Chapter 7

Complementary Material

7.2.3 Direct MRAC

***Proof of Theorem* 7.2.2.** We prove Theorem 7.2.2 in three steps.

Step 1. Show that $\theta(k)$ and $\varepsilon(k)$ are bounded Let us first rewrite the update equations in (7.32) as

$$\tilde{\theta}(k+1) = \tilde{\theta}(k) + \gamma\varepsilon(k)\phi_p(k) + \zeta(k), \tag{7C.1}$$

where $\tilde{\theta}(k) \triangleq \theta(k) - \theta^*$,

$$\zeta(k) \triangleq \begin{cases} 0_{2n} & \text{if } \left(\theta_{2n}(k) + \gamma\varepsilon(k)y_p(k)\right)\text{sgn}\left(c_0^*\right) \geq \beta_0, \\ \left[0_{2n-1}^T,\ \beta_0\text{sgn}\left(c_0^*\right) - \theta_{2n}(k) - \gamma\varepsilon(k)y_p(k)\right]^T & \text{otherwise,} \end{cases}$$

and 0_i denotes the zero vector of dimension i. Consider the function

$$V(k) = \tilde{\theta}^T(k)\tilde{\theta}(k). \tag{7C.2}$$

Using (7C.1), (7C.2) we get

$$\begin{aligned} V(k+1) - V(k) = \gamma^2\varepsilon^2(k)\phi_p^T(k)\phi_p(k) + 2\gamma\varepsilon(k)\tilde{\theta}^T(k)\phi_p(k) \\ + 2\tilde{\theta}^T(k+1)\zeta(k) - \zeta_n^2(k). \end{aligned} \tag{7C.3}$$

From (7C.1) we have

$$\tilde{\theta}^T(k+1)\zeta(k) = \tilde{\theta}_{2n}(k+1)\zeta_{2n}(k) \leq 0. \tag{7C.4}$$

Substituting (7C.4) into (7C.3), we obtain

$$V(k+1) - V(k) \leq \gamma^2\varepsilon^2(k)\phi_p^T(k)\phi_p(k) + 2\gamma\varepsilon(k)\tilde{\theta}^T(k)\phi_p(k) - \zeta_n^2(k). \tag{7C.5}$$

Using (7.30) and (7.32), we have

$$\varepsilon(k)m_s^2(k) = -\tilde{\theta}^T(k)\phi_p(k). \tag{7C.6}$$

Substituting (7C.6) into (7C.5), we get

$$\begin{aligned} V(k+1)-V(k) &\le \frac{\gamma^2\left(\tilde{\theta}^T(k)\phi_p(k)\right)^2\phi_p^T(k)\phi_p(k)}{m_s^4(k)} - 2\gamma\frac{\left(\tilde{\theta}^T(k)\phi_p(k)\right)^2}{m_s^2(k)} - \zeta_n^2(k) \\ &\le \frac{\gamma(\gamma-2)\left(\tilde{\theta}^T(k)\phi_p(k)\right)^2}{m_s^2(k)} - \zeta_n^2(k) \le 0 \end{aligned} \tag{7C.7}$$

since $0<\gamma<2$. From (7C.7) we see that $\tilde{\theta}(k)\in\ell_\infty$ and hence $\theta(k)\in\ell_\infty$ by definition. Furthermore, using (7C.6), we establish that $\varepsilon(k)\in\ell_\infty$.

Step 2. Define a fictitious normalizing signal m_f bounding u_p, y_p and establish an upper bound for m_f From (7.30) and (7.31), we have

$$y_p = W_m(\mathrm{z})\left[r+\frac{1}{c_0^*}\tilde{\theta}^T\omega\right]. \tag{7C.8}$$

Since $y_p = G_p(\mathrm{z})u_p$ and $Z_p(\mathrm{z})$ is Hurwitz, we can rewrite (7C.8) as

$$u_p = G_p^{-1}(\mathrm{z})W_m(\mathrm{z})\left[r+\frac{1}{c_0^*}\tilde{\theta}^T\omega\right], \tag{7C.9}$$

where $G_p^{-1}(\mathrm{z})W_m(\mathrm{z})$ is biproper because of assumption M2 (stated in section 7.2.2). Let us define a fictitious normalizing signal m_f as

$$m_f^2(k) \triangleq 1+\left\|u_p(k-1)\right\|^2+\left\|y_p(k-1)\right\|^2, \tag{7C.10}$$

where $\|\cdot\|$ denotes the $\ell_{2\delta}$ norm for some $0<\delta\le 1$ for which $G_p^{-1}(\sqrt{\delta}\mathrm{z}), W_m(\sqrt{\delta}\mathrm{z})$ have stable poles. Applying Lemma A.12.33 to (7C.8), (7C.9), together with the fact that $r\in\ell_\infty$, we get

$$\left\|u_p\right\|\le c+c\left\|\tilde{\theta}^T\omega\right\|, \qquad \left\|y_p\right\|\le c+c\left\|\tilde{\theta}^T\omega\right\|, \tag{7C.11}$$

where c denotes any finite constant. Substituting (7C.11) into (7C.10), we obtain

$$m_f(k)\le c+c\left\|\tilde{\theta}^T(k-1)\omega(k-1)\right\|. \tag{7C.12}$$

Step 3. Establish boundedness of all the signals and convergence of $e_1(k)$ to zero Defining $\omega_p \triangleq [\bar{\omega}^T, W_m^{-1}(\mathrm{z})y_p]^T$, we have

$$\tilde{\theta}^T\omega = \tilde{\theta}^T\omega_p + \tilde{c}_0\left(r-W_m^{-1}(\mathrm{z})y_p\right). \tag{7C.13}$$

Rewriting (7C.8) as

$$r = W_m^{-1}(\mathrm{z}) y_p - \frac{1}{c_0^*}\tilde{\theta}^T \omega$$

and substituting into (7C.13), we get

$$\tilde{\theta}^T \omega = \tilde{\theta}^T \omega_p - \frac{\tilde{c}_0}{c_0^*}\tilde{\theta}^T \omega,$$

i.e.,

$$\tilde{\theta}^T \omega = \frac{c_0^*}{c_0}\tilde{\theta}^T \omega_p. \tag{7C.14}$$

From Lemma A.12.35, we have

$$W_m(\mathrm{z})\left[\tilde{\theta}^T \omega_p\right] = \tilde{\theta}^T \phi_p + W_c(\mathrm{z})\left[W_b(\mathrm{z})\left[\omega_p^T\right]\Delta\tilde{\theta}\right], \tag{7C.15}$$

where $W_b(\mathrm{z}), W_c(\mathrm{z})$ are as defined in Lemma A.12.35 and, for any variable x, $\Delta x(k) \triangleq x(k+1) - x(k)$. From Lemma A.12.35, we have

$$\tilde{\theta}^T \omega_p = F_1\left[\left(\Delta\tilde{\theta}\right)^T \omega_p + \tilde{\theta}^T \Delta\omega_p + \left(\Delta\tilde{\theta}\right)^T \Delta\omega_p\right] + F\left[\tilde{\theta}^T \omega_p\right], \tag{7C.16}$$

where $F(\mathrm{z},\alpha_0) = \frac{{\alpha_0}^{n^*}}{(\mathrm{z}-1+\alpha_0)^{n^*}}$, $F_1(\mathrm{z},\alpha_0) = \frac{1-F(\mathrm{z},\alpha_0)}{\mathrm{z}-1}$ satisfy $\| F_1(\mathrm{z},\alpha_0) \|_{\infty\delta} \le \frac{c}{\alpha_0}$, $\| F(\mathrm{z},\alpha_0) W_m^{-1}(\mathrm{z}) \|_{\infty\delta} \le c\alpha_0^{n^*}$ for any $1 < \alpha_0$.

Using (7C.15) and (7C.16), we obtain

$$\tilde{\theta}^T \omega_p = F_1\left[\Delta\tilde{\theta}^T \omega_p + \tilde{\theta}^T \Delta\omega_p + \Delta\tilde{\theta}^T \Delta\omega_p\right] + FW_m^{-1}\left[\tilde{\theta}^T \phi_p + W_c(\mathrm{z})\left[W_b(\mathrm{z})\left[\omega_p^T\right]\Delta\tilde{\theta}\right]\right]. \tag{7C.17}$$

Using (7C.14), we have

$$\Delta\tilde{\theta}^T \omega_p + \tilde{\theta}^T \Delta\omega_p + \Delta\tilde{\theta}^T \Delta\omega_p = \frac{1}{c_0^*}\left(\Delta c_0\left(\tilde{\theta}^T + \Delta\tilde{\theta}^T\right)(\omega + \Delta\omega) + c_0 \Delta\tilde{\theta}^T (\omega + \Delta\omega) + c_0 \tilde{\theta}^T \Delta\omega\right). \tag{7C.18}$$

Using (7.32) and (7C.18), we can rewrite (7C.17) as

$$\begin{aligned}\tilde{\theta}^T \omega_p = {} & \frac{1}{c_0^*} F_1\left[\Delta c_0\left(\tilde{\theta}^T + \Delta\tilde{\theta}^T\right)(\omega + \Delta\omega) + c_0 \Delta\tilde{\theta}^T (\omega + \Delta\omega) + c_0 \tilde{\theta}^T \Delta\omega\right] \\ & + FW_m^{-1}\left[-\varepsilon m_s^2 + W_c(\mathrm{z})\left[W_b(\mathrm{z})\left[\omega_p^T\right]\Delta\tilde{\theta}\right]\right].\end{aligned} \tag{7C.19}$$

Due to boundedness of $\frac{1}{c_0}$, it follows from (7C.14) that

$$\left\|\tilde{\theta}^T \omega\right\| \le c\left\|\tilde{\theta}^T \omega_p\right\|. \tag{7C.20}$$

Applying Lemma A.12.33 in (7C.19) and using the fact that $\| F_1(z,\alpha_0) \|_{\infty\delta} \le \frac{c}{\alpha_0}$, we can establish that

$$\left\|\tilde{\theta}^T\omega\right\| \le \left\|\tilde{\theta}^T\omega_p\right\| \le \frac{c}{\alpha_0}\left(\left\|\Delta c_0 m_f\right\| + \left\|\Delta\tilde{\theta}^T m_f\right\| + m_f\right) + c\alpha_0^{n^*}\left(\left\|\varepsilon m_f\right\| + \left\|\Delta\tilde{\theta}^T m_f\right\|\right). \quad (7C.21)$$

Therefore, due to $\Delta c_0, \Delta\tilde{\theta} \in \ell_\infty$,

$$\left\|\tilde{\theta}^T\omega\right\| \le \frac{c}{\alpha_0} m_f + c\alpha_0^{n^*}\left\|\tilde{g} m_f\right\|, \quad (7C.22)$$

where $\tilde{g} = |\varepsilon m_s| + |\Delta\tilde{\theta}| \in \ell_2$ and $\tilde{g} \to 0$ as $k \to \infty$. From (7C.12), (7C.22) we obtain

$$m_f^2(k) \le c + \frac{c}{\alpha_0^2} m_f^2(k-1) + c\alpha_0^{2n^*}\left\|\tilde{g}(k-1)m_f(k-1)\right\|^2,$$

which for large α_0 implies that

$$m_f^2(k) \le c + c\alpha_0^{2n^*}\left\|\tilde{g}(k-1)m_f(k-1)\right\|^2$$

or

$$m_f^2(k) \le c + c\alpha_0^{2n^*}\sum_{i=0}^{k-1}\delta^{k-i}\tilde{g}^2(i)m_f^2(i).$$

Since $\delta^{k-1-i} \le 1$ for $i = 0,1,\ldots,k-1$, we have

$$m_f^2(k) \le c + c\alpha_0^{2n^*}\sum_{i=0}^{k-1}\tilde{g}^2(i)m_f^2(i).$$

Applying Lemma A.12.31(ii), we obtain

$$m_f^2(k) \le c\prod_{0\le i<k}\left(1 + c\alpha_0^{2n^*}\tilde{g}^2(i)\right).$$

Using the fact that the geometric mean is less than the arithmetic mean, we have

$$m_f^2(k) \le c\left(1 + \frac{1}{k}\sum_{i=0}^{k-1} c\alpha_0^{2n^*}\tilde{g}^2(i)\right)^k.$$

Since $\tilde{g} \in \ell_2$, we have

$$m_f^2(k) \le c\left(1 + \frac{c}{k}\right)^k \le ce^c,$$

which implies that $m_f \in \ell_\infty$ and hence, using the definition of m_f and Lemma A.12.33, that all the signals within the closed-loop system are bounded. From (7C.8) we have

$$e_1 = y_p - y_m = \frac{1}{c_0^*} W_m(\mathrm{z})\left[\tilde{\theta}^T \omega\right]. \tag{7C.23}$$

Using (7C.22), we can establish due to $m_f \in \ell_\infty$ and $\tilde{g} \in \ell_2$, $\tilde{g}(k) \to 0$ as $k \to \infty$ that

$$\limsup_{k\to\infty} \left\|\tilde{\theta}^T(k)\omega(k)\right\| \le \frac{c}{\alpha_0}.$$

Since α_0 is any arbitrary large number, the above inequality implies that

$$\limsup_{k\to\infty} \left\|\tilde{\theta}^T(k)\omega(k)\right\| = 0.$$

and hence, using (7C.23), we have that $e_1(k) \to 0$ as $k \to \infty$. □

***Proof of Theorem* 7.2.3.** We prove Theorem 7.2.3 in a similar way to Theorem 7.2.2. The boundedness of $\theta(k)$ and $\varepsilon(k)$ follows from Theorem 4.11.3. Next, to bound u_p, y_p, we define a fictitious normalizing signal

$$m_f^2(k) \triangleq 1 + \left\|u_p(k-1)\right\|^2 + \left\|y_p(k-1)\right\|^2,$$

where $\|\cdot\|$ denotes the $\ell_{2\delta}$ norm for some $0 < \delta \le 1$ for which $G_p^{-1}(\sqrt{\delta}\mathrm{z}), W_m(\sqrt{\delta}\mathrm{z})$ have stable poles. To find an upper bound for m_f, we express the plant input and output in terms of the error parameter term $\tilde{\theta}^T\omega$. Consider the SPM (7.30), which is valid for $\eta = 0$. For $\eta \neq 0$, we rewrite (7.33) as

$$y_p - \eta = G_p(\mathrm{z})u_p. \tag{7C.24}$$

The effect of η can be included in (7.30) as follows:

$$z(k) = \theta^{*T}\left(\phi_p(k) - \phi_{p\eta}(k)\right), \tag{7C.25}$$

where

$$\phi_{p\eta} \triangleq \left[0_{n-1}^T,\ \frac{W_m(\mathrm{z})\alpha^T(\mathrm{z})}{\Lambda(\mathrm{z})}\eta,\ W_m(\mathrm{z})\eta,\ \eta\right]^T.$$

Using (7C.25) and (7.38), we have

$$\varepsilon(k)m_s^2(k) = -\tilde{\theta}^T(k)\phi_p(k) - \theta^{*T}\phi_{p\eta}(k). \tag{7C.26}$$

From (7C.25) and (7.37) we have

$$y_p = W_m(\mathrm{z})\left[r + \frac{1}{c_0^*}\tilde{\theta}^T\omega\right] + \eta_y, \tag{7C.27}$$

where $\eta_y \triangleq \frac{1}{c_0^*}\theta^{*T}\phi_{p\eta} = (1+W_m(\mathrm{z})\frac{\theta_3^*\Lambda(\mathrm{z})+\theta_2^{*T}\alpha(\mathrm{z})}{c_0^*\Lambda(\mathrm{z})})\eta$. Since $Z_p(\mathrm{z})$ is Hurwitz, using (7C.24) we have

$$u_p = G_p^{-1}(\mathrm{z})W_m(\mathrm{z})\left[r+\frac{1}{c_0^*}\tilde{\theta}^T\omega\right]+\eta_u, \tag{7C.28}$$

where $\eta_u \triangleq G_p^{-1}(\mathrm{z})[\eta_y-\eta]$. Applying Lemma A.12.33 to (7C.27), (7C.28) and using stability of $W_m(\mathrm{z}), G_p^{-1}(\mathrm{z})W_m(\mathrm{z})$, we obtain

$$\|u_p\| \le c + c\|\tilde{\theta}^T\omega\| + \|\eta_u\|, \quad \|y_p\| \le c + c\|\tilde{\theta}^T\omega\| + \|\eta_y\|. \tag{7C.29}$$

Using the expressions for η_u and η_y, we have

$$\|\eta_y\| \le \left\|1+W_m(\mathrm{z})\frac{\theta_3^*\Lambda(\mathrm{z})+\theta_2^{*T}\alpha(\mathrm{z})}{c_0^*\Lambda(\mathrm{z})}\right\|_{\infty\delta}\|\eta\|,$$
$$\|\eta_u\| \le \left\|G_p^{-1}(\mathrm{z})W_m(\mathrm{z})\frac{\theta_3^*\Lambda(\mathrm{z})+\theta_2^{*T}\alpha(\mathrm{z})}{c_0^*\Lambda(\mathrm{z})}\right\|_{\infty\delta}\|\eta\|,$$

where

$$\|\eta\| \le \mu\left(\|\Delta_1(\mathrm{z})\|_{\infty\delta}\|u_p\| + \|\Delta_2(\mathrm{z})\|_{\infty\delta}\|y_p\|\right) + d_0,$$

which implies that

$$\|\eta_y\| \le \Delta_\infty \mu m_f + d_0, \qquad \|\eta_u\| \le \Delta_\infty \mu m_f + d_0 \tag{7C.30}$$

where

$$\Delta_\infty \triangleq \max\left\{\left\|1+W_m(\mathrm{z})\frac{\theta_3^*\Lambda(\mathrm{z})+\theta_2^{*T}\alpha(\mathrm{z})}{c_0^*\Lambda(\mathrm{z})}\right\|_{\infty\delta}, \left\|G_p^{-1}(\mathrm{z})W_m(\mathrm{z})\frac{\theta_3^*\Lambda(\mathrm{z})+\theta_2^{*T}\alpha(\mathrm{z})}{c_0^*\Lambda(\mathrm{z})}\right\|_{\infty\delta}\right\} \max\left\{\|\Delta_1(\mathrm{z})\|_{\infty\delta}, \|\Delta_2(\mathrm{z})\|_{\infty\delta}\right\}.$$

From (7C.29) and (7C.30), we obtain

$$m_f^2 \le c + c\|\tilde{\theta}^T(k-1)\omega(k-1)\|^2 + \Delta_\infty^2\mu^2 m_f^2,$$

which can be rewritten for $\mu < 1/\Delta_\infty$ as

$$m_f^2 \le c + c\|\tilde{\theta}^T(k-1)\omega(k-1)\|^2, \tag{7C.31}$$

noting that c denotes a generic positive constant. Similarly to the proof of Theorem 7.2.2, defining $\omega_p \triangleq [\bar{\omega}^T, W_m^{-1}(\mathrm{z})y_p]^T$ and rewriting (7C.27) as

$$r = W_m^{-1}(\mathrm{z})y_p - \frac{1}{c_0^*}\tilde{\theta}^T\omega - W_m^{-1}(\mathrm{z})\eta_y,$$

we obtain

$$\tilde{\theta}^T\omega = \frac{c_0^*}{c_0}\tilde{\theta}^T\omega_p - \frac{c_0^*}{c_0}\tilde{c}_0 W_m^{-1}(\mathrm{z})\eta_y. \tag{7C.32}$$

Applying the same arguments based on Lemma A.12.35 in the proof of Theorem 7.2.2 to $W_m(\mathrm{z})[\tilde{\theta}^T\omega_p]$, we have

$$\tilde{\theta}^T\omega_p = F_1\left[\Delta\tilde{\theta}^T\omega_p + \tilde{\theta}^T\Delta\omega_p + \Delta\tilde{\theta}^T\Delta\omega_p\right] + FW_m^{-1}\left[\tilde{\theta}^T\phi_p + W_c(\mathrm{z})\left[W_b(\mathrm{z})\left[\omega_p^T\right]\Delta\tilde{\theta}\right]\right], \tag{7C.33}$$

where $W_b(\mathrm{z}), W_c(\mathrm{z})$ are as defined in Lemma A.12.35, $\Delta x(k) \triangleq x(k+1) - x(k)$ for any variable x, and $F(\mathrm{z},\alpha_0) = \frac{{\alpha_0}^{n^*}}{(\mathrm{z}-1+\alpha_0)^{n^*}}$, $F_1(\mathrm{z},\alpha_0) = \frac{1-F(\mathrm{z},\alpha_0)}{\mathrm{z}-1}$ satisfy $\| F_1(\mathrm{z},\alpha_0)\|_{\infty\delta} \le \frac{c}{\alpha_0}$, $\| F(\mathrm{z},\alpha_0)W_m^{-1}(\mathrm{z})\|_{\infty\delta} \le c\alpha_0^{n^*}$ for any $\alpha_0 > 1$. Using (7C.32), we have

$$\begin{aligned}\Delta\tilde{\theta}^T\omega_p + \tilde{\theta}^T\Delta\omega_p + \Delta\tilde{\theta}^T\Delta\omega_p = {} & \frac{1}{c_0^*}\left(\Delta c_0\left(\tilde{\theta}^T + \Delta\tilde{\theta}^T\right)(\omega+\Delta\omega) + c_0\Delta\tilde{\theta}^T(\omega+\Delta\omega) + c_0\tilde{\theta}^T\Delta\omega\right) \\ & + \Delta c_0\overline{\eta}_y + c_0\Delta\overline{\eta}_y + \Delta c_0\Delta\overline{\eta}_y,\end{aligned} \tag{7C.34}$$

where $\overline{\eta}_y \triangleq W_m^{-1}(\mathrm{z})\eta_y$. Using (7.37)–(7.39) and (7C.34), we can rewrite (7C.33) as

$$\begin{aligned}\tilde{\theta}^T\omega_p = {} & \frac{1}{c_0^*}F_1\left[\Delta c_0\left(\tilde{\theta}^T + \Delta\tilde{\theta}^T\right)(\omega+\Delta\omega) + c_0\Delta\tilde{\theta}^T(\omega+\Delta\omega) + c_0\tilde{\theta}^T\Delta\omega\right] \\ & + F_1\left[\Delta c_0\overline{\eta}_y + \tilde{c}_0\Delta\overline{\eta}_y + \Delta c_0\Delta\overline{\eta}_y\right] + FW_m^{-1}\left[-\varepsilon m_s^2 + W_c(\mathrm{z})\left[W_b(\mathrm{z})\left[\omega_p^T\right]\Delta\tilde{\theta}\right]\right].\end{aligned} \tag{7C.35}$$

Due to boundedness of $\frac{1}{c_0}$, it follows from (7C.32) that

$$\left\|\tilde{\theta}^T\omega\right\| \le c\left\|\tilde{\theta}^T\omega_p\right\| + c\left\|\overline{\eta}_y\right\|. \tag{7C.36}$$

Applying Lemma A.12.33 in (7C.35) and using the fact that $\left\|F_1(\mathrm{z},\alpha_0)\right\|_{\infty\delta} \le \frac{c}{\alpha_0}$ and the boundedness of $\theta(k)$, we can establish that

$$\left\|\tilde{\theta}^T\omega\right\| \le \left\|\tilde{\theta}^T\omega_p\right\| + c\left\|\overline{\eta}_y\right\| \le \frac{c}{\alpha_0}\left(\left\|\Delta c_0 m_f\right\| + \left\|\Delta\tilde{\theta}^T m_f\right\| + m_f\right) + c\alpha_0^{n^*}\left(\left\|\varepsilon m_s^2\right\| + \left\|\Delta\tilde{\theta}^T m_f\right\|\right) + c\left\|\overline{\eta}_y\right\|. \tag{7C.37}$$

Therefore, using $\Delta c_0, \Delta\tilde{\theta} \in \ell_\infty$ together with the definition of $\overline{\eta}_y$ and (7C.30), we obtain

$$\left\|\tilde{\theta}^T\omega\right\| \le c\left(\frac{1}{\alpha_0} + \mu\right)m_f + c\alpha_0^{n^*}\left\|\tilde{g}m_f\right\| + cd_0, \tag{7C.38}$$

where from Theorem 4.11.3 and (7.40) we have $\tilde{g} \triangleq |\varepsilon m_s| + |\Delta\tilde{\theta}| \in \mathcal{S}(\mu^2 + d_0^2)$. From (7C.31) and (7C.38), we obtain

$$m_f^2(k) \le c + c\left(\frac{1}{\alpha_0^2} + \mu^2\right) m_f^2(k-1) + c\alpha_0^{2n^*} \left\|\tilde{g}(k-1) m_f(k-1)\right\|^2,$$

which for large α_0 implies that

$$m_f^2(k) \le c + c\alpha_0^{2n^*} \left\|\tilde{g}(k-1) m_f(k-1)\right\|^2 + c\mu^2 m_f^2(k-1)$$

or

$$m_f^2(k) \le c + c\alpha_0^{2n^*} \sum_{i=0}^{k-1} \delta^{k-i} \tilde{g}^2(i) m_f^2(i) + c\mu^2 m_f^2(k-1).$$

