



**FPGA-Based On-line Dynamic Tracking of Power System
Harmonics Using ADALINE Adaptive Filter**

Wang Jing Jing

**A Thesis Submitted in Partial Fulfillment of the Requirements for
the Degree of Master of Engineering in Electrical Engineering
Prince of Songkla University**

2015

Copyright of Prince of Songkla University

Thesis Title FPGA-Based On-line Dynamic Tracking of Power System
 Harmonics Using ADALINE Adaptive Filter

Author Miss Wang Jing Jing

Major Program Electrical Engineering

Major Advisor

.....
 (Assoc. Prof. Dr. Nattha Jindapetch)

Examining Committee:

.....Chairperson
 (Dr. Warit Wichakool)

.....Committee
 (Asst. Prof. Dr. Kusumal Chalermyanont)

.....Committee
 (Assoc. Prof. Dr. Pornchai Phukpattaranont)

.....Committee
 (Assoc. Prof. Dr. Wan Zuha Wan Hasan)

The Graduate School, Prince of Songkla University, has approved this thesis as partial fulfillment of the requirements for the Master of Engineering Degree in Electrical Engineering.

.....
 (Assoc. Prof. Dr. Teerapol Srichana)
 Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate's own investigations. Due acknowledgement has been made of any assistance received.

_____ Signature

(Assoc. Prof. Dr. Nattha Jindapetch)

Major Advisor

_____ Signature

(Wang Jing Jing)

Candidate

I hereby certify that this work has not been accepted in substance for any other degree, and is not being currently submitted in candidature for any degree.

_____ Signature

(Wang Jing Jing)

Candidate

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to Assoc. Prof. Dr. Nattha Jindapetch for her encouragement, teaching and help.

Also thanks all the teachers in Faculty of Engineering who once offered me valuable courses and advices during my study. Specially thanks committees Dr. Warit Wichakool, Asst. Prof. Dr. Kusumal Chalermyanont, Assoc. Prof. Dr. Pornchai Phukpattaranont and Assoc. Prof. Dr. Wan Zuha Wan Hasan for coming to be the committee in my graduation thesis presentation.

Then, I feel grateful to all of my friends in Department of Electrical Engineering for their kindness and help.

Wang Jing Jing

Contents

Chapter 1. Introduction	1
1.1 Background and Problem Statement.....	1
1.2 Objectives	1
1.3 Outcomes	2
1.4 Overview of the Methods for Tracking the Harmonic Components	2
1.4.1 Fast Fourier Transform (FFT) [1].....	3
1.4.2 Kalman Filter (KF)	7
1.4.3 Artificial Neural Network.....	8
1.4.4 Summary	12
1.5 Research Methodology	13
1.6 Research Scopes	14
1.7 Research Plan.....	14
Chapter 2. Theory	15
2.1. Neuron Model	15
2.2 ADALINE Network.....	17
2.2.1 ADALINE Network.....	17
2.2.2 Mean Square Error	17
2.2.3 LMS Algorithm.....	19
2.3 Xilinx System Generator	21
2.4 Lyapunov Function	22
2.4.1 Theory of Lyapunov	22
2.4.2 Lyapunov Advantages	22
Chapter 3. Research Methodology.....	24
3.1 ADALINE.....	25
3.2 Tracking of the Amplitudes of the Fundamental Signal and Harmonics	27
3.3 Tracking of the Fundamental Frequency	29
3.4 Adaptive Variation of the Learning Parameters α	29
3.5 MATLAB Code	31
3.6 FPGA Implementation.....	34
Chapter 4. Results and Discussions	38

4.1 MATLAB Simulation Results	39
4.2 FPGA Implementation results	45
4.3 On-line Dynamic Tracking	49
4.4 Hardware-in-the-Loop Verification.....	52
Chapter 5. Conclusions, Problems and Suggestions.....	57
5.1 Conclusions.....	57
5.2 Problems and Suggestions	57
References.....	59

List of Figures

Figure 1- 1	Aliasing ($f_1=60$ Hz, $f_2=1080$ Hz, $f_s=1.92$ kHz) [1].	3
Figure 1- 2	Avoid aliasing with increasing the sampling frequency ($f_1=60$ Hz, $f_2=1080$ Hz, $f_s=3,84$ kHz) [1].	4
Figure 1- 3	Picket-Fence Effect ($f_1=60$ Hz, $f_2=75$ Hz, $f_2=150$ Hz, $f_s=1.92$ kHz, $N=32$) [1].	5
Figure 1- 4	Picket-Fence Effect ($f_1=60$ Hz, $f_2=75$ Hz, $f_2=150$ Hz, $f_s=1.92$ kHz, $N=64$) [1].	5
Figure 1- 5	Avoid of Picket-Fence Effect ($f_1=60$ Hz, $f_2=75$ Hz, $f_2=150$ Hz, $f_s=1.92$ kHz, $N=128$) [1].	6
Figure 1- 6	Leakage ($f=55$ Hz, $N=32$) [1].	6
Figure 1- 7	Flowchart of the overall process of methodology works.	13
Figure 2- 1	Single-Input Neuron.	15
Figure 2- 2	Multiple-Input Neuron.	16
Figure 2- 3	Linear Transfer Function.	16
Figure 2- 4	ADALINE network.	17
Figure 2- 5	CORDIC SINCOS Xilinx Block.	21
Figure 2- 6	MCode Xilinx Block.	21
Figure 3- 1	Flowchart of research methodology.	24
Figure 3- 2	Block diagram of the ADALINE for estimating 3-phase voltages or currents (3-Phase ADALINE) [6].	26
Figure 3- 3	Block diagram of the ADALINE for estimating harmonic components and fundamental frequency.	27
Figure 3- 4	Data flow of the proposed ADALINE.	27
Figure 3- 5	Total Xilinx design for tracking fundamental and harmonics.	34
Figure 3- 6	Adaptive model for update learning parameters.	35
Figure 3- 7	Model for generating dynamic input vectors (Sin, Cos).	35
Figure 3- 8	Model for accumulating weight vectors.	36
Figure 3- 9	Xilinx design for tracking the amplitude.	36
Figure 3- 10	Xilinx design for tracking the fundamental frequency.	37
Figure 3- 11	Xilinx design for tracking the fundamental frequency.	37

