Chapter 2
Neural Network-Based State Estimation Schemes

2.1 Introduction

In this chapter, two neural network-based adaptive observers for a general model of MIMO nonlinear systems are proposed. The first proposed neural network is linear-in-parameter and the second one is *nonlinear* in its parameters which makes it applicable to many systems with arbitrary degrees of nonlinearity and complexity. The online weight-updating mechanism is a modified version of the backpropagation algorithm with a simple structure together with an e-modification term added to guarantee the robustness of the observer. The stability of the recurrent neural network observers are shown by Lyapunov's direct method. Moreover, the strictly positive real (SPR) assumption imposed on the output error equation is relaxed.

The reminder of this chapter is organized as follows: in Section 2.2, the observer problem is stated and the general structure of the neuro-observer is given. The proposed *linear-in-parameter* neuro-observer is introduced in Section 2.3. In Section 2.4, the results are extended to the case of *nonlinear-in-parameter* neuro-observer. Section 2.5 gives a model of flexible-joint manipulators. The observer performance is evaluated and demonstrated in Section 2.6 by simulation carried out on single- and two-link flexible-joint manipulators. Section 2.7 provides brief conclusions of this chapter.

2.2 Problem Formulation

Consider the general model of a nonlinear MIMO system

$$\begin{aligned} \dot{x}(t) &= f(x,u) \\ y(t) &= Cx(t), \end{aligned} \tag{2.1}$$

H.A. Talebi et al., *Neural Network-Based State Estimation of Nonlinear Systems*, Lecture Notes in Control and Information Sciences 395,
DOI 10.1007/978-1-4419-1438-5_2,

where $u \in R^m$ is the input, $y \in R^m$ is the output, $x \in R^n$ is the state vector of the system, and $f(.,.)$ is a vector-valued unknown nonlinear function.

The objective is to estimate the state x in the presence of unknown function $f(x,u)$ in the system dynamics (2.1). The following assumptions will help us in deriving proper update rules as well as in stability analysis:

Assumption 2.1 *The nonlinear system (2.1) is observable.*

Assumption 2.2 *Another reasonable assumption made here is that the open-loop system is stable. In other words, the states of the system, $x(t)$ are bounded in L_∞ which is a common assumption in identification schemes.*

Now, selecting a Hurwitz matrix, A such that the pair (C,A) is observable and adding Ax to and subtracting it from (2.1) yields

$$\begin{aligned} \dot{x}(t) &= Ax + g(x,u) \\ y(t) &= Cx(t), \end{aligned} \tag{2.2}$$

where $g(x,u) = f(x,u) - Ax$.

The key to designing a neuro-observer is to employ a neural network to identify the nonlinearity and a conventional observer to estimate the states. By invoking a Luenberger observer [21], the observer model of the system (2.2) can be defined as follows

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x} + \hat{g}(\hat{x},u) + G(y - C\hat{x}) \\ \hat{y}(t) &= C\hat{x}(t), \end{aligned} \tag{2.3}$$

where $\hat{x}$ denotes the state of the observer, and the observer gain $G \in R^{n\times m}$ is selected such that $A - GC$ becomes a Hurwitz matrix. It should be noted that the gain G is guaranteed to exist, since A can be selected such that (C,A) is observable.

The structure of a neuro-observer is shown in Fig. (2.1). In this figure, $\hat{x}$ denotes the state of the recurrent model (2.3). Corresponding to the Hurwitz matrix A, $M(s) := (sI - A)^{-1}$ is also shown which is an $n \times n$ matrix whose elements are stable transfer functions. To approximate the nonlinear function $g(x,u)$ a multilayer NN is considered. According to Theorem 1.1, a multilayer NN with sufficiently large number of hidden layer neurons can estimate the unknown function $g(x,u)$ as follows:

$$g(x,u) = W^T \sigma(V^T \bar{x}^T) + \varepsilon(x),$$

where W and V are the weight matrices of the output and hidden layers, respectively, $\bar{x} = [x \quad u]$, $\varepsilon(x)$ is the bounded neural network approximation error, and $\sigma(.)$ is the transfer function of the hidden neurons that is usually considered as a tangent hyperbolic function presenting below:

$$\sigma_i(V_i\bar{x}) = \frac{2}{1 + exp^{-2V_i\bar{x}}} - 1, \tag{2.4}$$

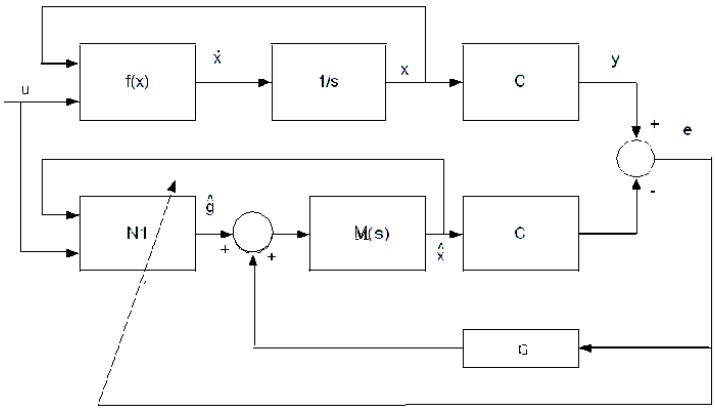

Fig. 2.1 The structure of the proposed neural network observer.

where V_i is the i^{th} row of V, and $\sigma_i(V_i\bar{x})$ is the i^{th} element of $\sigma(V\bar{x})$.

It is assumed that the upper bound on fixed ideal weights W and V exist such that

$$\|W\|_F \leq W_M \tag{2.5}$$

$$\|V\|_F \leq V_M. \tag{2.6}$$

It is also known that the tangent hyperbolic function is bounded by σ_m, i.e.,

$$\|\sigma(V\bar{x})\| \leq \sigma_m. \tag{2.7}$$

In the following two sections, a LPNN and a NLPNN observer are introduced and the stability of the observers are studied by Lyapunov's direct method.

2.3 Linear-in-Parameter Neural Network-Based Observer

In this section a stable LPNN neuro-observer is proposed for nonlinear systems whose dynamics are governed by (2.2). Recalling from Section 1.2.1 that, the linear-in-parameter neural network (LPNN) is obtained by fixing the weights of the first layer as $V = I$. Then, the model can be expressed as

$$g(x,u) = W^T\sigma(\bar{x}) + \varepsilon. \tag{2.8}$$

Thus, the function g can be approximated by an LPNN as

$$\hat{g}(\hat{x},u) = \hat{W}^T\sigma(\hat{\bar{x}}). \tag{2.9}$$

The proposed observer is then given by

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x} + \hat{W}^T\sigma(\hat{\bar{x}}) + G(y - C\hat{x})\\ \hat{y}(t) &= C\hat{x}(t).\end{aligned} \tag{2.10}$$

On the other hand, by defining the state estimation error as $\tilde{x} = x - \hat{x}$ and using (2.2), (2.9) and (2.10), the error dynamics can be expressed as

$$\begin{aligned}\dot{\tilde{x}}(t) &= Ax + W\sigma(\bar{x}) - A\hat{x} - \hat{W}\sigma(\hat{\bar{x}}) - G(Cx - C\hat{x}) + \varepsilon(x)\\ \tilde{y}(t) &= C\tilde{x}(t).\end{aligned} \tag{2.11}$$

Now, adding $W\sigma(\hat{\bar{x}})$ to and subtracting from (2.11) leads to

$$\begin{aligned}\dot{\tilde{x}}(t) &= A_c\tilde{x} + \tilde{W}\sigma(\hat{\bar{x}}) + w(t)\\ \tilde{y}(t) &= C\tilde{x}(t),\end{aligned} \tag{2.12}$$

where $\tilde{W} = W - \hat{W}$, $A_c = A - GC$, and $w(t) = W[\sigma(\bar{x}) - \sigma(\hat{\bar{x}})] + \varepsilon(x)$ is a bounded disturbance term i.e., $\|w(t)\| \leq \bar{w}$ for some positive constant $\bar{w}$, due to the tangent hyperbolic function and the boundedness of the ideal neural network weights, W.

