1 A Set of Challenging Control Problems

Charles W. Anderson W. Thomas Miller

GTE Laboratories Incorporated
40 Sylvan Road
Waltham, MA 02254

1.1 Introduction

In this chapter, we present a number of control problems as challenges to experimenters wishing to explore new ideas for building automatic controllers that improve their performance by learning from experience. The problems were chosen for the simplicity with which they can be stated and modeled and for their relevance to difficulties encountered in real control situations. The difficulties addressed by this collection of problems include incomplete system knowledge, nonlinearity, noise, and delays.

Two of the problems are taken from previous chapters—Ungar's bioreactor (Chapter ??) and Jorgensen and Schley's autolander (Chapter ??) These problems are repeated here in a concise, consistent, and complete format for the purpose of providing all of the details that would be needed by anyone wishing to test the performance of a controller on these problems in simulation. Particular constraints, parameter values, and control objectives are suggested to encourage researchers to perform the same experiments and thus facilitate quantitative performance comparisons of different control techniques.

Additional problems provided in this chapter concern balancing a pole, steering a tractor-trailer truck, steering a ship, and moving a robotic manipulator. These problems have been the subjects of research in neural network control methods or in other forms of learning control. Two more problems are from a comparative demonstration of adaptive control design techniques presented at the 1988 American Control Conference.

Each problem description includes a short introduction, the definition of the plant to be controlled, controller input and output variables, and discussions of the problem's objective, relevance, and possible extensions. Previous results are referenced if they exist. A plant is specified by its state variables, constraints, equations of motion, and parameters. Note that the plant details are for the sake of its simulation; ideally, a learning controller would function even when this knowledge of the plant is incomplete or incorrect. The problems are expressed in discrete time to ease the task of simulating the plants on a digital computer. The definitions of some of the plants are simplified by the use of intermediate variables that do not represent part of the plant's state. Intermediate variables are recognized by the lack of a time index; only state variables are indexed by time. The equations are presented in the order in which they should be evaluated, with the exception of the parameter values which follow the equations.

We hope that these problems will prove useful as testbeds for comparing the performance of neural network techniques for learning control with other approaches to control design. The best reported performance on these and similar control problems will serve as challenges to others and as benchmarks against which further results can be compared.

1.2 The Bioreactor

1.2.1 Introduction

The bioreactor (see Chapter ??) is a tank containing water, nutrients, and biological cells as shown in Figure **??**. Nutrients and cells are introduced into the tank where the cells mix with the nutrients. The state of this process is characterized by the number of cells and the amount of nutrients. The volume in the tank is maintained at a constant level by removing tank contents at a rate equal to the incoming rate. This rate is called the *flow rate* and is the variable by which the bioreactor is controlled. The bioreactor control problem is to maintain the amount of cells at a desired level.

1.2.2 Plant

State	c_1	amount of cells
	c_2	amount of nutrients
Control	r	flow rate
Constraints	$0 \leq c_1, c_2 \leq 1$	Cell and nutrient amounts are between 0 and 1.
	$0 \leq r \leq 2$	Flow rate is positive and less than or equal to 2.

Figure 1.1
The bioreactor is a tank of a liquid mixture of cells and nutrients. The objective is to control the amount of cells by adjusting the flow rate.

Initial Conditions

$c_1[0]$ a random variable from uniform distribution on the interval $(0.9c_1^*, 1.1c_1^*)$

$c_2[0]$ a random variable from uniform distribution on the interval $(0.9c_2^*, 1.1c_2^*)$

$r[0]$ a random variable from uniform distribution on the interval $(0.9r^*, 1.1r^*)$

where c_1^*, c_2^*, and r^* are defined below.

Equations of Motion

$$c_1[t+1] = c_1[t] + \Delta(-c_1[t]r[t] + c_1[t](1-c_2[t])e^{c_2[t]/\gamma})$$

$$c_2[t+1] = c_2[t] + \Delta(-c_2[t]r[t] + c_1[t](1-c_2[t])e^{c_2[t]/\gamma}\frac{1+\beta}{1+\beta-c_2[t]})$$

Parameters

β	0.02	growth rate parameter
γ	0.48	nutrient inhibition parameter
Δ	0.01	sampling interval

1.2.3 Controller Input and Output

Control Interval 0.5 s (50 times as long as the sampling interval, Δ)

Input $c_1[t]$ and $c_2[t]$ for $t = 0, 50, 100, \ldots$

Output $r[t]$ for $t = 0, 50, 100, \ldots$ and $r[t] = r[t-1]$ for all other values of t

1.2.4 Objective

The objective is to achieve and maintain a desired cell amount, $c_1^*[t]$, by altering the flow rate throughout a learning trial. If T is the number of time steps in a trial, then the objective is to minimize the cumulative measure

$$\sum_{t=0,50,100,\ldots,T} (c_1[t] - c_1^*[t])^2.$$

T should be on the order of 5000, which is equivalent to 50 seconds. After T time steps have elapsed, the state of the bioreacter is reset in the manner used to generate the initial conditions and the controller again attempts to maintain the desired cell amount. This procedure is repeated for some number of trials. The cumulative errors for each trial can be averaged, perhaps weighting the errors for the most recent trials more heavily, to form a final performance measure.

Ungar defined three kinds of bioreactor control problems. In the first problem, the bioreactor is started in a region of the state space from which a stable state is easily achieved. Let c_1^*, c_2^*, and r^* be desired values of the state and flow rate. For the first problem, $(c_1^*, c_2^*) = (0.1207, 0.8801)$ and $r^* = 0.75$. For these values, (c_1^*, c_2^*) is a stable state. Recall that the initial conditions specify that $c_1[0]$, $c_2[0]$, and $r[0]$ are within 10% of these values.

For the second problem, the desired state is $(c_1^*, c_2^*) = (0.2107, 0.7226)$ and $r^* = 1.25$. The fact that state $(0.2107, 0.7226)$ is unstable makes this problem much harder than the first.

The third problem is a combination of the first two. The desired state is first set to a stable value, $(c_1^*, c_2^*) = (0.1237, 0.8760)$ with $r^* = 1/1.3$. Then, after 100 control intervals, which equals 50 seconds, 0.05 is added to c_1^*, giving $c_1^* = 0.1737$. This shifts the problem from one of controlling about a stable desired state to one involving an unstable desired state. One increment in the value of c_1^* is sufficient.

1.2.5 Relevance

The bioreactor is a challenging problem for neural network controllers for several reasons. Although the task involves few variables and is easily simulated, its nonlinearity makes it difficult to control. For example, small changes in parameters value can cause the bioreactor to become unstable. The issues of delay, nonlinearty, and instability can be studied with the bioreactor control problem. Significant delays exist between changes in flow rate and the response in cell concentration. Nonlinearities in the bioreactor's dynamics present a challenge to networks for learning nonlinear models. Neural networks that learn to compensate for deficiencies in the performance of conventional controllers for this task can be tested. This is also a good problem for investigating combinations of methods for predicting future states with controllers that learn to avoid unstable regions of the state space.

The bioreactor easily satisfies our goals of relevance to real-world problems. Improvements in bioreactor control techniques can result in significant savings to the biochemical industries.

