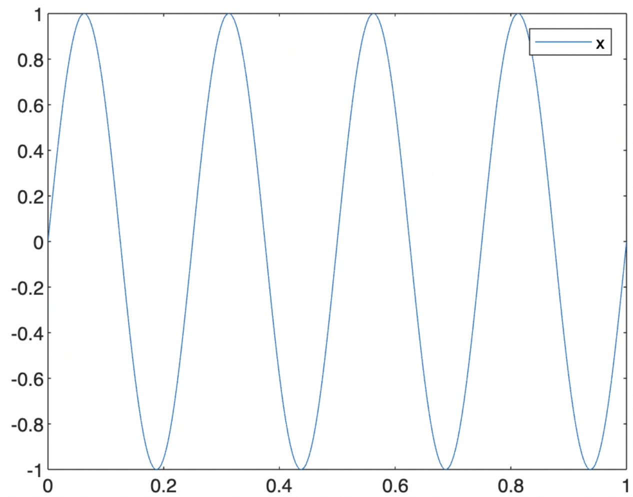
LMS Algorithm

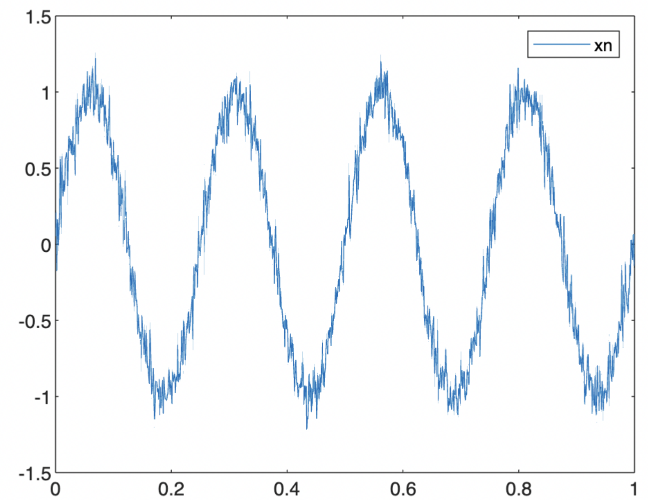
The code LMS.m implements an adaptive filter using the Least Mean Squares (LMS) algorithm to estimate the coefficients of a filter. Here is a step-by-step explanation of what the code does:

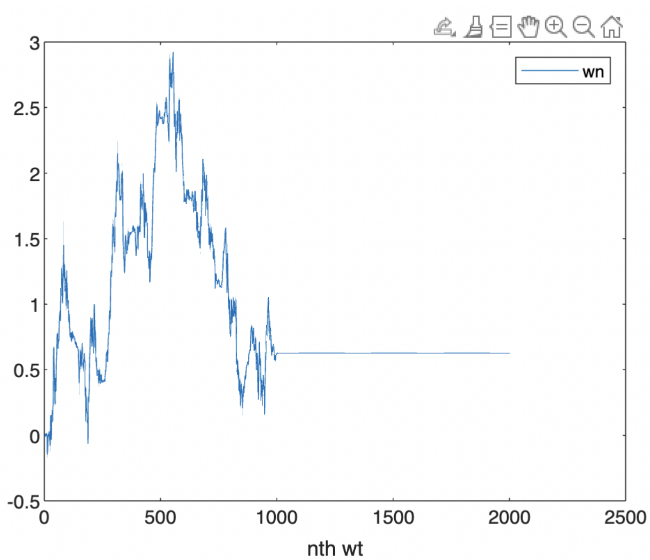
1. The code defines the sampling frequency `Fs`, the sampling interval `ts`, and creates a time vector `t` from 0 to 1 with a step of `ts`.
2. It defines a sinusoidal input signal `x` with a frequency of 8*pi Hz.
3. It adds white Gaussian noise to the input signal `x` to create a noisy input signal `xn`.
4. The code plots the original signal `x` and the noisy signal `xn` in separate figures.
5. It initializes the filter coefficients `wn` as an array of zeros and defines the impulse response `hn` as an array of ones divided by the length of `x`.
6. It generates a random reference signal `dn` with the same length as `xn`.
7. It initializes arrays for the filtered output `yn` and the error signal `en`.
8. The code enters a loop that iteratively updates the filter coefficients based on the LMS algorithm.
9. Within the loop, it adjusts the lengths of `xn` and `wn` to match the current iteration.
10. It calculates the filtered output `yn` by convolving `xn` with `wn`.
11. It computes the error signal `en` as the difference between the reference signal `dn` and the filtered output `yn`.
12. It updates the filter coefficients `wn` using the LMS update rule.
13. The loop continues for each sample of the input signal `x`.
14. Finally, the code plots the estimated filter coefficients `wn` and the error signal `en` in separate figures.