Once the structure of the neural network is known, a proper learning rule should be defined to train the network. This weight-updating mechanism is usually defined in such a way that the stability of the observer is guaranteed. Furthermore, the adaptive law should not be complicated or limited by some strong constraints. Backpropagation (BP) is one of the most popular algorithms that has been widely used for classification, recognition, identification, observation, and control problems. BP owes its popularity to the simplicity in structure which makes it a viable choice for practical problems. However, the main drawback of the previous work (e.g. [55] and [56]) is the lack of a mathematical proof of stability. The following theorem provides a stable neuro-observer by proposing an weight-updating mechanism based on the modified backpropagation algorithm plus an e-modification term to guarantee its robustness.

Theorem 2.1. *Consider the plant model (2.1) and the observer model (2.10). Given Assumptions 2.1 and 2.2, if the weights of the LPNN are updated according to*

$$\dot{\hat{W}} = -\eta(\tilde{y}^T C A_c^{-1})^T(\sigma(\hat{\bar{x}}))^T - \rho\|\tilde{y}\|\hat{W}, \tag{2.13}$$

where $\eta > 0$ is the learning rate, $J = \frac{1}{2}(\tilde{y}^T\tilde{y})$ is the objective function and ρ is a small positive number, then $\tilde{x}, \tilde{W}, \tilde{y} \in L_\infty$

Proof: To prove this theorem, it is firstly shown that the first term in (2.13) is the backpropagation term and the second term is the e-modification terms for incorporating damping in the equations, i.e.,

$$\dot{\hat{W}} = -\eta\left(\frac{\partial J}{\partial \hat{W}}\right) - \rho\|\tilde{y}\|\hat{W}. \tag{2.14}$$

By employing the chain rule $\frac{\partial J}{\partial \hat{W}}$ can be computed as follows

$$\frac{\partial J}{\partial \hat{W}} = \frac{\partial J}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial \hat{W}} = -\tilde{y}^T C \frac{\partial \hat{x}}{\partial \hat{W}}. \tag{2.15}$$

The above equation represent a set of nonlinear dynamical system and the so-called backpropagation in time (dynamic backpropagation) should be utilized for solving the gradient $\frac{\partial \hat{x}}{\partial \hat{W}}$. However, this adds to the complexity of the observer and makes the real-time implementation of the approach very difficult. To cope with this problem, it is suggested to use the static approximation of the gradient, i.e., by setting $\dot{\hat{x}} = 0$ in (2.10), one achieves

$$\frac{\partial \hat{x}}{\partial \hat{W}} \approx -A_c^{-1} \frac{\partial g}{\partial \hat{W}}. \tag{2.16}$$

Now by using (2.9), (2.15) and (2.16), the learning rule (2.14) can be written as

$$\dot{\hat{W}} = -\eta (\tilde{x}^T C^T C A_c^{-1})^T (\sigma(\hat{\tilde{x}}))^T - \rho \|C\tilde{x}\| \hat{W}. \tag{2.17}$$

Therefore, the dynamics of the weight error $\tilde{W} = W - \hat{W}$ can be expressed as

$$\dot{\tilde{W}} = \eta (\tilde{x}^T C^T C A_c^{-1})^T (\sigma(\hat{\tilde{x}}))^T \rho \|C\tilde{x}\| \hat{W}. \tag{2.18}$$

To study the stability of the proposed observer let us consider the Lyapunov function candidate as below

$$L = \frac{1}{2} \tilde{x}^T P \tilde{x} + \frac{1}{2} tr(\tilde{W}^T \rho^{-1} \tilde{W}), \tag{2.19}$$

where $P = P^T$ is a positive-definite matrix satisfying

$$A_c^T P + P A_c = -Q, \tag{2.20}$$

for the Hurwitz matrix A_c and some positive-definite matrix Q. Then, taking time derivative of (2.19) along the trajectories (2.12) yields

$$\dot{L} = \frac{1}{2} \dot{\tilde{x}}^T P \tilde{x} + \frac{1}{2} \tilde{x}^T P \dot{\tilde{x}} + tr(\tilde{W}^T \rho^{-1} \dot{\tilde{W}}). \tag{2.21}$$

Now, by substituting (2.12),(2.20), and (2.18) into (2.21), one can get

$$\begin{aligned} \dot{L} = & -\frac{1}{2} \dot{\tilde{x}}^T Q \tilde{x} + \tilde{x}^T P (\tilde{W} \sigma(\hat{\tilde{x}}) + w) \\ & + tr(-\tilde{W}^T l \tilde{x} \sigma^T + \tilde{W}^T \|C\tilde{x}\| (W - \tilde{W})), \end{aligned} \tag{2.22}$$

where $l = \eta \rho^{-1} A_c^{-T} C^T C$. On the other hand, we have

$$tr(\tilde{W}^T(W-\tilde{W}) \leq W_M\|\tilde{W}\| - \|\tilde{W}\|^2 \tag{2.23}$$

$$tr(\tilde{W}^T l\tilde{x}\sigma^T) \leq \sigma_m\|\tilde{W}^T\|\ \|l\|\ \|\tilde{x}\|, \tag{2.24}$$

where W_M and σ_m are given by (2.5) and (2.7). Note that the last inequality in (2.23) is obtained by using the fact that for two column vectors A and B, the following equality holds:

$$tr(AB^T) = B^T A. \tag{2.25}$$

Now, by using (2.23) and (2.24), one can get

$$\begin{aligned}\dot{L} \leq & -\frac{1}{2}\lambda_{min}(Q)\|\tilde{x}\|^2 + \|\tilde{x}\|\|P\|(\|\tilde{W}\|\sigma_m + \bar{w}) \\ & + \sigma_m\|\tilde{W}\|\|l\|\|\tilde{x}\| + (W_M\|\tilde{W}\| - \|\tilde{W}\|^2)\|C\|\|\tilde{x}\|.\end{aligned}$$

Furthermore, by completing the squares of $\|\tilde{W}\|$ the following condition is obtained to ensure negative definiteness of $\dot{L}$:

$$\|\tilde{x}\| \geq (2\|P\|\bar{w} + (\sigma_m\|p\| + W_M\|C\| + \sigma_m\|l\|)^2/2)/\lambda_{min}(Q).$$

In fact, $\dot{L}$ is negative definite outside the ball with radius b described as $\chi = \{\tilde{x} \mid \|\tilde{x}\| > b\}$, and $\tilde{x}$ is uniformly ultimately bounded. The region inside the ball is attractive, since the increase of $\dot{L}$ for smaller values of $\|\tilde{x}\|$ will increase L and $\tilde{x}$, which brings the $\tilde{x}$ outside the ball χ where $\dot{L}$ is negative semi-definite and results in reducing L and $\tilde{x}$. The above analysis shows the ultimate boundedness of $\tilde{x}$. Further details concerning the notion of ultimate boundedness is given in Appendix A.

To show the boundedness of the weight error $\tilde{W}$, consider (2.18) which can be rewritten as

$$\dot{\tilde{W}} = f_1(\tilde{x}) + \rho\|C\tilde{x}\|\hat{W} = f_1(\tilde{x}) + \alpha_1 W - \alpha_1\tilde{W}, \tag{2.26}$$

where

$$\begin{aligned}f_1(\tilde{x}) &= \eta(\tilde{x}^T C^T C A_c^{-1})^T(\sigma(\hat{\tilde{x}}))^T \\ \alpha_1 &= \rho\|C\tilde{x}\|.\end{aligned}$$

It can be seen that $f_1(.)$ is bounded since $\tilde{x}$ and $\sigma(\hat{V}\hat{\tilde{x}})$ are both bounded, C is bounded, and A_c is a Hurwitz matrix . Given the fact the the ideal weight W is fixed, (2.26) can be regarded as a linear system with bounded input $(f_1(\tilde{x}) + \alpha_1 W)$. It is clear that this system is stable since α_1 is positive and the system input remains bounded. Hence, the boundedness of $\tilde{W}$ is also ensured. This completes the proof of the theorem. □

A stable LPNN neuro-observer was introduced in this section. The next section provides a neuro-observer for larger class of nonlinear dynamics, called nonlinear-in-parameter (NLPNN) neuro-observer.