Figure 4- 1	Flowchart of experiments.....	38
Figure 4- 2	Distorted Signal (Fundamental,5 th , 7 th , 11 th)(time=2T).	39
Figure 4- 3	Amplitude Tracking of Fundamental signal and harmonics (5 th , 7 th , 11 th) in MATLAB simulation (comparison case).	39
Figure 4- 4	Tracking of Fundamental Frequency in MATLAB simulation (comparison case).	40
Figure 4- 5	Tracking of Mean Square Error in MATLAB simulation (comparison case).	40
Figure 4- 6	Amplitude Tracking of Fundamental signal and harmonics (5 th , 7 th , 11 th).	42
Figure 4- 7	Tracking of Fundamental Frequency..	42
Figure 4- 8	Tracking of Mean Square Error.	42
Figure 4- 9	Amplitude Tracking of Fundamental signal and harmonics (5 th , 7 th , 11 th).	44
Figure 4-10	Tracking of Fundamental Frequency..	44
Figure 4-11	Tracking of Mean Square Error.	44
Figure 4- 12	Total Xilinx design for tracking fundamental and harmonics.	45
Figure 4- 13	Xilinx Simulation in Case 1.	47
Figure 4- 14	Xilinx Simulation in Case 2.	48
Figure 4- 15	Register with reset port.	49
Figure 4- 16	Case a (Reference fundamental frequency is 50.2 HZ).	50
Figure 4- 17	Case b (Reference fundamental frequency is 50.15 HZ).	50
Figure 4- 18	Case c (Reference fundamental frequency is 50.1 HZ).	50
Figure 4- 19	Case d (Reference fundamental frequency is 50.05 HZ).	50
Figure 4- 20	Case e (Reference fundamental frequency is 49.95 HZ).	50
Figure 4- 21	Case f (Reference fundamental frequency is 49.9 HZ).	51
Figure 4- 22	Case g (Reference fundamental frequency is 49.85 HZ).	51
Figure 4- 23	Case h (Reference fundamental frequency is 49.8 HZ).	51
Figure 4- 24	Complete Design Ready for the Hardware-in-the-Loop Simulation.	52
Figure 4- 25	Model design of the distorted signal.	52

Figure 4- 26	Simulation result showing simlator output on top and hardware output at the bottom (Amplitude Tracking).....	53
Figure 4- 27	Simulation result showing simlator output on top and hardware output at the bottom (Frequency Tracking).....	53
Figure 4- 28	Simulation result showing simlator output on top and hardware output at the bottom (Error).....	54
Figure 4- 29	Simulation result showing simlator output on top and hardware output at the bottom (Amplitude Tracking).....	54
Figure 4- 30	Simulation result showing simlator output on top and hardware output at the bottom (Frequency Tracking).....	55
Figure 4- 31	Simulation result showing simlator output on top and hardware output at the bottom (Error).....	55

List of Tables

Table 1-1	Summary of Literature Review.	12
Table 1-2	Research Plan.	14
Table 4-1	Convergence Time.	41
Table 4-2	Convergence Time in case 1 and case 2.....	45
Table 4-3	Device Utilization Summary (Xilinx Spartan3E-1200 FPGA).....	46
Table 4-4	Static Timing Analysis Report (Xilinx Spartan3E-1200 FPGA).	46
Table 4-5	Device Utilization Summary (xc6slx100-3fgg484 FPGA).....	46
Table 4-6	Static Timing Analysis Report (xc6slx100-3fgg484 FPGA).	46

Chapter 1. Introduction

1.1 Background and Problem Statement

In the present AC power system, there is a obvious increase in harmonic currents and voltages. This is primarily due to the utilization of the non-linear loads in the power system.

The incidence of harmonic is not often, but there is a strong necessity to be aware of harmonic issues. We all know that the more harmonics the more harmful. The harmonics in a power system will produce additional harmonic losses, reducing the efficiency of electrical equipment. Harmonics can also increase heating in the equipment, conductors and even can cause a fire. Harmonics produce mechanical vibration on the electric motor and physical damage to aging cables insulation. So reduction of harmonics is considered desirable.

Active power-line filtering is conventionally performed by injecting equal-but-opposite of the distortion into the line. The phenomenon motivated us to find some method to tracking the harmonic components so then eliminate them. Then it will help to increase power system reliability. In the usual approach, estimation of harmonic is done by using the FFT (Fast Fourier Transform) [1], Kalman Filter (KF) [2-3] and Artificial Neural Network (ANN) [4-7]. However, there are several performance limitations in the FFT application. If we fail to satisfy the certain conditions will result in aliasing, picket fence effects and leakage. In the KF approach, the state equations, measurement equations, and covariance matrices need to be correctly defined. In the ANN, it requires too much data to get trained and may lead to inaccurate results due to the random-like behavior and large variations in the harmonic components. In order to overcome the drawbacks above, we proposed a method based on the adaptive linear neurons (ADALINE) in this research.

1.2 Objectives

- 1) To apply ADALINE neural network to on-line tracking signal and harmonic components.
- 2) To improve on-line harmonics tracking system for better speed and convergence performance.

- 3) To implement the ADALINE algorithm in FPGA to track a signal and harmonic components in the power system.

1.3 Outcomes

- 1) Obtain an ADALINE filter for tracking the harmonic components and the fundamental frequency with a faster and more accurate convergence speed.
- 2) Obtain an FPGA prototyping for tracking the fundamental and the harmonic components in the power system.

1.4 Overview of the Methods for Tracking the Harmonic Components

The technology of electricity rectification has been widely used in the power system. However, the rectification device is a non-sinusoidal electrical equipment. The voltage through the rectifier turned to distortion. Harmonics generated by the rectifier accounts for nearly 40% of all the harmonics, which is the largest source of harmonics.

The complex current waveform can be decomposed into a series of sinusoids with fundamental frequency and integer multiples of the fundamental frequency by Fourier series analysis. For the purpose, the distorted signal can be described as follows:

$$y(t) = \sum_{l=1}^{\infty} X_l \sin\left(\frac{l2\pi t}{T} + \varphi_l\right) + \epsilon(t), \quad (1.1)$$

where

$y(t)$ is the time domain function,

l is the harmonic number (only odd values of l are required),

X_l is the amplitude of the l^{th} harmonic component,

φ_l is the phase of the l^{th} harmonic component,

T is the length of a cycle in seconds,

and $\epsilon(t)$ is the random noise and the DC decaying components.