Defining $\tilde{g}_2(i) = (\tilde{g}^2(i) + \mu^2)^{1/2}\ \forall i$ and noting that $\tilde{g}_2 \in \mathcal{S}(\mu^2 + d_0^2)$ since $\tilde{g} \in \mathcal{S}(\mu^2 + d_0^2)$, we can rewrite this inequality as

$$m_f^2(k) \le c + c\alpha_0^{2n^*} \sum_{i=0}^{k-1} \delta^{k-i} \tilde{g}_2^2(i) m_f^2(i).$$

Applying Lemma A.12.31, we obtain

$$m_f^2(k) \le c + \sum_{i=0}^{k-1} \left(c\alpha_0^{2n^*} \delta^{k-i} \tilde{g}_2^2(i) \prod_{i<j<k} \left(1 + c\alpha_0^{2n^*} \delta^{k-j} \tilde{g}_2^2(j)\right)\right).$$

Using the fact that the geometric mean is less than arithmetic mean, we have

$$m_f^2(k) \le c + \sum_{i=0}^{k-1} \left(c\alpha_0^{2n^*} \delta^{k-i} \tilde{g}_2^2(i) \left(1 + \frac{1}{k-i-1} \left(\sum_{j=i+1}^{k-1} c\alpha_0^{2n^*} \delta^{k-j} \tilde{g}_2^2(j)\right)\right)^{k-i-1}\right).$$

Since $\tilde{g}_2 \in \mathcal{S}(\mu^2 + d_0^2)$ and $|\delta| < 1$, we have

$$m_f^2(k) \le c + \sum_{i=0}^{k-1} \left(c\alpha_0^{2n^*} \delta^{k-i} \tilde{g}_2^2(i) \left(1 + \mu^2 + d_0^2 + \frac{c}{k-i-1}\right)^{k-i-1}\right) \le \sum_{i=0}^{k-1} \left(c\alpha_0^{2n^*} \tilde{g}_2^2(i) \delta^{k-i} \left(1 + \mu^2 + d_0^2\right)^{k-i-1}\right).$$

Therefore, again since $\tilde{g}_2 \in \mathcal{S}(\mu^2 + d_0^2)$, for $\delta < 1/(1 + \mu^2 + d_0^2)$ we conclude that $m_f \in \ell_\infty$ and hence, using the definition of m_f and Lemma A.12.33, that all the signals within the closed-loop system are bounded. From (7C.27) we have

$$e_1 = y_p - y_m = \frac{1}{c_0^*} W_m(z) \left[\tilde{\theta}^T \omega\right] + \eta_y. \tag{7C.39}$$

In a way similar to the proof of Theorem 7.2.2, using (7C.30), (7C.38), (7C.39) together with $m_f \in \ell_\infty$, $\tilde{g} \in \mathcal{S}(\mu^2 + d_0^2)$, and Lemma A.12.33, we establish the convergence result of the theorem, i.e.,

$$\lim_{\bar{N}\to\infty} \sup_{0<N\le\bar{N}} \frac{1}{N}\sum_{k=0}^{N-1} |e_1(k)| \le c\left(\mu + \sqrt{\varepsilon_0} + d_0\right),$$

where $\varepsilon_0 = 1/\alpha_0$, which can be taken as an arbitrarily small positive number. □

7.2.4 Indirect MRAC

***Proofs of Theorems* 7.2.4 *and* 7.2.5.** We prove Theorems 7.2.4 and 7.2.5 following the same methodology as in Theorems 7.2.2 and 7.2.4. We mainly consider Theorem 7.2.5, and treat Theorem 7.2.4 as a special case of Theorem 7.2.5, where $\mu = 0$ and $d_0 = 0$.

Because of Theorem 4.11.3 and (7.46), we have that $\theta_p, 1/\hat{k}_p, \varepsilon, \varepsilon m_s \in \ell_\infty$ and $\Delta\theta_p$, $\varepsilon m_s \in \mathcal{S}(\mu^2 + d_0^2)$ ($\Delta\theta_p, \varepsilon m_s \in \ell_2$ for the ideal case with $\mu = 0$ and $d_0 = 0$). This also implies that $\theta \in \ell_\infty$ and $\Delta\theta \in \mathcal{S}(\mu^2 + d_0^2)$ ($\Delta\theta \in \ell_2$ for the ideal case with $\mu = 0$ and $d_0 = 0$).

Next, to bound u_p, y_p, we define a fictitious normalizing signal

$$m_f^2(k) \triangleq 1 + \left\|u_p(k-1)\right\|^2 + \left\|y_p(k-1)\right\|^2,$$

where $\|\cdot\|$ denotes the $\ell_{2\delta}$ norm for some $0 < \delta \le 1$ for which $G_p^{-1}(\sqrt{\delta}z), W_m(\sqrt{\delta}z)$ have stable poles. To find an upper bound for m_f, we express the plant input and output in terms of the error parameter term $\tilde{\theta}^T\omega$. Rewriting (7.33) as

$$y_p - \eta = G_p(z)u_p, \tag{7C.40}$$

the effect of η can be included in the SPM (7.43) as follows:

$$z(k) = \theta_p^{*T}\left(\phi(k) + \phi_\eta(k)\right) + \eta_z(k), \tag{7C.41}$$

where

$$\phi_\eta \triangleq \left[0_{m+1}^T,\ \frac{\alpha_{n-1}^T(z)}{\Lambda_p(z)}\eta\right]^T, \qquad \eta_z = \frac{z^n}{\Lambda_p(z)}\eta.$$

Note here that, as established in the proof of Theorem 7.2.3,

$$\|\eta\| \le \mu\left(\left\|\Delta_1(z)\right\|_{\infty\delta}\left\|u_p\right\| + \left\|\Delta_2(z)\right\|_{\infty\delta}\left\|y_p\right\|\right) + d_0,$$

and hence

$$\|\eta_z\| \le \mu \left\| \frac{\mathrm{z}^n}{\Lambda_p(\mathrm{z})} \right\|_{\infty\delta} \left(\|\Delta_1(\mathrm{z})\|_{\infty\delta} \|u_p\| + \|\Delta_2(\mathrm{z})\|_{\infty\delta} \|y_p\| \right) + d_0 \left\| \frac{\mathrm{z}^n}{\Lambda_p(\mathrm{z})} \right\|_{\infty\delta}.$$

Using (7C.41) and (7.45), we have

$$\varepsilon(k) m_s^2(k) = -\tilde{\theta}_p^T(k)\phi(k) + \theta_p^{*T}\phi_\eta(k) + \eta_z(k). \tag{7C.42}$$

Noting that θ^*, ω are the same as in the direct MRAC case, from (7C.27) and (7C.28) we have

$$y_p = W_m(\mathrm{z}) \left[r + \frac{1}{c_0^*} \tilde{\theta}^T \omega \right] + \eta_y, \tag{7C.43}$$

$$u_p = G_p^{-1}(\mathrm{z}) W_m(\mathrm{z}) \left[r + \frac{1}{c_0^*} \tilde{\theta}^T \omega \right] + \eta_u, \tag{7C.44}$$

where η_u, η_y are as defined in the proof of Theorem 7.2.3 and hence satisfy

$$\|u_p\| \le c + c\|\tilde{\theta}^T \omega\| + \|\eta_u\|, \qquad \|y_p\| \le c + c\|\tilde{\theta}^T \omega\| + \|\eta_y\|, \tag{7C.45}$$

$$\|\eta_y\| \le \Delta_\infty \mu m_f + d_0, \qquad \|\eta_u\| \le \Delta_\infty \mu m_f + d_0, \tag{7C.46}$$

where

$$\Delta_\infty \triangleq \max \left\{ \left\| 1 + W_m(\mathrm{z}) \frac{\theta_3^* \Lambda(\mathrm{z}) + \theta_2^{*T} \alpha(\mathrm{z})}{c_0^* \Lambda(\mathrm{z})} \right\|_{\infty\delta}, \right.$$
$$\left. \left\| G_p^{-1}(\mathrm{z}) W_m(\mathrm{z}) \frac{\theta_3^* \Lambda(\mathrm{z}) + \theta_2^{*T} \alpha(\mathrm{z})}{c_0^* \Lambda(\mathrm{z})} \right\|_{\infty\delta} \right\} \max \left\{ \|\Delta_1(\mathrm{z})\|_{\infty\delta}, \|\Delta_2(\mathrm{z})\|_{\infty\delta} \right\}.$$

From (7C.29) and (7C.30), we obtain

$$m_f^2 \le c + c\|\tilde{\theta}^T(k-1)\omega(k-1)\|^2 + \Delta_\infty^2 \mu^2 m_f^2,$$

which can be rewritten for $\mu < 1/\Delta_\infty$ as

$$m_f^2 \le c + c\|\tilde{\theta}^T(k-1)\omega(k-1)\|^2, \tag{7C.47}$$

noting that c denotes a generic positive constant.

Next, following the steps in the proof of Theorem 5.6.1 (for continuous-time indirect MRAC), with the only difference that the differential operator s and the continuous-time variable t are respectively replaced with the shift operator z and the discrete-time index k, we obtain

$$\tilde{\theta}^T W_m(\mathrm{z}) \bar{\omega}_1 = \bar{e}_1, \tag{7C.48}$$

where

$$\bar{\omega}_1 \triangleq \left[\omega_1{}^T, \omega_2{}^T, y_p, W_m^{-1}(\mathrm{z}) y_p\right]^T,$$
$$\bar{e}_1 \triangleq \frac{1}{\hat{k}_p}\left(\hat{Q}(\mathrm{z},k)\bullet\hat{R}_p(\mathrm{z},k)\frac{W_m(\mathrm{z})}{\Lambda(\mathrm{z})} y_p - \hat{Q}(\mathrm{z},k)\bullet\hat{\bar{Z}}_p(\mathrm{z},k)\frac{W_m(\mathrm{z})}{\Lambda(\mathrm{z})} u_p\right),$$

and $\bar{e}_1$ satisfies

$$\left\|\bar{e}_1\right\| \le c\left\|\varepsilon m_s m_f\right\| + c\left\|\Delta\tilde{\theta} m_f\right\|, \tag{7C.49}$$

where c denotes a generic positive constant.

From (7C.43), we have

$$e_1 = y_p - y_m = W_m\left[\frac{1}{c_0^*}\tilde{\theta}^T\omega\right] + \eta_y. \tag{7C.50}$$

Therefore,

$$\tilde{\theta}^T\bar{\omega}_1 - \tilde{\theta}^T\omega = \tilde{\theta}^T\left(\bar{\omega}_1 - \omega\right) = \left(c_0 - c_0^*\right)(W_m^{-1} y_p - r) = \frac{c_0 - c_0^*}{c_0^*}\tilde{\theta}^T\omega + \left(c_0 - c_0^*\right)\bar{\eta}_y,$$

where $\bar{\eta}_y \triangleq W_m^{-1}(\mathrm{z})\eta_y$; i.e., $\tilde{\theta}^T\bar{\omega}_1 = \frac{c_0}{c_0^*}\tilde{\theta}^T\omega + \left(c_0 - c_0^*\right)\bar{\eta}_y$. Hence, using Lemma A.12.36, we obtain

$$\tilde{\theta}^T\omega = \frac{c_0^*}{c_0}\left(F_1(\mathrm{z},\alpha_0)\left[\Delta\theta^T\bar{\omega}_1 + \tilde{\theta}^T\Delta\bar{\omega}_1 + \Delta\theta^T\Delta\bar{\omega}_1\right] + F(\mathrm{z},\alpha_0)\left[\tilde{\theta}^T\bar{\omega}_1\right] - \left(c_0 - c_0^*\right)\bar{\eta}_y\right), \tag{7C.51}$$

where $\Delta x(k) \triangleq x(k+1) - x(k)$ for any variable x and $F(\mathrm{z},\alpha_0) = \frac{\alpha_0{}^{n^*}}{(\mathrm{z}-1+\alpha_0)^{n^*}}$, $F_1(\mathrm{z},\alpha_0) = \frac{1-F(\mathrm{z},\alpha_0)}{\mathrm{z}-1}$ satisfy $\|F_1(\mathrm{z},\alpha_0)\|_{\infty\delta} \le \frac{c}{\alpha_0}$, $\|F(\mathrm{z},\alpha_0)W_m^{-1}(\mathrm{z})\|_{\infty\delta} \le c\alpha_0^{n^*}$ for any $\alpha_0 > 1$. Applying Lemma A.12.35, we obtain

$$\tilde{\theta}^T\bar{\omega}_1 = W_m^{-1}\left[\tilde{\theta}^T W_m\left[\bar{\omega}_1\right] + W_c\left[W_b\left[\bar{\omega}_1^T\right]\Delta\theta\right]\right] = W_m^{-1}\left[\bar{e}_1 + W_c\left[W_b\left[\bar{\omega}_1^T\right]\Delta\theta\right]\right], \tag{7C.52}$$

where $W_b(\mathrm{z}), W_c(\mathrm{z})$ are as defined in Lemma A.12.35. Substituting into (7C.51), we obtain

$$\tilde{\theta}^T\omega = \frac{c_0^*}{c_0}\left(F_1(\mathrm{z},\alpha_0)\left[\Delta\theta^T\bar{\omega}_1 + \tilde{\theta}^T\Delta\bar{\omega}_1 + \Delta\theta^T\Delta\bar{\omega}_1\right] + F(\mathrm{z},\alpha_0)W_m^{-1}(\mathrm{z})\left[\bar{e}_1 + W_c(\mathrm{z})\left[W_b(\mathrm{z})\left[\bar{\omega}_1^T\right]\Delta\theta\right]\right] - \left(c_0 - c_0^*\right)\bar{\eta}_y\right).$$

Since c_0 is bounded from below, i.e., using Lemma A.12.33 and the definition of m_f, we obtain

$$\left\|\tilde{\theta}^T\omega\right\| \le \frac{c}{\alpha_0}\left\|\Delta\tilde{\theta} m_f\right\| + \frac{c}{\alpha_0} m_f + c\alpha_0^{n^*}\left(\left\|\bar{e}_1\right\| + \left\|\Delta\tilde{\theta} m_f\right\|\right) + c\left\|\bar{\eta}_y\right\|. \tag{7C.53}$$

Using (7C.49) and (7C.53) together with $\theta,\theta_p,1/\hat{k}_p,\varepsilon,\varepsilon m_s \in \ell_\infty$, $\Delta\theta,\ \Delta\theta_p,\ \varepsilon m_s \in \mathcal{S}(\mu^2+d_0^2)$ ($\Delta\theta,\Delta\theta_p,\varepsilon m_s \in \ell_2$ for $\mu=0$ and $d_0=0$), the definition of $\overline{\eta}_y$, and (7C.46), we obtain

$$\left\|\tilde{\theta}^T\omega\right\| \le c\left(\frac{1}{\alpha_0}+\mu\right)m_f + c\alpha_0^{n^*}\left\|\tilde{g}m_f\right\| + cd_0, \tag{7C.54}$$

where $\tilde{g} \triangleq |\varepsilon m_s|+|\Delta\theta|+|\Delta\theta_p| \in \mathcal{S}(\mu^2+d_0^2)$ ($\tilde{g}\in\ell_2$ and $\tilde{g}\to 0$ as $k\to\infty$ for $\mu=0$ and $d_0=0$).

Note here that (7C.31) and (7C.38) are exactly the same as (7C.47) and (7C.54), respectively. Hence using these equations and exactly the same steps as in the proof of Theorem 7.2.3, we establish that $m_f \in \ell_\infty$, and hence all the signals within the closed-loop system are bounded as well as

$$\lim_{\bar{N}\to\infty}\sup_{0<N\le\bar{N}}\frac{1}{N}\sum_{k=0}^{N-1}|e_1(k)| \le c\left(\mu+\sqrt{\varepsilon_0}+d_0\right),$$

where $\varepsilon_0 = 1/\alpha_0$, which can be taken as an arbitrarily small positive number.

In the ideal case, where $\mu=0$ and $d_0=0$, using (7C.54), $m_f\in\ell_\infty$, $\tilde{g}\in\ell_2$, $\tilde{g}(k)\to 0$ as $k\to\infty$, and choosing α_0 arbitrarily large, we have that

$$\limsup_{k\to\infty}\left\|\tilde{\theta}^T(k)\omega(k)\right\| = 0.$$

This, together with (7C.50), implies that $e_1(k)\to 0$ as $k\to\infty$. □

7.4 APPC

***Proof of Theorem* 7.4.1.** The steps of the proof are similar to those of the proof of Theorem 6.7.3 (continuous-time robust APPC).

Step 1. Express u_p, y_p in terms of the estimation error We rewrite the control law (7.85) and the normalized estimation error as

$$\begin{aligned}\hat{L}Q_m\frac{1}{\Lambda}u_p &= -\hat{P}\frac{1}{\Lambda}(y_p-y_m),\\ \varepsilon m_s^2 &= z-\theta_p^T\phi = \hat{R}_p\frac{1}{\Lambda_p}y_p - \hat{Z}_p\frac{1}{\Lambda_p}u_p.\end{aligned} \tag{7C.55}$$

The proof is simplified if without loss of generality we design

$$\Lambda(z) = \Lambda_p(z)\Lambda_q(z),$$

where $\Lambda_q(z)$ is an arbitrary monic Hurwitz polynomial of degree $q-1$ if $q\ge 2$ and $\Lambda_q(z)=1$ for $q<2$. We should point out that the same analysis can also be carried out

with Λ, Λ_p being Hurwitz but otherwise arbitrary, at the expense of some additional algebra. Let us define

$$u_f \triangleq \frac{1}{\Lambda} u_p, \qquad y_f \triangleq \frac{1}{\Lambda} y_p$$

and write (7C.55) as

$$\begin{aligned} &\hat{P} y_f + \hat{L} Q_m u_f = y_{m1}, \\ &\hat{R}_p \Lambda_q y_f - \hat{Z}_p \Lambda_q u_f = \varepsilon m_s^2, \end{aligned} \tag{7C.56}$$

where

$$y_{m1} \triangleq \hat{P} \frac{1}{\Lambda} y_m \in \ell_\infty .$$

Using the expressions

$$\begin{aligned} &\hat{R}_p(\mathrm{z})\Lambda_q(\mathrm{z}) = \mathrm{z}^{n+q-1} + \overline{\theta}_1^T \alpha_{n+q-2}(\mathrm{z}), \qquad \hat{Z}_p(\mathrm{z})\Lambda_q(\mathrm{z}) = \overline{\theta}_2^T \alpha_{n+q-2}(\mathrm{z}), \\ &\hat{P}(\mathrm{z}) = p_0 \mathrm{z}^{n+q-1} + \overline{p}^T \alpha_{n+q-2}(\mathrm{z}), \qquad \hat{L}(\mathrm{z}) Q_m(\mathrm{z}) = s^{n+q-1} + \overline{l}^T \alpha_{n+q-2}(\mathrm{z}) \end{aligned}$$

in (7C.56), we obtain

$$\begin{aligned} y_f(k+n+q-1) &= -\overline{\theta}_1^T \alpha_{n+q-2}(\mathrm{z}) y_f(k) + \overline{\theta}_2^T \alpha_{n+q-2}(\mathrm{z}) u_f(k) + \varepsilon m_s^2(k), \\ u_f(k+n+q-1) &= (p_0 \overline{\theta}_1 - \overline{p})^T \alpha_{n+q-2}(\mathrm{z}) y_f(k) - (p_0 \overline{\theta}_2 + \overline{l})^T \alpha_{n+q-2}(\mathrm{z}) u_f(k) \\ &\quad - p_0 \varepsilon m_s^2(k) + y_{m1}(k). \end{aligned} \tag{7C.57}$$

Defining the state

$$x(k) \triangleq \left[y_f(k), y_f(k-1), \ldots, y_f(k-(n+q-2)), u_f(k), u_f(k-1), \ldots, u_f(k-(n+q-2)) \right]^T,$$

(7C.57) can be expressed in the form

$$x(k+1) = A(k)x(k) + b_1(k)\varepsilon m_s^2(k) + b_2 y_{m1}(k), \tag{7C.58}$$

where

$$A(k) = \begin{bmatrix} -\overline{\theta}_1^T & \overline{\theta}_2^T \\ I_{n+q-2} \quad 0_{(n+q-2)} & \mathrm{O}_{(n+q-2)x(n+q-1)} \\ p_o \overline{\theta}_1^T - \overline{p}^T & p_o \overline{\theta}_2^T - \overline{l}^T \\ \mathrm{O}_{(n+q-2)x(n+q-1)} & I_{n+q-2} \quad 0_{(n+q-2)} \end{bmatrix}, \quad b_1(k) = \begin{bmatrix} 1 \\ 0_{(n+q-2)} \\ -p_o \\ 0_{(n+q-2)} \end{bmatrix}, \quad b_2 = \begin{bmatrix} 0_{(n+q-1)} \\ 1 \\ 0_{(n+q-2)} \end{bmatrix}$$

and $\mathrm{O}_{(n+q-2)x(n+q-1)}$ and $0_{(n+q-2)}$ denote, respectively, an $(n+q-2)\times(n+q-1)$ matrix and an $(n+q-2)$-vector with all elements equal to zero. Since

$$u_p(k) = \Lambda u_f(k) = u_f(k+n+q-1) + \lambda^T \alpha_{n+q-2}(\mathrm{z}) u_f(k),$$
$$y_p(k) = \Lambda y_f(k) = y_f(k+n+q-1) + \lambda^T \alpha_{n+q-2}(\mathrm{z}) y_f(k),$$

where λ is the coefficient vector of $\Lambda(\mathrm{z}) - \mathrm{z}^{n+q-1}$, it follows that

$$\begin{aligned} u_p(k) &= [\underbrace{0,\ldots,0}_{n+q-1}, \underbrace{1,0,\ldots,0}_{n+q-1}] x(k+1) + [\underbrace{0,\ldots,0}_{n+q-1}, \lambda^T] x(k), \\ y_p(k) &= [\underbrace{1,0,\ldots,0}_{n+q-1}, \underbrace{0,\ldots,0}_{n+q-1}] x(k+1) + [\lambda^T, \underbrace{0,\ldots,0}_{n+q-1}] x(k). \end{aligned} \tag{7C.59}$$

Combining (7C.58) and (7C.59), we obtain

$$\begin{aligned} x(k+1) &= A(k)x(k) + b_1(k)\varepsilon m_s^2(k) + b_2 y_{m1}(k), \\ y_p(k) &= C_1^T(k)x(k) + d_1(k)\varepsilon m_s^2(k) + d_2 y_{m1}(k), \\ u_p(k) &= C_2^T(k)x(k) + d_3(k)\varepsilon m_s^2(k) + d_4 y_{m1}(k), \end{aligned} \tag{7C.60}$$

where

$$\begin{aligned} \left[C_1^T, d_1, d_2\right] &= [\underbrace{1,0,\ldots,0}_{n+q-1}, \underbrace{0,\ldots,0}_{n+q-1}] \left[A \;\; b_1 \;\; b_2\right] + [\lambda^T, \underbrace{0,\ldots,0}_{n+q+1}], \\ \left[C_2^T, d_3, d_4\right] &= [\underbrace{0,\ldots,0}_{n+q-1}, \underbrace{1,0,\ldots,0}_{n+q-1}] \left[A \;\; b_1 \;\; b_2\right] + [\underbrace{0,\ldots,0}_{n+q-1}, \lambda^T, 0, 0]. \end{aligned}$$

Step 2. Establish the e.s. property of $A(k)$ Using Theorem 4.11.3, we have that $\theta_p, \varepsilon, \varepsilon m_s \in \ell_\infty$ and $\Delta\theta_p, \varepsilon m_s \in \mathcal{S}\left(\mu^2 + d_0^2\right)$ ($\Delta\theta_p, \varepsilon m_s \in \ell_2$ for the ideal case with $\mu = 0$ and $d_0 = 0$). Consequently, we have $\| A \| \in \ell_\infty$ and $\|\Delta A\| \in \mathcal{S}(\mu^2 + d_0^2)$ ($\| \Delta A \| \in \ell_2$ for the ideal case with $\mu = 0$ and $d_0 = 0$).

The APPC law guarantees that $\det(\mathrm{z}I - A(k)) = A^*(\mathrm{z})$ for each k, where $A^*(\mathrm{z})$ is Hurwitz. Hence, because of Theorem A.12.23(i), the homogenous part of (7C.60) is e.s. in the large.

