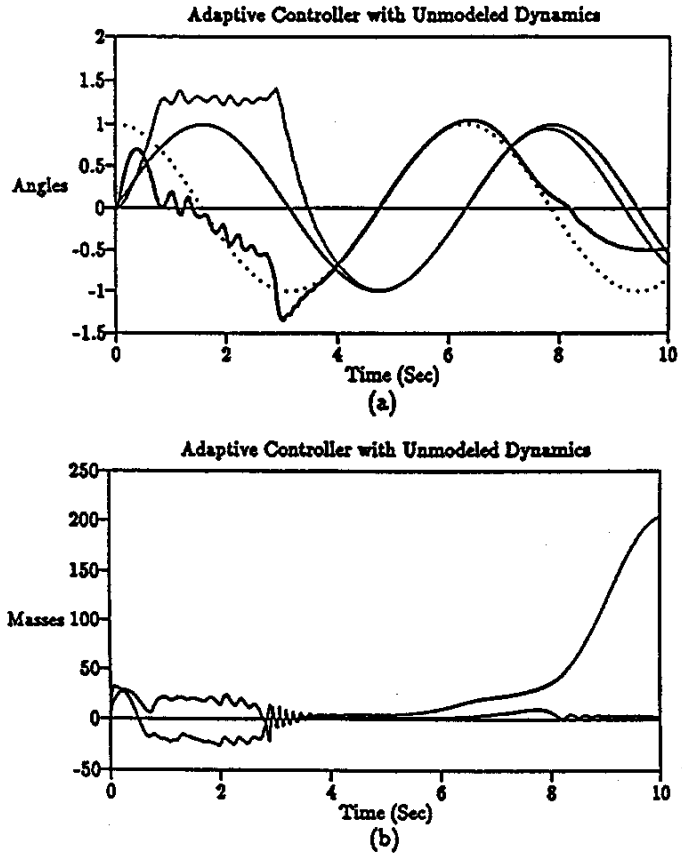


Figure 3.4.4: Response using adaptive controller with incorrect regression matrix, showing the effects of unmodelled dynamics. (a) Actual and desired joint angles. (b) Mass estimates.

the rest of $W(x)$ was not changed. The simulation results shown in Fig. 3.4.4 reveal that the performance of the adaptive controller is very bad. This corresponds to *unmodelled dynamics* which generally destroys the performance of adaptive controllers. Several techniques are available for making adaptive controllers *robust to unmodelled dynamics*, including the e-modification (Narendra and Annaswamy 1987), the σ-modification (Ioannou and Kokotovic 1984), and deadzone techniques (Kreisselmeier and Anderson 1986). These all involve adding correction terms to the tuning algorithm (3.4.14). □

3.4.4 Robust Control

Robust controllers for robot arms (Spong *et al.* 1987, Corless 1989, Dawson *et al.* 1990) comprise a large class of controllers, some of which are based on approximation techniques. Referring to Equation (3.4.8) and Fig. 3.4.1, in adaptive controllers the primary design effort goes into selecting a dynamic estimate $\hat{f}$ for the nonlinear function (3.4.7) that is tuned on-line. By contrast, in robust controllers, the primary design effort goes into selecting the robust term $v(t)$. An advantage of standard

robust controllers is that they have no dynamics, so they are generally simpler to implement. On the other hand, adaptive controllers are somewhat more refined in that the dynamics are learned on-line and less control effort is usually needed. Furthermore, in adaptive control it is necessary to compute a regression matrix $W(x)$, while in robust control it is necessary to compute a *bounding function* $F(x)$. In some modern techniques, robust and adaptive techniques are combined to provide the advantages of each class. These are called adaptive-robust controllers, or robust-adaptive controllers. Two popular robust controllers are now described.

Robust Saturation Controller. A *robust saturation controller* is given by

$$\begin{aligned} \tau &= \hat{f} + K_v r - v \\ v &= \begin{cases} -r\frac{F(x)}{\|r\|}, & \|r\| \geq \varepsilon \\ -r\frac{F(x)}{\varepsilon}, & \|r\| < \varepsilon \end{cases} \end{aligned} \tag{3.4.16}$$

where $\hat{f}$ is an estimate for $f(x)$ that is not changed on-line— for instance, a PD-gravity-based robust controller would use $\hat{f} = G(q)$, ignoring the other nonlinear terms. In computing the robust control term $v(t)$, ε is a small design parameter, $\|\cdot\|$ denotes the norm, and $F(x)$ is a known scalar function that bounds the uncertainties $\tilde{f} = f - \hat{f}$ so that

$$\|\tilde{f}\| \leq F(x). \tag{3.4.17}$$

The intent is that $F(x)$ is a simplified function that can be computed using the bounding properties in Table 3.2.1 even if the exact value of the complicated nonlinear function $f(x)$ is unknown.