2.4 Nonlinear-in-Parameter Neural Network-Based Observer

In more general neuro-observer structure, where the weight matrix of the first layer is not restricted to be constant, the nonlinear function g can be approximated by a multilayer NN according to the following model:

$$\hat{g}(\hat{x},u) = \hat{W}\sigma(\hat{V}\hat{\bar{x}}). \tag{2.27}$$

Therefore, the observer dynamics can be given as

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x} + \hat{W}\sigma(\hat{V}\hat{\bar{x}}) + G(y - C\hat{x}) \\ \hat{y}(t) &= C\hat{x}(t), \end{aligned} \tag{2.28}$$

and also the error dynamics can be expressed as

$$\begin{aligned} \dot{\tilde{x}}(t) &= A_c\tilde{x} + \tilde{W}\sigma(\hat{V}\hat{\bar{x}}) + w(t) \\ \tilde{y}(t) &= C\tilde{x}(t), \end{aligned} \tag{2.29}$$

where $\tilde{W} = W - \hat{W}$, $A_c = A - GC$, $w(t) = W[\sigma(V\bar{x}) - \sigma(\hat{V}\hat{\bar{x}})] + \varepsilon(x)$ is a bounded disturbance term i.e., $\|w(t)\| \leq \bar{w}$ for some positive constant $\bar{w}$, due to the tangent hyperbolic function and the boundedness of the ideal neural network weights V and W.

In Theorem 2.2, a learning rule is introduced for neuro-observer (2.28) which guarantees the stability of the observer and the boundedness of the error estimation.

Theorem 2.2. *Consider the plant model (2.1) and the observer model (2.28). Given Assumptions 2.1 and 2.2, if the weights of the NLPNN are updated according to*

$$\dot{\hat{W}} = -\eta_1(\tilde{y}^T CA_c^{-1})^T(\sigma(\hat{V}\hat{\bar{x}}))^T - \rho_1\|\tilde{y}\|\hat{W} \tag{2.30}$$

$$\dot{\hat{V}} = -\eta_2(\tilde{y}^T CA_c^{-1}\hat{W}(I - \Lambda(\hat{V}\hat{\bar{x}})))^T sgn(\hat{\bar{x}})^T - \rho_2\|\tilde{y}\|\hat{V}, \tag{2.31}$$

where $\Lambda(\hat{V}\hat{\bar{x}}) = diag\{\sigma_i^2(\hat{V}_i\hat{\bar{x}})\}, i = 1,2,...,m$ and $sgn(\hat{\bar{x}})$ is the sign function:

$$sgn(\hat{\bar{x}}) = \begin{cases} 1 & for\ \hat{\bar{x}} > 0 \\ 0 & for\ \hat{\bar{x}} = 0 \\ -1 & for\ \hat{\bar{x}} < 0 \end{cases},$$

then $\tilde{x}, \tilde{W}, \tilde{V}, \tilde{y} \in L_\infty$, i.e., the estimation error, weights error, and the output error are all bounded. In these equations, $\eta_1, \eta_2 > 0$ are the learning rates, $J = \frac{1}{2}(\tilde{y}^T\tilde{y})$ is the objective function and ρ_1, ρ_2 are positive numbers.

Proof: Similar to the proof of Theorem 2.1, at first it is shown that the first terms in (2.30) and (2.31) are the backpropagation terms and the second terms are the e-modification which incorporates proper damping in the equations. Then, the stability of the proposed observer is studied by Lyapunov's direct method. In other words, the weights are updated based on the following dynamics

$$\dot{\hat{W}} = -\eta_1(\frac{\partial J}{\partial \hat{W}}) - \rho_1\|\tilde{y}\|\hat{W} \tag{2.32}$$

$$\dot{\hat{V}} = -\eta_2(\frac{\partial J}{\partial \hat{V}}) - \rho_2\|\tilde{y}\|\hat{V}. \tag{2.33}$$

Let us define

$$net_{\hat{v}} = \hat{V}\hat{\bar{x}} \tag{2.34}$$

$$net_{\hat{w}} = \hat{W}\sigma(\hat{V}\hat{\bar{x}}). \tag{2.35}$$

Therefore, $\frac{\partial J}{\partial \hat{W}}$ and $\frac{\partial J}{\partial \hat{V}}$ can be computed as [42]

$$\frac{\partial J}{\partial \hat{W}} = \frac{\partial J}{\partial net_{\hat{w}}}.\frac{\partial net_{\hat{w}}}{\partial \hat{W}}$$
$$\frac{\partial J}{\partial \hat{V}} = \frac{\partial J}{\partial net_{\hat{v}}}.\frac{\partial net_{\hat{v}}}{\partial \hat{V}}.$$

On the other hand, we have

$$\frac{\partial J}{\partial net_{\hat{w}}} = \frac{\partial J}{\partial \tilde{y}}.\frac{\partial \tilde{y}}{\partial \hat{x}}.\frac{\partial \hat{x}}{\partial net_{\hat{w}}} = -\tilde{y}^T C.\frac{\partial \hat{x}}{\partial net_{\hat{w}}} = -\tilde{x}^T C^T C.\frac{\partial \hat{x}}{\partial net_{\hat{w}}}$$
$$\frac{\partial J}{\partial net_{\hat{v}}} = \frac{\partial J}{\partial \tilde{y}}.\frac{\partial \tilde{y}}{\partial \hat{x}}.\frac{\partial \hat{x}}{\partial net_{\hat{v}}} = -\tilde{y}^T C.\frac{\partial \hat{x}}{\partial net_{\hat{v}}} = -\tilde{x}^T C^T C.\frac{\partial \hat{x}}{\partial net_{\hat{v}}}, \tag{2.36}$$

and

$$\frac{\partial net_{\hat{w}}}{\partial \hat{W}} = \sigma(\hat{V}\hat{\bar{x}})$$
$$\frac{\partial net_{\hat{v}}}{\partial \hat{V}} = \hat{\bar{x}}. \tag{2.37}$$

Now, by employing (2.28) and the definitions of $net_{\hat{v}}$ and $net_{\hat{w}}$ as defined in (2.34) and (2.35), respectively one can obtain

$$\frac{\partial \dot{\hat{x}}(t)}{\partial net_{\hat{w}}} = A_c\frac{\partial \hat{x}}{\partial net_{\hat{w}}} + I$$
$$\frac{\partial \dot{\hat{x}}(t)}{\partial net_{\hat{v}}} = A_c\frac{\partial \hat{x}}{\partial net_{\hat{v}}} + \hat{W}(I - \Lambda(\hat{V}\hat{\bar{x}})). \tag{2.38}$$

Similar to the approach adopted in the proof of Theorem 2.1, using the static approximation, yields in

$$\frac{\partial \hat{x}}{\partial net_{\hat{w}}} \approx -A_c^{-1}$$
$$\frac{\partial \hat{x}}{\partial net_{\hat{v}}} \approx -A_c^{-1}\hat{W}(I - \Lambda(\hat{V}\hat{\bar{x}})). \tag{2.39}$$

Then, substituting (2.36), (2.37), and (2.39), in the the learning rules (2.32) and (2.33) leads to

$$\dot{\hat{W}} = -\eta_1(\tilde{x}^T C^T C A_c^{-1})^T(\sigma(\hat{V}\hat{\bar{x}}))^T - \rho_1\|C\tilde{x}\|\hat{W} \tag{2.40}$$

$$\dot{\hat{V}} = -\eta_2(\tilde{x}^T C^T C A_c^{-1}\hat{W}(I-\Lambda(\hat{V}\hat{\bar{x}})))^T\hat{\bar{x}}^T - \rho_2\|C\tilde{x}\|\hat{V}. \tag{2.41}$$

Therefore, the learning rules (2.40) and (2.41) in terms of the weight errors $\tilde{W} = W - \hat{W}$ and $\tilde{V} = V - \hat{V}$, can be written as

$$\dot{\tilde{W}} = \eta_1(\tilde{x}^T C^T C A_c^{-1})^T(\sigma(\hat{V}\hat{\bar{x}}))^T + \rho_1\|C\tilde{x}\|\hat{W} \tag{2.42}$$

$$\dot{\tilde{V}} = \eta_2(\tilde{x}^T C^T C A_c^{-1}\hat{W}(I-\Lambda(\hat{V}\hat{\bar{x}})))^T\hat{\bar{x}}^T + \rho_2\|C\tilde{x}\|\hat{V}. \tag{2.43}$$

In order to simplify the stability analysis, we replace $\hat{\bar{x}}$ by $sgn(\hat{\bar{x}})$ in the above equation:

$$\dot{\tilde{V}} = \eta_2(\tilde{x}^T C^T C A_c^{-1}\hat{W}(I-\Lambda(\hat{V}\hat{\bar{x}})))^T sgn(\hat{\bar{x}})^T + \rho_2\|C\tilde{x}\|\hat{V}. \tag{2.44}$$

As will be clear later, this modification is necessary to derive Equation (2.50), since $sgn(\hat{\bar{x}})$ is bounded but this is not necessarily true for $\hat{\bar{x}}$. Note that by using the sign of $\hat{\bar{x}}$, the weight update is guaranteed to move in the right direction. It can be seen that the learning rule (2.42) and (2.44) are equivalent to (2.30) and (2.31) expressed in terms of $\tilde{W}$ and $\tilde{V}$.

