227-0689-00L System Identification	**06 December 2023**

Solution 9: Prediction error methods

We will not provide solutions for the MATLAB parts of the exercises. These will instead be discussed in the exercise sessions.

Background reading

The background material for this exercise is Sections 3.2, 4.2, and 7.2 of Ljung (*System Identification; Theory for the User*, 2nd Ed., Prentice-Hall, 1999).

Problem 1:

We are given a linear, discrete time system of the form:

$$y(k) = G(\theta, z)u(k) + H(\theta, z)e(k),$$

where $e(k)$ is normally distributed, white noise with unit variance. Consider a prediction error identification of the plant with the quadratic cost function given as:

$$V_N(\theta, Z_N) = \frac{1}{N}\sum_{k=0}^{N-1}\frac{1}{2}\epsilon^2(k,\theta),$$

where $\epsilon(k,\theta)$ represents the parametrized prediction error. Write the cost function $V_N(\theta, Z_N)$ in terms of the transfer functions G and H in frequency domain and the empirical transfer function (ETFE) $\hat{G}(e^{2\pi jk/N}) = \frac{Y_N(2\pi k/N)}{U_N(2\pi k/N)}$, with $k = 0, 1, \dots, N-1$. Based on this derivation give an interpretation of the prediction error method in terms of frequency domain identification.

Hint: You may use the Parseval's relation: $\sum\limits_{k=0}^{N-1} u^2(k) = \sum\limits_{k=0}^{N-1} |U_N(2\pi k/N)|^2$.

Solution

From the Parseval's rule it follows that

$$V_n(\theta, Z_N = \frac{1}{2N}\sum_{k=0}^{N-1}\epsilon^2(\theta,k) = \frac{1}{2N}\sum_{k=0}^{N-1}|E_N(\theta, 2\pi k/N)|^2, \tag{9.1}$$

where $E_N(\theta, 2\pi k/N)$ represents the DFT of the prediction error $\epsilon(k,\theta)$. On the other hand, the prediction error is given by:

$$\epsilon(k,\theta) = H^{-1}(\theta,z)[y(k) - G(\theta,z)u(k)],$$

And therefore, it holds that:

$$E_N(2\pi k/N) = H^{-1}(e^{2\pi k/N})[Y_N(2\pi k/N) - G(e^{2\pi k/N})U_N(2\pi k/N)] + R_N(2\pi k/N), k = 0, 1, \ldots, N-1,$$

where $R_N(2\pi k/N)$ represents the transient that satisfies $|R_N(2\pi k/N)| \leq \frac{C}{\sqrt{N}}$, with C being a finite positive constant. By using the definition of the ETFE: $\hat{G}(e^{2\pi jk/N}) = \frac{Y_N(2\pi k/N)}{U_N(2\pi k/N)}$ and substituting everything into equation (9.1), we obtain:

$$V_N(\theta, Z_N) = \frac{1}{N}\sum_{k=0}^{N-1}\frac{1}{2}\left|\hat{G}(e^{2\pi jk/N}) - G(e^{2\pi jk/N})\right|^2 \frac{|U_N(2\pi k/N)|^2}{|H(e^{2\pi k/N})|^2} + \overline{R}_N,$$

where $\overline{R}_N$ represents an equivalent term due to transient, which goes to 0 as N goes to infinity. Therefore the prediction error method can be seen as a method for fitting the ETFE to the transfer function of the model with a weighted norm that is inversely proportional to the asymptotic variance (or in other words proportional to the signal to noise ratio) at a given frequency.

Problem 2:

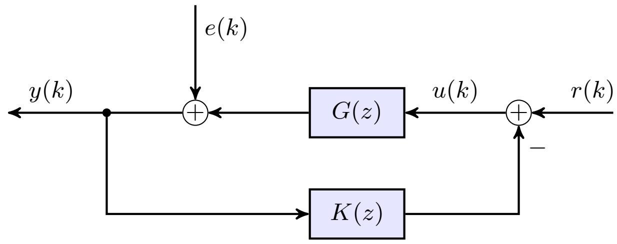

Figure 9.1: Closed-loop system

Consider the closed-loop system with the structure given in Figure 9.1. It is known that the plant has the form $G(z) = \dfrac{B(z)}{A(z)}$, where $B(z) = b_1 z^{-1}$ and $A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}$. The signal $e(k)$ entering at the output is zero-mean white noise. The controller has the form $K(z) = z^{-1} - k_2 z^{-2}$, with $k_2 = 0.1$. Assume that you are given measurement data of $r(k)$ and $y(k)$ from one single experiment.