The performance of the robust controller is described in the next theorem and illustrated in a subsequent example.

Theorem 3.4.3 (Robust Saturation Controller) :

Suppose the disturbance $\tau_d(t)$ in (3.4.1) is zero and the desired trajectory $q_d(t)$ is bounded according to (3.4.5). Then, using the robust controller (3.4.16), the tracking error norm $\|r(t)\|$ is eventually bounded to the neighborhood of ε.

Proof:

Using the proposed control yields the closed-loop error dynamics

$$M\dot{r} = -V_m r - K_v r + \tilde{f} + v,$$

with $\|\tilde{f}\| < F(x)$ the known function. Select the Lyapunov function candidate

$$L = \frac{1}{2} r^T M(q) r.$$

Differentiate to discover

$$\dot{L} = \frac{1}{2} r^T \dot{M} r + r^T M \dot{r}$$

whence substitution from the error dynamics yields

$$\begin{aligned} \dot{L} &= \frac{1}{2} r^T \dot{M} r - r^T V_m r - r^T K_v r + r^T (\tilde{f} + v) \\ &= \frac{1}{2} r^T (\dot{M} - 2V_m) r - r^T K_v r + r^T (\tilde{f} + v), \end{aligned}$$

so that skew-symmetry property P3 yields

$$\dot{L} = -r^T K_v r + r^T(\tilde{f} + v).$$

Recall from Chapter 2 the norm property $\sigma_{min}(K_v)\|r\|^2 \leq r^T K_v r$, with $\sigma_{min}(K_v)$ the smallest singular value of the gain matrix K_v. (If K_v is diagonal, this is simply equal to the smallest gain value.) Therefore,

$$\begin{aligned} \dot{L} &\leq -\sigma_{min}(K_v)\|r\|^2 + \|r\| \cdot \|\tilde{f}\| + r^T v \\ \dot{L} &\leq -\sigma_{min}(K_v)\|r\|^2 + \|r\|F(x) + r^T v. \end{aligned}$$

There are now two cases to consider.

Case 1. $\|r\| \geq \varepsilon$. In this case, according to the definition of the robust control term $v(t)$ one has

$$\begin{aligned} \dot{L} &\leq -\sigma_{min}(K_v)\|r\|^2 + \|r\|F(x) - \|r\|^2 F(x)/\|r\| \\ &\leq -\sigma_{min}(K_v)\|r\|^2 + \|r\|F(x) - \|r\|F(x) \\ \dot{L} &\leq -\sigma_{min}(K_v)\|r\|^2. \end{aligned}$$

Therefore, $\dot{L}$ is negative definite in terms of $\|r(t)\|$. Hence, L is decreasing in this region and $\|r\|$ decreases towards ε.

Case 2. $\|r\| < \varepsilon$. In this case, according to the definition of the robust control term $v(t)$ one has

$$\begin{aligned} \dot{L} &\leq -\sigma_{min}(K_v)\|r\|^2 + \|r\|F(x) - \|r\|^2 F(x)/\varepsilon \\ &\leq -\sigma_{min}(K_v)\|r\|^2 + \|r\|F(x)\left(1 - \frac{\|r\|}{\varepsilon}\right). \end{aligned}$$

The last term is generally positive in this region, so that nothing can be said about whether L is increasing or decreasing. In general, L may be increasing in this region so that $\|r\|$ increases towards ε. □

This proof is typical of robust control and recurs in various forms throughout the book— During the proof, various norm inequalities are used to manipulate $\dot{L}$, ending finally in the selection for the robust control term $v(t)$ overbounding some remaining uncooperative terms. It is often required to consider different disjoint regions. The result is generally that the tracking error is bounded by some 'small enough' value, though not equal to zero. Contrast this with the situation in adaptive control (e.g. Theorem 3.4.2)

Variable Structure Robust Controller. Another robust controller is the *variable structure robust controller* (Slotine 1985)

$$\begin{aligned} \tau &= \hat{f} + K_v r - v \\ v &= -(F(x) + \eta)sgn(r) \end{aligned} \tag{3.4.18}$$

where $sgn(\cdot)$ is the signum function and $F(x)$ is a known function computed using the properties in Table 3.2.1 to bound the uncertainties $\|f - \hat{f}\|$. The design parameter η is selected as a small value. This controller takes advantage of the properties of sliding mode or variable structure controllers to provide its robustness.