Now, to show ultimate boundedness of the estimation errors, let us consider the following positive definite Lyapunov function candidate

$$L = \frac{1}{2}\tilde{x}^T P\tilde{x} + \frac{1}{2}tr(\tilde{W}^T\tilde{W}) + \frac{1}{2}tr(\tilde{V}^T\tilde{V}), \tag{2.45}$$

where $P = P^T$ is a positive-definite matrix satisfying

$$A_c^T P + PA_c = -Q, \tag{2.46}$$

for the Hurwitz matrix A_c and some positive-definite matrix Q. The time derivative of (2.45) is given by

$$\dot{L} = \frac{1}{2}\dot{\tilde{x}}^T P\tilde{x} + \frac{1}{2}\tilde{x}^T P\dot{\tilde{x}} + tr(\tilde{W}^T\dot{\tilde{W}}) + tr(\tilde{V}^T\dot{\tilde{V}}). \tag{2.47}$$

Then, substituting (2.29), (2.42), (2.44) and (2.46) into (2.47) yields

$$\begin{aligned}\dot{L} = &-\frac{1}{2}\tilde{x}^T Q\tilde{x} + \tilde{x}^T P(\tilde{W}\sigma(\hat{V}\hat{\bar{x}}) + w) + tr(\tilde{W}^T l_1\tilde{x}\sigma(\hat{V}\hat{\bar{x}})^T + \tilde{W}^T\rho_1\|C\tilde{x}\|(W-\tilde{W}))\\ &+ tr(\tilde{V}^T(I-\Lambda(\hat{V}\hat{\bar{x}}))^T\hat{W}^T l_2\tilde{x}sgn(\hat{\bar{x}})^T + \tilde{V}^T\rho_2\|C\tilde{x}\|(V-\tilde{V})),\end{aligned} \tag{2.48}$$

where $l_1 = \eta_1 A_c^{-T}C^T C, l_2 = \eta_2 A_c^{-T}C^T C$. On the other hand, the following inequalities are always true

$$\begin{aligned} tr(\tilde{W}^T(W-\tilde{W})) &\leq W_M\|\tilde{W}\| - \|\tilde{W}\|^2 \\ tr(\tilde{V}^T(V-\tilde{V})) &\leq V_M\|\tilde{V}\| - \|\tilde{V}\|^2 \\ tr(\tilde{W}^T l_1 \tilde{x}\sigma(\hat{V}\hat{x})^T) &\leq \sigma_m\|\tilde{W}^T\|\,\|l_1\|\,\|\tilde{x}\|, \end{aligned} \tag{2.49}$$

where W_M, V_M and σ_m are given by (2.5)-(2.7). Now, using the facts that $\|\hat{W}\| = \|W-\tilde{W}\| \leq W_M + \|\tilde{W}\|$, $1-\sigma_m^2 \leq 1$, and (2.25) leads to the following inequality

$$tr(\tilde{V}^T(I-\Lambda(\hat{V}\hat{\bar{x}}))^T\hat{W}^T l_2\tilde{x}sgn(\hat{\bar{x}})^T) \leq \|\tilde{V}\|(W_M+\|\tilde{W}\|)\|l_2\|\|\tilde{x}\|. \tag{2.50}$$

Then, by employing (2.49) and (2.50), one can get

$$\begin{aligned} \dot{L} \leq & -\frac{1}{2}\lambda_{min}(Q)\|\tilde{x}\|^2 + \|\tilde{x}\|\|P\|(\|\tilde{W}\|\sigma_m + \bar{w}) + \sigma_m\|\tilde{W}\|\|l_1\|\|\tilde{x}\| \\ & + (W_M\|\tilde{W}\| - \|\tilde{W}\|^2)\rho_1\|C\|\|\tilde{x}\| + \|\tilde{V}\|\|l_2\|(W_M+\|\tilde{W}\|)\|\tilde{x}\| \\ & + \rho_2\|C\|\|\tilde{x}\|(V_M\|\tilde{V}\| - \|\tilde{V}\|^2) = F. \end{aligned} \tag{2.51}$$

By completing the squares for the terms involving $\|\tilde{W}\|$ and $\|\tilde{V}\|$, we look for conditions on $\|\tilde{x}\|$ which are independent of the neural network weights error and also make the time derivative of the Lyapunov function candidate, (2.45) negative. Toward this end, by defining $K_1 = \frac{\|l_2\|}{2}$ and adding $K_1^2\|\tilde{W}\|^2\|\tilde{x}\|$ and $\|\tilde{V}\|^2\|\tilde{x}\|$ to and subtracting them from the right hand side of (2.51), one can obtain

$$\begin{aligned} F = & -\frac{1}{2}\lambda_{min}(Q)\|\tilde{x}\|^2 + (\|P\|\bar{w} - (\rho_1\|C\| - K_1^2)\|\tilde{W}\|^2 - (K_1\|\tilde{W}\| - \|\tilde{V}\|)^2 \\ & + (\|P\|\sigma_m + \sigma_m\,\|l_1\| + \rho_1\|C\|W_M)\|\tilde{W}\| + (\rho_2\|C\|V_M + \|l_2\|W_M)\|\tilde{V}\| \\ & - (\rho_2\|C\| - 1)\|\tilde{V}\|^2)\|\tilde{x}\|. \end{aligned} \tag{2.52}$$

Next, let us introduce K_2 and K_3 as follows:

$$\begin{aligned} K_2 &= \frac{\rho_1 W_M\|C\| + \sigma_m\|l_1\| + \|P\|\sigma_m}{2(\rho_1\|C\| - K_1^2)} \\ K_3 &= \frac{\rho_2\|C\|V_M + \|l_2\|W_M}{2(\rho_2\|C\| - 1)}. \end{aligned}$$

Then, $K_2^2\|\tilde{x}\|$ and $K_3^2\|\tilde{x}\|$ are added to and subtracted from (2.52):

$$\begin{aligned} F = & -\frac{1}{2}\lambda_{min}(Q)\|\tilde{x}\|^2 + (\|P\|\bar{w} + (\rho_1\|C\| - K_1^2)K_2^2 + (\rho_2\|C\| - 1)K_3^2 \\ & - (\rho_1\|C\| - K_1^2)(K_2 - \|\tilde{W}\|)^2 - (\rho_2\|C\| - 1)(K_3 - \|\tilde{V}\|)^2 \\ & - (K_1\|\tilde{W}\| - \|\tilde{V}\|)^2)\|\tilde{x}\|. \end{aligned}$$

Now, assuming the $\rho_1 \geq \frac{K_1^2}{\|C\|}$, $\rho_2 \geq \frac{1}{\|C\|}$ and using the fact that the last three terms of (2.53) are negative yields

$$F \leq -\frac{1}{2}\lambda_{min}(Q)\|\tilde{x}\|^2 + \|\tilde{x}\|(\|P\|\bar{w} + (\rho_1\|C\| - K_1^2)K_2^2 + (\rho_2\|C\| - 1)K_3^2). \tag{2.53}$$

Therefore, the following condition on $\|\tilde{x}\|$ guarantees the negative semi-definiteness of $\dot{L}$:

$$\|\tilde{x}\| > \frac{2(\|P\|\bar{w} + (\rho_1\|C\| - K_1^2)K_2^2 + (\rho_2\|C\| - 1)K_3^2)}{\lambda_{min}(Q)} = b. \tag{2.54}$$

In fact, $\dot{L}$ is negative definite outside the ball with radius b described as $\chi = \{\tilde{x} \mid \|\tilde{x}\| > b\}$, and $\tilde{x}$ is uniformly ultimately bounded.

