NON-STATIONARY WIRELESS CHANNEL EQUALIZER ENHANCEMENT USING STA

COMPREHENSIVE OVERVIEW, IMPLEMENTATION AND COMPARATIVE ANALYSIS OF DIFFERENT DESIGN TECHNIQUES AIDED BY SINUSOID TRACKING ALGORITHMS

Non-stationary wireless channel equalizer enhancement using STA

COMPREHENSIVE OVERVIEW, IMPLEMENTATION AND COMPARATIVE ANALYSIS OF DIFFERENT DESIGN TECHNIQUES AIDED BY SINUSOID TRACKING ALGORITHMS

Submitted by:

Muhammad Abubakar

Supervised by:

Dr. Pervez Akhtar

THESIS

Submitted to:

Department of Electronic and Power Engineering

Pakistan Navy Engineering College Karachi

National University of Science and Technology, Islamabad Pakistan

In fulfillment of requirement for the award of the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

With Specialization in Communication

July 2013

بسم الله الرحمن الرحيم

ABSTRACT

This thesis is a comprehensive research work to design an estimator for the purpose of channel equalization in time varying channel characteristics. The purpose is to discover techniques which can help in improving bad channel reception for wireless channels. The study is initialized by the well-known least mean square (LMS) filter by studying its effectiveness and performance for the noise channel environments. Further on a recently developed tangent directed variable step LMS filter is implemented and its performance is compared to that of the LMS filter. The major step taken in this work is the implementation of a sinusoid tracking algorithm (STA) as a channel equalizer, it is a blind nonlinear adaptive algorithm having excellent tracking performance in high noise environments. The STA implementation variants are implemented and analyzed by their estimation performance for the same channel environments used for the earlier LMS and variable step LMS. Learning from the different implementation results a variable step LMS based preprocessing is implemented along with the STA core and STA frequency directed variants for the STA to add redundancy to the equalizer without compromising performance. This design outperforms the individual implementations and is more redundant to the non-stationary channel behavior. The Simulink based implementation results are shown and discussed in detail to conclude the performance and estimation effectiveness. Xilinx System generator based Implementation for the STA architecture is also designed and implemented successfully using XSG DSP block set. The architecture developed in XSG is also optimized in terms of area, speed and power utilization to create a XSG based sub block which can be used for FPGA based implementation of the sinusoid tracking algorithm in the development of real-time applications.

ACKNOWLEDGEMENT

This work concluded due to the support and technical knowledge I got form my teachers and my friends especially Dr. Pervez Akhter who is also my research advisor advised me to the core of knowledge relating the adaptive filter theory and design. The FPGA based implementations are solely possible due to the motivation provided by Dr. Arshad Aziz.

I thank my family for their support and patience which helped me achieve this momentous task.

I thank my friend Engr. Saad-bin-Ayaz for his support and help regarding the adaptive filter applications and design tasks.

The motivation which I got for such type of work is from many of my teachers. I also thank all of them for their trust and the knowledge they shared with me.

CONTENTS

TABLE OF FIGURES

LIST OF ABBREVIATIONS

Amplitude Directed (AD)

Direct Digital Synthesizer (DDS)

Digital Signal Processor (DSP)

Frequency Directed (FD)

Frequency Directed Sinusoid Tracking Algorithm (STA FD)

Field Programmable Gate Array (FPGA)

Finite impulse Response (FIR)

Inter Symbol Interference (ISI)

Least Mean Square (LMS)

Recursive Least squares (RLS)

Sinusoid Tracking Algorithm (STA)

State Space RLS (SSRLS)

System Generator (SYSGEN)

Xilinx System Generator (XSG)

CHAPTER NO 1:

INTRODUCTION

CHAPTER 1 OVERVIEW:

Channel equalization is a vague term used to define the approximately true estimation for the received signal in the presence of different types of channel behavior [b.1]. It depends on our modulation scheme that what is the reception area of interest for our signal, i.e. amplitude, frequency, phase (what type of modulation for our message is used and needs to be estimated correctly). Equalization is preferred to be adaptive as the response of the channel of interest usually varies with time and the behavior can also be nonlinear in nature [a.1].

FIGURE 1-1 OVERVIEW OF CHANNEL EQUALIZATION

1.1 EQUALIZER:

An entity or a subsystem which is capable of matching some desired response to a specific range of inputs and generate a desired output. It can be linear FIR structure to simplify the computation and the architecture as well as it can also be adaptive in nature to cater time varying desired responses [a.2]. Generically an equalizer can also be termed as a filter as both have same structure and same behavior when implemented to cater random channel effects [a.4].

1.2 ADAPTIVE FILTER:

A filter whose characteristics vary with time to reduce the error between output and desired output [a.1]. The error is used as a reference to the convergence to minimize the convergence time by implementing different error measurement schemes. Adaptive filter can be implemented on either FIR or IIR structures effectively [a.2]. Depending on the type of algorithm used for estimation of output they are broadly characterized into two classes' e.g. linear and non-linear adaptive filters.

1.3 CHANNEL:

It is a broad term used to mention a medium which is in use of any quantitative value for the purpose of its movement. There are many types of channels in general and the term is used freely in fields ranging from Science, agriculture to linguistics etc. [c.2] As an electrical researcher the channels we are concerned with are the ones referred as the communication channel.

13.1 COMMUNICATION CHANNEL:

Communication channels can be wired or wireless. Wired communication channels can be EM or optical in nature e.g. coaxial cables and optical fibers. Wireless communications are radio frequency channels, these can be line of sight or non-line of sight e.g. free space communication links like the deep space network or terrestrial communication links like the cellular networks [a.6].

1.3.2 CHANNEL EFFECTS:

As mentioned above no matter what type of channel is being used as long as it is not ideal it will deploy some of its own characteristics in the signal, which can be either any one or a combination of few of the characteristics the channel possesses, which can lead in the received signal as inter symbol interference, fading, delay or shift [b.2].

1.3.3 COMMON CHANNELS:

The most common model for a channel is the Gaussian model whose response follows the Gaussian probability density function. This behavior shows the inputs to the channel are distributed normally in frequency when they are received at the receiver. Another type of channel model is the multipath model modeled by the relay distribution function [a.7]. The distribution for relay pdf is non-symmetric and can significantly induce fading behaviors at the receiver.

1.4 INTRODUCTION TO THESIS:

This is a comprehensive comparative analysis and implementation of STA based algorithms for wireless channel equalization. Discrete models based on Simulink design environment are developed and successfully implemented to form a benchmark for the STA based implementations for signal processing applications.

In chapter 2 the literature review, adaptive linear combiner and the basic minimum mean square filter derivation is discussed along with different implementation strategies.

In chapter 3 the basic LMS algorithm is discussed, derived, implemented and evaluated.

In chapter 4 the arc tangent variable step LMS algorithm is discussed, derived, implemented and evaluated.

In chapter 5 the STA core algorithm is discussed, derived, implemented and evaluated.

In chapter 6 the STA frequency directed algorithm is discussed, derived, implemented and evaluated.

In chapter 7 the comparative analysis is performed between different filters implemented in previous chapters.

In chapter 8 the redundant technique of post processing using STA cores for an arc tangent LMS equalizer is introduced and evaluated and its effectiveness is shown and discussed.

In chapter 9 the STA based preprocessor is implemented for an arc tangent LMS to conclude even better results than before and a fair comparison is evaluated and concluded.

In chapter 10 the FPGA based efficient implementation for the sinusoid tracking algorithm using Xilinx system generator is performed and evaluated.

1.5 OBJECTIVE OF THESIS:

The work done in this thesis is to test the capability of the STA based algorithms for their advance implementation in the field of adaptive signal processing [a.4]. All the models that are developed for this thesis can easily be implemented on discrete hardware like the DSP or FPGA's. Similarly the performance comparisons done can also aid in the future development of more versatile and redundant systems.

CHAPTER NO 2:

LITERATURE REVIEW

CHAPTER 2 LITERATURE REVIEW:

2.1 ADAPTIVE SYSTEMS:

Any system whose working methodology readjusts itself to cater change in the environment and maintain or improve its performance [a.2]. In our line of research the adaptive systems are developed and implemented for adaptive control and adaptive signal processing majorly. Adaptive systems have the following major merits [a.1].

- Auto adaptation in non-stationary environments.
- Adaptive filtering and decision directed tasks via programming or training.
- They self-design themselves minimizing synthesis and design time.
- They adapt themselves to cater new situations.
- Some systems can adaptively reroute themselves to avoid damaged areas and remove defects. They are also can have limited auto repair.
- Their mathematical models are usually nonlinear and non-stationary.
- They are preferred for real-time applications.
- They can sustainably improve system performance even when the input signals or data streams have unknown characteristics or behavior.

2.1.1 AREAS OF APPLICATION:

Adaptive systems have wide area of applications ranging from linear filtering applications to extremely complex nonlinear systems design and management [b.3]. Some major application groups which benefit from adaptive processing architectures are as follows [a.2].

- Noise redundant filtering applications.
- Response identification systems.
- Equalization response matching.
- System response estimation.
- Adaptive predictor applications.

2.1.2 TYPES OF ADAPTIVE SYSTEMS:

2.1.2.1 OPEN LOOP:

Open loop adaptation is regarded when we have external related data which can be used to tune or adapt the subsystem.

2.1.2.2 CLOSED LOOP:

A feedback system is incorporated in closed loop adaptation for the purpose of minimizing the system deviation and is more suitable for quickly matching the response to our desired output [a.5].

Some general structure configurations for the closed loop system are pictorially illustrated below.

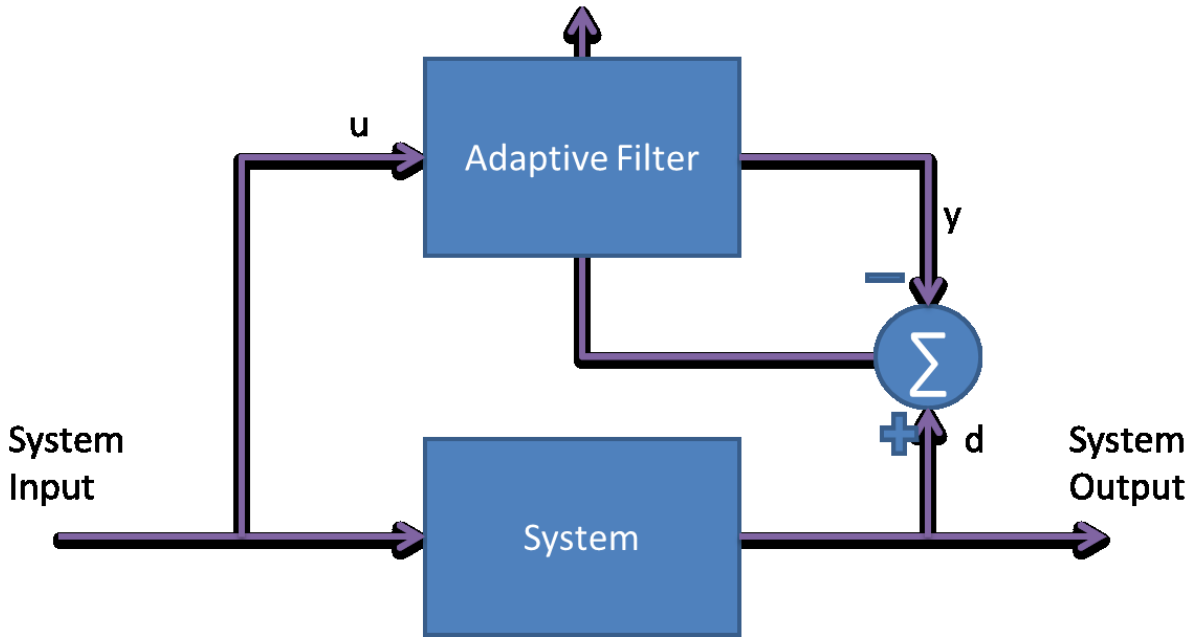


FIGURE 2-1 SYSTEM IDENTIFICATION

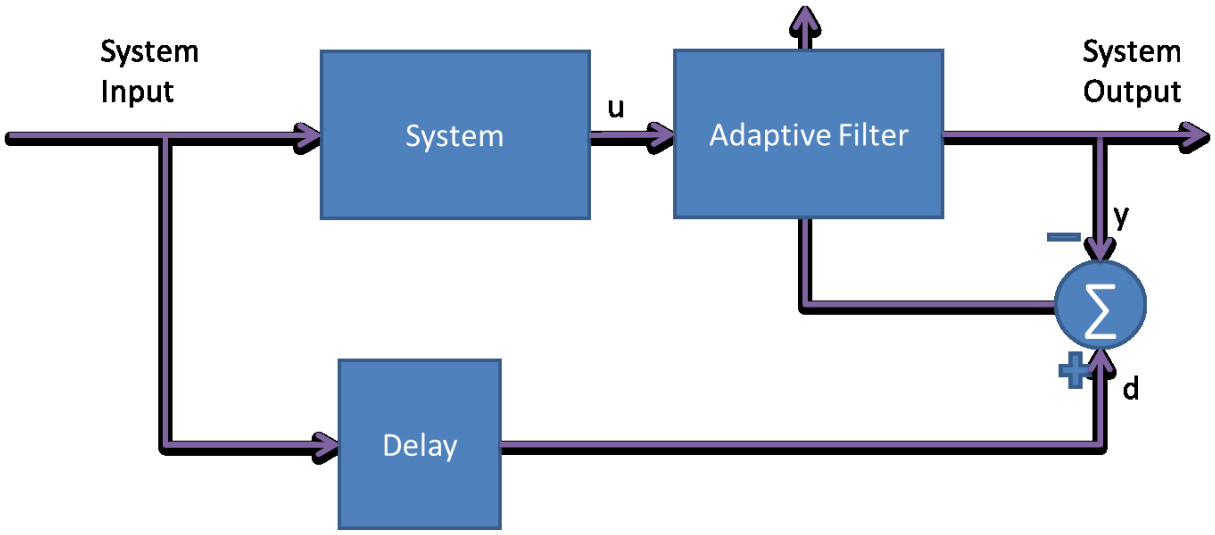


FIGURE 2-2 SYSTEM INVERSE MODELING

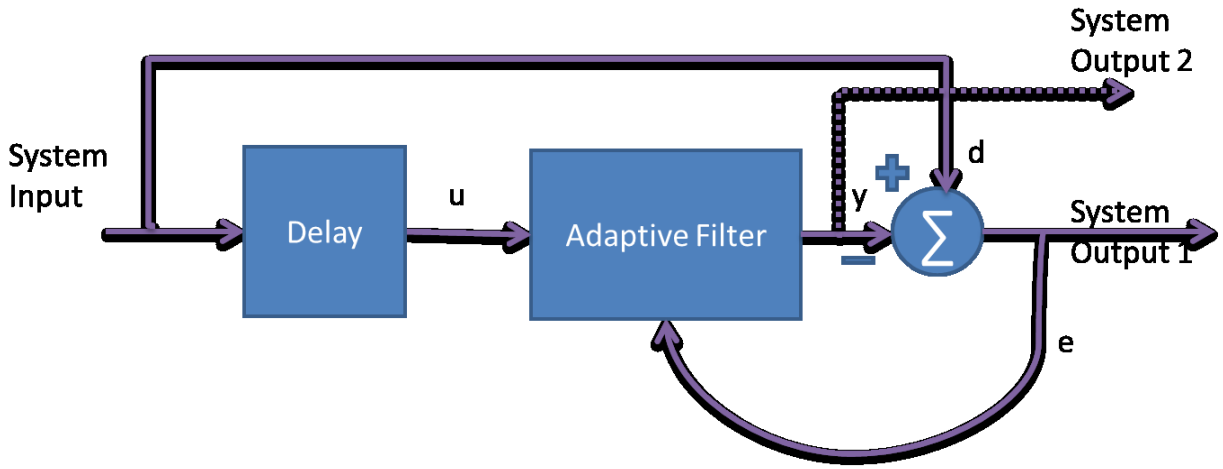


FIGURE 2-3 SYSTEM PREDICTION

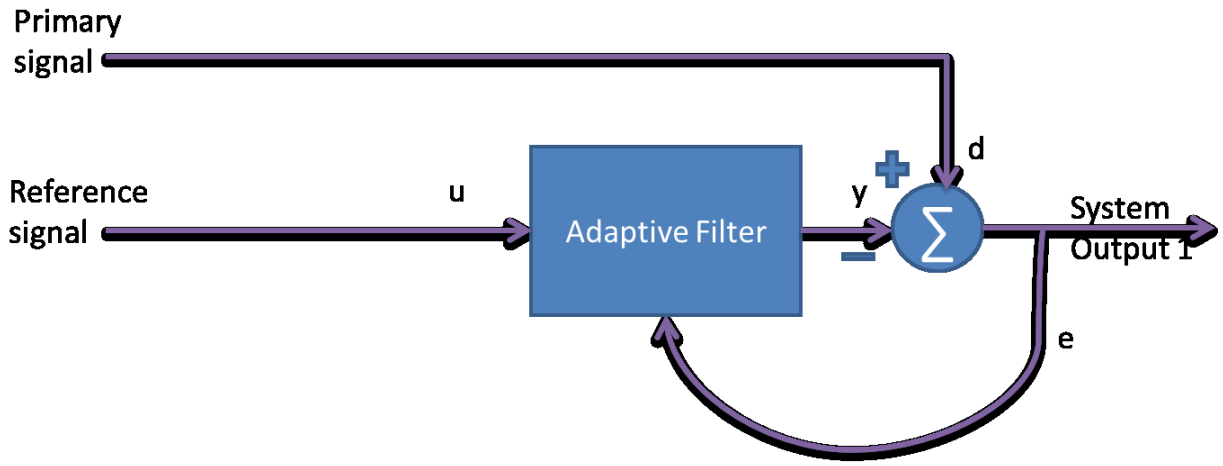


FIGURE 2-4 INTERFERENCE CANCELING

2.2 ADAPTIVE LINEAR COMBINER:

This type of structure is generic to adaptive signal processing. This general form is applied with little variation in many applications for the purpose of adaptive signal processing. It can be designed as a single input system or multiple inputs system [a.1].

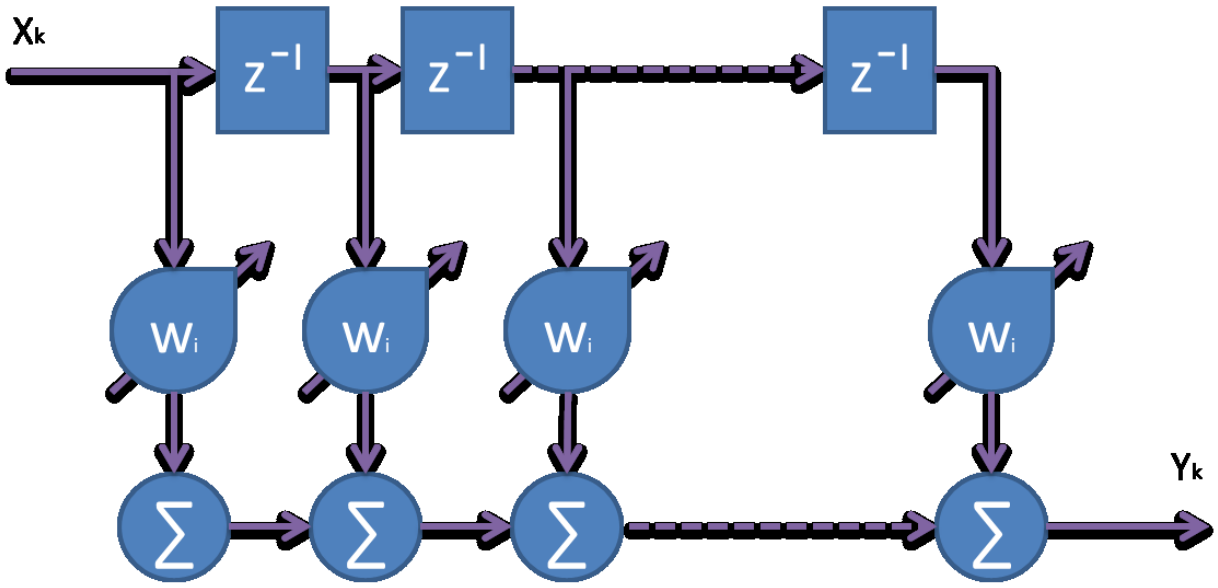


FIGURE 2-5 ADAPTIVE LINEAR COMBINER ARCHITECTURE

2.3 PERFORMANCE FUNCTION FOR MMSE:

It defines the criteria for the adaptive algorithm to follow for minimizing the error and to achieve convergence effectively. The Minimum mean square error (MMSE) estimator based adaptive algorithm is the most basic form of adaptive filter and its performance function can be elaborated by this simplified version of derivation [a.1].

The criteria for development of wiener filter are governed by the error in the output to that of our desired output.

$$e_k = d_k - y_k$$

As an adaptive linear combiner is being developed for the architectural implementation the output is the relation between the taps and the input vector.

$$e_k = d_k - X_k^T W = d_k - W^T X_k$$

Squaring the above equation to get square of error

$$e_k^2 = d_k^2 + W^T X_k X_k^T W - 2 d_k X_k^T W$$

Taking the expected value for the above equation for input we develop eq (1)

$$E[e_k^2] = E\left[d_k^2\right] + W^T E\left[X_k X_k^T\right] W - 2E\left[d_k X_k^T\right] W$$

The input autocorrelation matrix R is

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

The input and the desired output cross correlation vector P is

$$P = E[d_k X_k]$$

Substituting the above values in eq. (1)

$$MSE \triangleq \xi = E[e_k^2] = E[d_k^2] + W^T R W - 2P^T W$$

The gradient can be found by differentiating for **W,** and evaluating for =0

$$\nabla = 0 = 2RW^* - 2P$$

Rearranging and simplifying the above equation we can the optimum weight vector for gradient zero

$$W^* = R^{-1} P$$

Substituting above equation in eq. (1) we calculate the MMSE as follows

$$\xi_{min} = E[d_k^2] + W^{*T} R W^* - 2P^T W^*$$

$$\xi_{min} = E[d_k^2] + [R^{-1}P]^T R R^{-1} P - 2P^T R^{-1} P$$

$$\xi_{min} = E\left[d_k^2\right] - P^T R^{-1} P$$

The finalized equation for the MMSE is developed to minimize the error using optimum weights calculated above.

$$\xi_{min} = E\left[d_k^2\right] - P^T W^*$$

2.4 TEST SYSTEM DESIGN:

Our test system for all the upcoming chapters is developed on the basis of [b.13] and is described by the following Block diagram. The specifics of each subsystem block are also described in detail for the further evaluation comparison of the thesis literature and experimentation in the future work.

The input is a sampled sinusoid signal having about 167 samples per period creating a frequency of 60 HZ

The channel is modeled as a 3 tap fir filter with coefficients [0.3 0.9 0.3] to model the inter symbol interference (ISI) **Error! Reference source not found.**. ISI is a major challenge in wireless ransmissions due to multipath.

