EXAMPLE 6.2–1: Approximate Computed-Torque Controller

We wish to design and simulate an approximate computed-torque controller for the two-link arm given in Figure 6.2.1 (see Chapter 2 for the two-link revolute robot arm dynamics). Assuming that the friction is negligible, the link lengths are exactly known, and the masses m_1 and m_2 are known to be in the regions 0.8±0.05 kg and 2.3±0.1 kg, respectively, a possible approximated computed-torque controller can be written as

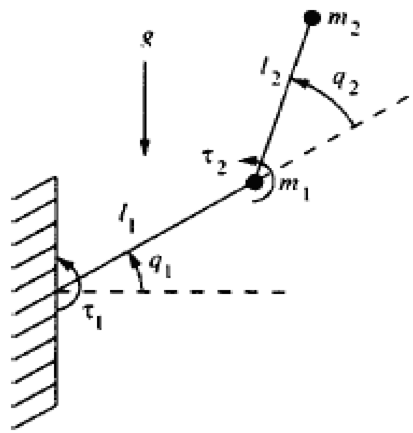

Figure 6.2.1: Two-link planar arm.

$$\begin{aligned}\tau_1 = &\left(2\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2 + (\hat{m}_1 + \hat{m}_2)\, l_1^2\right)(\ddot{q}_{d1} + k_{v1}\dot{e}_1 + k_{p1}e_1) \\ &+ \left(\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2\right)(\ddot{q}_{d2} + k_{v2}\dot{e}_2 + k_{p2}e_2) - \hat{m}_2 l_1 l_2 s_2 \dot{q}_2^2 \\ &- 2\hat{m}_2 l_1 l_2 s_2 \dot{q}_1 \dot{q}_2 + \hat{m}_2 l_2 g c_{12} + (\hat{m}_1 + \hat{m}_2)\, l_1 g c_1\end{aligned} \tag{1}$$

$$\begin{aligned}\tau_2 = &\left(\hat{m}_2 l_2^2 + \hat{m}_2 l_1 l_2 c_2\right)(\ddot{q}_{d1} + k_{v1}\dot{e}_1 + k_{p1}e_1) + \hat{m}_2 l_2 g c_{12} \\ &+ \hat{m}_2 l_2^2 (\ddot{q}_{d2} + k_{v2}\dot{e}_2 + k_{p2}e_2) + \hat{m}_2 l_1 l_2 s_2 \dot{q}_1^2,\end{aligned} \tag{2}$$

where l_1=l_2=1 m and g is the gravitational constant. We choose $\hat{m}_1$=0.85 kg and $\hat{m}_2$=2.2 kg since the actual values are assumed to be unknown. After substituting the control law above into the two-link robot dynamics, we can form the error system

$$\ddot{e} + K_v \dot{e} + K_p e = \hat{M}^{-1}(q)\, W(q, \dot{q}, \ddot{q})\, \tilde{\varphi}, \tag{3}$$

where $\hat{M}^{-1}(q)$ is the inverse of the inertia matrix $M(q)$ with m_1 and m_2 replaced by $\hat{m}_1$, and $\hat{m}_2$, respectively. The matrix $W(q, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, sometimes called the *regression* matrix [Craig 1985], is a 2×2 matrix given by

$$W(q, \dot{q}, \ddot{q}) = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}, \tag{4}$$

where

$$\begin{aligned} W_{11} =& l_1^2 \ddot{q}_1 + l_1 g c_1, \\ W_{12} =& l_2^2(\ddot{q}_1 + \ddot{q}_2) + l_1 l_2 c_2 (2\ddot{q}_1 + \ddot{q}_2) + l_1^2 \ddot{q}_1 - l_1 l_2 s_2 \dot{q}_2^2 \\ & - 2 l_1 l_2 s_2 \dot{q}_1 \dot{q}_2 + l_2 g c_{12} + l_1 g c_1, \\ W_{21} =& 0, \\ W_{22} =& l_1 l_2 c_2 \ddot{q}_1 + l_1 l_2 s_2 \dot{q}_1^2 + l_2 g c_{12} + l_2^2 (\ddot{q}_1 + \ddot{q}_2). \end{aligned}$$