To study the boundedness of the weight errors $\tilde{W}$ and $\tilde{V}$, let us rewrite (2.42) and (2.44) as

$$\dot{\tilde{W}} = f_1(\tilde{x}, \hat{V}) + \rho_1\|C\tilde{x}\|\hat{W} = f_1(\tilde{x}, \hat{V}) + \alpha_1 W - \alpha_1\tilde{W} \tag{2.55}$$

$$\dot{\tilde{V}} = f_2(\tilde{x}, \hat{W}, \hat{V}) + \rho_2\|C\tilde{x}\|\hat{V} = f_2(\tilde{x}, \hat{W}, \hat{V}) + \alpha_2 V - \alpha_2\tilde{V}, \tag{2.56}$$

where

$$\begin{aligned} f_1(\tilde{x}, \hat{V}) &= \eta_1(\tilde{x}^T C^T C A_c^{-1})^T(\sigma(\hat{V}\hat{\tilde{x}}))^T \\ f_2(\tilde{x}, \hat{W}, \hat{V}) &= \eta_2(\tilde{x}^T C^T C A_c^{-1}\hat{W}(I - \Lambda(\hat{V}\hat{\tilde{x}})))^T sgn(\hat{\tilde{x}})^T \\ \alpha_1 &= \rho_1\|C\tilde{x}\| \\ \alpha_2 &= \rho_2\|C\tilde{x}\|. \end{aligned}$$

Since $\tilde{x}$ and $\sigma(\hat{V}\hat{\tilde{x}})$ are both bounded, C is bounded, and A_c is a Hurwitz matrix, it can be concluded that $f_1(.)$ is bounded. Given that the ideal weight W is fixed and α_1 is positive, (2.55) can be regarded as a linear system with bounded input $(f_1(\tilde{x}, \hat{V}) + \alpha_1 W)$ and therefore, it can be assured that $\tilde{W}$ is bounded. Given that $\tilde{W} \in L_\infty$, it can be observed that $f_2(.)$ is also bounded since all its arguments are bounded including $\Lambda(.)$ as defined below equation (2.31). Consequently, similar analysis shows that (2.56) also represents a stable bounded input linear system and hence $\tilde{V} \in L_\infty$. The key to the above analysis is that $\tilde{V}$ only appears in $f_1(.)$ and $f_2(.)$ as bounded functions ($\sigma(.)$ and $\Lambda(.)$). This completes the proof. □

It is worth mentioning that the size of the estimation error bound b can be kept small by proper selection of the damping factors, A_c and the learning rates (through K_2 and K_3) such that a higher accuracy can be achieved. It should be noted that since ρ_1 and ρ_2 are design parameters, the conditions on them do not restrict the applicability of the proposed approach.

Remark 2.1 *In many cases, not all system states directly appear in the output of the system. Hence, some elements of C would be zero and this will slow down the learning process because of the structure of the backpropagation algorithm (see equations (2.42) and (2.44)). It is suggested that for the purpose of training only, the output matrix C is modified to C_1 such that all the states appear in the output of the system and directly contribute to the observation error, i.e., (2.30) and (2.31)*

can be redefined as

$$\dot{\hat{W}} = -\eta_1(\tilde{y}^T C_1 A_c^{-1})^T(\sigma(\hat{V}\hat{\bar{x}}))^T - \rho_1\|\tilde{y}\|\hat{W} \tag{2.57}$$

$$\dot{\hat{V}} = -\eta_2(\tilde{y}^T C_1 A_c^{-1}\hat{W}(I-\Lambda(\hat{V}\hat{\bar{x}})))^T sgn(\hat{\bar{x}})^T - \rho_2\|\tilde{y}\|\hat{V}. \tag{2.58}$$

However, the rest of the proof and conditions remain unchanged.

Remark 2.2 *There are different parameters that can be adjusted to control the convergence of the neural network e.g. the learning rates η_1 and η_2, the damping factors ρ_1 and ρ_2, and the Hurwitz matrix A. It is well known that larger learning rates can lead to faster convergence but extra care should be taken to avoid overshoot. Moreover, although increasing the damping factor can improve the stability of the system, too much damping can lead to premature convergence of the weights which might be far from the ideal weights. The Hurwitz matrix A which is primarily used for stable integration has considerable effect on convergence as well as accuracy of the state estimation. One has to be concerned with different issues when selecting this matrix. First, it should be selected such that the pair (C,A) is observable. Since, the states of the estimator are obtained through $M(s)$ whose poles are the eigenvalues of A, a more stable matrix (matrix with eigenvalues farther to the left in the complex plane) helps the states of the estimator track those of the actual system with better accuracy (in terms of delays). However, a more stable A and hence A_c might slow down the convergence of the weights since A_c^{-1} is used in updating the weights (see equations (2.42) and (2.44)). One solution is to use a more stable A for better accuracy and use higher learning rates for better convergence.*

2.5 A Case Study: Application to State Estimation of Flexible-Joint Manipulators

The state estimation of flexible-joint manipulators is considered as a case study to evaluate the performance of the proposed observer. Robot manipulators with joint or link flexibility are proper examples of systems with high nonlinearity, unmodeled dynamics, and parameter variations.

A dominant source of compliance in robotic systems is the result of flexibility in the motor transmissions. . For instance, the unconventional gear-tooth meshing action of the harmonic drive makes it possible to acquire higher gear ratio and high torque capability in a compact geometry. On the other hand, a harmonic drive transmission is much more flexible than a conventional gear transmission. In Appendix B, the harmonic drives are explained in more details. The flexibility of the joint causes difficulty in modeling manipulator dynamics and becomes a potential source of uncertainty that can degrade the performance of a manipulator and in some cases can even destabilize the system [38]. Consequently, addressing this issue is important for calibration as well as modeling and control of robot manipulators. Joint

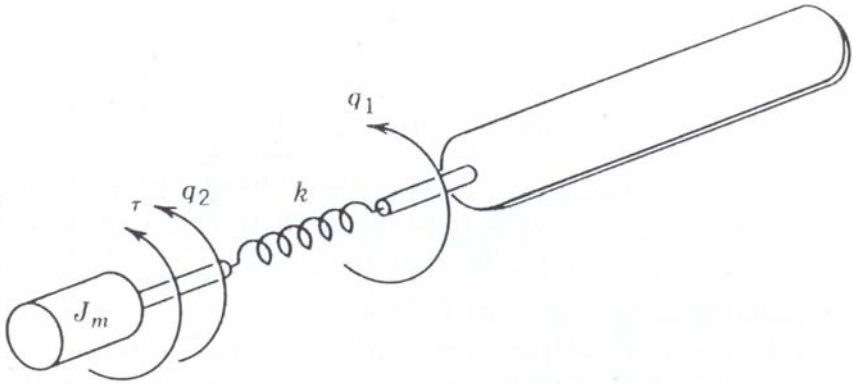

Fig. 2.2 The schematic of flexible-joint manipulator modeled by torsional spring [3]

elasticity can be modeled as a torsional spring between the input shaft (motor) and the output shaft (link) of the manipulator, as shown in Fig. 2.2. Due to the presence of joint flexibility, there are twice as many degrees of freedom compared to the rigid joint case.

To compensate for joint flexibility, many sophisticated control algorithms have been proposed both in constrained [57, 58] and unconstrained motions [59, 60, 61, 62, 63, 64, 65]. In [65], an adaptive control scheme is addressed for flexible-joint robots. Most of these schemes however, assume the availability of both the link and the motor positions, a condition that may not always be satisfied. Luenberger observers, reduced-order high-gain observers, and Kalman filter based observers have been used to relax the requirement of measurement from both sides of the transmission device [66, 67, 68, 28, 69, 70]. However, a fundamental assumption underlying all of these methods is that the system nonlinearities are completely known *a priori*. In this section, our proposed neural network observer is applied to a flexible-joint robot when the motor positions and velocities are available and the link positions and velocities need to be estimated. This choice of measured variables is the most practical.

Although, no *a priori* knowledge about the system dynamics is required for our state estimation approach, the analytical model of the manipulator is still needed for simulation purposes. In the following section, a dynamic model of a flexible-joint manipulator is introduced.

