International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 5, Number 2 (2012), pp. 113-124

http://www.irphouse.com

Interference Cancellation in Adaptive Filtering through LMS Algorithm using TMS320C6713DSK

Prabira Kumar Sethy and Dr. Subrata Bhattacharya

Associate Professor, ISM, Dhanbad, Jharkhand, India
E-mail: prabirsethy.05@gmail.com

Abstract

The scope of this paper is interference cancellation which is concerned with removal of noise superposed on speech signal. Interference cancelling makes use of an auxiliary or reference input derived from one or more sensor located in noise field where the signal is undetectable. This input is filtered from primary input containing both signal and interference. Adaptive filtering which are able to lock- in on the frequency of interference and to tracks its changes is required. In order to achieve this, a reference signal should be available which is strongly correlated with the interference only. To this purpose LMS algorithm implementation is considered.

Simulation
When the desired signal is sine_sound.wav:

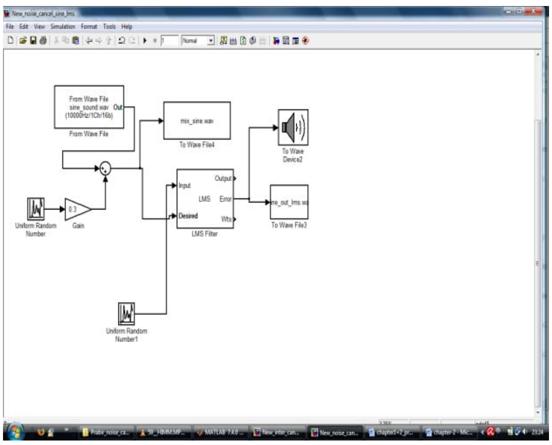


Figure 1: Simulink *Model-1*

At first I generate “sine_sound” using following MATLAB instruction.

```
n=0:4999;
s=10*sin(0.4*pi*n);
sound(s,10000);
```

Now this sound is stored in MATLAB workspace.

Here the “sine_sound.wav” is the desired signal corrupted by the noise ie; uniform random number. The corrupted signal is fed to the desired input of LMS. Another reference noise ie; uniform random number is fed to the input of LMS. At the error output of LMS the “sine_sound.wav” present which is not contaminated by noise.

Observation

Table 1

Uniform Random No. (Noise) (Minimum & Maximum Value)	Perceptual Grading 10
±*0.5*	8
±*1*	9
±*2*	10
±*3*	5

Table 2

Uniform Random No. (Reference Noise) (Minimum & Maximum Value)	Perceptual Grading 10
±0.4	7
±0.6	10
±0.8	8

Table 3

Filter Length	Perceptual Grading(10)
10	03
15	04
18	06
22	08
32	09
35	10
38	09
45	08
50	05
58	03
64	01

Table 4

Step Size	Perceptual Grading(10)
0.001	4
0.005	9
0.1	1

When the desired signal is speech (theforce.wav):

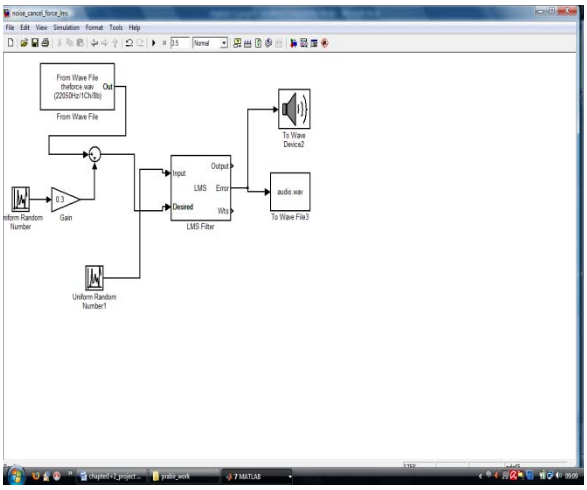


Figure 2: Simulink *Model-2*

Here the "theforce.wav" is the desired signal corrupted by the noise ie; uniform random number. The corrupted signal is fed to the desired input of LMS. Another reference noise ie; uniform random number is fed to the input of LMS. At the error output of LMS the "theforce.wav" presents this is not contaminated by noise.