The vector $\tilde{\varphi}$ called the parameter error vector, is a 2×1 vector given by

$$\tilde{\varphi} = \begin{bmatrix} \tilde{\varphi}_1 \\ \tilde{\varphi}_2 \end{bmatrix}, \tag{5}$$

where

$$\tilde{\varphi}_1 = m_1 - \hat{m}_1$$

and

$$\tilde{\varphi}_2 = m_2 - \hat{m}_2.$$

The associated tracking error 2×1 vector and 2×2 gain matrices in (3) are given by

$$e = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \quad K_v = \begin{bmatrix} k_{v1} & 0 \\ 0 & k_{v2} \end{bmatrix}, \quad \text{and} \quad K_p = \begin{bmatrix} k_{p1} & 0 \\ 0 & k_{p2} \end{bmatrix}.$$

For m_1=0.8 kg and m_2=2.3 kg, the approximate computed torque controller (1)–(2) was simulated with $q(0)=\dot{q}(0)=0$, with the controller gains set at

$$K_p = K_v = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \quad (6)$$

and with a desired trajectory of

(7)

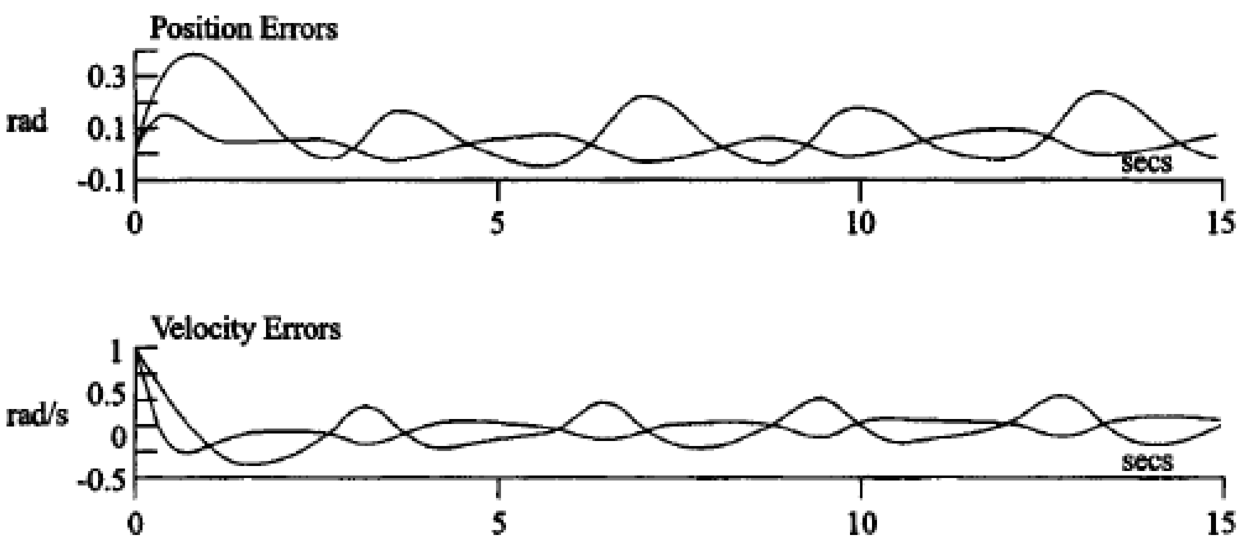

Figure 6.2.2: Simulation of approximate computed-torque controller.

Torque Controller:

$$\tau = \hat{M}(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + \hat{V}_m(q,\dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q})$$

Update Rule:

$$\dot{\hat{\varphi}} = \Gamma W^T(q,\dot{q},\ddot{q})\hat{M}^{-1}(q)B^T P\mathbf{e}$$

where

$$\mathbf{e} = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, \quad B = \begin{bmatrix} O_n \\ I_n \end{bmatrix}, \quad A = \begin{bmatrix} O_n & I_n \\ -K_p & -K_v \end{bmatrix}$$

$$W(q,\dot{q},\ddot{q})\hat{\varphi} = \hat{M}(q)\ddot{q} + \hat{V}_m(q,\dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q})$$

$$A^T P + PA = -Q$$

for some positive-definite, symmetric matrices P and Q.