Derive, for the given closed-loop system, the output predictor in the pseudo-linear regressor form:

$$y_{\text{est}}(k|\theta) = \varphi^T(k,\theta)\theta.$$

Solution

In order to derive the output predictor, we intend to use the following formula that has been derived for open loop systems (Equation (4.6) in Ljung):

$$y_{\text{est}} = H^{-1}(z)G(z)u + (1 - H^{-1}(z))y, \tag{9.2}$$

where $H(z)$ is the transfer function acting on a white noise signal $e(t)$ before it affects the signal $G(z)u$. However, in the problem, we consider a closed loop system where $u = K(z)y$. We can derive an equivalent open loop system which has $r(t)$ as the input and $y(t)$ as the output. Consider the dependence of y on r, e as:

$$\begin{aligned} y &= e + G(z)u = e + G(z)(r - K(z)y) \\ \implies \quad (1 + G(z)K(z))y &= e + G(z)r \\ \implies \quad y &= \frac{G(z)}{1 + G(z)K(z)} r + \frac{1}{1 + G(z)K(z)} e = \hat{G}(z)r + \hat{H}(z)e \end{aligned} \tag{9.3}$$

By plugging (9.3) into (9.2), we obtain

$$\begin{aligned} y_{\text{est}} &= \hat{H}^{-1}(z)\hat{G}(z)r + (1 - \hat{H}^{-1}(z))y \\ &= \frac{B(z)}{A(z)} r + \frac{-B(z)K(z)}{A(z)} y, \end{aligned}$$

which can be multiplied by $A(z)$ to obtain

$$y_{\text{est}} = (1 - A(z))y_{\text{est}} + B(z)(r - K(z)y)$$

Note that this is a pseudolinear form, since the predictor uses past values of y_{est} in the regressor. Defining the vector $\theta = \begin{bmatrix} a_1 & a_2 & a_3 & b_1 \end{bmatrix}^\intercal$, the output predictor can be written as

$$\begin{aligned} y_{\text{est}}(k|\theta) &= \begin{bmatrix} -y_{\text{est}}(k{-}1|\theta) & -y_{\text{est}}(k{-}2|\theta) & -y_{\text{est}}(k{-}3|\theta) & \big(r(k{-}1){-}y(k{-}2){+}0.1y(k{-}3)\big) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \end{bmatrix} \\ &= \varphi^T(k,\theta)\theta. \end{aligned}$$

MATLAB **Exercise:**

Consider the ARMAX model

$$A(z)y(k) = B(z)u(k) + C(z)e(k),\ k = 1, \ldots, N$$

with the matrices defined by

$$\begin{aligned} A(z) &= 1 - 1.5z^{-1} + 0.7z^{-2} \\ B(z) &= 1.0z^{-1} + 0.4z^{-2} \\ C(z) &= 1 - 1.1z^{-1} + 0.4z^{-2}, \end{aligned}$$

where $e(k) \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$. Fix an input sequence $u(k)$ from the following ARMAX process

$$u(k) = 0.14u(k-1) + 0.12\,u(k-2) + 0.9e_u(k-1) + 0.3\,e_u(k-2).$$

where $e_u \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$.

Assuming that the transfer function $C(z)$ is exactly known:

1. Obtain least-squares (LS) estimates $\hat{A}_{LS}(z)$ and $\hat{B}_{LS}(z)$ for $A(z)$ and $B(z)$ respectively.
2. Plot the predicted values along with the true response $y(k)$ for validation set.
3. Repeat (1) for a different set of realizations of $e(k)$ and plot the histograms of the parameters.

Solution hints:

The first thing that should be done is to form the discrete time systems that will be used in order to generate relevant system data. These are the transfer functions given by $\frac{B(z)}{A(z)}$ and $\frac{C(z)}{A(z)}$ and the transfer function needed to generate $u(k) = L(z)e_u(k)$. These transfer functions can be generated by using the *tf* command. Random signals e and v can be created by using the *randn* command. The signal u can then be generated from the signal v and the output signal y can be generated from u and e by using the *lsim* command.

1. The output can be written in the following form

   $$y(k) = \frac{B(z)}{A(z)}u(k) + \frac{C(z)}{A(z)}e(k)$$

   Since $C(z)$ is exactly known by assumption, it can be verified that it is stably invertible and we can write the following

   $$C^{-1}(z)y(k) = \frac{B(z)}{A(z)}C^{-1}(z)u(k) + \frac{1}{A(z)}e(k).$$

By bringing the ARMAX model into the ARX model form (see slide 10.39), we can proceed with the Least Square Estimate after redefining the input as $u_F(k) = C^{-1}(z)u(k)$ and the output $y_F(k) = C^{-1}(z)y(k)$. We first write the regressor form,

$$y_F(k) = \varphi(k)^T\theta,$$

where

$$\varphi(k) = \begin{bmatrix} -y_F(k-1) \\ -y_F(k-2) \\ u_F(k-1) \\ u_F(k-2) \end{bmatrix}, \theta = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix}.$$

The least squares estimate of θ can then be calculated in MATLAB by the comand $\theta = \Phi \backslash Y_F$, where $Y_F = [y_F(1), \ldots, y_F(N)]^T$ and,

$$\Phi = \begin{bmatrix} \varphi(1)^T \\ \vdots \\ \varphi(N)^\top \end{bmatrix}.$$

Note that if the transfer function $C(z)$ was not known and to be estimated along with the matrices $A(z)$ and $B(z)$, then the function `fmincon` can be used to identify $C(z)$.

