9. Balanced-Realization Based Adaptive Filtering

There has always been a desire to use balanced realizations for adaptive filtering due to many of its interesting properties [10, 12]. For example, a balanced realization is known to have the least parameter sensitivity. Based on the discussions in Chapter 8, this suggests that the balanced realization will have good noise rejection characteristics (robust in the presence of noise), since the wrong parameter estimates, due to the misadjustment caused by noise, will describe a model that is still close to the true system [10]. The balanced realization minimizes the ratio of maximum-to-minimum eigenvalues of the Grammian matrices. DeBrunner heuristically argues that this property should lead to fast convergence of adaptive filters based on a balanced realization [10]. Furthermore, the balanced realization is very useful in model reduction [13]. Due to the absence of a parameterization that can guarantee that the realization stays balanced upon adaptation of the parameters, researchers could not develop a balanced-realization based adaptive filtering algorithm [10, 12]. Ober filled the void in balanced parameterization by developing a canonical representation for balanced realizations [14].

9.1 Balanced Realizations

The state-space system, as described in (8.2), has controllability Grammian **K** and observability Grammian **W**, which are the solutions to the Lyapunov equations

$$\begin{aligned} \mathbf{K} &= \mathbf{A}\mathbf{K}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T \\ \mathbf{W} &= \mathbf{A}^T\mathbf{W}\mathbf{A} + \mathbf{C}^T\mathbf{C} \end{aligned} \tag{9.1}$$

In (9.1) we do not explicitly indicate the dependence of the matrices on θ. We drop θ in the sequel unless the dependence on θ is not obvious. The realization is said to be in the *balanced form* (sometimes *internally balanced form* is used), if the two Grammians are diagonal and equal. That is

$$\mathbf{K} = \mathbf{W} = diag\left(\sigma_1, \sigma_2, \ldots, \sigma_N\right) \tag{9.2}$$

The positive diagonal elements $\sigma_1, \sigma_2, \ldots, \sigma_N$ are usually referred to as the Hankel singular values.

9.2 Parameterization

Ober's parameterization [14] of the balanced realizations of discrete-time systems is based on the corresponding parameterization for continuous time systems. The balanced continuous-time model is then transformed into its discrete-time equivalent using the bilinear transformation. Fortunately, the bilinear transformation preserves the observability and controllability Grammians [14]. Hence, the resulting discrete-time model is also in balanced form.

The balanced realization of an *N*th-order, stable, single-input single-output (SISO), LTI continuous-time system with distinct Hankel singular values is defined in terms of the parameters $\boldsymbol{\theta}_{BAL} = (\sigma_1, \ldots \sigma_N, b_1, \ldots, b_N, s_1, \ldots s_N, d)^T$, where

$$\begin{aligned} &\sigma_1 > \sigma_2 > \cdots > \sigma_N > 0 \\ &b_j > 0,\ s_j = \pm 1, \qquad\qquad \forall j. \end{aligned} \tag{9.3}$$

The Hankel singular values of any randomly chosen *N*th-order SISO, LTI plant will almost surely be distinct. Here, "almost surely" means that the set of exceptions has Lebesgue measure zero.

The inequality constraints shown in (9.3) render the parameterization unique for a given system. These parameters are related to the system matrices $\{\mathbf{A}_c, \mathbf{B}_c, \mathbf{C}_c, \mathbf{D}_c\}$ as follows (here, we use the subscript *c* to emphasize the fact that the underlying system is continuous):

$$\mathbf{A}_c(\boldsymbol{\theta}_{BAL}) = \lfloor a_{ij} \rfloor_{1 \le i,j \le N}, \text{ where } a_{ij} = -\frac{b_i b_j}{s_i s_j \sigma_i + \sigma_j} \tag{9.4a}$$

$$\mathbf{B}_c(\boldsymbol{\theta}_{BAL}) = [b_1 \quad b_2 \quad \cdots \quad b_N]^T \tag{9.4b}$$

$$\mathbf{C}_c(\boldsymbol{\theta}_{BAL}) = [s_1 b_1 \quad s_2 b_2 \quad \cdots \quad s_N b_N] \tag{9.4c}$$