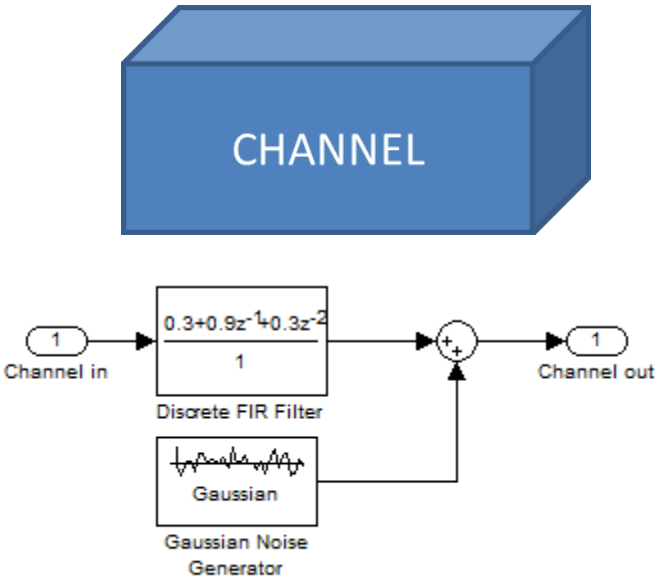


FIGURE 2-6 SIMULINK BLOCK DIAGRAM OF CHANNEL

The Gaussian noise generator is used to add a noise variance of 10 relevant to the input sinusoid of value 1**Error! Reference source not found.**.

The desired signal is created by introducing a delay in the input signal which is then fed to the equalizer.

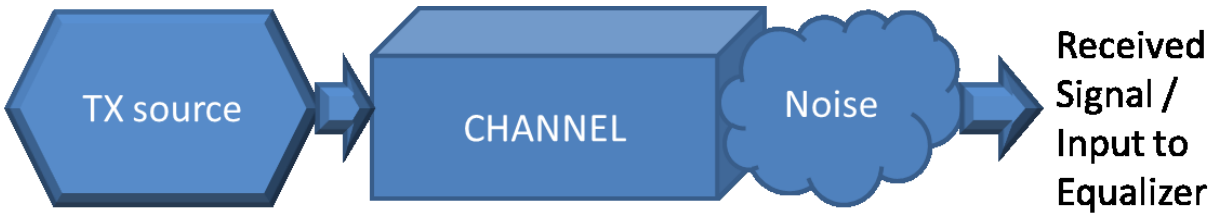


FIGURE 2-7 TEST SYSTEM MODEL

CHAPTER NO 3:

LEAST MEAN SQUARE ADAPTIVE ALGORITHM

CHAPTER 3 LEAST MEAN SQUARE ADAPTIVE ALGORITHM:

3.1 INTRODUCTION:

LMS is a worldwide well known algorithm for its easy and successful use along with the redundancy which it provides to the overall system. It is simple and it has computational ease as it does not require offline gradient estimation or repetitions of data. If we have the desired output at each step of iteration the LMS implementation is the best choice for the adaptive signal processing applications [b.19]. LMS algorithm uses the special case of gradient that is valid for implementation when we use an adaptive linear combiner [a.1].

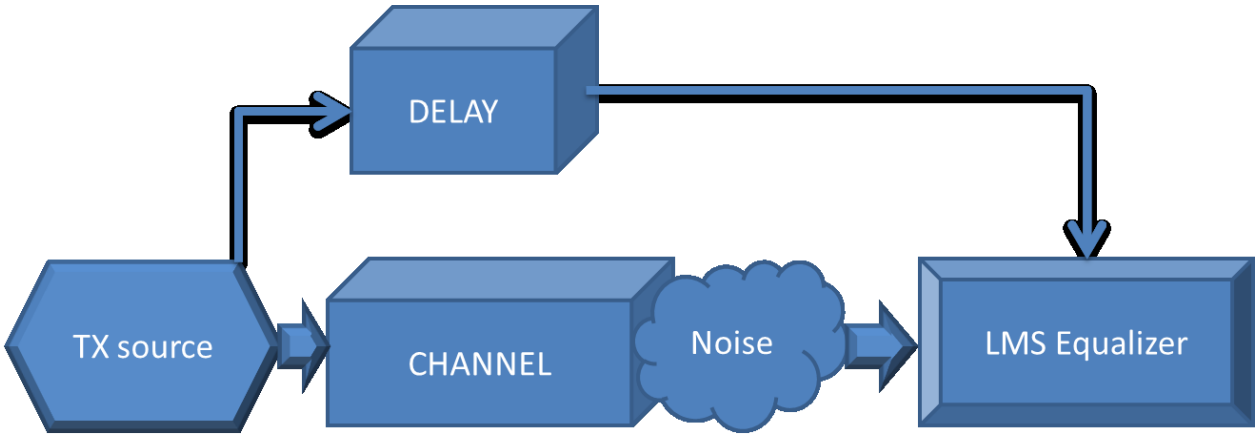


FIGURE 3-1 LMS TEST SYSTEM BLOCK DIAGRAM

3.2 DERIVATION:

If we assume an adaptive linear combiner of order p the LMS algorithm for it can be developed for the provided $x(n)$and $d(n)$[a.2],[c.2] .

Stating that:

p =filter order

μ = step size

The single input to the adaptive linear combiner can be defined as a multiple value with incrementing delays to represent their arrival at the combiner taps e.g. one shift at every iteration of the algorithm.

For n=0, 1, 2…..

$$x(n) = [x(n), x(n-1), \ldots., x(n-p+1)]^T$$

The error $e(n)$ between output and desired values can be calculated as their difference

$$e(n) = d(n) - \hat{h}^H(n)x(n)$$

$$\hat{h}(n+1) = \hat{h}(n) + \mu e^*(n)x(n)$$

$$0 < \mu < \frac{2}{\lambda_{max}}$$

Where λ_{max} is the largest Eigen value for the autocorrelation matrix R=E{x(n)x(n)}

$$0 < \mu < \frac{2}{tr[R]}$$

Where $tr[R]$ *is trace of autocorrelation matrix*

If an unsuitable value for step size is chosen the algorithm tends to be unstable and diverge.

The above described algorithm can also be perceived in the following for the output $y(n)$ of the filter and in terms of weight vector $w(n)$ for adjusting the taps in the adaptive linear combiner structure.

$$y(n) = w(n)^T x(n) = x^T(n)w(n)$$

$$e(n) = d(n) - y(n)$$

The weight vector estimated for each iteration can be represented as

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

3.3 IMPLEMENTATION:

LMS filter was implemented in Simulink [d.1] to test for the channel model and noise variance effects. Simulink filter block for LMS filter was used with our created channel model and the Gaussian noise addition.

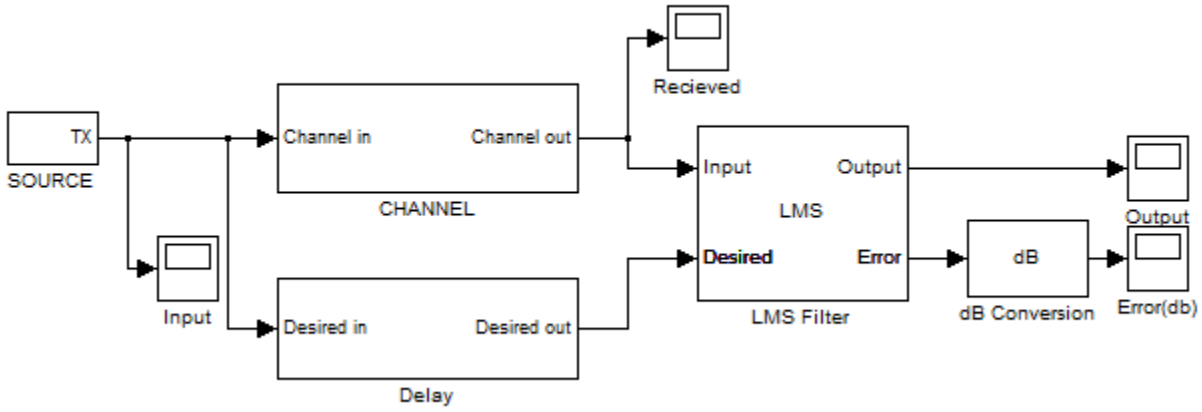


FIGURE 3-2 SIMULINK BLOCK DIAGRAM

3.4 EVALUATION:

The filter implemented was evaluated as a channel equalizer trying to estimate the inverse model of the channel. The capability and effectiveness for the LMS filter was assessed by the convergence speed and convergence stability. The LMS filter implemented has length 7 and the step size used is 0.06**Error! Reference source not found.**.

3.5 RESULTS:

The x-axis of all graphs represent the number of samples having actual range between 0 – 2000. This value is scaled down by 10000, so the range on the axis is between 0 – 0.2. The y – axis represents true values.

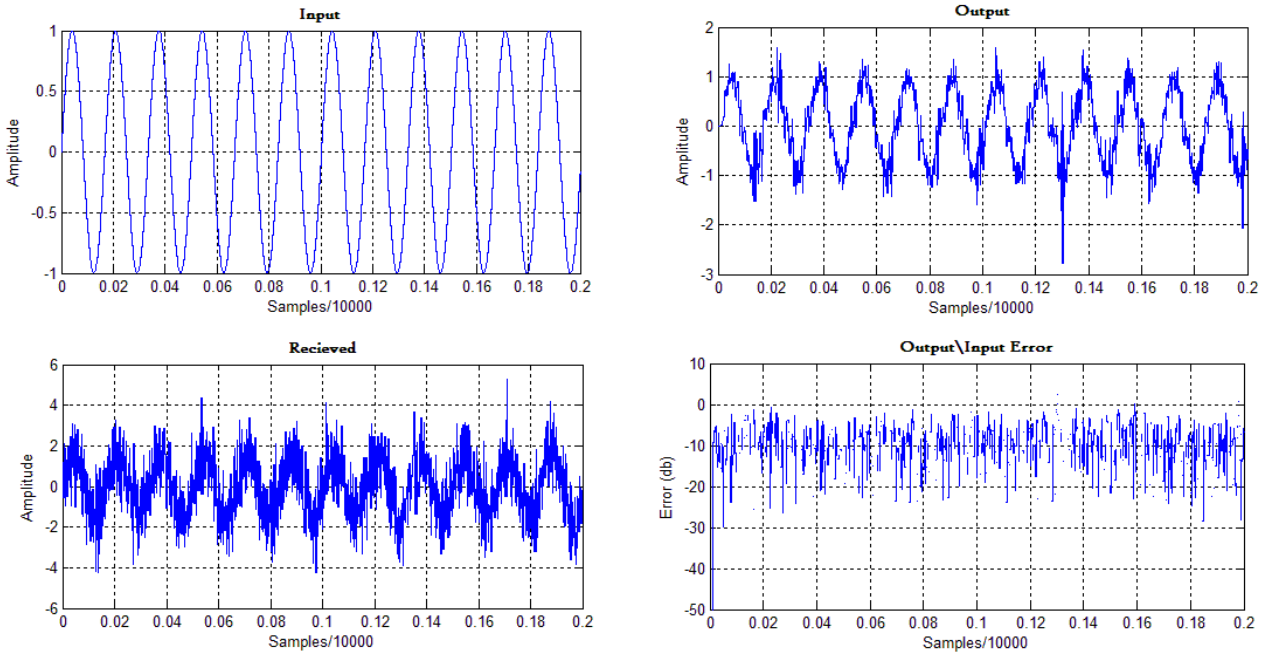


FIGURE 3-3 RESULTS FOR LMS

3.6 CONCLUSION:

The results for the implemented and evaluated LMS filter show that the response of the adaptive filter is satisfactory but if we choose wrong step size value the filter output response diverges. We see the output is diverging momentarily at around 0.13 in the above graph. This is due to the behavior of the noise at that certain moment which makes it impossible for the LMS algorithm to track the output as the additive property of noise is too large to remove for the small step size used to create stability in the filtered output so the filter response diverges momentarily but is soon converged as more samples of input are processed and an equalized response is established. This concludes that the step size vale has to be carefully chosen to perform significant filtering for applications and to avoid instability in system.

CHAPTER NO 4:

VARIABLE STEP LEAST MEAN SQUARE ADAPTIVE ALGORITHM

CHAPTER 4 VARIABLE STEP LEAST MEAN SQUARE ADAPTIVE ALGORITHM:

4.1 INTRODUCTION:

A lot of work has been done and referred to as to develop a variable step size LMS feedback mechanism to overcome certain LMS issues [b.9][b.12][b.18]. The algorithm that is used in this work is the arc-tan based variable step algorithm **Error! Reference source not found.**. It has been iscussed in the derivation portion and is simpler and effective for its complexity and structure. Such algorithms are a technique to add more redundancy and resistivity in the adaptive system.

The principle of variable step LMS algorithm is to overcome the contradiction between convergence speed and the convergence accuracy present in the fixed step LMS algorithm. In the initial stages a large step-size vale is used as to have a faster convergence achievement whereas in the later phases the step size value is reduced to achieve the lesser steady state error **Error! Reference source not ound.**[b.12].

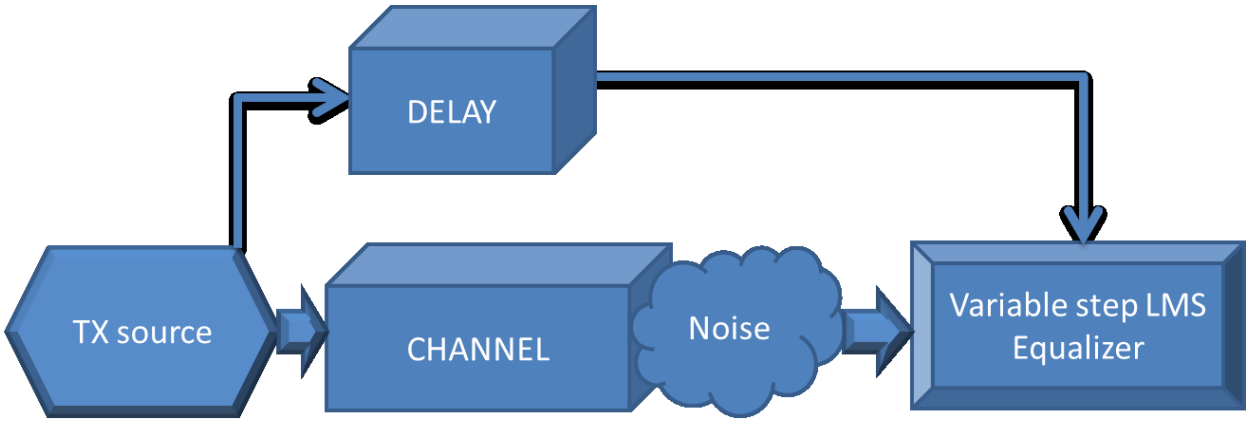


FIGURE 4-1 BLOCK DIAGRAM FOR THE VARIABLE STEP LMS IMPLEMENTATION

4.2 DERIVATION:

The output for the variable step LMS algorithm is developed by the adaptive linear combiner where the weights to the filter taps and the inputs are processed.

$$y(n) = w(n)^T x(n) = x^T(n) w(n)$$

Similarly the error to the variable step function is simple difference between the desired output and the estimated output. This can be described simply as the difference equation as follows

$$e(n) = d(n) - y(n)$$

Weights are updated by using the gradient descent function to converge the error to minimum value computationally possible.

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

By analysis of the convergence of an LMS algorithm we can easily deduce that the error has a direct relation with the step size. To achieve faster convergence we can increase the step size when error is greater and similarly vice versa we can have stability in algorithm when error is small by having a small step size. Therefore monotone as well as smooth curve characteristic for the function can be easily concluded.

The arc-tangent based LMS step size algorithm **Error! Reference source not found.** has the same haracteristics by the nonlinear function of arc-tan. It is evaluated that the arctangent curve is consistent to that of the variation for our step factor.

The atan-LMS algorithm is represented by the following equations.

$$\mu(n) = \text{atan}(e(n))$$

We can have better control over the algorithm curve characteristics by introducing some control factors

$$\mu(n) = \beta\text{atan}(\alpha e(n)^{\gamma})$$

These control factors give us better control for the rate of convergence of step function by introducing different curve characteristics and rates.

4.3 IMPLEMENTATION:

Variable step LMS is implemented using Simulink [d.1] block set of LMS filter with the step size parameter set to be configured externally by port instead of dialogue box. The equation for Step size configuration is implemented using discrete function blocks from the Simulink math library for arctangent, multipliers and exponent functions.

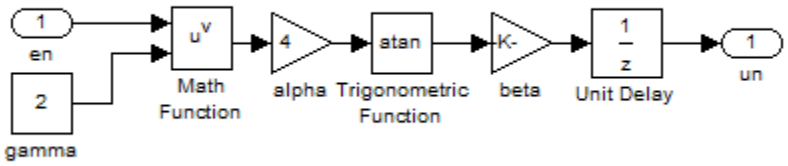


FIGURE 4-2 SIMULINK BASED ARCHITECTURE FOR VARIABLE STEP SIZE FUNCTION

The development of this function block was done to ease and discretize the sample based implementation for this variable step technique algorithm [b.1][b.2]. The test system was applied and the variable step LMS based adaptive implementation was done using Simulink.

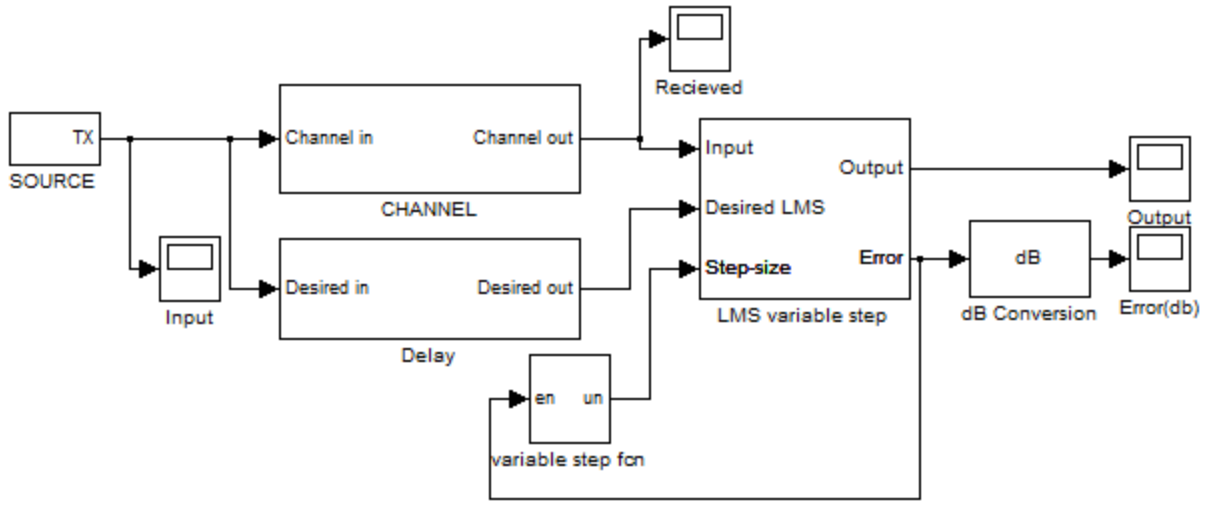


FIGURE 4-3 SIMULINK BASED BLOCK IMPLEMENTATION

4.4 EVALUATION:

Variable step LMS architecture including the variable step function created by the Simulink block set was evaluated for the design of the channel equalizer and relevant results were generated. The length for the LMS filter used is same as that for the previous fixed step implementation. The initial condition for the variable step function is taken as 0.06. The values for α, β & γ are 4, 0.04 & 2 respectively.

4.5 RESULTS:

The x-axis of all graphs represent the number of samples having actual range between 0 – 2000. This value is scaled down by 10000, so the range on the axis is between 0 – 0.2. The y – axis represents true values.

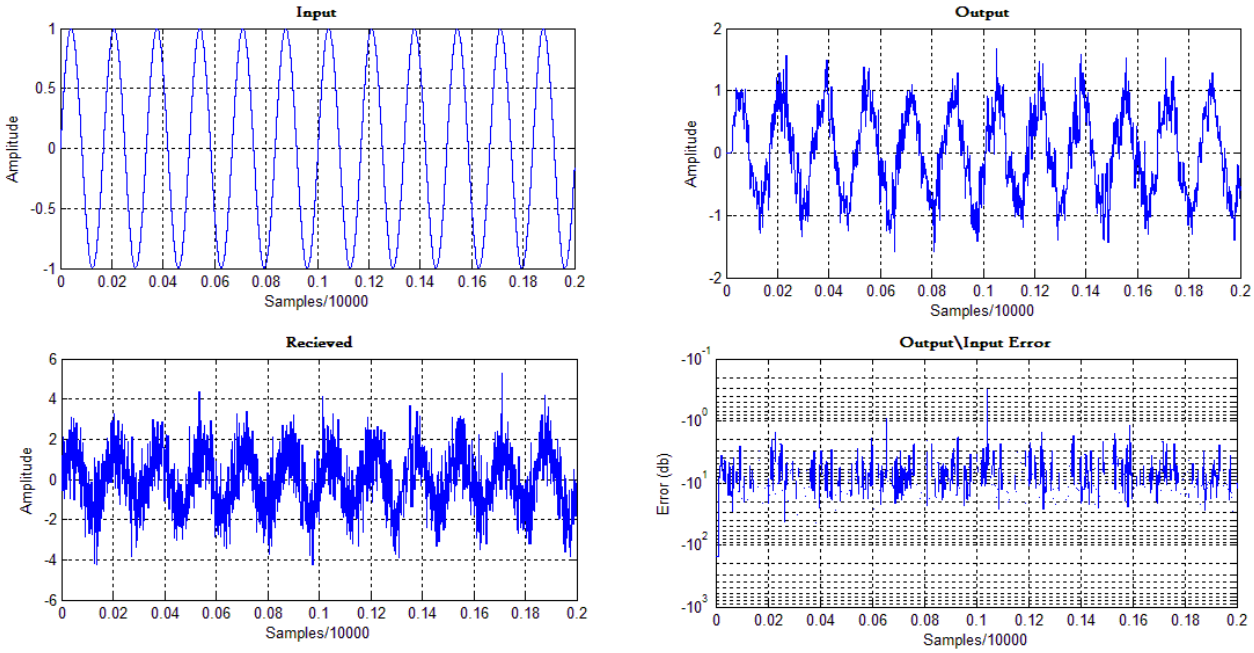


FIGURE 4-4 RESULTS FOR ATAN LMS

4.6 CONCLUSION:

The results shown help us conclude the fact that the variable step reduces the overall tracking error to a very little extent due to the high Gaussian variance input but it has a significant impact in introducing stability in the output for an ordinary LMS filter as the step size available to the LMS algorithm is adaptively varying with respect to the nonlinear tan function available to the variable step size function. This eliminates the instability so that the overall filter has a convergent behavior.