Stability:

Tracking error vector $\mathbf{e}$ is asymptotically stable.

Restrictions:

Parameter resetting method is required. Measurement of $\ddot{q}$ is required.

EXAMPLE 6.2–2: Adaptive Computed-Torque Controller

It is desired to design and simulate the adaptive computed-torque controller given in Table 6.2.1 for the two-link arm given in Figure 6.2.1.

Assuming that the friction is negligible and that the link lengths are exactly known, the adaptive computed-torque controller can be written in the same form as that given in Example 6.2.1, with the exception that *we* must find the update rules for $\hat{m}_1$ and $\hat{m}_2$. That is, we use Equations (1) and (2) in Example 6.2.1 for the joint torque control and then formulate the update rule for m_1 and m_2 according to Table 6.2.1.

For simplicity, in this example we select the servo gains as

$$K_v = k_v I_n \quad \text{and} \quad K_p = k_p I_n, \tag{1}$$

where k_v and k_p are positive, scalar constants and for this case I_n is the 2×2 identity matrix. We propose that the matrix P in Table 6.2.1 be selected as

$$P = \begin{bmatrix} P_1 I_n & P_2 I_n \\ P_2 I_n & P_3 I_n \end{bmatrix} = 1/2 \begin{bmatrix} (K_p + 1/2 k_v) I_n & 1/2 I_n \\ 1/2 I_n & I_n \end{bmatrix}. \tag{2}$$

Note that P is symmetric, and that it is positive definite if k_v is selected to be greater than 1 (see the Gerschgorin Theorem in Chapter 1). To see if our selection of P gives a positive-definite Q, perform the matrix operation

$$A^T P + PA = -Q \tag{3}$$

$$Q = \begin{bmatrix} 1/2 k_p I_n & O_n \\ O_n & (K_v + 1/2)\, I_n \end{bmatrix}. \tag{4}$$

Since we have already restricted $k_v > 1$, it can be verified that Q is a positive definite, symmetric matrix. We note here that the process of finding a positive definite, symmetric P and Q for the general Lyapunov approach is not always an easy task.

Now that we have found an appropriate P, we can formulate the adaptive update rule given in Table 6.2.1. The associated parameter estimate vector is

$$\hat{\varphi} = \begin{bmatrix} \hat{m}_1 \\ \hat{m}_2 \end{bmatrix}$$

$$\begin{aligned}\dot{\hat{m}}_1 =& \gamma_1[(W_{11}MI_{11} + W_{21}MI_{21})(P_2e_1 + P_3\dot{e}_1) \\ &+ (W_{11}MI_{21} + W_{21}MI_{22})(P_2e_2 + P_3\dot{e}_2)]\end{aligned} \tag{5}$$

and

$$\begin{aligned}\dot{\hat{m}}_2 =& \gamma_2[(W_{12}MI_{11} + W_{22}MI_{21})(P_2e_1 + P_3\dot{e}_1) \\ &+ (W_{12}MI_{21} + W_{22}MI_{22})(P_2e_2 + P_3\dot{e}_2)],\end{aligned} \tag{6}$$

where

$$\begin{aligned}MI_{11} &= \frac{1}{\Delta}\left(\hat{m}_2 l_2^2\right), \\ MI_{21} &= -\frac{1}{\Delta}\left(\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2\right), \\ MI_{22} &= \frac{1}{\Delta}\left(2\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2 + (\hat{m}_1 + \hat{m}_2)\, l_1^2\right), \\ \Delta &= \left(2\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2 + (\hat{m}_1 + \hat{m}_2)\, l_1^2\right)\left(\hat{m}_2 l_2^2\right) - \left(\hat{m}_2 l_2^2 + \hat{m}_2 l_1 l_2 c_2\right)^2,\end{aligned}$$