2.5.1 Manipulator Model

Using the Lagrangian approach, the flexible-joint manipulator can be modeled by the following equations [3].

$$D_l(q_1)\ddot{q}_1 + C_1(q_1,\dot{q}_1) + g(q_1) + B_1\dot{q}_1 = \tau_s$$
$$J\ddot{q}_2 + \tau_s + B_2\dot{q}_2 = \tau, \tag{2.59}$$

where $q_1 \in \Re^n$ is the vector of link positions, $q_2 \in \Re^n$ is the vector of motor shaft positions, $g(q_1) \in \Re^n$ is the gravity loading force, $C_1(q_1,\dot{q}_1) \in \Re^n$ is the term corresponding to the centrifugal and Coriolis forces, $B_1 \in \Re^{n\times n}$ and $B_2 \in \Re^{n\times n}$ are the viscous damping matrices at the output and input shafts respectively, $D_l(q_1) \in \Re^{n\times n}$ and $J \in \Re^{n\times n}$ are the robot and the actuator inertia matrices respectively, and τ is the input torque. The reaction torque τ_s from the rotational spring is often considered as

$$\tau_s = K(q_2 - q_1) + \beta(q_1,\dot{q}_1,q_2,\dot{q}_2),$$

where $K \in \Re^{n\times n}$ is the positive-definite stiffness matrix of the rotational spring that represents the flexibility present between the input and the output shafts. In general, there is an unknown nonlinear force $\beta(q_1,\dot{q}_1,q_2,\dot{q}_2)$ which can be regarded as a combination of a nonlinear spring and friction at the output shafts of the manipulator. The reaction torque τ_s cannot be modeled accurately and is assumed to be unknown for observer design and is included for simulation purposes only. For more details regarding flexible-joint manipulator dynamics refer to Appendix B.

It should be noted that the flexible-joint manipulator system is marginally stable, i.e., its states are bounded for all bounded inputs except the step input. All results except those obtained for generalization are obtained for sinusoidal inputs. However, it is well known that the flexible-joint system can be stabilized using PD control of joint position feedback with arbitrary positive gains [3, 71]. Hence, a joint PD controller can be used to make the system stable without any *a priori* knowledge about the system dynamics. The joint position measurements are readily available. Hence, for a practical application this remedy can be used as in [72]. This technique is utilized for simulations concerning the generalization issue.

2.6 Simulation Results

In this section, the performance of the proposed observer is investigated on single-link and two-link flexible-joint manipulators.

2.6.1 A Single-link Flexible-Joint Manipulator

Consider a single-link flexible-joint manipulator, (2.59) whose its state vector is defined as $x = [q_1\ \dot{q}_1\ q_2\ \dot{q}_2]$ and numerical value of its parameters are given below:

$$J = 1.16kg.m^2,\ \ m = 1Kg,\ \ l = 1m,\ \ K = 100N/m$$
$$\eta = 10,\ \rho = 1.5,$$

$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$, and A is an 4×4 block diagonal matrix whose blocks A_{ii} are selected as $\begin{bmatrix} -20 & 1 \\ 0 & -20 \end{bmatrix}$, $i = 1...2$, where J is the motor inertia, m is the link mass, l is the link length, and K is the stiffness of the joint.

To estimate the states of this system a three-layer neural network is applied which has 5 neurons in the input layer, 5 neurons in the hidden layer, and 4 neurons in the output layer. u, $\hat{q}_1$, $\dot{\hat{q}}_1$, $\hat{q}_2$ and $\dot{\hat{q}}_2$ are considered as input of the neural network. The input layer neurons have tangent hyperbolic transfer functions and the output neurons use linear activation functions. The initial weights of the network are selected as small random numbers.

Fig. 2.3 depicts the state estimation obtained by our proposed neural network. Fig. 2.3–a illustrate the responses of q_2 and $\hat{q}_2$ and Fig. 2.3–b shows those of q_1 and $\hat{q}_1$. Fig. 2.3–c and 2.3–d depict the responses of the motor and link velocities, respectively. It is clear that the states of the observer follow the states of the actual system.

2.6.1.1 Off-line Training:

In the next step, to evaluate the performance of the neural network when it is working as an off-line training scheme the training is stopped and the weights obtained from the last simulation results are considered as initial weights. Therefore, the network is used in recall mode. Fig. 2.4 illustrates the estimated stats when the neural network is trained off-line, i.e., the weights are not updated. These results demonstrate that the neural network has learned the system dynamics very accurately and can estimate the stats after training, properly.

2.6.1.2 Generalization

To investigate the generalization issue, another simulation is performed on the system. Toward this end, the training trajectory are defined as $0.1sint + 0.2sin2t + 0.05sin4t$. The results are given in Fig. 2.5–a to 2.5–e. Fig. 2.5–a depicts the state response of q_2 and $\hat{q}_2$ for the closed-loop system and Fig. 2.5–b illustrates those for q_1 and $\hat{q}_1$. Figs. 2.5–c to 2.5–e depict the state responses of the system in a short interval after the learning.

Later, we stop the training and applied a totally different trajectory namely $0.075sin3t$ to the manipulator. The simulation results obtained in this case are shown in Figs. 2.6–a to 2.6–d. Fig. 2.6–a depicts the state response of q_2 and $\hat{q}_2$ for the closed-loop system and Fig. 2.6–b shows those for q_1 and $\hat{q}_1$. Figs. 2.6–c and 2.6–d illustrate the responses for the joint and link velocities and their estimations. As can be observed, the proposed neural observer exhibits the desired generalization property.

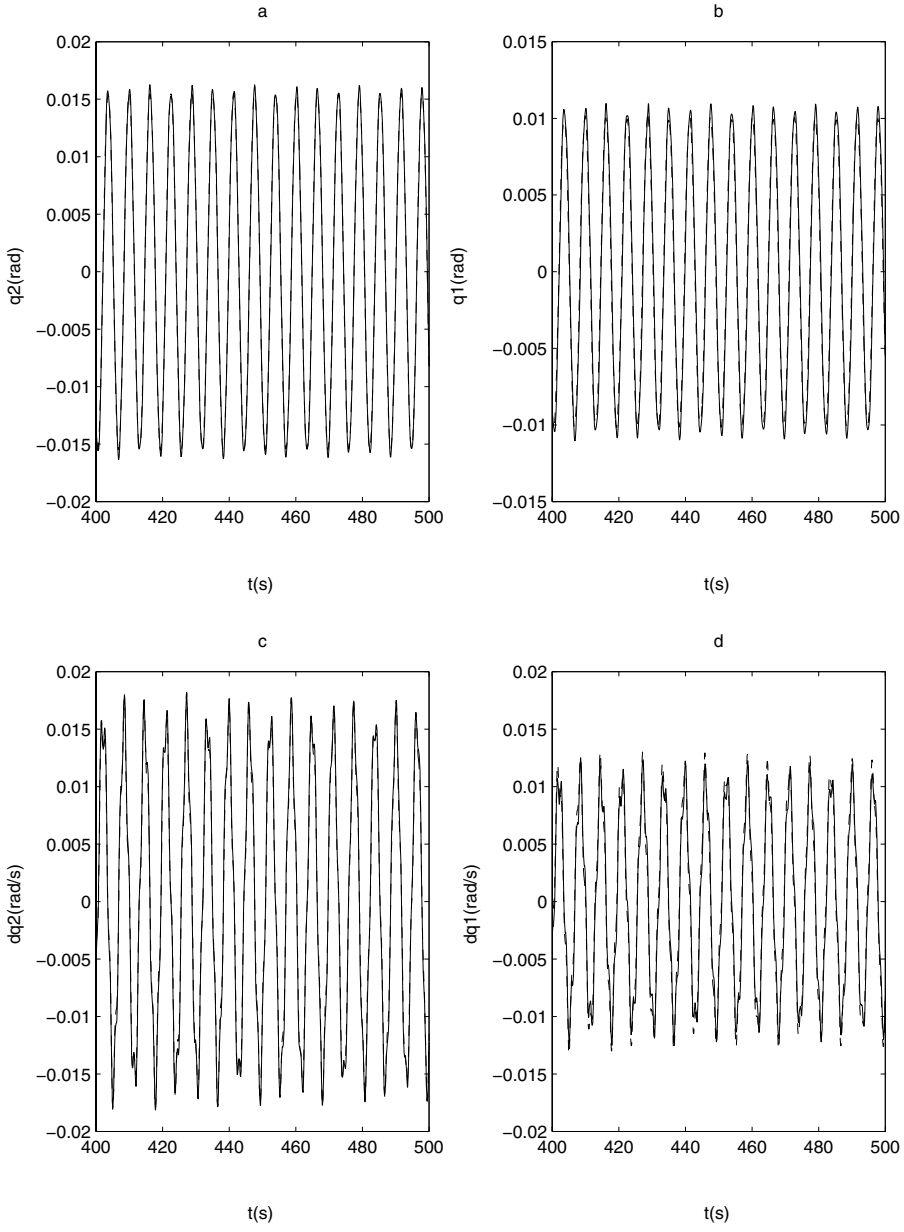

Fig. 2.3 The state responses of the single-link flexible-joint manipulator to $\sin(t)$ reference trajectory for NLPNN: (a) motor position, (b) link position, (c) motor velocity, (d) link velocity. The solid lines correspond to the actual states and the dashed lines correspond to the states of the observer.