The common algorithm for tracking the harmonic components in the literature are based on the FFT, the KF and ANN.

1.4.1 Fast Fourier Transform (FFT) [1]

The Fast Fourier Transform (FFT) is always used as an analysis tool. However, there are three serious problems in the FFT application, i.e. aliasing, picket-fence effect, and leakage.

1.4.1.1 Aliasing

If the sampling rate is too low. The high-frequency components of a time function can translate into low frequencies. And this is the phenomenon of aliasing. This problem may still be present even if the highest frequency component is higher than $f_s/2$.

Assume that we have a known input signal $x(t)$. The sampling frequency of the signal is 1920 Hz. The signal consists of fundamental signal with frequency of 60 Hz and a sinusoid of 1080 Hz. The 1080 Hz is higher than $f_s/2$, and it resulted in the amplitude of 840 Hz is nonzero, as shown in Figure 1-1. Then we changed the sampling frequency to 3840 Hz. Then there is no aliasing and the spectrum is shown in Figure 1-2 [1].

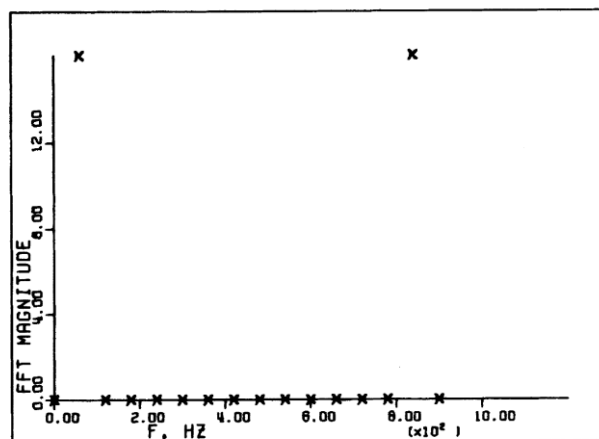


Figure 1- 1 Aliasing ($f_1=60$ Hz, $f_2=1080$ Hz, $f_s=1.92$ kHz) [1].

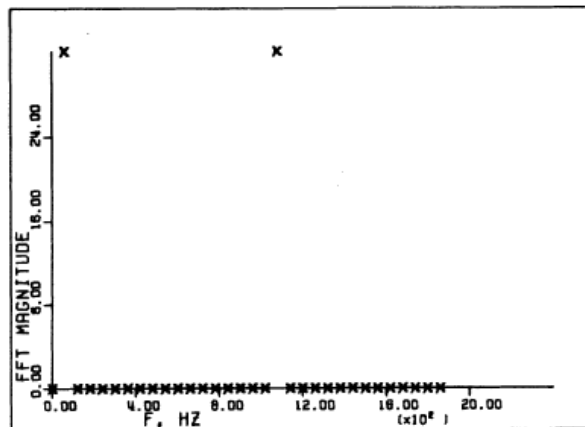


Figure 1- 2 Avoid aliasing with increasing the sampling frequency ($f_1=60$ Hz, $f_2=1080$ Hz, $f_s=3,84$ kHz) [1].

1.4.1.2 Picket-Fence Effect

The picket-fence effect occurs if the analyzed waveform includes a frequency which is not an integer times the fundamental frequency. A frequency lying between the n^{th} and the $(n+1)^{\text{th}}$ harmonics affects the magnitudes of the n^{th} and the $(n+1)^{\text{th}}$ harmonics and secondarily the magnitude of all other harmonics. Also, this frequency can cause leakage which in turn may cause pseudoaliasing.

The picket-fence effect is illustrated by adding together three sinusoidal signals. The first has the fundamental frequency 60 Hz with a peak amplitude of unity; the second has a frequency of 75 Hz with a peak amplitude of $1/8$; and the third has a frequency of 150 Hz with a peak amplitude of $1/8$. To begin, the sampling frequency was 1092 Hz. Figure 1-3 shows the frequency spectrum. From this, it is obvious that the three frequencies cannot be resolved since the frequency sample interval is 60 Hz. The 75 and the 150 Hz components do not have an integer number of cycles in T ; therefore, leakage occurs which in turn causes pseudoaliasing.

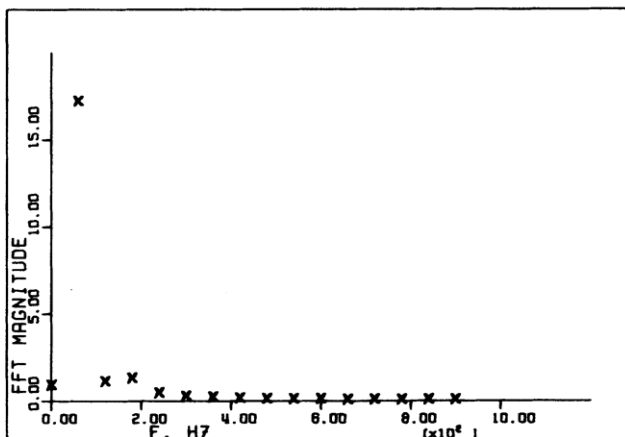


Figure 1-3 Picket-Fence Effect ($f_1=60$ Hz, $f_2=75$ Hz, $f_2=150$ Hz, $f_s=1.92$ kHz, $N=32$) [1].

The sampling frequency was left the same, but the number of points was increased to 64. The results are shown in Figure 1-4. It is obvious that the 60 Hz and the 150 Hz components can be resolved since the frequency sample interval is 30 Hz. Since the 75 Hz component does not have an integer number of cycles with 30 Hz, it will cause leakage.

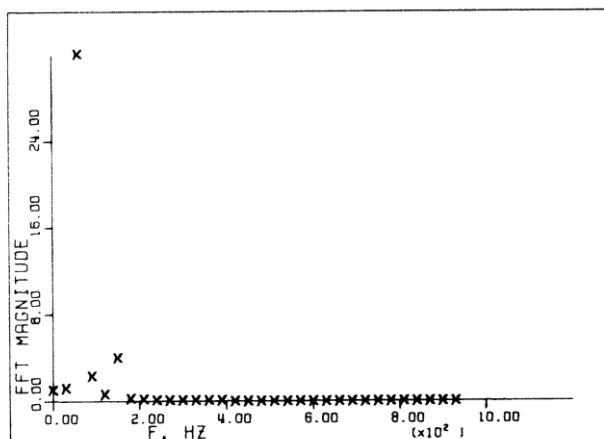


Figure 1-4 Picket-Fence Effect ($f_1=60$ Hz, $f_2=75$ Hz, $f_2=150$ Hz, $f_s=1.92$ kHz, $N=64$) [1].