CHAPTER NO 5:

SINUSOID TRACKING ADAPTIVE ALGORITHM

CHAPTER 5 SINUSOID TRACKING ADAPTIVE ALGORITHM:

5.1 INTRODUCTION:

The sine tracking algorithm proposed by Zirani [b.4][b.5] is unique in its performance for nonlinear applications like those applied in the field of Biomedical engineering [b.15][b.16][b.17][b.21], however the algorithm is relatively new and has since been applied to only on a few specific applications, but it's simple structure is relatively easy to understand and comprehend which makes this algorithm the prime candidate for digital signal processing based applications. Figure [5-2] shows the unique architecture for STA. As easily observable in the figure the algorithm tries to minimize the error that is been fed into the algorithm and by the help of step sizes or tuning factors μ1,μ2 and μ3 the algorithms convergence rate decreases or increases affecting the immunity for the noise it can handle for certain system [b.5][b.11][b.10]. The algorithm tracks the amplitude, frequency and phase of the input signal.

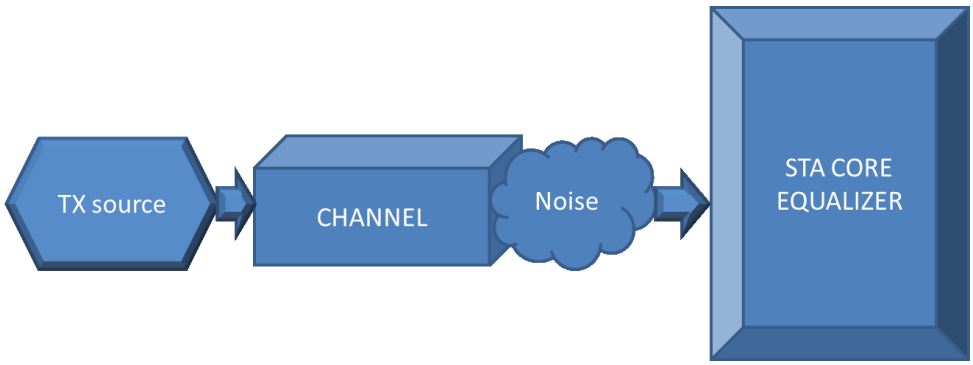


FIGURE 5-1 BLOCKS DIAGRAM FOR STA IMPLEMENTATION

5.2 DERIVATION:

General mathematical form of input

$$u(t) = \sum_{i=0}^{\infty} A_i \sin \emptyset_i + n(t)$$

To extract the desired sinusoidal component its amplitude, frequency and phase have to be tracked.

$$y(t) = A(t) \sin\left(\int_{\cdot}^{t} \omega\,(\tau) d\tau + \delta(t) \right)$$

The amplitude is estimated and tracked by the following nonlinear differential equation.

$$\frac{d\hat{A}(t)}{dt} = 2\mu_1 e(t) \sin \widehat{\emptyset}(t)$$

Similarly the frequency estimate is evaluated as follows

$$\frac{d\hat{\omega}(t)}{dt} = 2\mu_2 e(t)\hat{A}(t)\cos\hat{\emptyset}(t)$$

The phase of the input is tracked to be used for proper frequency and amplitude estimation.

$$\frac{d\hat{\emptyset}(t)}{dt} = \hat{\omega}(t) + \mu_3\frac{d\hat{\omega}(t)}{dt}$$

The error between the input and output is calculated by subtraction to be fed back in the filter for the next step error minimization.

$$e(t) = u(t) - \hat{A}(t)\sin\hat{\emptyset}(t)$$

A discrete form of equations for the above discussed algorithm can be formed using the T_s as the step function value for evaluating the discrete system correctly.

The equations [b.6] below are representing the discrete tracking of the algorithm for discrete input $u[n]$ and discrete output $y[n]$.

$$A[n+1] = A[n] + 2T_s\mu_1 e[n]\, sin\emptyset[n]$$

$$\omega[n+1] = \omega[n] + 2T_s\mu_2 e[n]A[n]\, cos\emptyset[n]$$

$$\emptyset[n+1] = \emptyset[n] + T_s\omega[n] + 2T_s\mu_2\mu_3 e[n]A[n]\, cos\emptyset[n]$$

$$y[n] = A[n]sin\emptyset[n]$$

$$e[n] = u[n] - y[n]$$

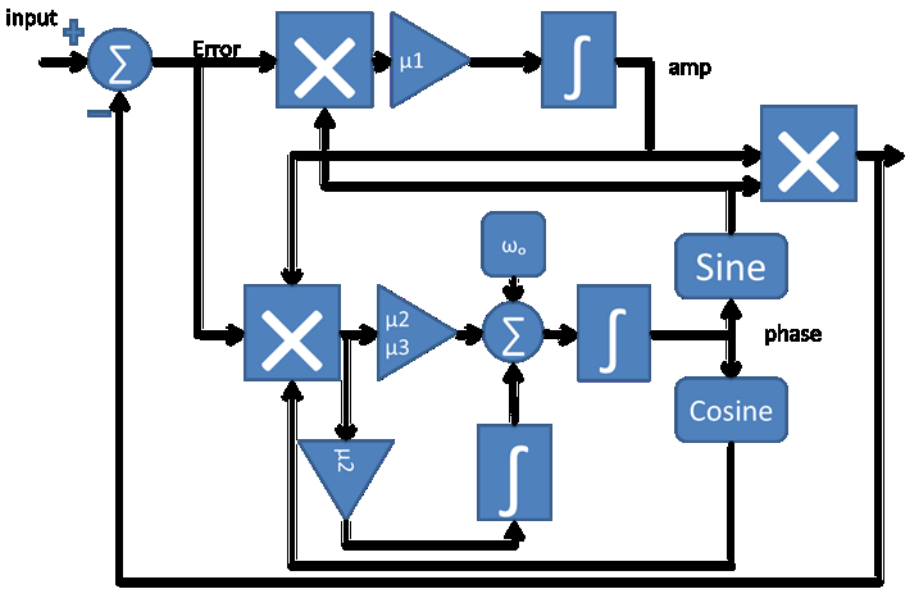


FIGURE 5-2 STA CORE ARCHITECTURE

5.3 IMPLEMENTATION:

A discrete form of the algorithm can easily be implemented on digital systems for its DSP performance. The relatively simple discrete equations shown above make the discrete implementation easy. A Simulink based implementation is performed for the above mentioned STA. This implementation is eased by the development of the block architecture.

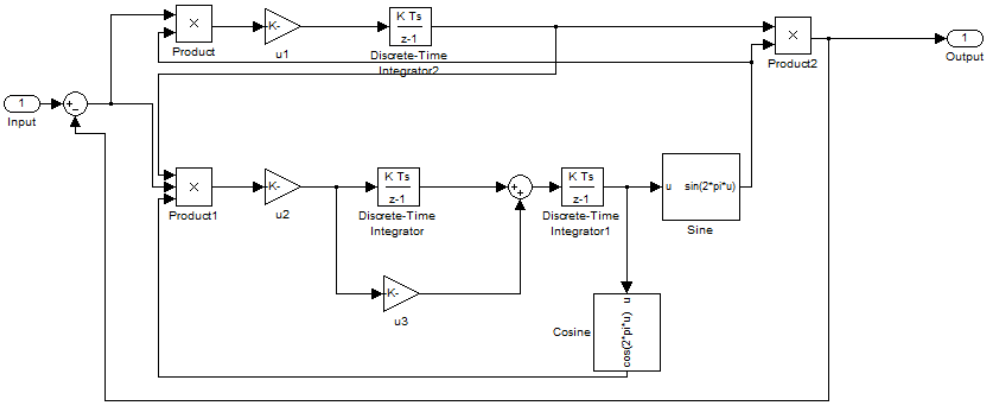


FIGURE 5-3 SIMULINK BASED STA IMPLEMENTATION

5.4 EVALUATION:

The discrete algorithm implemented is evaluated by a sine wave given at the input is tracked and various results are shown outlining the tracking performances for the STA. These results show the convergence capability for the algorithm along with its frequency and phase tracking performance[b.5].

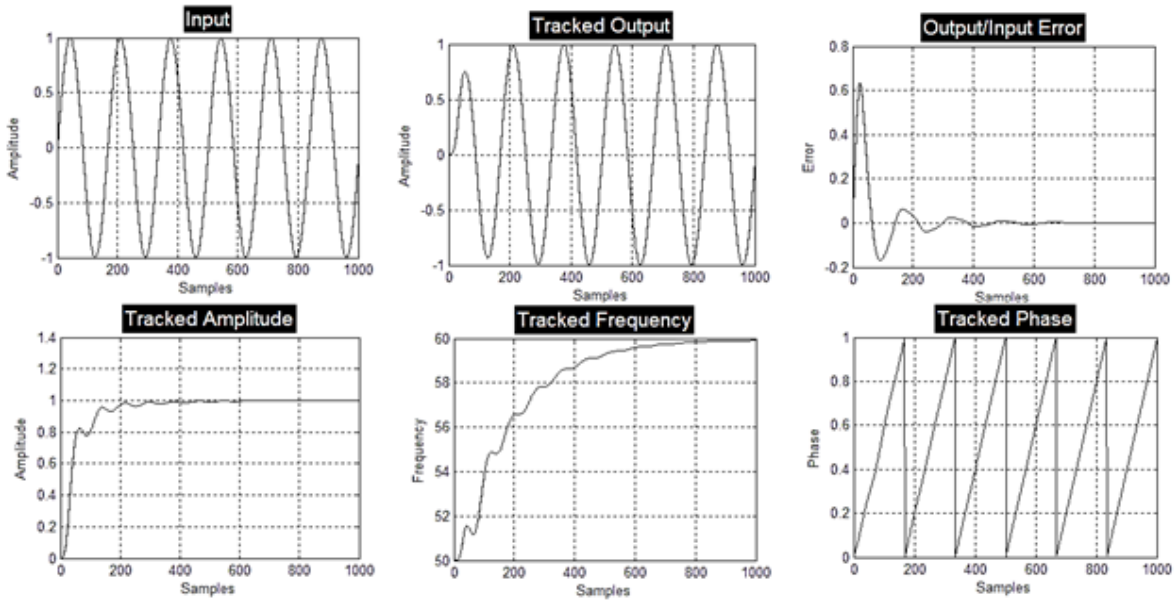


FIGURE 5-4 VARIOUS TRACKING RESULTS OF STA

A blind channel equalizer is implemented using the STA core structure and the error between input and output is calculated by the help of the reference signal d(n) and output is tracked along with the various results are shown. The desired frequency component that is to be tracked is the 60 Hz desired sinusoid.

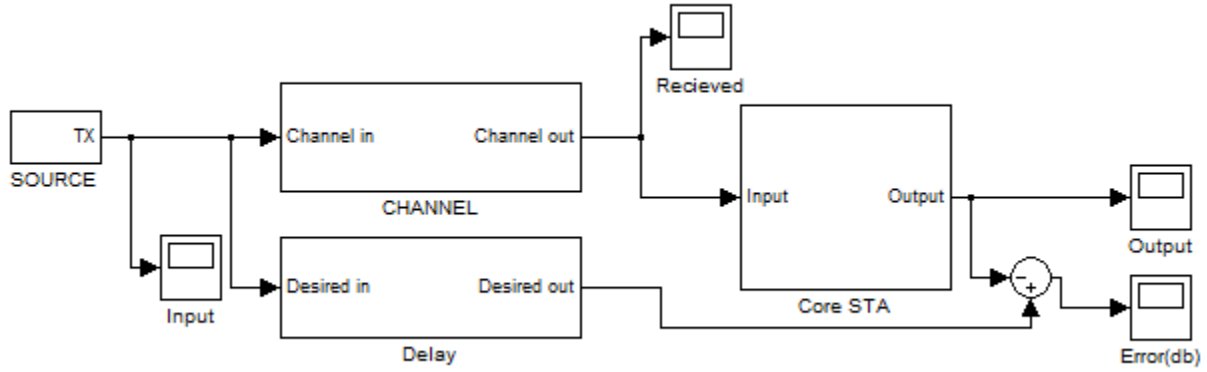


FIGURE 5-5 STA BASED EQUALIZER IMPLEMENTATION

5.5 RESULTS:

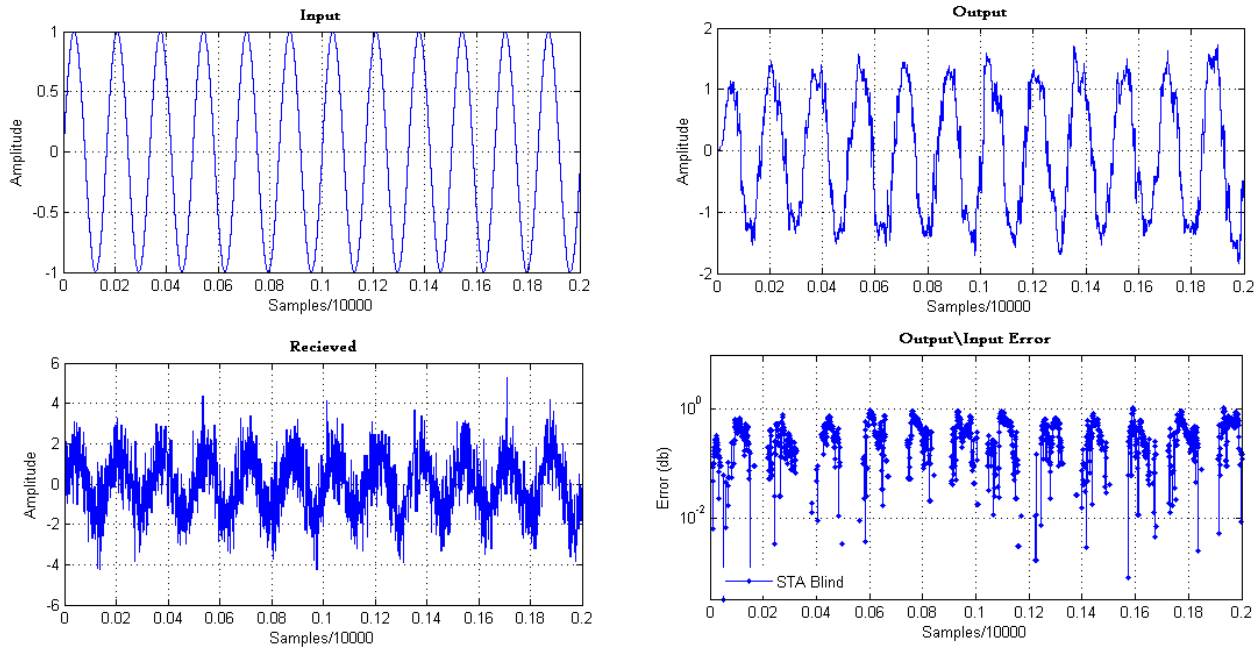


FIGURE 5-6 STA EQUALIZER RESULTS

The x-axis of all graphs represent the number of samples having actual range between 0 – 2000. This value is scaled down by 10000, so the range on the axis is between 0 – 0.2. The y – axis represents true values.

5.6 CONCLUSION:

The results help us conclude that the STA has the capability to track the sinusoid based component of the desired frequency and the performance is relevantly close to that of the LMS algorithm. The stability of the algorithm is good due to the convergent behavior and the tracking error has an oscillatory behavior. The error plot shows that the algorithm tracks the phase and frequency with stable behavior to have overlapping response for error. The tracking error varies between 0.01 and 1. The output shows us the tracked sinusoid component which is visually distorted a little relevant to the received signal.

CHAPTER NO 6:

FREQUENCY DIRECTED SINUSOID TRACKING ADAPTIVE ALGORITHM

CHAPTER 6 FREQUENCY DIRECTED SINUSOID TRACKING ADAPTIVE ALGORITHM:

6.1 INTRODUCTION:

The STA discussed in the previous section can easily be modified to have directions form the desired frequency[b.7][b.8][b.9][b.21][b.22]. This is done to enhance the convergence of the STA as well as to decrease the computational complexity to that of the pure STA as it does not have to estimate and track the frequency of the signal, it is available as an input. The result is an algorithm that is less complex and fast convergent to the directed frequency and thus can be used freely in our designs.

6.2 DERIVATION:

Mathematically the algorithm can be easily developed by substituting the STA derivation equations bit the directed frequency in place of the tracked\estimated frequency [b.8]. This results in simplified equations as follows.

The directed frequency amplitude is tracked and estimated as

$$\frac{d\hat{A}(t)}{dt} = \mu e(t) \sin\left(\int_{\cdot}^{t} \omega(\tau) d\tau(t) + \hat{\delta}(t)\right)$$

The initial phase component for the directed frequency is evaluated as

$$\frac{d\hat{\delta}(t)}{dt} = ve(t)\, \hat{A}(t) \cos\left(\int_{\cdot}^{t} \omega(\tau) d\tau(t) + \hat{\delta}(t)\right)$$

The evaluated output is developed by using the estimated amplitude and phase for the input signal.

$$y(t) = \hat{A}(t) \sin\left(\int_{\cdot}^{t} \omega(\tau) d\tau(t) + \hat{\delta}(t)\right)$$

The error criterion is same as that of the pure STA

$$e(t) = u(t) - y(t)$$

The total phase component is

$$\widehat{\emptyset}(t) = \int_{\cdot}^{t} \omega(\tau) d\tau(t) + \hat{\delta}(t)$$

The discrete form of above equations is also easy to develop like the pure STA discrete development taking the step-size T_s

The estimate equations for the amplitude initial phase and the total phase for the given directed frequency $\omega[n]$ are stated as.

$$A[n+1] = A[n] + T_s \mu e[n]\, sin\emptyset[n]$$

$$\delta[n+1] = \delta[n] + T_s ve[n]A[n]\, cos\emptyset[n]$$

$$\emptyset[n+1] = \emptyset[n] + T_s \omega[n] + T_s ve[n]A[n]\, cos\emptyset[n]$$

$$y[n] = A[n] sin\emptyset[n]$$

$$e[n] = u[n] - y[n]$$

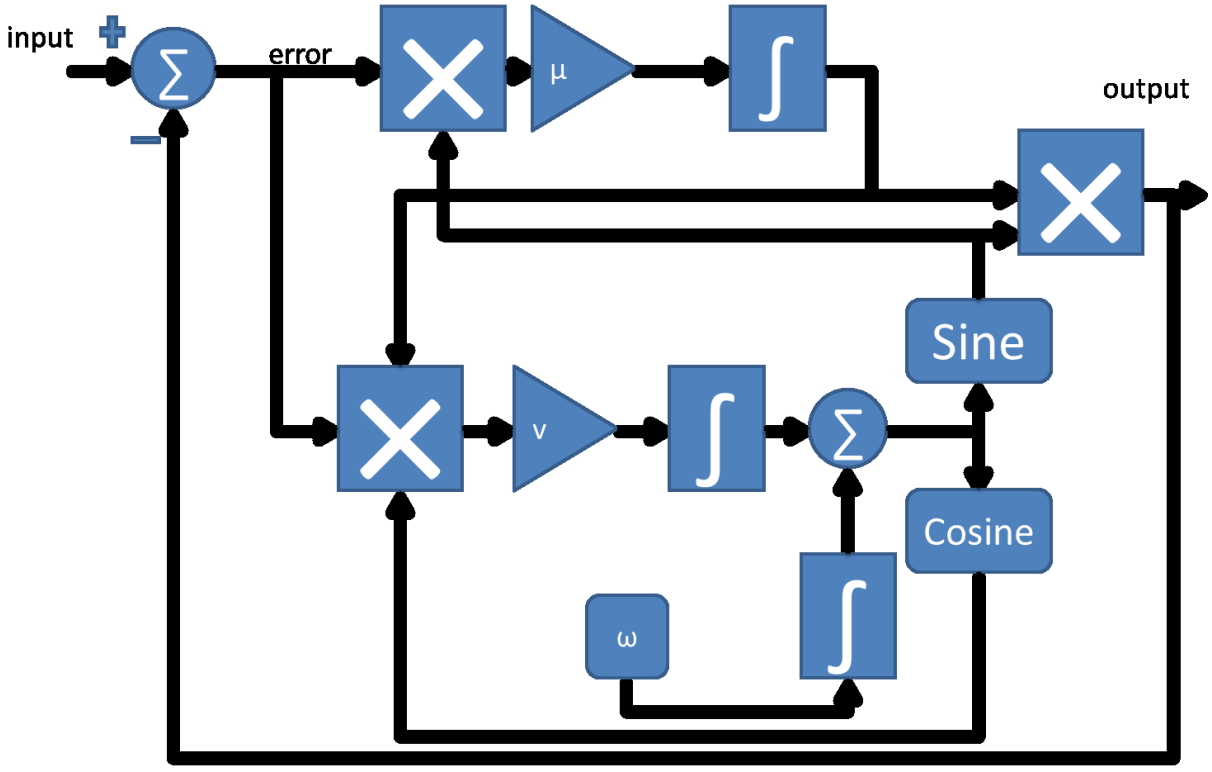


FIGURE 6-1 BLOCK DIAGRAM FOR THE STA FREQUENCY DIRECTED ARCHITECTURE

6.3 IMPLEMENTATION:

The architecture which is implemented on Simulink design environment is developed form the discrete equations discussed above. The simplified architecture is easier to implement than STA and is consistent purely of discrete support Simulink blocks form Simulink block set library.

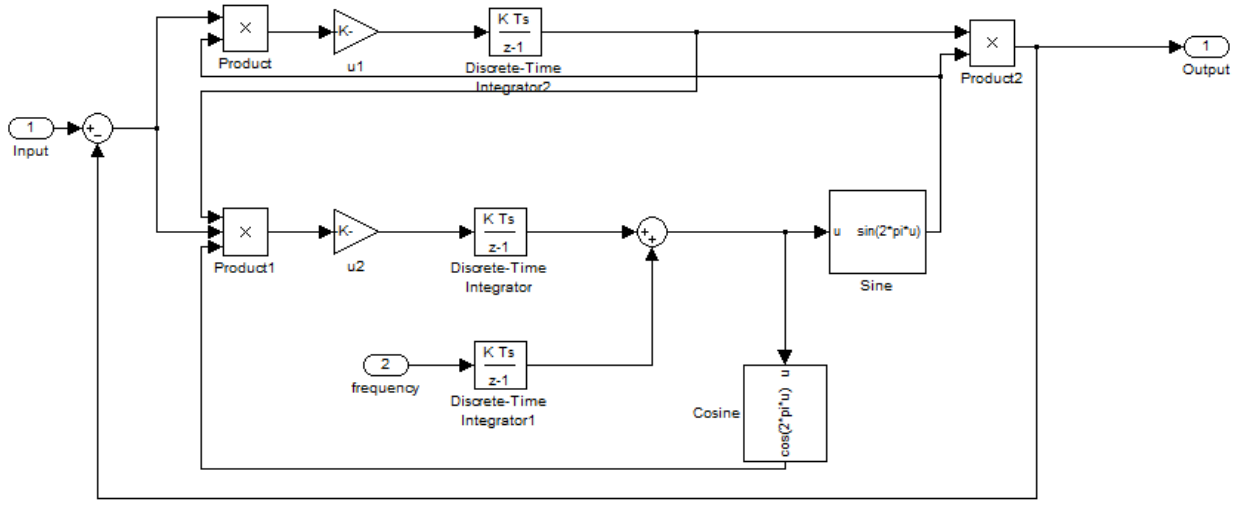


FIGURE 6-2 STA FREQUENCY DIRECTED ALGORITHM IMPLEMENTATION WITH SIMULINK BLOCKS

6.4 EVALUATION:

The convergence capability for sinusoid input tracking is evaluated with the frequency directed case and frequency direction along with amplitude limiting is applied to form and evaluate the equalizer design for the post processed frequency directed STA.

