
**Proceedings of the First**

# University Journal of Research and Innovation

**December, 2019**

Organized by
University of Computer Studies (Pakokku)

# Proceeding of

# The First University Journal of Research and Innovation 2019

**December , 2019**

**Organized by**

**University of Computer Studies (Pakokku)**

**Department of Higher Education ,**

**Ministry of Education , Myanmar**

# University Journal of Research and Innovation

Volume 1, Issue 1 2019

# University Journal of Research and Innovation 2019

Volume 1 , Issue 1 , 2019

This journal and individual papers published at www.ucspkku.edu.mm.



**Copies:110**

All research papers in this journal have undergone rigorous peer-reviewed which is published annually. Full papers submitted for publication are refereed by the Associate Editorial Board through an anonymous referee process.

The authors of the paper bear the responsibility for their content.

Papers presented at the First University Journal of Research and Innovation(UJRI), University of Computer Studies (Pakokku), December 2019.

# UJRI 2019 Editorial Board

# UJRI 2019 Editorial Board

# Proceedings of

# The First University Journal of

# Information and Computing Science 2019

# December, 2019

## Contents

### Artificial Intelligence & Machine Learning



### Natural Language Processing



### Big Data Analysis

## Parallel & Distributed Computing



## Image Processing



## Human Computer Interaction

## Database Management System & Information Retrieval



## Network & Security

## Data Mining & Machine Learning



## Digital Business Management

## Electronics



## Embedded System



## Cloud Computing

## Software Engineering and Web Engineering



## Digital Signal Processing



## Theoretical Nuclear Physics

## Material Science



## Mathematics

## Experimental Nuclear Physics



## English Language



## Myanmar Language in Literature

# Author Index

# Analysis of Noise Cancellation using LMS and RLS Algorithms

Aye Theingi Oo[#1], Theingi Ait[*2], Nay Win Zaw[#3]
[#]*Department of Electronic Engineering,West Yangon Technological University*
*Yangon, Myanmar*
[1]*ayetheingioo1989@gmail.com*
[2]*theingiait@gmail.com*
[3]*drnaywinzaw@gmail.com*

**Abstract**

*In the field of Digital Communication & signal processing, signals are associated with noise and distortions. This is due to the fact that system undergoes the time varying physical process which are unknown some time. These variations can be eliminated using adaptive filtering as it senses the unknown variation in the signal properties. Finite impulse response (FIR) and Infinite impulse response (IIR) filters are available for adaptation process. Amongst both of them FIR is widely used by adaptive filter. The approaches of adaptation can be achieved by least mean square (LMS), wiener filter, recursive least squares filter (RLS) etc. Noise cancellation can be achieved by using proper value of the parameters. In this paper the performance of LMS filter and RLS filter are used for noise cancellation in voice signal. The step size (μ) is considered to be the parameter where we can make the changes and a comparison has been drawn based on their performance.*

**Keywords -** Adaptive Filtering Algorithms, LMS, RLS, Acoustic Echo Cancellation, ERLE, MSE, SNR

## 1. Introduction

Digital filters are a very important part of DSP. In fact, their extraordinary performance is one of the key reasons that DSP has become so popular. In the introduction, filters have two uses: signal separation and signal restoration. Signal separation is needed when a signal has been contaminated with interference, noise, or other signals. Signal restoration is used when a signal has been distorted in some way. Analog filters are cheap, fast, and have a large dynamic range in both amplitude and frequency. Digital filters, in comparison, are vastly superior in the level of performance that can be achieved. Digital filters can be implemented in two ways, by convolution (also called finite impulse response or FIR) and by recursion (also called infinite impulse response or IIR). Filters carried out by convolution can have far better performance than filters using recursion, but execute much more slowly [1].

Noise has proven to be the bottleneck in deciding the performance of communication system and its random nature makes it difficult in designing these systems. The two types of filtering are used fixed and adaptive. The difference being, adaptive filters do not require prior information and hence are more effective. Adaptive filters are used in many diverse applications such as echo cancellation, radar signal processing, navigation systems, and equalization of communication channels and in biomedical signal enhancement. The two efficient algorithms for designing of adaptive filters are RLS and LMS algorithm [9].

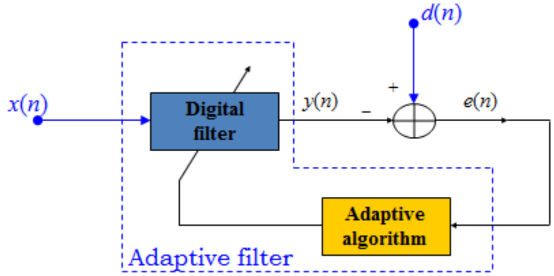



**Figure 1. General Adaptive Filter Algorithm**

## 2. Adaptive Filter

An adaptive filter adapts to changes in its input signals automatically according to a given algorithm. The algorithm will vary the coefficients according to a given criteria, typically an error signal to improve its performance. Basically an adaptive filter is a digital filter combined with an adaptive algorithm, which is used to modify the coefficients of the filter [5].

Adaptive filters are used in many diverse applications in today's world for example telephone echo canceling, radar signal processing, equalization of communication channels and biomedical signal enhancement [9].

Input Signal: For the case of multiple input [8]

$$X_n=[X_{0n}X_{1n}\ldots\ldots\ldots\ldots X_{ln}]^T \qquad (1)$$

Weight Vectors: Corresponding to weight vector input signal are $W_n=[W_{0n}W_{1n}\ldots\ldots W_{ln}]^T$ (2)

So, on multiplying eqution (1) and (2) , we get output signal as $y_n=X_n^T W_n=W_n X^T_n$ (3)

and since error signal is: $e_n=d_n–y_n$ (4)

Calculating error signal from equation (3) :

$$e_n=d_n - X_n^T W=d_n–WX^T_n \qquad (5)$$

On squaring equation (5),the instantaneous squared error is: $e^2_n=d^2_n+W^T X_n X_n^T W–2d_n X_n^T W$ (6)

from equation (6) calculating the expected mean square value as mentioned in equation (7)

$$E[e^2_n]=E[d^2_n]+W^T E[X_n X^T_n]W– 2E[d_n X_n^T]W \qquad (7)$$

$X_n X_n^T$ is the autocorrelation matrix of input signal and it is represented by the term R as given in the eqution (8)

$$R=E[X_n X_n^T]$$

$$= E\begin{bmatrix} X^2_{0n} & X_{0n}X_{1n} & \cdots & X_{0n}X_{ln} \\ \vdots & \ddots & & \vdots \\ X_{ln}X_{0n} & X_{ln}X_{1n} & \cdots & X^2_{ln} \end{bmatrix} \qquad (8)$$