$$\mathbf{D}_c(\boldsymbol{\theta}_{BAL}) = d \tag{9.4d}$$

By direct substitution, it may be easily verified that the above set of system matrices satisfies the continuous-time Lyapunov equations

$$\begin{aligned} \mathbf{A}_c \boldsymbol{\Sigma} + \boldsymbol{\Sigma} \mathbf{A}_c^T &= -\mathbf{B}_c \mathbf{B}_c^T \\ \mathbf{A}_c^T \boldsymbol{\Sigma} + \boldsymbol{\Sigma} \mathbf{A}_c &= -\mathbf{C}_c^T \mathbf{C}_c \end{aligned} \tag{9.5}$$

where $\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, \ldots, \sigma_N)$ is the Grammian matrix. Hence, (9.4) constitutes a balanced realization. The bilinear transformation of the system defined by the continuous-time system matrices in (9.4) yields the equivalent discrete-time system defined by the following system matrices (the subscript *d* is used to denote that the underlying system is discrete):

$$\mathbf{A}_d(\boldsymbol{\theta}_{BAL}) = (\mathbf{I} - \mathbf{A}_c)^{-1}(\mathbf{I} + \mathbf{A}_c) \tag{9.6a}$$

$$\mathbf{B}_d(\boldsymbol{\theta}_{BAL}) = \sqrt{2}(\mathbf{I} - \mathbf{A}_c)^{-1}\mathbf{B}_c \tag{9.6b}$$

$$\mathbf{C}_d(\boldsymbol{\theta}_{BAL}) = \sqrt{2}\mathbf{C}_c(\mathbf{I} - \mathbf{A}_c)^{-1} \tag{9.6c}$$

$$\mathbf{D}_d(\boldsymbol{\theta}_{BAL}) = \mathbf{D}_c + \mathbf{C}_c(\mathbf{I} - \mathbf{A}_c)^{-1}\mathbf{B}_c \tag{9.6d}$$

where $\mathbf{I}$ is the identity matrix of dimension $N \times N$.

It has been proved that the bilinear transformation defined in (9.6) preserves the observability and controllability Grammians and, hence, continuous-time balanced systems are mapped to discrete-time balanced systems under bilinear transformation [14].

9.3 Adaptive Filtering Algorithm

This section will present the least mean squares (LMS) based adaptation algorithm [1] to adapt the parameters $\boldsymbol{\theta}_{BAL}$ of the balanced realization. We first derive the sensitivity formulas that are needed for LMS adaptation. The following identity [17] is used frequently in deriving the gradients:

$$\frac{\partial \mathbf{A}^{-1}}{\partial \alpha} = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial \alpha}\mathbf{A}^{-1} \tag{9.7}$$

The transfer function $H(z)$ in terms of the state-space system matrices is given by

$$H(z; \boldsymbol{\theta}_{BAL}) = \mathbf{C}_d(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d + \mathbf{D}_d \tag{9.8}$$

Using (9.7) and (9.8),

$$\frac{\partial H}{\partial \theta_i} = \frac{\partial \mathbf{C}_d}{\partial \theta_i}\left(z\mathbf{I} - \mathbf{A}_d\right)^{-1}\mathbf{B}_d + \mathbf{C}_d\left(z\mathbf{I} - \mathbf{A}_d\right)^{-1}\frac{\partial \mathbf{B}_d}{\partial \theta_i} + \mathbf{C}_d\left(z\mathbf{I} - \mathbf{A}_d\right)^{-1}\frac{\partial \mathbf{A}_d}{\partial \theta_i}\left(z\mathbf{I} - \mathbf{A}_d\right)^{-1}\mathbf{B}_d + \frac{\partial \mathbf{D}_d}{\partial \theta_i}. \tag{9.9}$$