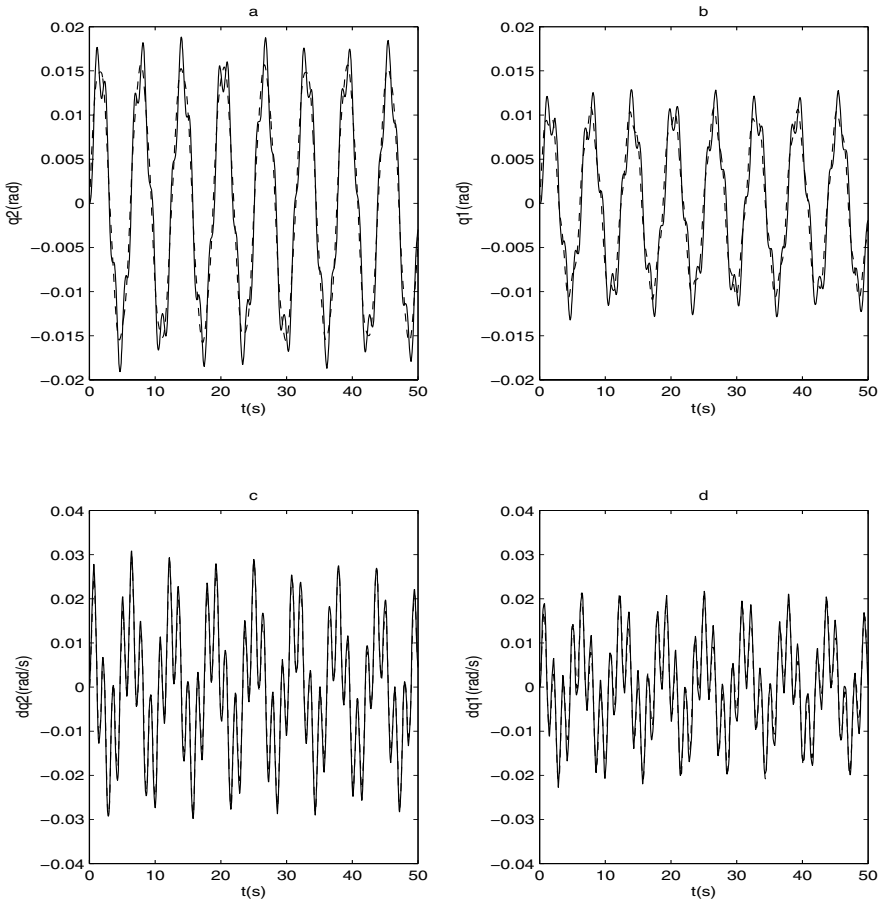

Fig. 2.4 The state responses of the single-link flexible-joint manipulator to $\sin(t)$ reference trajectory for NLPNN after the learning stops: (a) motor position, (b) link position, (c) motor velocity, (d) link velocity. The solid lines correspond to the actual states and the dashed lines correspond to the states of the observer.

2.6.2 A Two-link Flexible-Joint Manipulator

Simulation results for a two–link planar manipulator are presented in this section. The dynamics of a two-link manipulator are far more complicated than those of a single-link manipulator. The manipulator consists of two flexible-joints with state vector $x = [x_1\ x_2]$ where $x_i = [q_{1i}\ \dot{q}_{1i}\ q_{2i}\ \dot{q}_{2i}]$ is the state vector of i^{th} link for $i = 1,2$ and the following numerical data

$$J = diag\{1.16, 1.16\},\ m = diag\{1,1\},\ l_1 = l_2 = 1m,$$
$$K = diag\{100, 100\},\ \eta_1 = \eta_2 = 100,\ \text{and}\ \rho_1 = \rho_2 = 1.5,$$

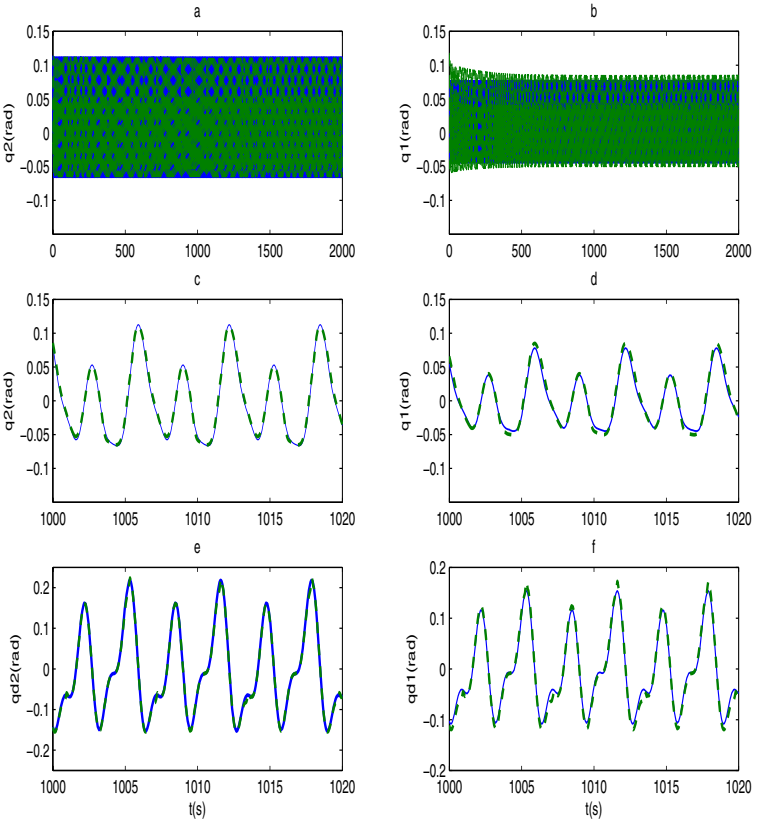

Fig. 2.5 The state responses of the single-link flexible-joint manipulator to $0.1sint + 0.2sin2t + 0.05sin4t$ reference trajectory for NLPNN: (a) motor position during the learning, (b) link position during the learning, (c) motor position at the end of learning phase, (d) link position at the end of learning phase, (e) motor velocity, (f) link velocity. The solid lines correspond to the actual states and the dashed lines correspond to the states of the observer.

and A is an 8×8 block diagonal matrix whose blocks A_{ii} are selected as $\begin{bmatrix} -20 & 1 \\ 0 & -20 \end{bmatrix}$, $i = 1...4$. The neural network has three layers including 10 neurons in input layer,10 neurons with tangent hyperbolic activation functions in hidden layer, and the 8 neurons with linear transfer functions in output layer. The input of the network is $\hat{\bar{x}}$. Fig. 2.7 depicts the result of the state estimation. As can be observed, despite the increased complexity in the manipulator model, the neural network has learned the manipulator dynamics, and all the states of the neural network track the corresponding states of the system. In the last step, for the sake of comparison a simulation