Diagonal terms mean square of input components and cross terms are cross correlation among the input components [8]. Next term E [$d_n$ $X_n$] is the cross correlation between the input signal and desired signal and it is expressed by P and represented in equation (9)

$$P=E[d_n X_n]=E[d_n X_{0n} \quad d_n X_{1n} \ldots\ldots d_n X_{ln}]^T \qquad (9)$$

Now calculating mean square error from equation (7)

$$MSE \triangleq \zeta=E[e^2 n]=E[d^2_n] +W^T RW– 2P^T W \qquad (10)$$

Gradient and Mean square error:

$$\nabla \triangleq \partial\zeta/\partial W=[\partial\zeta/\partial w0 \quad \partial\zeta/\partial w1 \quad \ldots \quad \partial\zeta/\partial w_L]^T \qquad (11)$$

$$=2RW–2P \qquad (12)$$

To obtain the minimum mean-square error the weight vector W is set at its optimal value W*, where the gradient is zero [8]:

$$\nabla = 0 = 2RW*2P \qquad (13)$$

W* is wiener weight vector

$$W* = R^{-1} P \qquad (14)$$

Mean square error is obtained by substituting W* from equation (14) for W in equation (10)

$$\zeta min = E[d^2 n] + W^{*T} RW^* - 2P^T W^* \quad (15)$$

$$= E[d^2_n] + [R^{-1}P]^T RR^{-1}P - 2P^T R^{-1}P \quad (16)$$

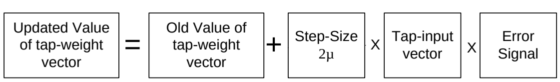
On simplifying equation (16) we get :

$$\zeta min = E[d^2_n] - P^T R^{-1} P = E[d^2_n] - P^T W^* \quad (17)$$

### 2.1. The LMS Algorithm

The properties of Least Mean Square (LMS) algorithm also that makes it the best choice for many real-time systems are simplicity and ease of implementation. The LMS algorithm is introduced by Widrow & Hoff in 1959 [3]. Simple, no matrices calculation involved in the adaptation. The Least Mean Square, or LMS, algorithm is a stochastic gradient algorithm that iterates each tap weight in the filter in the direction of the gradient of the squared amplitude of an error signal with respect to that tap weight. The LMS is an approximation of the steepest descent algorithm, which uses an instantaneous estimate of the gradient vector. It is the more successful of the algorithms because it is the most efficient in terms of storage requirement and indeed computational complexity, the basic LMS algorithm updates the filter coefficients after every sample [3].

The Least-Mean-Square algorithm in words is described below [7]:




The LMS Algorithm consists of two basic processes: Filtering process and adaptation process. The filtering process includes two steps [7] :

(i) Calculate the output of FIR filter by convolving input and taps.

$$y[n] = w^T[n]x[n] \quad (18)$$

(ii) Calculate estimation error by comparing the output to desired signal.

$$e[n] = d[n] - y[n] \quad (19)$$

The adaptation process adjust tap weights based on the estimation error.

$$w[n+1] = w[n] + 2\mu e[n]x[n] \quad (20)$$

The LMS algorithm using Matlab is as follow:

(1)Read voice signal into Matlab:Desired vector d(n), Input vector x(n);

(2)Initialization column weight vector W(n)=0, Error vector e(n)=0

(3)Calculate Filter Output $y\hat{}(n)=w^T(n)x(n)$

(4)If input signal length is greather than filter size

a. Calculate error vector e(n)
b. Update weight vector w[n+1]=w[n] +2 µ e[n] x[n]
c. Go to step (3)

(5) If input signal length is less than filter size

a. Calculate MSE, ERLE and SNR
b. End of the system

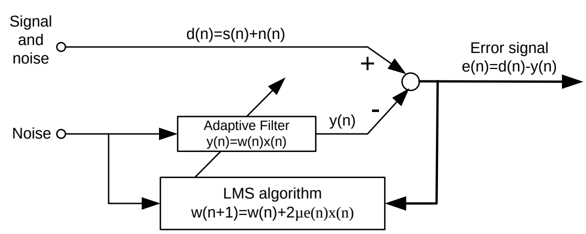
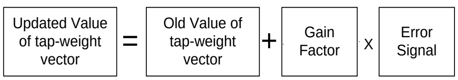



**Figure 2. Architecture of LMS algorithm**

### 2.2. Recursive Least-Squares Algorithm (RLS)

The Recursive Least Squares (RLS) algorithm was introduced in order to provide superior performance compared to those of the Least Mean Squares (LMS) algorithm at the expense of increased computational complexity. In the RLS algorithm, an estimate of the autocorrelation matrix is used to decorrelate the voice signal. Also, the quality of the steady state solution keeps on improving over time, eventually leading to an optimal solution [4]. The RLS algorithm recursively solves the least squares problem. In the following equations, the constants λ and δ are user defined that represent the forgetting factor and regularization parameter respectively. The forgetting factor is a positive constant less than unity, which gives a measure of the memory of the algorithm; and the regularization parameter's value is determined by the signal-to-noise ratio (SNR) of the signals [6].

The Recursive Least-Squares algorithm in words is described below [2]:




The RLS Algorithm consists of two basic processes: Filtering process and adaptation process. The filtering process includes two steps [2]:

(1) Calculate the output of FIR filter by convolving input and taps.

$$y(n)=w^T(n)x(n) \quad (21)$$

(2) Calculate estimation error by comparing the output to desired signal.

$$e[n] = d[n] - y[n] \quad (22)$$

The adaptation process adjust tap weights based on the estimation error and gain factor [2].

$$w(n)=w(n-1)+k(n)e(n) \quad (23)$$

Computing the gain vector:

$$k(n)= \frac{\lambda^{-1}\Phi_A^{-1}(n-1)x(n)}{1+\lambda^{-1}x^T(n)\Phi_A^{-1}(n-1)x(n)} \quad (24)$$

$\Phi_A^{-1}(n)$ *update* is

$$\Phi_A^{-1}(n) =\lambda^{-1}\Phi_A^{-1}(n-1) - \lambda^{-1}k(n)x^T(n)\Phi_A^{-1}(n-1) \quad (25)$$

The adaptive filter algorithm is measured by using performance measure parameters of Echo Return Loss Enhancement (ERLE) and Mean Square Error (MSE) [2].