From (9.6) and (9.7), the derivatives (with respect to the balanced realization parameters $\boldsymbol{\theta}_{BAL}$) of the discrete-time system matrices needed in (9.9) are given by

$$\frac{\partial \mathbf{A}_d}{\partial \theta_i} = \left(\mathbf{I} - \mathbf{A}_c\right)^{-1}\frac{\partial \mathbf{A}_c}{\partial \theta_i}\left[\mathbf{I} + \left(\mathbf{I} - \mathbf{A}_c\right)^{-1}\left(\mathbf{I} + \mathbf{A}_c\right)\right] \tag{9.10a}$$

$$\frac{\partial \mathbf{B}_d}{\partial \theta_i} = \sqrt{2}\left(\mathbf{I} - \mathbf{A}_c\right)^{-1}\left[\frac{\partial \mathbf{A}_c}{\partial \theta_i}\left(\mathbf{I} - \mathbf{A}_c\right)^{-1}\mathbf{B}_c + \frac{\partial \mathbf{B}_c}{\partial \theta_i}\right] \tag{9.10b}$$

$$\frac{\partial \mathbf{C}_d}{\partial \theta_i} = \sqrt{2}\left[\frac{\partial \mathbf{C}_c}{\partial \theta_i} + \mathbf{C}_c\left(\mathbf{I} - \mathbf{A}_c\right)^{-1}\frac{\partial \mathbf{A}_c}{\partial \theta_i}\right]\left(\mathbf{I} - \mathbf{A}_c\right)^{-1} \tag{9.10c}$$

$$\frac{\partial \mathbf{D}_d}{\partial \theta_i} = \frac{\partial \mathbf{D}_c}{\partial \theta_i} + \frac{\partial \mathbf{C}_c}{\partial \theta_i}\left(\mathbf{I} - \mathbf{A}_c\right)^{-1}\mathbf{B}_c + \mathbf{C}_c\left(\mathbf{I} - \mathbf{A}_c\right)^{-1}\frac{\partial \mathbf{A}_c}{\partial \theta_i}\left(\mathbf{I} - \mathbf{A}_c\right)^{-1}\mathbf{B}_c \tag{9.10d}$$

From (9.4), the non-zero derivatives of the continuous-time system matrices that appear in (9.10) are given by

$$\frac{\partial \mathbf{A}_c}{\partial \sigma_i} = \left[a_{jk}^{c\sigma}\right]_{1 \le j,k \le N}, \text{ where} \tag{9.11a}$$

$$a_{jk}^{c\sigma} = \begin{cases} \dfrac{b_i^2}{2\sigma_i^2} & \text{if } i = j = k \\ \dfrac{s_i s_k b_i b_k}{\left(s_i s_k \sigma_i + \sigma_k\right)^2} & \text{if } i = j \neq k \\ \dfrac{b_j b_i}{\left(s_j s_i \sigma_j + \sigma_i\right)^2} & \text{if } i = k \neq j \\ 0 & \text{otherwise} \end{cases} \tag{9.11b}$$

$$\frac{\partial \mathbf{A}_c}{\partial b_i} = \left[a_{jk}^{cb}\right]_{1 \le j,k \le N}, \text{ where} \tag{9.12a}$$

$$a_{jk}^{cb} = \begin{cases} -\dfrac{b_i}{\sigma_i} & if \;\; i = j = k \\ -\dfrac{b_k}{s_i s_k \sigma_i + \sigma_k} & if \;\; i = j \neq k \\ -\dfrac{b_j}{s_j s_i \sigma_j + \sigma_i} & if \; i = k \neq j \\ 0 & otherwise \end{cases} \tag{9.12b}$$

$\frac{\partial \mathbf{A}_c}{\partial s_i} = \left[a_{jk}^{cs}\right]_{1 \le j,k \le N}$, where (9.13a)

$$a_{jk}^{cs} = \begin{cases} \dfrac{s_k \sigma_i b_i b_k}{\left(s_i s_k \sigma_i + \sigma_k\right)^2} & if \;\; i = j \neq k \\ \dfrac{s_j \sigma_j b_j b_i}{\left(s_j s_i \sigma_j + \sigma_i\right)^2} & if \; i = k \neq j \\ 0 & otherwise \end{cases} \tag{9.13b}$$

