IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 7, Issue 2, Ver. I (Mar. - Apr. 2017), PP 20-25
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Real Time System Identification of Speech Signal Using Tms320c6713

Rathnakara.S[1] and Dr.V.Udayashankara[2]
[1]*Research Student, JSS Research Foundation, Mysore, India*
[2] *Research Guide, JSS Research Foundation, Mysore, India*

Abstract : *In this paper real time system identification is implemented on TMS320C6713 for speech signal using standard IEEE sentence (SP23) of NOIZEUS database with different types of real world noises at different level SNR taken from AURORA data base. The performance is measured in terms of signal to noise ratio (SNR) improvement. The implementation is done with "C' program for least mean square (LMS) and recursive least square (RLS) algorithms and processed using DSP processor with Code composer studio.*
Keywords: *System Identification, LMS, RLS, SNR, TMS3206713*

I. Introduction

The quality and intelligibility of speech reduced at the receiving end when the input speech is mixed with some form of the noise components. So the original speech becomes unpleasant to listen. The amount of unpleasant is depending upon the type of noise corrupted to the speech signal. The quality and intelligibility of the speech in the presence of background noise can be improved by speech enhancement algorithms. Over many decades researchers are focused in this area and developed the different algorithms to remove the noise which is present along with the speech signal. [7-10].The adaptive filter is still better tool to suppress the noise content in the speech signal. Recently many researchers are working on implementing system identification and noise canceller using adaptive filters on fixed point processor, floating point processor, simulink and other hardware solutions [4,11] for deterministic signals and time varying signals. In this work LMS and RLS algorithms are implemented on floating point processor TMS320C6713 for system identification and performance is compared in terms of output SNR

II. LMS Algorithm

The least mean square (LMS) algorithm is basic adaptive filter used for system identification and noise cancellation. The basic block diagram of adaptive transversal filter is shown in Figure 1.In which input signal $u(n)$ and a desired signal $d(n)$ determine the filter coefficient 'w', that minimizes the error $e(n)$, between the output of the filter $y(n)$, and the desired signal d(n).In LMS algorithm the step size remains constant in updating filter coefficient equation [1, 2]

The output, error and weight equation of conventional LMS algorithms are shown below.

The output; $y(n)=w(n).u^T(n)$ (1)

The error; $e(n)=d(n)-y(n)$ (2)

The weight equation:

$w(n+1) = w(n) + 2\mu u(n) e(n)$ (3)

Where μ is the step size..

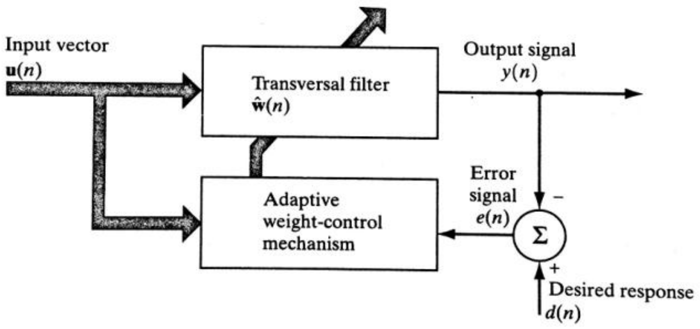


Figure 1. Adaptive transversal filter

III. RLS Algorithm

The recursive least square (RLS) adaptive algorithm is based on least square estimation of the filter coefficients for (n-1) iterations. This algorithm is very powerful, when input is very dynamic and requires very high speed of response. To reduce the number of iterations RLS algorithm uses the autocorrelation matrix. Autocorrelation matrix in RLS algorithm uses current tap-input information as well as all past input samples. Important feature of the RLS algorithm is the inverse correlation matrix (similar to 'mu' in Least Mean Square Algorithm) replaced at each step by a simple scalar division. The RLS exhibits extremely fast convergence compared to least mean square algorithms but computational complexity is very high.