1.2.6 Extensions

In real bioreactors additional controls are available. The contents of a bioreactor are often heated and cooled to maintain the temperature at a level most conducive to cell growth. Also, since cell metabolism depends on contact between cells and nutrients, the contents are often stirred. Models incorporating temperature and stirring controls would result in more challenging and realistic multiple-control problem.

Other extensions involve more realistic models of concentration measurements. The accuracy of these measurements depends on the accuracy of the measurement device and on how well mixed the bioreactor's contents are. As a first step, noise can be added to the concentration variables.

1.2.7 References

Agrawal, P., Lee, C., Lim, H.C., & Ramkrishna, D., "Theoretical Investigations of Dynamic Behavior of Isothermal Continuous Stirred Tank Biological Reactors," *Chemical Engineering Science*, Vol. 37, No. 3, pp. 453–462, 1982.

Ungar, L.H., Powell, B.A., & Kamens, S., "Adaptive Networks for Fault Diagnosis and Process Control," *Computers and Chemical Engineering*, submitted, 1989.

1.3 Aircraft Autolander

1.3.1 Introduction

The autolander problem of Jorgensen and Schley (Chapter ??) concerns the landing of an aircraft as it is subjected to wind disturbances. The aircraft is represented by a linearized model with parameter values chosen to match the model to a commercial aircraft. The aircraft model includes two feedback controllers typically found in commercial aircraft—an autothrottle and a pitch autopilot. The aircraft's descent is controlled by specifying the desired elevator angle to the pitch autopilot. The speed of the aircraft is maintained at a constant value by the autothrottle.

Input to an autoland controller includes the aircraft's altitude and vertical speed and the desired values of these variables obtained from an Instrument Landing System (ILS). The ILS determines a trajectory such as the one shown in Figure **??**. The controller must generate a sequence of desired elevator angles that result in the aircraft touching down on the runway within the given ranges of horizontal position, speed, and pitch.

Figure 1.2
An autolander must adjust the elevators of the aircraft in order to guide the plane along a trajectory that is as close as possible to that specified by the Instrument Landing System as shown.

1.3.2 Plant

State

Aircraft State

u	longitudinal velocity (ft/s)
w	vertical velocity (degrees/s)
q	pitch rate (degrees/s)
θ	pitch angle (degrees)
h	altitude (ft)
x	horizontal position as negative of ground distance to desired touchdown position (ft)

Autothrottle State

u_T

Wind Disturbance State

u_{d1}, w_{d1}, w_{d2}

Control θ_c pitch angle command

Constraints $-10^{\text{o}} \leq \theta_c \leq 5^{\text{o}}$

Initial Conditions

$$u[0] = w[0] = q[0] = \theta[0] = 0$$

$$x[0] = -h[0]/\tan\gamma$$

$$h[0] = 500$$

$$u_T[0] = 0$$

$$u_{d1}[0] = w_{d1}[0] = w_{d2}[0] = 0$$

Equations of Motion

Wind Disturbances

$$
\begin{aligned}
u_{gc} &= \begin{cases} -u_h\,(1+\ln(h[t]/510)/\ln 51), & h[t] \geq 10 \\ \qquad\qquad 0 \qquad\qquad\quad , & h[t] < 10 \end{cases} \\
u_d &= u_{d1}[t] + u_{gc} \\
a_u &= \begin{cases} U_0/(100\sqrt[3]{h[t]}), & h[t] > 230 \\ \quad U_0/600 \quad\ , & h[t] \leq 230 \end{cases} \\
N_1, N_2 &= \text{random variables from standard normal distribution} \\
u_{d1}[t+1] &= u_{d1}[t] + \Delta(0.2|u_{gc}|\sqrt{2a_u}N_1/\sqrt{\Delta} - a_u u_{d1}[t]) \\
a_w &= U_0/h[t] \\
\sigma_w &= \begin{cases} \qquad 0.2|u_{gc}| \qquad\quad , & h[t] > 500 \\ 0.2|u_{gc}|(0.5+0.00098h[t]), & h[t] \leq 500 \end{cases} \\
w_d &= \sigma_w\sqrt{a_w}(a_w w_{d1}[t] + \sqrt{3}w_{d2}[t]) \\
w_{d1}[t+1] &= w_{d1}[t] + \Delta w_{d2}[t] \\
w_{d2}[t+1] &= w_{d2}[t] + \Delta(N_2/\sqrt{\Delta} - a_w^2 w_{d1}[t] - 2a_w w_{d2}[t])
\end{aligned}
$$

Pitch Autopilot

$$
\delta_E = \begin{cases} K_1(\theta_c[t] - \theta[t]) - K_2 q[t], h[t] \geq h_f \\ K_3(\theta_c[t] - \theta[t]) - K_4 q[t], h[t] < h_f \end{cases}
$$

Autothrottle

$$
\begin{aligned}
\delta_T &= K_5(u_c - u[t]) + K_5 \omega u_T[t] \\
u_T[t+1] &= u_T[t] + \Delta(u_c - u[t])
\end{aligned}
$$

Aircraft

$$
\begin{aligned}
u[t+1] &= u[t] + \Delta\{X_u(u[t]-u_d) + X_w(w[t]-w_d) + \\
&\quad X_q q[t] - g\cos\gamma\theta[t]\pi/180 + X_e\delta_E + X_T\delta_T\} \\
w[t+1] &= w[t] + \Delta\{Z_u(u[t]-u_d) + Z_w(w[t]-w_d) + \\
&\quad (Z_q - U_0\pi/180)q[t] + g\sin\gamma\theta[t]\pi/180 + \\
&\quad Z_E\delta_E + Z_T\delta_T\} \\
q[t+1] &= q[t] + \Delta\{M_u(u[t]-u_d) + M_w(w[t]-w_d) + \\
&\quad M_q q[t] + M_E\delta_E + M_T\delta_T\} \\
\theta[t+1] &= \theta[t] + \Delta q[t] \\
\dot{h} &= U_0\theta[t]\pi/180 - w[t] \\
h[t+1] &= h[t] + \Delta\dot{h} \\
V_g &= U_0\cos\gamma + u_{gc} \\
x[t+1] &= x[t] + \Delta V_g
\end{aligned}
$$

Parameters

Autopilot and Autothrottle

K_1	2.8	K_2	2.8
K_3	11.5	K_4	6.0
K_5	3.0	ω	0.1

Aircraft Response

X_u	-0.038	X_w	-0.0513	X_q	0.00152
X_E	0.00005	X_T	0.158	Z_u	0.313
Z_w	-0.605	Z_q	-0.0410	Z_E	-0.146
Z_T	0.031	M_u	-0.0211	M_w	0.157
M_q	-0.612	M_E	0.459	M_T	0.0543

Other

u_c	0 ft/s	throttle comand
u_h	20 ft/s	wind speed at 510 ft. altitude
U_0	235 ft/s	nominal speed
γ	-3^o	flight path angle
h_f	45 ft	altitude at which flare begins
g	32.2 ft/s^2	acceleration due to gravity
Δ	0.01 s	sampling interval

1.3.3 Controller Input and Output

Control Interval 0.1 s (10 times as long as the sampling interval, Δ)

Input

	$h[t]$	current altitute
	$\dot{h}$	current altitude rate of change (defined above)
	h_c	desired altitute (defined below)
	$\dot{h}_c$	desired altitude rate of change (defined below)

where $t = 0, 10, 20, \ldots$.