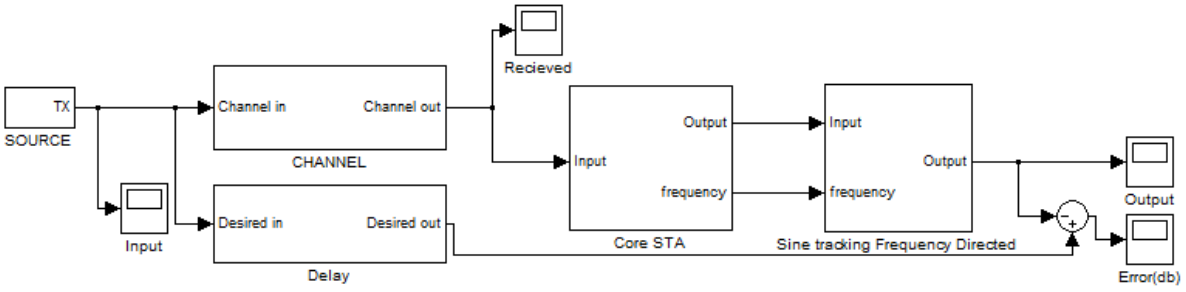


FIGURE 6-3 STA EQUALIZER POST PROCESSING WITH STA FD

6.5 RESULTS:

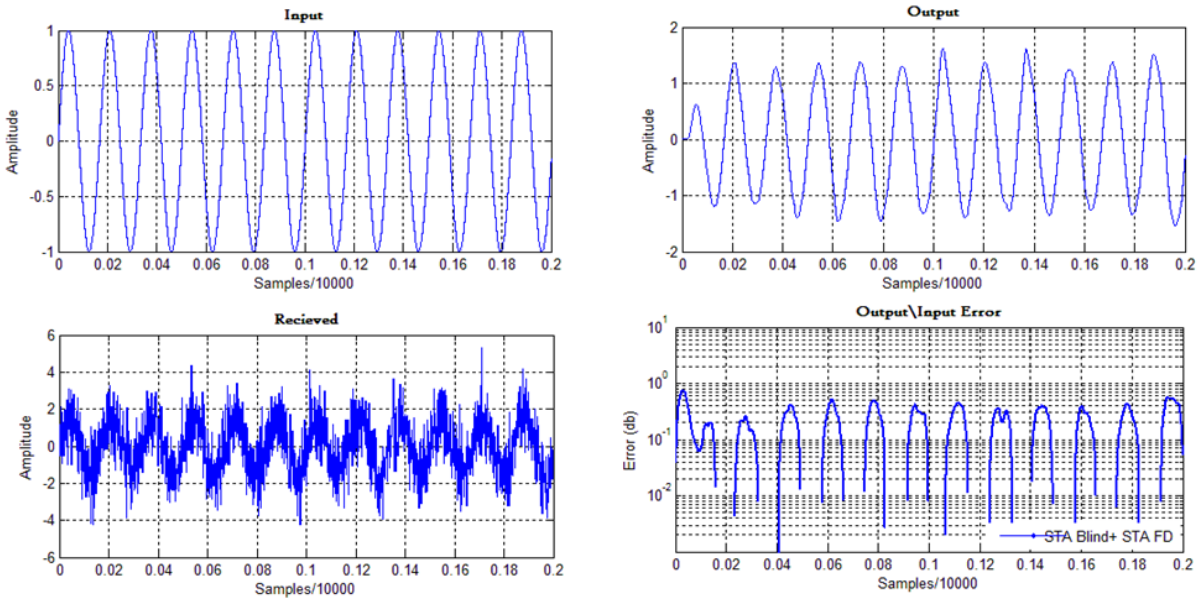


FIGURE 6-4 RESULTS FOR STA+FD

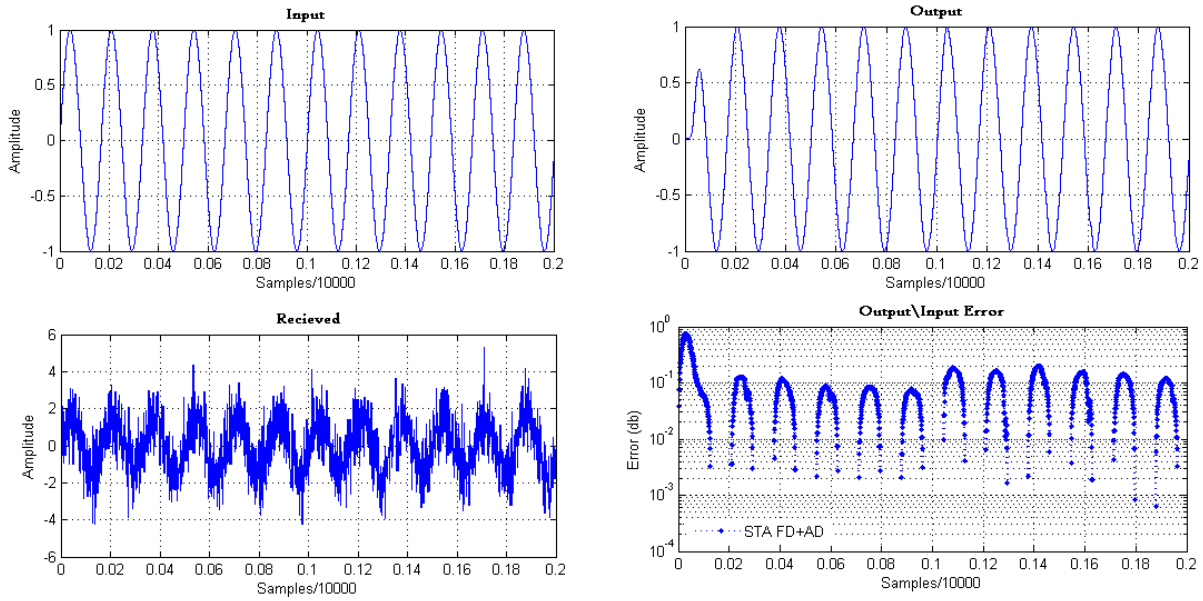


FIGURE 6-5 RESULTS FOR STA+FD+AMPLITUDE LIMITER

The x-axis of all graphs represent the number of samples having actual range between 0 – 2000. This value is scaled down by 10000, so the range on the axis is between 0 – 0.2. The y – axis represents true values.

6.6 CONCLUSION:

The results shown for the blind STA aided with the STA frequency directed for the cases of free running amplitude and the amplitude limiting mode show us the behavior of the algorithms and their equalizers. We observe that when we post processed the core STA output with a frequency directed STA core we are able to successfully eliminate the Gaussian interference but the amplitude stability has not been achieved with perfection. By observing the results it is concluded that the amplitude is above the required and needs to be limited. The case for the amplitude limiter based design makes the output same as the input but the phase tracking is not perfected which leaves some error in the results. The results for the STA+STA FD with amplitude limiter show that the error is lower than the ones achieved for the case of without amplitude limiter.

CHAPTER NO 7 :

COMPARATIVE ANALYSIS

CHAPTER 7 COMPARATIVE ANALYSIS:

7.1 LMS VS. VARIABLE STEP LMS:

LMS and the arc tan directed variable step LMS is compared to verify [b.14] as well as to create a benchmark for the performance comparison for the later STA based comparisons to develop a level of performance improvement. As discussed earlier in their relevant sections both are performance capable to our equalizer design test, but the variable step size LMS has certain design constraints which enhance its performance for the design.

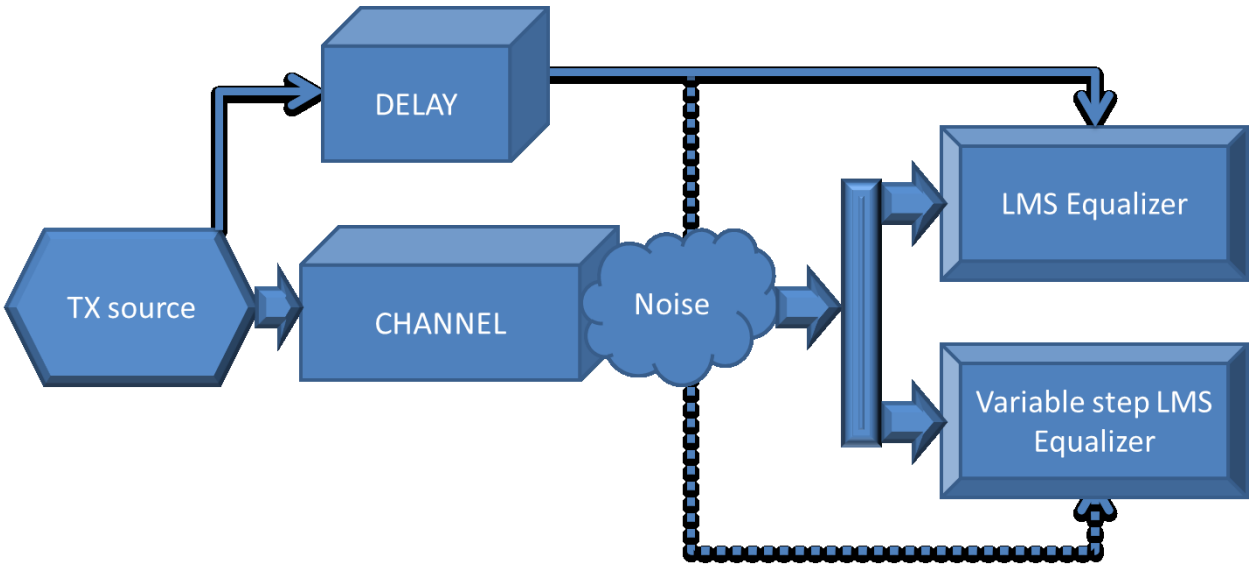


FIGURE 7-1 BLOCK DIAGRAM FOR COMPARISON

7.1.1 IMPLEMENTATION:

Both algorithms were driven by the same reception signal at the receiver frontend and were implemented such that both perform in parallel design status to have a fair comparison between them.

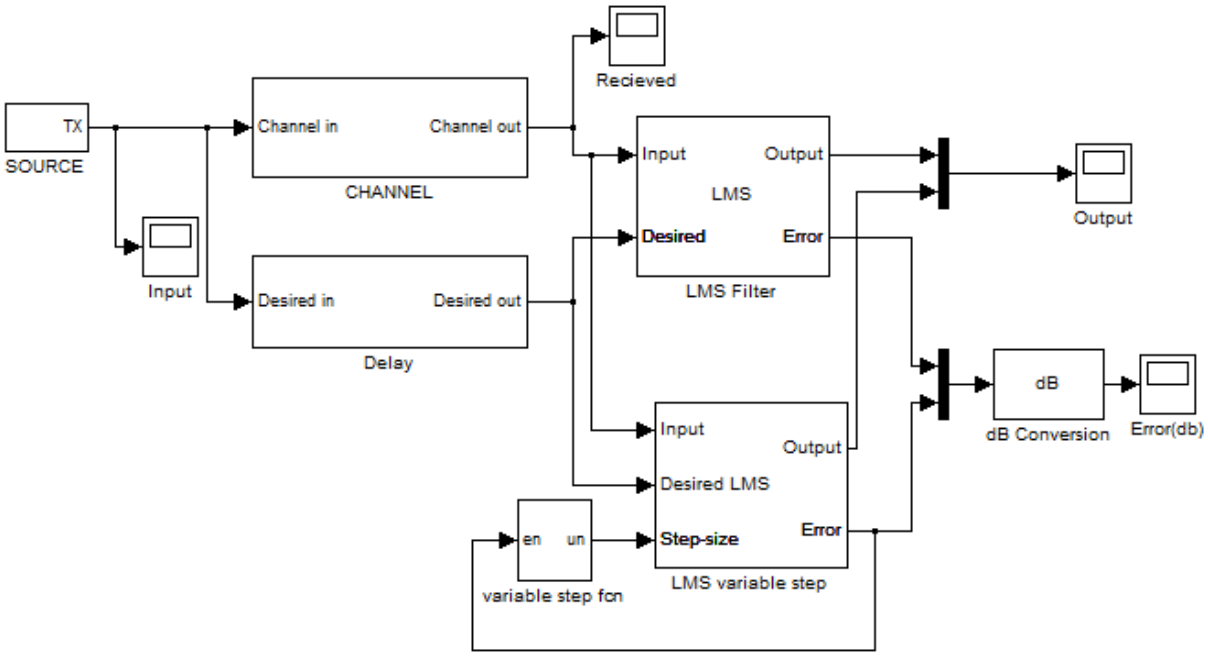


FIGURE 7-2 SIMULINK MODEL

7.1.2 RESULTS:

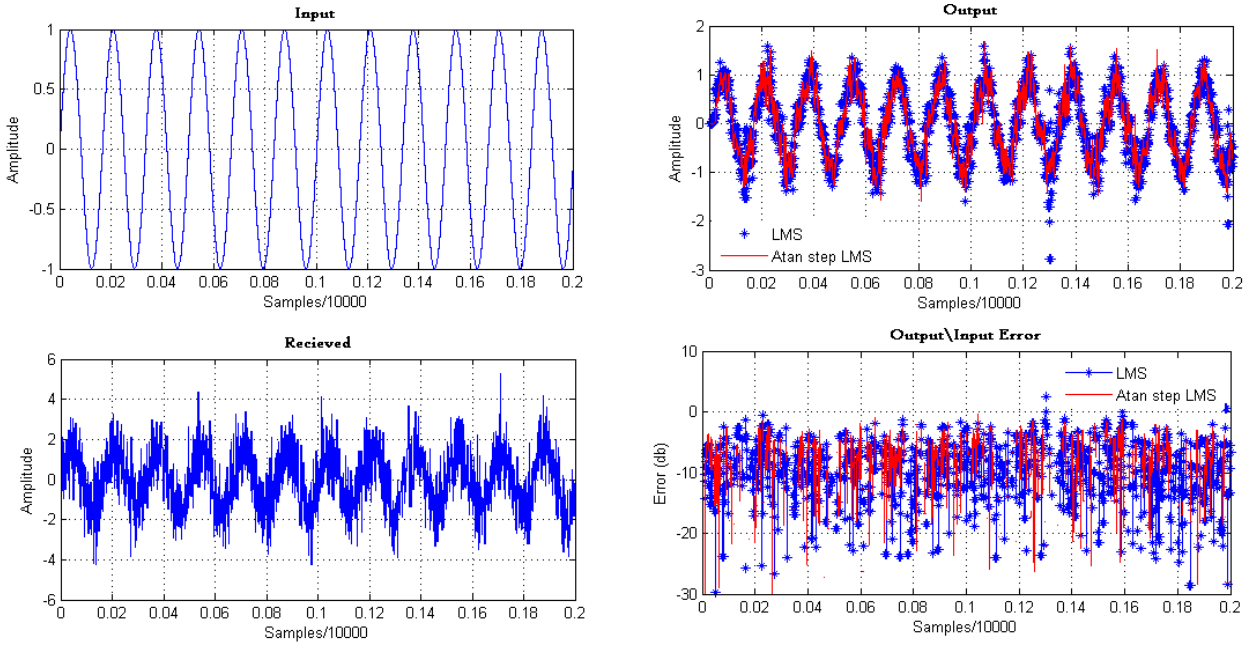


FIGURE 7-3 COMPARISON RESULTS

The x-axis of all graphs represent the number of samples having actual range between 0 – 2000. This value is scaled down by 10000, so the range on the axis is between 0 – 0.2. The y – axis represents true values.

7.1.3 CONCLUSION:

By comparing both filters for the same test system the evaluated results as shown in the above show that the arc tan variable step LMS is more stable in its convergence period compared to the fixed step LMS. This increase in stability gives us a convergent system and the target output is more optimized.

7.2 STA VS. STA WITH FREQUENCY DIRECTED STA:

The equalizer designs based on both STA core and STA frequency directed were compared to show the related enhancement in tracking the input and to form a benchmark for the combined algorithm to introduce redundancy as well as to introduce advance features advantages.

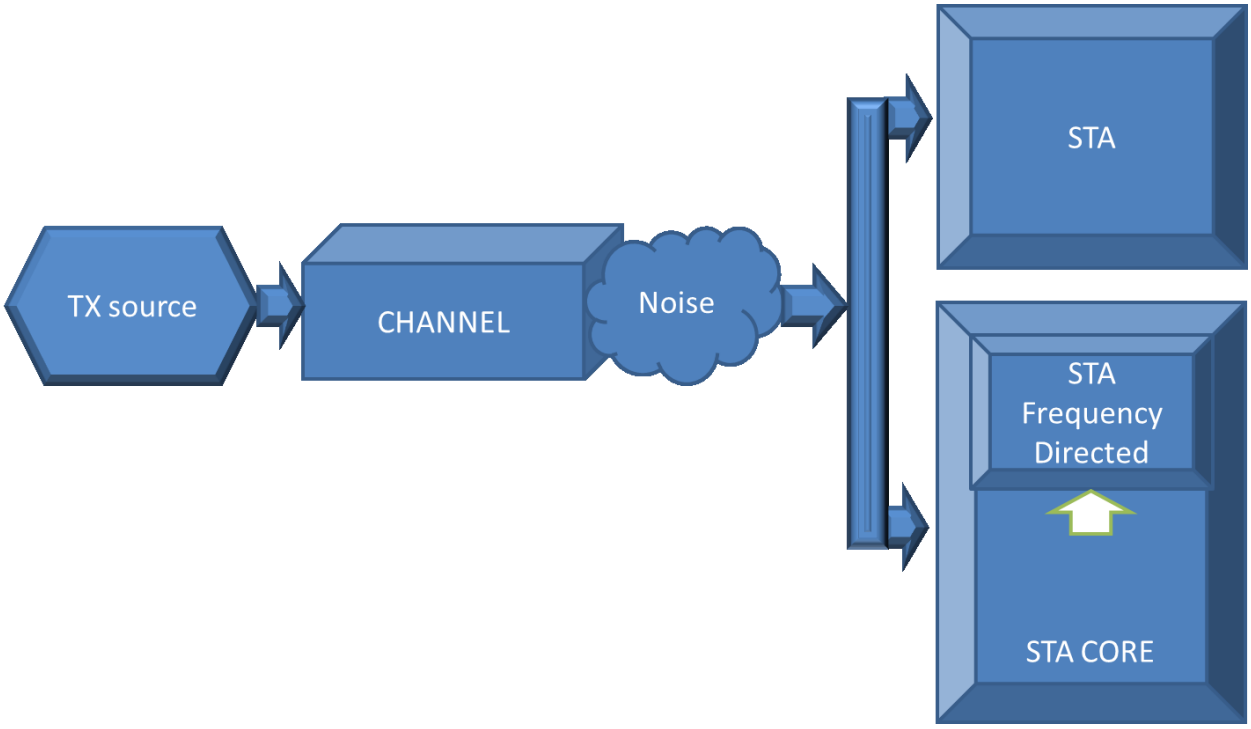


FIGURE 7-4 COMPARISON BLOCK MODEL

7.2.1 IMPLEMENTATION:

Both algorithms were driven by the same reception signal at the receiver frontend and were implemented such that both perform in parallel design status to have a fair comparison between them.

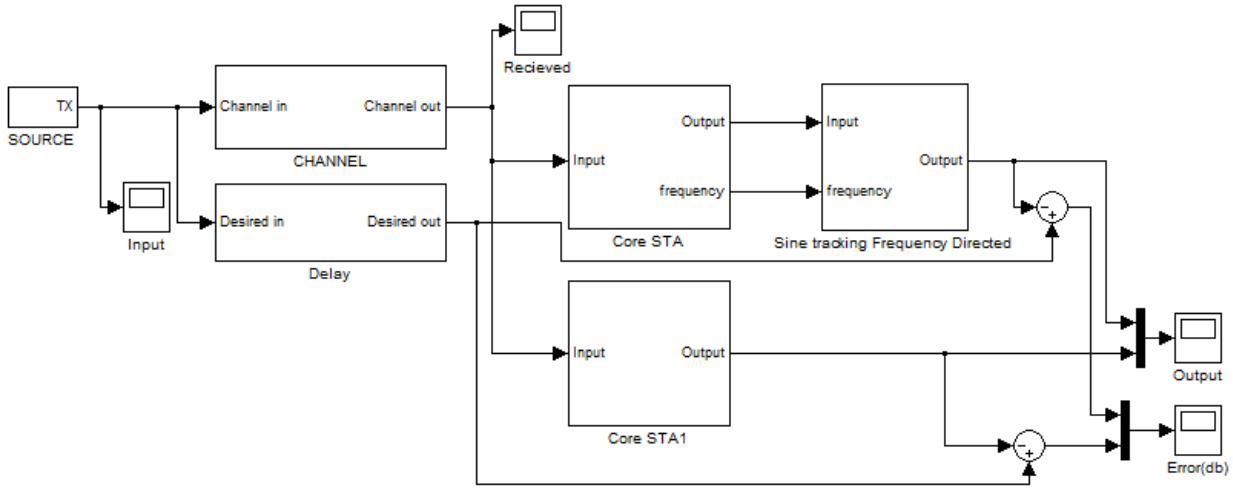


FIGURE 7-5 COMPARISON SIMULINK MODEL

7.2.2 RESULTS:

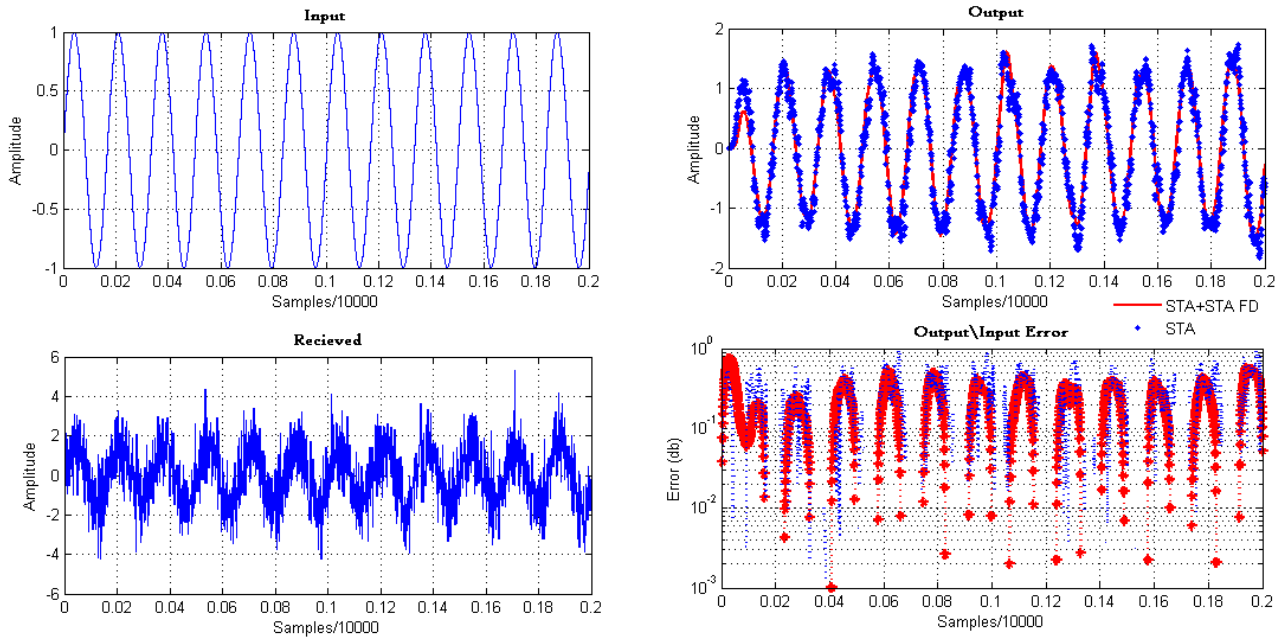


FIGURE 7-6 COMPARISON RESULTS

The x-axis of all graphs represent the number of samples having actual range between 0 – 2000. This value is scaled down by 10000, so the range on the axis is between 0 – 0.2. The y – axis represents true values.