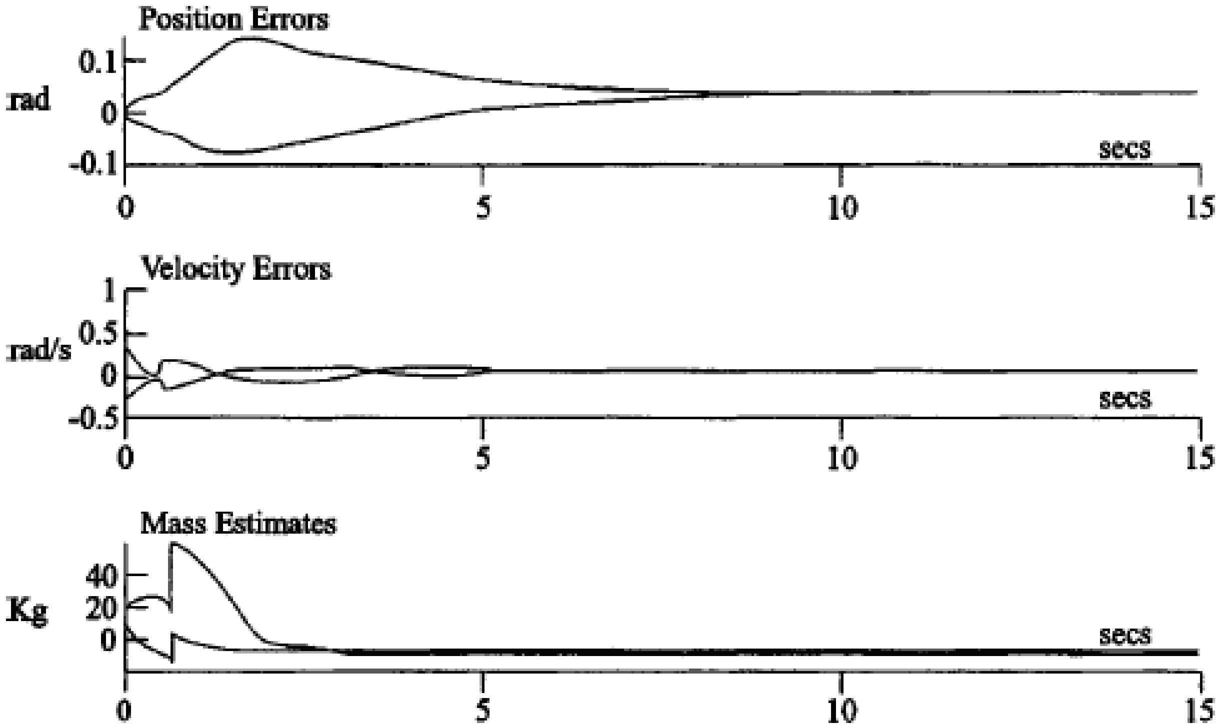

Figure 6.2.4: Simulation of the adaptive computed-torque controller.

EXAMPLE 6.3–1: Adaptive Inertia-Related Controller

We wish to design and simulate the adaptive inertia-related controller given in Table 6.3.1 for the two-link arm given in Figure 6.2.1. Assuming that the friction is negligible and the link lengths are exactly known, the adaptive inertia-related torque controller can be written as

$$\tau_1 = Y_{11}\hat{m}_1 + Y_{12}\hat{m}_2 + k_{v1}\dot{e}_1 + k_{v1}\lambda_1 e_1 \tag{1}$$

$$\tau = Y(\cdot)\,\hat{\varphi} + K_v\dot{e} + K_v\Lambda e \tag{2}$$

In the expression for the control torques, the regression matrix $Y(\cdot)$ is given by

$$Y(\ddot{q}_d, \dot{q}_d, q_d, q, \dot{q}) = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}, \tag{3}$$

where

$$Y_{11} = l_1^2(\ddot{q}_{d1} + \lambda_1\dot{e}_1) + l_1 g c_1, \tag{4}$$

$$\begin{aligned} Y_{12} = {} & (l_2^2 + 2l_1l_2c_2 + l_1^2)(\ddot{q}_{d1} + \lambda_1\dot{e}_1) \\ & + (l_2^2 + l_1l_2c_2)(\ddot{q}_{d2} + \lambda_2\dot{e}_2) - l_1l_2s_2\dot{q}_2(\dot{q}_{d1} + \lambda_1 e_1) \\ & - l_1l_2s_2(\dot{q}_1 + \dot{q}_2)(\dot{q}_{d2} + \lambda_2 e_2) + l_2 g c_{12} + l_1 g c_1, \end{aligned} \tag{5}$$

$$Y_{21} = 0, \tag{6}$$

and

$$\begin{aligned} Y_{22} = &\left(l_1 l_2 c_2 + l_2^2\right)\left(\ddot{q}_{d1} + \lambda_1 \dot{e}_1\right) + l_2^2\left(\ddot{q}_{d2} + \lambda_2 \dot{e}_2\right) \\ &- l_1 l_2 s_2 \dot{q}_1\left(\dot{q}_{d1} + \lambda_1 e_1\right) + l_2 g c_{12}. \end{aligned} \tag{7}$$