Example 3.4.2 (Design and Performance of Robust Controller) :
To implement the robust controllers just described, it is necessary to determine a bound $F(x)$ on the functional estimation error $\tilde{f}$. This may be accomplished using the bounding properties in Table 3.2.1. For the 2-link planar arm, for instance, the required bounds are located in Example 3.2.2.

a. Control Design and Bounding Function. Assume, for instance, that a robust PD gravity controller is to be implemented. Then one has $\hat{f} = G(q)$ and the control input (3.4.8) is

$$\tau = K_v r + G(q) - v.$$

Compare this to the computed-torque PD-gravity controller (3.3.16).

For this choice of control, according to (3.4.7) one has

$$\tilde{f} = f(x) - \hat{f} = M(q)(\ddot{q}_d + \Lambda\dot{e}) + V_m(q,\dot{q})(\dot{q}_d + \Lambda e) + F(\dot{q}).$$

Now, some norm inequalities (Chapter 2) and the bounding properties in Table 3.2.1 can be used to determine that

$$\begin{aligned} \|\tilde{f}\| &\leq \|M\| \cdot \|\ddot{q}_d + \Lambda\dot{e}\| + \|V_m(q,\dot{q})\| \cdot \|\dot{q}_d + \Lambda e)\| + \|F(\dot{q})\| \\ &\leq \mu_2\|\ddot{q}_d + \Lambda\dot{e}\| + v_B\|\dot{q}\| \cdot \|\dot{q}_d + \Lambda e\| + f_B\|\dot{q}_d - \dot{e}\| + k_B \\ &\equiv F(x), \end{aligned}$$

where the inertia matrix and Coriolis/centripetal bounds are given for the 2-link arm in Example 3.2.2 as

$$\begin{aligned} \mu_2 &= (m_1 + m_2)a_1^2 + 2m_2a_2^2 + 3m_2a_1a_2. \\ v_B &= m_2a_1a_2 \end{aligned}$$

and the friction bounds f_B, k_B depend on the arm, gears, and actuators used. In this example friction will be neglected. One recalls that $x \equiv [e^T \;\; \dot{e}^T \;\; q_d^T \;\; \dot{q}_d^T \;\; \ddot{q}_d^T]^T$.

b. Computer Simulation. A function M file was written in MATLAB (1994) to simulate the robust saturation controller— this procedure is very direct. The robust controller is an easy modification of the function routine given in Example 3.3.1. The CT controller equations in Fig. 3.3.2 are simply replaced by the robust saturation controller equations (3.4.16). The lines of code implementing the robust saturation controller are given in Fig. 3.4.5. Note that this code is significantly simpler than either the CT controller in Example 3.3.1 or the adaptive controller in Example 3.4.1. No dynamics are contained in the robust controller so that additional states for ode23 are not needed.

In typical robust controllers, there are no controller dynamics so that the performance does not improve with time. However, with good designs (and large enough control gains) the errors are bounded so that they are small enough. Typical plots are like those in Fig. 3.4.6, where the errors are always small, though nonzero, but do not become smaller with time. □