Step 3. Use the properties of l_2 norm and the B–G lemma to establish boundedness We apply Lemma A.12.33 to (7C.60). Using the e.s. property established in Step 2 and Lemma A.12.32, we have

$$\left\|x_k\right\| \le c\sqrt{\delta}^k \, |\, x(0)\, | + c\left\|(\varepsilon m_s^{\,2})_k\right\| + c,$$

where $\left\|(\cdot)_k\right\|$ denotes the $\ell_{2\delta}$ norm $\left\|(\cdot)_k\right\|_{2\delta}$ for any $\delta > \alpha^2$, where $\alpha \in (0,1)$ is the exponential convergence rate (as used in Definition A.12.15) of the homogenous part of (7C.60). Since $\frac{x}{m_s} \in \ell_\infty$, it follows that

$$\left\|x_k\right\| \le c\sqrt{\delta}^k m_s(0) + c\left\|(\varepsilon m_s^{\,2})_k\right\| + c.$$

Since $\| y_{pk} \|, \| u_{pk} \| \le c \| x_k \| + c \| (\varepsilon m_s^{\,2})_k \| + c$, it follows that

$$\left\|y_{pk}\right\|, \left\|u_{pk}\right\| \le \left\|x_k\right\| \le c\sqrt{\delta}^{k} m_s(0) + c\left\|(\varepsilon m_s^{\,2})_k\right\| + c. \tag{7C.61}$$

From (7.86) , we have $m_s^{\,2}(k) = 1 + \phi^T(k)\phi(k) + n_d(k)$ and

$$n_d(k) = \delta_0^{\,k} n_d(0) + \left\|y_{p(k-1)}\right\|_{2\delta_0}^2 + \left\|u_{p(k-1)}\right\|_{2\delta_0}^2. \tag{7C.62}$$

Since $\|(\cdot)_{(k-1)}\|_{2\delta_0} \le \|(\cdot)_{(k-1)}\|$ for $\delta_0 \le \delta$, it follows that

$$m_s^{\,2}(k) = 1 + \phi^T(k)\phi(k) + n_d(k) \le 1 + \phi^T(k)\phi(k) + \delta_0^{\,k} m_s^{\,2}(0) + \left\|y_{p(k-1)}\right\|^2 + \left\|u_{p(k-1)}\right\|^2 f \qquad \forall k \ge 0.$$

Substituting (7C.61) and noting that (7C.62) and Lemma A.12.33 imply $\phi^T(k)\phi(k) \le c n_d(k)$, we obtain

$$m_s^{\,2}(k) \le c + c\delta^k m_s^{\,2}(0) + c\left\|(\varepsilon m_s^{\,2})_k\right\|^2 \qquad \forall k \ge 0$$

or

$$m_s^{\,2}(k) \le c + c\delta^k m_s^{\,2}(0) + c\sum_{i=0}^{k-1}\left(\delta^{k-i}\varepsilon^2 m_s^2(i) m_s^2(i)\right) \qquad \forall k \ge 0.$$

Applying Lemma A.12.31, we obtain

$$\begin{aligned} m_s^{\,2}(k) &\le c + c\delta^k m_s^{\,2}(0) + c\sum_{i=0}^{k-1}\left(\prod_{i<j<k}\left(1+\delta^{k-j}\varepsilon^2 m_s^2(j)\right)\delta^{k-i}\varepsilon^2 m_s^2(i)\left(1+\delta^{i-k_1} m_s^{\,2}(0)\right)\right) \\ &= c + c\delta^k m_s^{\,2}(0)\left(1 + \sum_{i=0}^{k-1}\left(\varepsilon^2 m_s^2(i)\prod_{i<j<k}\left(1+\delta^{k-j}\varepsilon^2 m_s^2(j)\right)\right)\right) \\ &\quad + c\sum_{i=0}^{k-1}\left(\delta^{k-i}\varepsilon^2 m_s^2(i)\prod_{i<j<k}\left(1+\delta^{k-j}\varepsilon^2 m_s^2(j)\right)\right) \qquad \forall k \ge 0. \end{aligned}$$

Using the fact that geometric mean is less than arithmetic mean, we have

$$\begin{aligned} m_s^{\,2}(k) &\le c + c\delta^k m_s^{\,2}(0)\left(1 + \sum_{i=0}^{k-1}\left(\varepsilon^2 m_s^2(i)\left(1 + \frac{1}{k-i-1}\sum_{i<j<k}\delta^{k-j}\varepsilon^2 m_s^2(j)\right)^{k-i-1}\right)\right) \\ &\quad + c\sum_{i=0}^{k-1}\left(\delta^{k-i}\varepsilon^2 m_s^2(i)\left(1 + \frac{1}{k-i-1}\sum_{i<j<k}\delta^{k-j}\varepsilon^2 m_s^2(j)\right)^{k-i-1}\right) \qquad \forall k \ge 0. \end{aligned}$$

Since $\tilde{g}_2 \in \mathcal{S}(\mu^2 + d_0^2)$ and $|\delta| < 1$, for $\delta < 1/(1+\mu^2+d_0^2)$, we have

$$
\begin{aligned}
m_s{}^2(k) &\le c + c\delta^k m_s{}^2(0)\left(1+\sum_{i=0}^{k-1}\left(\varepsilon^2 m_s^2(i)\left(1+\frac{c}{k-i-1}\right)^{k-i-1}\right)\right) \\
&\quad + c\sum_{i=0}^{k-1}\left(\delta^{k-i}\varepsilon^2 m_s^2(i)\left(1+\frac{c}{k-i-1}\right)^{k-i-1}\right) \\
&\le c + c\delta^k m_s{}^2(0)\left(1+e^c\sum_{i=0}^{k-1}\left(\varepsilon^2 m_s^2(i)\right)\right) + ce^c\sum_{i=0}^{k-1}\left(\delta^{k-i}\varepsilon^2 m_s^2(i)\right) \\
&\le c + c\delta^k m_s{}^2(0)\left(1+k\left(\mu^2+d_0^2\right)\right)+c \\
&\le c \qquad \forall k \ge 0.
\end{aligned}
$$

Therefore we have m_s, and hence all the closed loop signals bounded.

Step 4. Establish bounds for the tracking error A bound for the tracking error e_1 is obtained by expressing e_1 in terms of signals that are guaranteed by the adaptive law to be of the order of the modeling error in the m.s.s. The tracking error equation is derived following the steps of the proofs of Theorems 6.3.2 and 6.7.2, replacing the differential operator s with the shift operator z and applying the discrete time counterparts of the B-G and swapping lemmas used; it has the same form as in these proofs and is given by

$$
e_1 = \frac{\Lambda(\mathrm{z})\mathrm{z}^{n-1}Q_m(\mathrm{z})}{A^*(\mathrm{z})}\varepsilon m_s^2 + \frac{\Lambda(\mathrm{z})\alpha_{n-2}^T(\mathrm{z})}{A^*(\mathrm{z})}v_0,
$$

where v_0 is the output of proper stable transfer functions whose inputs are elements of $\Delta\theta_p$ multiplied by bounded signals. Since $\theta_p, \varepsilon, \varepsilon m_s, m_s \in \ell_\infty$ and $\Delta\theta_p, \varepsilon m_s \in \mathcal{S}\left(\mu^2+d_0^2\right)$ ($\Delta\theta_p, \varepsilon m_s \in \ell_2$ for the ideal case with $\mu = 0$ and $d_0 = 0$), following exactly the same steps as in the proof of Theorem 7.2.3 (and Theorem 7.2.2), we establish the convergence results. □

7.5 Examples Using the Adaptive Control Toolbox

The discrete-time MRAC, adaptive prediction, one-step-ahead control, and APPC algorithms presented in Chapter 7 can be implemented using a set of MATLAB functions and Simulink blocks provided in the Adaptive Control Toolbox. In this section, we demonstrate use of the Adaptive Control Toolbox in various discrete-time adaptive control problems via some simulation examples.

7.5.1 MRAC

The MATLAB function **`dmrc`** can be used to simulate the MRC algorithm described in section 7.2, as demonstrated in Examples 7.5.1 and 7.5.2.

Example 7.5.1 Consider the plant

$$
y(t) - 1.9y(t-1) + 0.9y(t-2) = u(t-3) + 0.5u(t-4) + 0.25u(t-3),
$$

where the signal y is initially at rest, i.e., $y(0) = y(-1) = \cdots = 0$. Let us design a model reference controller to bring y to 1, i.e., $y_m(t) = 1\ \forall t$. We can do this by choosing a

stable reference model with a steady-state value of 1 and applying the desired reference to this reference model. Let the reference model be

$$E\left(q^{-1}\right)y_m(t) = q^{-3}\bar{H}\left(q^{-1}\right)r(t),$$

where

$$E\left(q^{-1}\right) = 1 - 1.5q^{-1} + 0.56q^{-2},$$
$$\bar{H}\left(q^{-1}\right) = 0.04 + 0.02q^{-1}.$$

Solving the matching equation

$$F\left(q^{-1}\right)A\left(q^{-1}\right) + q^{-3}G\left(q^{-1}\right) = E(q^{-1}),$$

we obtain

$$F\left(q^{-1}\right) = 1 + 0.4q^{-1} + 0.42q^{-2},$$
$$G\left(q^{-1}\right) = 0.438 - 0.378q^{-1},$$
$$\beta\left(q^{-1}\right) = F\left(q^{-1}\right)\bar{B}\left(q^{-1}\right) = 1 + 0.9q^{-1} + 0.87q^{-2} + 0.31q^{-3} + 0.105q^{-4}.$$

Hence the control law is given by

$$\begin{aligned} u(t) &= \frac{1}{\beta_0}\bar{H}\left(q^{-1}\right)r(t) - \bar{G}\left(q^{-1}\right)y(t) - q\bar{\beta}\left(q^{-1}\right)u(t-1) \\ &= 0.04r(t) + 0.02r(t-1) - 0.438y(t) + 0.378y(t-1) \\ &\quad -0.9u(t-1) - 0.87u(t-2) - 0.31u(t-3) - 0.105u(t-4). \end{aligned}$$

The plant controlled with the model reference controller above can be simulated for $t \in [1,100]$ using the following code:

```
A = [1 -1.9 0.9];
Bbar = [1 0.5 0.25];
n = 2; m = 2; d = 3;
E = [1 -1.5 0.56]; Hbar = [0.04 0.02];

t_final = 100;
t = 1:t_final;

[Cr,Cu,Cy] = dmrc(d,Bbar,A,Hbar,E);

[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
[nstatem,xm] = uarma('init',d-1,Hbar,E);  %ym(t)/r(t-1)
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
ym(1) = uarma('output',xm,1,d-1,Hbar,E);
xm = uarma('state',xm,1,d-1,Hbar,E);
for k = 1:d-1,
   r(k) = 1;
   ym(k+1) = uarma('output',xm,r(k),d-1,Hbar,E);
   xm = uarma('state',xm,r(k),d-1,Hbar,E);
```

```
end
Wr = [r(1); 0];
Wu = zeros(4,1);
Wy = [y(1);0];

for k = 1:t_final,
   r(k+d-1) = 1;
   ym(k+d) = uarma('output',xm,r(k+d-1),d-1,Hbar,E);
   xm = uarma('state',xm,r(k+d-1),d-1,Hbar,E);
   u(k) = Cr*Wr+Cu*Wu+Cy*Wy;
   y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
   x = uarma('state',x,u(k),d-1,Bbar,A);
   Wr = [r(k+1);Wr(1)];
   Wu = [u(k);Wu(1:3)];
   Wy = [y(k+1);Wy(1)];
end;
```

The responses for $r(t)=1$ and $r(t)=1+\sin(0.1t)$ are plotted in Figures 7C.1 and 7C.2. The response is smoother but the settling time is greater than those of the one-step-ahead control. ■

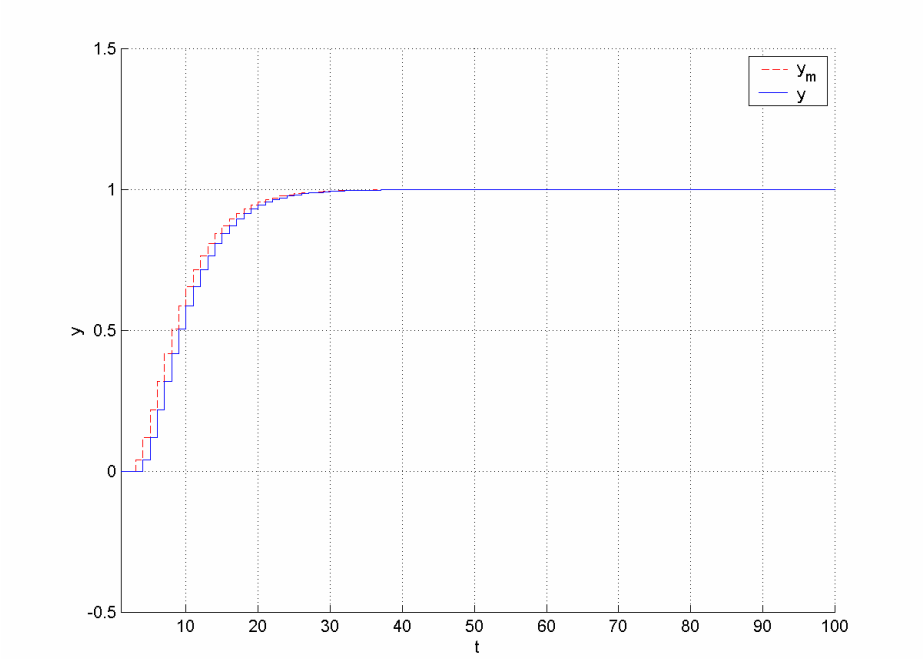

Figure 7C.1 *Time history of the output* y *for* $r(t)=1$ *in Example* 7.5.1.

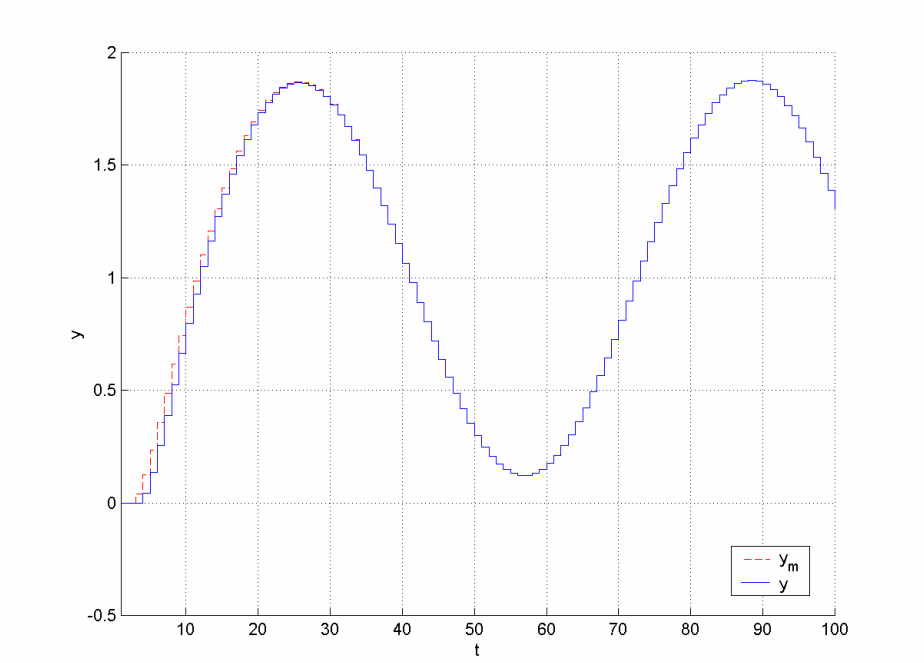

Figure 7C.2 *Time history of the output* y *for* $r(t) = 1 + \sin(0.1t)$ *in Example* 7.5.1.

Example 7.5.2 The following is a reduced model for the dynamics of a hard-disk drive (HDD) servo system which is obtained by ignoring the resonance characteristics and higher-frequency flexible modes in the system

$$y = H(s)u \quad \text{with} \quad H(s) = \frac{K_p}{s^2},$$

where y and u are the HDD sensor output, which represent the position of the reading magnet of the HDD with respect to the center of the desired disk track and actuator input, respectively. Assume that it is required for the output (the position of the reading head) to track a square reference signal, a piecewise constant signal switching between the values $0\ \mu\text{m}$ and $1\ \mu\text{m}$ (switching occurs every 2 msec).

Let $K_p = 10^6$. To apply a digital control, we first need to discretize our plant. Consider a sampling frequency of 10 KHz ($T_s = 0.1$ msec). The zero-order hold (ZOH) equivalent of $H(s)$ at this sampling frequency is

$$H(q) = \frac{0.005q + 0.005}{q^2 - 2q + 1}.$$

Hence, denoting $u(kT_s)$ and $y(kT_s)$ as $u[k]$ and $y[k]$, respectively, the corresponding ARMA model takes the form

$$A\left(q^{-1}\right)y[k] = B\left(q^{-1}\right)u[k],$$

where

$$A\left(q^{-1}\right)=1-2q^{-1}+q^{-2},$$
$$B\left(q^{-1}\right)=q^{-1}\left(0.005+0.005q^{-1}\right)=q^{-1}\bar{B}\left(q^{-1}\right).$$

In order to design a model reference controller for the above control task, we first need to choose a stable reference model with a steady-state value of 1 as before. Let this reference model be

$$E\left(q^{-1}\right)y_m(t)=q^{-3}\bar{H}\left(q^{-1}\right)r(t)$$

with

$$E\left(q^{-1}\right)=1-0.9q^{-1}+0.2q^{-2},$$
$$\bar{H}\left(q^{-1}\right)=0.2+0.1q^{-1}.$$

Solving

$$F\left(q^{-1}\right)A\left(q^{-1}\right)+q^{-1}G\left(q^{-1}\right)=E(q^{-1}),$$

we obtain

$$F\left(q^{-1}\right)=1,$$
$$G\left(q^{-1}\right)=1.1-0.8q^{-1},$$
$$\beta\left(q^{-1}\right)=F\left(q^{-1}\right)\bar{B}\left(q^{-1}\right)=0.005+0.005q^{-1}.$$

Hence the control law is obtained as

$$u[k]=\frac{1}{\beta_0}\bar{H}\left(q^{-1}\right)r[k]-\bar{G}\left(q^{-1}\right)y[k]-q\bar{\beta}\left(q^{-1}\right)u[k-1]$$
$$=40r[k]+20r[k-1]-220y[k]+160y[k]-u[k].$$

Applying the square reference signal above, simulation results for the MRC-controlled plant are shown in Figure 7C.3. Compared to the one-step-ahead control of the system, the response is slower and smoother, and the energy of the command signal is much smaller. ■

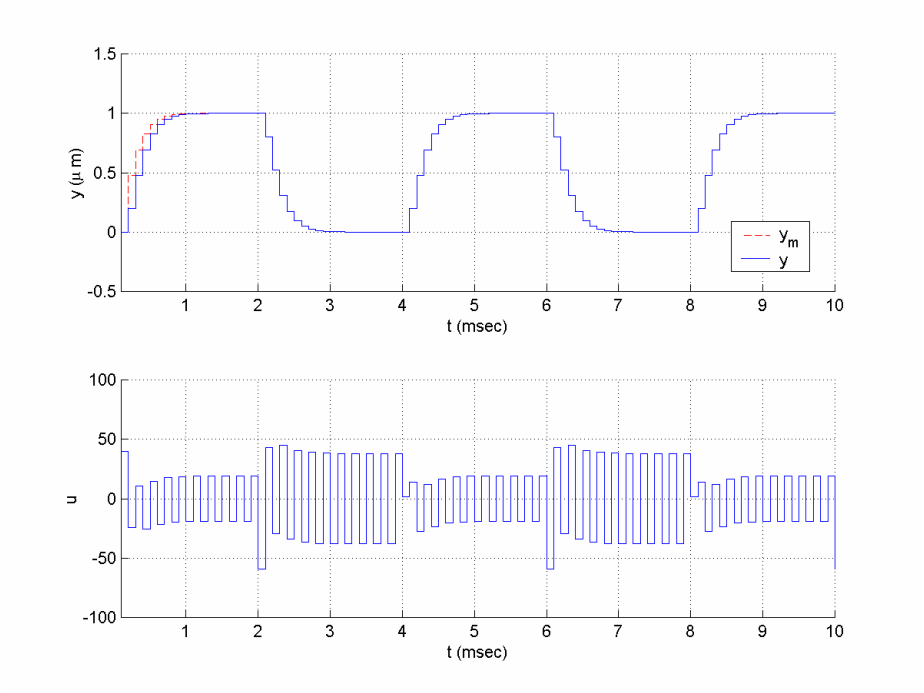

Figure 7C.3 *Time histories of the input and output signals in Example* 7.5.2.

The MATLAB function **udmracdr** incorporating some functions to generate the parameter estimates can be used to implement the direct MRAC algorithm presented in section 7.2.3, as demonstrated in Examples 7.5.3 and 7.5.4. In order to implement the indirect MRAC algorithm given in section 7.2.4, the function **udmracidr** can be used, as demonstrated in Example 7.5.3. If any robustness modification is needed, **urobust** can be incorporated to perform this modification.

Example 7.5.3 Let us perform the control task of Example 7.5.1 assuming that the plant parameters are not available this time and using a direct model reference adaptive controller. The control scheme will be based on the parametric model

$$u(t) = \theta^{*T}\phi(t),$$

where

$$\phi(t) = \left[E\left(q^{-1}\right)y(t+3), -y(t), -y(t-1), -u(t-1), \ldots, -u(t-4)\right]^T,$$

$$\theta^* = \left[\frac{1}{\beta_0} \quad \bar{g}_0 \quad \bar{g}_1 \quad \bar{\beta}_1 \quad \cdots \quad \bar{\beta}_4\right]^T.$$

The control law is given by

$$u(t) = \theta^T(t)\bar{\phi}(t),$$

where

$$\bar{\phi}(t)=\left[\bar{H}\left(q^{-1}\right)r(t),-y(t),-y(t-1),-u(t-1),\ldots,-u(t-4)\right]^T.$$

We can use the following recursive least-squares (RLS) algorithm with start-up forgetting ($\beta(0)=0.1,\beta_1=0.9$) to generate the parameter estimates:

$$\bar{\theta}(t)=\theta(t-1)+\frac{P(t-1)\phi(t-3)}{\beta(t)+\phi^T(t-3)P(t-1)\phi(t-3)}\left(y(t)-\theta^T(t-1)\phi(t-3)\right),\quad \theta(0)=[1,0,\ldots,0]^T,$$

$$P(t)=\frac{1}{\beta(t)}\left[P(t-1)-\frac{P(t-1)\phi(t-3)\phi^T(t-3)P(t-1)}{\beta(t)+\phi^T(t-3)P(t-1)\phi(t-3)}\right],\quad P(0)=I,$$

$$\beta(t)=\beta_1\beta(t-1)+1-\beta,$$

$$\theta(t)=\Pr(\bar{\theta}(t)),$$

where the projection mapping $\Pr(\cdot)$ is chosen to guarantee $0.1\le\theta_1\le 10$. Noting that the reference model is

$$E\left(q^{-1}\right)y_m(t)=q^{-3}\bar{H}\left(q^{-1}\right)r(t),$$

where

$$E\left(q^{-1}\right)=1-1.5q^{-1}+0.56q^{-2},$$
$$\bar{H}\left(q^{-1}\right)=0.04+0.02q^{-1},$$

the following code can be used to simulate the control of the plant:

```
A = [1 -1.9 0.9];
Bbar = [1 0.5 0.25];
n = 2; m = 2; d = 3; kE = 2; l = 1;
E = [1 -1.5 0.56]; Hbar = [0.04 0.02];

t_final = 100;
t = 1:t_final;

theta0 = zeros(7,1);
theta0(1) = 1;
Pj = uparproj('hyperplane',1,0.1,10);
P = eye(7);
ArgLS = urlsarg('startup forgetting',0.1,0.9);
x_pi = [theta0; P(:);0.1];
[n_c, x_c] = udmracdr('init',[n d m kE l]);
[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
[nstatem,xm] = uarma('init',d-1,Hbar,E);  %ym(t)/r(t-1)
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
ym(1) = uarma('output',xm,1,d-1,Hbar,E);
xm = uarma('state',xm,1,d-1,Hbar,E);
for k = 1:d-1,
   r(k) = 1;
   ym(k+1) = uarma('output',xm,r(k),d-1,Hbar,E);
   xm = uarma('state',xm,r(k),d-1,Hbar,E);
```

```
end

for k = 1:t_final,
   r(k+d-1) = 1;
   ym(k+d) = uarma('output',xm,r(k+d-1),d-1,Hbar,E);
   xm = uarma('state',xm,r(k+d-1),d-1,Hbar,E);
   [u_old phi] = udmracdr('regressor',x_c,y(k),[n d m kE], E);
   x_pi = udrls(x_pi, u_old, phi, 0, ArgLS, [], Pj);
   theta(:,k) = x_pi(1:7);
   u(k)=udmracdr('control',x_c,[y(k) r(k)],[n d m kE
l],theta(:,k),Hbar);
   y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
   x = uarma('state',x,u(k),d-1,Bbar,A);
   x_c = udmracdr('state', x_c, [u(k) y(k) r(k)], [n d m kE l]);
end;
```

The simulation results are plotted in Figures 7C.4, 7C.5, and 7C.6. The difference compared to the adaptive one-step-ahead control is significant. The overshoot is brought to an acceptable range.