Formulating the adaptive update rule as given in Table 6.3.1, the associated parameter estimate vector is

$$\hat{\varphi} = \begin{bmatrix} \hat{m}_1 \\ \hat{m}_2 \end{bmatrix}$$

with the adaptive update rules

$$\dot{\hat{m}}_1 = \gamma_1 \left[Y_{11}\left(\lambda_1 e_1 + \dot{e}_1\right) + Y_{21}\left(\lambda_2 e_2 + \dot{e}_2\right)\right] \tag{8}$$

and

$$\dot{\hat{m}}_2 = \gamma_2 \left[Y_{12}\left(\lambda_1 e_1 + \dot{e}_1\right) + Y_{22}\left(\lambda_2 e_2 + \dot{e}_2\right)\right]. \tag{9}$$

For m_1=0.8 kg and m_2=2.3 kg, the adaptive inertia-related controller was simulated with $k_{v1}=k_{v2}=10$, $\lambda_1=\lambda_2=2.5$, $\lambda_1=\lambda_2=20$, $\hat{m}_1(0)=0$, $\hat{m}_2(0)=0$, and with

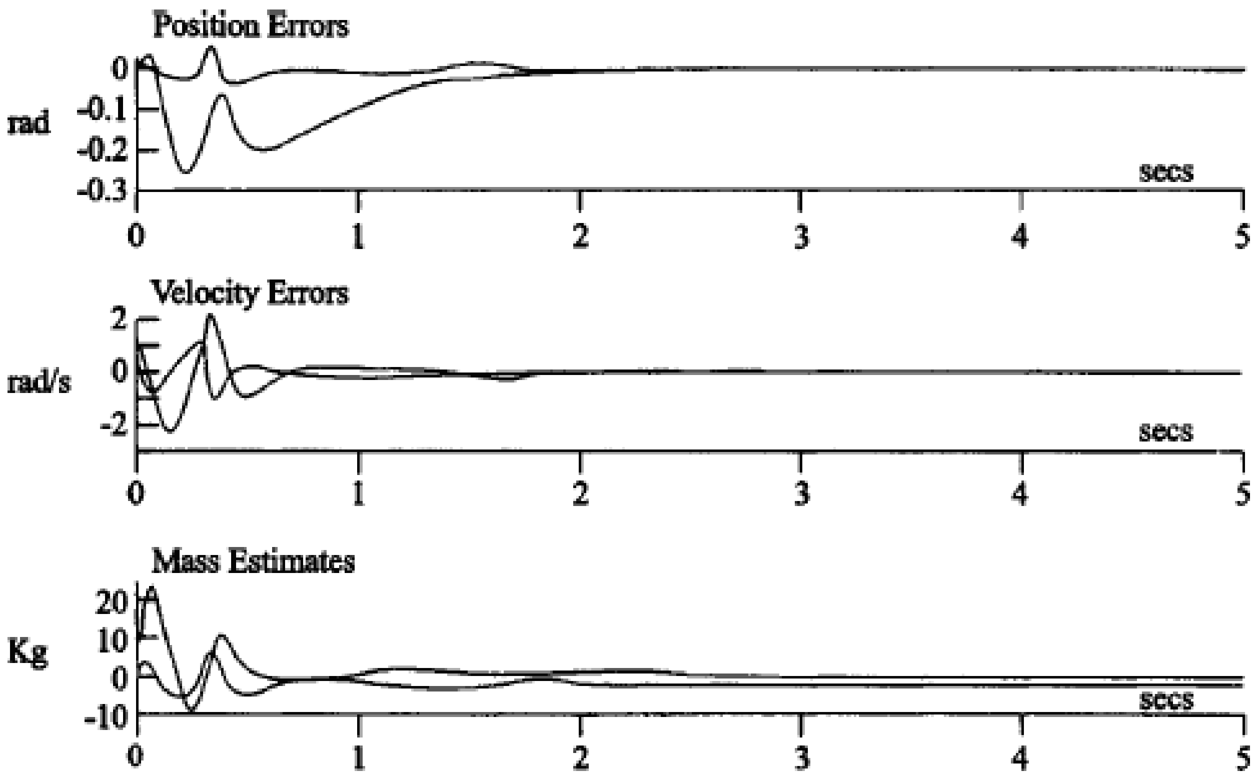

Figure 6.3.2: Simulation of the adaptive inertia-related controller.

Table 6.3.1: Adaptive Inertia-Related Controller