If u(1),u(2)… are input samples, d(1),d(2)… are desired samples and "Λ 'is forgetting factor varies between 0 and 1.The forgetting factor gives exponentially less weight to older error samples and gives more weightage to recent data. 'δ' is small positive constant The error e(n) is calculated by taking the difference of desired response d(n)and the output y(n)The output, error and weight equation of conventional RLS algorithms are shown below

Initialization

$$P(0) = \delta^{-1} I \tag{4}$$

$$w(0)=0 \tag{5}$$

The filter output is $y(n) = u(n)^T w(n-1)$ (6)

$$k(n) = \frac{\lambda^{-1} P(n-1) u(n)}{1 + \lambda^{-1} u^T(n) P(n-1) u(n)} \tag{7}$$

Inverse correlation matrix

$$P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} k(n) u^T(n) P(n-1) \tag{8}$$

$$e(n) = d(n) - u(n)^T w(n-1) \tag{9}$$

$$w(n) = w(n-1) + k(n) e(n) \tag{10}$$

IV. Tms320c6713 DSP Processor

The DSK is a low-cost standalone development platform also serves as a hardware reference design for the TMS320C6713 DSP [3-5]. The DSK uses the 32-bit EMIF for the SDRAM (CE0) and daughter card expansion interface (CE2 and CE3).An on-board AIC23 codec allows the DSP to transmit and receive analog signals. Analog audio I/O such as microphone input, line input, line output and headphone output is done through four 3.5mm audio jacks the codec can select the microphone or the line input as the active input. The analog output is driven to both the fixed gain line connector and adjustable gain headphone connectors. McBSP0 is used for the codec control interface and McBSP1 is used for data. The block diagram of C6713 DSK is shown in figure 2. Code composer studio provides an integrated development environment (IDE) to incorporate software tools, which includes tools for code generation such as a C compiler followed by an assembler and a finally linker. The C compiler compiles a C source program with extension .c to produce an assembly source file with extensioin.asm. The assembler assembles an .asm source file to produce machine language object file with extension obj. the linker combines object files and object libraries as input to produce an executable file with extension .out. This executable file can be loaded and run directly on the C6713 processor.

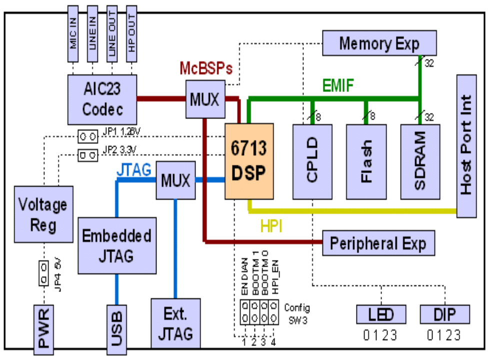


Figure 2 Block diagram of C6713 DSK.

V. Simulated Results

The simulations are carried with standard IEEE sentence sp23 (NOIZEUS database) of male voice saying" stop whistling and watch the boys march". The original signal has 21209 samples. The performance of the LMS and RLS are tested with babble noise, restaurant noise, airport noise and train noise at different level of SNR taken from AURORA data base. The results are plotted using Matlab. The different stages of implementation are illustrated below. The trouble shooting DSK connectivity is shown in Figure 3, which is initial stage for configuration of CCS to configure the ports. Figure 4 shows the window corresponding to create a new project. Figure 5 shows the successful code building stage of the project. Figure 6 shows final execution stage 'run'. The output file will be stored in out.dat The Figure 7 shows the desired signal, signal+noise, output and the mean square error for LMS with original speech corrupted with babble noise of' 0'dB for 'µ' =0.1 with filter order 8.Figure 8 shows the same for RLS algorithm with forgetting factor λ=0.97 and δ=0.01.The performance of RLS algorithm has shown in Table 1 for different values of forgetting factor. It is clearly evident that the output SNR observed maximum when Λ=0.97 for different input SNR. Table 2 demonstrates the performance of output SNR for both LMS and RLS algorithms. The RLS gives better output SNR for all types of nonstationary noises at different level of input SNR compares to LMS algorithm