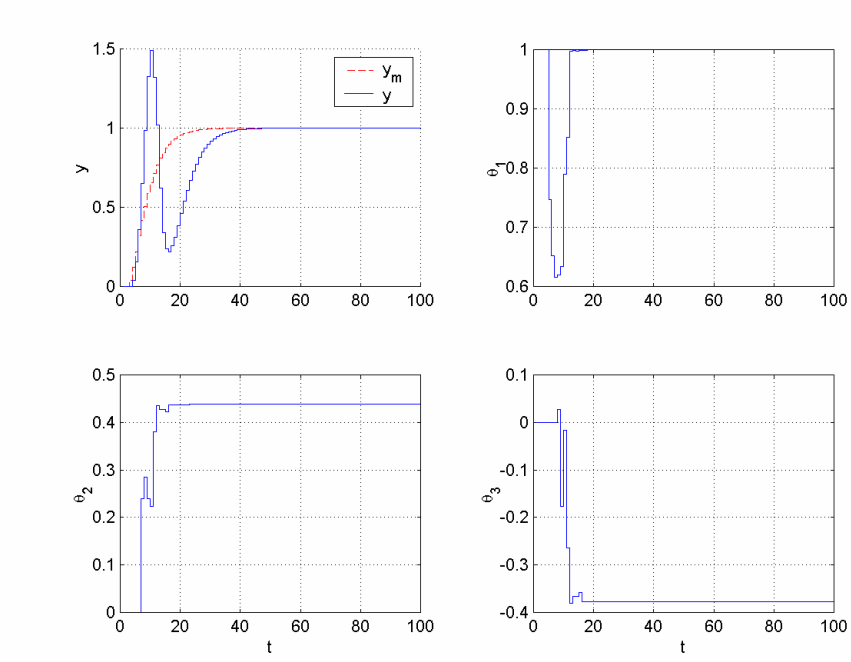

Figure 7C.4 *Time histories of the output* y *and parameter estimates* $\theta_1, \theta_2, \theta_2$ *for* $r(t) = 1$ *in Example* 7.5.3.

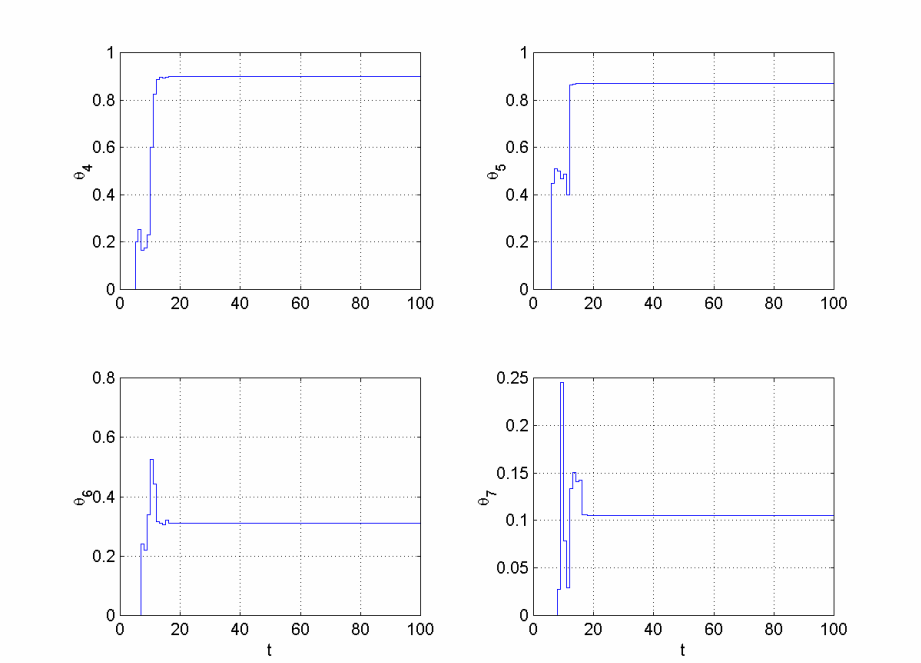

Figure 7C.5 *Time histories of the parameter estimates* $\theta_4,\ldots,\theta_7$ *for* $r(t)=1$ *in Example* 7.5.3.

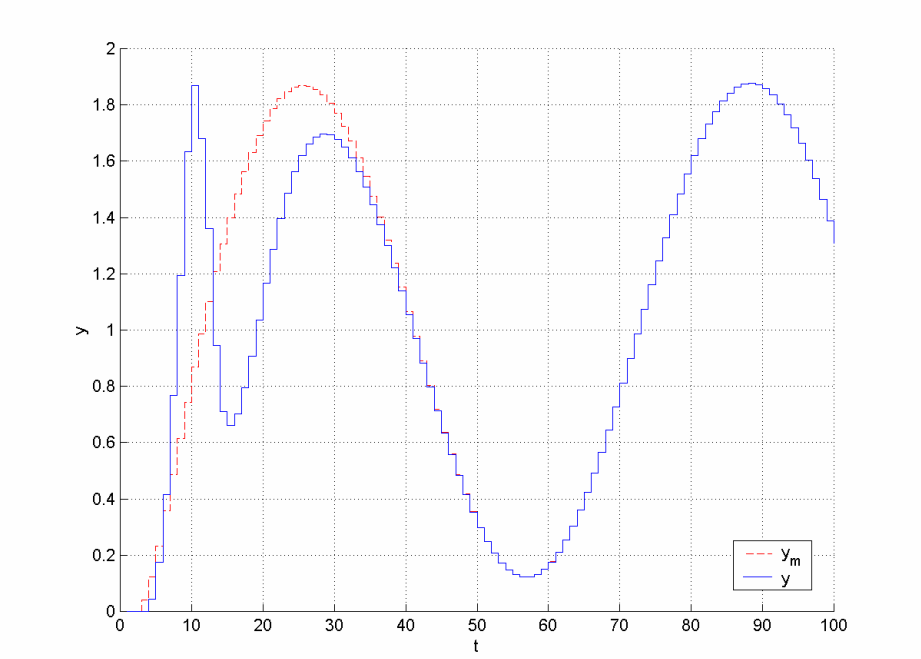

Figure 7C.6 *Time history of the output* y *for* $r(t)=1+3\sin(0.5t)$ *in Example* 7.5.3.

We can repeat the same task using indirect MRAC. In order to do this, we first estimate the system parameters based on the linear parametric model

$$y(t)=\theta^{*T}\phi(t),$$

where

$$\phi(t) = [u(t-3), u(t-4), u(t-5), -y(t-1), -y(t-2)]^T,$$
$$\theta^* = \left[b_0, b_1, b_2, a_1, a_2\right]^T.$$

The parameter estimate vector $\theta(t) = [\hat{b}_0(t), \hat{b}_1(t), \hat{b}_2(t), \hat{a}_1(t), \hat{a}_2(t)]^T$ can be obtained using any PI algorithm with parameter projection to keep $\hat{b}_0$ away from zero, e.g., $0.1 \le \theta_1 \le 10$. Let us use the following RLS algorithm with start-up forgetting as before:

$$\overline{\theta}(t) = \theta(t-1) + \frac{P(t-1)\phi(t)}{\beta(t) + \phi^T(t)P(t-1)\phi(t)}\left(y(t) - \theta^T(t-1)\phi(t)\right), \quad \theta(0) = [1, 0, \ldots, 0]^T,$$
$$P(t) = \frac{1}{\beta(t)}\left[P(t-1) - \frac{P(t-1)\phi(t)\phi^T(t)P(t-1)}{\beta(t) + \phi^T(t)P(t-1)\phi(t)}\right], \quad P(0) = I,$$
$$\beta(t) = \beta_1\beta(t-1) + 1 - \beta_1,$$
$$\theta(t) = \left[\hat{b}_0(t), \hat{b}_1(t), \hat{b}_2(t), \hat{a}_1(t), \hat{a}_2(t)\right]^T = \Pr(\overline{\theta}(t)).$$

Computing $\hat{A}(q^{-1}, t) = 1 + \hat{a}_1(t)q^{-1} + \hat{a}_2(t)q^{-2}$ and $\hat{\bar{B}}(q^{-1}, t) = \hat{b}_0(t) + \hat{b}_1(t)q^{-1} + \hat{b}_2(t)q^{-2}$, we solve the equation

$$\hat{F}\left(q^{-1}, t\right)\hat{A}\left(q^{-1}, t\right) + q^{-d}\hat{G}\left(q^{-1}, t\right) = E(q^{-1});$$

form the polynomials

$$\hat{\beta}(q^{-1}, t) = \hat{\beta}_0(t) + \hat{\beta}_1(t)q^{-1} + \cdots + \hat{\beta}_4(t)q^{-4} = \hat{F}(q^{-1}, t)\hat{\bar{B}}(q^{-1}, t),$$
$$\hat{\bar{G}}(q^{-1}, t) = \frac{1}{\hat{\beta}_0(t)}\hat{G}(q^{-1}, t),$$
$$\hat{\bar{\beta}}(q^{-1}, t) = \frac{1}{\hat{\beta}_0(t)}\left(\hat{\beta}(q^{-1}, t) - \hat{\beta}_0(t)\right);$$

and generate the control signal u as

$$u(t) = \frac{1}{\hat{\beta}_0(t)}\overline{H}\left(q^{-1}\right)r(t) - \hat{\bar{G}}\left(q^{-1}, t\right)y(t) - q\hat{\bar{\beta}}\left(q^{-1}, t\right)u(t-1).$$

The following code can be used for simulation:

```
A = [1 -1.9 0.9];
Bbar = [1 0.5 0.25];
n = 2; m = 2; d = 3; kE = 2; l = 1;
E = [1 -1.5 0.56]; Hbar = [0.04 0.02];

t_final = 100;
t = 1:t_final;

theta0 = zeros(5,1);
```

```
theta0(1) = 1;
Pj = uparproj('hyperplane',1,0.1,10);
P = eye(5);
ArgLS = urlsarg('startup forgetting',0.1,0.9);
x_pi = [theta0; P(:);0.1];
[n_c, x_c] = udmracidr('init',[n d m kE l]);
[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
[nstatem,xm] = uarma('init',d-1,Hbar,E);  %ym(t)/r(t-1)
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
ym(1) = uarma('output',xm,1,d-1,Hbar,E);
xm = uarma('state',xm,1,d-1,Hbar,E);
for k = 1:d-1,
   r(k) = 1;
   ym(k+1) = uarma('output',xm,r(k),d-1,Hbar,E);
   xm = uarma('state',xm,r(k),d-1,Hbar,E);
end

for k = 1:t_final,
   r(k+d-1) = 1;
   ym(k+d) = uarma('output',xm,r(k+d-1),d-1,Hbar,E);
   xm = uarma('state',xm,r(k+d-1),d-1,Hbar,E);
   phi = udmracidr('regressor',x_c,[n d m kE]);
   x_pi = udrls(x_pi, y(k), phi, 0, ArgLS, [], Pj);
   theta(:,k) = x_pi(1:5);
   u(k)=udmracidr('control',x_c,[y(k) r(k)],[n d m kE
l],theta(:,k),Hbar,E);
   y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
   x = uarma('state',x,u(k),d-1,Bbar,A);
   x_c = udmracidr('state', x_c, [u(k) y(k) r(k)], [n d m
kE l]);
end;
```

The results are plotted in Figures 7C.7 and 7C.8. The transient behavior is worse than the direct MRAC case and better than the behaviors obtained via adaptive one-step-ahead controllers. ■

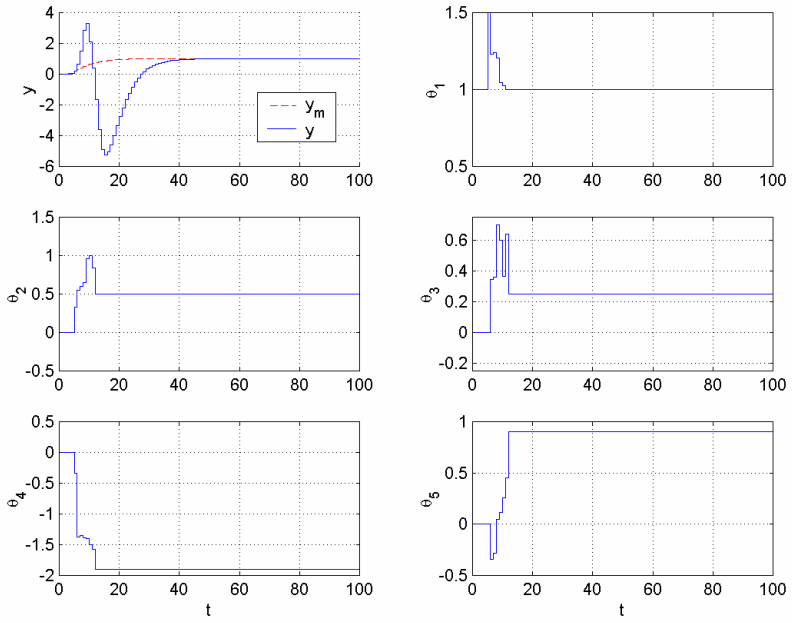

Figure 7C.7 *Time histories of the output* y *and parameter estimates* θ_i *for* $y_m(t)=1$ *using indirect MRAC in Example* 7.5.3.

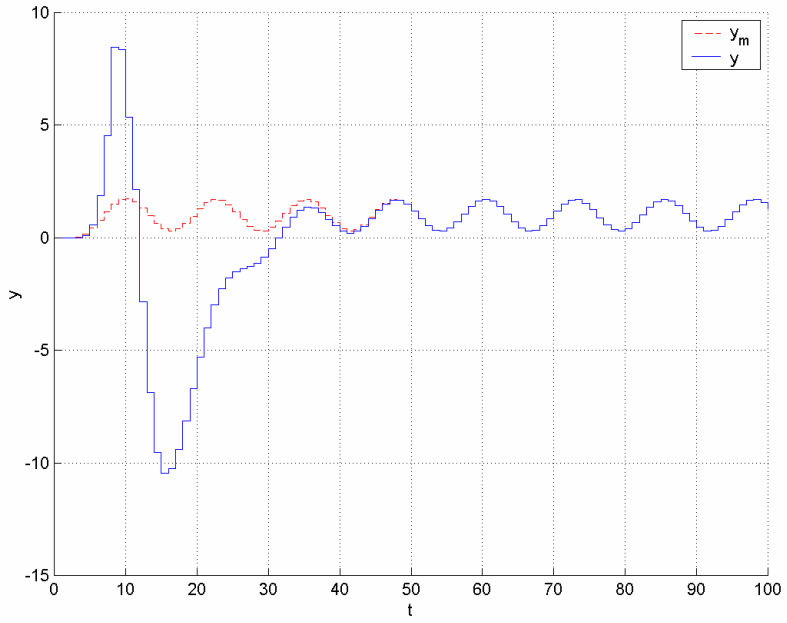

Figure 7C.8 *Time history of the output* y *for* $y_m(t)=1+3\sin(0.5t)$ *using indirect MRAC in Example* 7.5.3.

Example 7.5.4 Let us perform the control task of Example 7.5.2 assuming that the plant parameters are not available this time and using a direct model reference adaptive controller. The control scheme will be based on the parametric model

$$u[k] = \theta^{*T}\phi[k]$$

with

$$\phi[k] = \left[E\left(q^{-1}\right)y[k+1], -y[k], -y[k-1], -u[k-1]\right]^T,$$
$$\theta^* = \left[\frac{1}{\beta_0}, \frac{\bar{g}_0}{\beta_0}, \frac{\bar{g}_1}{\beta_0}, \frac{\bar{\beta}_1}{\beta_0}\right]^T.$$

The control law is given by

$$u[k] = \theta^T[k]\bar{\phi}[k],$$

where

$$\bar{\phi}[k] = [\bar{H}(q^{-1})r[k], -y[k], -y[k-1], -u[k-1]]^T.$$

Let $r[k]$ be a square signal as before. We can use the RLS algorithm with start-up forgetting of Example 7.5.2 without any change other than choosing the initial estimate vector as $[500,0,0,0]^T$ and the projection mapping $\Pr(\cdot)$ so that $1 \le \theta_1 \le 1000$ is guaranteed. The following code can be used to simulate the control of the HDD servo system:

```
A A = [1 -2 1];
Bbar = 0.005*[1 1];
n = 2; m = 1; d = 1; kE = 2; l = 1;
Ts = 0.1;
E = [1 -0.9 0.2]; Hbar = [0.15 0.15];

t_final = 100;
t = Ts*(1:t_final);

theta0 = zeros(4,1);
theta0(1) = 500;
Pj = uparproj('hyperplane',1,1,1000);
P = eye(4);
ArgLS = urlsarg('startup forgetting',0.1,0.9);
x_pi = [theta0; P(:);0.1];
[n_c, x_c] = udmracdr('init',[n d m kE l]);
[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
[nstatem,xm] = uarma('init',d-1,Hbar,E);  %ym(t)/r(t-1)
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
ym(1) = uarma('output',xm,0,d-1,Hbar,E);
xm = uarma('state',xm,1,d-1,Hbar,E);

for k = 1:t_final,
   r(k) = mod(floor(k/20)+1,2);
   ym(k+1) = uarma('output',xm,r(k),d-1,Hbar,E);
   xm = uarma('state',xm,r(k),d-1,Hbar,E);
   [u_old phi] = udmracdr('regressor',x_c,y(k),[n d m kE], E);
   x_pi = udrls(x_pi, u_old, phi, 0, ArgLS, [], Pj);
   theta(:,k) = x_pi(1:4);
```

```
   u(k)=udmracdr('control',x_c,[y(k) r(k)],[n d m kE
l],theta(:,k),Hbar);
   y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
   x = uarma('state',x,u(k),d-1,Bbar,A);
   x_c = udmracdr('state', x_c, [u(k) y(k) r(k)], [n d m kE l]);
end;
```

The results are plotted in Figure 7C.9. The transient behavior is significantly better than the cases with the adaptive one-step-ahead controllers. Energy of the control signal is less as well. ■

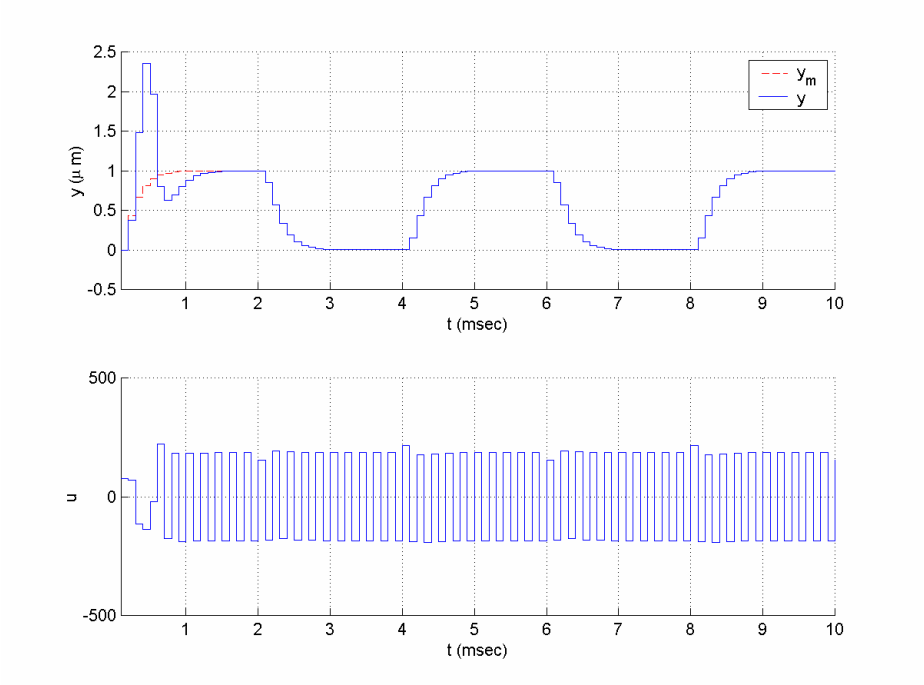

Figure 7C.9 *Time histories of the input and output signals in Example* 7.5.4.

7.5.2 Adaptive Prediction and Control

The Adaptive Control Toolbox provides a rich tool set for adaptive prediction and one-step-ahead predictive control. Before introducing the features provided in the toolbox, we revisit the adaptive prediction approaches to cover some concepts omitted in the main text. After this revisit, a number of examples will be provided to demonstrate the use of the Adapative Control Toolbox for designing and implementing adaptive prediction and one-step-ahead control schemes.

Adaptive Prediction

Consider the ARMA model

$$A\left(q^{-1}\right)y(t)=B\left(q^{-1}\right)u(t), \tag{7C.63}$$

where

$$A\left(q^{-1}\right)=1+a_1q^{-1}+\cdots+a_nq^{-n},$$
$$B\left(q^{-1}\right)=q^{-d}\left(b_0+\cdots+b_mq^{-m}\right)=q^{-d}\overline{B}\left(q^{-1}\right),$$

d is the so-called plant delay, u is the input, and y is the output. The output of the model at the time $t+d$ can be expressed as

$$R\left(q^{-1}\right)y(t+d)=G\left(q^{-1}\right)y(t)+F\left(q^{-1}\right)\overline{B}\left(q^{-1}\right)u(t), \tag{7C.64}$$

where $R(q^{-1})$ is a design polynomial usually chosen as $R(q^{-1})=1$ and $F(q^{-1})$, $G(q^{-1})$ are computed by solving the Bezout equation

$$F\left(q^{-1}\right)A\left(q^{-1}\right)+q^{-d}G\left(q^{-1}\right)=R\left(q^{-1}\right). \tag{7C.65}$$

Equation (7C.65) has a unique solution in the form

$$F\left(q^{-1}\right)=1+f_1q^{-1}+\cdots+f_{d-1}q^{-d+1},$$
$$G\left(q^{-1}\right)=g_0+g_1q^{-1}+\cdots+g_{n-1}q^{-n+1}$$

for $R(q^{-1})=1$ [1].

Direct Adaptive Predictor

When the coefficients of $A(q^{-1})$, $B(q^{-1})$ in (7C.63) are unknown, we express (7C.64) in the form of an SPM. Choosing $R(q^{-1})=1$ for simplicity and defining

$$\beta(q^{-1})=F(q^{-1})\overline{B}(q^{-1})=\beta_0+\beta_1q^{-1}+\cdots+\beta_{d+m-1}q^{-d-m+1},$$

we obtain

$$y(t+d)=\theta^{*T}\phi(t), \tag{7C.66}$$

where

$$\phi(t)=[u(t),u(t-1),\ldots,u(t-d-m+1),-y(t),-y(t-1),\ldots,-y(t-n+1)]^T,$$
$$\theta^*=\left[\beta_0,\beta_1,\ldots,\beta_{d+m-1},-g_0,\ldots,-g_{n-1}\right]^T.$$

Since $y(t+d)$ is not available for measurement at time t we rewrite (7C.66) as

$$y(t) = \theta^{*T}\phi(t-d). \tag{7C.67}$$

Any of the PI algorithms discussed in Chapter 4 can be used to generate $\theta(t)$, the estimate of θ^* at time t. The only modification needed is replacement of $\phi(t)$ and z with $\phi(t-d)$ and y, respectively. Then the estimate of $y(t+d)$ is given by

$$\hat{y}(t+d) = \theta^T(t)\phi(t). \tag{7C.68}$$

Indirect Adaptive Predictor

In this case, the DARMA model

$$A\left(q^{-1}\right)y(t) = q^{-d}\overline{B}\left(q^{-1}\right)u(t)$$

is first expressed as

$$y(t) = \theta_p^{*T}\phi(t), \tag{7C.69}$$

where

$$\theta_p^* = \left[b_0,\ \ldots,\ b_m,\ a_1,\ \ldots,\ a_n\right]^T,$$
$$\phi(t) = [u(t-d),\ldots,u(t-d-m),-y(t-1),\ldots,-y(t-n)]^T.$$

Using a PI algorithm and (7C.69), we generate the estimate $\theta_p(t)$ of θ_p^* at time t and form the estimated polynomials

$$\begin{aligned}\hat{A}\left(q^{-1},t\right) &= 1+\hat{a}_1(t)q^{-1}+\cdots+\hat{a}_n(t)q^{-n},\\ \hat{B}\left(q^{-1},t\right) &= \hat{b}_0(t)+\cdots+\hat{b}_m(t)q^{-m}.\end{aligned} \tag{7C.70}$$

Solving

$$\hat{F}\left(q^{-1},t\right)\hat{A}\left(q^{-1},t\right)+q^{-d}\hat{G}\left(q^{-1},t\right)=1 \tag{7C.71}$$

for $\hat{F}(q^{-1},t),\hat{G}(q^{-1},t)$ at each time t, the indirect adaptive predictor estimate is obtained as

$$\hat{y}(t+d) = \hat{G}\left(q^{-1},t\right)y(t)+\hat{F}\left(q^{-1},t\right)\hat{B}\left(q^{-1},t\right)u(t). \tag{7C.72}$$

When the coefficients of $A(q^{-1})$, $B(q^{-1})$ in (7C.63) are known, the MATLAB functions **`diophant`** and **`dprd`** can be used to solve the Bezout equation and construct the predictor, respectively, as demonstrated in Examples 7.5.5 and 7.5.6. If the coefficients of $A(q^{-1})$, $B(q^{-1})$ in (7C.63) are unknown, the MATLAB functions **`udprddr`** and **`udprdidr`** can be used to simulate, respectively, the direct and indirect adaptive prediction algorithms above, as demonstrated in Examples 7.5.7–7.5.10.