Output $\theta_c[t]$ pitch angle command for $t = 0, 10, 20, \ldots$. For other values of t, $\theta_c[t] = \theta_c[t-1]$.

The desired altitude and altitude rate of change are determined by the following ILS system:

Equations of Motion

Equations of motion for plant must be calculated first to determine values of u_{gc} and $\dot{h}$.

When $h[t] > h_f$:

$$\begin{aligned} h_c &= -x[t] \tan\gamma \\ \dot{h}_c &= -V_g \tan\gamma \end{aligned}$$

When $h[t] \leq h_f$ and $h[t-1] > h_f$:

$$\begin{aligned} \dot{h}_f &= \dot{h} \\ x_{c0} &= x[t] \end{aligned}$$

When $h[t] \leq h_f$ and $h[t-1] \leq h_f$:

$$\begin{aligned} \tau_x &= -\frac{h_f V_g}{\dot{h}_f - \dot{h}_{TD}} \\ h_c &= h_f \left(\frac{\dot{h}_f e^{-(x[t]-x_{c0})/\tau_x} - \dot{h}_{TD}}{\dot{h}_f - \dot{h}_{TD}} \right) \\ \dot{h}_c &= -\frac{h_f V_g \dot{h}_f e^{-(x[t]-x_{c0})/\tau_x}}{\tau_x (\dot{h}_f - \dot{h}_{TD})} \end{aligned}$$

Parameters	$\dot{h}_{TD}$	-1.5 ft/s	desired altitude rate of change on touchdown

1.3.4 Objective

Let T be the time step at which the airplane either lands or crashes, i.e., $h[T] \leq 0$. The landing is judged by four performance measures. Listed in their order of importance, they are the plane's vertical speed, horizontal position, pitch, and horizontal speed. They are defined as follows with the given desired ranges:

$$\begin{array}{cc} \text{vertical speed:} & -3 \le \dot{h} \le -1 \text{ ft/s} \\ \text{horizontal position:} & -300 \le x[T] \le 1000 \\ \text{pitch:} & -10 \le \theta[T] \le 5 \\ \text{horizontal speed:} & 200 \le V_g \le 270 \end{array}$$

1.3.5 Relevance

Flying aircraft are subject to wind disturbances that can be fatal when they occur close to the ground while landing. Autoland systems that are routinely employed as commercial aircraft are not designed to handle large wind gusts that do ocassionally occur. Systems that could learn to improve the performance of a current autoland controller in large wind conditions stand to increase the reliability and safety of landing.

1.3.6 References

Holley, W.E., “Wind Modeling and Lateral Control for Automatic Landing,” Stanford University Thesis, 1975.

Neuman, F. & Foster, J.D., “Investigation of a Digital Automatic aircraft Landing System in Turbulence,” NASA Technical Note TN D-6066, Ames Research Center, October, 1970.

Pallett, E.H.J., “Autopilot Logic for Flare Maneuver of STOL Aircraft,” *Automatic Flight Control*, London, Grenada, 1983.

1.4 Pole Balancing

1.4.1 Introduction

The pole balancing problem is the problem of learning to balance an upright pole, sometimes called an inverted pendulum. The bottom of the pole is attached by a pivot to a cart that travels along a track as shown in Figure **??**. Movement of both cart and pole is constrained to the vertical plane. The state of this system is given by the pole's angle and angular velocity and the cart's horizontal position and velocity. The only available control actions are to exert forces of fixed magnitude on the cart that push it to the left or right.

The event of the pole falling past a certain angle or the cart running into the bounds of its track is called a *failure*. A sequence of forces must

Figure 1.3
The objective of the pole-balancing problem is to keep the pole upright and the cart away from the ends of its track.

be applied that avoid failure as much as possible by balancing the pole in the center of the track. A naive controller, before learning much about this task, will be unable to avoid failures. The pole and cart system is reset to its initial state after each failure and the controller must learn to balance the pole for as long as possible.

1.4.2 Plant

State	θ	angle of pole from upright position
	$\dot{\theta}$	angular velocity of pole
	x	horizontal position of cart's center
	$\dot{x}$	velocity of cart
Control	f	force on cart

Constraints	$-12^o < \theta < 12^o$	Pole limited to small angles. Falling past this is a failure. 12^o is limit often used in conventional control to allow use of linear control law.
	$-2.4 < x < 2.4$ m	Center of cart cannot exceed limits of track. The cart moving past these limits is also a failure.
	$f = -10$ or 10 N	fixed-magnitude force (bang-bang control)

Initial Conditions $\theta[0] = \dot{\theta}[0] = x[0] = \dot{x}[0] = 0$

Equations of Motion

$$\begin{aligned}
\theta[t+1] &= \theta[t] + \Delta\,\dot{\theta}[t] \\
\dot{\theta}[t+1] &= \dot{\theta}[t] + \\
&\quad \Delta\,\frac{mg\sin\theta[t] - \cos\theta[t]\left(f[t] + m_p l\dot{\theta}[t]^2\sin\theta[t]\right)}{(4/3)ml - m_p l\cos^2\theta[t]} \\
x[t+1] &= x[t] + \Delta\, x\dot{[t]} \\
\dot{x}[t+1] &= \dot{x}[t] + \\
&\quad \Delta\,\frac{f[t] + m_p l\left(\dot{\theta}[t]^2\sin\theta[t] - \ddot{\theta}[t]\cos\theta[t]\right)}{m}
\end{aligned}$$

Parameters	g	9.8	acceleration due to gravity
	m	1.1	combined mass of pole and cart
	m_p	0.1	mass of pole
	l	0.5	distance from pivot to pole's center of mass
	Δ	0.02	sampling interval

1.4.3 Controller Input and Output

Control Interval	0.02 s (equal to the sampling interval, Δ)
Input	$\theta[t]$, $\dot{\theta}[t]$, $x[t]$, and $\dot{x}[t]$
Output	$f[t]$

1.4.4 Objective

The objective of this problem is to avoid failures. Note that this objective is equally satisfied by balancing the pole within a very narrow or a wide range in angle about the upright position as long as failure is avoided. This objective can be formalized by defining a failure signal $F[t]$ as

$$F[t] = \begin{cases} 1, & \text{if } |\theta[t]| > 12^{\mathrm{o}} \text{ and } |x[t]| > 2.4 \text{ m;} \\ 0, & \text{otherwise.} \end{cases}$$

The objective is to minimize the sum

$$\sum_{t=0}^{T} F[t]$$

over the length of an experiment of T time steps.

A measure of progress during an experiment is the number of steps between failures, i.e., the balancing time until failure. The level of performance achieved at the conclusion of a run is the balancing time since the last failure.

1.4.5 Relevance

The inverted pendulum is one of the simplest inherently unstable systems. It has been used to demonstrate a number of conventional control techniques (Cannon, 1967; Cheok & Loh, 1987; Eastwood, 1968; Roberge, 1960). The relatively large set of conventional controllers for this problem can be the basis for comparisons to neural network control methods and for the development of hybrid schemes incorporating conventional and neural network control techniques.

There are also considerable precedents for the application of neural networks to this problem, referred to in the following Results section. Results already exist with which novel approaches can be compared.

This is a failure avoidance task that involves an unstable system and is therefore related to a wide class of real problems requiring the avoidance of costly or harmful conditions, as in the control of power generation and flight control. Another advantage of this task is that many extensions can be made that facilitate the exploration of a number of different control issues.