7.2.3 CONCLUSION:

When the results for the STA core and the STA core with post processed frequency dictated STA are evaluated together the fact for the noise removal of the post processing technique is revealed and the outputs shown are a clear indication of this. The error plot shown on logarithmic scale also confirms the post processing advantage to that of using simple core of STA. The unwanted error is reduced more in this case.

7.3 VARIABLE STEP LMS VS. STA FREQUENCY DIRECTED:

The variable step LMS equalizer was compared to the STA based Equalizer and the performance effects are shown in the form of error plots and comparison graphs. The Variable step LMS is chosen for the comparison instead of fixed step LMS because in the above comparison it was concluded that the variable step algorithm is relatively stable.

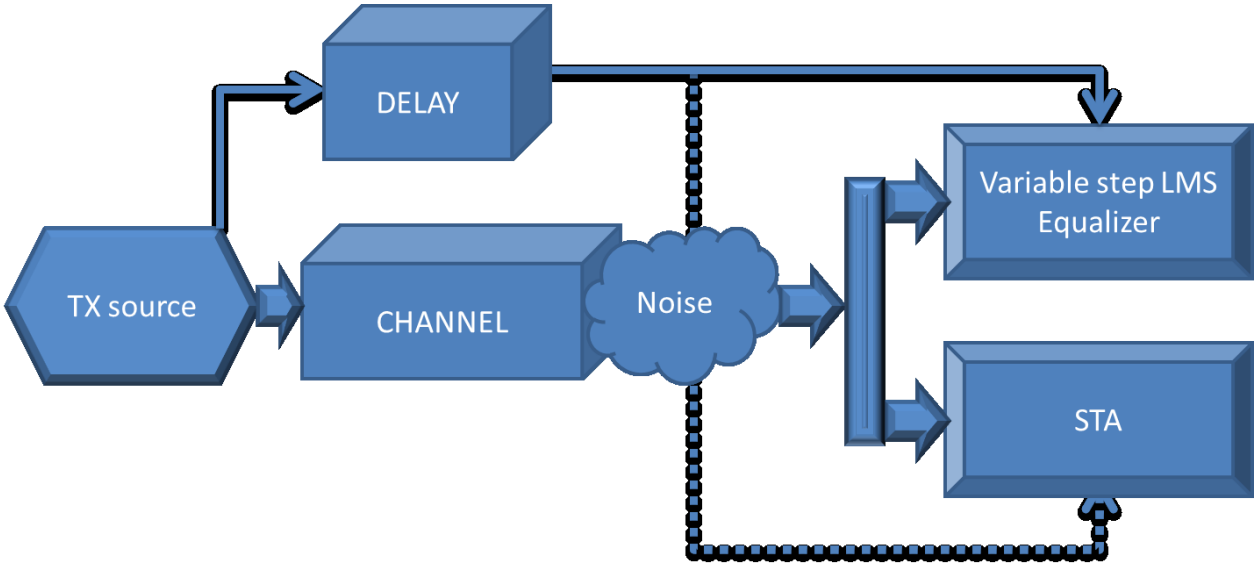


FIGURE 7-7 COMPARISON MODEL

7.3.1 IMPLEMENTATION:

Both algorithms were driven by the same reception signal at the receiver frontend and were implemented such that both perform in parallel design status to have a fair comparison between them.

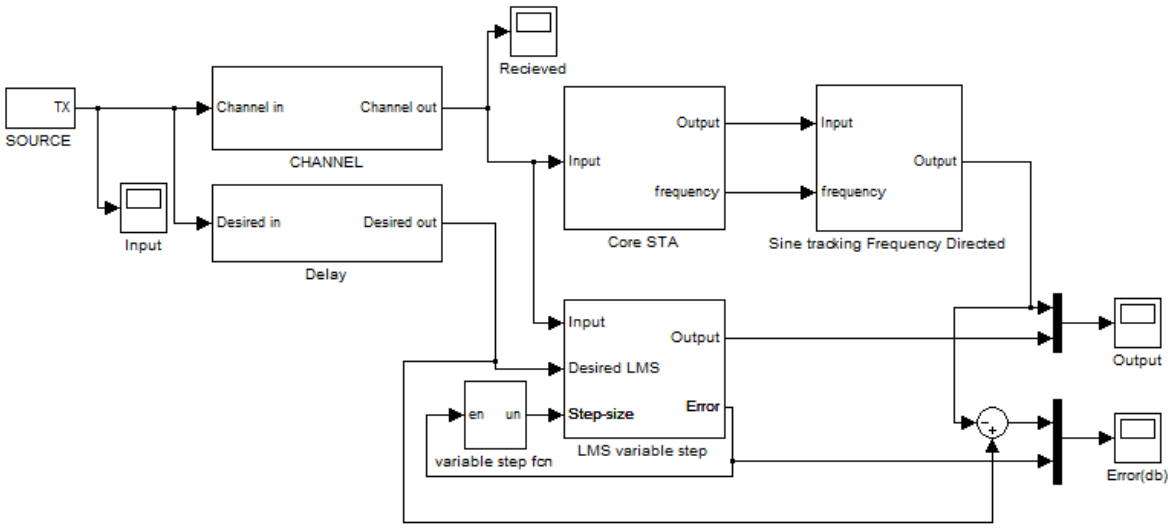


FIGURE 7-8 SIMULINK MODEL

7.3.2 RESULTS:

The x-axis of all graphs represent the number of samples having actual range between 0 – 2000. This value is scaled down by 10000, so the range on the axis is between 0 – 0.2. The y – axis represents true values.

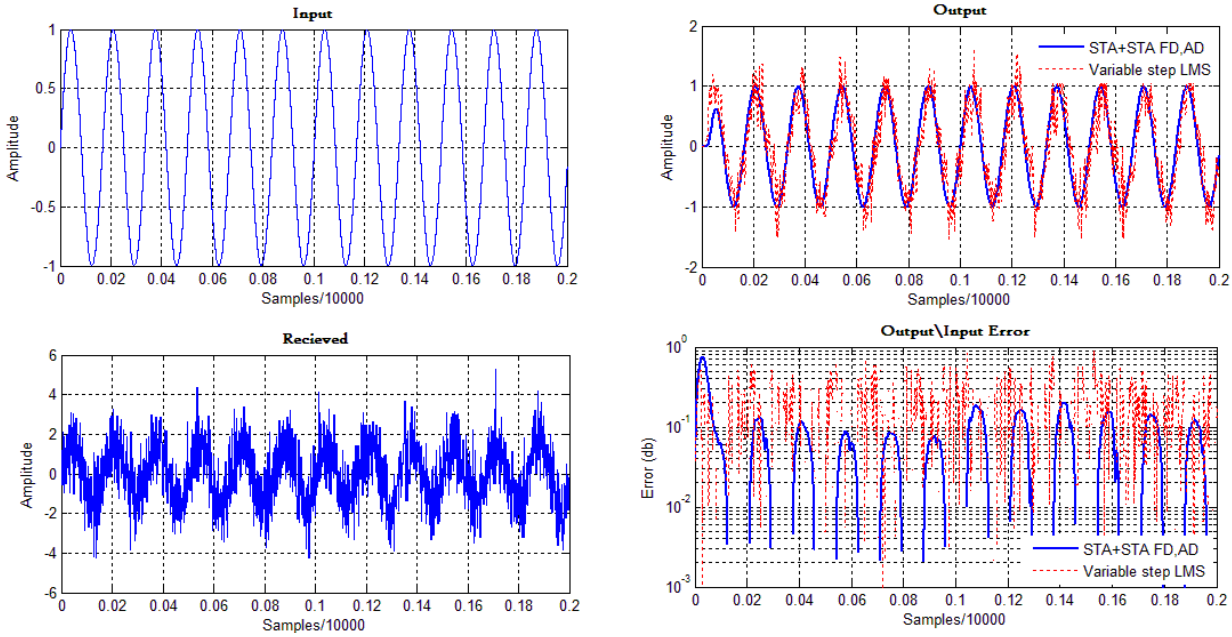


FIGURE 7-9 COMPARISON RESULTS

7.3.3 CONCLUSION:

As in the above implementations we have seen that the best performance was shown by the Variable step LMS and the Post processed STA with frequency dictation, so a comparison is conducted in this case between the best performers in both cases. The STA algorithm is also amplitude limited to see the best performance difference between the two.

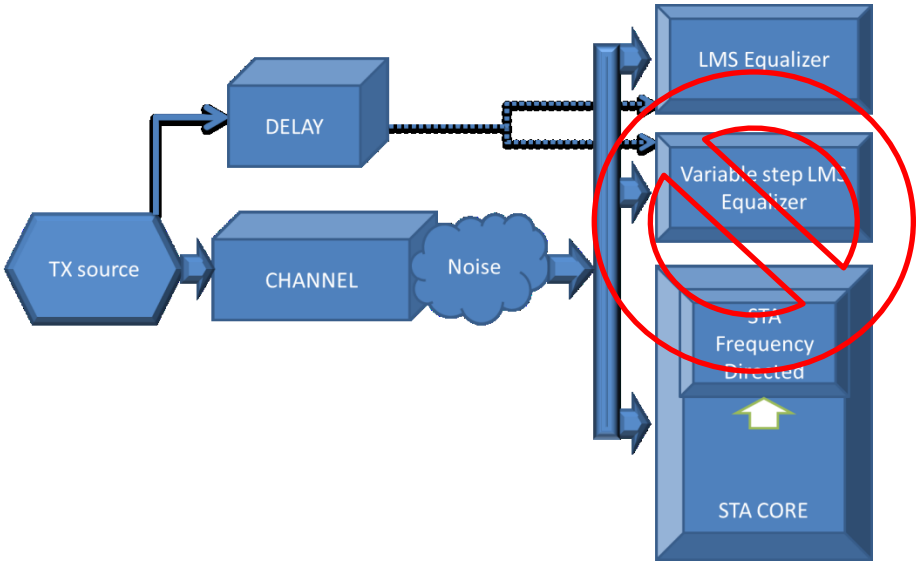
As shown in results a clear comparative behavior and large difference between the two is displayed and the STA technique with post processing gets all the marks for this comparative result.

7.4 CONCLUSIONS:

We have successfully implemented and evaluated the known techniques for their merits and demerits. However while this implementation a lot of problems were seen for both the techniques either from LMS or from STA group.

The LMS is capable of eliminating the ISI but has poor noise performance whereas the noise performance for the STA is good but as it is implemented Blind so all the parameters have to be set according to the nature of the input being tracked for the multi distorted signal which improves the performance as compared to that of the LMS based implementation but it is not so generic and redundant as compared to the LMS.

So what is required is a solution which satisfies both the characteristics and has a stable performance.



CHAPTER NO 8:

VARIABLE STEP LMS AIDED SINUSOID TRACKING ADAPTIVE REDUNDANT ALGORITHM

CHAPTER 8 VARIABLE STEP LMS AIDED SINUSOID TRACKING ADAPTIVE REDUNDANT ALGORITHM:

8.1 INTRODUCTION:

The comparison analysis portion deduced the results and the conclusions concluded the performance merits for different algorithms. This valuable data was used in this combined adaptive redundant algorithm to form a post processing for the variable step LMS filter to combine the merits of two algorithms [b.19][b.18].

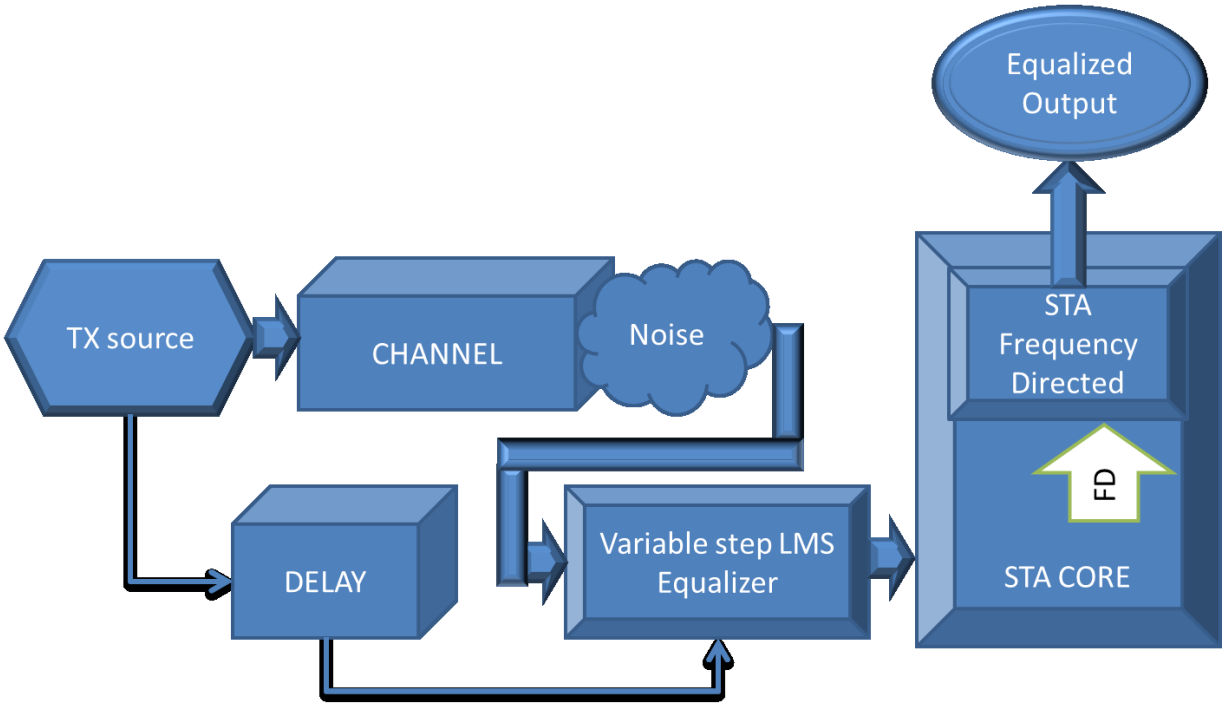


FIGURE 8-1PROPOSED FILTERING BLOCK DIAGRAM

8.2 IMPLEMENTATION:

The basic design is based on the fact that LMS filter is prone to noise and it needs a post processor to estimate and eliminate interferences that is generated or passed by the variable step LMS filter. So the technique of post processing the output of the arc tan-LMS by STA with frequency directed postprocessor is implemented and evaluated.

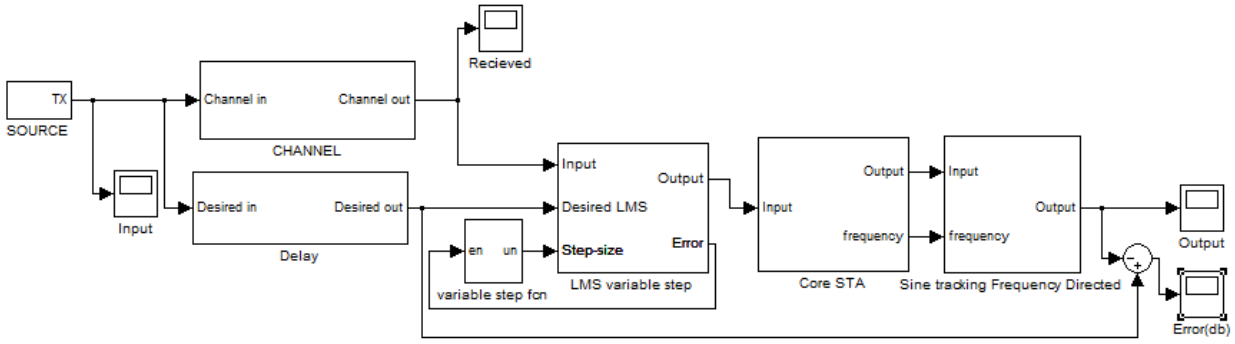


FIGURE 8-2 SIMULINK MODEL

8.3 EVALUATION:

The equalizer model developed is used and evaluated for the test system designed earlier. The relevant advantages and disadvantages of this technique are discussed.

8.4 RESULTS:

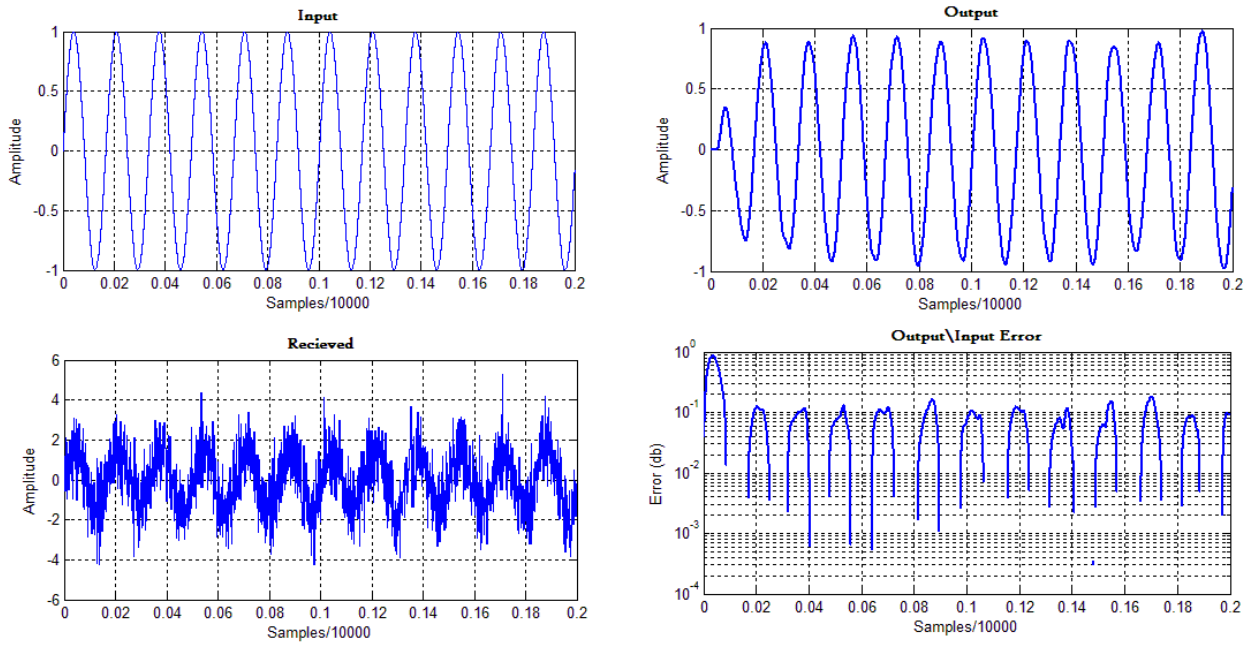


FIGURE 8-3 RESULTS

The x-axis of all graphs represent the number of samples having actual range between 0 – 2000. This value is scaled down by 10000, so the range on the axis is between 0 – 0.2. The y – axis represents true values.

8.5 CONCLUSION:

The performance for the proposed algorithm is stable for the period of its convergence and has a stable behavior throughout. As shown in the results we can clearly see that it has the best behavior for the same test channel system which is used as the bench mark for this work. The instability or the poor noise filtering of LMS is overcome by the STA whereas the blindness of the phase and frequency of STA is directed by the LMS which gives us a redundant stable and immune to noise filter combination.

The post processing has the disadvantage of reducing the gain which is shadowed as the error in results.

CHAPTER NO 9:

STA AIDED VARIABLE STEP LMS ADAPTIVE REDUNDANT ALGORITHM

CHAPTER 9 STA AIDED VARIABLE STEP LMS ADAPTIVE REDUNDANT ALGORITHM:

9.1 INTRODUCTION:

In chapter 8 we concluded that the STA aided post processor has a demerit of reducing the gain which is induced as an error in the tracking result limiting the overall effective performance. Though the results of such a design are valid but the combined design approach which has been taken in this work has shown a horizon which can be explored further to improve the tracking performance even more than what is termed as a satisfactory to excellence.

The main conclusion concluded until now is that the best results can be obtained if we combine the merits of the involved techniques and propose a redundant strategy which can far outweigh their demerits in terms of the performance. In this design the sinusoid tracking algorithm has been used as a preprocessor to remove noise component from the received signal [b.19] which gives the ease of equalizing the preprocessed less noisy input to the LMS or the variable step LMS algorithm hence improving the performance[b.18]. The figure below elaborates the block flowchart diagram for the design.

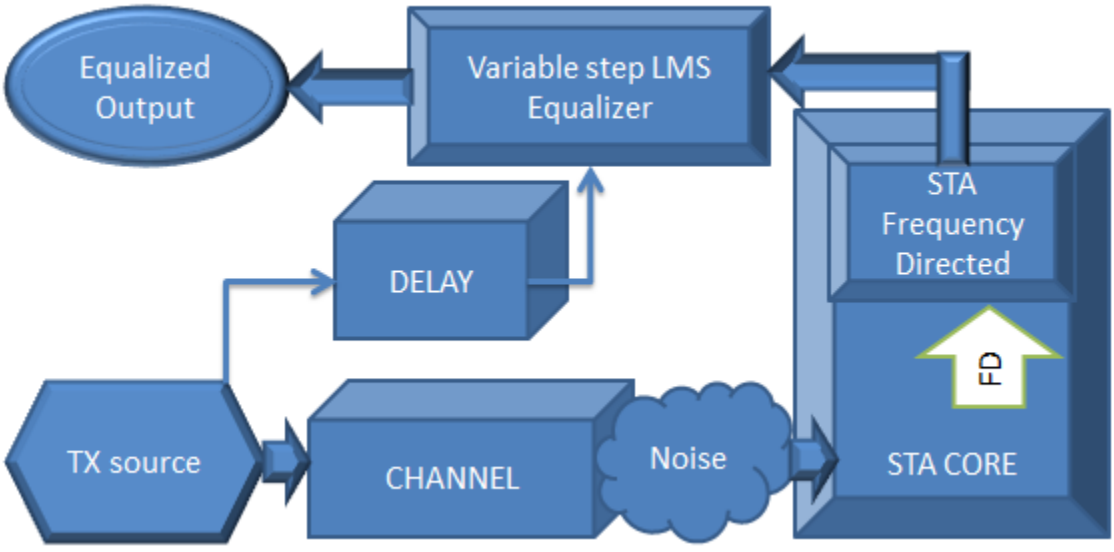


FIGURE 9-1 PROPOSED FILTERING BLOCK DIAGRAM

9.2 IMPLEMENTATION:

The design is developed so that to maximize tracking capability by using the sinusoid tracking algorithm cores. As from the earlier results and the work done [b.22] it is known that STA is a very redundant and good noise eliminator which can have an improved performance aided by a STA frequency directed smoother. This smoothened output is then equalized with a LMS filter to reduce the estimation error.