5.1Mean Square Error

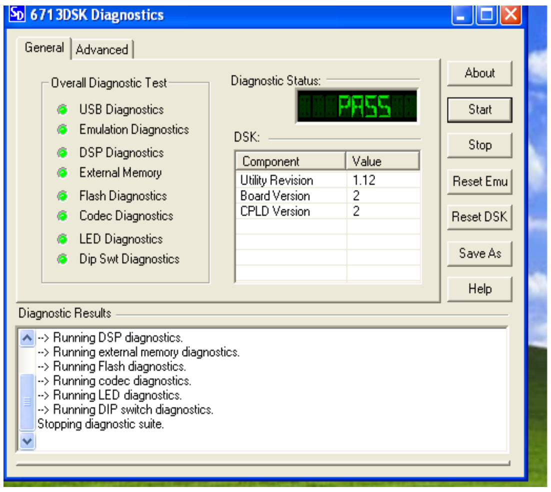
Mean square error (MSE) is important parameter to estimate the performance of the algorithms. The corresponding equation is given below.

$$\mathrm{MSE}=\frac{1}{\mathrm{N}}\sum_{\mathrm{n}=1}^{\mathrm{N}}(\mathrm{d}[\mathrm{k}]-\mathrm{y}[\mathrm{k}])^2 \qquad (11)$$

Where d(n) is the desired signal, y(n) is the output of the filter and N is the total number of samples. The MSE defiens the mean square error between the real model to estimated model[11]. MSE in decibels given by

$$\mathrm{MSE\ \ dB}=10\log_{10}(\mathrm{MSE}) \qquad (12)$$

Figure 9 demonstrates the variations of mean square error (MSE) for both the algorithms in dB. The average MSE obtained for LMS and RLS is 28.73 dB & -52.9 dB – respectively


6713DSK Diagnostics

General | Advanced

Overall Diagnostic Test
- USB Diagnostics
- Emulation Diagnostics
- DSP Diagnostics
- External Memory
- Flash Diagnostics
- Codec Diagnostics
- LED Diagnostics
- Dip Swt Diagnostics

Diagnostic Status: PASS

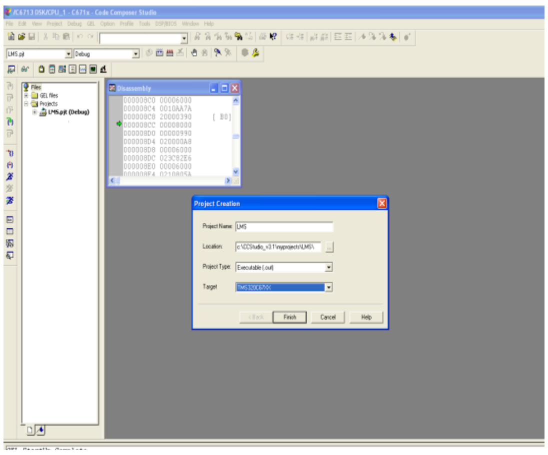
DSK:

Component	Value
Utility Revision	1.12
Board Version	2
CPLD Version	2

About | Start | Stop | Reset Emu | Reset DSK | Save As | Help

Diagnostic Results
```
--> Running DSP diagnostics.
--> Running external memory diagnostics.
--> Running Flash diagnostics.
--> Running codec diagnostics.
--> Running LED diagnostics.
--> Running DIP switch diagnostics.
Stopping diagnostic suite.
```

Figure3. Trouble shooting DSK connectivity


Project Creation

Project Name: LMS

Location: c:\CCStudio_v3.1\myprojects\LMS\

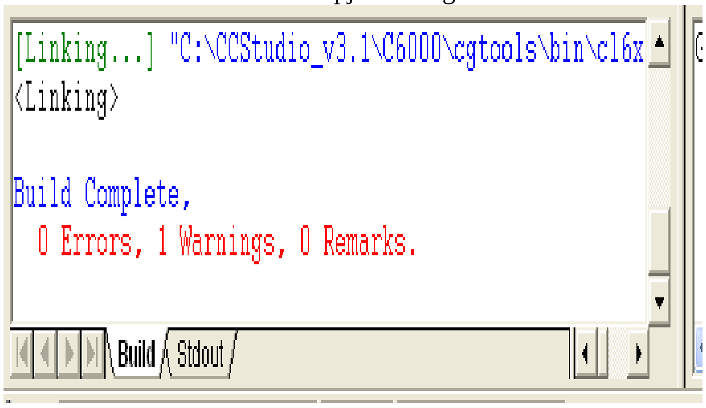
Project Type: Executable (.out)