1.4.6 Extensions

This problem can be extended in many ways. Friction can be added to the pole's pivot and to the cart's wheels. Other disturbances, like wind effects and inclinations of the track, can also be added. Such extensions can be used to compare conventional adaptive control techniques for dealing with unknown disturbances with neural network controllers.

Other extensions result from different contraints and objectives. The fixed-magnitude force can be replaced by a real-valued force bounded within a realistic range. Linear control laws can be used to balance the pole when restricted to small angles as described here. A pole allowed to swing through 360 degrees would require a nonlinear control law. Multilayer networks for learning nonlinear control laws could be tested on this extension to the problem. More than one pole can be either mounted on the cart or stacked one upon the other. Another extension is to use different actuators. Rather than directly controlling force, the velocity of a wheel motor can be controlled to move the cart back and forth.

The balancing problem becomes much more challenging when multiple tasks are defined. For example, if the pole is allowed to swing in a complete circle, the controller can be given the tasks of either balancing the pole or spinning it at a given velocity, depending on the value of a command input specifying which goal is desired.

Finally, a major extension is to move out of the world of simulation into the real world. Many physical pole and cart systems have been constructed and interfaced to computer control as part of educational courses on real-time control. The real-world system will undoubtedly exhibit difficulties not captured in simulation.

1.4.7 Results

A number of results have been obtained using additional knowledge not included in the above problem description. Guez and Selinsky (1988), Tolat and Widrow (1988), and Widrow and Smith (1964) applied supervised learning methods to either learn to mimic a human controller or a given control law. Widrow (1988) showed how supervised learning can be used when the requirement of an existing controller is replaced with the information that the desired state of the pole and cart is $(\theta, \dot{\theta}, x, \dot{x}) = (0, 0, 0, 0)$.

Methods for learning solely from the relatively un-informative failure signal have been applied to the pole balancing problem by Michie and Chambers (1968), Barto, Sutton, and Anderson (1983), Sutton (1984), Selfridge, Sutton, and Barto (1985), and Anderson (1987, 1989). In Barto, et al., (1983) a neural network reliably learned to balance a simulated pole for at least 30 minutes after experiencing less than 100 failures.

1.4.8 References

Anderson, C.W., "Learning to Control an Inverted Pendulum Using Neural Networks," *IEEE Control Systems Magazine*, vol. 9, no. 3, pp. 31–37, April 1989.

Anderson, C.W., "Strategy Learning with Multilayer Connectionist Representations," Technical Report TR87-509.3, GTE Laboratories, Incorporated, Waltham, MA, 1987. (This is a corrected version of report published in *Proc. Fourth International Workshop on Machine Learning*, Irvine, CA, pp. 103–114, June 1987.)

Barto, A.G., Sutton, R.S., & Anderson, C.W., "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, pp. 834–846, Sept.–Oct. 1983.

Bridge, I.D., "Learning to Control a Dynamically Unstable System," M.S. Thesis, College of Manufacturing, Cranfield Institute of Technology, England, Sept. 1988.

Cannon, R.H., Jr., *Dynamics of Physical Systems*, McGraw-Hill, Inc., 1967.

Cheok, K.C. & Loh, N.K., "A Ball-Balancing Demonstration of Optimal and Disturbance-Accommodating Control," *IEEE Control Systems Magazine*, vol. 7, no. 1, pp. 54–57, Feb. 1987.

Connell, M.E. & Utgoff, P.E., "Learning to Control a Dynamic Physical System," *Proc. AAAI-87*, vol. 2, pp. 456–460, American Association for Artificial Intelligence, 1987.

Eastwood, E., “Control Theory and the Engineer,” *Proc. IEE*, vol. 115, no. 1, Jan. 1968.

Guez, A. & Selinsky, J., “A Trainable Neuromorphic Controller,” *J. of Robotic Systems*, vol. 5, no. 4, pp. 363–388, August 1988.

Higdon, D.T. & Cannon, R.H., Jr., “On the Control of Unstable Multiple Output Mechanical Systems,” In *ASME Winter Annual Meeting*, Philadelphia, PA, 1963.

Jørgensen, V., “A Ball-Balancing System for Demonstration of Basic Concepts in the State-Space Control Theory,” *Int. J. Elect. Enging. Educ.*, vol. 11, pp. 367-376, 1974.

Michie, D. & Chambers, R.A., “BOXES: An Experiment in Adaptive Control,” *Machine Intelligence 2*, E. Dale & D. Michie, Eds., Edinburgh: Oliver and Boyd, pp. 137–152, 1968.

Roberge, J.K., “The Mechanical Seal,” MIT SB Thesis, May 1960.

Rosen, B.E., Goodwin, J.M., & Vidal, J.J., “State Recurrence Learning,” *First Annual Int. Neural Network Society Meeting*, Boston, MA, Sept. 1988 (Abstract appears in *Neural Networks*, vol. 1, Supplement 1, 1988, pp. 48).

Sammut, C., “Experimental Results from an Evaluation of Algorithms that Learn to Control Dynamic Systems,” *Proc. Fifth Int. Conf. on Machine Learning*, pp. 437-443, 1988.

Selfridge, O.G., Sutton, R.S., & Barto, A.G., “Training and Tracking in Robotics,” *Proc. IJCAI-85*, pp. 670–672, International Joint Conference on Artificial Intelligence, 1985.

Sutton, R.S., “Temporal Credit Assignment in Reinforcement Learning,” Doctoral Dissertation, COINS Tech. Report 84-02, University of Massachusetts, Amherst, 1984.

Tolat, V.V. & Widrow, B., “An Adaptive “Broom Balancer” with Visual Inputs,” *IEEE Int. Conf. on Neural Networks*, San Diego, CA, July 1988.

White, M.J., “Never at Rest: Learning to Control a Dynamically Unstable System,” M.S. Thesis, College of Manufacturing, Cranfield Institute of Technology, England, Sept. 1988.

Widrow, B., "The Original Adaptive Neural Net Broom-Balancer," *Int. Symposium on Circuits and Systems*, pp. 351–357, May 1987.

Widrow, B. & Smith, F.W., "Pattern-Recognizing Control Systems," *1963 Computer and Information Sciences (COINS) Symposium Proceedings*, Washington, D.C.: Spartan, 1964.

1.5 Tractor-Trailer Truck Steering

1.5.1 Introduction

Nguyen and Widrow (1989) demonstrated the application of a neural network to the problem of learning to steer a tractor-trailer truck backing up at a constant speed. Their problem definition, with minor modifications, is repeated here. The cab's front wheels move a fixed distance backward with each step. Steering is accomplished by changing the angle of the front tires with respect to the orientation of the cab. The goal is to guide the back of the trailer to a point on a loading dock with the trailer perpendicular to the dock. Refer to the diagram in Figure **??** to relate the following state variables to the physical layout of the problem.

1.5.2 Plant

State	x, y	coordinates of center of rear of trailer
	θ_s	angle of trailer, measured from positive x with counterclockwise being positive
	θ_c	angle of cab, measured from positive x with counterclockwise positive

Control	u	steering angle of front wheels relative to cab orientation, counterclockwise positive

Constraints	$x > 0$	the loading dock is at $x = 0$
	$\|\theta_s - \theta_c\| \leq 90^o$	the angle between the cab and trailer cannot be more than 90^o
	$-70^o \leq u \leq 70^o$	steering range of front wheels is limited

Figure 1.4
The tractor-trailer truck backs up at a constant speed. The objective is to position the rear of the trailer at $(x; y) = (0; 0)$ by adjusting the angle, u, of the front wheels.