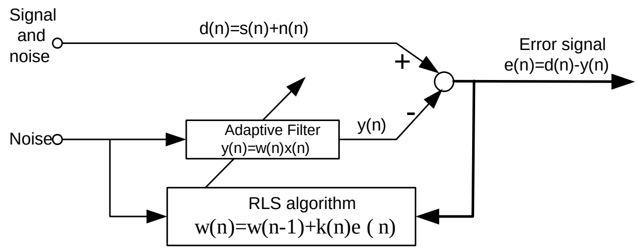



**Figure 3. Architecture of RLS algorithm**

## 3. Signal to Noise Ratio (SNR)

SNR is defined as the ratio of signal power to the noise power corrupting the signal. The Signal to Noise Ratio is the defining factor when it comes to the measurement of quality of signal. A high SNR means good quality of signal with low distortions.

$$SNR=10log_{10}\left[\frac{rms(speech)}{rms(noise)}\right] \quad (26)$$

### 3.1. Echo Return Loss Enhancement (ERLE)

The Echo Return Loss Enhancement (ERLE) is a measure of the amount of echo suppressed by the acoustic echo canceller. It is defined as the ratio of power of original echo over the power of the residual echo signal after cancellation.

$$ERLE=10log_{10}\frac{Pd}{Pe} \quad (27)$$

ERLE measured in dB and for a good echo canceller circuit, an ERLE in the range from 30 dB to 40dB is considered to be ideal. The higher the ERLE, the better the Acoustic Echo Cancellation works.

## 4. Mean Square Error (MSE)

Mean Square Error (MSE) is the sequence of mean squared error. This column vector contains predictions of the mean squared error of adaptive filter at each time instant. The mean squared error is calculated as

$$MSE=\frac{\sum e^2}{n} \quad (28)$$

Measure how can adapt to get accurately model.

## 5. Simulation Result

The main drawback of the LMS algorithm is that it requires a careful choice of the only parameter used for adjusting its behavior, called step size. A too large step size give a fast response to plant changes but results in a large excess mean square error (MSE), and may even cause loss of convergence. A too small step size degrades tracking capabilities of the algorithm.

An optimal step size, giving a trade-off between the speed of convergence and residual error, depends on the power of the input data. Hence, different step size analysis is chosen to compare two adaptive algorithms: LMS and RLS.

In this part, the simulation results for different step size using Matlab.

Methods chosen

LMS and RLS

Filter size

128

Step sizes

1, 0.1, 0.01, 0.001

Forgetting factor for RLS algorithm 1

Sampling Frequency

8kHz

### 5.1. Results of LMS Algorithm

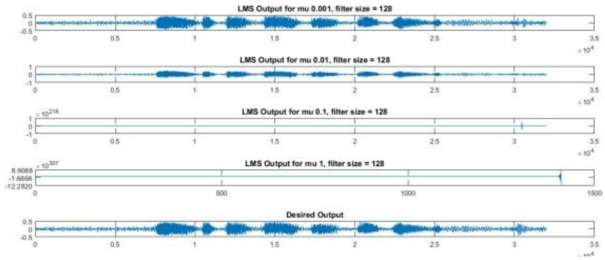



**Figure 4. Sound waveforms for different step sizes using LMS algorithm**

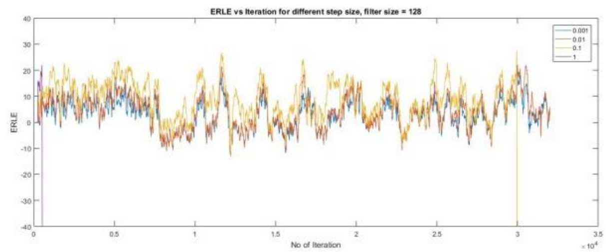



**Figure 5. Comparison value of ERLE using different step size for LMS algorithm (L=128)**

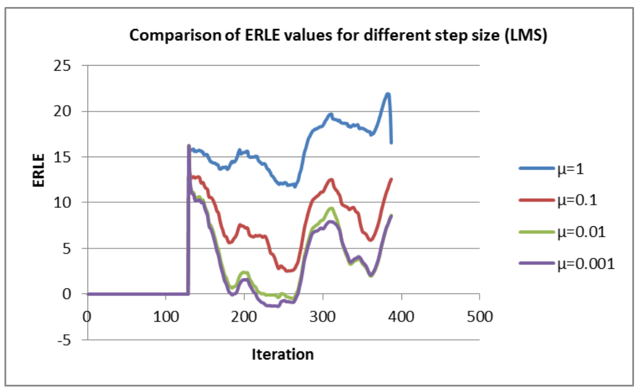



**Figure 6. Comparison value of ERLE using filter length 128 and step size value of 1, 0.1, 0.01, 0.001 (LMS).**

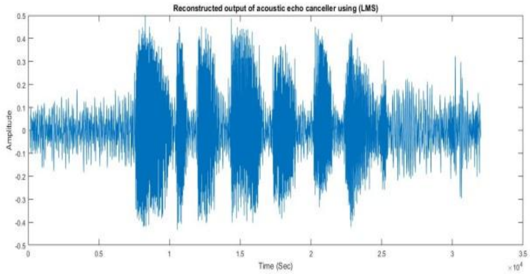



**Figure 7. Output of acoustic echo canceller using step size value of 0.001 and filter length of 128 (LMS).**

**Table 1. Comparison value using different Step Size for LMS algorithm (L=128)**

| Step Size (μ) | 0.001 | 0.01 | 0.1 | 1 |
|---|---|---|---|---|
| ERLE(dB) | 1.386 | 2.1262 | INF | NAN |
| MSE | 0.0156 | 0.0287 | INF | NAN |
| SNR(dB) | 0.2456 | 0.9863 | INF | NAN |