$$\frac{\partial \mathbf{B}_c}{\partial b_i} = \mathbf{e}_i \tag{9.14}$$

$$\frac{\partial \mathbf{C}_c}{\partial b_i} = s_i \mathbf{e}_i \tag{9.15}$$

$$\frac{\partial \mathbf{C}_c}{\partial s_i} = b_i \mathbf{e}_i \tag{9.16}$$

$$\frac{\partial \mathbf{D}_c}{\partial d} = 1 \tag{9.17}$$

Figure 9.1 shows the configuration of the state-space adaptive filter. The adaptive filter attempts to minimize the mean square of the error signal e_n, which is the difference between the desired output y_n and the estimated output $\hat{y}_n$. That is,

$$e_n = y_n - \hat{y}_n . \tag{9.18}$$

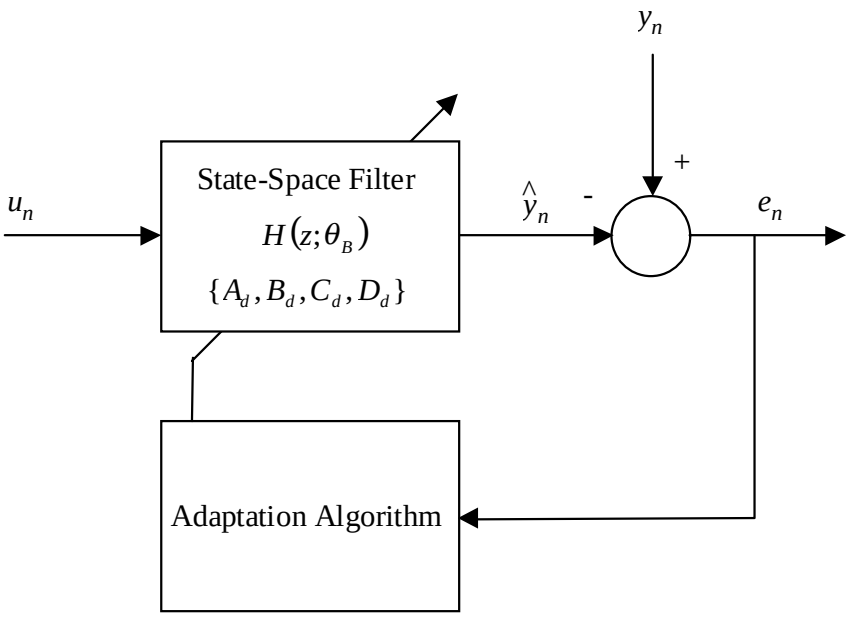

Figure 9.1 State-Space Adaptive Filter.

The LMS algorithm approximates the expected value of the squared error signal using its instantaneous value. With such an approximation, the parameter adaptation equation is given by

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \mu e_n \frac{\partial \hat{y}_n}{\partial \boldsymbol{\theta}}, \tag{9.19}$$

where

$$\frac{\partial \hat{y}_n}{\partial \boldsymbol{\theta}} = \frac{\partial H}{\partial \boldsymbol{\theta}} u_n \tag{9.20}$$

is the gradient of the estimated output with respect to the parameter vector. From (9.9) and (9.20), we recognize that this gradient may be computed using a bank of state-space filters. The structure of the gradient-computing filter is shown in Figure 9.2. The system matrices corresponding to each block are indicated within the block.