Target: TMS320C67XX

< Back | Finish | Cancel | Help

Figure 4. To create a new project file

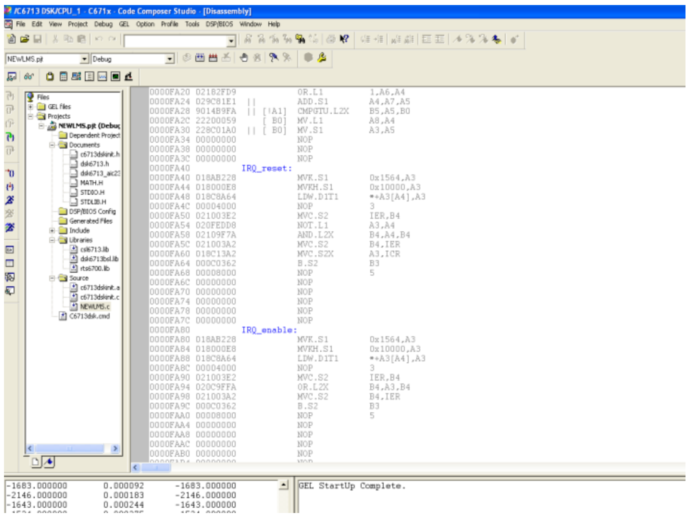
------------------------------ LMS.pjt - Debug ---------------------------



```
[Linking...] "C:\CCStudio_v3.1\C6000\cgtools\bin\cl6x
<Linking>

Build Complete,
  0 Errors, 1 Warnings, 0 Remarks.
```

Figure5. Code building stage window.



```
0000FA20 02182FD9                OR.L1       1,A6,A4
0000FA24 029C81E1  ||            ADD.S1      A4,A7,A5
0000FA28 9014B9FA  || [!A1]      CMPGTU.L2X  B5,A5,B0
0000FA2C 22200059     [ B0]      MV.L1       A8,A4
0000FA30 228C01A0  || [ B0]      MV.S1       A3,A5
0000FA34 00000000                NOP
0000FA38 00000000                NOP
0000FA3C 00000000                NOP
0000FA40           IRQ_reset:
0000FA40 018AB228                MVK.S1      0x1564,A3
0000FA44 018000E8                MVKH.S1     0x10000,A3
0000FA48 018C8A64                LDW.D1T1    *+A3[A4],A3
0000FA4C 00004000                NOP         3
0000FA50 021003E2                MVC.S2      IER,B4
0000FA54 020FEDD8                NOT.L1      A3,A4
0000FA58 02109F7A                AND.L2X     B4,A4,B4
0000FA5C 021003A2                MVC.S2      B4,IER
0000FA60 018C13A2                MVC.S2X     A3,ICR
0000FA64 000C0362                B.S2        B3
0000FA68 00008000                NOP         5
0000FA6C 00000000                NOP
0000FA70 00000000                NOP
0000FA74 00000000                NOP
0000FA78 00000000                NOP
0000FA7C 00000000                NOP
0000FA80           IRQ_enable:
0000FA80 018AB228                MVK.S1      0x1564,A3
0000FA84 018000E8                MVKH.S1     0x10000,A3
0000FA88 018C8A64                LDW.D1T1    *+A3[A4],A3
0000FA8C 00004000                NOP         3
0000FA90 021003E2                MVC.S2      IER,B4
0000FA94 020C9FFA                OR.L2X      B4,A3,B4
0000FA98 021003A2                MVC.S2      B4,IER
0000FA9C 000C0362                B.S2        B3
0000FAA0 00008000                NOP         5
0000FAA4 00000000                NOP
0000FAA8 00000000                NOP
0000FAAC 00000000                NOP
0000FAB0 00000000                NOP
```

```
-1683.000000      0.000092      -1683.000000
-2146.000000      0.000183      -2146.000000
-1643.000000      0.000244      -1643.000000
```

```
GEL StartUp Complete.
```