## 5.2. Results of LMS and RLS Algorithms

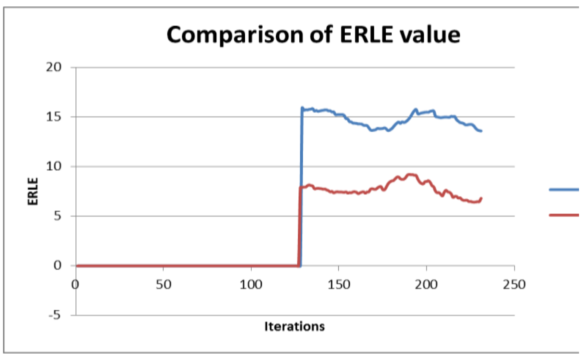



**Figure 8. Comparison value of ERLE using different algorithms**

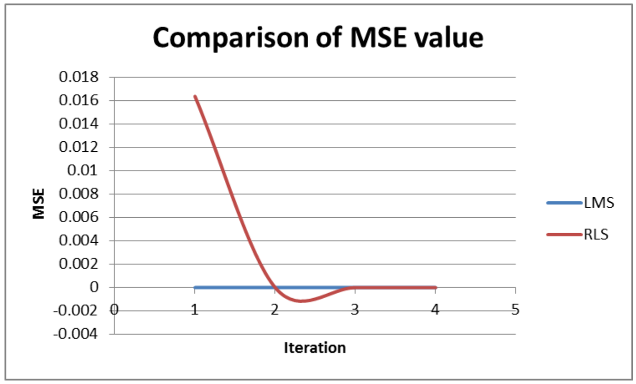



**Figure 9. Comparison value of MSE using filter length 128 and step size value of 0.001**.

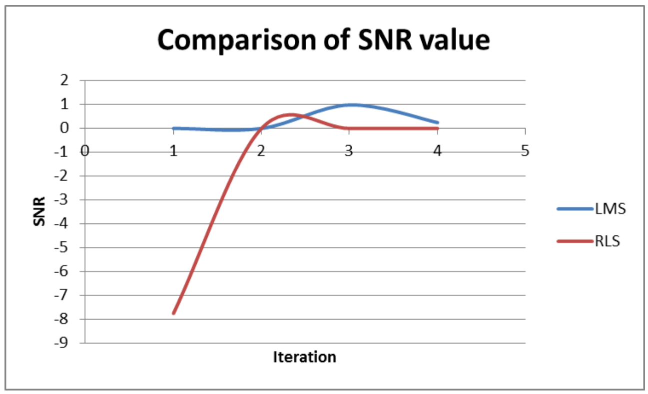



**Figure 10. Comparison value of SNR using filter length 128 and step size value of 0.001.**

## 5.3. Results of RLS Algorithm

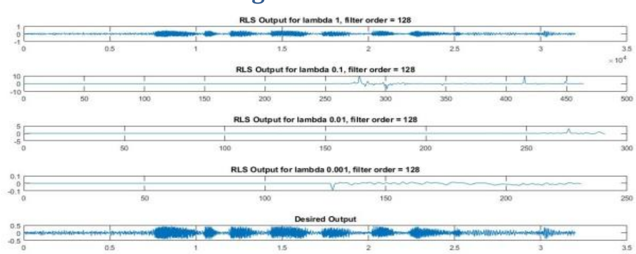



**Figure 11. Sound waveforms for different λ(lambda) sizes using RLS algorithm**

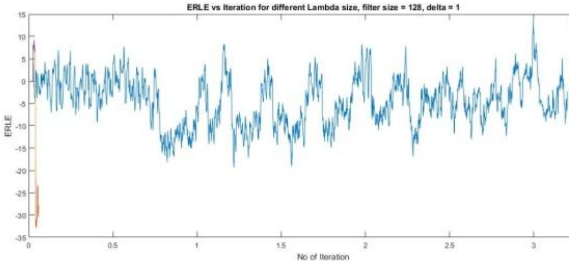



**Figure 12. Comparison value of ERLE using different step size for RLS algorithm with Filter Size value of 128.**

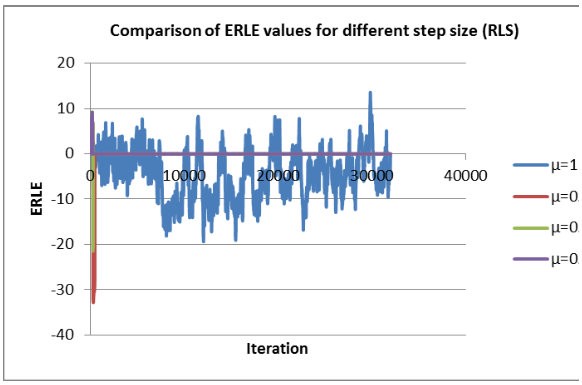



**Figure 13. Comparison value of ERLE using filter length 128 and step size value of 1, 0.1, 0.01, 0.001 (RLS).**

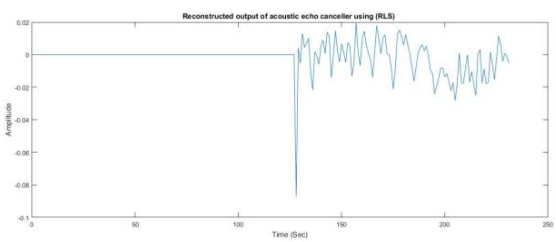



**Figure 14. Output of acoustic echo canceller using lambda 0.001 and filter length 128 (RLS).**

**Table 2. Comparison value using different Step Size for RLS algorithm (L=128)**

| Step Size (μ) | 0.001 | 0.01 | 0.1 | 1 |
|---|---|---|---|---|
| ERLE(dB) | 5.937 | NaN | NaN | NaN |
| MSE | 0.0164 | - | - | - |
| SNR(dB) | 7.7509 | - | - | - |

**Table 3. Advantage and Disadvantage of LMS and RLS algorithms**

| Algorithm | Advantage | Disadvantage |
|---|---|---|
| LMS | The advantage of the LMS algorithms produces fast convergence speed while its shortcoming sub- optimal solution in low signal-to-noise ratio (SNR) environment. | If the value of step size μ is so small then the adaptive filter takes a long time to converge on the optimal solution and in case of large value of the adaptive filter speed convergence will be diverged and become unstable.<br><br>Where, μ is a step-size parameter and it controls the immediate change of the updating factor. It shows a great impact on the performance of the LMS algorithm in order to change its value.<br><br>In the LMS algorithm, the weight vector w(n) changes depending on the input signal x(n) . Thus it will get the problem which is called gradient noise amplification when x(n) is too large. |
| RLS | The RLS algorithm has an advantage of fast convergence. | It has the problem of high computational complexity. |

## 6. Conclusion

In this paper, a performance comparison between the LMS and RLS algorithms has been drawn using the Matlab. The simulations have been done with real time voice signal. Simulations have shown that the RLS algorithm outperforms the LMS algorithm but this high performance is with a trade-off with the high computational complexity of the RLS algorithm. One of the disadvantages of the RLS algorithm inspite of its higher convergence rate is its stability if the autocorrelation matrix is singular.


## Acknowledgements

The authors would like to express their sincere gratitude to Dr. Kyi Soe, Rector of West Yangon Technological University and Dr. Min Zaw Aung, Rector of Technological University Mawlamyine, for their kind pleasure on to carry out this paper. The author wishes to express the deepest gratitude to Dr. Nay Win Zaw, Professor, Head of Department of Electronic Engineering, West Yangon Technological University and Dr. Seint Seint Htwe, Professor, Head of Department of Electronic Engineering, Technological University Mawlamyine, for their helpful and invaluable suggestions, opinion and guidance in preparation of this paper. The author would like to express their sincere to Daw Theingi Ait, supervisor, and Dr. Nay Win Zaw, co-supervisor for their kindly help and invaluable permission, support and encouragements throughout the paper writing period. The authors would like to express their thanks to all persons who have given support during the preparation period of this paper.


## References


[1] J.L.Jini Mary, B.Sree Devi, G.Monica Bell Aseer, 2016, " *Performance Evaluation of Adaptive Filters for Noise Cancellation* ", Department of ECE, VV college of Engineering, Tisaiyanvilai, Volume 4, Issue 2, March-April, 2016

ISSN 2091-2730

[2] JYOTI DHIMAN, May 2013, " *Comparison between Adaptive filter Algorithms (LMS, NLMS and RLS)* ",India, ISSN: 2278 – 7798 International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 5, May 2013.

[3] Thomas Drumright, 1998, " *Adaptive Filtering* ".