Initial Conditions	$x[0]$	a random variable from uniform distribution from 0 to 100
	$y[0]$	a random variable from uniform distribution from -50 to 50
	$\theta_s[0]$	a random variable from uniform distribution from -90^o to 90^o
	$\theta_c[0]$	a random variable from uniform distribution from $\theta_s[0] - 10$ to $\theta_s[0] + 10$

Equations of Motion

$$\begin{aligned}
A &= r \cos u[t] \\
B &= A \cos(\theta_c[t] - \theta_s[t]) \\
C &= A \sin(\theta_c[t] - \theta_s[t]) \\
x[t+1] &= x[t] - B \cos \theta_s \\
y[t+1] &= y[t] - B \sin \theta_s \\
\theta_c[t+1] &= \tan^{-1}\left(\frac{d_c \sin \theta_c[t] - r \cos \theta_c[t] \sin u[t]}{d_c \cos \theta_c[t] + r \sin \theta_c[t] \sin u[t]}\right) \\
\theta_s[t+1] &= \tan^{-1}\left(\frac{d_s \sin \theta_s[t] - C \cos \theta_s[t]}{d_s \cos \theta_s[t] + C \sin \theta_s[t]}\right)
\end{aligned}$$

where $\tan^{-1}$ is from -180^o to 180^o.

Parameters	r	0.2 m	distance front tires move in one time step
	d_c	6.0 m	length of cab, from pivot to front axle
	d_s	14.0 m	length of trailer, trailer rear to pivot

1.5.3 Controller Input and Output

Input	$x[t]$, $y[t]$, $\theta_s[t]$, and $\theta_c[t]$
Output	$u[t]$

1.5.4 Objective

The objective of this control problem is to move the center of the rear of the trailer having coordinates (x, y) as close as possible to the center of the loading dock at coordinates $(x_{dock}, y_{dock}) = (0, 0)$ using only backward movements of the cab. The cab starts in a random position and orientation relative to the dock, with a random angle between the cab and trailer. Each trial terminates when any corner of the trailer or cab contacts the plane of the dock. Let T be the time step at which a trial is terminated. The squared error in the positioning of the truck is given by

$$(x_{dock} - x[T])^2 + (y_{dock} - y[T])^2 + (0 - \theta_s[T])^2,$$

since x_{dock}, y_{dock}, and 0^o are the desired values for $x[T]$, $y[T]$, and $\theta_s[T]$, respectively. Additionally, the desire for a minimum time solution can be incorporated by minimizing T.

1.5.5 Relevance

The truck backer-upper is representative of many sequential decision problems. The control decisions made early in the backing up process have substantial effects upon final results. Early moves may not always be in a direction to reduce error, but they position the truck and trailer for ultimate success. In many respects, this truck steering problem requires a control strategy that is like a dynamic programming problem solution.

1.5.6 Extensions

The problem could be extended by including obstacles in the problem. A trial would then terminate either by contacting the loading dock or one of the obstacles. The same set of fixed obstacles could be utilized for all training and testing, adding fixed constraints to the control problem. New positions for the stationary obstacles could be determined for each trial, requiring a more general capability of planning with constraints. The obstacles could also be allowed to move during each trial, providing a difficult test of planning and dynamic replanning.

The objective can be modified to incorporate the desire for particular truck paths over others in addition to the goal of reaching the loading dock. The results of learning can be biased towards shorter paths or paths requiring smaller steering angles by adding appropriate terms to the error function.

1.5.7 Results

Computer simulations of the truck and a neural network controller by Nguyen and Widrow [1989] have demonstrated workability, although no mathematical proof yet exists. Their approach involved the combination of a neural network that learned to generate good steering commands and a second neural network that learned to predict the next state of the truck. After much experience, the truck could be initially "jackknifed" and aimed in many different directions, toward and away from the dock, but as long as there was sufficient clearance the controller appeared to be capable of finding a solution.

1.5.8 References

Nguyen, D. & Widrow, B., "The Truck Backer-Upper: An Example of Self-Learning in Neural Networks." *Proc. of the Int. Joint Conference on Neural Networks*, Washington, D.C., June 18-22, 1989.

1.6 Ship Steering

1.6.1 Introduction

Figure **??**a shows an overhead view of a ship cruising on an ocean. The ship's state is given by its position, orientation, and turning rate. The ship starts at a particular position and orientation and is to be maneuvered at a constant speed through a sequence of gates. In a real ship, the turning rate would be indirectly controlled by changing the rudder angle. Here, a desired turning rate is specified directly by the controller. There is a time lag between changes in the desired turning rate and the actual rate, modeling the effects of a real ship's inertia and the resistance of the water.

1.6.2 Plant

State	x, y	the coordinates of the ship
	θ	the orientation of the ship
	$\dot{\theta}$	the actual turning rate of the ship
Control	r	the desired turning rate of the ship
Constraints	$\lvert r \rvert < 15$ degrees/second	ship cannot turn faster than 15 degrees/second

Initial Conditions $x[0] = 0.5; y[0] = \theta[0] = \dot{\theta}[0] = 0$

Equations of Motion

$$\begin{aligned}
\dot{\theta}[t+1] &= \dot{\theta}[t] + \Delta\left(r[t] - \dot{\theta}[t]\right)/T \\
\theta[t+1] &= \theta[t] + \Delta\dot{\theta}[t] \\
x[t+1] &= x[t] + \Delta V \sin\theta[t] \\
y[t+1] &= y[t] + \Delta V \cos\theta[t]
\end{aligned}$$

Figure 1.5
The ship must be steered through a sequence of gates.

Parameters	T	5	time constant of convergence to desired turning rate
	V	3 m/s	constant speed of ship
	Δ	0.2	sampling interval

1.6.3 Controller Input and Output

Control Interval	0.2 s (= Δ)
Input	$x[t]$, $y[t]$, $\theta[t]$, and $\dot{\theta}[t]$
Output	$r[t]$

1.6.4 Objective

The ship's center starts at coordinate $(x, y) = (0.5, 0)$ moving straight up ($\theta = 0$, $\dot{\theta} = 0$). The goal is to generate sequences of r values that steer the center of the ship through a number of gates in a particular order using a minimum amount of time.

A simple problem having a single gate should be attempted first. Place the sides of the gate at coordinates $(0.2, 1)$ and $(0.3, 1)$. The ship must be steered through this gate as quickly as possible. A typical trajectory is shown in Figure **??**b.

The problem becomes more challenging with additional gates. For example, using the same final gate as before, we can add four gates with sides at $(0.8, 0.3)$ and $(0.9, 0.3)$, $(0.3, 0.6)$ and $(0.4, 0.6)$, $(0.5, 0.7)$ and $(0.6, 0.7)$, and $(0.3, 0.8)$ and $(0.4, 0.8)$, as shown in Figure **??**c. This sequence of gates is similar to the one used by Anzai (1984).