Torque Controller:

$$\tau = Y(\cdot)\,\hat{\varphi} + K_v\dot{e} + K_v\Lambda e$$

Update Rule:

$$\dot{\hat{\varphi}} = \Gamma Y^T(\cdot)(\Lambda e + \dot{e})$$

where

$$Y(\cdot)\hat{\varphi} = \hat{M}(q)(\ddot{q}_d + \Lambda\dot{e}) + \hat{V}_m(q,\dot{q})(\dot{q}_d + \Lambda e) + \hat{G}(q) + \hat{F}(\dot{q})$$

Stability:

Tracking error e and $\dot{e}$ are asymptotically stable. Parameter estimate $\hat{\varphi}$ is bounded.

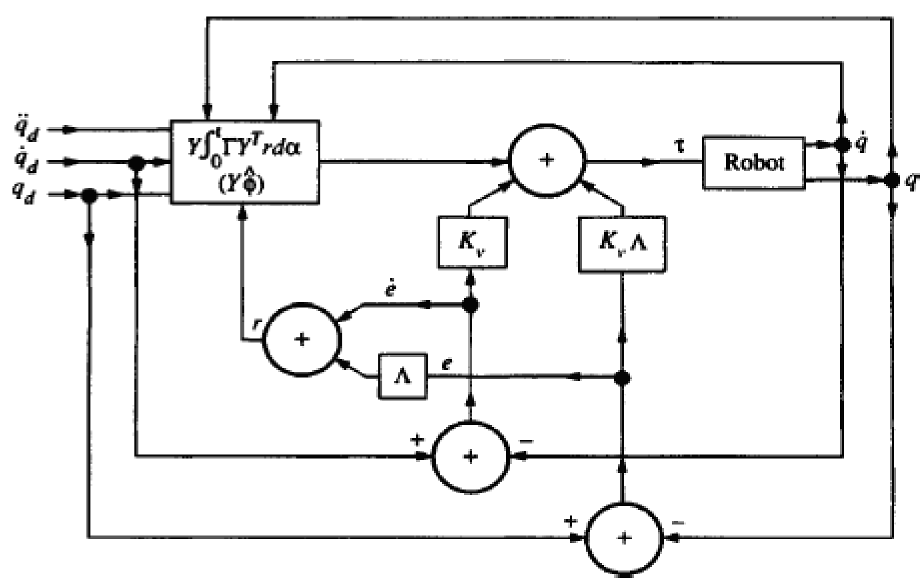

Figure 6.3.1: Block diagram of the adaptive inertia-related controller.

EXAMPLE 6.6–2: Least-Squares Estimator for a One-Link Robot Arm

Using the dynamics of the one-link robot arm given in Example 6.5.1, it is desired to find the least-squares estimator given by (6.6.16) and (6.6.17). Since the number of unknown parameters is two, define the matrix P to be