[4] Noor K. Muhsin, (2011) " *Noise Removal of ECG Signal Using Recursive Least Square Algorithms* ",Department of Biomedical Engineering/ Al-Khwarizmi College of Engineering/ University of Baghdad, Al-Khwarizmi Engineering Journal, Vol. 7, No. 1, PP 13 - 21 (2011), (Received 11 May 2010; Accepted 17 February 2011).

[5] Thamer M. Jamel, January 2013. "*Performance Enhancement of Adaptive Acoustic Echo Canceller Using a New Time Varying Step Size LMS Algorithm (NVSSLMS)*", University of Technology, department of Electrical Engineering, Baghdad, Iraq, International Journal of Advancements in Computing Technology(IJACT), Korea , Vol. 3, No. 1.

[6] Aman Kumar Sahu, (2014),"*Noise Cancellation Using Adaptive Filters of Speech Signal by RLS Algorithm in Matlab*", C.V.RAMAN University, Kota, Bilaspur, C.G. India, (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611.

[7] Jonathan Cedarleaf, Steve Philbert, Arvind Ramanathan, " *NOISE CANCELLATION USING LEAST MEAN SQUARES ADAPTIVE FILTER*", University of Rochester, Department of Electrical and Computer Engineering.

[8] Vedansh Thakkar, 2017, " *Noise Cancellation using Least Mean Square Algorithm*", Medicaps Institute of Technology and anagement, (IOSR-JECE), e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 12, Issue 5, Ver. I (Sep.- Oct. 2017), PP 64-75.

[9] Prof. Vaishali M. Karne,2013, " *LEAST MEAN SQUARE (LMS) ADAPTIVE FILTER FOR NOISE CANCELLATION* ", G.N.I.E.T, Nagpur, India, ISSN 2319 - 4847 Special Issue for National Conference On Recent Advances in Technology and Management for Integrated Growth 2013

# Stability of Transfer Function in Discrete-Time System Using MATLAB SIMULINK

**Khaing Zin Win, Myint Myint Yi, Zay Oo Maung, Phyu Pyar Wai**
*Electronic Engineering Department, TU(Hmawbi)*
*khaingzinnthet@gmail.com*
*myintmyintyitdu@gmail.com*
*zayoomaung@gmail.com*
*ppw131180@gmail.com*

## Abstract

*Digital Signal Processing is important in Electronics Engineering, control and computer communication, telecommunication, biotech-nological, seismology, economic and etc. the performance of the signal processing can be investigated in time, space and frequency domain. The transfer function helps to find the range of stability of any system. In this paper, stability condition of IIR filters is checked in z-plane by using system network. The main aim of this paper is to analyze the IIR filters response with different orders in time domain and frequency domain based on stability condition. IIR filters are implemented by using the concept of Digital Signal Processing and MATLAB SIMULINK.*

**Keywords:** Stability, Transfer function, Z-plane, IIR filter, MATLAB SIMULINK

## 1. Introduction

Filters can be used in many applications such as in noise reduction signal processing and communication systems, channel equalization, audio-video signal processing, radar field, data analysis. A filter is and electrical network that can transmit signal within a specified frequency range, pass band and the spread signal is called stop band. The frequency separates the pass band and the stop band is known as cut-off frequency. There are two types of digital filters. They are Finite Impulse Response Filter (FIR) and Infinite Impulse Response Filter (IIR). An FIR filter requires more computation time on the digital signal processing and more memory. On the other hand, Infinite Impulse Response (IIR) with feedback filters is fundamental elements of digital signal processing. IIR filters can achieve the desired filtering characteristics using less memory and computations than FIR filters.

IIR Filters are the digital filters that have an infinite impulse response to remove unwanted signal. They are also known as recursive filters as they have a feedback and hence they produce a better frequency response [2]. The Z-transform is important in digital signal processing to perform filter design and system analysis. Stability of the transfer function plays a vital role in the desired system. In this paper, the IIR low pass filters have been designed and implemented stability condition with different transfer function using MATLAB software. The Z-transform is defined as follow:

$$X(Z) = \sum_{K=0}^{N} x[k] z^{-k} \quad (1)$$

where the sequence support interval is [0,N] and Z is any complex.

## 2. Stability of Transfer Function

The transfer function of the discrete system, H(Z) is defined by

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z-trnsform\ of\ output\ sequence}{z-trnasform\ of\ input\ sequence} \quad (2)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{H_0 \prod_{i=1}^{Z}(z-z_i)^{m_i}}{\prod_{i=1}^{p}(z-p_i)^{n_i}} \quad (3)$$

The transfer function can be calculated from (i) difference equation characterizing the system (ii) a network representation of the system (iii) a state-space characterization. [1] In this paper, the transfer function is calculated by using network representation to analyze stability of the discrete time system.

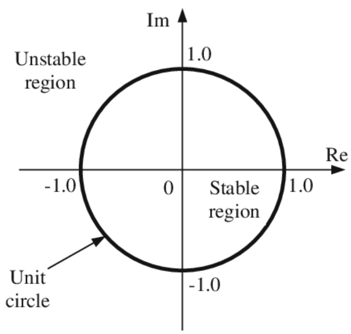



**Figure 1. Pole Location in Z-plane**

In Z plane, poles on the positive real z axis and within the unit circle $(-1 < p_i < 1)$ produce a converging series and a stable response. To stable the system, all poles of transfer function are inside the unit circle or the magnitude of poles is less than 1, $|P_i| < 1\ for\ i = 1,2, \dots\dots, N$ .The pole's location for the stable system is illustrated in figure 1. [1]

## 3. Implementation of Transfer Function for IIR Filter

IIR filters depends linearly on finite number of input samples and a finite number of Previous outputs of filter. The transfer function H(Z) of the implemented IIR is calculated from the network representation in figure 2.

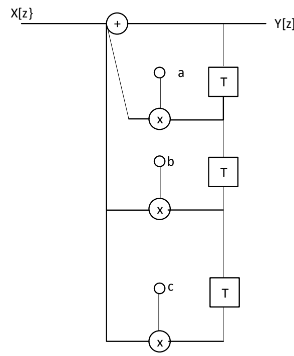



**Figure 2. Network for Recursive System**

This network is represented third order low pass filter. So, its transfer function can be calculated into three parts: first order, second order and third order. The related transfer functions are:

The transfer function for first order LPF is

$$H(Z) = \frac{1}{1 + aZ^{-1}}$$

The transfer function for second order LPF is

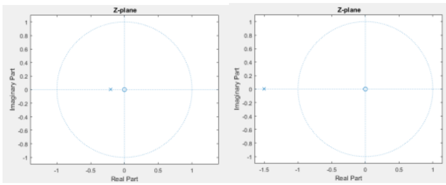
$$H(Z) = \frac{1}{1 + aZ^{-1} + bZ^{-2}}$$

The transfer function for third order LPF is

$$H(Z) = \frac{1}{1 + aZ^{-1} + bZ^{-2} + cZ^{-3}}$$

### 3.1. Testing for Stable and Unstable Condition Z-plane using MATLAB

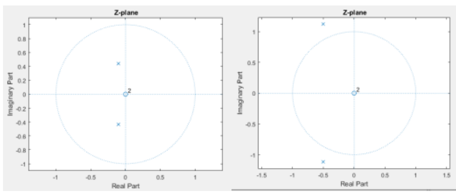
The transfer functions of the desired system network for IIR filter is tested whether it is stable or not in z-plane using MATLAB software. The pole location is selected and outside within the unit circle. Testing condition of first order LPF is shown in figure 3.