1.6.5 Relevance

This problem is a good test of a controller's ability to deal with long delays and to plan for future consequences. The delays involved in the steering of mobile vehicles depend on the interactions between a vehicle and its supporting medium. Nautical ships exhibit particularly long time delays in response to rudder movements. Planning for future encounters with gates should be part of the current control decision, because the ship's position and orientation as it moves through one gate can greatly affect the ease of navigating through successive gates.

Anzai (1984) used this model of a ship steering problem to study the development of cognitive strategies in the control of systems with long delays. The evolution of ship trajectories during learning can be compared to those published by Anzai that were exhibited by humans as they learned to control Anzai's simulation.

1.6.6 Extensions

As formulated, this problem requires a controller to learn to guide the ship through a single set of gates. A straightforward extension to multiple sets of gates is realized by uniquely labeling each set of gates and providing the label for the current set of gates as an additional input to the controller. Alternatively, one can attempt a solution to the more general problem of deciding how to steer the ship when given some representation of the position of the next gate and train on a wide variety of gate sets. One representation that is similar to what the controller of a ship might experience is the distance from the ship to the next gate and the gate's direction relative to the ship's orientation. Including relative distances and directions to subsequent gates as input would allow a controller to optimize current actions for future encounters with those gates.

Other simulated effects, such as water currents and wind gusts, can be added to the model and treated either as unknown disturbances or as environmental variables that are sensed by the ship. Other additions to the model can simulate more accurately the interaction between the ship's motion and its rudder, requiring a controller to set the rudder rather than specify a desired turning rate.

1.6.7 References

Anzai, Y., "Cognitive Control of Real-Time Event-Driven Systems," *Cognitive Science*, vol. 8, pp. 221–254, 1984.

1.7 Manipulator Dynamics

1.7.1 Introduction

The complete model of a typical five or six axis industrial robot is too complicated to qualify as a straightforward control task for comparative studies of alternative control techniques (Neuman and Murray, 1987b), though considerable effort has been applied to the problem of efficient representation of manipulator dynamics, for both simulation and real time control applications (Luh, et al., 1980; Neuman and Murray, 1987a). The three axis robot described in this section (Morgan and Ozguner, 1985) and shown in Figure **??** is a reasonable compromise between system complexity (and thus realism) and ease of implementation. It is similar to the three major axes (base, upper arm, and forearm) of typical industrial robots. The model is complete, in that all joint cou-

Figure 1.6
The control of this three-joint robotic manipulator must deal with complex interactions among the links.

pling terms (centripetal and Coriolis torques, variable effective moments of inertia, etc.) are included. It is still an idealized model, however, in that all masses are assumed to be lumped at discrete points and effects such as drive train stiction are not modeled.

1.7.2 Plant

State	θ_1	angular position of robot base axis (radians)
	θ_2	angular elevation of upper arm above horizontal (radians)
	θ_3	angular elevation of forearm above horizontal (radians)
	$\dot{\theta}_1$	angular velocity of robot base axis (radians/s)
	$\dot{\theta}_2$	angular velocity of upper arm (radians/s)
	$\dot{\theta}_3$	angular velocity of forearm (radians/s)
Control	T_1	torque of base actuator (kg $\mathrm{m}^2/\mathrm{s}^2$)
	T_2	torque of upper arm actuator (kg $\mathrm{m}^2/\mathrm{s}^2$)
	T_3	torque of forearm actuator (kg $\mathrm{m}^2/\mathrm{s}^2$)

Constraints	$0 \leq \theta_2 \leq 180^{\mathrm{o}}$	upper arm cannot pass below plane of base
	$-1000 \leq T_n \leq 1000$ kg $\mathrm{m}^2/\mathrm{s}^2$	actuator torque range is limited

Equations of Motion

$$
\begin{aligned}
A_1 &= (M_1 + M_2)L_1^2 \cos^2 \theta_2[t] + \\
&\quad M_2 L_1 L_2 \cos\theta_2[t] \cos\theta_3[t] + \\
&\quad M_2 L_2^2 \cos^2 \theta_3[t] + J \\
A_2 &= (M_1 + M_2)L_1^2 \\
A_3 &= M_2 L_1 L_2 \sin(\theta_2[t] + \theta_3[t]) \\
A_4 &= M_2 L_2^2 \\
B_1 &= 2(M_1 + M_2)L_1^2 \sin\theta_2[t] \cos\theta_2[t] + \\
&\quad M_2 L_1 L_2 \sin\theta_2[t] \cos\theta_2[t] \\
B_2 &= M_2 L_1 L_2 \cos\theta_2[t] \sin\theta_3[t] + 2M_2 L_2^2 \sin\theta_3[t] \cos\theta_3[t] \\
B_3 &= M_2 L_1 L_2 \sin(\theta_2[t] - \theta_3[t]) \\
B_4 &= (M_1 - M_2)L_1^2 \sin\theta_2[t] \cos\theta_3[t] + \\
&\quad M_2 L_1 L_2 \sin\theta_2[t] \cos\theta_3[t] \\
B_5 &= M_2 L_2^2 \sin\theta_3[t] \cos\theta_3[t] + M_2 L_1 L_2 \cos\theta_2[t] \sin\theta_3[t] \\
G_1 &= -(M_1 + M_2) L_1 \cos\theta_2[t] \\
G_2 &= -M_2 L_2 \cos\theta_3[t] \\
C_1 &= B_1 \dot\theta_1[t] \dot\theta_2[t] + B_2 \dot\theta_1[t] \dot\theta_3[t] - K\dot\theta_1[t] + T_1[t] \\
C_2 &= -B_3 \dot\theta_3^2[t] - B_4 \dot\theta_1^2[t] - K\dot\theta_2[t] + G_1 g + T_2[t] \\
C_3 &= B_3 \dot\theta_2^2[t] - B_5 \dot\theta_1^2[t] - K\dot\theta_3[t] + G_2 g + T_3[t]
\end{aligned}
$$

$$
\begin{aligned}
\theta_1[t+1] &= \theta_1[t] + \Delta\dot\theta_1[t] \\
\theta_2[t+1] &= \theta_2[t] + \Delta\dot\theta_2[t] \\
\theta_3[t+1] &= \theta_3[t] + \Delta\dot\theta_3[t] \\
\dot\theta_1[t+1] &= \dot\theta_1[t] + \Delta C_1 / A_1 \\
\dot\theta_2[t+1] &= \dot\theta_2[t] + \Delta \frac{C_2 A_4 - C_3 A_2}{A_2 A_4 - A_3^2} \\
\dot\theta_3[t+1] &= \dot\theta_3[t] + \Delta \frac{C_2 A_3 - C_3 A_2}{A_3^2 - A_2 A_4}
\end{aligned}
$$

Parameters	J	0.5 kgm^2	rotational inertia of base
	M_1	10 kg	point mass between upper arm and forearm
	M_2	5 kg → 20kg	point mass at end of arm (including payload)
	L_1	0.6 m	length of upper arm
	L_2	0.8 m	length of forearm
	K	20 kgm^2/s	friction coefficient for all actuators
	g	9.8 m/s^2	acceleration due to gravity
	Δ	0.001	sampling interval

1.7.3 Controller Input and Output

Control Interval 0.01 s (10 times as long as the sampling interval, Δ)

Input $f(\theta_i[t]), i = 1,2,3$ where $t = 0, 10, \ldots$ and $f(\theta_i)$ models the limited resolution of joint angle sensors. Let the resolution be 0.0002 radians, so $f(\theta_i) \in \{\pm 0.0002n | n = 0, 1, 2, \ldots\}$. There are no other inputs. Sensors for joint velocities and accelerations are assumed to be unavailable.

