INVERSE SYSTEM IDENTIFICATION TO COMPENSATE FOR THE COPPER TRANSMISSION MEDIUM BY ADAPTIVE FILTER

Engr. Lubna Farhi

lubnafarhi@yahoo.com
Assistant Professor
Sir Syed Engineering University & Technology, Karachi

ABSTRACT

This paper has been written to explain by placing to compensate for the copper transmission medium (unknown system) in series with adaptive filter, filter becomes the inverse of the unknown system . The process requires a delay inserted in the desired signal path to keep the data at the summation synchronized Adding the delay keeps the system causal.

Keywords: Adaptive Filter,FIR, Steepest Descent ,Widrow-Hopf LMS algorithm ,MATLAB

1. INTRODUCTION

In this paper we investigated the influence of adaptive filters on the performance of inverse system identification by using the LMS algorithm. An innovative and efficient adaptive algorithm constructed to improve both convergence rate and steady-state MSE.

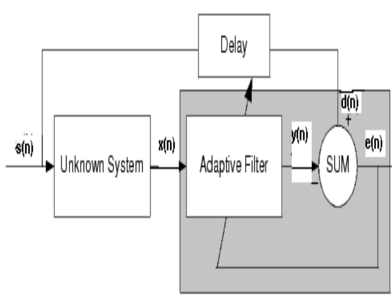
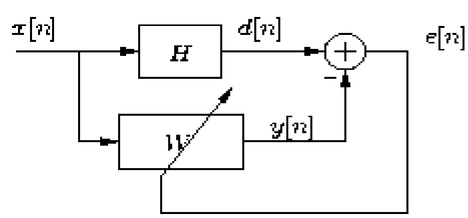


Figure 1: Inverse System identification

The adaptive filter adjusts its coefficients to minimize the mean- square error between its output and that of an unknown system. Fig.2 shows a block diagram of system identification using adaptive filtering. The objective is to change (adapt) the coefficients of an FIR filter, W, to match as closely as possible the response of an unknown system, H. The unknown system and the adapting filter process the same input signal x[n] and have outputs



d[n](also referred to as the desired signal) and y[n].

Figure 2: System identification

Without the delay element, the adaptive filter algorithm tries to match the output from the adaptive filter (y(n) to input data (x(n) that has not yet reached the adaptive elements because it is passing through the unknown system. In essence, the filter ends up trying to look ahead in time. As hard as it tries, the filter can never adapt: e(n) never reaches a very small value and your adaptive filter never compensates for the unknown system response. And it never provides a true inverse response to the unknown system. Including a delay equal to the delay caused by the unknown system prevents this condition.

2. ADAPTIVE SIGNAL IDENTIFICATION

An Adaptive filter consists of two distinct components:

- A digital filter with adjustable coefficients
- An Adaptive algorithm to modify the coefficients of the filter to the input changes.
- Main object: to produce an optimum estimate desired signal

2.1 Algorithm

In Figure 3 two input signals yk & xk are applied to the Adaptive filter simultaneously.

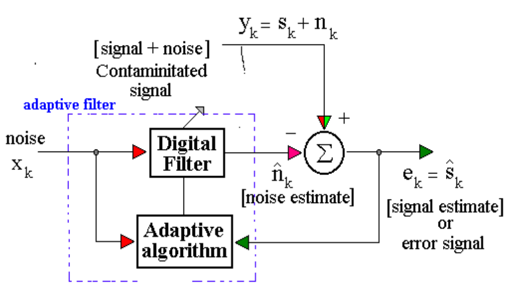


Figure 3: Adaptive identification

Here the yk is the contaminated signal containing both noise nk and the desired signal sk & xk is a measure of yk.

$$y_k = s_k + n_k \quad \cdots\cdots\cdots (1)$$

The digital filter of the system is used to process the nk by producing ; an estimation of nk.
An FIR digital filter is the single input single output system. The FIR Filter is used here instead of IIR because of its simplicity and stability