(a) (b)

**Figure 3. Testing Condition of First Order LPF (a) Stable Condition (b) Unstable Condition**

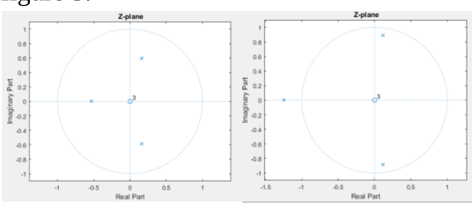
Testing condition of second order LPF is shown in figure 4.




(a) (b)

**Figure 4. Testing Condition of Second Order LPF (a) Stable Condition (b) Unstable Condition**

Testing condition of third order LPF is shown in figure 5.




(a) (b)

**Figure 5. Testing condition of third order LPF (a) Stable Condition (b) Unstable Condition**

## 4. Simulation Results

After testing the stability of the transfer function, IIR LPF is implemented by using MATLAB SIMULINK as shown in figure 6. The output of discrete time system with different order and different stability conditions are analyzed in time domain and frequency domain.

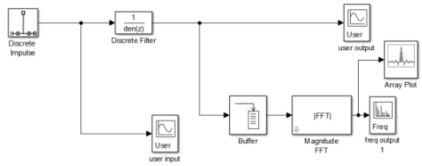



**Figure 6. Implemented System Block Diagram for the System**

The testing condition of input time domain for various order is shown in figure 7.

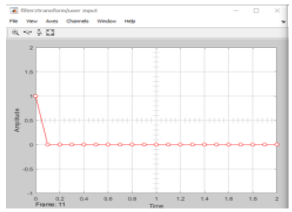
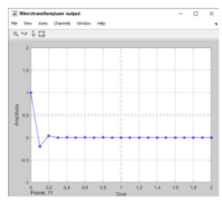
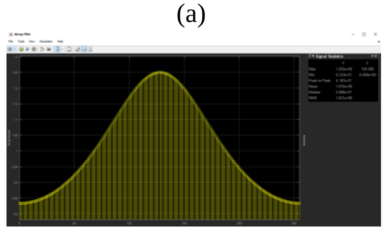


**Figure 7. Testing Condition Input System in Time Domain**

The Stability testing condition of discrete time system with first order LPF in time domain and frequency domain are shown in figure 8.



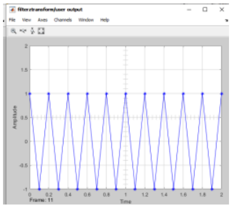
(a)



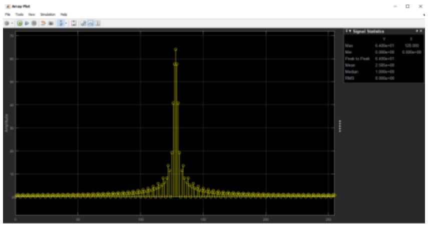
(b)

**Figure 8. First Order Stable Condition of Low Pass filter (a) Time Domain (b) Frequency domain**

The Instability testing condition of discrete time system with first order LPF in time domain and frequency domain are shown in figure 9.



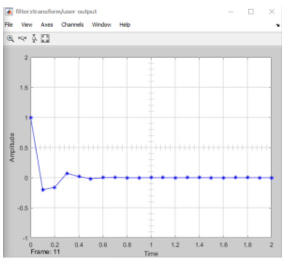
(a)



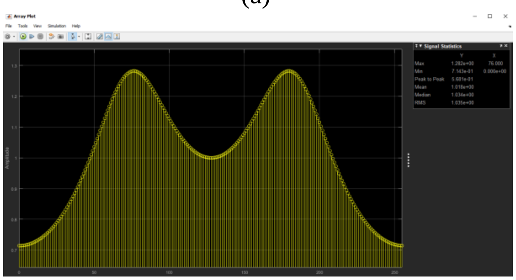
(b)

**Figure 9. First Order Unstable Condition of LPF (a) Time Domain (b) Frequency Domain**

The Stability testing condition of discrete time system with second order LPF in time domain and frequency domain are shown in figure 10.



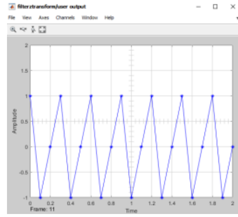
(a)



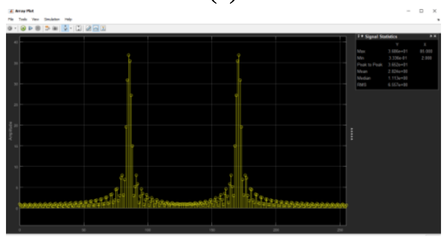
(b)

**Figure 10. Second Order Stable Condition of LPF (a) Time Domain (b) Frequency Domain**

The Instability testing condition of discrete time system with second order LPF in time domain and frequency domain are shown in figure 11.



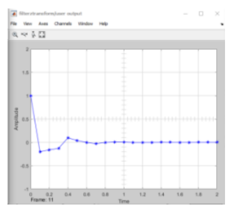
(a)



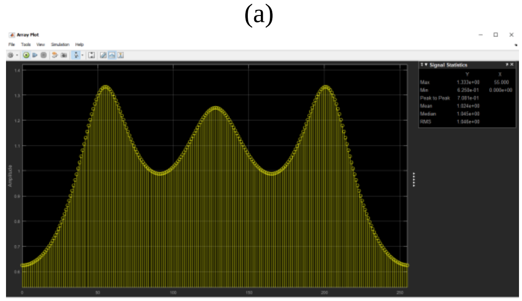
(b)

**Figure 11. Second Order Unstable Condition of LPF (a)Time Domain (b) Frequency Domain**

The Stability testing condition of discrete time system with third order LPF in time domain and frequency domain are shown in figure 11.