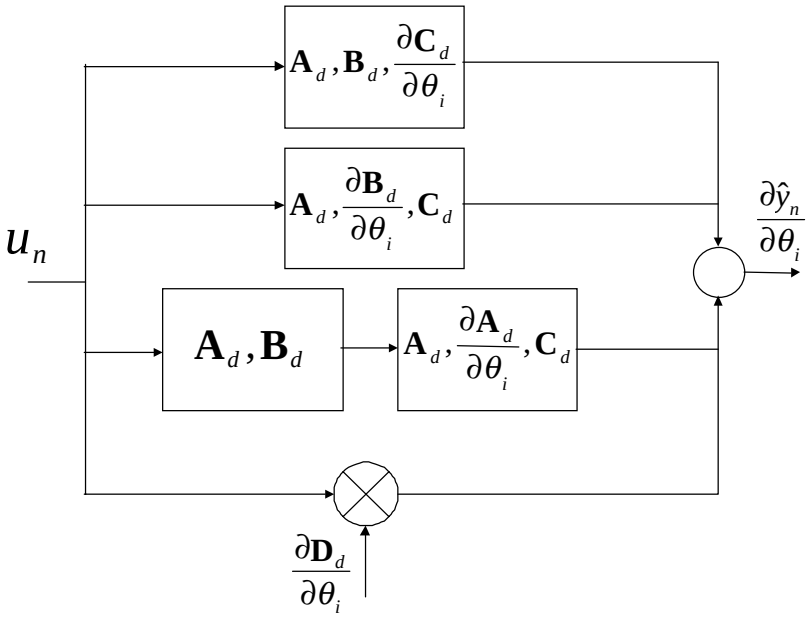

Figure 9.2 Gradient-Computing Filter.

The following is a summary of the adaptive filtering algorithm.

1. Initialize $\boldsymbol{\theta}_0$. Repeat Steps (2)-(6) for $n > 0$.
2. Transform the balanced parameters $\boldsymbol{\theta}_n$ into discrete-time system matrices using (9.4) and (9.6).
3. Compute the estimate for output $\hat{y}_n$ using the state-space equations as in (8.2).
4. Estimate the error e_n in the estimate using (9.18).
5. Compute the gradient of the output with respect to the parameters using (9.9) and (9.20).
6. Adapt the parameters using (9.19). Use any projection technique [39] to ensure that the constraints in (9.3) are satisfied.

Thus, (9.4), (9.6), (8.2), (9.18), (9.9), (9.20) and (9.19), in the specified order, constitute the balanced-realization based LMS adaptation algorithm.

The adaptive filtering algorithm presented above attempts to minimize the mean-squared output error. The global minimum of the output-error surface is known to give an estimate for the true parameters of the system [12]. However, it is not easy to locate the global minimum of the output-error surface due to the presence of additional local minima, unless we have a "good" starting point. We surmount this problem by using an equation-error (EE) based adaptation algorithm [12] to obtain an initial estimate for the parameters. That is, the stable, biased estimate

obtained from the equation-error based adaptive filter is used as the initial estimate for the balanced-realization based (output-error minimization) algorithm.

9.4 Simulation Results

The proposed algorithm is simulated in MATLAB. The plant to be modeled is a fourth-order elliptic low-pass filter (LPF) with $0.05 f_s$ as cutoff frequency. We assume that 16,000 input-output measurements (free of any measurement noise) from this plant are available. The initial 4,000 data pairs are used to adapt the equation-error based adaptive filter. The estimate obtained from the equation-error adaptive filter is transformed into balanced parameters. The latter 12,000 data pairs are used to adapt the balanced-realization based adaptive filter. Figure 9.3 shows the learning curve, which is a plot of the output estimation error vs. iteration number, for the adaptation algorithm. The true and estimated magnitude responses (estimate obtained using 16,000 data samples) are shown in Figure 9.4. The difference between the true and estimated magnitude responses is barely noticeable.

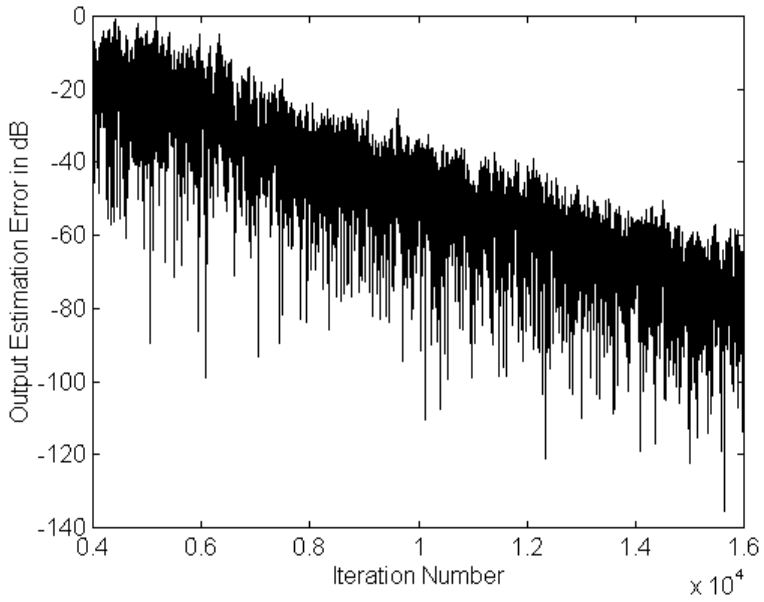

Figure 9.3 Learning Curve of the Proposed Algorithm. (Plant: 4th Order Elliptic LPF)