Observation

Table 5

Uniform Random No. (Noise) (Minimum & Maximum Value)	Perceptual Grading 10
±0.5	8
±1	9
±2	10
±3	5

Table 6

Uniform Random No. (Reference Noise) (Minimum & Maximum Value)	Perceptual Grading 10
±0.4	7
±0.6	10
±0.5	8

Table 7

Filter Length	Perceptual Grading(10)
10	03
15	04
18	06
22	08
32	09
35	10
38	09
45	08
50	05
58	03
64	01

Table 8

Step Size	Perceptual Grading(10)
0.001	4
0.005	9
0.1	1

When the music signal is desired i.e. (hare Krishna.wav):

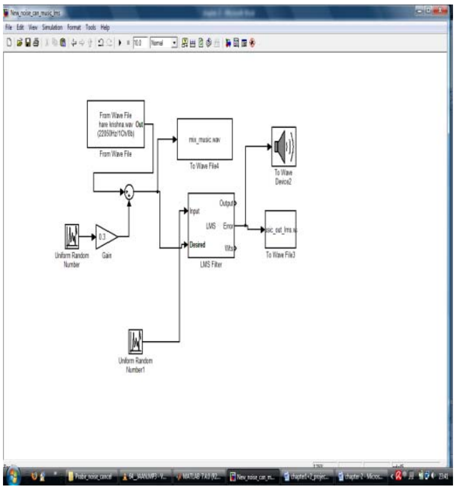


Figure 3: Simulink *Model-3*

Here the “hare Krishna.wav” is the desired signal corrupted by the noise ie; uniform random number. The corrupted signal is fed to the desired input of LMS. Another reference noise ie; uniform random number is fed to the input of LMS. At the error output of LMS the “hare Krishna.wav” present which is not contaminated by noise.

Observation

Table 9

Uniform Random No. (Noise) (Minimum & Maximum Value)	Perceptual Grading 10
±0.5	8
±1	9
±2	10
±3	5

Table 10

Uniform Random No. (Reference Noise) (Minimum & Maximum Value)	Perceptual Grading 10
±0.4	7
±0.6	9
±0.8	8

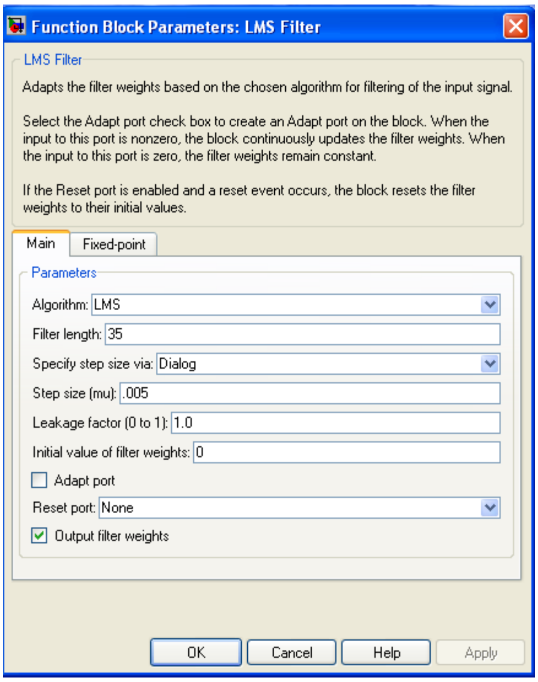
Table 11

Filter Length	Perceptual Grading(10)
10	03
15	04
18	06
22	08
32	09
35	10
38	09
45	08
50	05
58	03
64	01