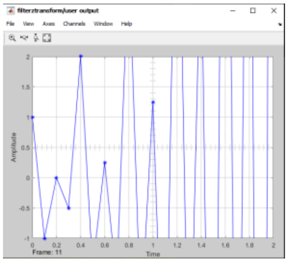
(a)



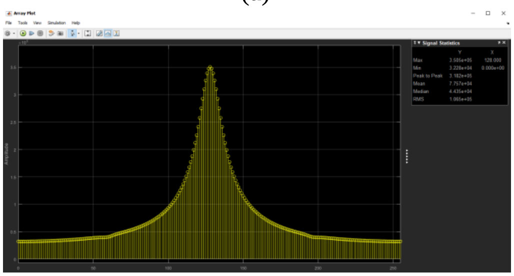
(b)

**Figure 11. Third Order Stable Condition of LPF (a) Time Domain (b) Frequency Domain**

The Instability testing condition of discrete time system with second order LPF in time domain and frequency domain are shown in figure 12.



(a)



(b)

**Figure 12. Third Order Unstable Condition of LPF (a) Time Domain (b) Frequency Domain**

From the experimental results, the response of filter in stable condition is closer to the desired passband with less attenuation. However, the unstable condition deviates from desired and bandwidth of passband is narrow with more attenuation.

## 5. Conclusion

The stability conditions of IIR lowpass filter are especially analyzed in Z-plane using system network with various orders. According to the experimental results, the passband of the stable system is wide and closer to the ideal response with less attenuation. On the other hand, response of the unstable system is narrow and deviate from ideal response with more attenuation. Similarly, the stability and response for Highpass, Bandpass and Bandstop filters can be investigated with this method.


## ACKNOLEDGMENTS

The author is deeply gratitude to Dr. Phyu Pyar Wai, Professor and Head, Department of Electronic Engineering, Technological University (Hmawbi), for her kind help and invaluable suggestions. The author wishes to extend grateful thanks to Dr. Myint Myint Yi, Professor and Dr. Zay Oo Maung, Associate Professor, Department of Electronic Engineering, Technological University (Hmawbi) for their supervision and kindly support in all the time of this research work.


## References


[1] Andreas Antonious, "*Signals, Systems and Filters*", McGraw-Hill Compaines, 2006.

[2] ModiRishabhkumar N. and PramitiParashar, "*Design and Implementation of Butterworth, chebyshev.I filters for Digital Signal Analysis*", volume:05 Issue:03, IRJET, Mar-2018.

[3] C.SidneyBurrus, "*Properties of IIR filters*", openstax.CNX module:m116898.

[4] John Semmlow, "*IIR filters, signals and systems for Bionegineers*", second Edition,2012.

[5] Nasser Kehtarnavaz, *"Digital filter,Digital Signal Processing System Design*", second Edition,2008.

[6] Lester Liu, *"Z-transform FIR and IIR Filters Lowpass, Band-pass and High-pass filters"*, Lecture2, October, 2014

# Interconversion of Various Number Systems In Digital Technology

Moe Moe Thein
*Faculty of Computer Science and Technology*
*cummthein.74@email.com*
Thae Thae Han
*Information Technology Supporting and Maintenance Department*
*thae2hann@email.com*
Nyein Nyein Hlaing
*Facuty of Computer Science and Technology*
*nyeinnyeinhlaing@email.com*

## Abstract

*The A number system is a set of rules and symbols used to represent a number. Any system that used for representing numbers called numeral systems. Everyone is familiar with decimal number system using ten digits. However, computers use binary number system, using only two digits 0 and 1 based on concept of decimal number system. Other various number systems used this concept based on decimal system i.e. quaternary, secondary, octal, duodecimal, Quadra decimal, hexadecimal and vigesimal number system using four, six, eight, twelve, fourteen, sixteen and twenty digits . Conversion of number systems is essential for understanding of computer. The programming for digital devices requires precise understanding of these formats. Conversion of number system requires a lot of techniques. In this paper, we illustrate s the concepts of number systems, arithmetic calculations using tabulated form. It will offer a step towards theses number systems to understand.*

**Keywords:** Number System, Conversion, number's arithmetic

## 1. Introduction

We are familiar with decimal numbers using the numbers 0-9 since childhood. When we deal with computer and digital devices, it needs how a number will be used in digital world; we need knowledge of various number systems such as binary, octal, decimal and hexadecimal. Moreover, the use of the microprocessor requires these number systems. Although human beings use the decimal system with ten digits 0-9, computers communicate and run on binary digits 0, 1.

The number systems used in digital technology have a great variety. Few of these systems are; unary (base 1),

These all number systems use unique and distinct symbols. Some of these systems use numeric digits (0, 1, 2, -----, 9). For example, in unary (base-1) to represent a number T, an arbitrarily chosen symbol repeats T times. The number 3 is represented as III using the symbol I (tally mark). Similarly, binary system (base-2) represents 0& 1 while ternary system (base-3) uses 0, 1 and 2.

Undecimal (base-11) requires eleven symbols represent 0 to 10. These "10" symbol is an alphabet A in hexadecimal vigesimal system (base-20) uses 20 distinct symbols:0, -----9, A, B, ----, E, F, G, H, I and J (I can avoid so that not confused with I). Quadsexavigesimal system (base-64) uses A-Z, a-z and 0-9 for the first 62 values, + and / for the last two values.

In digital world, we will need to learn about number systems involved mathematics calculation, such as addition, subtraction, multiplication and division. In case, when data store in computer memory to represent negative numbers, compliments use.

## 2. Ease of Use

Humans are speaking to one another in a particular language made of words and letters the computer does not understand the words and letters when we type these. These words are translated into numbers understudied by computer.

### 2.1. Digits

The digit of number system must understood before number convert from one number to another. The first digit is always zero larger numbers are constructs by using positional notation. The units' position of $10^0$ is 1, $10^1$ is 10. The exponential powers are critical in numbering systems.

### 2.2. Number Representation

A number is represented as a general format'

$N=A_nB^n+A_{n-1}B^{n-1}+\text{-----}+A_1B^1+A_0B^0$

Where, N=Number, B=Base, A=Any digit

For example number 207 can be represented in various number systems as follows in table1

**Table 1. Number Representation in Various Number System**

| | | | |
|---|---|---|---|
| Decimal | 207 | $2*10^2+0*10^1+7*10^0=200+0+7$ | 207 |
| Binary | 11001111 | $1*2^7+1*2^6+0*2^5+....+0*2^0$ | 207 |
| Quaternary | 3033 | $3*4^3+0*4^2+3*4^1+3*4^0$ | 207 |
| Senary | 543 | $5*6^2+4*6^1+3*6^0$ | 207 |
| Octal | 317 | $3*8^2+1*8^1+7*8^0$ | 207 |
| Duodecimal | 153 | $1*12^2+5*12^1+3*12^0$ | 207 |
| Quadrodecimal | 10B | $1*14^2+0*14^1+B*14^0$ | 207 |
| Hexadecimal | 12F | $1*16^2+2*16^1+F*16^0$ | 207 |
| Vigesimal | 17 | $1*20^1+7*20^0$ | 207 |

### 2.3. MSD and LSD

The MSD in a number that has the greatest effect while the LSD in a number that has the least effect

MSD LSD

3 0 6 . 0 6 3

MSD LSD

5 0 4 0 . 0

### 2.4. Binary Number

The number system with base 2 is known as the binary system. Only two symbols 0 4 1 are used in this system.