3.4.5 Learning Control

In many industrial applications robot manipulators are used to perform the same task repeatedly, such as in spray painting, short assembly operations, component insertion, and so on. In such repetitive motion situations, information from one iteration can be recorded and used to improve the performance on the next iteration.

```
% file robrob.m, to be called by MATLAB routine ode23

function xdot= robrob(t,x) ;

% compute desired trajectory as in Fig. 3.3.2

% Use period= 2*pi ; amp1= 1 ; amp2= 1 ;

% Robust control input

   m1= 0.8 ; m2= 2.3 ; a1= 1 ; a2= 1 ; g= 9.8 ;  % arm parameters
   Kv= 20*eye(2) ; lam= 5*eye(2);  eps= 0.1 ;  % controller parameters

   % tracking errors
      e= qd - [x(1) x(2)]' ;
      ep= qdp - [x(3) x(4)]' ;
      r= ep + lam*e ;

   % compute bounding function
      mu2= (m1+m2)*a1^2 + 2*m2*a2^2 + 3*m2*a1*a2 ;
      vB= m2*a1*a2 ;
      fsig= qdp + lam*e ;
      fsigp= qdpp + lam*ep ;
      F= mu2*norm(fsigp) + vB*norm([x(3) x(4)]')*norm(fsig) ;

   % control torques.
      G= [(m1 + m2)*g*a1*cos(x(1)) + m2*g*a2*cos(x(1) + x(2))
            m2*g*a2*cos(x(1) + x(2))] ;
      div= max([norm(r)  eps]') ;
      v= -r*F/div ;
      tau= Kv*r + G - v ;
      tau1= tau(1) ;  tau2= tau(2) ;

% ROBOT ARM DYNAMICS from Fig. 3.3.2
```

Figure 3.4.5: Robust controller.

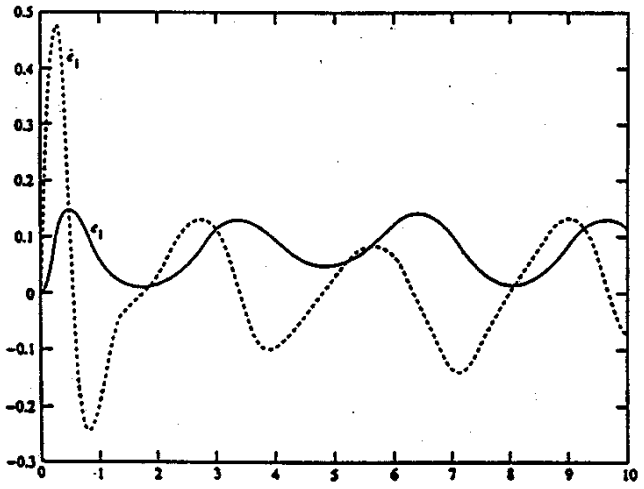


Figure 3.4.6: Typical behavior of robust controller.

This is termed *repetitive motion learning control.* Using the filtered error approach of section 3.4.1, it is direct to derive the learning controller of Sadegh *et al.* (1990) for the robot arm (3.4.1).

Let $\ell = 1, 2, \ldots$ denote the iteration number of the trajectory repetition. Then, using information from the $(\ell - 1)$-st iteration, the controller for the ℓ-th iteration is given by

$$\begin{aligned} \tau_\ell &= \hat{f}_\ell + K_v r - v \\ v &= -(K_p e + K_s \|e\|^2 r) \\ \hat{f}_\ell &= \hat{f}_{\ell-1} + K_L r \end{aligned} \tag{3.4.19}$$

where the filtered error is $r = \dot{e} + \Lambda e$ and the tracking error is $e = q_d - q$, with $q_d(t)$ the specified trajectory to be followed repeatedly. The gains K_v, K_p, K_s are positive diagonal design matrices, and K_L is a positive diagonal learning gain matrix. The function $\hat{f}_\ell$ is a *learning term* that uses its value on the previous iteration to improve on an estimate for the nonlinear function appearing in the error analysis. The functions $e(t), r(t)$ appearing in the equations may be interpreted as $e_\ell(t), r_\ell(t)$, the errors ocurring on the ℓ-th repetition.

3.5 CONCLUSIONS

In this chapter we have provided the basics for robot servo-controller design by summarizing kinematics, Jacobians, and dynamics for serial-link robot manipulators. Computed-torque control was reviewed; this technique is important because it provides a general framework for unifying a variety of robot control algorithms. It was shown how to simulate robot controllers using a digital computer. We discussed classical joint control, digital control, adaptive control, robust control, and repetitive learning control. Extensive examples were provided to show the sort of behavior that can be expected from each sort of controller.