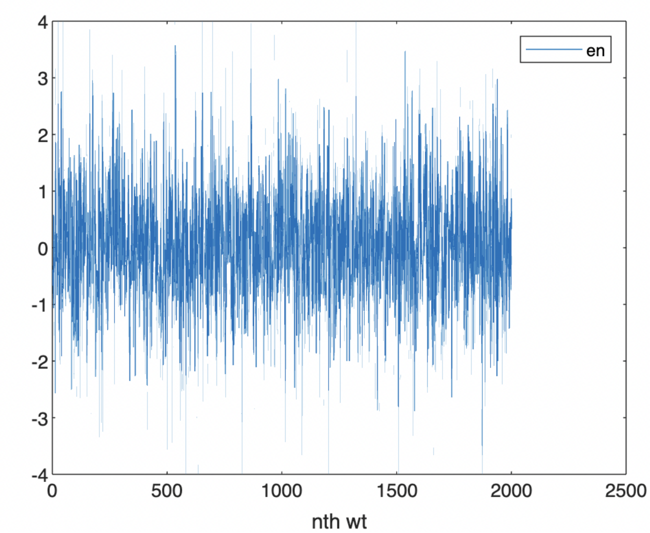
In summary, this code demonstrates the application of the LMS algorithm to estimate the filter coefficients in an adaptive filter by iteratively adjusting the coefficients based on the input signal and the desired response.

Graphs :









FxLMS

The code FxLMS.m implements an adaptive control system using the Least Mean Square (LMS) algorithm for system identification and the Filtered-x Least Mean Square (FxLMS) algorithm for active control. Here is a brief explanation of the code:

1. Initialization: The code initializes the parameters, such as the system impulse response coefficients (Pw and Sw), the length of the data (T), and the learning rate (mu).

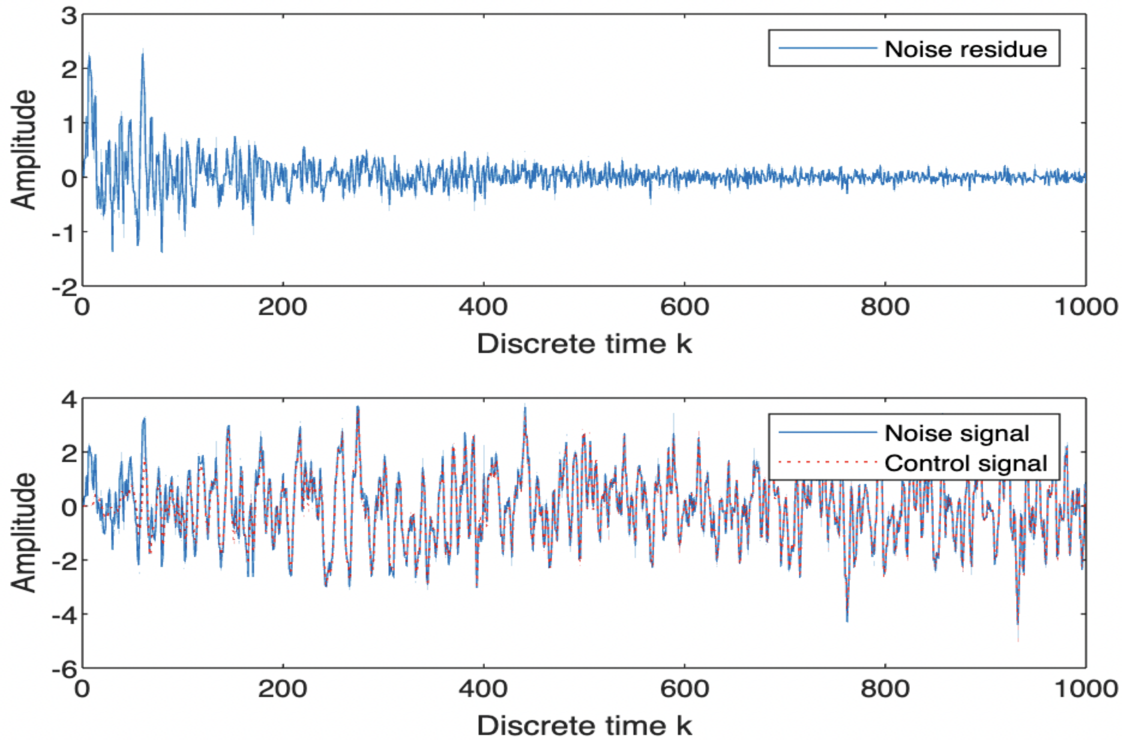
2. System Identification: A white noise signal (x_iden) is generated and passed through a simulated system with impulse response coefficients Sw. The output of the system (y_iden) is measured, and the identification process begins. The LMS algorithm is applied to update the weights (Shw) of the adaptive filter Sh(z) to minimize the identification error (e_iden).