Example 7.5.5 Consider the plant

$$y(t)-1.9y(t-1)+0.9y(t-2)=u(t-3)+0.5u(t-4)+0.25u(t-3).$$

Let us design a 3-steps-ahead predictor for this plant. First, we need to put the system into the generic ARMA form as

$$A\left(q^{-1}\right)y(t)=B\left(q^{-1}\right)u(t),$$

where

$$A\left(q^{-1}\right)=1-1.9q^{-1}+0.9q^{-2},$$
$$B\left(q^{-1}\right)=q^{-3}\overline{B}\left(q^{-1}\right)=q^{-3}\left(1+0.5q^{-1}+0.25q^{-2}\right).$$

To construct the predictor, we need to solve the Bezout equation

$$F\left(q^{-1}\right)A\left(q^{-1}\right)+q^{-3}G\left(q^{-1}\right)=1;$$

the solution in the form

$$F\left(q^{-1}\right)=1+f_1q^{-1}+f_2q^{-2},$$
$$G\left(q^{-1}\right)=g_0+g_1q^{-1}$$

is unique. One can use different methods [1,2,3] to solve this equation by hand. We will use the MATLAB function **`diophant`** for this task. The code

```
A = [1 -1.9 0.9];
Bbar = [1 0.5 0.25];
q_d = [0 0 0 1];
POLYTYPE = 1; % showing the polynomials are in the form
               % a0 + a1 q^-1 + ...
[F, G, RETYPE] = diophant(A, q_d, 1, POLYTYPE)
beta = conv(F,Bbar)
```

results in

$$F\left(q^{-1}\right)=1+1.9q^{-1}+2.71q^{-2},$$
$$G\left(q^{-1}\right)=3.439-2.439q^{-1},$$
$$\beta\left(q^{-1}\right)=F\left(q^{-1}\right)\overline{B}\left(q^{-1}\right)=1+2.4q^{-1}+3.91q^{-2}+1.83q^{-3}+0.6775q^{-4}.$$

Hence, the predictor equation is found as

$$\hat{y}(t+3)=\left(3.439-2.439q^{-1}\right)y(t)+\left(1+2.4q^{-1}+3.91q^{-2}+1.83q^{-3}+0.6775q^{-4}\right)u(t).$$

The same predictor could be constructed directly by using **`dprd`** as follows:

```
d = 3;
A = [1 -1.9 0.9];
```

```
Bbar = [1 0.5 0.25];
[beta, G] = dprd(d, Bbar, A)
```

Since there is no disturbance or uncertainty affecting the system, $\hat{y}(t+3) = y(t+3)$ will be satisfied for $t \geq 3$ (assuming that the prediction is started at $t = 0$). ■

Example 7.5.6 Consider the signal $y(t) = \cos \omega t$, with $\omega = \frac{\pi}{7}$. Let us construct a 10-steps-ahead linear predictor to predict the future values of $y(t)$ using the current and previous measurements. To do this, we first observe that

$$\cos \omega t + \cos(\omega t - 2\omega) = 2 \cos \omega \cos(\omega t - \omega) .$$

Based on this observation, we construct the ARMA model as

$$A\left(q^{-1}\right) y\left(t\right) = 0$$

with $A(q^{-1}) = 1 - 2\cos \omega q^{-1} + q^{-2}$. To construct the predictor, we solve the Bezout equation

$$F\left(q^{-1}\right) A\left(q^{-1}\right) + q^{-10} G\left(q^{-1}\right) = 1$$

to obtain $G(q^{-1})$. Note that since $\bar{B}(q^{-1}) = 0$, we do not need $F(q^{-1})$. Using the function **`diophant,`** we obtain $G(q^{-1}) = -2.247 + 2.247 q^{-1}$. Hence the predictor is obtained as

$$\hat{y}\left(t+10\right) = -2.247 y(t) + 2.247 y(t-1). \qquad \blacksquare$$

Example 7.5.7 Consider the plant of Example 7.5.5 driven by the input signal

$$u(t) = \cos\left(\frac{\pi}{20} t\right) - 0.6 \cos\left(\frac{\pi}{35} t\right) + 0.3 \sin\left(\frac{\pi}{12} t + \frac{\pi}{3}\right).$$

Assume that the plant parameters are not known, i.e., we know only that the plant is in the form

$$\begin{aligned} A\left(q^{-1}\right) y\left(t\right) &= B\left(q^{-1}\right) u\left(t\right), \\ A\left(q^{-1}\right) &= 1 + a_2 q^{-1} + a_2 q^{-2}, \\ B\left(q^{-1}\right) = q^{-3} \bar{B}\left(q^{-1}\right) &= q^{-3} \left(b_0 + b_1 q^{-1} + b_2 q^{-2}\right). \end{aligned}$$

Let us design a 3-steps-ahead adaptive predictor for this plant. The adaptive predictor will have the form

$$\hat{y}(t+d) = \theta^T(t) \phi(t),$$

where

$$\phi(t) = [u(t), u(t-1), \ldots, u(t-4), -y(t), -y(t-1)]^T$$

and $\theta(t)$ is the estimate of the parameter vector $\theta^* = [\beta_0, \beta_1, \ldots, \beta_4, -g_0, -g_1]^T$, which consists of parameters of the polynomials $\beta(q^{-1})$ and $G(q^{-1})$ described in the text. We need to perform the parameter estimation based on the parametric model

$$y(t) = \theta^{*T}\phi(t-d).$$

Let us use normalized parameter projection algorithms for estimation; i.e., to obtain $\theta(t)$, let us use

$$\theta(t) = \theta(t-1) + \frac{P(t-1)\phi(t-d)}{\phi^T(t-d)P(t-1)\phi(t-d)}\left(y(t) - \theta^T(t-1)\phi(t-d)\right), \quad \theta(0) = [0,\ldots,0]^T,$$

$$P(t) = P(t-1) - \frac{P(t-1)\phi(t-d)\phi^T(t-d)P(t-1)}{\phi^T(t-d)P(t-1)\phi(t-d)}, \quad P(0) = I.$$

Then we can use the following code to simulate the process and the prediction for $t \in [1,100]$:

```
A = [1 -1.9 0.9];
Bbar = [1 0.5 0.25];
n = 2; m = 2; d = 3;

t_final = 100;
t = 1:t_final;

theta0 = zeros(7,1);
P = eye(7);
x_pi = [theta0; P(:)];
[nstate,x] = ufilt('init',Bbar,A);
[n_pr, x_pr] = udprddr('init',[n d m]);

for k = 1:t_final,
   k3 = k-3;
   u_k = cos(pi/20*k)-0.6*cos(pi/35*k)+0.3*sin(pi/12*k+pi/3);
   u_k3 = cos(pi/20*k3)-0.6*cos(pi/35*k3)+0.3*sin(pi/12*k3+pi/3);
   y(k) = ufilt('output', x, u_k3, Bbar, A, 0);
   x = ufilt('state', x, u_k3, Bbar, A);
   phi = udprddr('regressor', x_pr, [u_k y(k)], [n d m]);
   x_pi = udproj('orth', x_pi, y(k), phi);
   theta(:,k) = x_pi(1:7);
   yp(k+3) = udprddr('predict', x_pr, [u_k y(k)], [n d m], theta(:,k));
   x_pr = udprddr('state', x_pr, [u_k y(k)], [n d m]);
end;
```

The results are plotted in Figure 7C.10. We can see from these results that the settlement step for parameter estimation is about 50. Comparing with the results of Example 7.5.5, we also see that all of the parameter estimates converge to their real values. Settlement for output prediction is much faster. The output signal is predicted perfectly for $t > 12$. ■

Example 7.5.8 Consider the sinusoidal signal prediction problem of Example 7.5.6, i.e., prediction of the signal $y(t) = \cos\omega t$, with $\omega = \frac{\pi}{7}$. This time, assume that it is known that the signal y is in the form $y(t) = \alpha\cos\omega t$, but the coefficients α, ω are unknown.

Let us construct a 10-steps-ahead linear predictor to predict the future values of $y(t)$, as before.

Similar to Example 7.5.6, first observe that

$$\alpha\cos\omega t+\alpha\cos(\omega t-2\omega)=2\cos\omega\big(\alpha\cos(\omega t-\omega)\big),$$

and then construct the ARMA model as

$$A\big(q^{-1}\big)y\big(t\big)=0,$$

where $A(q^{-1})=1-2\cos\omega q^{-1}+q^{-2}$ ($\bar{B}=0$). We can write the adaptive predictor equations in the format of (7C.68) as

$$\hat{y}(t+10)=\theta^T(t)\phi(t),$$

where

$$\phi(t)=[\underbrace{0,\ldots,0}_{10},-y(t),-y(t-1)]^T$$

and $\theta(t)$ is the estimate of the parameter vector

$$\theta^*=[\underbrace{0,\ldots,0}_{10},-g_0,-g_1]^T,$$

where g_0,g_1 are coefficients of $G(q^{-1})$ as described in the text. We can get rid of the zero entries to simplify our model. An adaptive prediction scheme can be obtained by integrating the simplified model and the pure projection algorithm as follows:

$$\begin{aligned}
&\hat{y}(t+10)=\bar{\theta}^T(t)\bar{\phi}(t),\\
&\bar{\phi}(t)=[-y(t),-y(t-1)]^T,\\
&\bar{\theta}\big(t\big)=\bar{\theta}\big(t-1\big)+\frac{\bar{\phi}\big(t-10\big)}{\bar{\phi}^T\big(t-10\big)\bar{\phi}\big(t-10\big)}\Big(y\big(t\big)-\bar{\theta}^T\big(t-1\big)\bar{\phi}\big(t-10\big)\Big),\quad \bar{\theta}(0)=\begin{bmatrix}0 & 0\end{bmatrix}.
\end{aligned}$$

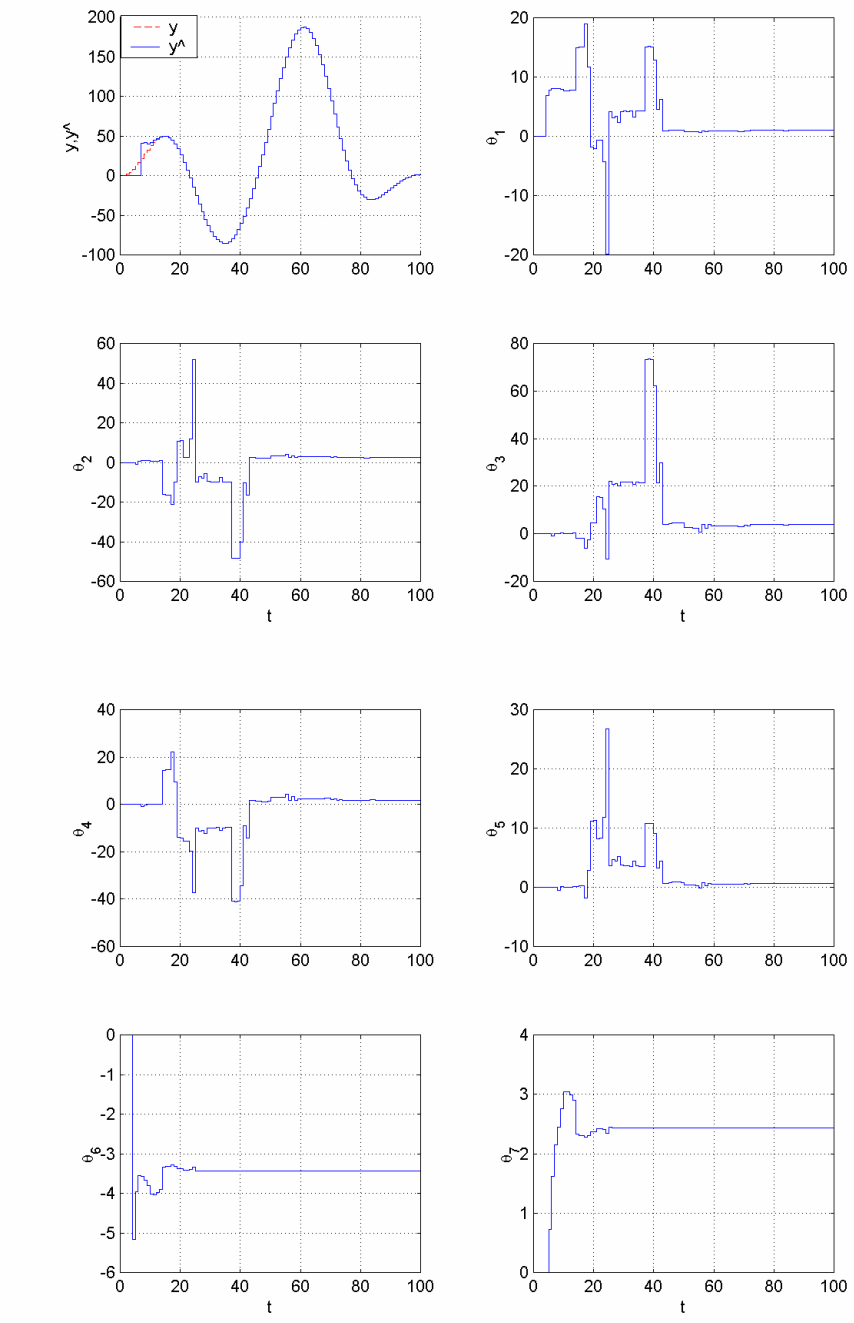

Figure 7C.10 *Time histories of the output* y, *the output prediction* $\hat{y}$, *and the parameter estimates* θ_i *in Example* 7.5.7.

The following code, which is based on the first parameterization above, can be used to simulate the prediction for $t \in [1,100]$:

```
w = pi/7;
n = 2; m = 0; d = 10;

t_final = 100;
t = 1:t_final;

x_pi = zeros(12,1);
[n_pr, x_pr] = udprddr('init',[n d m]);

for k = 1:t_final,
   y(k) = cos(w*k);
   phi = udprddr('regressor', x_pr, [0 y(k)], [n d m]);
   x_pi = udproj('pure', x_pi, y(k), phi);
   theta(:,k) = x_pi;
   yp(k+10) = udprddr('predict', x_pr, [0 y(k)], [n d m],
theta(:,k));
   x_pr = udprddr('state', x_pr, [0 y(k)], [n d m]);
end;
```

The results are plotted in Figure 7C.11. The sinusoidal signal $y(t)$ is predicted perfectly after the fortieth step. ■

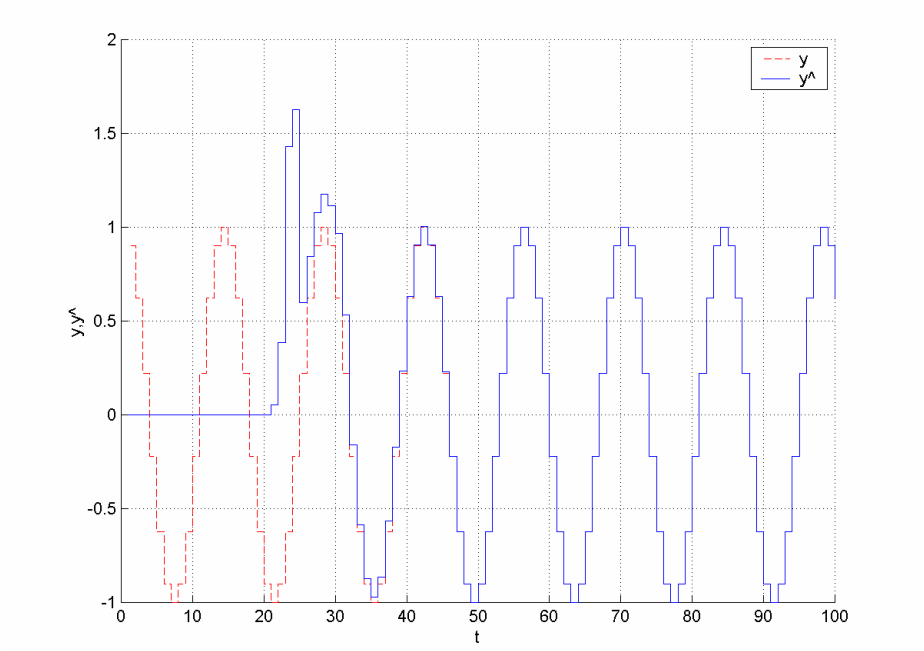

Figure 7C.11 *Time histories of the signal* y *and the signal prediction* $\hat{y}$ *in Example* 7.5.8.

Example 7.5.9 Let us repeat the adaptive prediction task of Example 7.5.7 using an indirect adaptive predictor and RLS algorithm with start-up forgetting. The integrated prediction scheme will be as follows:

$$\phi(t) = [u(t-3), u(t-4), u(t-5), -y(t-1), -y(t-2)]^T,$$
$$\beta(t) = \beta_1 \beta(t-1) + 1 - \beta_1,$$
$$P(t) = \frac{1}{\beta(t)}\left[P(t-1) - \frac{a(t)P(t-1)\phi(t)\phi^T(t)P(t-1)}{\beta(t) + a(t)\phi^T(t)P(t-1)\phi(t)}\right],$$
$$\theta(t) = \left[\hat{b}_0(t), \hat{b}_1(t), \hat{b}_2(t), \hat{a}_1(t), \hat{a}_2(t)\right]^T = \theta(t-1) + \frac{P(t-1)\phi(t)}{\beta(t) + \phi^T(t)P(t-1)\phi(t)}\left(z(t) - \theta^T(t-1)\phi(t)\right),$$

$$\hat{A}(q^{-1}, t) = 1 + \hat{a}_1(t)q^{-1} + \hat{a}_2(t)q^{-2},$$
$$\hat{B}(q^{-1}, t) = \hat{b}_0(t) + \hat{b}_1(t) + \hat{b}_2(t)q^{-2},$$
$$\hat{F}(q^{-1}, t)\hat{A}(q^{-1}, t) + q^{-d}\hat{G}(q^{-1}, t) = 1,$$

$$\hat{y}(t+3) = \hat{G}(q^{-1}, t)y(t) + \hat{F}(q^{-1}, t)\hat{B}(q^{-1}, t)u(t).$$

Selecting $\theta(0) = [0, \ldots, 0]^T, P(0) = I, \beta(0) = 0.7, \beta_1 = 0.99$, the following code can be used to simulate the process and the prediction for $t \in [1,100]$:

```
A = [1 -1.9 0.9];
Bbar = [1 0.5 0.25];
n = 2; m = 2; d = 3;

t_final = 100;
t = 1:t_final;

theta0 = zeros(5,1);
P = eye(5);
ArgLS = urlsarg('startup forgetting',0.7,0.99);
x_pi = [theta0; P(:);0.7];
[nstate,x] = ufilt('init',Bbar,A);
[n_pr, x_pr] = udprdidr('init',[n d m]);

for k = 1:t_final,
   k3 = k-3;
   u_k = cos(pi/20*k)-0.6*cos(pi/35*k)+0.3*sin(pi/12*k+pi/3);
   u_k3 = cos(pi/20*k3)-0.6*cos(pi/35*k3)+0.3*sin(pi/12*k3+pi/3);
   y(k) = ufilt('output', x, u_k3, Bbar, A, 0);
   x = ufilt('state', x, u_k3, Bbar, A);
   phi = udprdidr('regressor', x_pr, [u_k y(k)], [n d m]);
   x_pi = udrls(x_pi, y(k), phi, 0, ArgLS);
   theta(:,k) = x_pi(1:5);
   yp(k+3) = udprdidr('predict', x_pr, [u_k y(k)], [n d m], theta(:,k));
   x_pr = udprdidr('state', x_pr, [u_k y(k)], [n d m]);
end;
```

The results, which are comparable to those of Example 7.5.7, are plotted in Figure 7C.12. ■

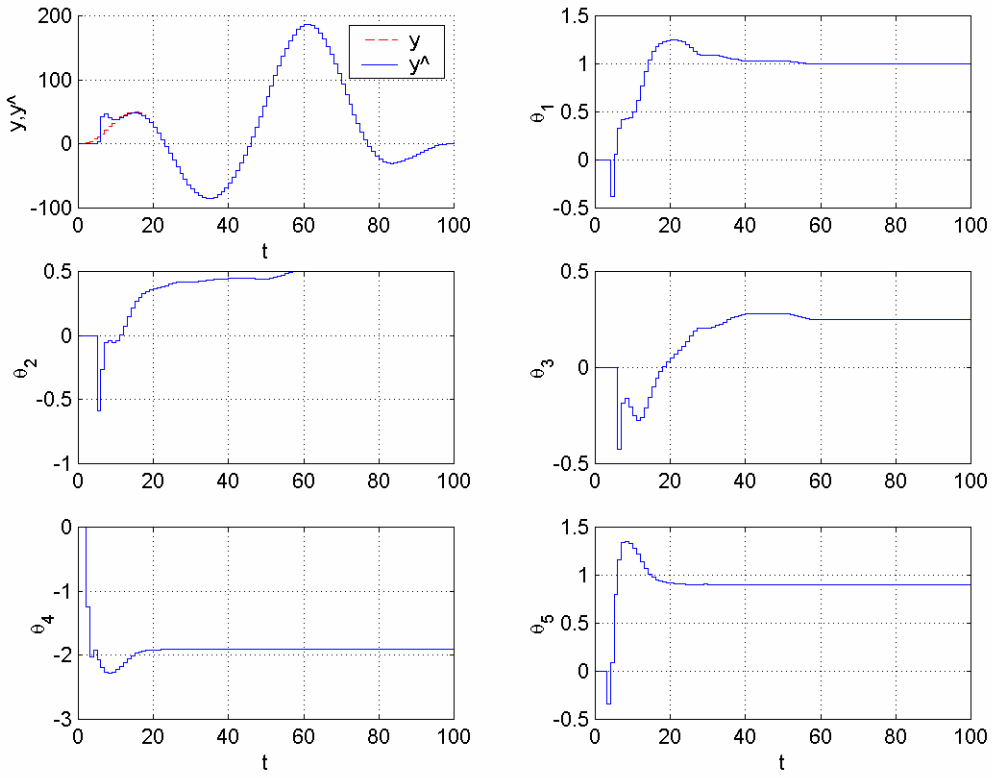

Figure 7C.12 *Time histories of the output* y, *the output prediction* $\hat{y}$, *and the parameter estimates* θ_i *in Example* 7.5.9.

Example 7.5.10 Consider the adaptive signal prediction problem of Example 7.5.8. Let us perform the same task using an indirect adaptive predictor and RLS with start-up forgetting.