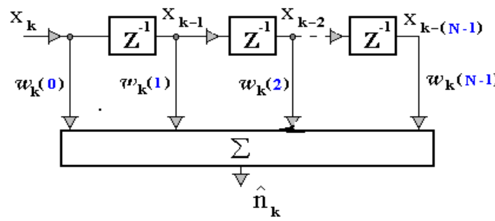


Figure 4: FIR Filter Structure

tap-weight vector, $w_k(i)$

$$W_k = [w_k(0), w_k(1), w_k(2), \ldots, w_k(N-1)]$$

the tap-input vector, x_k

$$X_k = [x_k, x_{k-1}, x_{k-2}, \ldots, x_{k-(N-1)}]$$

FIR filter output, $\hat{n}_k$

$$\hat{n}_k = \sum_{i=0}^{N-1} w_k(i)\, x_{k-1}$$

Here tap-weight vectors are the coefficients to be adjusted.
An estimate of the desired signal is then obtained by subtracting the digital filter output from contaminated signal.

$$\hat{s}_k = y_k - \hat{n}_k = (s_k + n_k) - \hat{n}_k \quad \cdots\cdots\cdots (2)$$

The main objective in noise cancellation is to produce an optimum estimate of the noise in the contaminated signal & hence an optimum estimates of the desired signal.
The $\hat{s}_k$ or ek is feedback to Adaptive algorithm & performs two tasks:

- Desired signal estimation
- Adjustment of filter coefficients

2.2 Adaptive Algorithm

The Adaptive filter algorithm is used to adjust the digital filter coefficients to minimize the error signal, according to some criterion e.g., in the Least Square sense. Thus taking square & mean of error signal:

$$E[\hat{s}_k^2] = E[s_k^2] + E[(n_k - \hat{n}_k)^2] + 2E[s_k(n_k - \hat{n}_k)] \quad \ldots (3)$$

$\hat{n}_k$

Last term in eq. (3) becomes zero because of un-correlation of desired signal with noise & noise estimate.

$$E[\hat{s}_k^2] = E[s_k^2] + E[(n_k - \hat{n}_k)^2] \cdots (4)$$

signal power estimated | total signal power | noise power

The first term in the above equation is estimated of signal power; the second one is the total signal power while the last one is the noise power. If then is the exact replica of nk , the output power contain only the signal power i.e., by adjusting Adaptive filter towards the optimum position, the remnant noise power & hence the total output power are minimized. The desired signal power remains unaffected by this adjustment since sk is uncorrelated with nk. Thus

$$\min E[\hat{s}_k^2] = E[s_k^2] + \min E[(n_k - \hat{n}_k)^2] \cdots (5)$$

This shows that minimizing the total power at the output of the canceller maximizes the signal to noise ration of the output.

Having exact estimate of the noise the last term becomes zero & estimate of desired signal becomes equal to the desired signal. i.e., the output of the canceller becomes noise free:

$$E[\hat{s}_k^2] = E[s_k^2]$$

At this stage adaptive filter turns off (ideally) by setting its weights to zero.

A number of adaptive algorithms are being used like:

- LMS
- RLS
- Kalman

We used LMS here in this system because of the following advantages:

- More efficient because of easy computation & better storage capabilities
- Numerically stable

2.3 Lms Adaptive Algorithm

Many adaptive algorithms can be viewed as an approximation of the discrete Wiener filter. This filter produces an optimum estimate of the part of contaminated signal (that is correlated with input signal), which is then subtracted from the contaminated signal to yield the error signal. Therefore from esq. (2):

$$e_k = y_k - \hat{n}_k$$

or

from FIR output:

$$e_k = y_k - \sum_{i=0}^{N-1} w_k(i)\, x_{k-1} \cdots (6)$$

or

$$e_k = y_k - X_k^T W_k \cdots (7)$$

where

$$W_k = [w_k(0), w_k(1), w_k(2), \ldots, w_k(N-1)]$$

$$X_k = [x_k, x_{k-1}, x_{k-2} \ldots, x_{k-(N-1)}]$$

Instead of computing noise weights in one go, like above the LMS coefficients are adjusted from sample to sample in such a way as to minimize the MSE (mean square error).