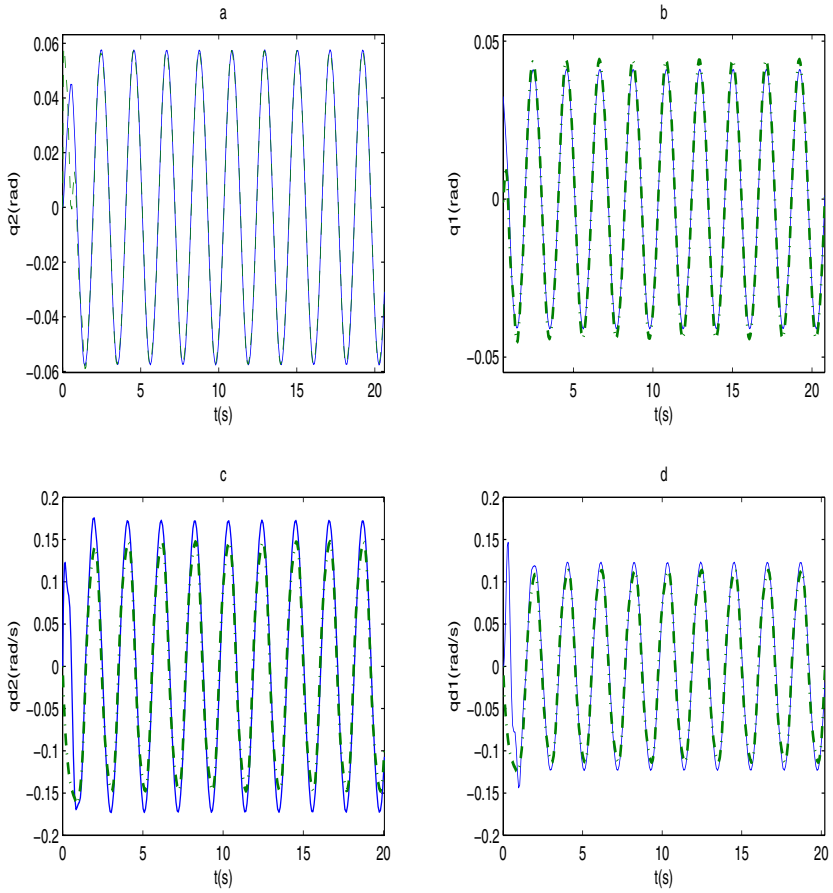

Fig. 2.6 The state responses of the single-link flexible-joint manipulator to $0.075sin3t$ reference trajectory for NLPNN during the recall (testing) phase: (a) motor position, (b) link position, (c) motor velocity, (d) link velocity. The solid lines correspond to the actual states and the dashed lines correspond to the states of the observer.

is performed using LPNN and NLPNN in the same situation. Fig. 2.8–a illustrates the state responses, q_2 and $\hat{q}_2$ and Fig. 2.8–c shows the state responses, q_1 and $\hat{q}_1$ for the LPNN observer. The simulation results of the NLPNN observer in the same situation are shown in Figs. 2.8–b and 2.8–d for q_2 and q_1, respectively. As it is expected, the figures confirm that the obtained results by using the NLPNN observer are much superior to those obtained using the LPNN observer.

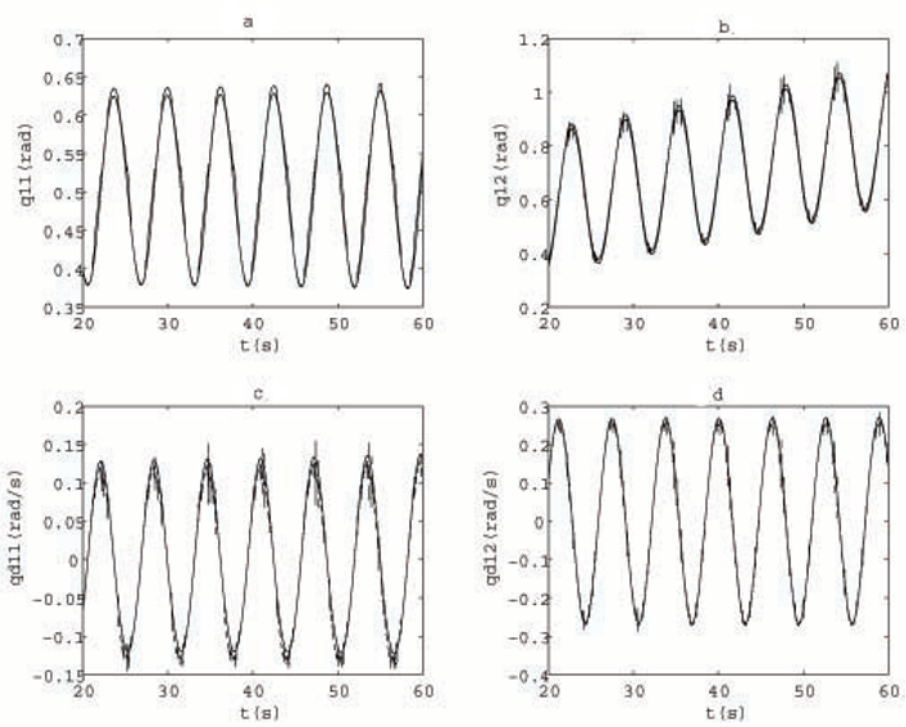

Fig. 2.7 The state responses of the two-link flexible-joint manipulator (after the learning period) to $sin(t)$ reference trajectory for NLPNN: (a) The position of the first link, (b) the position of the second link, (c) the velocity of the first link, (d) the velocity of the second link. The solid lines correspond to the actual states and the dashed lines correspond to the states of the observer.

2.7 Conclusions

Two recurrent neuro-adaptive observers for a general model of MIMO nonlinear systems have been introduced in this chapter. The structure of the proposed stable observers are considered to be either *linear* or *nonlinear* in parameters. The neural network weights have been updated based on the combination of a modified back-

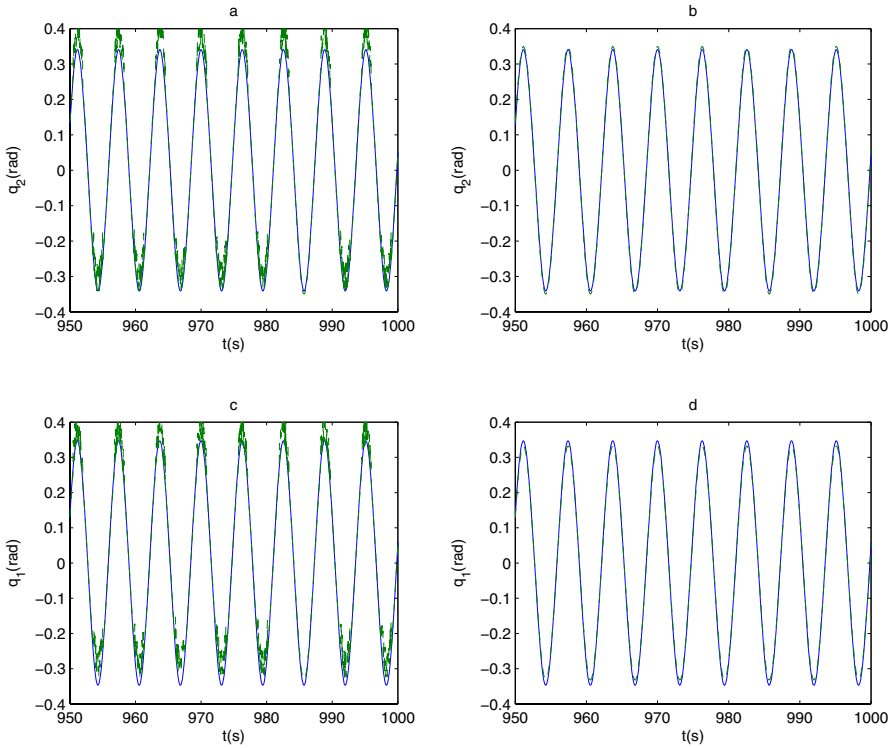

Fig. 2.8 The state responses of the flexible-joint manipulator (after the learning period) to $\sin(t)$ reference trajectory : (a) motor position for LPNN, (b) motor position for NLPNN, (c) link position for LPNN, (d) link position for NLPNN. The solid lines correspond to the actual states and the dashed lines correspond to the states of the observer.

propagation algorithm and an e-modification that guarantees the boundedness of the state estimation error. The stability of the overall system was shown by Lyapunov's direct method. It is worth noting that no SPR assumption or any other constraints that restrict the applicability of the approach was imposed on the system. The proposed observer can be applied both as an online and an off-line estimator. Simulation results performed on a flexible-joint manipulator confirm the reliable performance of the proposed observer.

http://www.springer.com/978-1-4419-1437-8

Neural Network-Based State Estimation of Nonlinear Systems
Application to Fault Detection and Isolation
Talebi, H.A.; Abdollahi, F.; Patel, R.V.; Khorasani, K.
2010, Softcover
ISBN: 978-1-4419-1437-8