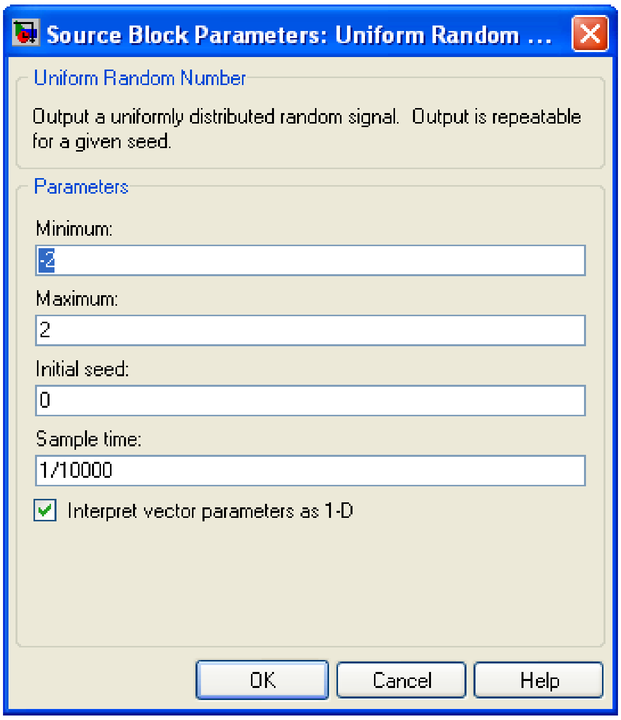
Table 12

Step Size	Perceptual Grading(10)
0.001	4
0.005	9
0.1	1

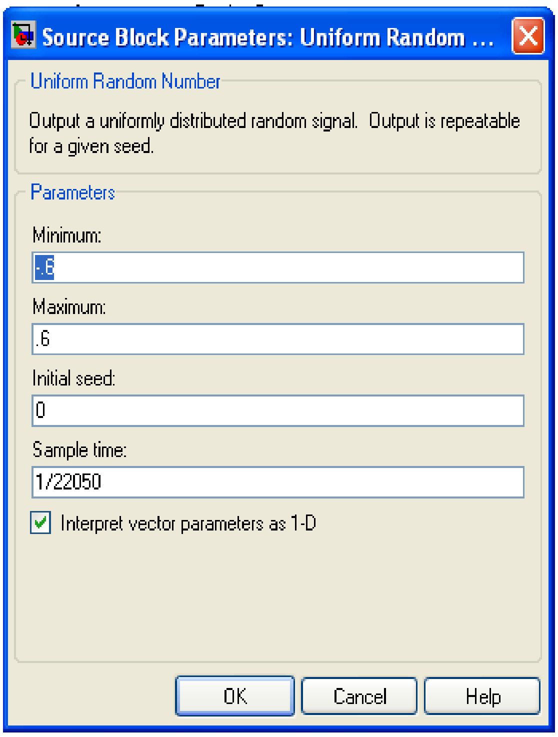
Optimum Parameters of LMS



Source Block Parameter of Additive Noise



Source Block Parameter of Reference Noise



Real-Time Implementation of Noise Cancellation when Desired Signal is Square Wave from Function Generator using LMS

At first I generate square wave from function generator of frequency 1KHz and amplitude 1Vpp. Now that square wave is added by noise signal by using below real-time block.