The LMS adaptive algorithm is based on the Steepest Descent algorithm, which updates weight vectors sample to sample:

$$W_{k+1} = W_k - \mu \nabla_k \cdots (8)$$

where

W_k weight of signal at kth sampling instant

μ adaptation parameter

∇_k gradient vector

The gradient vector, the cross-correlation between the primary & the secondary inputs P, and the autocorrelation of the primary input, R, are related as

$$\nabla = -2P + 2RW \cdots (9)$$

where

$$P_k \triangleq E[\,y_k X_k\,]$$

$$R_k \triangleq E[\,X_k X_k^T\,]$$

For instantaneous estimates of gradient vector, we can write as

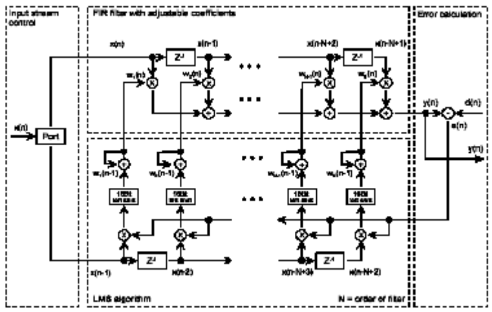


Figure 5: LMS –Filter implementation

$$\nabla_k = -2P_k + 2R_k W_k = -2y_k X_k + 2X_k X_k^T W_k$$

$$= -2\,X_k[\,y_k - X_k^T W_k\,] \cdots (10)$$

From eq. (7) & (10) eq. (8) can be rewritten as

$$W_{k+1} = W_k + 2\mu X_k e_k \cdots (11)$$

This is Widrow-Hopf LMS algorithm.
The LMS algorithm of eq. (11) does not require the prior knowledge signal statistics (R & P) & uses instantaneous estimates to make the system more accurate. These estimates improve gradually with time as the weights of the filter

are adjusted by learning the signal characteristics. But practically the Wk never reaches theoretical optimum of Wiener theory, but fluctuates about it.

2.4 Properties Of LMS Adaptive Filter

Time- varying, self-adjusting.

Deals with Linear and as well as Non-linear systems.

Becomes linear system after their adjustments are held constant after adaptation. Automatically adapts in the face of changing environments and changing system requirements.

Performs specific filtering and decision making tasks, i.e.can be programmed by a training process.

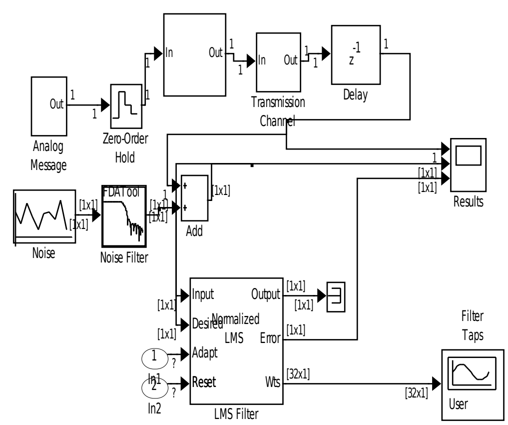
Extrapolates a model of behavior to deal with new situations after having been trained on a finite and often small number of training signals or patterns.

Complex and difficult to analyze than non-adaptive systems, but they offer the possibility of substantially increased system performance when input signal characteristics are unknown or time varying.

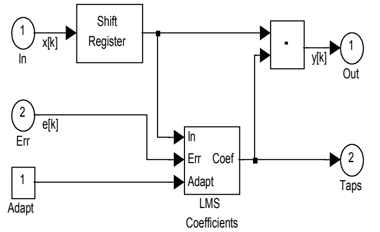
Easier to design than other forms of adaptive systems.

3. MATLAB ALGORITHM

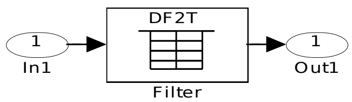
Model implements the LMS Adaptive Filter algorithm by using the MATLAB,