Figure.6. Final execution stage

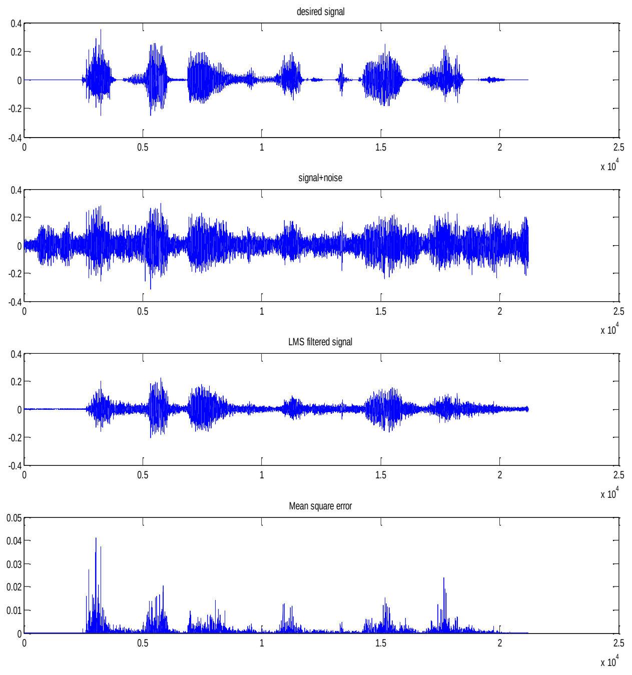


Figure 7 The desired signal, signal+noise, output and the mean square error for LMS with '0' dB Babble noise

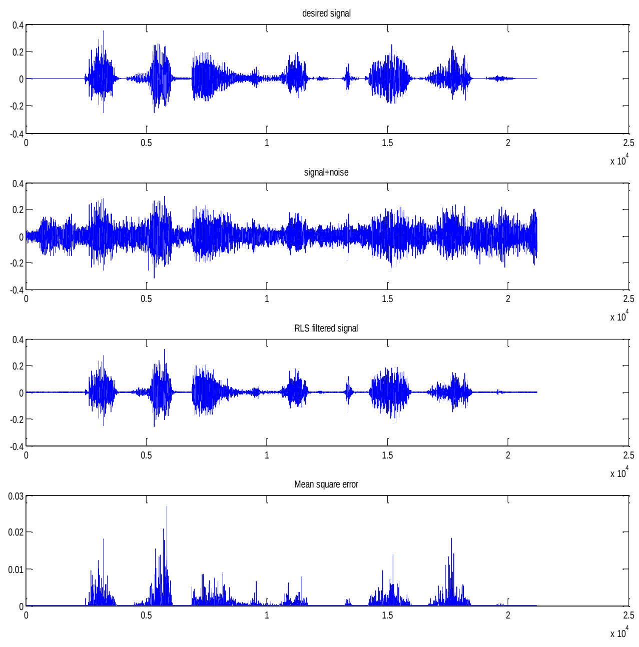


Figure 8 The desired signal, signal+noise, output and the mean square error for RLS with '0' dB Babble noise

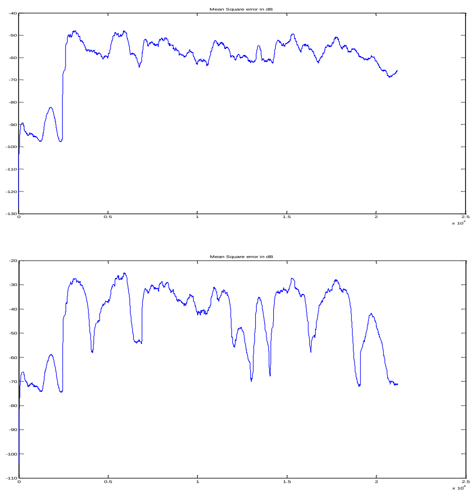


Figure 9. Top to bottom, the excessive mean square error of LMS and excessive mean square error of RLS algorithm in dB.