The STA core algorithm is applied to the raw channel signal received at the receiver end. It serves as the tracker of the desired relevant frequencies closer to the transmission bandwidth of the signal desired. As it is a nonlinear tracker and effectively tracks its noisy input and removes the noise from the desired sinusoidal component as well as it extracts the desired sinusoidal component as the output. The noise elimination using STA does not destroy any major properties of the desired information carrying component of the desired sinusoidal component. This is a major advantage of using such non stationary tracker. As a simple single core is used some jitter in the desired component phase is left and needs to be filter out. The second stage of processing occurs at the STA frequency directed sub core which is driven by the output of STA Core as input signal to be tracked and the tracked frequency generated by the STA core which is the direction frequency for the STA FD. This core acts as to smooth the desired component and further filter any unwanted components. The output is a pure sinusoid majorly free of any interference or jitter and is the noise free multipath reception. So the signal at this stage is free of channel noise which was added to it in the signal composition. A simple fixed step LMS filter with length 6 and step size 0.06 is used to remove multipath and act as inverse model to the channel as well as the preprocessor cores implemented which effectively equalizes the desired signal at the receiver to produce far better results from the typical LMS in the noisy environments. The LMS filter used is replaced by a variable step LMS filter with length 6 and the value for $\alpha=4$ $\beta=0.04$ and $\gamma=2$ to further stabilize the response in non-stationary channels.

The Simulink model developed shows the block structure where STA cores are followed by an arc-tan variable step LMS filter.

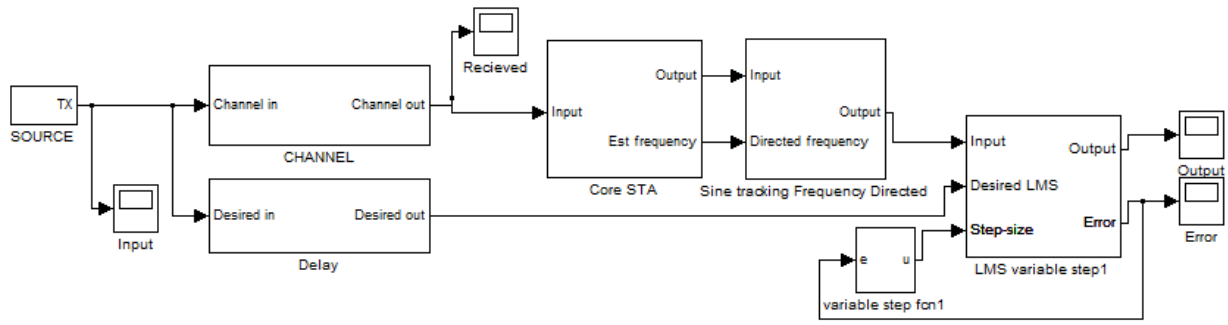


FIGURE 9-2 SIMULINK MODEL

9.3 EVALUATION:

The equalizer model developed is used and evaluated for the test system. The relevant advantages and disadvantages of this technique are discussed.

9.4 RESULTS:

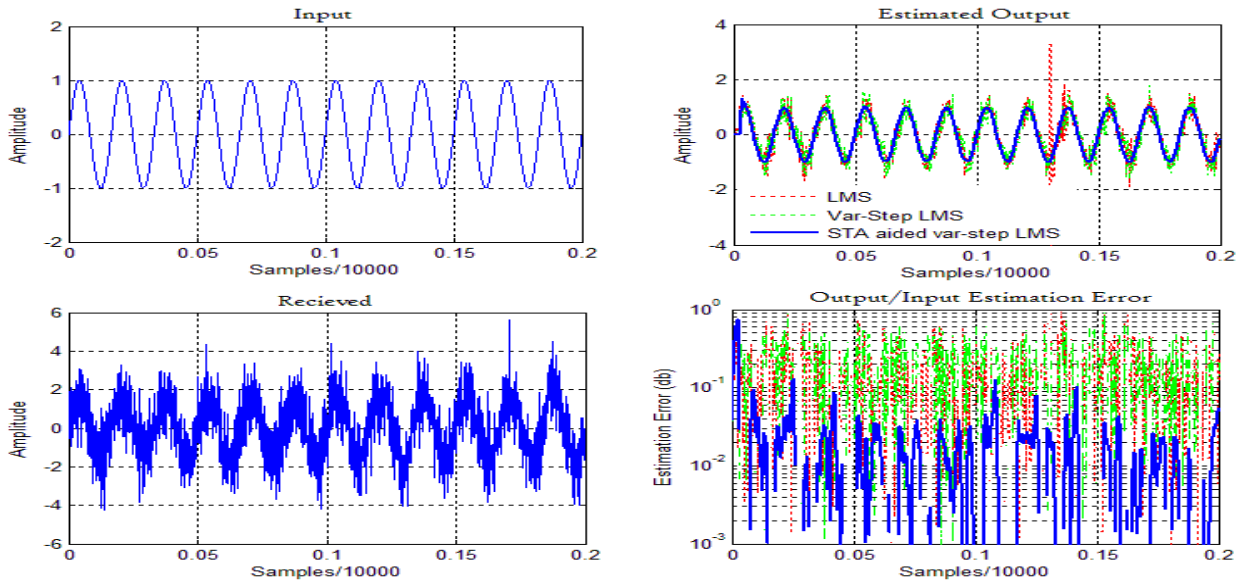


FIGURE 9-3 RESULTS

The x-axis of all graphs represent the number of samples having actual range between 0 – 2000. This value is scaled down by 10000, so the range on the axis is between 0 – 0.2. The y – axis represents true values.

The simulation results shown in figure 9-3 are a fair comparison between the LMS equalizer, an arc-tan variable step LMS equalizer and an STA aided variable step LMS equalizer. Section 3.6 explains the large spike induced in LMS tracked output at around 0.13. The performance improvement shown by the STA aided equalizer is significant and can easily be seen by the graph axis. The convergence error achieved by the variable step LMS is around -10 dB and that of STA aided design is -20 to -30dB. This means the STA aided equalizer has performance enhancement of about 10 to 20 dB when compared to the variable step LMS design. That is a significant improvement for a non-stationary channel in extreme noisy environments. These estimation performances are controlled by the multiple tune able factors of the STA, STA FD and the curve characteristics of the variable step LMS filter and can be varied to suit different applications.

9.5 CONCLUSION:

The implementation and evaluation of the STA as a preprocessor for noise removal aiding the variable step LMS equalizer is successful and the results shown help us conclude this to be an effective strategy to decrease estimation error for simple filters like the LMS and the variable step LMS. This is a great advantage for the current mobile communication systems which are designed from simple and discrete hardware to effectively improve their reception using simple architecture of the STA which will increase their capacity in the current channel bandwidth. This effective design strategy can be exploited for other simple algorithms which are affected by the noise induction as well as to increase performance for current applications.

CHAPTER NO 10:

FPGA BASED EFFICIENT IMPLEMENTATION OF SINUSOID TRACKING ADAPTIVE FILTER USING XILINX SYSTEM GENERATOR

CHAPTER 10 FPGA BASED ACHITECTURE DESIGN FOR STA:

10.1 ABSTRACT:

An equaling discrete time model is created to the proposed algorithm of A.K. Ziarani and A. Konrad. It is then evaluated and implemented on Matlab [d.1] using Simulink [d.2]. Its simulation results are verified for a Simulink step size of discrete value '1'. This discrete model architecture is developed to aid the system generator based design. Xilinx block set is used to find the alternatives to the discrete blocks of Simulink of whom some blocks are used in unique combinations to generate those functions which are not directly available. The step by step implementation is achieved on system generator based architecture by carefully adjusting the path delays between parallel paths as Xilinx blocks have different latency settings. A brief description is presented below about all the blocks and their settings and the unique challenges faced during this architecture development.

10.2 INTRODUCTION:

Field programmable gate arrays [a.9] present us with the facility of developing custom architectures efficiently and with greater ease and accuracy than any other hardware based solution. This is an indication of the bright future FPGA based efficient architectures have in future [a.9]. The STA algorithm proposed by A.K Ziarani is an efficient adaptive filter which is most likely to be a part of many applications due to its exceptional performance for the nonlinear environments. The best of the Digital signal processing applications are being implemented on reconfigurable hardware like the FPGA. Doing a Verilog code [a.9] to implement this architecture is a hefty task, so a high level design and analysis approach for implementation using Xilinx system generator is taken for this task.

10.3 IMPLEMENTATION:

10.3.1 IMPLEMENTING CONTINUOUS TIME MODEL:

A continuous time model for the proposed architecture is created using Simulink to verify the results presented in paper as well as to understand the working of the algorithms. The notable fact in this diagram is the use of real continuous time integrator, all the blocks have inheritance step size setting of '-1', whereas the input is a continuous time sine signal of frequency ~60 Hz about 377 radians. The initial condition for integrator 1 is set to be 50 as it is close to our desired 60 Hz so the filter is to converge quickly. The Simulink model shown below is implementation of the architecture and its results are consistent to the paper described by the author.

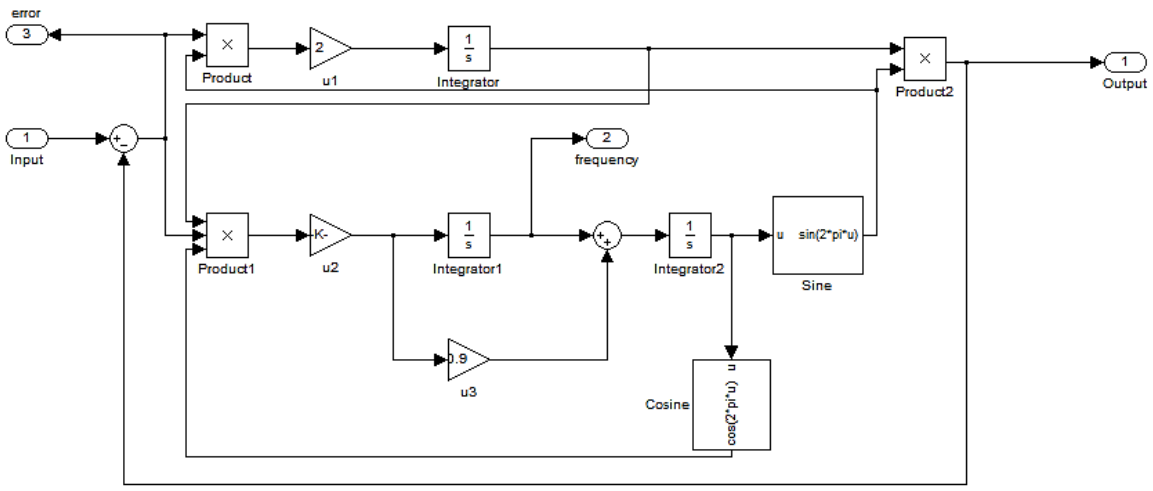


FIGURE 10-1 SIMULINK BASED CONTINUOUS IMPLEMENTATION

10.3.2 IMPLEMENTING DISCRETE TIME MODEL:

The discrete time algorithm was implemented on a discrete time with step size 1. The inbuilt discrete time integrator was not used as it is unavailable in the system generator so a structure was developed and implemented using discrete delay and sum components to act as the desired discrete time integrator. One notable fact is that the use of a 'u' gain to determine the precision of the integrator or its internal step size. The setting of this parameter determines the actual precision and the rate of convergence to be achieved for discrete filter. The settings for this parameter are in the range 0.001-0.0001 etc.

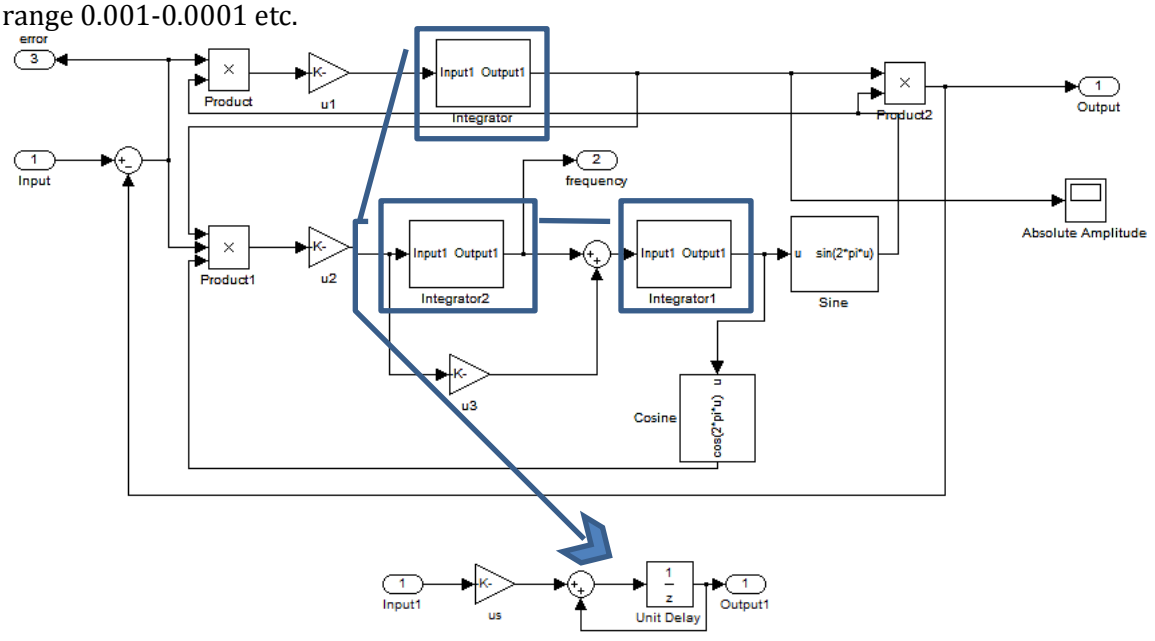


FIGURE 10-2 DESIGN OF DISCRETE TIME INTEGRATOR

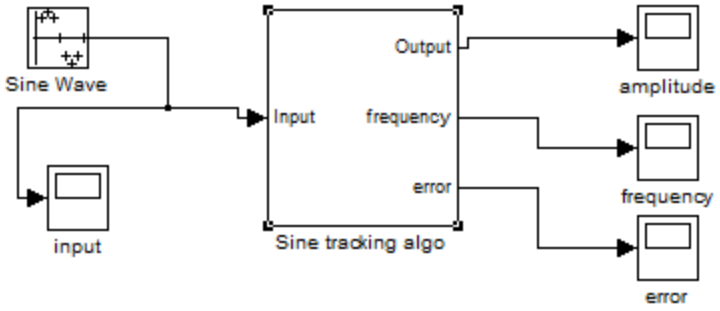
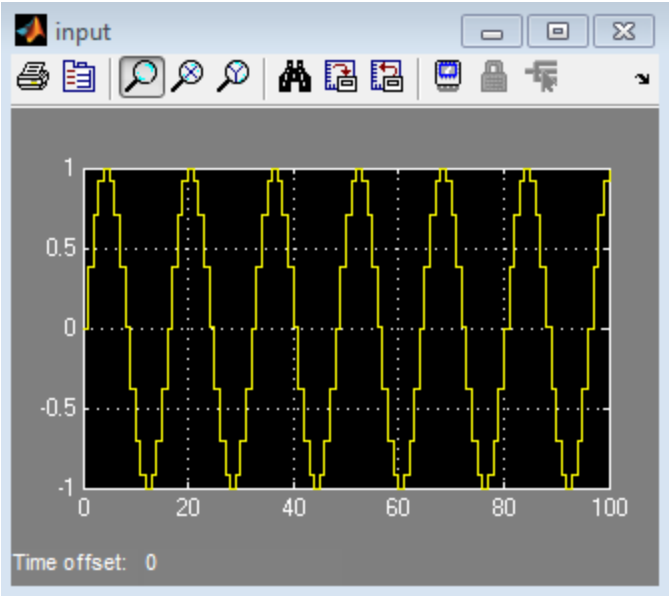


FIGURE 10-3 TOP LEVEL MODULE

For this discrete time model the input is a sampled sine wave as shown in this figure it is discrete in nature and has about 16 samples in one of its periods. A significant amount of periods are generated to allow the filter to converge.

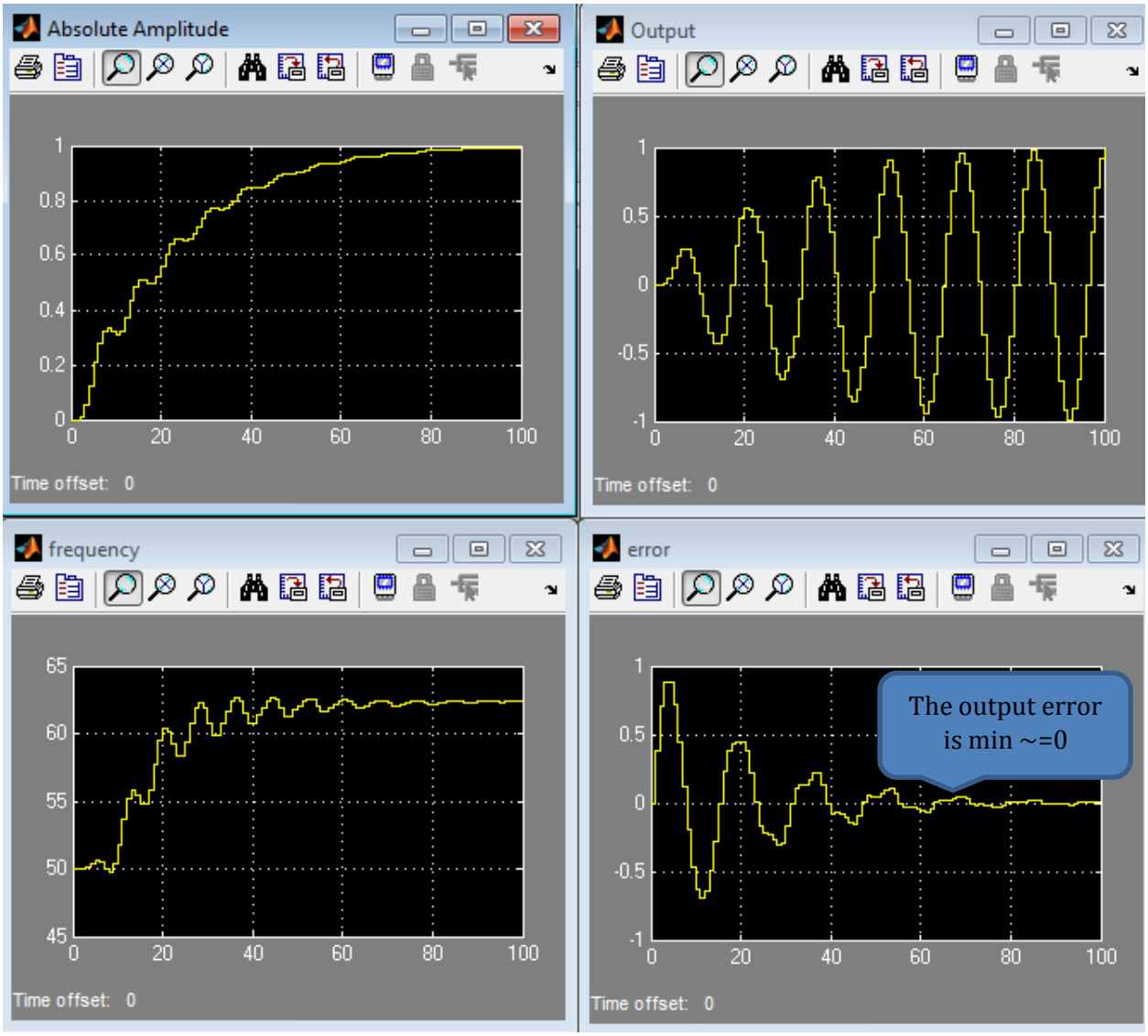
As shown in scope 100 samples are generated as input to the above discrete time sine tracking algorithm. The sampling frequency can be increased to generate a smoother input which will result in more precise and accurate output values for the discrete filter.



SIMULATION RESULT FOR INPUT SCOPE

10.3.2.1 RESULTS:

The results on next page depict the successful implementation and development of the algorithm in discrete domain, which is one of the fundamental requirements for Xilinx system generator based implementation.
The absolute amplitude shows the tracked amplitude of the input signal which can be clearly seen as the signal approaches '1' at around 80 samples of the input.
The tracked frequency is at around ~62 Hz, which shows also minimum oscillatory behavior @ > 80 samples.



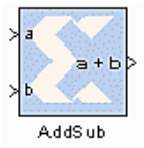
OUTPUT RESULTS FOR TRACKED AMPLITUDE, OUTPUT, FREQUENCY AN ESTIMATION ERROR

The results also show minimum error @ > 80 samples of the input. This behavior is satisfactory for all such inputs or the related ones.

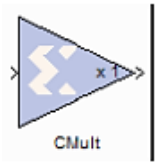
The above system was also tested for the step input response and it passed that with reasonable behavior and output was a successfully tracked step function.

10.4 EXPLORING SYSTEM GENERATOR:

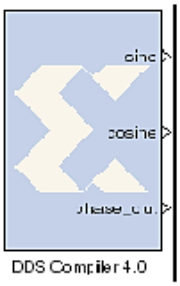
A working SYSGEN model can only be developed dealing with in the domain of reconfigurable computing especially on high level synthesis tools. So the best help available is from Xilinx website [c.4] and the manual user guide for the Xilinx system generator [d.3]. Xilinx University program provides XSG labs for students which are relevant to understanding the basics of fixed point implementation and Xilinx implementation and analysis Block sets provided along the MATLAB.
The XSG blocks which are of the most assistance in successfully implementing and evaluating the above architecture are the following.



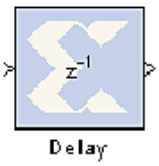
The Xilinx AddSub block implements an adder/subtractor. The operation can be fixed (Addition or Subtraction) or changed dynamically under control of the sub mode signal.



The Xilinx CMult block implements a *gain* operator, with output equal to the product of its input by a constant value. This value can be a MATLAB expression that evaluates to a constant.

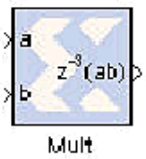


The Xilinx DDS Compiler block is a direct digital synthesizer, also commonly called a numerically controlled oscillator (NCO). The block uses a lookup table scheme to generate sinusoids. A digital integrator (accumulator) generates a phase that is mapped by the lookup table into the output sinusoidal waveform.



The Xilinx Delay block implements a fixed delay of L cycles.

The delay value is displayed on the block in the form z^{-L}, which is the Z-*transform* of the block's transfer function. Any data provided to the input of the block will appear at the output after L cycles. The rate and type of the data of the output will be inherited from the input. This block is used mainly for



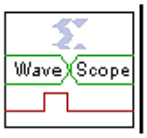
The Xilinx Mult block implements a multiplier. It computes the product of the data on its two input ports, producing the result on its output port.