The same sampling frequency was used with 128 points and the results is shown in Figure 1-5. Now, the frequency sample interval is 15 Hz and the 75 Hz component has an integer times of 15 Hz, also no aliasing was detected.

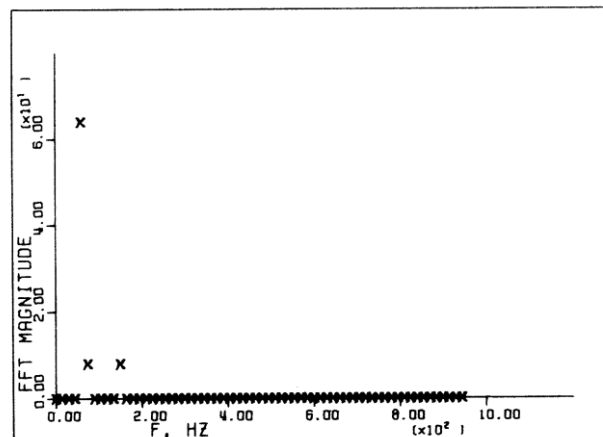


Figure 1- 5 Avoid of Picket-Fence Effect ($f_1=60$ Hz, $f_2=75$ Hz, $f_2=150$ Hz, $f_s=1.92$ kHz, $N=128$) [1].

Therefore, the frequency component must be an integer multiple of the frequency sample interval in the FFT [1].

1.4.1.3 Leakage

Due to the truncation of the time sequence such that a fraction of a cycle exists in the waveform that is subjected to the FFT. "leakage" will spread the energy from one frequency into adjacent ones.

Use the same the sampling frequency, but decrease the frequency of a 60 Hz unity amplitude sine waveform by 1 Hz each time down to 55 Hz. The sampling frequency was 1.92 kHz. A frequency of 58 Hz aliasing was detected by the computer program. Figure 1-6 shows the spectrum of the 55 Hz waveform sampled at 1.92 kHz [1].

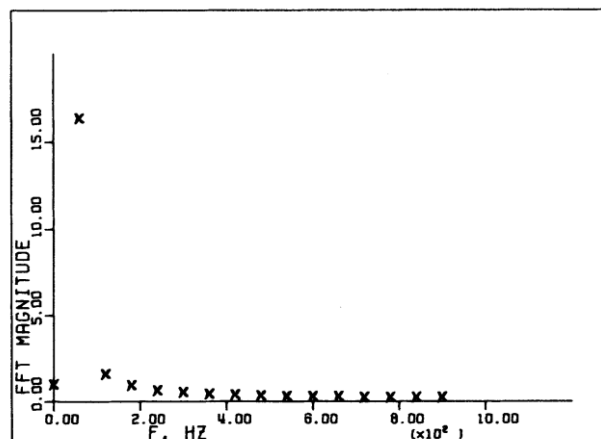


Figure 1- 6 Leakage ($f=55$ Hz, $N=32$) [1].

So, there are many limitations in the FFT application. The performance limitation in the FFT application [1]:

- 1) The signal must be stationary and periodic;
- 2) The highest frequency of the signal is less than $f_s/2$;
- 3) The window length of data must be an exact integer multiple of power-frequency cycles;
- 4) Each frequency in the signal is an integer multiple of the fundamental frequency.

1.4.2 Kalman Filter (KF)

Another method for estimating harmonic components is KF. Due to KF is a time-domain stochastic optimal estimator. It's suitable for tracking time-varying parameters of harmonic signals.

1.4.2.1 An adaptive Kalman filter for dynamic estimation of harmonic signals [2]

The standard KF algorithm can be used to estimate the state variable vector i.e. for tracking the harmonic components.

However, the performance of KF algorithm is strongly dependent on the a priori information of the process and measurement noise. On the other hand, KF often suffers from “dropping off” which make it lose the ability to match abrupt parameter changes. Another major problem of KF application is that we should define the noise covariance matrix Q and the measurement noise variance value r_k carefully.

There are two adaptive techniques for adjusting the KF. The first basic idea to improve the steady-state performance of KF is to find a way for estimating the actual value of measurement variance r_k additionally. Another usage of the correlation function of the innovation process is to perform an optimality and stationary test for the KF.

Although the estimation of measurement variance r_k can improve the stationary performance of the KF, the question of filter “dropping off” can not be avoided. Once we detected a transient, we should assume a large estimation error covariance P_k , i.e. to reset the value of P_k to the predefined value. An obvious error covariance will increase the sensitivity of the KF, and the new parameter variations can be matched quickly. By this way a drop off of the used filter can be solved. The

recursive algorithm which based on the adaptive KF is able to detect abrupt changes of harmonic parameters [2].

1.4.2.2 A digital recursive measurement scheme for on-line tracking of power system harmonics [3]

An optimal measurement scheme for tracking the harmonics in power system voltage and current waveforms does not require an integer number of samples in an integer number of cycles. It is not limited to stationary signals, and it can track harmonics with time-varying amplitudes. The scheme is based on KF theory for the optimal estimation of the parameters of time-varying harmonics. The recursive optimal estimator based on KF is suitable for on-line applications. The parameters and a measurement equation that relates the discrete measurement to the state variables which to be estimated require a state variable model. The KF can be applied for any number of samples over a half cycle. The KF algorithm would track dynamic magnitude after the initialization stage (half a cycle).

The paper presented the basic assumptions in the FFT algorithms and the principles of KF in tracking the time variation of power system harmonics. The pitfalls in the FFT are illustrated by two examples. It is obvious that the KF algorithm is more accurate but is not sensitive to a certain sampling frequency. If we use the KF gain vector as the time-varying magnitudes, the estimator can make the tracking of harmonics come true [3].

1.4.3 Artificial Neural Network

The artificial neural network (ANN) technique requires to train a lot of data. And it is based on backpropagation. The neural method of estimating harmonics uses the optimization technique to minimize error.