The integrated prediction scheme will be as follows:

$$\phi(t) = [-y(t-1), -y(t-2)]^T,$$
$$\beta(t) = \beta_1 \beta(t-1) + 1 - \beta,$$
$$P(t) = \frac{1}{\beta(t)}\left[P(t-1) - \frac{a(t)P(t-1)\phi(t)\phi^T(t)P(t-1)}{\beta(t) + a(t)\phi^T(t)P(t-1)\phi(t)}\right],$$
$$\theta(t) = [\hat{a}_1(t), \hat{a}_2(t)]^T = \theta(t-1) + \frac{P(t-1)\phi(t)}{\beta(t) + \phi^T(t)P(t-1)\phi(t)}\left(z(t) - \theta^T(t-1)\phi(t)\right),$$

$$\hat{A}(q^{-1}, t) = 1 + \hat{a}_1(t)q^{-1} + \hat{a}_2(t)q^{-2},$$
$$\hat{F}(q^{-1}, t)\hat{A}(q^{-1}, t) + q^{-d}\hat{G}(q^{-1}, t) = 1,$$
$$\hat{y}(t+10) = \hat{G}(q^{-1}, t)y(t).$$

Selecting $\theta(0)=[0,0]^T, P(0)=I, \beta(0)=0.7, \beta_1=0.99$, the results plotted in Figure 7C.13 are obtained for $t\in[1,100]$. The prediction behavior is comparable to that in Example 7.5.8. ■

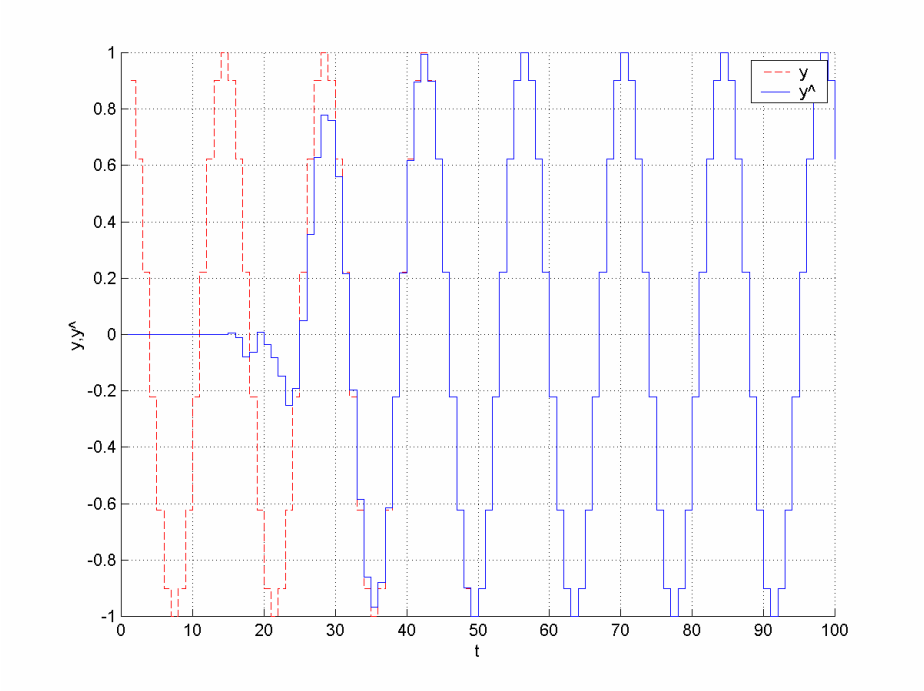

Figure 7C.13 *Time histories of the signal y and the signal prediction $\hat{y}$ in Example* 7.5.10.

7.5.3 One-Step-Ahead Control

The MATLAB function **`dosac`** can be used to simulate the pure and weighted one-step-ahead control algorithms above, as demonstrated in Examples 7.5.11 and 7.5.12.

Example 7.5.11 Consider the plant

$$y(t)-1.9y(t-1)+0.9y(t-2)=u(t-3)+0.5u(t-4)+0.25u(t-5)$$

of Examples 7.5.1 and 7.5.5. In Example 7.5.5 we had constructed a linear 3-steps-ahead predictor for this plant, and we had found the predictor polynomials as

$$F\left(q^{-1}\right)=1+1.9q^{-1}+2.71q^{-2},$$
$$G\left(q^{-1}\right)=3.439-2.439q^{-1},$$
$$\beta\left(q^{-1}\right)=F\left(q^{-1}\right)\bar{B}\left(q^{-1}\right)=1+2.4q^{-1}+3.91q^{-2}+1.83q^{-3}+0.6775q^{-4}.$$

Assume that the signal y is initially at rest, i.e., $y(0)=y(-1)=\cdots=0$. Now let us design a pure 3-steps-ahead controller to bring y to 1, i.e., $y_m(t)=1\ \forall t$. Our controller will be in the form (7.64). Hence computing $\alpha(q^{-1})$ and $\bar{\beta}(q^{-1})$ as

$$\alpha\left(q^{-1}\right) = G\left(q^{-1}\right) = 3.439 - 2.439q^{-1},$$
$$\beta\left(q^{-1}\right) = 2.4q^{-1} + 3.91q^{-2} + 1.83q^{-3} + 0.6775q^{-4},$$

we end up with the control law

$$u(t) = y_m(t+d) - 3.439y(t) + 2.439q^{-1}y(t-1) - 2.4u(t-1) - 3.91u(t-2) - 1.83u(t-3) - 0.6775u(t-4).$$

One can use the MATLAB function **dosac** to generate the coefficients of the control law directly. The plant controlled with the 3-steps-ahead controller above can be simulated for $t \in [1,20]$ using the following code:

```
A = [1 -1.9 0.9];
Bbar = [1 0.5 0.25];
n = 2; m = 2; d = 3;
t_final = 20;
t = 1:t_final;
[Cym,Cu,Cy] = dosac(d,Bbar,A);

[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
Wu = zeros(4,1);
Wy = [y(1);0];
ym(1:d) = ones(1,d);

for k = 1:t_final,
   ym(k+d) = 1;
   u(k) = Cym*ym(k+d)+Cu*Wu+Cy*Wy;
   y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
   x = uarma('state',x,u(k),d-1,Bbar,A);
   Wu = [u(k);Wu(1:3)];
   Wy = [y(k+1);Wy(1)];
end;
```

The result is plotted in Figure 7C.14. As seen in the figure, since the system parameters are known, the reference signal is caught immediately.

To see the response for a nonconstant reference signal, let us repeat the simulation for $y_m(t) = 1 + \sin\left(0.1t\right)$ and $t \in [1,100]$. The result is shown in Figure 7C.15. Tracking is perfect as before, since the system parameters are known. ■

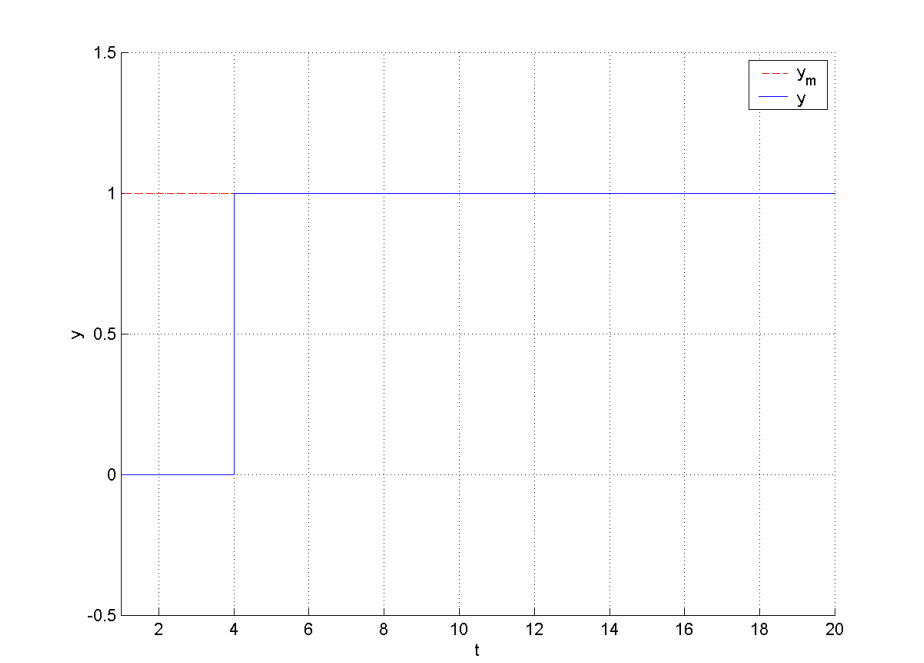

Figure 7C.14 *Time history of the output* y *for* $y_m(t) = 1$ *in Example* 7.5.11.

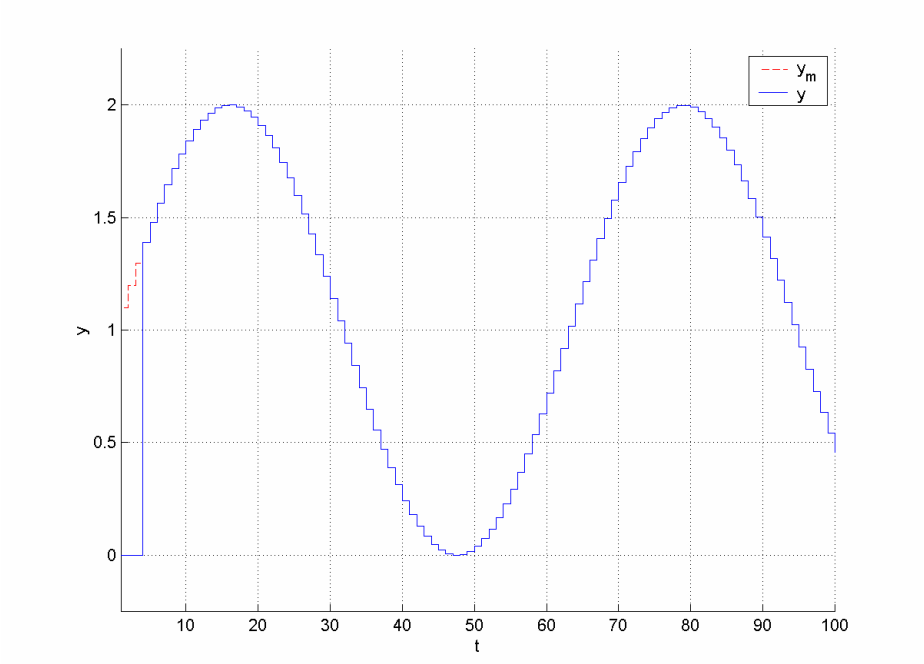

Figure 7C.15 Time *history of the output* y *for* $y_m(t) = 1 + \sin(0.1t)$ *in Example* 7.5.11.

Example 7.5.12 Consider the HDD servo system of Example 7.5.2. Let us perform the same control task using a one-step-ahead controller. Solving the Bezout equation

$$F\left(q^{-1}\right)A\left(q^{-1}\right)+q^{-1}G\left(q^{-1}\right)=1,$$

we obtain

$$G\left(q^{-1}\right)=2-q^{-1},$$
$$\beta\left(q^{-1}\right)=0.005+0.005q^{-1}.$$

Now let us consider minimization of three different cost functions to construct our controller, the cost function J_λ in (7.66), with λ equal to 0, $0.1\,\beta_0^2$, and β_0^2. Although for $\lambda=0$ the controller takes the form (7.64), let us use the generic form (7.68) for all of the three cases. Hence, plugging $\beta'(q^{-1})=q(\beta(q^{-1})-\beta_0)=5\times10^7$ into (7.68), we obtain

$$u[k]=\frac{0.005\left(y_m[k+1]-2y[k]-0.005u[k-1]\right)}{2.5\times10^{-5}+\lambda}.$$

The control coefficients can also be obtained by MATLAB directly, using the code

```
[Cym,Cu,Cy] = dosac(d,Bbar,A,lambda);
```

The simulation results for the plant and the controller above are shown in Figures 7C.16, 7C.17, and 7C.18. The trade-off between tracking and energy of the input signal can easily be seen in these plots. ■

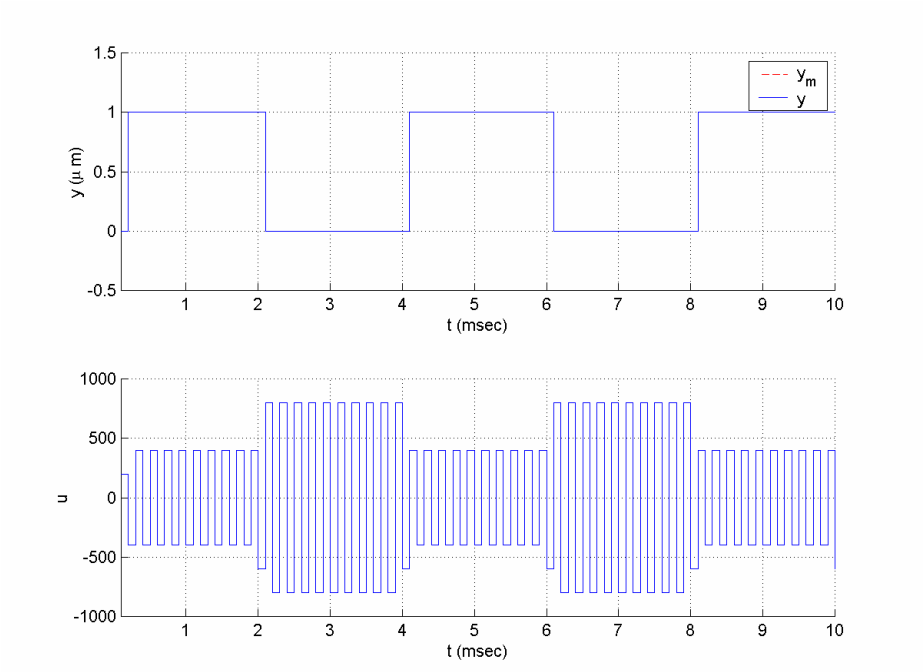

Figure 7C.16 *Time histories of the input and output signals for* $\lambda=0$ *in Example* 7.5.12.

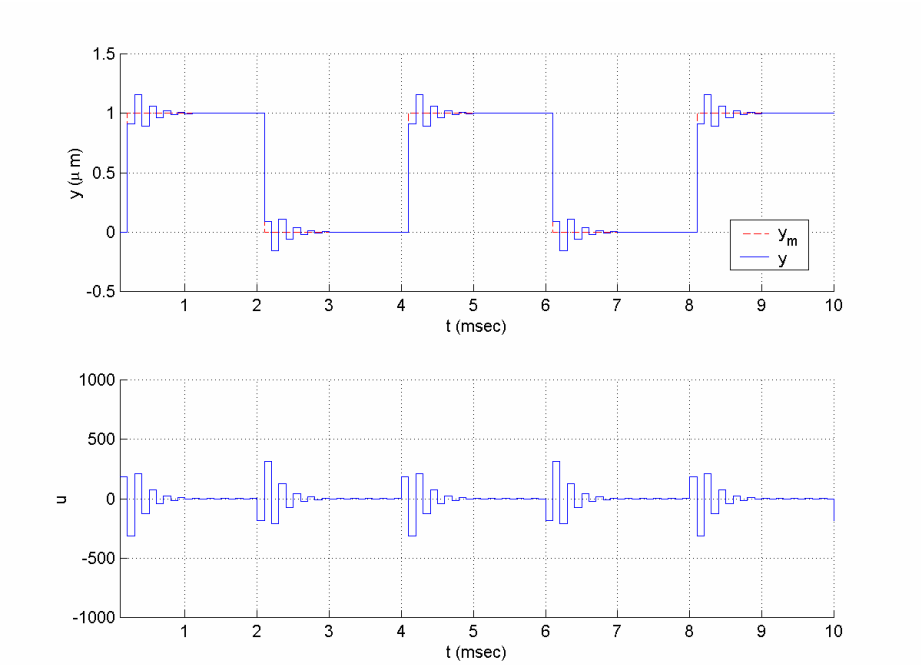

Figure 7C.17 *Time histories of the input and output signals for* $\lambda = 0.1\beta_0^2$ *in Example* 7.5.12.

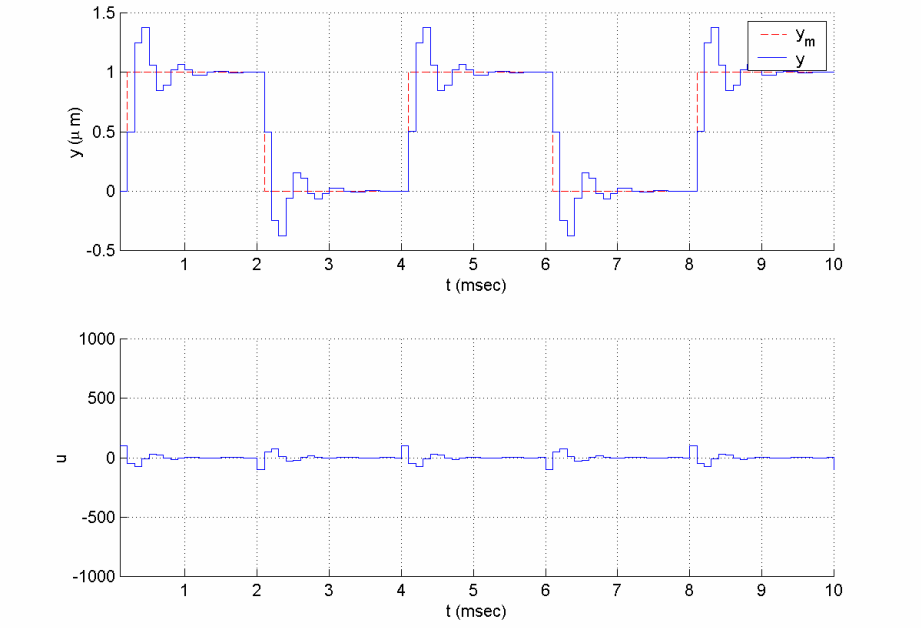

Figure 7C.18 *Time histories of the input and output signals for* $\lambda = \beta_0^2$ *in Example* 7.5.12.

The MATLAB function **`udosacdr`** used with some functions to generate the parameter estimates can be used to run the one-step-ahead adaptive control algorithm with direct approach, as demonstrated in Examples 7.5.13 and 7.5.14. In order to run the pure and weighted one-step-ahead adaptive control algorithms based on the linear control

form approach, the function **udosaclcf** incorporated into some functions to generate the parameter estimates can be used, as demonstrated in Examples 7.5.15 and 7.5.16. Similarly, the function **udosacidr** can be used to run the indirect one-step-ahead adaptive control scheme, as demonstrated in Example 7.5.17. If we desire any robustness modification to be applied as well, **urobust** can be used to perform this modification.

Example 7.5.13 Consider the plant and the control tasks of Example 7.5.11 to track constant and sinusoidal reference signals. Now assume that the plant parameters are not known; i.e., we know only that the plant is in the form

$$\begin{aligned}
&A\left(q^{-1}\right)y(t) = B\left(q^{-1}\right)u(t),\\
&A\left(q^{-1}\right) = 1 + a_2 q^{-1} + a_2 q^{-2},\\
&B\left(q^{-1}\right) = q^{-3}\overline{B}\left(q^{-1}\right) = q^{-3}\left(b_0 + b_1 q^{-1} + b_2 q^{-2}\right).
\end{aligned}$$

Let us repeat the same control task under this condition. From the model orders we see that the linear model to be used in parameter estimation will be

$$y(t+3) = \theta^{*T}\phi(t),$$

where

$$\begin{aligned}
&\phi = [u(t), u(t-1), \ldots, u(t-4), -y(t), -y(t-1)]^T,\\
&\theta^* = \left[\beta_0, \beta_1, \ldots, \beta_4, g_0, g_1\right]^T.
\end{aligned}$$

Hence, the control law will be

$$u(t) = \frac{1}{\theta_1(t)}[-\theta_2(t)u(t-1) - \cdots - \theta_5(t)u(t-4) + \theta_6(t)y(t) + \theta_7(t)y(t-1) + y_m(t+3)].$$

We can use the following RLS algorithm with start-up forgetting ($\beta(0) = 0.1, \beta_1 = 0.9$) to generate the parameter estimates:

$$\begin{aligned}
&\overline{\theta}\left(t\right) = \theta\left(t-1\right) + \frac{P\left(t-1\right)\phi\left(t-3\right)}{\beta\left(t\right) + \phi^T\left(t-3\right)P\left(t-1\right)\phi\left(t-3\right)}\left(y(t) - \theta^T\left(t-1\right)\phi\left(t-3\right)\right), \quad \theta(0) = [1,0,\ldots,0]^T,\\
&P\left(t\right) = \frac{1}{\beta\left(t\right)}\left[P\left(t-1\right) - \frac{P\left(t-1\right)\phi\left(t-3\right)\phi^T\left(t-3\right)P\left(t-1\right)}{\beta(t) + \phi^T\left(t-3\right)P\left(t-1\right)\phi\left(t-3\right)}\right], \quad P(0) = I,\\
&\beta\left(t\right) = \beta_1\beta\left(t-1\right) + 1 - \beta,\\
&\theta(t) = \Pr(\overline{\theta}(t)).
\end{aligned}$$

Choosing the projection mapping $\Pr(\cdot)$ so that $0.5 \le \theta_1 \le 100$ is guaranteed, the following code can be used to simulate the control of the plant for $t \in [1,100]$ and $y_m = 1$:

```
A = [1 -1.9 0.9];
Bbar = [1 0.5 0.25];
n = 2; m = 2; d = 3;

t_final = 100;
```

```
t = 1:t_final;

theta0 = zeros(7,1);
theta0(1) = 1;
Pj = uparproj('hyperplane',1,0.5,100);
P = eye(7);
ArgLS = urlsarg('startup forgetting',0.1,0.9);

x_pi = [theta0; P(:);0.1];
[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
[n_c, x_c] = udosacdr('init',[n d m]);
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
ym(1:d) = ones(1,d);

for k = 1:t_final,
   ym(k+d) = 1;
   phi = udosacdr('regressor', x_c, [n d m]);
   x_pi = udrls(x_pi, y(k), phi, 0, ArgLS, [], Pj);
   theta(:,k) = x_pi(1:7);
   u(k) = udosacdr('control',x_c,[y(k) ym(k+d)],[n d m],theta(:,k));
   y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
   x = uarma('state',x,u(k),d-1,Bbar,A);
   x_c = udosacdr('state', x_c, [u(k) y(k)], [n d m]);
end;
```

The result is plotted in Figures 7C.19 and 7C.20. The result of application of the same control structure for a sinusoidal y_m is plotted in Figure 7C.21. For both cases, although the transient behavior is poor, the parameter estimates converge to their true values, and the output converges to the desired value. ■

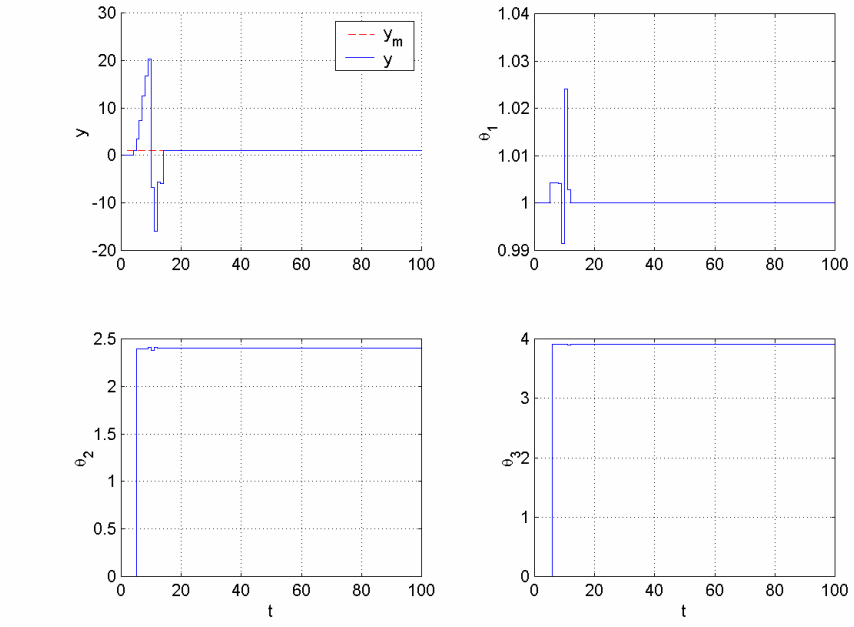

Figure 7C.19 *Time histories of the output* y *and parameter estimates* $\theta_1, \theta_2, \theta_2$ *for* $y_m(t) = 1$ *in Example* 7.5.13.

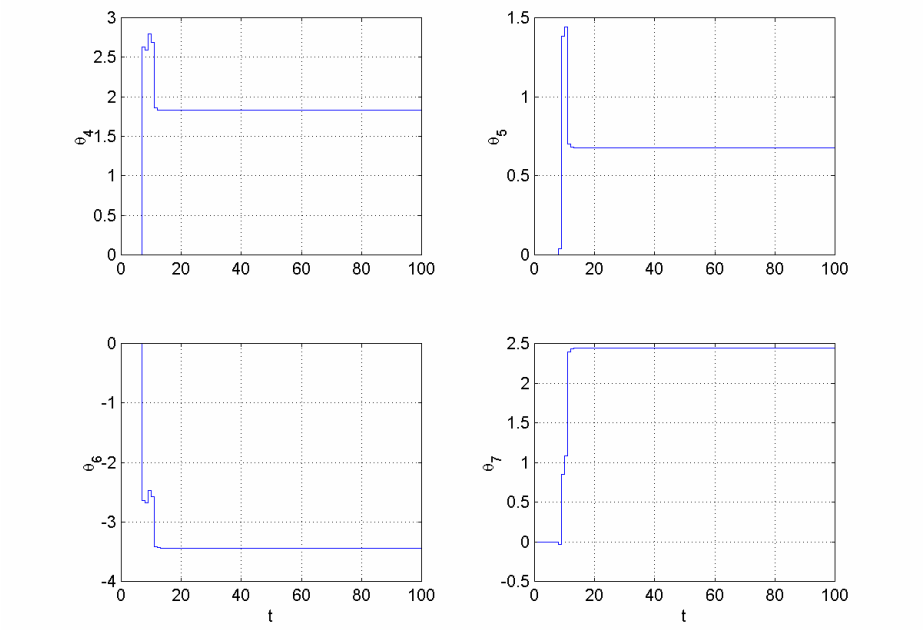

Figure 7C.20 *Time histories of the parameter estimates* $\theta_4,\ldots,\theta_7$ *for* $y_m(t)=1$ *in Example* 7.5.13.

