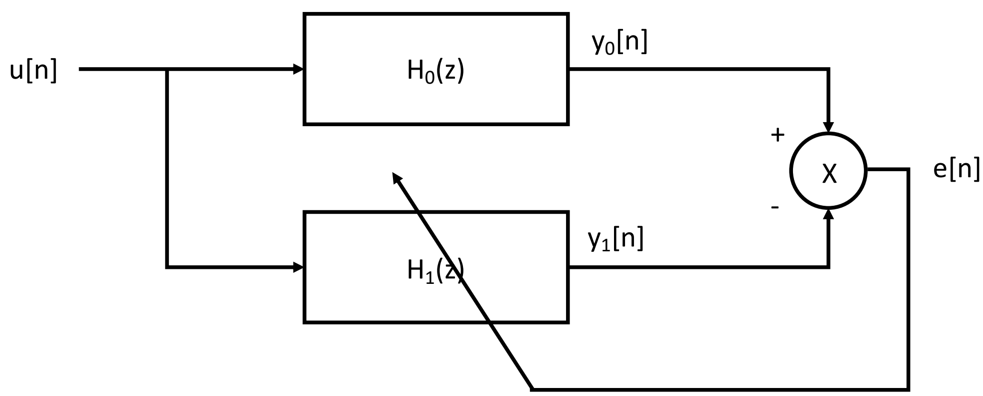
System Identification Problem

Hailan Shanbhag and Naresh Shanbhag

June 2020

1 Problem Statement



Consider an unknown system described by $H_0(z)$. The unknown system is a black-box, i.e., we can control only its input $u[n]$ and observe its output $y_0[n]$. Here $n = 0, \ldots, \infty$ is the time-index. It is known that output $y_0[n]$ of the unknown system $H_0(z)$ is related to its input via the following *difference equation*:

$$y_0[n] = \sum_{k=0}^{N-1} h_0[k]u[n-k] + \eta[n] \tag{1}$$

where $h_0[k]$ for $k = 0, \ldots, N-1$ are the coefficients and N is the order of the unknown system $H_0(z)$. Also, $\eta[n]$ is the *measurement noise* at the output of the unknown system. The coefficients $h_0[k]$, the order N and the noise $\eta[n]$ are *unknown* and need to be estimated.

We wish to estimate these unknown quantities by implementing an adaptive filter $H_1(z)$ that *learns* the unknown parameters of $H_0(z)$ using the configuration shown in Figure 1. The adaptive filter

$H_1(z)$ takes the same input $u[n]$ as the unknown system $H_0(z)$. The outputs of $H_0(z)$ and $H_1(z)$ are subtracted to generate an error signal $e[n]$. This error signal is used to update the values of the coefficients of $H_0(z)$. These operations are described below:

$$y_1[n] = \sum_{k=0}^{M-1} h_1[k,n]u[n-k] \tag{2}$$

$$e[n] = y_0[n] - y_1[n] \tag{3}$$

$$h_1[k,n+1] = h_1[k,n] + \mu e[n]u[n-k] \ , \quad k = 0, 1, ..., M-1 \tag{4}$$

Equation 3 computes $e[n]$ which is then used to find $h_1[k, n+1]$ via Equation 4, which is the value of the filter coefficient in the next time index.

You need to write a Python code that takes as inputs $(u[n], y_0[n])$ and generates the following outputs:

a) an estimate $\hat{N}$ of the value of N.

b) estimates $\hat{h}_0[k] = h_1[k, \infty]$.

c) an estimate of the variance $\hat{\sigma}_\eta^2$ of $\eta[n]$.

In order to realize this objective, you are provided with the following information:

- 800 samples of the 2-tuple $(u[n], y_0[n])$, i.e., the input-output pair at time index n of the unknown system. This data can be found in the file in the attached CSV file ”data.csv”.

2 Solution Steps

Please follow the steps below to obtain the solution:

1. **FIR filter**: write a Python code that implements the following difference equation:

$$y[n] = \sum_{k=0}^{M-1} h[k]x[n-k] \tag{5}$$

   Validate your code by assigning random values for the coefficients $h[k]$, your choice of filter order M and setting $\{x[0] = 1.0, x[1] = 0.0, x[2] = 0.0, \ldots, x[799] = 0.0\}$. The resulting output $y[n]$ should equal the coefficients $h[k]$ for $k = 0, \ldots, M-1$ and zero everywhere else.

2. **Analyzing data**:

   (a) Import the data from data.csv.

   (b) Initialize $h_1[k, 0] = 0$ for $k = 0, \ldots, M-1$ for your choice of M.

   (c) Feed the input signal $u[n]$ into the filter $h_1[k, n]$ to compute $y_1[n]$, $e[n]$, and update $h_1[k, n+1]$ according to Equations 2, 3 and 4.

(d) Record $e[n]$ and $h_1[k,n]$ for $n = 0, \ldots, 799$.

(e) Execute steps (b), (c) and (d) 50 times ($i = 0, 1, ..., 49$).

3. **Evaluating data**:

   (a) Find the ensemble averages $E[e^2[n]]$ and $E[h_1^{(i)}[k,n]]$:

$$E[e^2[n]] = \frac{1}{50}\sum_{i=0}^{49} e_i^2[n] \qquad \text{where } e_i[n] = e[n] \text{ in the } i^{th} \text{ run}$$

$$E[h_1^{(i)}[k,n]] = \frac{1}{50}\sum_{i=0}^{49} h_1^{(i)}[k,n] \qquad \text{where } h_1^{(i)}[k,n] = h_1[k,n] \text{ in the } i^{th} \text{ run}$$

   (b) Plot $E[e^2[n]]$ vs. n and $E[h_1[k,n]]$ vs. n. The plot of $E[e^2[n]]$ vs. n should reduce over time indicating that $E[h_1[k,799]]$ is a good estimate of $h_0[k]$.

   (c) Report $\hat{N}$, $\hat{h}_0[k] = h_1[k,799]$, and the variance $\hat{\sigma}_\eta^2$.

4. **Random initial coefficients**: Repeat steps 2 and 3 with random values for $h_1[k,0]$. Do the estimates in 3(c) change with intial values $h_1[k,0]$?

Submit your Python code properly commented and a short report describing the work you did and results.