1.4.3.1 An adaptive linear combiner for on-line tracking of power system harmonics [4]

In the paper, a Fourier linear combiner is used for adaptive estimation of harmonics. The linear combiner using the ADALINE. The weight vector of the ADALINE generates the Fourier coefficients of the signal using a nonlinear weight adjustment algorithm.

The approach is essentially different from the backpropagation method and allows well control the stability and speed of convergence by appropriate choice of parameters of the error difference equation. Further, the estimation method is highly adaptive and is capable of tracking the variations of amplitudes of the harmonic components. The performance of the method is compared with the KF, it shown the superiority in on-line tracking power system harmonics.

The amplitude vector of the ADALINE is updated with using Widrow-Hoff delta rule as

$$W(k+1) = W(k) + \alpha \frac{e(k)X(k)}{X^T(k)X(k)}. \quad (1.2)$$

In order to produces a fast convergence and introduces nonlinearity to the learning technique. Another adaptation algorithm for the weight adaptation of ADALINE is adapted as

$$W(k+1) = w(k) + \frac{\alpha e(k)\theta_k(X)}{\lambda + X^T\theta_k(X)}, \quad (1.3)$$

where the value of λ is chosen very nearly zero ($\lambda = 0.01$).

The method is based on the weight vector estimation of an ADALINE which based on the Least Mean Squares (LMS). An adaptive learning step α is used to bring a faster convergence and noise rejection in tracking the harmonics.

The results shows the good accuracy and convergence speed of the method when compare with the DFT. Furthermore, the adaptive algorithm is more suitable for tracking harmonics with time varying amplitude [4].

1.4.3.2 Modular approach to active power-line harmonic filtering [5]

The method is based on the extraction of harmonic components with using an ADALINE. A neural network is a system of inputs and outputs and many processing elements called neurons. The weights are adjusted when we are training the neural network. The weight vector is updated as follow

$$W(k+1) = W(k) + \alpha \frac{e(k)X(k)}{X^T(k)X(k)}. \quad (1.4)$$

The main advantages of ADALINE are speed and noise rejection. It is proved to be superior to KF technique in tracking the magnitudes and phases of the harmonics. It will be capable of tracking the harmonic distortion in a time-vary in a nonlinear load

environment. It improves the reliability of the active power filters and it saves the costs compared to the conventional method of filtering all the harmonics using converters. However, the algorithm presented in these two papers are adaptive and we should assume that the fundamental frequency is known apriori [5].

1.4.3.3 On-line tracking and mitigation of power system harmonics using ADALINE-based active power filter system [6]

The paper presents an improved method for online tracking of amplitudes which combines the fundamental frequency tracking with an ADALINE. The proposed method used two ADALINES to process the signals obtained from the power-line. The first ADALINE extracts the harmonic components of the distorted signal and the second ADALINE estimates the fundamental frequency of the voltage signal.

In the original ADALINE algorithm, it is assumed that the fundamental frequency is known. In this paper, the ADALINE algorithm has been modified to track both the fundamental frequency and the harmonic amplitudes.

The theoretical expectations are verified by digital simulation using EMTDC (Electric Magnetic Transient of Direct Current) simulation package. The proposed scheme exhibits better performance of noise rejection, fast and accurate tracking of harmonic components and system frequency of the signal with noise and decaying DC components when compared with KF and FFT approaches. [6].

1.4.3.4 Neural network for estimation of harmonic components in a power system [7]

In this paper, the author proposed two neural optimization approaches to find the optimum point that minimizes the mean square error. One is based on the measurements done in a time window. It uses massive parallel processing of the information taken within one time window, and delivers the solution in real time. The other one uses the neural optimization approach to find the optimum point then built the appropriate circuitry that minimizes the instantaneous mean square error.

The first solution used the principle of parallel processing of many samples of the line signal. Thus the speed of estimation is very high. At the same time, the complexity of circuitry is proportional to the number of samples. The higher this

number is, the more parallel branches of the circuit are needed, and the higher is the cost of implementation of the circuit.

The second approach has already been simulated and investigated on microcomputer and the results have proved its capabilities of estimating the parameters of harmonics and the adaptation time kept below one period T of the fundamental harmonic in the power system. But the adaptive way of estimation is slower [7].

1.4.4 Summary

Table 1-1 Summary of Literature Review.

Algorithm	Advantage	Disadvantage	
FFT [1] [14] [15]	It converts time to frequency, we can see the spectrum directly.	<p>The signal should be stationary and periodic.</p> <p>The sampling frequency must be twice greater than the signal.</p> <p>The window length of data must be an exact integer multiple of power-frequency cycles.</p> <p>It causes computational error and may lead to inaccurate results if the signal is contaminated by noise or the DC component.</p>	A
KF [2] [3] [13]	The Kalman filter technique estimates the harmonic components by utilizing a small number of samples in a relatively short time.	<p>The state equations, measurement equations, and covariance matrices need to be correctly defined.</p> <p>The proposed KF in the literature can not track the fundamental frequency.</p>	B
ANN [4] [5] [6] [7]	The delta rule which minimizes the mean square error between the signal sample and the estimate signal over all the time, can simplified the calculation.	<p>Requires too much data to get trained and may lead to inaccurate results due to the random-like behavior and large variations in the harmonic components.</p> <p>Weight adaption algorithm is not perfect.</p>	B C
ADALIN E [9] [12] [11] [25] [20][22]	<p>Can on-line tracking the harmonic components.</p> <p>There is no need to training the data.</p>	<p>Fundamental frequency should be fixed.</p> <p>The convergence of tracking the amplitude is slow.</p>	B C

A. Analysis studying

B. MATLAB Simulation

C. EMTDC simulation (Electric Magnetic Transient of Direct Current)

1.5 Research Methodology

In this research, an ADALINE based on the algorithm least-mean-square (LMS) shows better dynamic tracking capability of harmonic amplitudes and the fundamental frequency in the power system. The LMS-based ADALINE is very easy and simple to implement on FPGA. However, there is a trade-off between the convergence speed and the state-steady-misadjustment error. So, in order to balance the trade-off, we also proposed an appropriate method for adaptive updating the learning parameters.