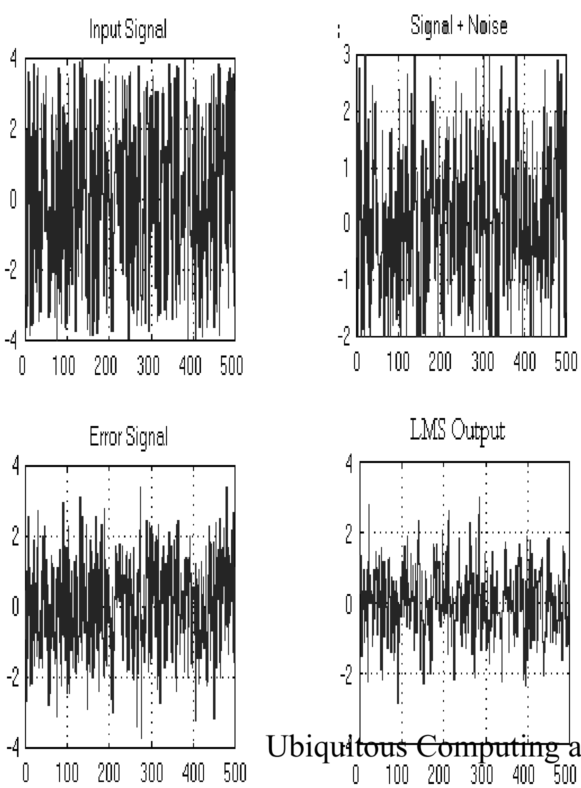
3.1 LMS Adaptive Filter



3.2 Noise Filter



4. SIMULATIONS & RESULTS



5. MATLAB PROGRAM

```
%LMS Signal Identification(lms.mfile)
clear all;close all;
refGain = 1;
worder = 8;
N = 2048;
t=1:N;
signal = sin(2*pi.*t.*t/N/N*8);
%.*fliplr(cos(2*pi.*t.*t/N/N*15))
wreal = randn(1,worder);
ref = conv(additivenoise,wreal);
primary = signal + ref(1:length(signal));
fref = additivenoise*refGain; %real reference
mic
w(1,:) = ones(1,worder);
mu = .1;
%Zero pad so we can start filter at 0 and not
throw of the index
frefpad = [zeros(1,worder -1) fref];

start = flops;
for n = 1:N;
    %offset n so we can reference the correct
value in zero-padded fref
  m = n + worder -1;
  frefblock = frefpad(m-worder+1:1:m)';
     refP(n) = w(n,:)*(frefblock);%adding extra
random noise to reference mic
  output(n) = primary(n) - refP(n);
  w(n+1,:) = w(n,:) + mu.*frefblock'.*output(n);
%we are using the output as our error signal
end;
work = flops-start
w(length(w),:)
%*Plot of w vs. time
figure;hold on
for ii = 1:worder;
 (rv,'blue');
  plot(w(:,ii),'r');
end;
figure;
subplot(3,1,1);
plot(primary);axis([0 length(primary)
min(primary) max(primary)]);
title('primary microphone signal');
subplot(3,1,2);
plot(output);axis([0 length(primary)
min(signal)-.5 max(signal)+.5]);
title('filtered output');
subplot(3,1,3);
```

```
plot((ref(1:length(refP))-refP).^2);%axis([0
length(primary) min(primary) max(primary)]);
%We start calculating the noise at 4 since the
early values of the output
%can be VERY large, and bias our SNR
measurement.
sv = 2*worder;
sw = length(signal);
SNRpre
=norm(signal(sv:sw))/norm(ref(sv:sw));
SNRpre = 10*log10(SNRpre)
```

6. CONCLUSIONS

In this paper, we successfully realized the adaptive inverse system identification in MATLAB. The results show that LMS is an effective algorithm used for the adaptive filter in the inverse system identification to compensate copper transmission. Following Conclusions are founds:

- Estimation of signal to have better approximations.
- Weight coefficient optimization of FIR filter.
- Updating weights from sample-to-sample.
- Inverse system identification is done producing an optimum estimate of the noise from contaminated signal and hence an optimum estimate of desired signal.

7. RECOMMENDATIONS

- Complex LMS algorithm:
  deals with complex data
- Fast LMS algorithm
  Data processing in blocks instead of sample to sample processing
- DSP
  The MATLAB code can be adjusted for DSP processing

8. REFERENCES

8.1 Journals

[1] Bernard Widrow, Robert C. Goodling et al., " Adaptive Noise Canceling: Principles and Applications", Proceedings of the IEEE, vol. 63, pp. 1692-1716, Dec. 1975

[2] S. Ikeda, A. Sugiyama, "An adaptive noise canceller with low signal distortion for speech codes," IEEE Trans. Signal Processing, vol. 47,pp. 665-674, Mar. 1999

8.2 Books

[1] Simon Haykin, Adaptive Filter Theory, Prentice Hall, 1996

[2] B. Widrow, S. Stearns, Adaptive Signal Processing. Englewood Cliffs, NJ: Prentice-Hall,1985

[3] G. Goodwin, K. Sin, Adaptive Filtering, Prediction and Control. Englewood Cliffs, NJ: Prentice-Hall, 1985.

8.3 Website

[1] www.mathworks.com

[2]http://www.spd.eee.strath.ac.uk/~interact/AF/aftutorial