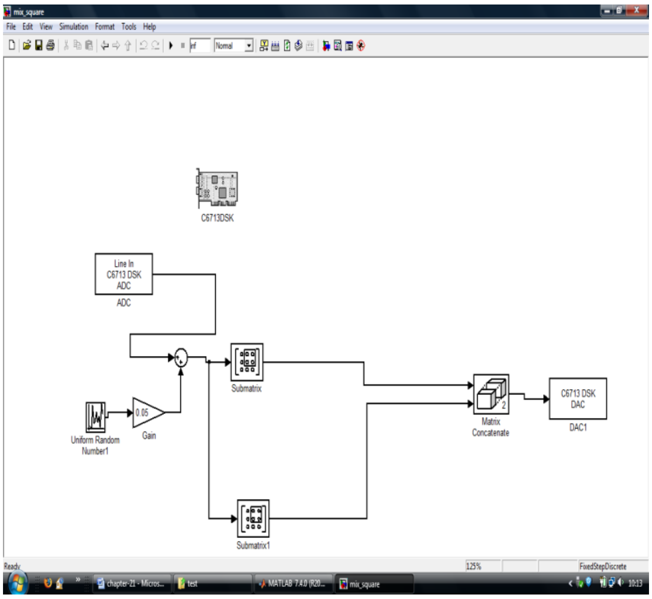


Figure 4: *Real-Time Model-1*

Square Wave and its Spectrum

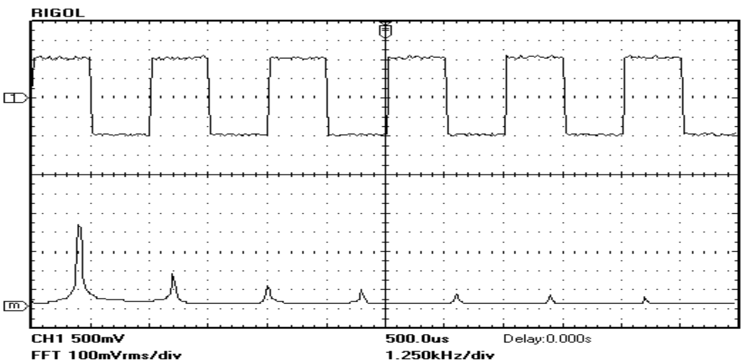


Figure 2.22

It is the square wave of frequency 1KHz and amplitude 1Vpp. Its FFT peak appears at 1KHz, 3KHz,5KHz, 7KHz which are the odd harmonics.

Output Wave Form (Square Wave+Noise) and its Spectrum

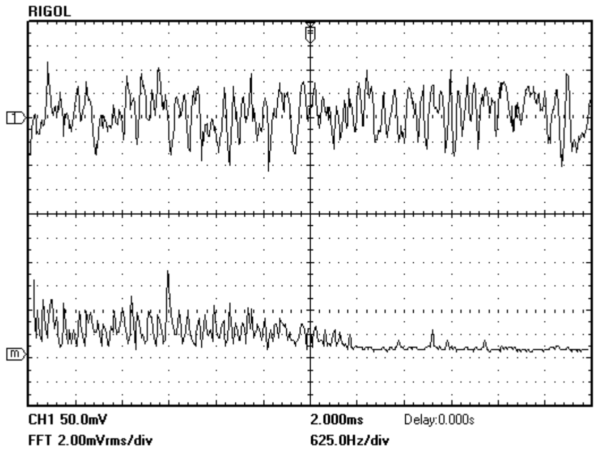


Figure 2.23

From the above spectral analysis it observed that one FFT peak appears at 1KHz which is the FFT of square wave. Now the noise which corrupts the square wave is cancelled by below real-time block.

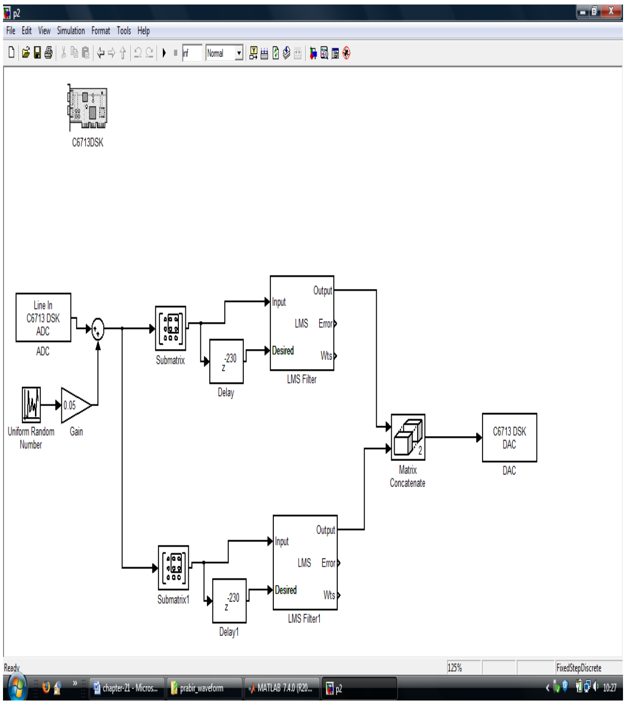


Figure 5: *Real-Time Model-2*

Output Wave Form after Noise Cancellation and its Spectrum

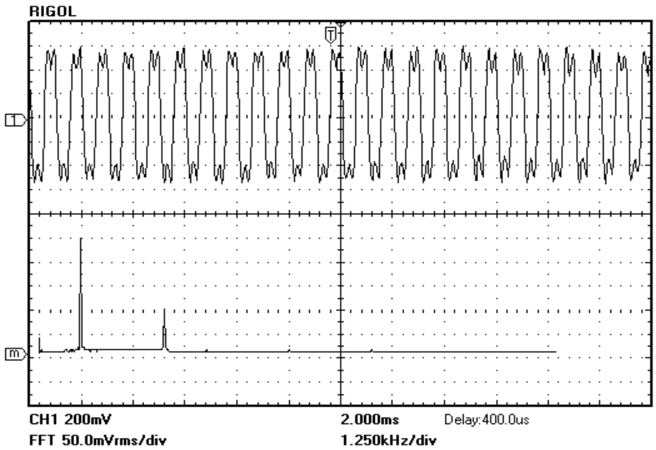


Figure 2.25

From this spectral analysis I observed that the FFT peak appears at 1KHZ and 3KHz which are dissolving for square wave. Here are almost no FFT peaks for noise which clear that noise is totally filtered out.