Figure1-7 shows the flowchart of the overall process of methodology works. We verified the learning parameters by simulated the LMS algorithm on MATLAB, then the design has been done by using Xilinx System Generator in Simulink. Finally, we implemented the proposed method on a Xilinx Spartan3E-1200 FPGA (Field Programmable Gate Array). The hardware-in-the loop verification results show that the proposed ADALINE adaptive filter can accurately estimate the dynamic harmonic amplitudes and adaptively track the dynamic fundamental frequency in the present AC power system. Moreover, the adaptive learning parameters bring the faster and more accurate convergence.

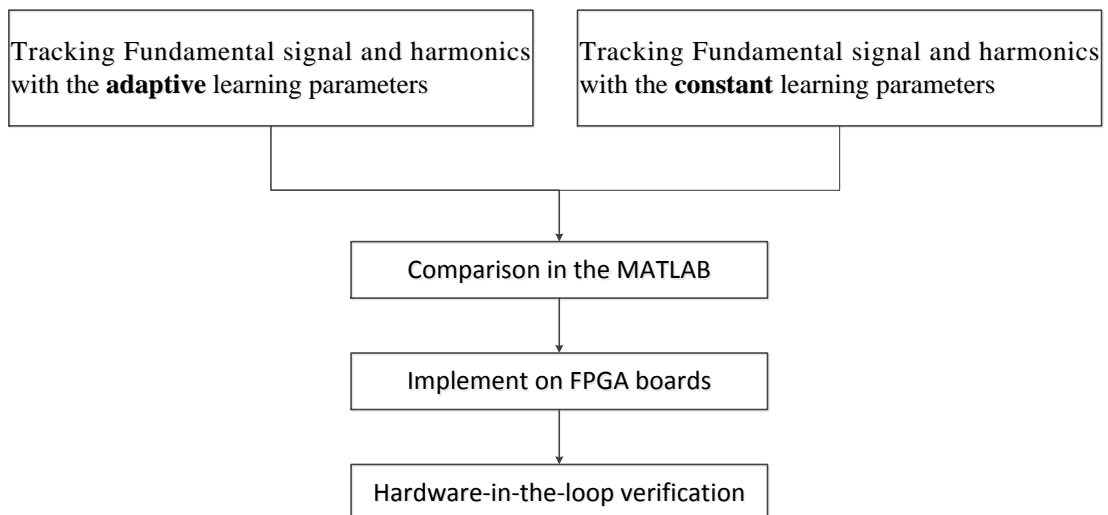


Figure 1- 7 Flowchart of the overall process of methodology works.

1.6 Research Scopes

- 1) Assume that the fundamental frequency and the amplitude of the fundamental voltage or current is known apriori.
- 2) It focuses on tracking the fundamental frequency and the amplitude of power system whose possible deviation is in a small allowable range.
- 3) Due to accumulate in the FPGA design, we have to scale the fix-point number. So, we may get some errors between the estimate results and the actual situation.

1.7 Research Plan

Table 1-2 Research Plan.

Activities	Period					
	Month (2014)				Month (2015)	
	1-3	4-6	7-11	10-12	1-3	3-6
1. Study Verilog Hardware Description Language						
2. Study a commercial software (Xilinx System Generator)						
3. Study neural network						
4. Design a algorithm for the research						
5. Simulation in MATLAB and improve the algorithm						
6. Implement the algorithm on the FPGA and improve it						
7. Analyze and Conclusion						
8. Report						

Chapter 2. Theory

2.1. Neuron Model

We already have a rudimentary understanding of biological neural network, so people constructed some artificial “neurons” and train them to perform certain functions, and they are artificial neural networks. The history of artificial neural networks is filled with colorful, and they are creatively utilized in many different fields. Application areas include the system identification and control, quantum chemistry, pattern recognition, visualization, e-mail spam filtering and so on.

A single-input neuron is shown in Figure 2-1. The scalar input p is multiplied by the scalar weight w to form wp . The other input, 1, is multiplied by a bias b and then passed to the summer. The summer output n goes into a transfer function f which produces the scalar neuron output a .

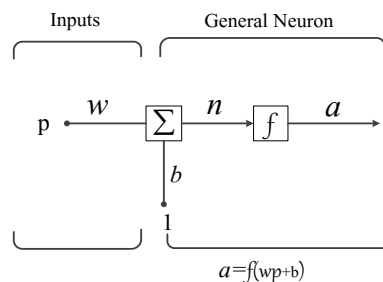


Figure 2- 1 Single-Input Neuron.

The neuron output is calculated as

$$a = f(wp + b). \quad (2.1)$$

Typically, a neuron has more than one input. A neuron with R inputs is shown in Figure 2-2. The individual inputs $p_1, p_2, p_3, \dots, p_R$ are each weighted by corresponding elements w_1, w_2, \dots, w_R of the weight matrix W .

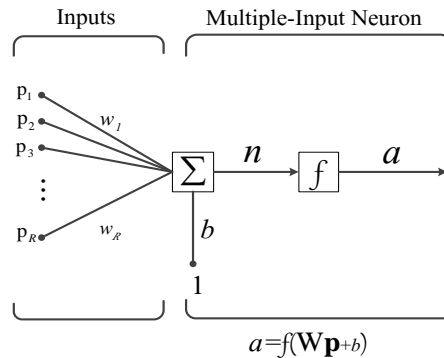


Figure 2- 2 Multiple-Input Neuron.

The neuron has a bias b , the output expression can be written:

$$a = f(n) = f(w_1p_1 + w_2p_2 + \dots + w_Rp_R + b) = f(W\mathbf{p} + b). \tag{2.2}$$

Fortunately, neural networks can often be described with matrices. The actual output depends on the particular transfer function that been chosen. We can also see the bias as a constant input of 1. However, if we do not want to have a bias in a particular neuron, it can be omitted.

Note that weight vectors and bias are both adjustable scalar parameters of the neuron. Typically the transfer function is chosen by the designer. And we have different transfer functions for different purposes.

The transfer function in Figure 2-1 may be a liner or a nonlinear function. A particular transfer function is chosen to satisfy some specifications of the problem so that the neuron is attempting to solve.

The output of a linear transfer function is equal to its input:

$$a = n = wp + b. \tag{2.3}$$

As illustrated in Figure 2-3, neurons with this transfer function are used in the ADALINE networks, which are discussed in this thesis.