Output $T_i[t], i = 1, 2, 3$, for $t = 0, 10, \ldots$ and $T_i[t] = T_i[t-1]$ for other values of t.

1.7.4 Objective

Trajectory Following Task: The trajectory following task involves tracking predetermined trajectories $(\theta_1^*[t], \theta_2^*[t], \theta_3^*[t])$ representing high speed movements of the arm. The trajectories used should traverse a wide range of arm configurations with high velocities and accelerations relative to the maximum values possible given the torque limits on the actuators. The controller should be able to accommodate variable payload masses, M_2, within the specified range, either by including the mass as an explicit network input or by adapting to each mass individually. Performance should be evaluated both in terms of root-mean-square position error, computed over each test trajectory, and in terms of maximum instantaneous position error during each trajectory. Let time be indexed from 0 to T over a test trajectory and let N be the number of control intervals within this period. Then the root-mean-square position error is given by

$$\sqrt{\frac{1}{N}\sum_{t=0,10,\ldots,T}\sum_{i=1}^{3}(\theta_i^*[t]-\theta_i[t])^2}$$

and the maximum instantaneous position error by

$$\max_{t=0,10,\ldots,T}\sqrt{\sum_{i=1}^{3}(\theta_i^*[t]-\theta_i[t])^2}.$$

Tests should be designed to evaluate speed of convergence and absolute performance using repetitions of a single trajectory, to evaluate learning interference using multiple trajectories, and to evaluate generalization using random trajectories.

Trajectory Planning Task: The same dynamic model can be used to study optimal trajectory planning. The goal is to move the arm from an initial position at time step $t = 0$ to a final position at a specified time step $t = T$, minimizing the objective function:

$$C_T = \sum_{t=0,10,\ldots,T}\sum_{i=1}^{3}\left(T_i[t]-T_i[t-1]\right)^2.$$

This is the minimum torque-change criteria suggested by Uno, Kawato, and Suzuki (1989; and see Chapter ??). Performance should be evaluated both in terms of the objective function above and in terms of the position error at the final time T.

1.7.5 Relevance

The three axis articulated manipulator model contains many of the same characteristics as real world control problems involving mechanical systems. The parameter values have been chosen such that the nonlinear joint coupling effects will be important and the control characteristics will be sensitive to payload over the range specified. The problem dynamics are dependent on velocity and acceleration, but these quantities cannot be measured in the model. Only serial position measurements are available. Additional real world difficulties are included in the torque limits and the fixed position measurement resolution. The challenges involved in learning to control the dynamics of such a system with high accuracy, over a reasonable subset of the operating space, with good generalization to new movements and with good resistance to learning interference, are formidable.

1.7.6 Extensions

The problem can be extended in many ways. Noise can be added to the sensor measurements. A more realistic motor model can be used for the actuators. The actuator model can be changed to represent the more complicated dynamics of pneumatic actuators. Assume that the hand position can be measured in a cartesian frame of reference (with a limited measurement resolution such as 0.1 mm). The trajectory following and trajectory planning tasks can then be implemented in hand coordinates rather than joint coordinates.

1.7.7 Results

Different approaches to the control of simulated robot dynamics using neural networks can be found in several places (Kawato, Furukawa, and Suzuki, 1987; Miller, Glanz, and Kraft, 1987; Guez and Selinsky, 1988; Goldberg and Pearlmutter, 1988). The models used in those studies vary, but are sufficiently similar to that described in this section to be useful for obtaining insight into possible control architectures.

Goldberg, K. & Pearlmutter, B., "Using Neural Networks to Learn the Dynamics of the CMU Direct-drive Arm II," CMU Robotics Institute Report, Pittsburgh, PA., Aug. 1988.

Guez, A. & Selinsky, J., "A Trainable Neuromorphic Controller," *Journal of Robotic Systems*, vol. 5, pp. 363–388, 1988.

Kawato, M., Furukawa, K., & Suzuki, R., "A hierarchical neural-network model for control and learning of voluntary movement," *Biological Cybernetics*, vol. 57, 1987, pp. 169–185.

Luh, J. Y. S., Walker, M. W., & Paul, R. P., "On-line Computational Scheme for Mechanical Manipulators." *Trans. ASME, J. of Dynamic Systems, Measurement and Control*, vol. 120, pp. 69–76, 1980.

Miller, III, W.T., Glanz, F.H., & Kraft, L.G., "Application of a general learning algorithm to the control of robotic manipulators," *Int. J. of Robotics Research*, vol. 6, no. 2, 1987, pp. 84–98.

Morgan, R. G. & Ozguner, U., "A Decentralized Variable Structure Control Algorithm for Robotic Manipulators." *IEEE J. of Robotics and Automation*, vol. RA-1, pp. 57–65, 1985.

Neuman, C. P. & Murray, J. J., “Customized Computational Dynamics.” *J. of Robotic Systems*, vol. 4, pp. 503–526, 1987(a).

Neuman, C. P. & Murray, J. J., “The Complete Dynamic Model and Customized Algorithms of the PUMA Robot.” *IEEE Trans. Systems, Man and Cybernetics*, vol. SMC-17, pp. 635–644, July/August, 1987(b).

Uno, Y., Kawato, M. & Suzuki, R., “Formulation and Control of Optimal Trajectory in Human Multijoint Arm Movement - Minimum Torque-Change Model,” *Biological Cybernetics*, in press.

1.8 Problems from the ACC Showcase of Adaptive Controller Design

1.8.1 Introduction

At the 1988 American Control Conference in Atlanta, leading researchers in adaptive control assembled to present a “Showcase of Adaptive Controller Designs” (Astrom, 1988; Goodwin, Salgado, and Middleton, 1988; Huang and Morse, 1988; Johnson, 1988; M'Saad, Landau, Samaan, and Duque, 1988; Masten and Cohen, 1988; Narendra and Duarte, 1988). The purpose was to highlight current adaptive control techniques and to enable comparisons of methodology and capabilities. Each group of researchers applied their technique to two simple control problems.

The comparison of neural network control methods on these problems to the results published in the ACC showcase could initiate a series of benchmark studies spanning the neural network and adaptive control fields. Additional showcases are planned for future ACC conferences and would be excellent forums for introducing neural network control approaches to an audience of researchers active in the control field.

The ACC problem definitions are tailored to the application of adaptive control techniques. The structures of the plants to be controlled are provided, along with the parameters of the plant. Control designers can assume knowledge of the plant's structure and particular values of the parameters. A command signal is provided as input to the controller. The adaptive controllers are to respond to the command with control actions that force the plant to follow a given “idealized” response, in spite of variations in the plant parameters and unknown disturbances added to the controller's output.

To make these problems more suitable for the application of learning methods, we make the following changes. The a priori knowledge of plant structure and parameter value ranges are replaced with a phase of learning during which the learning controller interacts with the simulated plant operating in various parameter value regimes. The controller must learn a control law that performs well under the experienced parameter values and command and disturbance signals. The learning phase is followed by a testing phase that is identical to the testing conditions specified in the ACC problem definitions, including the parameter values and command and disturbance signals.

Our reformulation of the ACC problems are stated below. The two problems differ only in their plant equations and ideal responses—one problem involves a first-order plant and the other a second-order plant.