$2^2$ $2^1$ $2^0$ . $2^{-1}$ $2^{-2}$

## 2.5. Quaternary Number

The number system with base 4 is known as the quaternary number system. Only four symbols such as 0, 1, 2 and 3 are used in this system. These systems are used in 2D Hilbert curves.

$4^2$ $4^1$ $4^0$ . $4^{-1}$ $4^{-2}$

## 2.6. Senary (Heximal) Number System

The number system with base is also known as the senary number systems. Only 6 symbols such as 0, 1, 2, 3, 4 and 5 are used in this system. This system is used in the study of prime numbers.

## 2.7. Octal Number

The number system with base 8 is also known as the octal number systems. Only 8 symbols are used in this system. Octal numbers are used for engineering binary data and displaying certain information.

## 2.8. Decimal Number Systems

The decimal number system is called intersectional system of number. It has ten as its base

6 2 6 . 4

$10^2 = 6*100 = 6000$

$10^1 = 2*10 = 20$

$10^0 = 6*1 = 6$

$10^{-1} = 4*0.1 = 0.4$

## 2.9. Duodecimal (or Dozental) Number Systems

The number system with base (or radix) 12 is known as the duodecimal number system. It is more convenient number system for computing fractions than other systems.

## 2.10. Hexadecimal Number Systems

The number system is base 16 which requires 16 distinct symbols to represent the number. his system is very popular in computer uses.

## 2.11. Hexadecimal Number Systems

The number system is base 20 which requires twenty symbols. Since there are more than ten common digits, the notation can be extended by using letters A,B,C,D,E,F,G,H,I and J to represent 10,11,12,13,14,15,16,17, 18 and 19.this system is widely used nearly all over the world in various languages.

# 3. Arithmetic Calculations

The arithmetic is the most basic branch mathematics used by everyone from simple day-to-day counting to advanced science and business calculations.. As the present, binary number system is the most common number system used by computer systems. Long ago, computer system used the decimal number system. Systems designers have discovered that binary arithmetic is better than the decimal arithmetic for calculations. But, decimal arithmetic is used in many software systems. Therefore, the need for decimal is persists.

## 3.1. Addition

In addition, a one quantity is added to another.Table2 shows addition table.

| | |
|---|---|
| Binary | $(0110)_2+(010)_2=(1000)_2$ |
| Quaternary | $(22)_4+ (11)_4=(33)_4$ |
| Senary | $(30)_6+ (20)_6=(50)_6$ |
| Octal | $(46)_8+ (51)_8=(117)_8$ |
| Decimal | $(09)_{10}+ (61)_{10}=(70)_{10}$ |
| Duodecimal | $(27)_{12}+ (0A)_{12}=(35)_{12}$ |
| Quadrodecimal | $(3B)_{14}+ (24)_{14}=(61)_{14}$ |
| Hexadecimal | $(2C)_{16}+ (18)_{16}=(45)_{16}$ |
| Vigesimal | $(8E)_{20}+ (10)_{20}=(9E)_{20}$ |

### 3.2. Subtraction

Subtraction is the opposite of addition.Table3 shows subtraction table in various number systems.

| | |
|---|---|
| Binary | $(1000)_2-(001)_2=(110)_2$ |
| Quaternary | $(20)_4- (13)_4=(1)_4$ |
| Senary | $(40)_6- (12)_6=(24)_6$ |
| Octal | $(67)_8- (22)_8=(45)_8$ |
| Decimal | $(87)_{10}- (16)_{10}=(71)_{10}$ |
| Duodecimal | $(72)_{12}- (06)_{12}=(62)_{12}$ |
| Quadrodecimal | $(4B)_{14}- (12)_{14}=(39)_{14}$ |
| Hexadecimal | $(FF)_{16}- (99)_{16}=(66)_{16}$ |
| Vigesimal | $(8E)_{20}- (0B)_{20}=(83)_{20}$ |

### 3.3. Multiplication

It combines two numbers into a single number called product.

| | |
|---|---|
| Binary | $(110)_2*(11)_2=(10010)_2$ |
| Quaternary | $(12)_4* (10)_4=(120)_4$ |
| Senary | $(40)_6*(02)_6=(120)_6$ |
| Octal | $(70)_8*(20)_8=(2000)_8$ |
| Decimal | $(65)_{10}*(22)_{10}=(1430)_{10}$ |
| Duodecimal | $(2A)_{12}*(11)_{12}=(2CA)_{12}$ |
| Quadrodecimal | $(5A)_{14}*(16)_{14}=(824)_{14}$ |
| Hexadecimal | $(2E)_{16}*(10)_{16}=(2E0)_{16}$ |
| Vigesimal | $(3B)_{20}*(10)_{20}=(3B0)_{20}$ |

### 3.4. Division

It is basically the opposite of multiplication. Division obtains the quotient of two numbers, when the divided by the divisor.

| | |
|---|---|
| Binary | $(0110)_2/(011)_2=(10)_2$ |
| Quaternary | $(22)_4/(11)_4=(02)_4$ |
| Senary | $(40)_6/(20)_6=(02)_6$ |
| Octal | $(60)_8/(10)_8=(06)_8$ |
| Decimal | $(90)_{10}/(05)_{10}=(18)_{10}$ |
| Duodecimal | $(88)_{12}/(44)_{12}=(02)_{12}$ |
| Quadrodecimal | $(5A)_{14}/(16)_{14}=(04)_{14}$ |
| Hexadecimal | $(2A)_{16}/(02)_{16}=(15)_{16}$ |
| Vigesimal | $(3C)_{20}/(03)_{20}=(14)_{20}$ |

## 4. Conclusion

In this paper, we present the various number systems used in the digital technology specifically computing devices. The proposed table cover almost everything associated to those most common the number representation, allowed digits in each number system, arithmetic of each number system. It will be helpful for the people who are new in the field of computer science or digital electronics. As a future work, more number systems and techniques can add in it. Software and Hardware can implement with the help of this proposed table.

## References


[1] BARRY B, BREY, " The Intel Microprocessors, Sixth edition prentice hall of India private limited", New Delhi-110 001, 2002.

[2] Chu, y..Digital Engineering New York: McGraw-Hill Book Co.,1962.

[3] D.Nasib S.Gill, JB Dixit,"Digital Design nd Computer".

[4] Shahid Latif, Rahat Ullah, Hamid Jan, "A Step towards an Easy Interconversion of Various Number Systems"

[5] R.P.Jain "Modern Digital Electronics", 4th edition, Published by Tata McGraw Hill Education Private Limited, 7 West Patel Nagar, New Delhi 110 008.


## Copyright Policy


**University of Computer Studies (Pakokku)**
**Department of Higher Education**
**Ministry of Education**
**Myanmar**