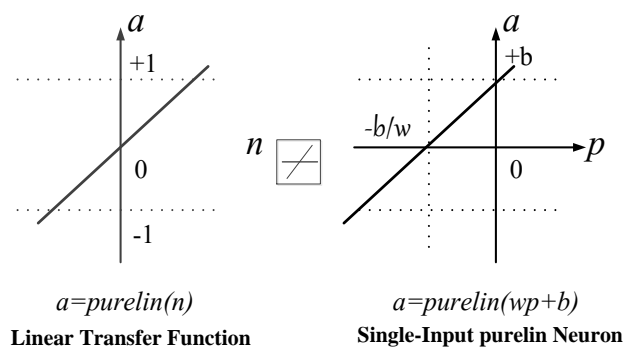


Figure 2- 3 Linear Transfer Function.

The output (a) versus input (p) characteristic of a single-input linear neuron with a bias is shown on the right side of Figure 2-3.

2.2 ADALINE Network

2.2.1 ADALINE Network

The ADALINE network is illustrated in Figure 2-4.

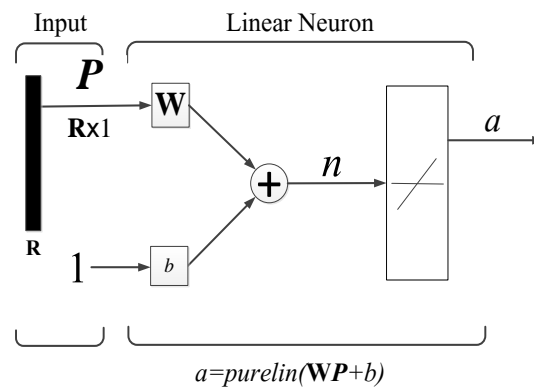


Figure 2- 4 ADALINE network.

The output of the network is given by

$$a = \text{purelin}(\mathbf{W}\mathbf{P} + b) = \mathbf{W}\mathbf{P} + b. \quad (2.4)$$

2.2.2 Mean Square Error

The LMS algorithm is an example of supervised training. The learning rule is provided with a series of examples of expected network behavior:

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\}.$$

Where p_q is an input to the network, and t_q is the corresponding target output. Once each input is fed into the network, the output of the network will be in comparison with the target.

In the LMS algorithm, the weights and the biases of the ADALINE is adjusted to minimize the mean square error, in which the error of the target output differs from the network output.

To simplify our development, we will lump all the parameters and bias adjusted into one vector:

$$X = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}. \quad (2.5)$$

Similarly, we include the bias input “1” as a component of the input vector:

$$Z = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}. \quad (2.6)$$

Now the network outputs are often written in the form:

$$a = \mathbf{w}^T \mathbf{p} + b = X^T Z. \quad (2.7)$$

This allows us to conveniently write out an expression for the mean square error of ADALINE network, which we used it as the performance index in the neural network:

$$F(x) = E[e^2] = E[(t - a)^2] = E[(t - X^T Z)^2]. \quad (2.8)$$

The expression is expanded as follows:

$$\begin{aligned} F(x) &= E[t^2 - 2tX^T Z + X^T Z Z^T X] \\ &= E[t^2] - 2X^T E[tZ] + X^T E[ZZ^T] X. \end{aligned} \quad (2.9)$$

It can be written in the convenient form:

$$F(x) = c - 2X^T h + X^T R X, \quad (2.10)$$

where

$$c = E[t^2], h = E[tZ] \text{ and } R = E[ZZ^T]. \quad (2.11)$$

Here the vector h gives the cross-correlation between the input vector and its associated target, while R is the input correlation matrix. The diagonal elements of this matrix are equal to the mean square values of the elements of the input vectors.

Compare equation (2.9) with the general form of the quadratic function:

$$F(x) = c + d^T x + \frac{1}{2} x^T A x. \quad (2.12)$$

We can see the mean square error performance index for the ADALINE network is a quadratic function, where

$$d = -2h \text{ and } A = 2R. \quad (2.13)$$

We know the characteristics of the quadratic function depend primarily on the Hessian matrix A . For example, if the eigenvalues of the Hessian are all positive, then the function will have one unique global minimum.

In this case the Hessian matrix is twice the correlation matrix R , and it can be shown that all correlation matrices are either positive definite or positive semi-definite, which means that they can never have negative eigenvalues. We are left with two possibilities. If the correlation matrix has only positive eigenvalues, the performance index will have one unique global minimum. If the correlation matrix has some zero eigenvalues, the performance index will either have a weak minimum or no minimum, depending on the vector $d = -2h$.

Now let's locate the stationary point of the performance index. From our previous discussion of quadratic function we know that the gradient is

$$\nabla F(\mathbf{x}) = \nabla \left(c + d^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T A \mathbf{x} \right) = d + A \mathbf{x} = -2h + 2R\mathbf{x}. \quad (2.14)$$

The stationary point of $F(\mathbf{x})$ can be found by setting the gradient equal to zero:

$$-2h + 2R\mathbf{x} = 0. \quad (2.15)$$

Therefore, if the correlation matrix is positive definite there will be a unique stationary point, which will be a strong minimum:

$$\mathbf{x}^* = R^{-1}h. \quad (2.16)$$

The existence of a unique solution depends only on the correlation matrix R . Therefore the characteristics of the input vectors determine whether or not a unique solution exists.

2.2.3 LMS Algorithm

Now that we have analyzed our performance index, the next step is to design an algorithm to locate the minimum point. If we could calculate the statistical quantities h and R , we could find the minimum point directly from equation (2.16). If we did not want to calculate the inverse of R , we could use the steepest descent algorithm with the gradient calculated from equation (2.14). In general, however, it is not desirable or convenient to calculate h and R . For this reason, we will use an approximate steepest descent algorithm, in which we use an estimated gradient.

The key insight of Widrow and Hoff was that they could estimate the mean square error $F(\mathbf{x})$ by

$$F(\mathbf{x}) = (t(k) - a(k))^2 = e^2(k), \quad (2.17)$$

where the expectation of the squared error has been replaced by the squared error at iteration k . Then, we have a gradient estimate of the form at each iteration

$$\nabla F(\mathbf{x}) = \nabla e^2(k). \quad (2.18)$$

The first R elements of $\nabla e^2(k)$ are derivatives with respect to the network weights, while the $(R+1)^{\text{st}}$ element is the derivative with respect to the bias. Thus we have

$$[\nabla e^2(k)]_j = \frac{\partial e^2(k)}{\partial w_j} = 2e(k) \frac{\partial e(k)}{\partial w_j} \quad (\text{for } j = 1, 2, \dots, R) \quad (2.19)$$

and

$$[\nabla e^2(k)]_{R+1} = \frac{\partial e^2(k)}{\partial b} = 2e(k) \frac{\partial e(k)}{\partial b}. \quad (2.20)$$

Now let's consider the partial derivative terms at the ends of these equations.