1.8.2 First-Order Plant

State x

Control u

Disturbance d unknown disturbance added to controller output

Initial Conditions $x[0] = 0$

Equation of Motion

$$x[t+1] = x[t] + \Delta\ (-ax[t] + K(u[t] + d[t]))$$

Parameters	K	$-0.5 \leq K \leq 2.0$	
	a	$-2.0 \leq a \leq 2.0$	
	Δ	0.1	sampling interval

1.8.3 Second-Order Plant

State $x, \dot{x}$

Control u

Disturbance d unknown disturbance added to controller output

Initial Conditions $x[0] = \dot{x}[0] = 0$

Equations of Motion

$$\begin{aligned} x[t+1] &= x[t] + \Delta\, \dot{x}[t] \\ \dot{x}[t+1] &= \dot{x}[t] + \Delta\ (-a_2\dot{x}[t] - a_1 x[t] + K(u[t] + d[t])) \end{aligned}$$

Parameters	K	$-0.5 \leq K \leq 2.0$	
	a_1	$-3.0 \leq a_1 \leq 3.0$	
	a_2	$-2.0 \leq a_2 \leq 2.0$	
	Δ	0.1	sampling interval

1.8.4 Controller Input and Output

Control Interval	0.1 s $(= \Delta)$
Input	$x[t]$ and $\dot{x}[t]$ (for the second-order plant)
Output	$u[t]$

1.8.5 Objective

The objective of this control problem, as stated in the ACC Showcase, is to force the output of the plant to follow an ideal response, r, defined differently for the two problems:

for first-order problem: $r[t+1] = r[t] + \Delta\,(-r[t] + c[t])\,;$

for second-order problem: $r[t+1] = r[t] + \Delta\,\dot{r}[t],$
$\dot{r}[t+1] = \dot{r}[t] + \Delta\,(-1.4\dot{r}[t] - r[t] + c[t])\,;$

where c is the command and $r[0] = \dot{r}[0] = 0$. This response is to be followed for the given command and disturbance trajectories spanning a testing phase of 20 seconds. Let t_0 be the initial time step of the testing phase and t_f be the final time step, i.e., $t_f = t_0 + 20/\Delta$. The goal, then, is to minimize

$$\sum_{t=t_0}^{t_f} (r[t] - x[t])^2,$$

where c and d are as defined in the ACC Showcase:

$$c[t] = \begin{cases} 1, & t_0 \leq t < t_0 + 8/\Delta; \\ -1, & t_0 + 8/\Delta \leq t \leq t_f, \end{cases}$$

and

$$d[t] = \begin{cases} -1, & t_0 \leq t < t_0 + 6/\Delta; \\ 2, & t_0 + 6/\Delta \leq t < t_0 + 15/\Delta; \\ 1, & t_0 + 15/\Delta \leq t \leq t_f, \end{cases}$$

These signals are piecewise constant; c changes value at 8 seconds and d changes value at 6 and at 15 seconds.

1.8.6 Details of Training Procedure

In the ACC Showcase, the goal was to design controllers that behave well for a range of parameter values. In order for a learning controller to converge on a control law that results in good performance, it must have experience with the plant when it is operating under a variety of values for its parameters.

The parameter values can be changed in many ways. For example, their ranges can be divided into a number of equally-spaced values. The parameter values can remain constant for short periods of time and then one parameter value can be changed to its next higher or lower value. The set of possible parameter values are stepped through until every combination of values has been experienced, after which the cycle is repeated. A random sequence of parameter values can be generated by periodically setting the parameters to values selected from a uniform probability distribution over their allowable ranges. An alternative is for the parameter values to take uniform random walks. It is important to specify in any report of experimental results the method used to vary the parameter values.

In addition to various parameter values, the controller should also be exposed to a large variety of commands and disturbances during learning. The set of possible command and disturbance signals is not constrained by the ACC Showcase problem definitions, except for the particular trajectories to be used for testing the controller. One reasonable assumption is that they should not exceed the values used for testing. Thus, commands can be limited to the range $[-1, 1]$ and disturbances to the range $[-2, 2]$.

One strategy for varying commands and disturbances is to change their values at random times, setting them to values selected randomly from the above ranges. Additional assumptions can be made about the maximum frequency of changes, since the command and disturbance vary only once and twice, respectively, during the 20 second testing phase.

1.8.7 Relevance

There are two philosophies of adaptation to unknown variations in a plant. The conventional philosophy is to adjust the parameters of a control law in order to produce control actions more appropriate for the current plant parameters and disturbances. This adaptation must take place whenever parameters and disturbances change. The learning philosophy is to remember these adaptations so that when similar changes in the plant occur in the future, good control actions are produced without waiting for further adaptation. This presupposes that changes in the plant can be represented by the input to the controller. The controller must be given more than just the current state and error in the plant's output. For some problems, providing a recent history of values taken by available plant outputs might be sufficient to represent the plant changes. For example, a large disturbance might be indicated by a sudden change in the trajectory of the plant's outputs. Neural networks could learn associations from the possibly large input vectors of previous plant outputs to appropriate control actions.

The ACC Showcase problems, with their obvious relevance to adaptive control, would provide a good testbed for comparing these two philosophies of adaptation. Results from testing neural network controllers on these problems can be directly compared with the existing results from the application of conventional adaptive control designs. Such comparisons might even lead to the inclusion of neural network controllers in future ACC Showcases.

1.8.8 References

Astrom, K.J., "Robust and Adaptive Pole Placement," *Proceedings of the 1988 American Control Conference*, Atlanta, GA, pp. 2423–2428, June, 1987.

Goodwin, G.C., Salgado, M.E., & Middleton, R.H., "Indirect Adaptive Control: An Integrated Approach," *Proceedings of the 1988 American Control Conference*, Atlanta, GA, pp. 2440–2445, June, 1987.

Huang, J. & Morse, A.S., "A Computer Study of Adaptive Control Systems," *Proceedings of the 1988 American Control Conference*, Atlanta, GA, pp. 2446–2451, June, 1987.

Johnson, C.D., "Applications of a New Approach to Adaptive Control," *Proceedings of the 1988 American Control Conference*, Atlanta, GA, pp. 2452–2460, June, 1987.

Masten, M.K. & Cohen, H.E., "Introduction to a Showcase of Adaptive Controller Design," *Proceedings of the 1988 American Control Conference*, Atlanta, GA, pp. 2418–2422, June, 1987.

M'Saad, M., Landau, I.D., Samaan, M., & Duque, M., "Performance Oriented Robust Adaptive Controller," *Proceedings of the 1988 American Control Conference*, Atlanta, GA, pp. 2434–2439, June, 1987.

Narendra, K.S. & Duarte, M.A., "Robust Adaptive Control Using Combined Direct and Indirect Methods," *Proceedings of the 1988 American Control Conference*, Atlanta, GA, pp. 2429–2433, June, 1987.

1.8.9 Acknowledgements

Details of the bioreactor problem were provided by Lyle Ungar. Charles Jorgensen and Charles Schley helped clarify the description of the autolander problem. Melvyn White and Ian Bridge added several items to the list of references for the pole balancing problem. Derrick Nguyen contributed details of his implementation of the truck steering problem. Richard Sutton provided useful suggestions on the form and content of the problem descriptions.