Real-Time Implementation of Noise Cancellation when Desired Signal is Speech Sound:

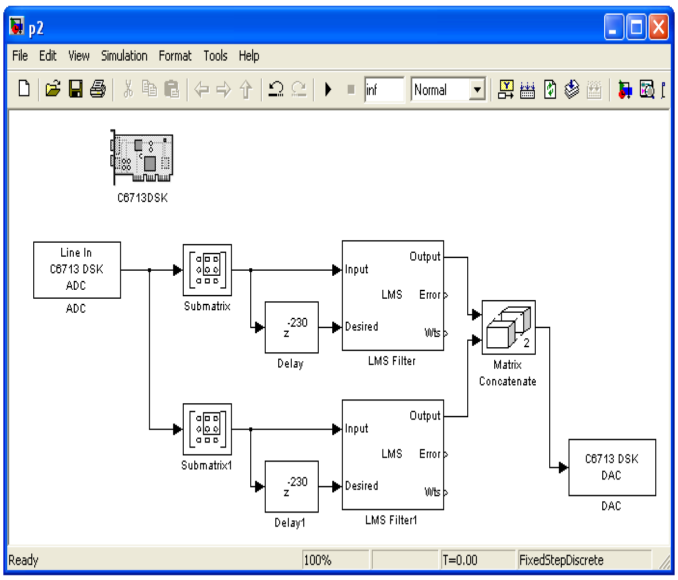


Figure 2.28*: Real-Time Model-8*

Result and Discussion

Noise used for corrupting sine_sound.wav and theforce.wav

Uniform random number:
maximum value=+2
Minimum value=-2
The noise is weighted by gain=3
So, the variance of noise actually added=$(0.3)^2 * [(2)^2/3]$=0.12

Noise used for reference

Uniform random number: maximum value=+0.6
Minimum value=-0.6
So, the variance of reference noise = $(0.6)^2 / 3$ =0.12
Now, the interfering signal is easily filtered out.

When reference noise is random (Gaussian noise) with same variance ie; 0.12 the interfering signal is not totally removed. Some portion of interfering signal appears at the out put. From this it clear that the interfering signal is filtered out when the additive noise and reference noise characteristics are same.

References

[1] Vinay K. Ingel, Digital Signal Processing using MATLAB.
[2] Bernard Widrow, Adaptive Signal Processing.
[3] Simon Haykin, Adaptive Filter Theory.
[4] Rulph Chassaing, Digital signal processing and applications with the C6713 and C6416 DSK.
[5] R. Chassaing, DSP Applications Using C and the TMS320C6x DSK, Wiley, New York, 2002
[6] TMS320C6713 DSK user manual, Cranes Software international limited (TI-Solutions).
[7] M.H. Hayes and Subrata Bhattacharya, Digital Signal Processing, Tata McGraw-Hill Publishing Company Limited, 3rd Ed., 2004.
[8] HWEI HSU and Dr. D. Mitra, Analog and Digital Communications, Tata McGraw-Hill Publishing Company Limited, 3rd Ed., 2004.
[9] N. Kehtarnavaz and M. Keramat, DSP System Design Using the TMS320C6000, Prentice Hall, Upper Saddle River, NJ, 2001. Kluwer Academic, New York, 2003.
[10] Code Composer Studio Getting Started Guide, SPRU509,Texas Instruments, Dallas,TX, 2001.
[11] TMS320C6000 Code Composer Studio Tutorial, SPRU301C, Texas Instruments, Dallas, TX, 2000.
[12] R. Chassaing, Digital Signal Processing Laboratory Experiments Using C and the TMS320C31 DSK,Wiley, New York, 1999.

[13] Carlos R. Martins, An Adaptive Noise Canceller and Its Implementation On DSP.
[14] Bernard Widrow, Adaptive Noise Cancelling Principles and Applications.
[15] Harrison, Adaptive Noise Cancellation.