2. In order to compare the measurements with the predictions, the appropriate transfer function $\frac{B_{LS}(z)}{A_{LS}(z)}$ needs to be formed from the parameters in $\hat{\theta}$ by using the *tf* command. Then by using the *lsim* command on the same input and noise sequence but with real and estimated transfer functions, one can obtain the actual and predicted outputs. The comparison is shown in Fig. 9.2. An alternative method to calculate the residual at each time step k is to take the difference between the true output $y(k)$ and the estimated output $\hat{y}(k) = C\hat{y}_F(k), \hat{y}_F(k) = \varphi(k)\hat{\theta}$, where $\hat{\theta}$ is the vector containing the estimated parameters.

3. By repeating the procedure described in point 1. for different realizations of $e(k)$, we get a sequence of estimated values for each of the parameters a_1, a_2, b_1 and b_2. The histograms of the obtained values can be plotted by using the *histogram* command. Fig. 9.3 shows the obtained histograms.

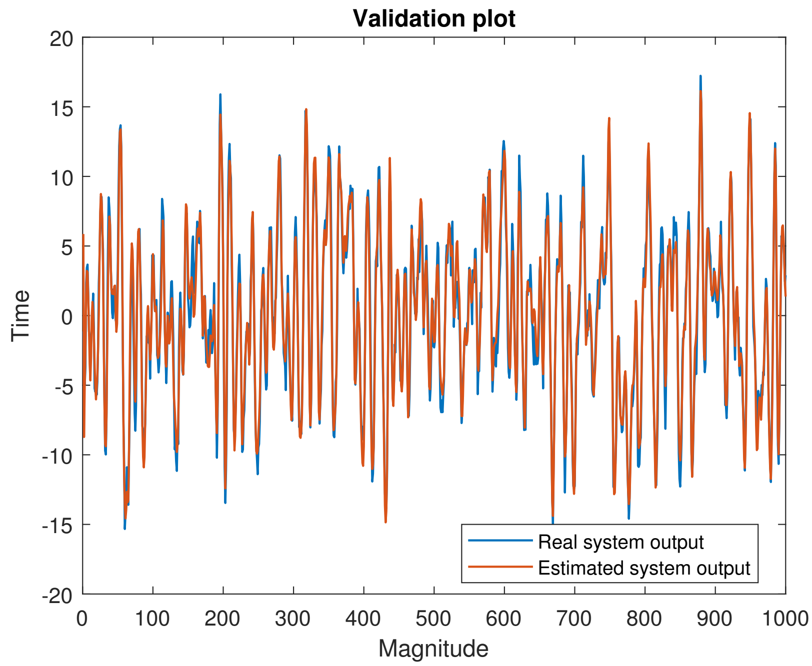

Figure 9.2: Comparison of the true and predicted values of $y(k)$.

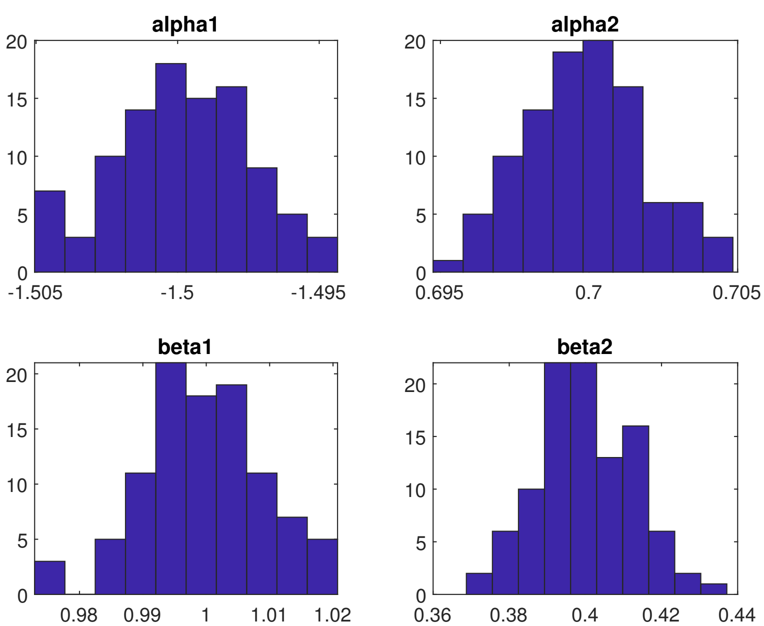

Figure 9.3: Histograms of the estimated parameters a_1, a_2, b_1, b_2.