$$P = \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix}. \tag{1}$$

$$\tau = m\ddot{q} + b\dot{q}, \tag{1}$$

$$\tau = m\ddot{q} + b\dot{q},$$

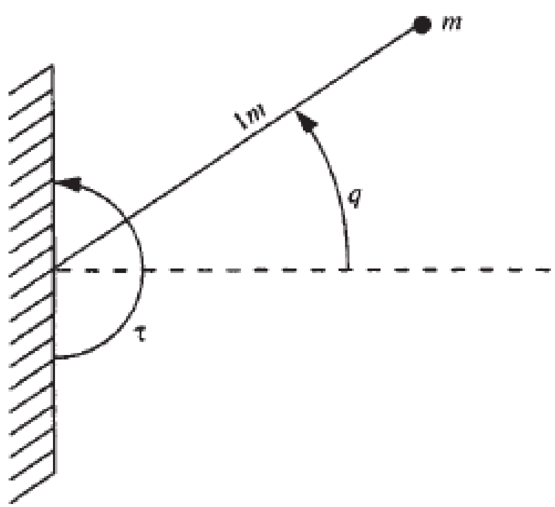

Figure 6.5.1: One-link revolute arm.

Utilizing the filtered regression matrix from Example 6.6.1, we have

$$W_f(q,\dot{q}) = \begin{bmatrix} W_{f11} & W_{f12} \end{bmatrix}, \tag{2}$$

where

$$W_{f11} = \left(\dot{f} * \dot{q}\right) + \dot{q} - f\dot{q}(0) \quad \text{and} \quad W_{f12} = \dot{f} * \dot{q}.$$

Using (6.6.17), it is easy to see that the matrix P should be updated in the following manner:

$$\dot{P}_1 = -\left(W_{f11}P_1 + W_{f12}P_2\right)^2 \tag{3}$$

$$\dot{P}_2 = -W_{f11}^2 P_1 P_2 - W_{f12}^2 P_2 P_3 - W_{f11}W_{f12}\left(P_1P_3 + P_2^2\right), \tag{4}$$

and

$$\dot{P}_3 = -\left(W_{f11}P_2 + W_{f12}P_3\right)^2. \tag{5}$$

Now using (6.6.16), the parameter update rules are

$$\dot{\hat{m}} = (P_1 W_{f11} + P_2 W_{f12})\,\tilde{\tau}_f \tag{6}$$

$$\dot{\hat{b}} = (P_2 W_{f11} + P_3 W_{f12})\,\tilde{\tau}_f, \tag{7}$$

where, from (6.6.16), τ_f is given by

$$\tilde{\tau}_f = \tau_f - W_{f11}\hat{m} - W_{f12}\hat{b}. \tag{8}$$

For insight into how the least-squares estimation method extracts parameter information, we now show how (6.6.18) is obtained. Utilizing (6.6.13) and the fact that the parameters are constant, we write (6.6.16) as

$$\dot{\tilde{\varphi}} = -PW_f^T(\cdot)\,W_f(\cdot)\,\tilde{\varphi}. \tag{6.6.20}$$

Using the matrix identity $\dot{P} = -P\dot{P}^{-1}P$ we can write (6.6.17) as

$$\dot{P}^{-1} = W_f^T(\cdot)\,W_f(\cdot)\,. \tag{6.6.21}$$

Substituting (6.6.21) into (6.6.20) yields the differential equation

$$\dot{\tilde{\varphi}} = -P\dot{P}^{-1}\tilde{\varphi}. \tag{6.6.22}$$

We claim that

$$\tilde{\varphi} = -PP^{-1}(0)\,\tilde{\varphi}(0) \tag{6.6.23}$$

is the solution to (6.6.22). This fact can be verified by substituting (6.6.23) into the right-hand and left-hand sides of (6.6.22). That is, we obtain

$$-\dot{P}P^{-1}(0)\,\tilde{\varphi}(0) = P\dot{P}^{-1}PP^{-1}(0)\,\tilde{\varphi}(0)\,; \tag{6.6.24}$$

therefore, (6.6.23) is the solution. Now from (6.6.21) it is easy to see that the solution for P is given by

$$P = \left\{ P^{-1}(0) + \int_0^t W_f^T(\sigma) W_f(\sigma) \, d\sigma \right\}^{-1}. \tag{6.6.25}$$

After examining (6.6.25), we can intuitively see that if the infinite integral condition is satisfied, then

$$\lim_{t \to \infty} \lambda_{\max} \{P\} = 0 \tag{6.6.26}$$

and

$$\lim_{t \to \infty} \lambda_{\min} \{P^{-1}\} = \infty. \tag{6.6.27}$$