The Xilinx Register block models a D flip flop-based register, having latency of one sample period.

The Xilinx Resource Estimator block provides fast estimates of FPGA resources required to implement a System Generator subsystem or model.

These estimates are computed by invoking block-specific estimators for Xilinx blocks, and summing these values to obtain aggregated estimates of lookup tables (LUTs), flip-flops (FFs), block memories (BRAM), 18x18 multipliers, tristate buffers, and I/Os.

The System Generator token serves as a control panel for controling system and simulation parameters, and it is also used to invoke the code generator for netlisting. Every Simulink model containing any element from the Xilinx Blockset must contain at least one System Generator token. Once a System Generator token is added to a model, it is possible to specify how code generation and simulation should be handled.



The System Generator WaveScope block provides a powerful and easy-to-use waveform viewer for analyzing and debugging System Generator designs.

The viewer allows you to observe the time-changing values of any wires in the design after the conclusion of the simulation. The signals may be formatted in a logic or analog format and may be viewed in binary, hex, or decimal radices.

10.5 CREATING THE SYSGEN MODEL:

10.5.1 AN APPROACH TOWARDS SYSTEM EXPLORATION:

Once you know about the working of the model and about all the blocks you are to use, that it is relatively simple just like Simulink coding drag and drop and then connect. You have a model but it's not even the beginning yet for the implementation step procedure as much more concentration is needed for the successful implementation.

The output we get is never the one we expect as the hardware plays its tricks introducing lags and leads and unforeseen delays to destroy the desired output so we need to come up with some tricks of our own to tackle these problems.

After successful implementation of the Simulink based design for the sampling time of '1' the system generator based design was implemented on equaling basis. Even when it comes to system generator based implementation for simple designs it is much trickier than implementation of the Simulink designs. So a block by block implementation and verification technique was implemented in order to avoid wasting time during debugging of potential problems by eliminating most of their causes at the design step. A mixed (15_15, 15_13, 20_5, 15_6, 15_7 & 14_13) bit fixed point architecture is developed to efficiently allocate resources.

Each block of the Simulink based reference model is designed separately for the system generator based model by studying the block behavior for the outputs generated when the specific inputs are applied. The full system implementation can be performed separately by many designers at the same time while cutting short the product development time. This technique also helps to eliminate any design compatibility bugs at the implementation stage. The technique works by the procedure of one problem at a time and it can be successfully implemented in most of the designs which possess feedback behavior as the case with this design. Figure 10-4 shows the flowchart for the technique used. Verification is preformed to check the outputs match those desired by our system.

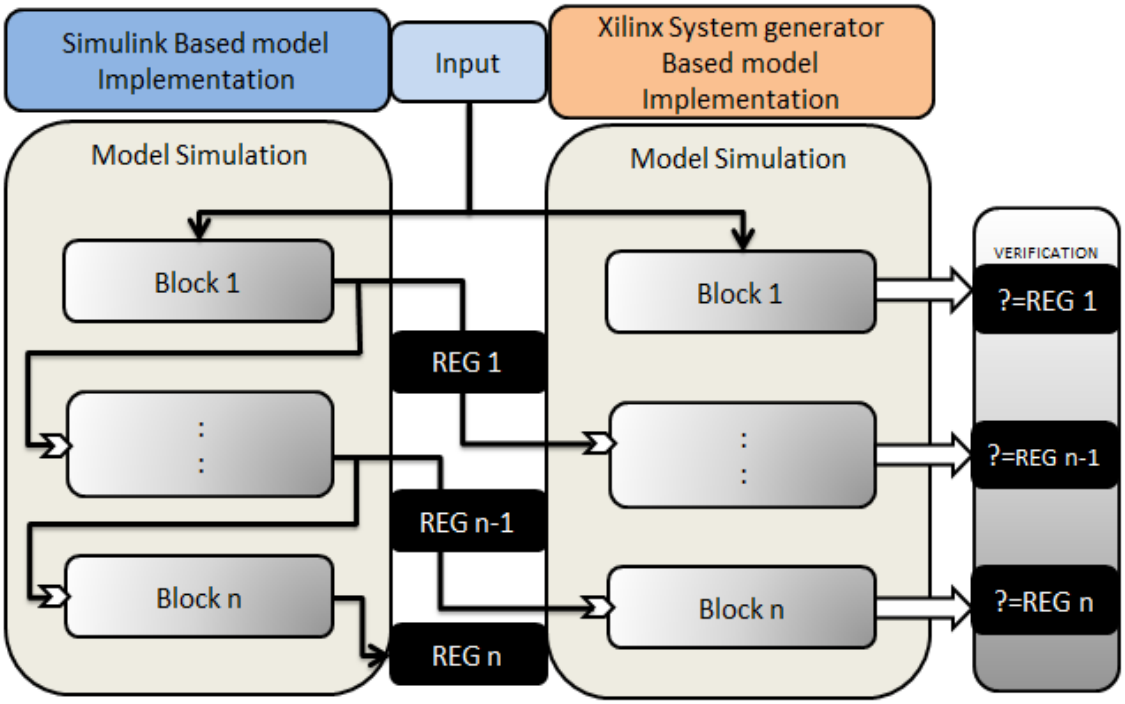


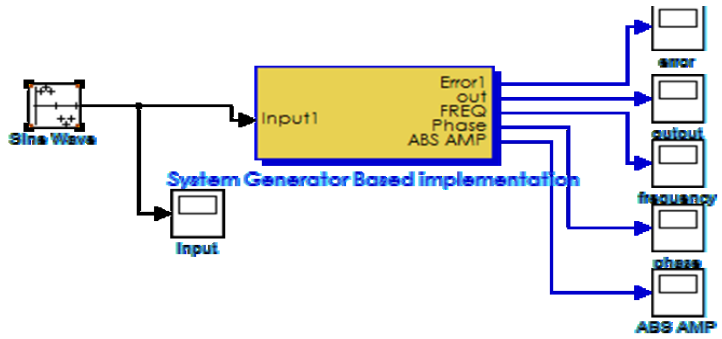
FIGURE 10-4 FLOWCHART FOR THE DESIGN

What is needed is some kind of visualization about what's going on in there. For this sole purpose the Xilinx block set includes the Wave scope. It is a compact tool; just include it in the model and add all your desired wires to its watch list you want to see. This scope is the developers Eyes and ears about all that is happening during execution at each of the blocks input and output.

Another useful tool is the Xilinx single step simulation as it allows going through the design step by step and identifying the problem which is the root cause to the faulty undesired output.

10.5.2 MODEL DEVELOPED:

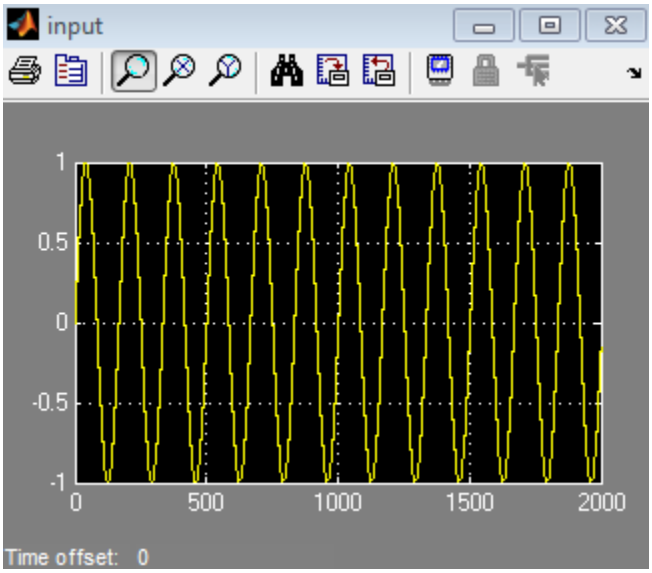
The model developed is implemented in system generated and its block diagram is included below for viewing.



TOP LEVEL MODULE IMPLEMENTED

The input is a sampled sine wave as shown in this figure it is discrete in nature and has about 167 samples in one of its periods. A significant amount of periods are generated to allow the filter to converge.

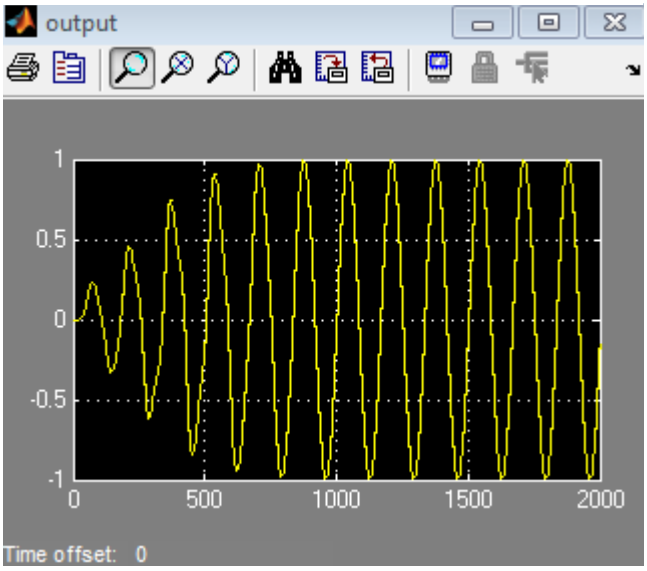
As shown in scope about 2000 samples are generated as input to the above system generator based sine tracking algorithm. The sampling frequency can be increased to generate a smoother input so that more precise and accurately tracked filter outputs can be processed.



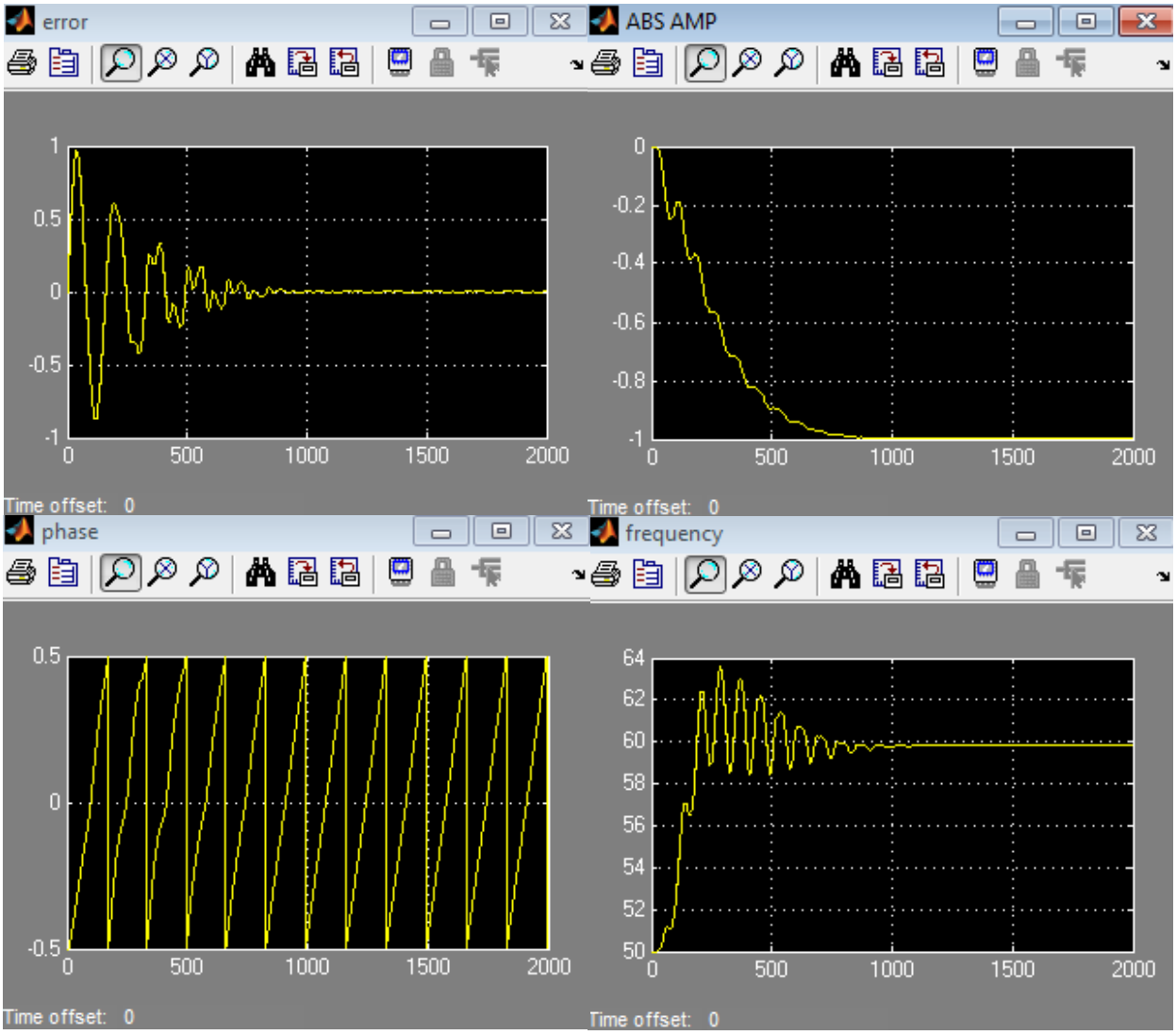
SIMULATION RESULT FOR THE INPUT SCOPE

The output scope depicts the successful implementation and development of the algorithm on system generator model based implementation.

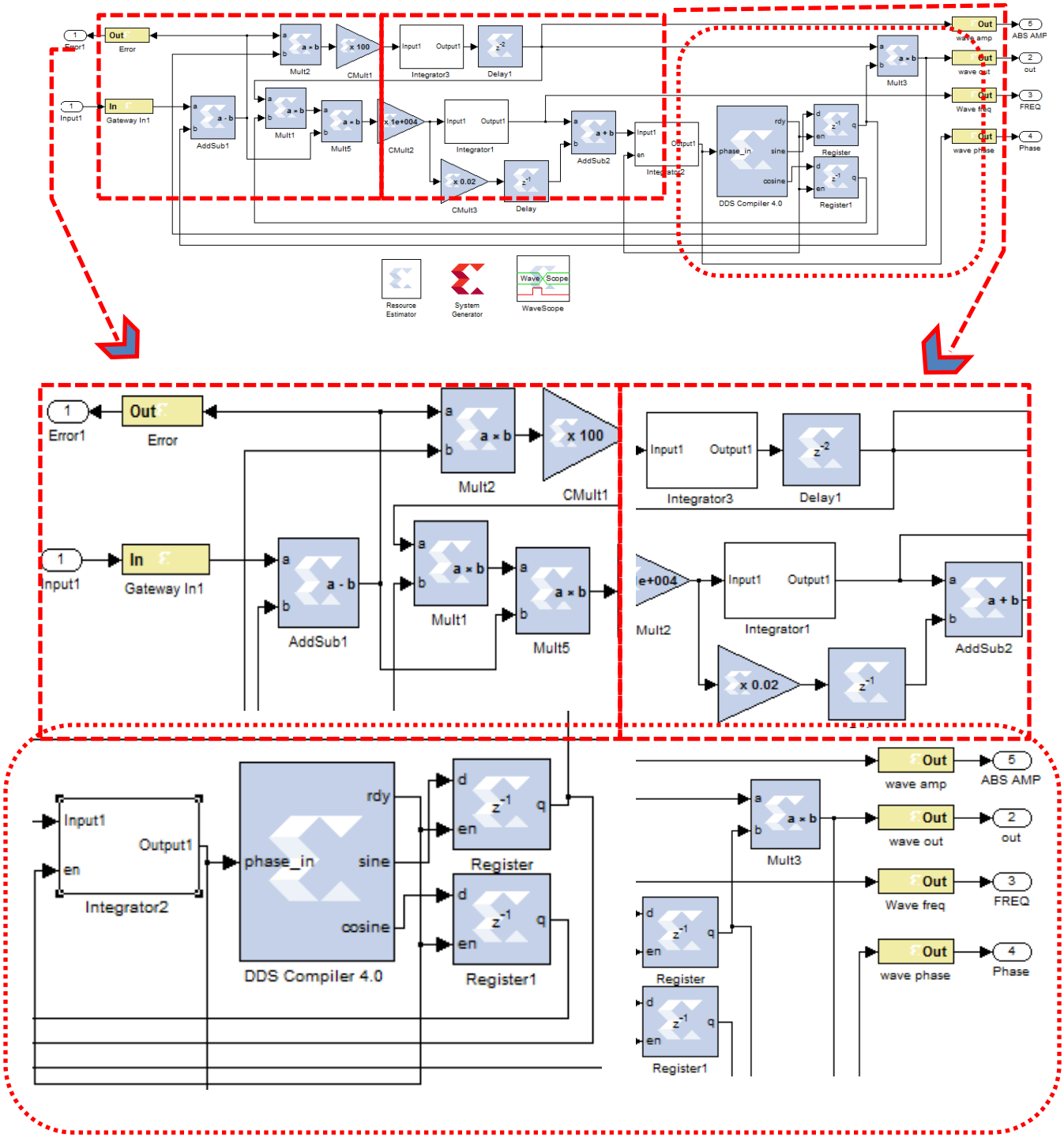
On next page the System generator output for error, absolute amplitude, frequency and phase are displayed and it can be seen clearly that the system model developed is well in range with the outputs depicted earlier in Discrete time model using Simulink.



SIMULATION RESULT FOR THE TRACKED OUTPUT SCOPE



SCOPE OUTPUTS FOR TRACKED AMPLITUDE, ESTIMATION ERROR, PHASE AND FREQUENCY FOR THE DISCRETE TIME MODEL SIMULATED USING THE XILINX SYSTEM GENERATOR



ZOOMED IN VIEW FOR DIFFERENT BLOCKS IMPLEMENTED IN XSG TO FORM THE TOP LEVEL MODULE

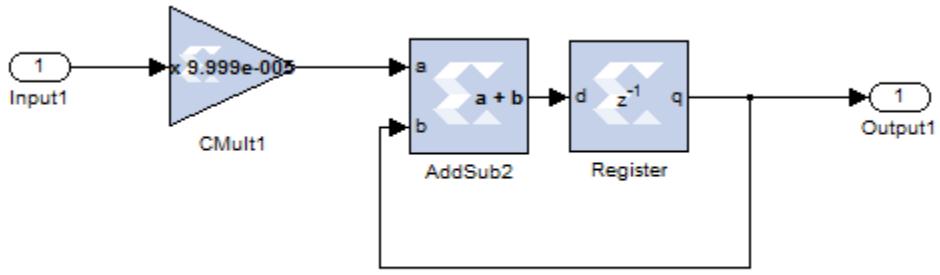


FIGURE 10-5 GENERAL INTEGRATOR DESIGN

Seen in the above figure is a standard integrator design used in the model to integrate the discrete time signals. It is same in design to the one we developed earlier for the Simulink discrete time model.

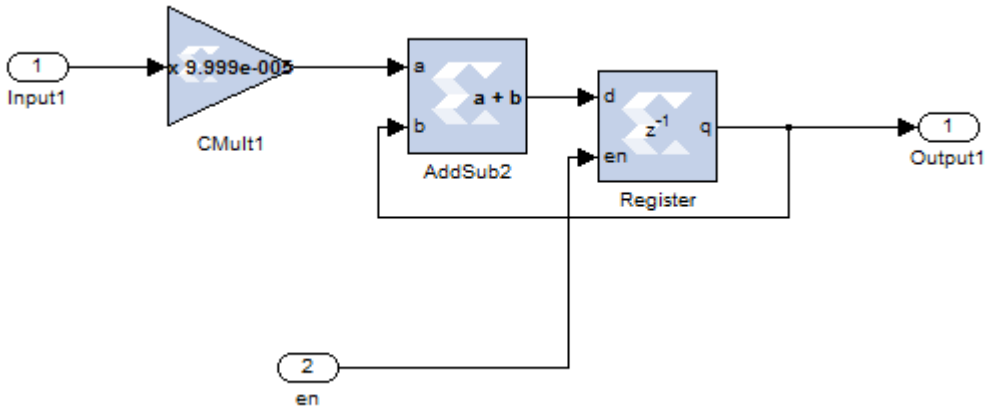
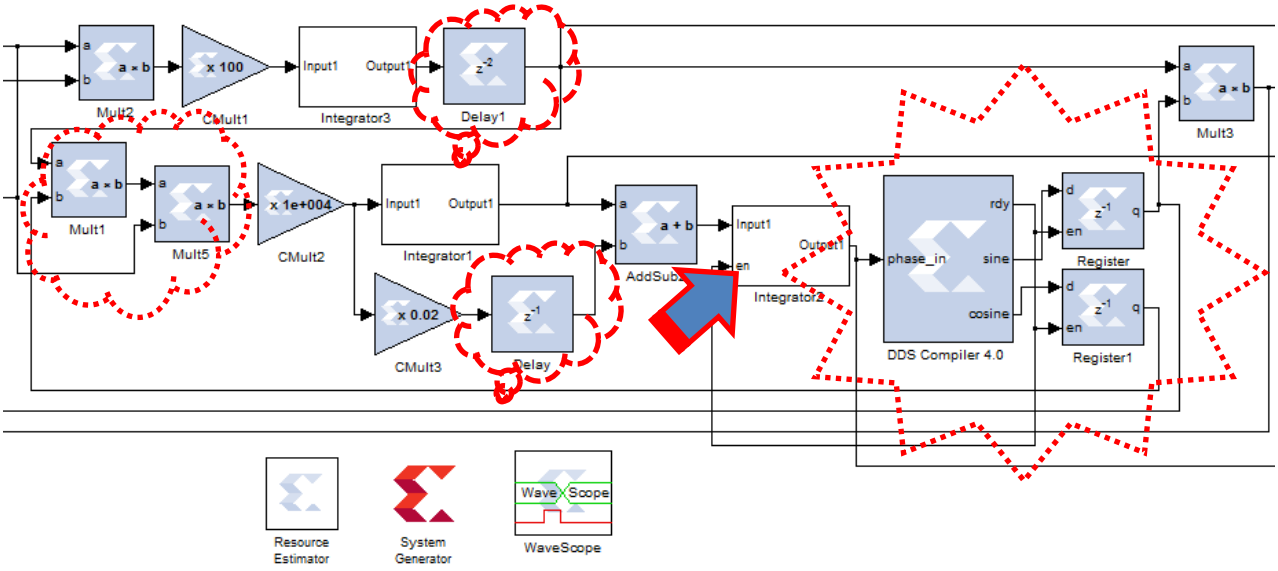


FIGURE 10-6 SPECIAL DESIGN FOR INTEGRATOR2 WITH START ENABLE

The phase integrator shown above has to be started only after the DDS block has a ready signal otherwise a large phase error is generated in the start of the filter which tends it unstable and is not a good practice.

10.6 MAJOR HURDLES SUCCESSFULLY NAVIGATED THROUGH:



HIGHLIGHTING MAJOR DESIGN ISSUES WHICH ARE RESOLVED TO IMPLEMENT THE DESIGN SUCESSFULLY

As seen in the above model it has two major parallel paths terminating at Mult1 and Mult3, if these have different latency then we have a problematic output. So what we need to do is add custom delay units in both to estimate the correct output. The delay1 introduces a delay of 2 clock cycles to overcome the path lead problem.