Standard commercially available robot controllers generally employ PD or PID classical joint control, though the ADEPT controller does provide velocity feedforward and a basic sort of force control. PID control allows positioning at a sequence of 'via' points taught by a human user using a teach pendant, but does not allow accurate dynamic following of a prescribed trajectory. Since adaptation and learning techniques are seldom used, there is no mechanism for accurately correcting for unknown payload masses, varying and unknown frictions, and unknown motor dynamics. As performance requirements for improved positioning accuracy increase in semiconductor manufacturing applications and elsewhere, outmoded PID control technology lags further behind in providing desired accuracy and speed of response. In subsequent chapters we shall describe biologically motivated learning adaptation techniques for improving the capabilities of robot controllers.

3.6 REFERENCES

Åström, K.J. and B. Wittenmark, *Adaptive Control*, Addison-Wesley, Reading, MA, 1989.

Craig, J., *Adaptive Control of Mechanical Manipulators*, Addison-Wesley, Reading, MA, 1985.

Colbaugh, R., H. Seraji, and K. Glass, "A new class of adaptive controllers for robot trajectory tracking," *J. Robotic Systems*, vol. 11, no. 8, pp. 761-772, 1994.

Colbaugh, R., K. Glass, and H. Seraji, " Performance-based adaptive tracking control of robot manipulators," *J. Robotic Systems*, vol. 12, no. 8, pp. 517-530, 1995.

Corless, M., "Tracking controllers for uncertain systems: applications to a Manutec R3 robot," *J. Dynam. Sys. Meas. Control*, vol. 111, pp. 609-618, Dec. 1989.

Dawson, D., Z. Qu, F.L. Lewis, and J.F. Dorsey, "Robust control for the tracking of robot motion," *Int. J. Control*, vol. 52, no. 3, pp. 581-595, 1990.

Goodwin, G.C., and K.S. Sin, *Adaptive Filtering, Prediction, and Control*, Prentice-Hall, NJ, 1984.

Ioannou, P.A., and P.V. Kokotovic, "Instability analysis and improvement of robustness of adaptive control," *Automatica*, vol. 20, no. 5, pp. 583-594, 1984.

Ioannou, P.A., and J. Sun, *Robust Adaptive Control*, Prentice-Hall, NJ, 1996.

Kreisselmeier, G., and B. Anderson, "Robust model reference adaptive control," *IEEE Trans. Automat. Control*, vol. AC-31, no. 2, pp. 127-133, Feb. 1986.

Landau, Y.D., *Adaptive Control: The Model Reference Approach*, Marcel Dekker, Inc., 1979.

Lewis, F.L., *Applied Optimal Control and Estimation*, Prentice Hall, New Jersey, 1992.

Lewis, F.L., C.T. Abdallah, and D.M. Dawson, *Control of Robot Manipulators*, Macmillan, New York, 1993.

Lewis, F.L., M. Fitzgerald, and K. Liu "Robotics," in *The Computer Science and Engineering Handbook*, ed. Allen B. Tucker, Jr., Chapter 33, CRC Press, 1997.

MATLAB version 4.2, July 1994, The Mathworks, Inc., 24 Prime Park Way, Natick, MA 01760, USA.

Narendra, K.S., and A.M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Trans. Automat. Control*, vol. AC-32, no. 2, pp. 134-145, Feb. 1987.

Narendra, K.S., and A.M. Annaswamy, *Stable Adaptive Systems*, Prentice-Hall, NJ, 1989.

Sadegh, N., R. Horowitz, W. Kao, and M. Tomizuka, "A unified approach to the design of adaptive and repetitive controllers for robot manipulators," *Trans. ASME*, vol. 112, pp. 618-629, Dec. 1990.

Sastry, S., and M. Bodson, *Adaptive Control*, Prentice-Hall, NJ, 1989.

Slotine, J.-J.E., "The robust control of robot manipulators," *Int. J. Robotics Research*, vol. 4, no. 4, pp. 49-64, 1985.

Slotine, J.-J., "Putting physics in control: the example of robotics," *Control Systems Magazine*, vol. 8., pp. 12-15, Dec. 1988.

Slotine, J.-J.E., and W. Li, Applied Nonlinear Control, Prentice-Hall, New Jersey, 1991.

Spong, M.W., J.S. Thorp, and J.M. Kleinwaks, "Robust microprocessor control of robot manipulators," *Automatica*, vol. 23, no. 3, pp. 373-379, 1987.

Spong, M.W., and M. Vidyasagar, *Robot Dynamics and Control*, Wiley, New York, 1989.