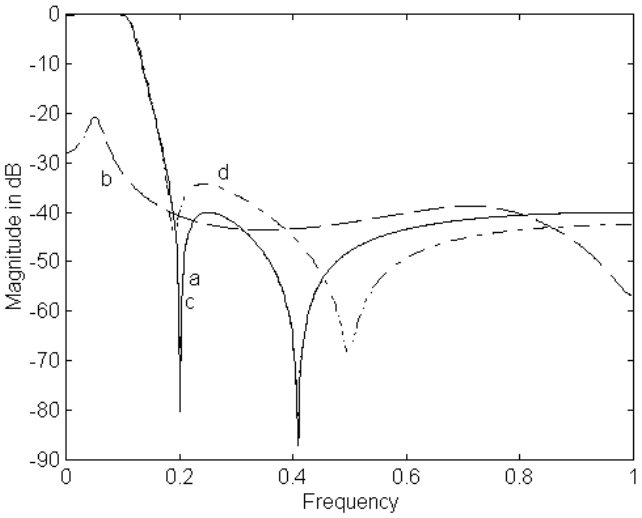

Figure 9.4 True and Estimated Magnitude Responses: (a) True Response, (b) Initial Estimate (from EE method), and Estimate from Proposed Algorithm (c) Noise-free case and (d) SNR = 20 dB.

Figure 9.5 shows the learning curve for the proposed algorithm under the same conditions as above, but with a desired signal that includes measurement noise at -29 dB. The signal strength is –9 dB resulting in a signal-to-measurement-noise ratio of 20 dB. We find that the steady-state MSE is -27.54 dB. Due to the presence of the measurement noise in the desired signal, the mean-squared estimation error is limited to -29 dB. The estimated magnitude is shown in Figure 9.4.

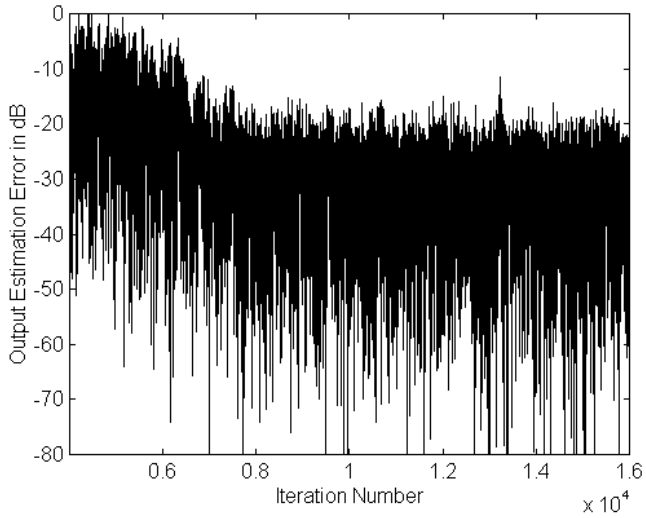

Figure 9.5 Learning Curve of the Proposed Algorithm. (Plant: 4th Order Elliptic LPF, Output SNR: 20 dB)

9.5 Conclusion

The proposed algorithm attempts to minimize the mean-squared output error. Hence, the resulting estimates are unbiased. The problem of the existence of local minimums in the output-error surface is surmounted by first using an equation-error based adaptation algorithm, which provides a good initial estimate for the parameters.

The balanced parameterization presented here inherently ensures that the adaptive filter always remains stable. The adaptation algorithm insures that the Hankel singular values are always positive and this positivity, in turn, guarantees that the system is stable. Consequently, the usual complex stability checks are not needed while using our balanced-realization based adaptive filter.