Another parallel path exists along the integrator1 block this is also adjusted by adding a delay after Cmult3.With this optimization all parallel paths in our model have same latency and are good to go.

The phase integrator shown by the arrow has to be started only after the DDS block has a ready signal otherwise a large phase error is generated in the start of the filter which tends it unstable and is not a good practice. A solution to this is implemented by a custom ready to enable integrator for proper phase generation control and to optimize the output.

Designing the NCO to generate the Sine and cosine was a major hurdle several methods were evaluated including cordic sin/cos generator block but the most relevant was found out to be DDS compiler. It was implemented to get the sine cosine output and two registers fed with enable were placed at the outputs to cater the unknown state X form propagating to the rest of the model.

Major challenges like parallel path delay optimization and NCO control for bounded output along with implementing multipliers were resolved and then these were put together to form the successful system we just observed as above.

10.7 OPTIMIZATION:

The DDS compiler output for sine and cosine was buffered to avoid the unknown state 'X' from propagating during the initializing phase. Figure 10-6 shows the enable controlled integrator2 to avoid phase jitter during initialization. The phase register is set at the initial value '-0.5'.After the design was successfully implemented in XSG the timing and power analysis for the architecture was performed. Post place and route timing report was generated for the XSG design which showed the max clock frequency supported by the design to be around 40.912 MHz which may not support certain timing critical applications so optimization was performed on the architecture to increase the overall clock speed for the design. The critical path was identified from the timing report studying the overall latency values for path delays and was partially pipelined which decreased the latency for the critical path to increase the operating frequency to 100.482 MHz which is much more compatible with time sensitive real-time systems. Figure 10-7 shows the partially pipelined optimized architecture for the algorithm [b.5] implemented using XSG. Figure 10-9 shows the histogram for path delay after its critical path was optimized.

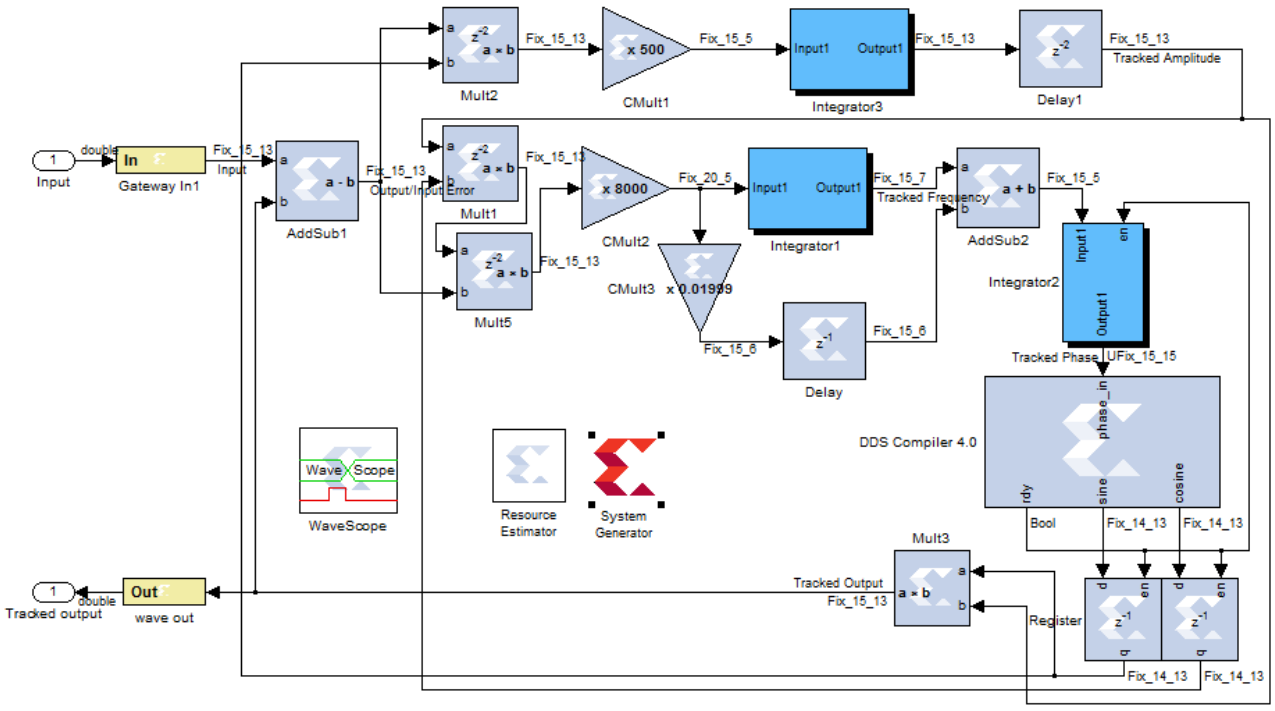


FIGURE 10-7 OPTIMIZED ARCHITECTURE

The wave scope tool gives us unprecedented view of all the activity going on and it was successfully used to overcome major hurdles and limitations. As seen in figure 10-8 it was successfully implemented to see the step by step output and in path results for the model debugging.

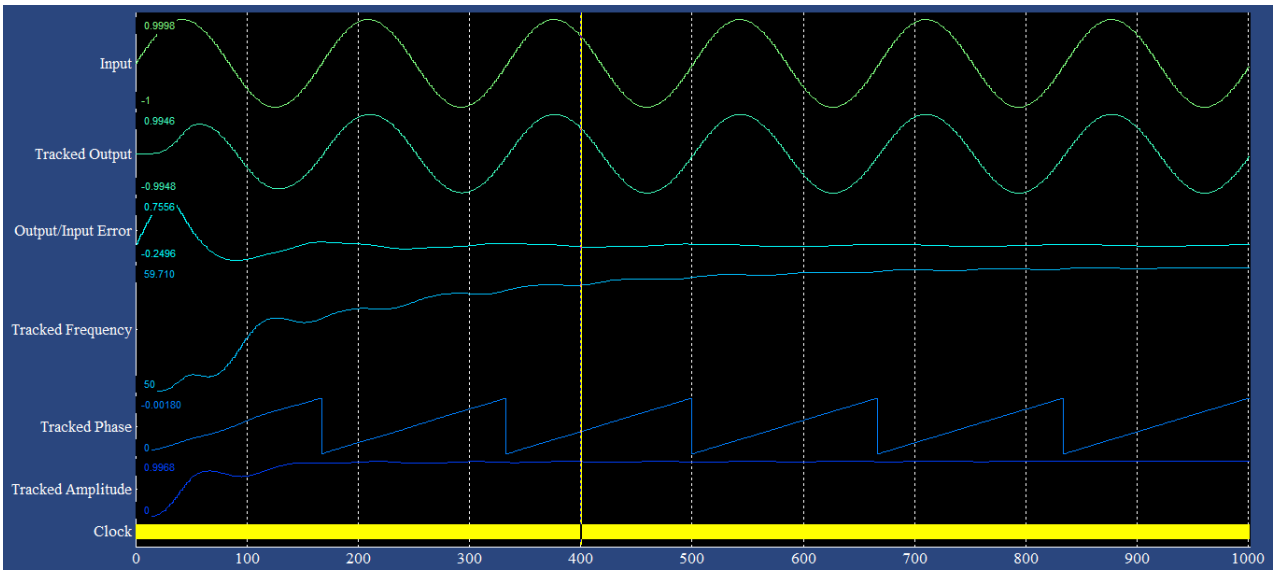


FIGURE 10-8 WAVE SCOPE VIEW

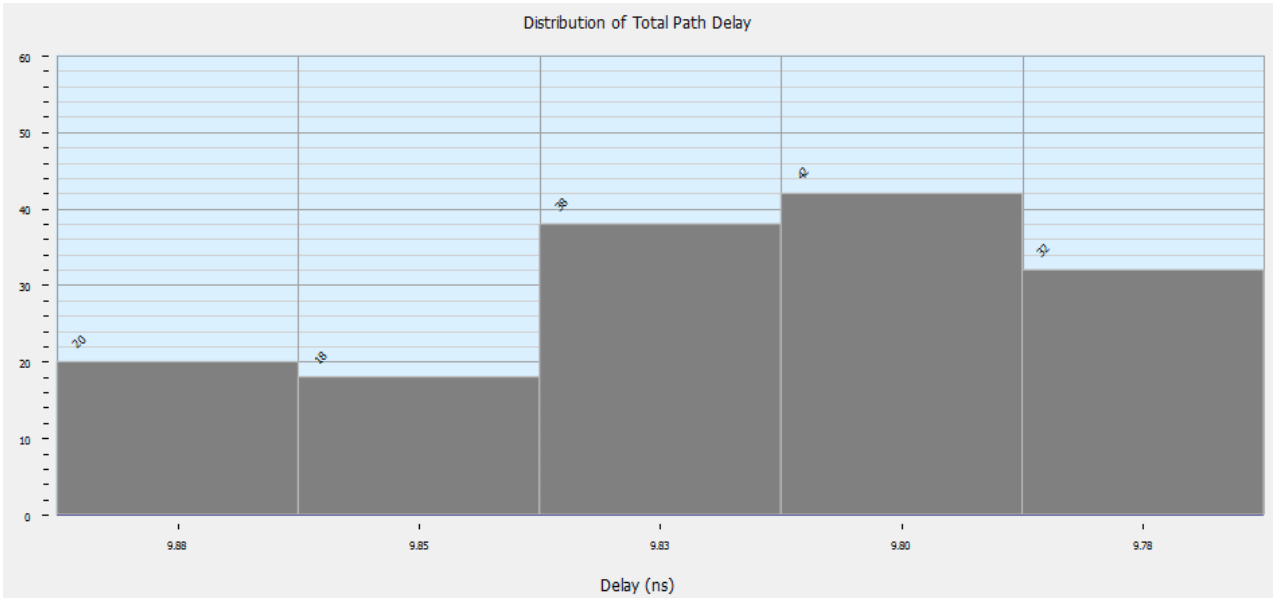


FIGURE 10-9 PATH DELAY HISTOGRAM

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.003	1	---	---
Logic	0.005	358	9112	4
Signals	0.005	511	---	---
BRAMs	0.009	*	*	*
DSPs	0.003	4	32	13
IOs	0.025	31	232	13
Leakage	0.020			
Total	0.071			

FIGURE 10-10 POWER ANALYSIS

10.7.1 DESIGN ANALYSIS:

Device Spartan6-xc6slx16-csg324	**Optimization**	
	Before	**After**
Device Utilization Summary		
Slice Logic Utilization: Number of Slice Registers: Number of Slice LUTs:	 109 1,160	 116 358
Slice Logic Distribution: Number of occupied Slices: Number of MUXCYs used: Number of LUT-FF pairs:	 327 1,288 1,163	 117 328 360
IO Utilization: Number of bonded IOBs:	 31	 31
Specific Feature Utilization: Number of RAMB **16**: Number of RAMB **8**: Number of BUFGs: Number of DSP48A1s:	 6 1 1 4	 6 1 1 4
Maximum Operating Frequency		
MHz	40.912	100.482
Power Utilization		
Watts	0.098	0.071

This section presents a comparison between the pre and post optimized architectures in terms of the resources usage, the maximum operable design frequency and the power usage.

10.7.2 RESULTS:

The optimized partially pipelined architecture was simulated and the results were verified which show the successful implementation of the model. Figure 10-8 show internal signal waveforms generated by wave scope for the architecture implemented. Section 10.7.1 shows maximum operating frequency and power utilization before and after optimization with partial pipeline in the critical path. Figure 10-10 shows power analysis. The designed architecture uses a total power of 0.071(W). The junction temperature obtained is also close to room temperature at 27.0(C). These power and temperature ratings validate the design for usage in portable devices. The resource utilization is shown in section 10.7.1 depicting reasonable usage in Spartan 6 based devices for development of larger integrated architectures on available resources. Xilinx Spartan6-xc6slx16-csg324 was targeted as the design chip to serve as reference for the results published here. The tracking validation for the developed hardware based architecture is obtained by tracking a noisy input sinusoid. Figure 10-11 shows the simulation results for the noisy input and tracked output generated by the filter implemented. Sampled based noisy input sinusoid is filtered to obtain the source signal.

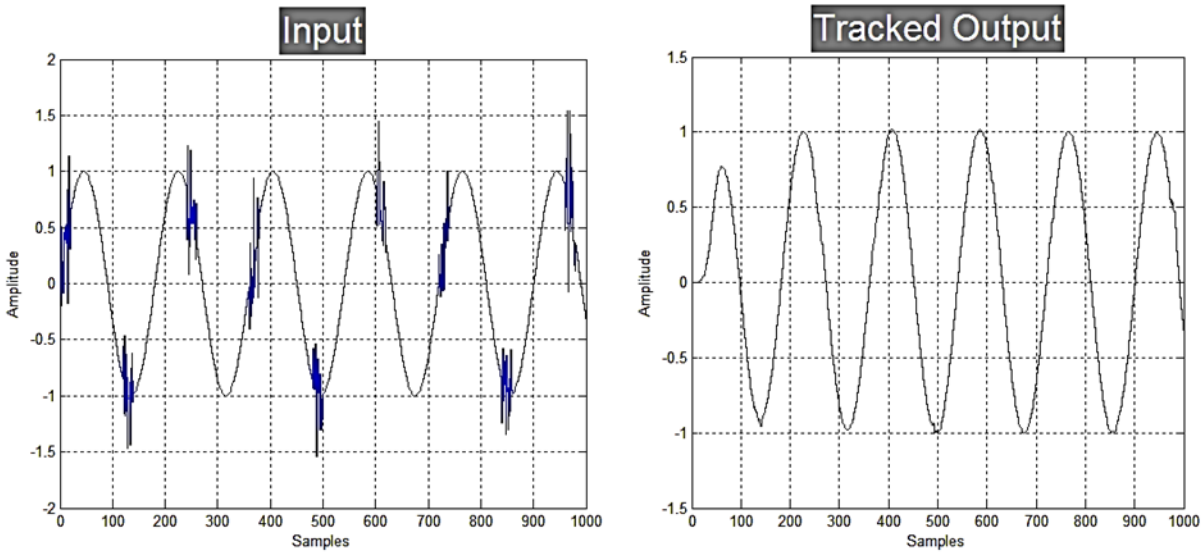


FIGURE 10-11 TRACKING RESULTS OF XSG MODEL

10.8 CONCLUSION:

The XSG based architecture for the novel adaptive filter has been designed and implemented successfully showing promising results. This filter block can be integrated within large systems [b.20] fulfilling the design requirements for different systems for their XSG based implementation and ultimately their hardware based development for use in real-time user based applications [b.22][b.21][b.15][b.16][b.17][b.8][b.3]. The developed design can serve as a reference model for further improvement in this design or future XSG based development of similar models.

RESEARCH RECOMMENDATIONS

The work presented in this thesis can serve as a guideline for the upcoming research interests for those students who have a motivational approach towards the relevant subjects of adaptive signal processing in wireless non stationary channels along with its applications. Although the specifics are discussed in detail and the merit is met but such innovative and broad spectrum fields are a fair game for anyone with the right approach of understanding and implementation. Some research recommendations have been put together as follows for those who are willing to take the challenge.

- Implementation and evaluation of the STA for other unexplored applications like beam forming echo cancellation and neural networks.
- Developing a new STA aided back propagation neural network for complex behavior analysis and matching.
- Aiding the STA equalizer design using other nonlinear advance algorithms like Recursive Least squares (RLS), State Space RLS (SSRLS), SSRLS with adaptive memory, Extended Kalman filter.
- A performance comparison between STA and Particle filters.
- Implement a variable discrete step size approach like arctangent for discrete implementation of STA algorithm for increasing convergence and steady-state stability.
- Image processing applications aided by STA.
- Removing flicker from video feeds using the STA flicker elimination technique.
- Aiding the Discrete wavelet transform using the STA frequency component extraction property.
- Audio compression techniques development using the STA interpolation feature.
- Encrypting data in a single component (e.g. amplitude, frequency, and phase) or between different component properties for the development of a new encryption technique.
- Design of a multidimensional STA architecture.
- FPGA based implementation of the different equalizers developed in this work or other relevant applications using either Verilog HDL or Xilinx system generator.

Researchers can benefit from the bibliography section for new innovative ideas as well as to understand and implement their research.

BIBLIOGRAPHY

BOOKS:

[a.1] Adaptive Signal Processing by Bernard Widrow and Samuel D. Stearns

[a.2] Adaptive Filter Theory 4e Simon Haykin and Thomas Kailath

[a.3] Digital Signal Processing: A Computer-Based Approach Second Edition Sanjit K. Mitra

[a.4] Fundamentals of adaptive filtering By Ali H. Sayed

[a.5] Adaptive Filtering: Algorithms and Practical Implementation 3e By Paulo S. R. Diniz

[a.6] Wireless Communications: Principles and Practice by Rappaport

[a.7] Wireless Communications Over Rapidly Time-Varying Channels by Franz Hlawatsch, Gerald M

[a.8] Verilog Hdl: A Guide to Digital Design and Synthesis By Samir Palnitkar

[a.9] Reconfigurable Computing: From FPGAs to Hardware/Software Co-design by João M. P. Cardoso

RESEARCH PAPERS:

[b.1] S. Qureshi, “Adaptive equalization,” Proc. IEEE,vol. 73, pp. 1349-1387, Sept. 1985.

[b.2] B. Petersen, D. Falconer, “Suppression of adjacent-channel, cochannel and intersymbol interference by equalizers and linear combiners,”IEEE Trans. Commun., vol. 42, pp. 3109-3118, Dec.1994.

[b.3] Real-Time Adaptive Signal Processing Development for US Navy Torpedoes Matthew A. Alexander, Robert Bond, William Payne, James C. Anderson, Nigel Lee, Francine Rayson MIT Lincoln Laboratory.

[b.4] A. K. Ziarani, “Extraction of Nonstationary Sinusoids,” Ph.D., Uni.Toronto, Toronto, Canada, 2002

[b.5] A.K Ziarani, A Konrad, A method of extraction of nonstationary sinusoids, Signal Processing, Volume 84, Issue 8, August 2004, Pages1323-1346,ISSN0165-1684, 10.1016/j.sigpro.2004.05.008.

[b.6] Ziarani, A.K.; Blumenfeld, I.M.; Konrad, A.; "Experimental verification of a novel method of extraction of nonstationary sinusoids," Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on , vol.1, no., pp. I- 455-8 vol.1, 4-7 Aug. 2002.

[b.7] McNamara, D.M.; Majo, J.P.; Ziarani, A.K.; , "Elimination of Power Line Interference on Telephone Cables Under Frequency-Varying Conditions," Instrumentation and Measurement, IEEE Transactions on , vol.57, no.2, pp.321-331, Feb. 2008 doi: 10.1109/TIM.2007.909496

[b.8] McNamara, D.M.; Majo, J.P.; Ziarani, A.K.; , "Power Line Interference Elimination on Telephone Loops," Instrumentation and Measurement Technology Conference, 2005. IMTC 2005. Proceedings of the IEEE , vol.2, no., pp.952-955, 16-19 May 2005 doi: 10.1109/IMTC.2005.1604278

[b.9] WANG Tian-lei, Developments in Adaptive Equalization Algorithm Research, Journal of Wuyi University [J],2009, 02(23),pp37-42

[b.10] DIAO Shu-lin; ZHONG Jian-bo, Analysis and Application of a Time Domain Adaptive Equalizer, Radio Engineering of China[J], 2009,09(39),pp44-47

[b.11] Shen Fu-min. Adaptive signal processing [M]. Xi'an: Xidian University Press, 2001

[b.12] ZHONG Hui-xiang, ZHENG Sha-sha, FENG Yue-ping, A Variable Step Size LMS Algorithm in Smart Antennas Based on Hyperbolic Tangent Function, JOURNAL OF JILIN UNIVERSITY (SCIENCE EDITION)[J].2008.5(46),pp935-939

[b.13] Wang Junfeng, A Variable Forgetting Factor RLS Adaptive Filtering Algorithm; International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, 2009

[b.14] Wang Junfeng, Zhang Bo, “Design of Adaptive Equalizer Based on Variable Step LMS Algorithm,” Proceedings of the Third International Symposium on Computer Science and Computational Technology(ISCSCT ’10) Jiaozuo, P. R. China, 14-15, pp. 256-258, August 2010.

[b.15] A.K. Ziarani, A. Konrad, A nonlinear adaptive method of elimination of power line interference in ECG signals, IEEE Trans.Biomed. Eng. 49 (6) (June 2002) 540–547.

[b.16] A.K. Ziarani, A. Konrad, A novel method of estimation of DPOAE signals, IEEE Trans. Biomed. Eng. 51 (5) (May 2004)864–868

[b.17] A.K. Ziarani, et al., A novel time-domain method of analysis of pulsed sine wave signals, IEEE Trans. Inst. Meas. 52 (3) (June2003) 809–814

[b.18] Georgi Iliev and Nikola Kasabov Department of Information Science, University of Otago CHANNEL EQUALIZATION USING ADAPTIVE FILTERING WITH AVERAGING

[b.19] Channel Equalizer Design Based on Wiener Filter and Least Mean Square Algorithms Hani Mehrpouyan.

[b.20] Fitzpatrick, M.J.; McNamara, D.M.; Ziarani, A.K.; "Real-time hearing assessment device based on distortion product otoacoustic emissions," Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on , vol.5, no., pp. v/625 - v/628 Vol. 5, 18-23 March 2005doi: 10.1109/ICASSP.2005.1416381

[b.21] McNamara, D. M.; Goli, A.; Ziarani, A. K.;"A novel approach for Doppler blood flow measurement," Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE, vol., no., pp.1883-1885, 20-25 Aug. 2008doi: 10.1109/IEMBS.2008.4649553.

[b.22] Ziarani, A.K.; Konrad, "An adaptive noise reduction technique," Circuits and Systems, 2001. MWSCAS 2001. Proceedings of the 44th IEEE 2001 Midwest Symposium on, vol.1, no., pp.251-254 vol.1, 2001.

WEBSITES:

[c.1] http://www.mathworks.com/

[c.2] http://www.wikipedia.org/

[c.3] http://see.stanford.edu/see/lecturelist.aspx?coll=523bbab2-dcc1-4b5a-b78f-4c9dc8c7cf7a

[c.4] http://www.xilinx.com/tools/sysgen.htm

[c.5] http://www.mathworks.com/help/comm/ug/equalization.html

[c.6] http://adaptivefilterapplications.blogspot.com/

[c.7] http://ucesp.ws.binghamton.edu/~xli/tutor/channel.htm

[c.8]http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-341-discrete-time-signal-processing-fall-2005/readings/

SOFTWARE & SOFTWARE MANUALS:

[d.1] MATLAB

- Getting Started – Math Works MATLAB

[d.2] Simulink

- Getting Started with Simulink – MathWorks
- MATLAB SIMULINK ® - Simulation and Model Based Design
- Using simulink to implement adaptive filters

[d.3] Xilinx system generator

- An introduction to Xilinx System Generator
- System Generator for DSP Getting Started Guide – Xilinx
- Xilinx System Generator for DSP User Guide
- Integrating Xilinx System Generator with Simulink HDL