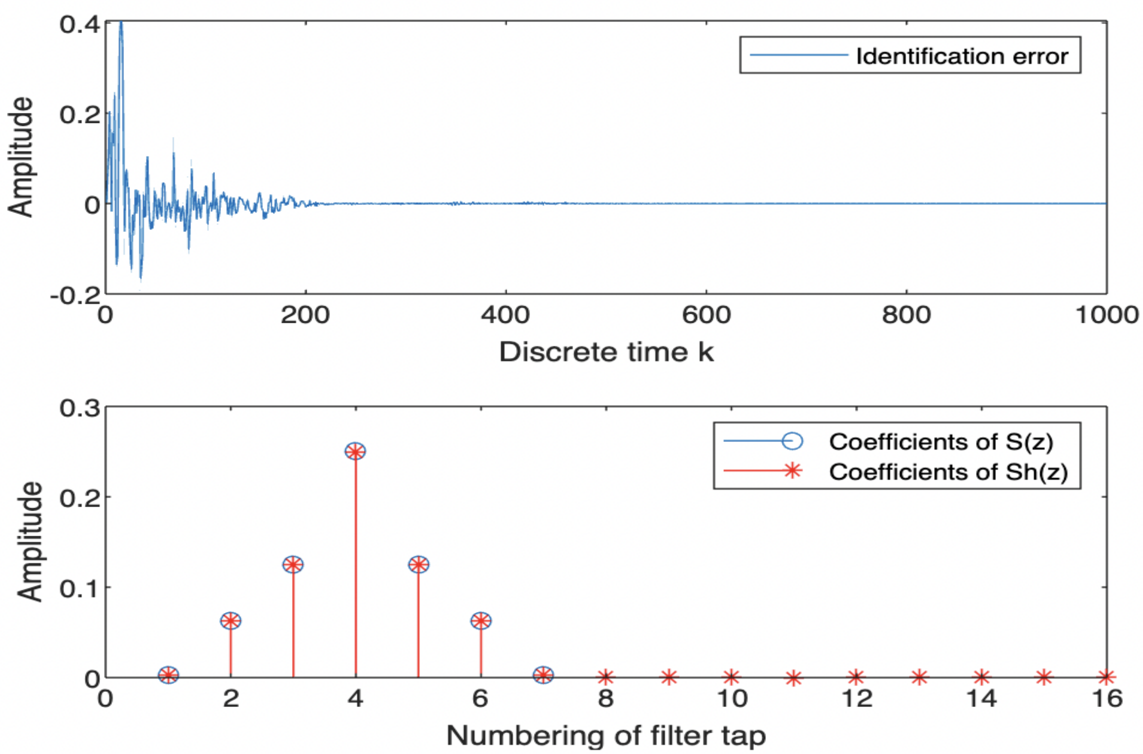
3. Active Control: Another white noise signal (X) is generated as the control input. The arriving noise at the sensor position (Yd) is measured by passing X through a simulated system with impulse response coefficients Pw. The control process starts by initializing the controller state (Cx), weights (Cw), and other variables. The FxLMS algorithm is applied, which includes the adaptation of both the controller weights (Cw) and the weights of the adaptive filter (Shw) based on the control error (e_cont) and the filtered input signal (Xhx).

4. Result Visualization: The code plots the identification error (e_iden) and the coefficients of the true system (Sw) and the adapted system (Shw) in the first subplot. In the second subplot, it displays the control error (e_cont) and compares the noise signal (Yd) with the control signal (Yd - e_cont).

The code demonstrates how the LMS and FxLMS algorithms can be used for system identification and active noise control, respectively. By adapting the filter weights based on the error signals, the algorithms aim to minimize the difference between the desired and actual system responses.

Graphs


Noise residue
Amplitude
Discrete time k
Noise signal
Control signal
Amplitude
Discrete time k


Identification error
Amplitude
Discrete time k
Coefficients of S(z)
Coefficients of Sh(z)
Amplitude
Numbering of filter tap