Table1: Performance RLS algorithm for different Values of forgetting factor (Λ)

Type of Noise	**File Name**	**Input SNR in dB**	**Output SNR for different Forgetting Factor(Λ)**			
			0.96	**0.97**	**0.98**	**0.99**
Babble Noise	sp23_babble_sn0	0	6.343	6.385	6.384	6.209
	sp23_babble_sn5	5	9.402	9.414	9.37	9.101
	sp23_babble_sn10	10	13.507	13.546	13.525	13.259
	sp23_babble_sn15	15	16.687	16.886	16.811	16.457

VI. Conclusion

In this work LMS and RLS algorithm were successfully implemented and analyzed for system identification on floating point processor TMS320C6713. The results are analyzed with standard speech IEEE sentence (SP23) of NOIZEUS database with different types of real world noises at different level SNR.. The performance is measured in terms of SNR improvement. RLS algorithm gives better SNR improvement and reduces mean square error due to estimated weight of the unknown system matching with actual weight of the desired signal. So this can be used in digital hearing aid for real time noise cancellation

Table2: Performance of the LMS and RLS algorithm for different types of noises at different input SNR

Type of Noise	**input File name**	**Input SNR in dB**	**Output SNR in dB**	
			LMS	**RLS**
Babble Noise	sp23_babble_sn0	0	4.64	6.39
	sp23_babble_sn5	5	6.34	9.41
	sp23_babble_sn10	10	8.72	13.55
	sp23_babble_sn15	15	10.68	16.89
Restaurant Noise	sp23_resturant_sn0	0	4.18	5.76
	sp23_resturant_sn5	5	6.22	8.7
	sp23_resturant_sn10	10	6.59	14.7
	sp23_resturant_sn15	15	10.85	17.38
Airport Noise	sp23_airport_sn0	0	3.96	5.21
	sp23_airport_sn5	5	6.52	9.07
	sp23_airport_sn10	10	8.65	12.64
	sp23_airport_sn15	15	10.66	17.19
Train Noise	sp23_train_sn0	0	4.9	6.68
	sp23_train_sn5	5	7.34	10.77
	sp23_train_sn10	10	9.19	13.62
	sp23_train_sn15	15	11.05	18.4
Exhibition Noise	sp23_exhbition_sn0	0	4.64	6.19
	sp23_exhbition_sn5	5	6.69	9.47
	sp23_exhbition_sn10	10	8.92	13.18
	sp23_exhbition_sn15	15	10.82	17.52

References

[1]. Simon Haykin " Adaptive Filter Theory" Fourth edition pearson education

[2]. V.Udayashankara " Modern digital signal processing" second edition PHI

[3]. Gaurav Saxena, SubramaniamGanesan, and Manohar Das "Real time implementation of adaptive noise cancellation" 978-1-4244-2030-8/08,2008 IEEE. PP431-436

[4]. Texas Instruments Tutorial, TMS320C6713 Hardware Designers Resource Guide",(July 2004), SPRAA33.

[5]. D. Reay, "Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK," John Wiley and Sons, Inc,Edition- 2nd 2008.

[6]. J. M. Górriz, et al "A Novel LMS Algorithm Applied to Adaptive Noise Cancellation "IEEE 2008 pp34-37

[7]. Julie E. Greenberg "Modified Lms Algorithms For Speech Processing With An Adaptive Noise Canceller", Ieee Transactions On Speech And Audio processing Vol6.No4 1998 PP338-351

[8]. Ma Shengqian, XuGuowei, Ma Zhifeng, Wei Shuping, Fan Manhong "Research on adaptive noise canceller of an improvement LMS algorithm 2011 IEEE pp1611-1614

[9]. Mohamed DjendiEt.Al "A New DualForwardBss Based Rls (Dfrls) AlgorithmFor Speech Enhancement." IEEE 2016

[10]. Ted S. And Biing-Hwang JuangWada "Enhancement Of Residual Echo For Robust Acoustic Echo Cancellation", IEEE Transactions On Audio, Speech And Language Processing, Vol. 20, No. 1, 2012 175-189

[11]. Fabián Rolando Jiménez- López and et.al" Adaptive filtering implanted over TMS 320C6713"ITECKNE Vol. 11, 2014 pp157-171