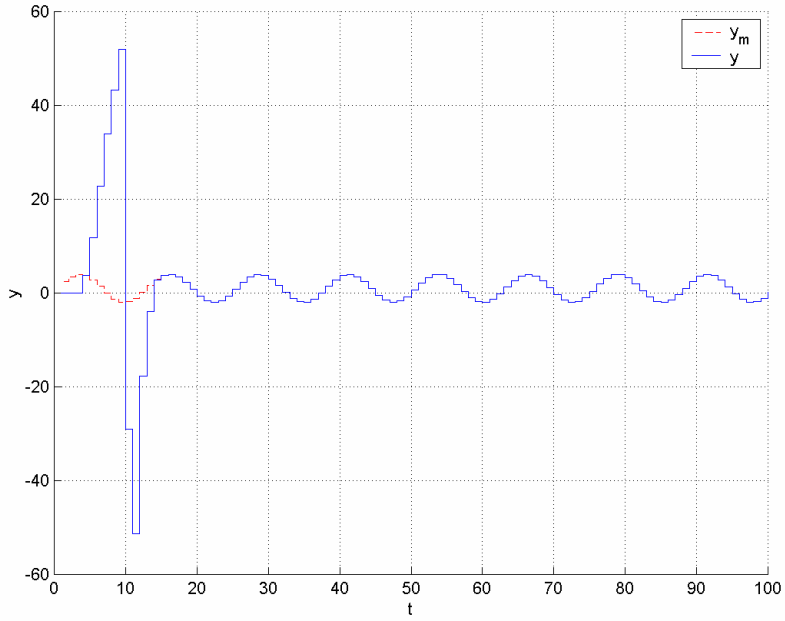

Figure 7C.21 *Time history of the output* y *for* $y_m(t)=1+3\sin(0.5t)$ *in Example* 7.5.13.

Example 7.5.14 Consider the HDD system of 7.5.2 and the one-step-ahead control design task of Example 7.5.12. This time, assume that the plant parameters are not known; i.e., we have only a model with unknown coefficients. Using the knowledge

about the orders n,d,m of the system, we deduce that the linear model to be used in parameter estimation is

$$y[k+1] = \theta^{*T}\phi[k],$$

where

$$\phi[k] = \left[u[k], u[k-1], -y[k], -y[k-1]\right]^T,$$

and the control law is

$$u[k] = \frac{1}{\theta_1[k]}\left(-\theta_2[k]u[k-1] + \theta_3[k]y[k] + \theta_4[k]y[k-1] + y_m[k+1]\right).$$

Using the RLS algorithm with start-up forgetting ($\beta(0) = 0.7, \beta_1 = 0.99$) as before, we generate the parameter estimates

$$\begin{aligned}
\bar{\theta}[k] &= \theta[k-1] + \frac{P[k-1]\phi[k-1]}{\beta[k] + \phi^T[k-1]P[k-1]\phi[k-1]}\left(y[k] - \theta^T[k-1]\phi[k-1]\right),\\
\theta[0] &= [0.003, 0, 0, 0]^T,\\
P[k] &= \frac{1}{\beta[k]}\left[P[k-1] - \frac{P[k-1]\phi[k-1]\phi^T[k-1]P[k-1]}{\beta[k] + \phi^T[k-1]P[k-1]\phi[k-1]}\right], \quad P[0] = I,\\
\beta[k] &= \beta_1\beta[k-1] + 1 - \beta,\\
\theta[k] &= \Pr(\bar{\theta}[k]).
\end{aligned}$$

Choosing the projection mapping $\Pr(\cdot)$ so that $0.003 \le \theta_1 \le 1$ is guaranteed, the simulation results shown in Figures 7C.22 and 7C.23 are obtained. Although the tracking is perfect at steady state, there is a very large overshoot in the response. ■

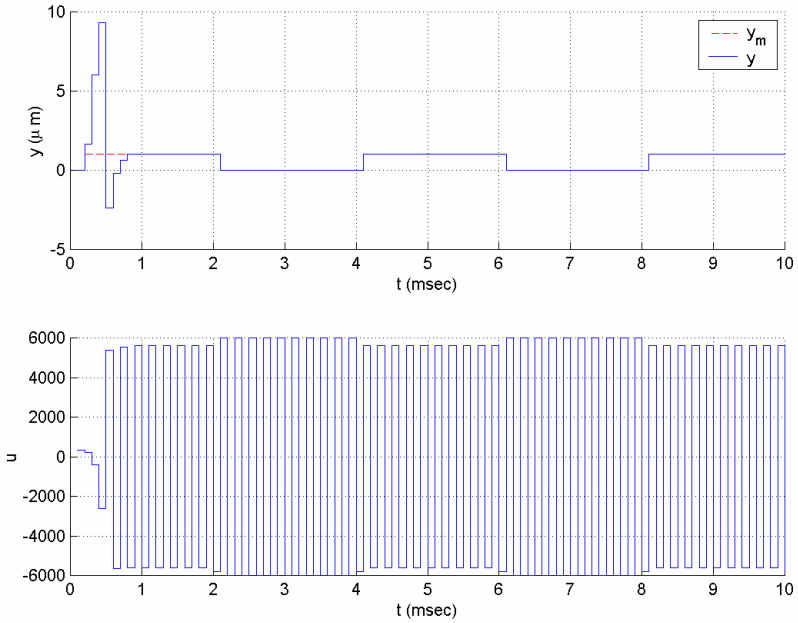

Figure 7C.22 *Time histories of the input and output signals in Example* 7.5.14.

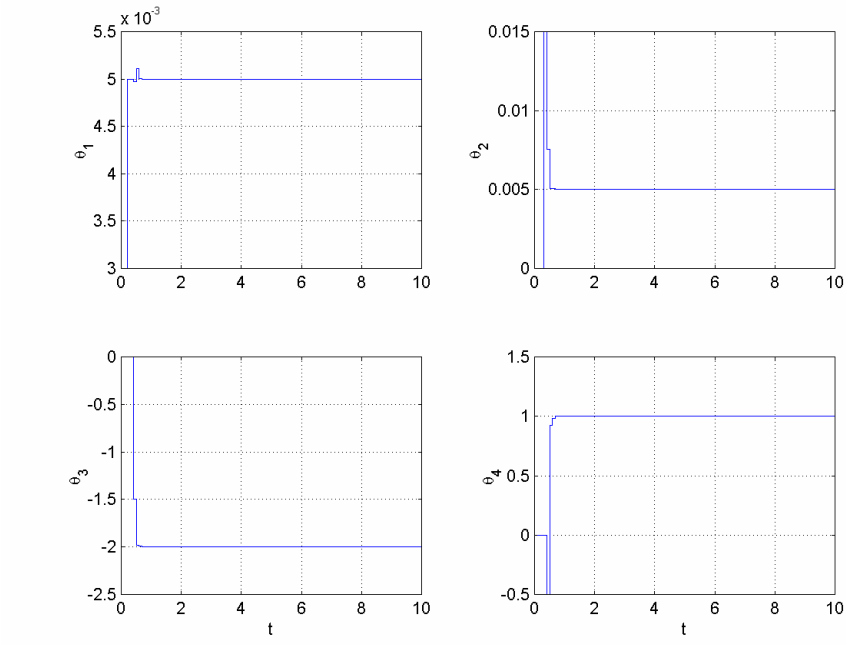

Figure 7C.23 *Time histories of the parameter estimates in Example* 7.5.14.

Example 7.5.15 Let us repeat Example 7.5.13 using the linear control form approach instead of the direct approach. The control scheme will be based on the parametric model

$$u(t) = \theta^{*T}\phi(t),$$

where

$$\phi(t) = [y(t+3), -y(t), -y(t-1), -u(t-1), \ldots, -u(t-4)]^T,$$
$$\theta^* = \left[\frac{1}{\beta_0} \quad \bar{g}_0 \quad \bar{g}_1 \quad \bar{\beta}_1 \quad \cdots \quad \bar{\beta}_4\right]^T.$$

The control law is given by

$$u(t) = \theta^T(t)\bar{\phi}(t),$$

where

$$\bar{\phi}(t) = [y_m(t+3), -y(t), -y(t-1), -u(t-1), \ldots, -u(t-4)]^T.$$

If we use the RLS algorithm with start-up forgetting as before, the adaptive law to generate the parameter estimates will be as follows:

$$\bar{\theta}(t) = \theta(t-1) + \frac{P(t-1)\phi(t-3)}{\beta(t) + \phi^T(t-3)P(t-1)\phi(t-3)}\left(u(t-3) - \theta^T(t-1)\phi(t-3)\right), \quad \theta(0) = [1,0,\ldots,0]^T,$$
$$P(t) = \frac{1}{\beta(t)}\left[P(t-1) - \frac{P(t-1)\phi(t-3)\phi^T(t-3)P(t-1)}{\beta(t) + \phi^T(t-3)P(t-1)\phi(t-3)}\right], \quad P(0) = I,$$
$$\beta(t) = \beta_1\beta(t-1) + 1 - \beta_1,$$
$$\theta(t) = \Pr(\bar{\theta}(t)).$$

Choosing the projection mapping $\Pr(\cdot)$ so that $0.1 \le \theta_1 \le 10$ is guaranteed, the following code can be used to simulate the control of the plant for $t \in [1,100]$ and $y_m = 1$:

```
A = [1 -1.9 0.9];
Bbar = [1 0.5 0.25];
n = 2; m = 2; d = 3;

t_final = 100;
t = 1:t_final;

theta0 = zeros(7,1);
theta0(1) = 1;
Pj = uparproj('hyperplane',1,0.1,10);
P = eye(7);
ArgLS = urlsarg('startup forgetting',0.1,0.9);
x_pi = [theta0; P(:);0.1];
%x_pi = [theta0;P(:)];
[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
[n_c, x_c] = udosaclcf('init',[n d m]);
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
ym(1:d) = ones(1,d);

for k = 1:t_final,
   ym(k+d) = 1;
   [u_old phi] = udosaclcf('regressor',x_c,y(k),[n d m]);
```

```
    x_pi = udrls(x_pi, u_old, phi, 0, ArgLS, [], Pj);
    %x_pi = udproj('orth',x_pi, u_old, phi, 0, Arg, Pj);
    theta(:,k) = x_pi(1:7);
    u(k) = udosaclcf('control',x_c,[y(k) ym(k+d)],[n d m],theta(:,k));
    y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
    x = uarma('state',x,u(k),d-1,Bbar,A);
    x_c = udosaclcf('state', x_c, [u(k) y(k)], [n d m]);
end;
```

The simulation results are plotted in Figures 7C.24, 7C.25, and 7C.26. The behaviors are comparable to those with the direct approach . ■

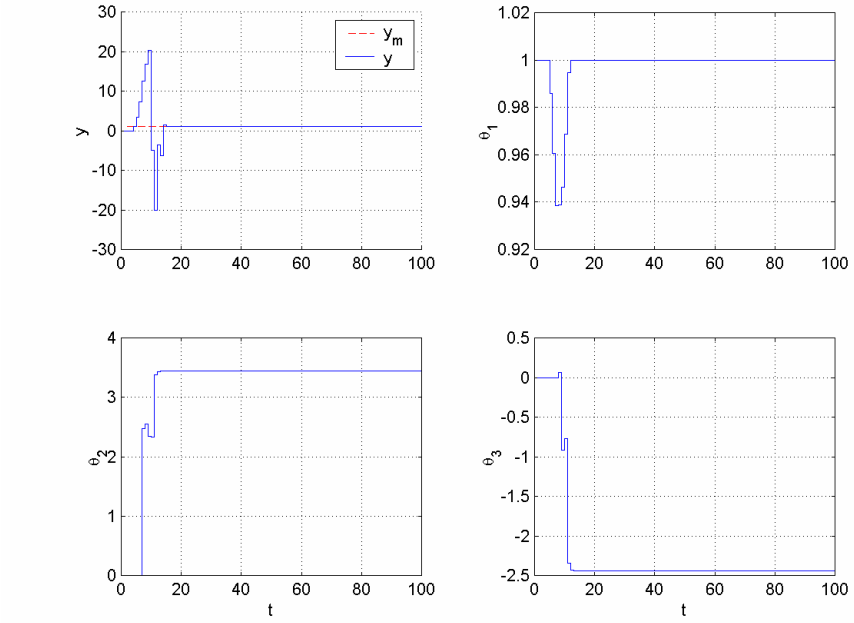

Figure 7C.24 *Time histories of the output* y *and parameter estimates* $\theta_1, \theta_2, \theta_2$ *for* $y_m(t) = 1$ *in Example* 7.5.15.

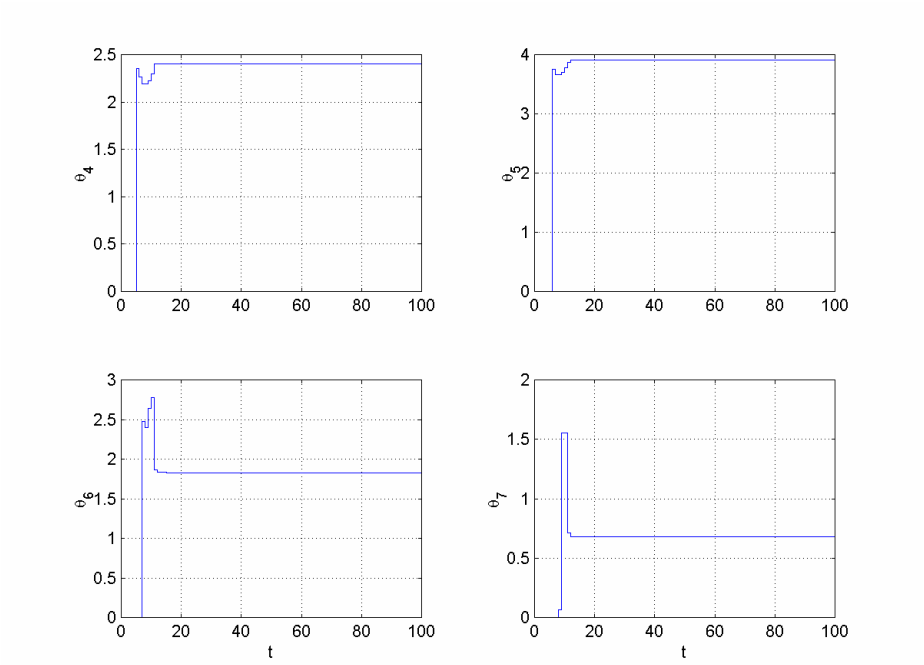

Figure 7C.25 *Time histories of the parameter estimates* $\theta_4,\ldots,\theta_7$ *for* $y_m(t)=1$ *in Example* 7.5.15.

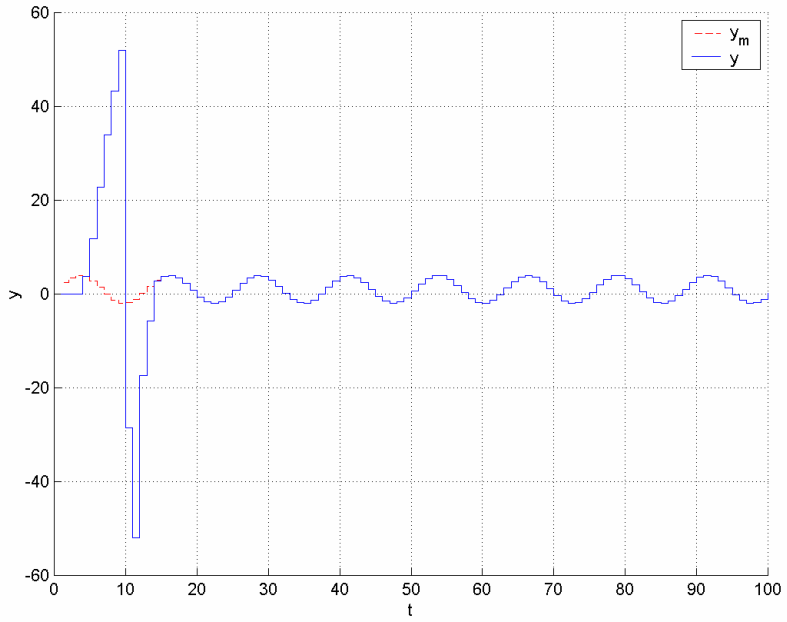

Figure 7C.26 *Time history of the output* y *for* $y_m(t)=1+3\sin(0.5t)$ *in Example* 7.5.15.

Example 7.5.16 Let us repeat Example 7.5.13 using the linear control form approach instead of the direct approach with $\gamma=0,\gamma=0.0005$, and $\gamma=0.005$. The control scheme will be based on the parametric model

$$u[k] = \theta^{*T}\phi[k],$$

with

$$\phi[k] = [y[k+1], -y[k], -y[k-1], -u[k-1]]^T,$$
$$\theta^* = \left[\frac{1}{\beta_0}, \overline{g}_0, \overline{g}_1, \overline{\beta}_1\right]^T.$$

The control law is given by

$$u[k] = \theta^T[k]\overline{\phi}[k],$$

where

$$\overline{\phi}[k] = [y_m[k+1], -y[k], -y[k-1], -u[k-1]]^T.$$

We can use the RLS algorithm with start-up forgetting of Example 7.5.14 without any change other than choosing the initial estimate vector as $[500,0,0,0]^T$ and noting the difference between the first entries of θ^* for the two cases. Choosing the projection mapping $\Pr(\cdot)$ so that $1 \le \theta_1 \le 1000$ is guaranteed, the following code can be used to simulate the control of the HDD system for $t \in [1,100]$:

```
A = [1 -2 1];
Bbar = 0.005*[1 1];
n = 2; m = 1; d = 1;
Ts = 0.1;

t_final = 100;
t = Ts*(1:t_final);

gamma = 0.005;
theta0 = zeros(4,1);
theta0(1) = 500;
Pj = uparproj('hyperplane',1,1,1000);
P = eye(4);
ArgLS = urlsarg('startup forgetting',0.1,0.9);
x_pi = [theta0; P(:);0.1];
[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
[n_c, x_c] = udosaclcf('init',[n d m]);
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
ym(1) = 0;

for k = 1:t_final,
   ym(k+d) = mod(floor(k/20)+1,2);
   [u_old phi] = udosaclcf('regressor', x_c, y(k), [n d m],gamma);
   x_pi;
   x_pi = udrls(x_pi, u_old, phi, 0, ArgLS, [], Pj);
   theta(:,k) = x_pi(1:4);
   u(k) = udosaclcf('control',x_c,[y(k) ym(k+d)],[n d
m],theta(:,k),gamma);
   y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
   x = uarma('state',x,u(k),d-1,Bbar,A);
```

```
   x_c = udosaclcf('state', x_c, [u(k) y(k)], [n d m]);
end;
```

The results are plotted in Figures 7C.27, 7C.28, and 7C.29. The transient behavior is comparable to that of Example 7.5.14, and the effect of the weighting coefficient is comparable to the nonadaptive case, which was examined in Example 7.5.12. ■

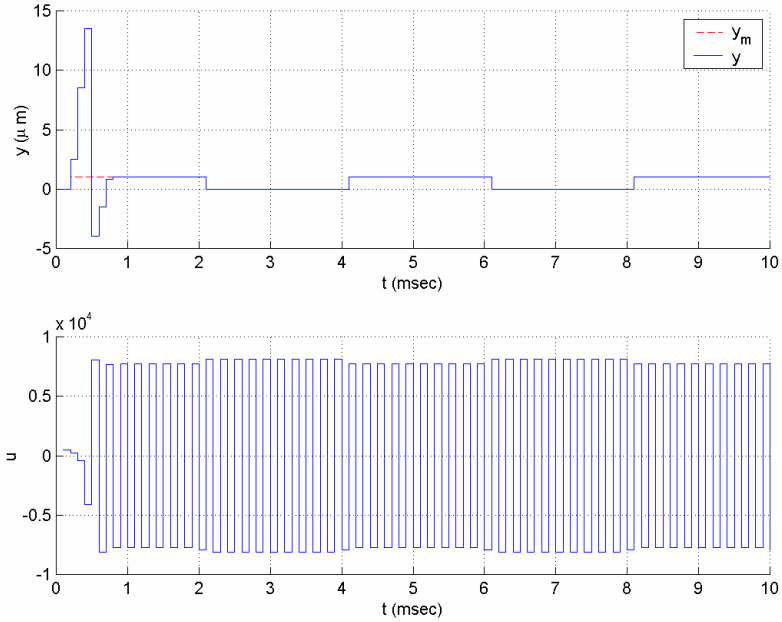

Figure 7C.27 *Time histories of the input and output signals for* $\gamma = 0$ *in Example* 7.5.16.

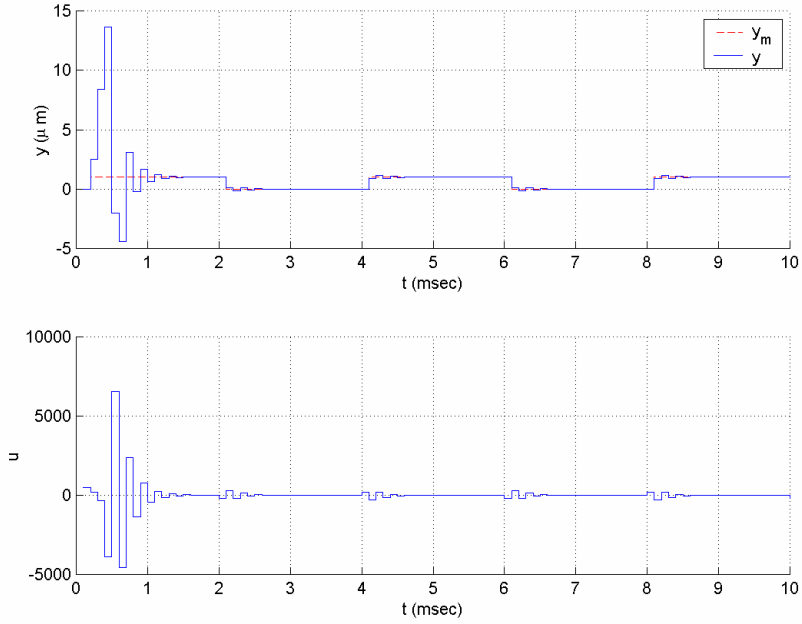

Figure 7C.28 *Time histories of the input and output signals for* $\gamma = 0.0005$ *in Example* 7.5.16.

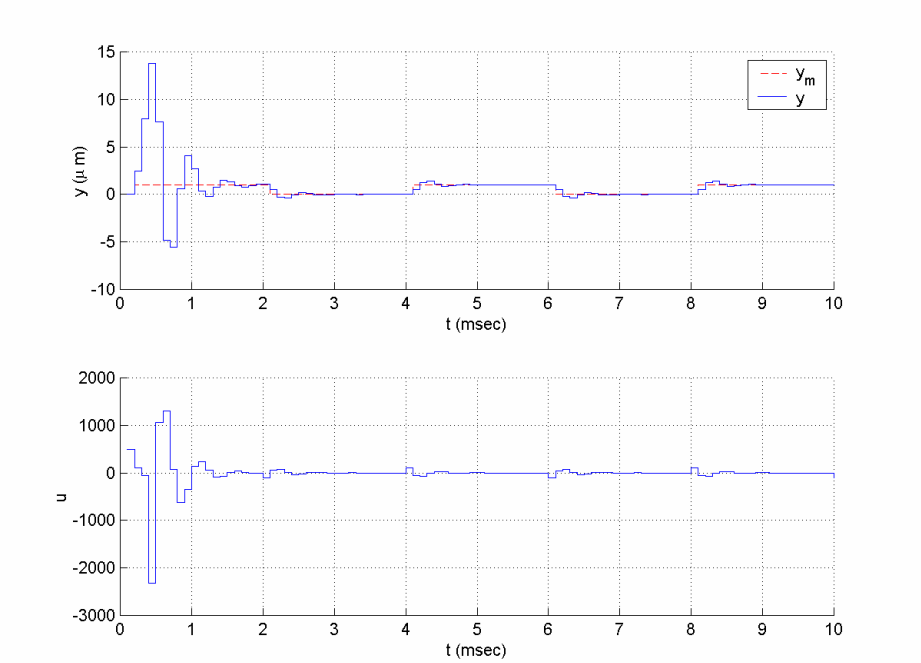

Figure 7C.29 *Time histories of the input and output signals for* $\gamma = 0.005$ *in Example* 7.5.16.

