and-Magnus model has a spin state, which the bending of the flight can inform

(b) The drag model overstates the air drag, so its predicted ball arrives later than the real one

(c) The drag-and-Magnus model reads the spin directly from the camera frames of the ball

(d) The drag model has no bounce, so it predicts the ball to pass straight through the table

11.4 Theory and calculation · 6 min

The ball has radius $r = 20\,\text{mm}$ and mass $m = 2.7\,\text{g}$. The app uses $\rho = 1.2\,\text{kg/m}^3$ and $C_D = 0.45$ in Equation (11.1), with $g = 9.81\,\text{m/s}^2$.

(a) Compute the drag factor k_D. *Checkpoint:* $0.1257\,\text{m}^{-1}$.

(b) Compute the drag deceleration of a ball moving at 10 m/s with no wind, and the terminal speed of a ball falling with no spin. *Checkpoint:* $12.6\,\text{m/s}^2$ and 8.83 m/s.

(c) Repeat (a) and the deceleration of (b) for the fitted $C_D = 0.54$, and say in which direction the teaching value biases the app against real air. *Checkpoint:* $0.1508\,\text{m}^{-1}$ and $15.1\,\text{m/s}^2$.

11.5 Theory and calculation · 7 min

Use the impact law of Equation (11.2) for a thin hollow sphere, $\alpha = 2/3$, with the table's $e_t = 0.905$ and $\mu_t = 1.0$ and the rubber's $e_p = 0.7$. Speeds are magnitudes along the direction stated.

(a) A ball arrives at the paddle at 6 m/s along the paddle's normal. What normal speed must the paddle have for the ball to leave at 10 m/s? *Checkpoint:* 3.41 m/s.

(b) A spinless ball reaches the table moving 6.10 m/s along the table and 2.21 m/s down. Show that the bounce grips, and find the speed along the table and the topspin with which the ball leaves. *Checkpoint:* 3.66 m/s and 183 rad/s, or 29.1 rev/s.

(c) The same serve with 40 rev/s of backspin reaches the table at 6.14 m/s along the table and 2.08 m/s down. Show that the bounce slides, and find the speed along the table after it. *Checkpoint:* 2.18 m/s.

11.6 Theory and calculation · 7 min

The paddle must move 0.3 m in 0.25 s, from rest to rest.

(a) Compute the peak acceleration of a minimum-jerk (quintic) move and of a bang-bang move, and their ratio. Equation (1.5) gives the peaks. *Checkpoint:* $27.7\,\text{m/s}^2$, $19.2\,\text{m/s}^2$ and 1.44.

(b) The base joint is limited to $360\,^\circ\text{/s}$ and $3000\,^\circ\text{/s}^2$. Convert these limits to paddle speed and acceleration at a lever of 0.5 m, and compare them with (a). *Checkpoint:* 3.14 m/s and $26.2\,\text{m/s}^2$.

(c) A drive obeys Equation (11.4) in its linear range with $\zeta = 1$ and $f_j = 3\,\text{Hz}$, but its commands are shaped by Equation (11.10) for 15 Hz. For a reference that ramps at a constant rate, find how far in time the joint lags the reference once the transient has passed. *Checkpoint:* 84.9 ms.

11.7 App challenge · 8 min

Load scenario 3, The commit time, and use the default seed. Each run ends by itself after 20 serves, about 50 s of simulated time, so run at 2x. Read the Returns on the table tile when the run ends. The counts are exact.

(a) Run the scenario as shipped, committing 50 ms after each serve. *Checkpoint:* 16 of 20.

(b) Press the reload button, set Commit time to 250 ms and run again. *Checkpoint:* 19 of 20.

(c) Press the reload button, set Commit time to 350 ms and run again. *Checkpoint:* 10 of 20.

Explain the shape of the three results with the time left at the commit that Figure 11.2c gives for each setting.

11.8 App challenge · 7 min

Load scenario 7, The hidden spin, and use the default seed. Each run ends by itself after ten serves, about 22 to 26 s of simulated time. Read the Returns on the table tile when the run ends.

(a) Run the scenario as shipped, with the Fit estimator. *Checkpoint:* 0 of 10.

(b) Press the reload button, set Estimator to EKF and run again. Export the CSV file and find the first row whose `phase` column reads COMMITTED. Read `est_topspin_rps` and `truth_ball_topspin_rps` in that row. Backspin is negative. *Checkpoint:* 10 of 10. At 1.25 s the estimate reads −33.5 rev/s, within 0.5 rev/s, against a true −40 rev/s.

(c) Press the reload button, set Estimator to EKF and Predictor model to Drag, and run again. *Checkpoint:* 0 of 10.

11.9 App challenge · 8 min

Load the challenge, a game to eleven against the tuned arm, and use the default seed. A game ends by itself, so run at 2x. Read the Points won tile and the clock when the game ends.

(a) Run the challenge as shipped. *Checkpoint:* 0–11, lost at 28.2 s, within 0.1 s.

(b) Press the reload button, press Reveal suggested tuning and run again. *Checkpoint:* 11–9, won at 66.3 s, within 0.1 s.

(c) Keep the suggested settings, but set Commit rule to Time and Commit time to 150 ms. Press Reset and run again. *Checkpoint:* 4–11, lost at 44.3 s, within 0.1 s.

Which of the tuned arm's weaknesses does the student's arm share in (c)?

11.10 Application area · 3 min

The app's table grip $\mu_t = 1.0$ is four times the 0.25 measured on a real table. What would change for the 10 m/s planning serve on a real table, and what should a robot builder do before trusting a bounce model?

11.11 Application area · 4 min

A 13 m/s drive reaches the hitting plane 351 ms after it leaves the far paddle. The predictor plans after eight frames, frame k is exposed k/f_{cam} after the launch with the first at zero, and each frame arrives τ_{cam} late. A swing needs a quarter of a second or more. For the cameras (50 Hz, 100 ms), (100 Hz, 50 ms) and (200 Hz, 10 ms), find when the first plan can be made and how much time is left. Which camera would you specify?

11.12 Review the agent · 5 min

A team uses the Uncertainty commit rule with the default threshold of 10 mm. An AI assistant advises: "The threshold is loose. Lower Commit uncertainty to 2 mm so that the arm commits only once the hitting point is sharp. In scenario 3 this will raise the returns above 17 of 20." Find the flaw in the advice and say how you would check it in the app.