Example 7.5.17 Let us repeat the adaptive one-step-ahead control task of Examples 7.5.13 and 7.5.15 using the indirect approach. Estimation of the system parameters will be based on the linear parametric model

$$y(t) = \theta^{*T}\phi(t)$$

with

$$\theta^* = [b_0, b_1, b_2, a_1, a_2]^T,$$
$$\phi(t) = [u(t-3), u(t-4), u(t-5), -y(t-1), -y(t-2)]^T.$$

The parameter estimate vector $\theta(t) = [\hat{b}_0(t), \hat{b}_1(t), \hat{b}_2(t), \hat{a}_1(t), \hat{a}_2(t)]^T$ can be obtained using any PI algorithm with parameter projection to keep $\hat{b}_0$ away from zero, e.g., $0.1 \le \theta_1 \le 10$. For example, the RLS algorithm with start-up forgetting can be used again as follows:

$$\overline{\theta}(t) = \theta(t-1) + \frac{P(t-1)\phi(t)}{\beta(t) + \phi^T(t)P(t-1)\phi(t)}\left(y(t) - \theta^T(t-1)\phi(t)\right), \quad \theta(0) = [1, 0, \ldots, 0]^T,$$

$$P(t) = \frac{1}{\beta(t)}\left[P(t-1) - \frac{P(t-1)\phi(t)\phi^T(t)P(t-1)}{\beta(t) + \phi^T(t)P(t-1)\phi(t)}\right], \quad P(0) = I,$$

$$\beta(t) = \beta_1\beta(t-1) + 1 - \beta_1,$$

$$\theta(t) = \left[\hat{b}_0(t), \hat{b}_1(t), \hat{b}_2(t), \hat{a}_1(t), \hat{a}_2(t)\right]^T = \Pr(\overline{\theta}(t)).$$

Having the parameter estimates, the control signal can be generated as follows:

$$u(t) = \frac{y_m(t+d) - \hat{G}(q^{-1},t)y(t) - \hat{\beta}'(q^{-1},t)u(t-1)}{\hat{\beta}_0(t)},$$

where

$$\begin{aligned}
&\hat{A}(q^{-1},t) = 1 + \hat{a}_1(t)q^{-1} + \hat{a}_2(t)q^{-2},\\
&\hat{\bar{B}}(q^{-1},t) = \hat{b}_0(t) + \hat{b}_1(t)q^{-1} + \hat{b}_2(t)q^{-2},\\
&\hat{F}(q^{-1},t)\hat{A}(q^{-1},t) + q^{-d}\hat{G}(q^{-1},t) = 1,\\
&\hat{\beta}(q^{-1},t) = \hat{\beta}_0(t) + \hat{\beta}_1(t)q^{-1} + \cdots + \hat{\beta}_4(t)q^{-4} = \hat{\bar{B}}(q^{-1},t)\hat{G}(q^{-1},t),\\
&\hat{\beta}'(q^{-1},t) = q\left(\hat{\beta}(q^{-1},t) - \hat{\beta}_0\right) = \hat{\beta}_1(t) + \hat{\beta}_2(t)q^{-1} + \cdots + \hat{\beta}_4(t)q^{-3}.
\end{aligned}$$

The following code can be used to simulate the control of the plant for $t \in [1,100]$ and $y_m = 1$:

```
A = [1 -1.9 0.9];
Bbar = [1 0.5 0.25];
n = 2; m = 2; d = 3;

t_final = 100;
t = 1:t_final;

theta0 = zeros(5,1);
theta0(1) = 1;
Pj = uparproj('hyperplane',1,0.1,10);
P = eye(5);
ArgLS = urlsarg('startup forgetting',0.1,0.9);
x_pi = [theta0; P(:);0.1];
[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
[n_c, x_c] = udosacidr('init',[n d m]);
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
ym(1:d) = ones(1,d);

for k = 1:t_final,
   ym(k+d) = 1;
   phi = udosacidr('regressor',x_c,[n d m]);
   x_pi = udrls(x_pi, y(k), phi, 0, ArgLS, [], Pj);
   theta(:,k) = x_pi(1:5);
   u(k) = udosacidr('control',x_c,[y(k) ym(k+d)],[n d m],theta(:,k));
   y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
   x = uarma('state',x,u(k),d-1,Bbar,A);
   x_c = udosacidr('state', x_c, [u(k) y(k)], [n d m]);
end;
```

The simulation results are plotted in Figures 7C.30 and 7C.31. The behaviors are comparable to those with the previous approaches. ■

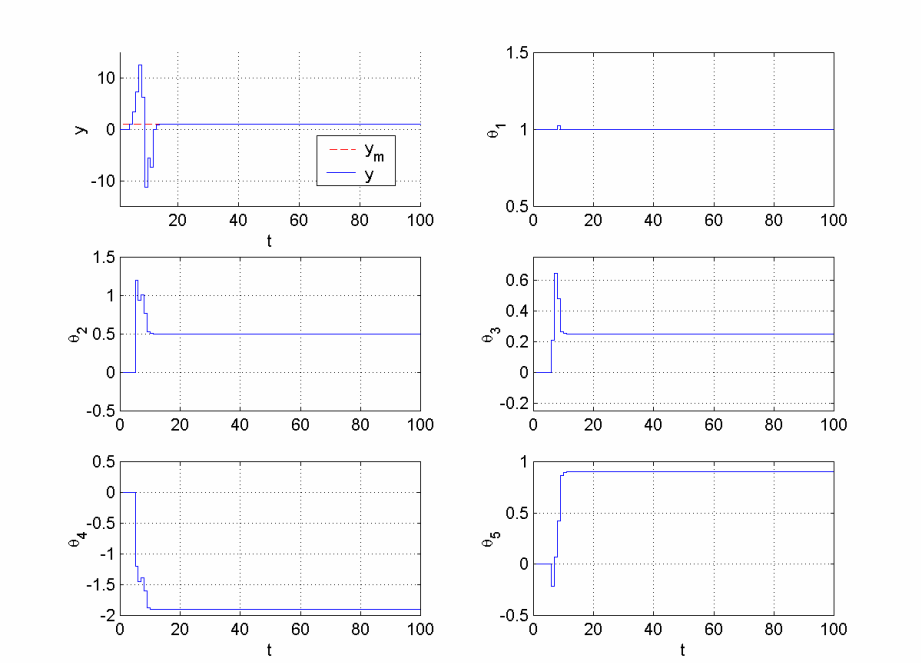

Figure 7C.30 *Time histories of the output* y *and parameter estimates* θ_i *for* $y_m(t) = 1$ *using indirect one-step-ahead adaptive control in Example* 7.5.17.

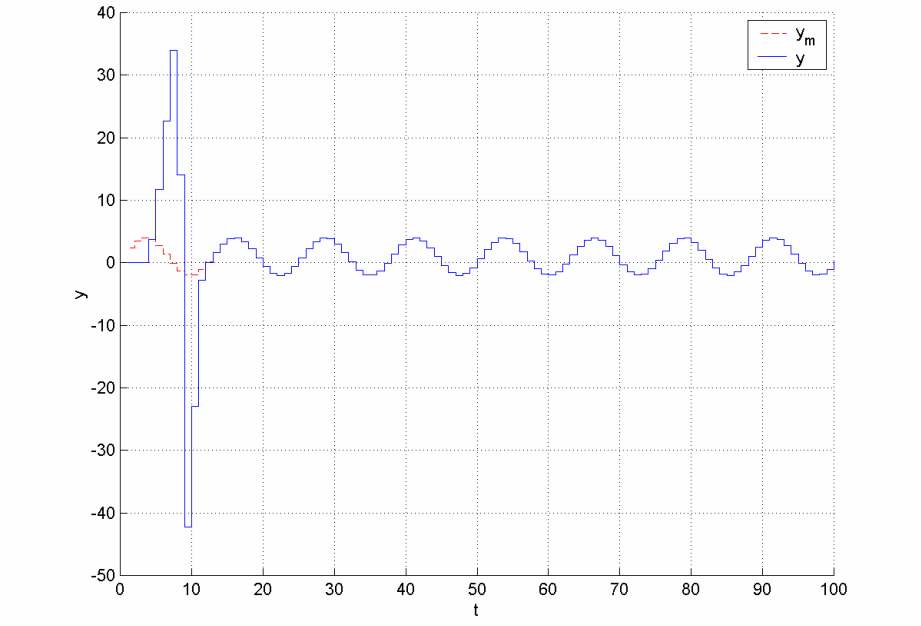

Figure 7C.31 *Time history of the output* y *for* $y_m(t) = 1 + 3\sin(0.5t)$ *using indirect one-step-ahead adaptive control in Example* 7.5.17.

7.5.4 APPC

The MATLAB functions **dppcdo**, **udppcdo**, **dppcsf**, **udppcsf**, **dppcimp**, and **udppcimp** can be used to run the three PPC algorithms presented in this chapter, as demonstrated in Examples 7.5.18 and 7.5.19.

Example 7.5.18 Consider the ARMA model

$$\left(1-2q^{-1}+q^{-2}\right)y = q^{-1}(0.5+0.5q^{-1})u,$$

which corresponds to a double integrator sampled with a sampling rate of 1 Hz. Let us design a pole placement controller for this system so that the closed loop is stable and the output tracks $y_m = 1$. First, we need to fix a monic Hurwitz polynomial of order $2n^* - 1 = 3$. Let this polynomial be $A^*(q^{-1}) = (1-0.1q^{-1})(1+0.2q^{-1}+0.02q^{-2})$. Next, we need to solve the Bezout equation

$$A\left(q^{-1}\right)L\left(q^{-1}\right)+B\left(q^{-1}\right)P\left(q^{-1}\right) = A^*\left(q^{-1}\right),$$

where $A(q^{-1}) = 1-2q^{-1}+q^{-2}$ and $B(q^{-1}) = q^{-1}(0.5+0.5q^{-1})$. The solution is found as

$$L(q^{-1}) = 1+0.7745q^{-1}, \quad P(q^{-1}) = 2.651-1.553q^{-1}.$$

Choosing $M(q^{-1}) = P(q^{-1})$, the final control law is found as

$$\left(1+0.7745q^{-1}\right)u(t) = -\left(2.651-1.553q^{-1}\right)\left(y(t)-y_m(t+d)\right),$$

or explicitly as

$$u(t) = 2.651(y_m(t+1)-y(t))-1.553(y_m(t)-y(t-1))-0.7745u(t-1).$$

This system and the control scheme can be simulated using the following code:

```
A = [1 -2 1]; Bbar = [0.5 0.5];
n = 2; m = 2; d = 1;
As = conv([1 0.2 0.02],[1 -0.1]);

t_final = 20; t = 1:t_final;

[Cym,Cu,Cy] = dppcdo(d,Bbar,A,As);

[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
Wu = 0;
Wy = [y(1);0];
Wym = ones(2,1);
ym(1) = 1;

for k = 1:t_final,
   ym(k+d) = 1;
   Wym = [ym(k+d);Wym(1)];
   u(k) = Cym*Wym+Cu*Wu+Cy*Wy;
```

```
    y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
    x = uarma('state',x,u(k),d-1,Bbar,A);
    Wu = u(k);
    Wy = [y(k+1);Wy(1)];
end;
```

The results are plotted in Figure 7C.32. ■

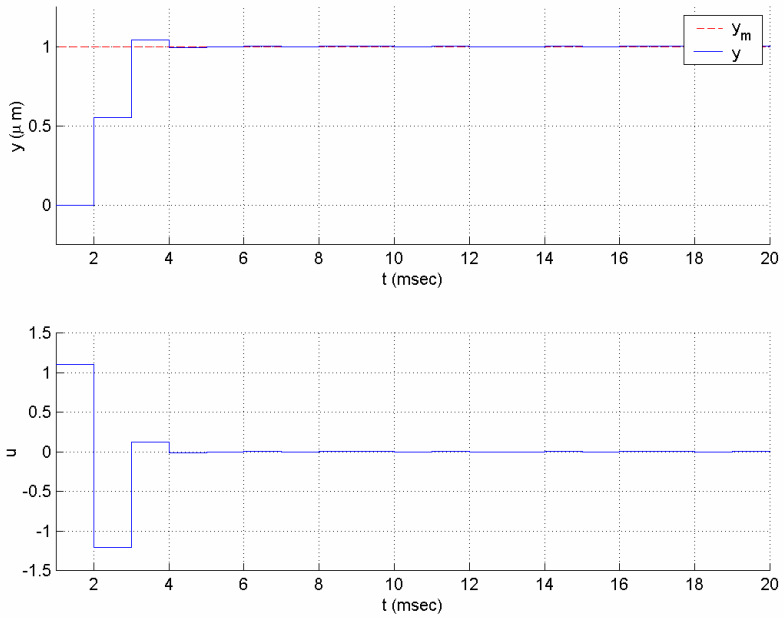

Figure 7C.32 *Time histories of the input and output signals in Example* 7.5.18.

Example 7.5.19 Consider the ARMA model of Example 7.5.18 in the existence of a periodic disturbance so that the disturbed plant can be modeled as

$$\left(1+q^{-1}+q^{-2}\right)\left(1-2q^{-1}+q^{-2}\right)y=\left(1+q^{-1}+q^{-2}\right)q^{-1}(0.5+0.5q^{-1})u\,.$$

Let us again design a pole placement controller for this system so that the closed loop is stable and the output tracks $y_m=1$. First, we need to fix a monic Hurwitz polynomial of order $2n^*-1=7$. Let this polynomial be $A^*(q^{-1})=(1-0.1q^{-1})^7$. Noting that the reference signal satisfies $(1-q^{-1})y_m=0$, we need to solve the Bezout equation

$$\bar{L}\left(q^{-1}\right)S\left(q^{-1}\right)D\left(q^{-1}\right)\tilde{A}\left(q^{-1}\right)+P\left(q^{-1}\right)\tilde{B}\left(q^{-1}\right)=A^*\left(q^{-1}\right),$$

where $\tilde{A}(q^{-1})=1-2q^{-1}+q^{-2}$, $\tilde{B}(q^{-1})=q^{-1}(0.5+0.5q^{-1})$, $S(q^{-1})=1-q^{-1}$, and $D(q^{-1})=1+q^{-1}+q^{-2}$. The solution is found as

$$\begin{aligned}&\bar{L}(q^{-1})=1+0.7564q^{-1},\\&P(q^{-1})=1.0872+0.3585q^{-1}+0.0587q^{-2}-2.5389q^{-3}+1.5128q^{-4}\,.\end{aligned}$$

Hence the control law is found as

$$\left(1+0.7564q^{-1}-q^{-3}-0.7564q^{-4}\right)u(t)$$
$$=-\left(1.0872+0.3585q^{-1}+0.0587q^{-2}-2.5389q^{-3}+1.5128q^{-4}\right)\left(y(t)-y_m(t+d)\right).$$

This system and the control scheme can be simulated using the following code:

```
Atilde = [1 -2 1];
Btilde = [0.5 0.5];
D = [1 1 1];
A = conv(Atilde,D);
Bbar = conv(Btilde,D);
n = 2; m = 2; d = 1;

As = [1 -0.1];
for l = 1:6,
   As = conv(As,[1 -0.1]);
end

t_final = 20;
t = 1:t_final;

S = [1 -1];
[Cym,Cu,Cy] = dppcimp(d,Btilde,Atilde,D,S,As);

[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
Wu = zeros(7,1);
Wy = [y(1);zeros(4,1)];
Wym = ones(5,1);
ym(1) = 1;

for k = 1:t_final,
   ym(k+d) = 1;
   Wym = [ym(k+d);Wym(1:4)];
   u(k) = Cym*Wym+Cu*Wu+Cy*Wy;
   y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
   x = uarma('state',x,u(k),d-1,Bbar,A);
   Wu = [u(k);Wu(1:6)];
   Wy = [y(k+1);Wy(1:4)];
end;
```

The results are plotted in Figure 7C.33. ■

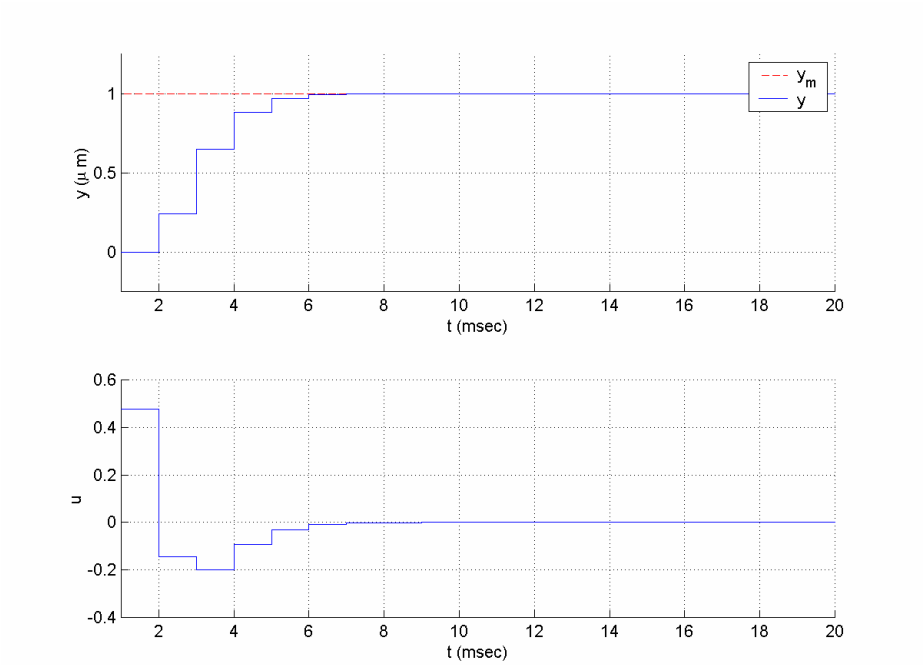

Figure 7C.33 *Time histories of the input and output signals in Example* 7.5.19.

As explained in section 7.4, discrete-time adaptive pole placement controllers can be constructed by combining a PPC scheme with one of the PI algorithms presented in Chapter 4. The implemetation of discrete-time APPC algorithms using the Adaptive Control Toolbox is demonstrated in the following examples.

Example 7.5.20 Consider the ARMA system and the control task of Example 7.5.18. Assume that the coefficients of the ARMA model are unknown. Let us apply indirect APPC (difference operator approach) to perform the control task. Choosing the desired closed-loop characteristics polynomial as $A^*(q^{-1}) = (1-0.1q^{-1})(1+0.2q^{-1}+0.02q^{-2})$, as before, we can construct the control scheme based on the linear parametric model $y(t) = \theta^{*T}\phi(t)$ with $\theta^* = [b_0, b_1, a_1, a_2]^T$ and $\phi(t) = [u(t-1), u(t-2), -y(t-1), -y(t-2)]^T$ as follows (see next page):

$$\bar{\theta}(t) = \theta(t-1) + \frac{P(t-1)\phi(t)}{\beta(t) + \phi^T(t)P(t-1)\phi(t)}\left(y(t) - \theta^T(t-1)\phi(t)\right), \quad \theta(0) = [1,0,0,0]^T,$$

$$P(t) = \frac{1}{\beta(t)}\left[P(t-1) - \frac{P(t-1)\phi(t)\phi^T(t)P(t-1)}{\beta(t) + \phi^T(t)P(t-1)\phi(t)}\right], \quad P(0) = I,$$

$$\beta(t) = \beta_1\beta(t-1) + 1 - \beta_1,$$

$$\theta(t) = \left[\hat{b}_0(t), \hat{b}_1(t), \hat{a}_1(t), \hat{a}_2(t)\right]^T = \Pr(\bar{\theta}(t)),$$

$$\hat{A}(q^{-1},t) = 1 + \hat{a}_1(t)q^{-1} + \hat{a}_2(t)q^{-2},$$

$$\hat{B}(q^{-1},t) = q^{-1}(\hat{b}_0(t) + \hat{b}_1(t)q^{-1}),$$

$$\hat{A}(q^{-1},t)\hat{L}(q^{-1},t) + \hat{B}(q^{-1},t)\hat{P}(q^{-1},t) = A^*(q^{-1}),$$

$$\hat{L}(q^{-1},t)u(t) = -\hat{P}(q^{-1},t)y(t) + \hat{P}(q^{-1},t)y_m(t+1).$$

Above, the LS algorithm with start-up forgetting and parameter projection is used for parameter estimation as before. Other algorithms could be used as well. The system and the control scheme presented above can be simulated using the following code:

```
A = [1 -2 1];
Bbar = [0.5 0.5];
n = 2; m = 1; d = 1;
As = conv([1 0.2 0.02],[1 -0.1]);

t_final = 100;
t = 1:t_final;

theta0 = zeros(4,1);
theta0(1) = 1;
Pj = uparproj('hyperplane',1,0.1,10);
P = eye(4);
ArgLS = urlsarg('startup forgetting',0.1,0.9);
x_pi = [theta0; P(:);0.1];
[nstate,x] = uarma('init',d-1,Bbar,A);  %y(t)/u(t-1)
[n_c, x_c] = udppcdo('init',[n d m]);
y(1) = uarma('output',x,0,d-1,Bbar,A);
x = uarma('state',x,0,d-1,Bbar,A);
ym(1) = 1;

for k = 1:t_final,
   ym(k+d) = 1;
   phi = udppcdo('regressor',x_c,[n d m]);
   x_pi = udrls(x_pi, y(k), phi, 0, ArgLS, [], Pj);
   theta(:,k) = x_pi(1:4);
   u(k) = udppcdo('control',x_c,[y(k) ym(k+d)],[n d m],theta(:,k),As);
   y(k+1) = uarma('output',x,u(k),d-1,Bbar,A);
   x = uarma('state',x,u(k),d-1,Bbar,A);
   x_c = udppcdo('state', x_c, [u(k) y(k) ym(k+d)], [n d m]);
end;
```

The results are plotted in Figure 7C.34. ■

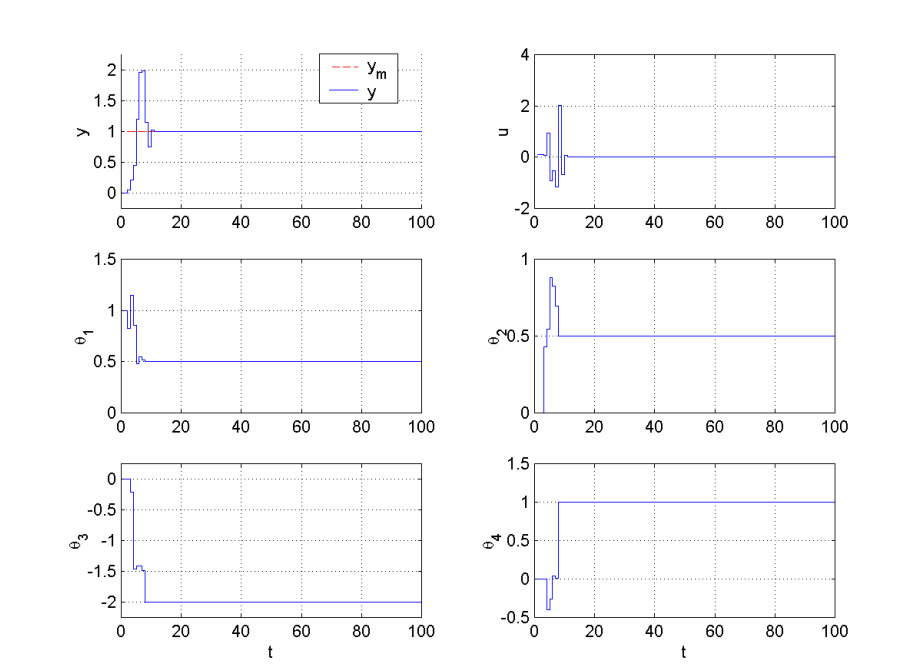

Figure 7C.34 *Time histories of the input and output signals and the parameter estimates in Example* 7.5.20.

Bibliography

[1] G. C. GOODWIN AND K. S. SIN, *Adaptive Filtering Prediction and Control*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[2] P. A. IOANNOU and J. Sun, *Robust Adaptive Control*, Prentice-Hall, Englewood Cliffs, NJ, 1996; also available at

http://www-rcf.usc.edu/~ioannou/Robust_Adaptive_Control.htm.

[3] K. J. ASTROM AND B. WITTENMARK, *Adaptive Control,* Addison-Wesley, Reading, MA, 1995.