First evaluate the partial derivative of $e(k)$ with respect to the weight w_j :

$$\begin{aligned} \frac{\partial e(k)}{\partial w_j} &= \frac{\partial [t(k) - a(k)]}{\partial w_j} = \frac{\partial}{\partial w_j} [t(k) - (\mathbf{w}^T \mathbf{p}(k) + b)] \\ &= \frac{\partial}{\partial w_j} \left[t(k) - \left(\sum_{i=1}^R w_i p_i(k) + b \right) \right], \end{aligned} \quad (2.21)$$

where $p_i(k)$ is the i th element of the input vector at the k^{th} iteration. This simplifies to

$$\frac{\partial e(k)}{\partial w_j} = -p_j(k). \quad (2.22)$$

In a similar way, we can obtain the final element of the gradient:

$$\frac{\partial e(k)}{\partial b} = -1. \quad (2.23)$$

Note that $p_j(k)$ and 1 are the elements of the input vector also the gradient of the squares error at iteration k can be written as

$$\nabla F(\mathbf{x}) = \nabla e^2(k) = -2e(k)\mathbf{Z}(k). \quad (2.24)$$

To calculate this approximate gradient, we need only multiply the error and the input. This approximation of $\nabla F(\mathbf{x})$ can now be used in the steepest descent algorithm. The steepest descent algorithm which with learning rate is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}. \quad (2.25)$$

If we substitute $\nabla F(x)$, from equation (2.24), we find

$$x_{k+1} = x_k + 2\alpha e(k)Z(k). \quad (2.26)$$

Then the LMS algorithm can be written conveniently in matrix notation:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\alpha e(k)\mathbf{p}^T(k), \quad (2.27)$$

and

$$\mathbf{b}(k+1) = \mathbf{b}(k) + 2\alpha e(k). \quad (2.28)$$

Note that the error e and the bias \mathbf{b} are now vectors. This is also referred to as the delta rule or the Widrow-Hoff learning algorithm.

2.3 Xilinx System Generator

Xilinx System Generator is a tool for model-based design. Mathematic algorithms can be directly mapped to the arithmetic operations of Xilinx blocksets. For the more complicated arithmetic functions, such as sine and cosine, the CORDIC SINCOS in Figure 2-5 is provided. In addition, the MATLAB function can be written in the Xilinx MCode block shown in Figure 2-6 to provide the control flow condition.

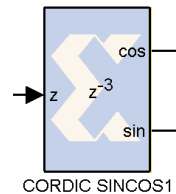


Figure 2- 5 CORDIC SINCOS Xilinx Block.

The Xilinx CORDIC SINCOS reference block implements Sine and Cosine generator circuit using a fully parallel CORDIC algorithm. Given input angle z , it computes the output $\cos(z)$ and $\sin(z)$. The CORDIC processor is implemented using building blocks from the Xilinx blockset.

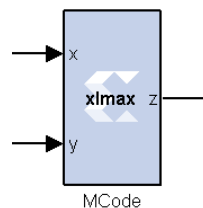


Figure 2- 6 MCode Xilinx Block.

The Xilinx MCode block is a container for executing a user supplied MATLAB function within Simulink. The block executes the M-code to calculate block outputs during a Simulink simulation. The same code is translated in a straightforward way into equivalent behavioral VHDL/ Verilog when the hardware is generated.

The simulation can be done within Simulink environments. The test vectors are generated by signal blocks. Finally, Xilinx System Generator generates the Hardware Description Language (HDL) codes in both Verilog HDL and VHDL for the FPGA implementation which be done by the Xilinx ISE tool.

2.4 Lyapunov Function

2.4.1 Theory of Lyapunov

Various types of asymptotic stability have been discussed for the solutions of differential equations describing dynamical systems. The most important type is that concerning the stability of solutions near to a point of equilibrium. This has been discussed by the theory of Lyapunov. The Russian mathematician Aleksandr Mikhailovich Lyapunov proposed two methods for demonstrating stability. The first method developed the solution in a series which was then proved convergent within limits. The second method, which is widely used nowadays, makes use of a Lyapunov function $V(x)$ which has an analogy to the potential function of classical dynamics. It is introduced as follows for a system having a point of equilibrium at $x=0$. Consider a function $V(x): \mathbb{R}^n \rightarrow \mathbb{R}$ such that

$V(x) \geq 0$ with equality if and only if $x = 0$ (positive definite).

$V(x) = \frac{d}{dt}V(x) \leq 0$ with equality is not constrained to only $x = 0$ (negative semidefinite).

Then $V(x)$ is called a Lyapunov function candidate and the system is stable in the sense of Lyapunov, whereas there is no general technique for constructing Lyapunov functions for ordinary differential equations (ODEs).

2.4.2 Lyapunov Advantages

In this research, the error in the ADALINE is the difference between the estimate output and reference signal. The error in ADALINE will reduce corresponding to iterations until the error is almost equal to zero. So the system should be global

asymptotic stability. As we know, the Lyapunov function is widely used in control system for providing stability and fast convergence of the system oscillations to their steady state values. So if we define a suitable Lyapunov function depends on the error, we can make the error reduce to zero faster and more accurate.

Chapter 3. Research Methodology

In the present AC power system, the amplitudes and phase of the currents and voltages are dynamic. On the other hand, we also should take the possible deviation of the fundamental frequency of the power system into consideration. To obtain the solution for on-line estimation of the harmonics, we proposed the ADALINE approach to the problem. As shown in Figure 3-1. The method provided the block for tracking the amplitudes and the block for tracking the fundamental frequency. We also proposed adaptive learning parameters block for bring a better and more accurate convergence in tracking the harmonic components. Due to we simulated in different cases with different reference fundamental frequency. We can get the convergence in the certain time, so we should reset the amplitude registers and the fundamental register every certain time.

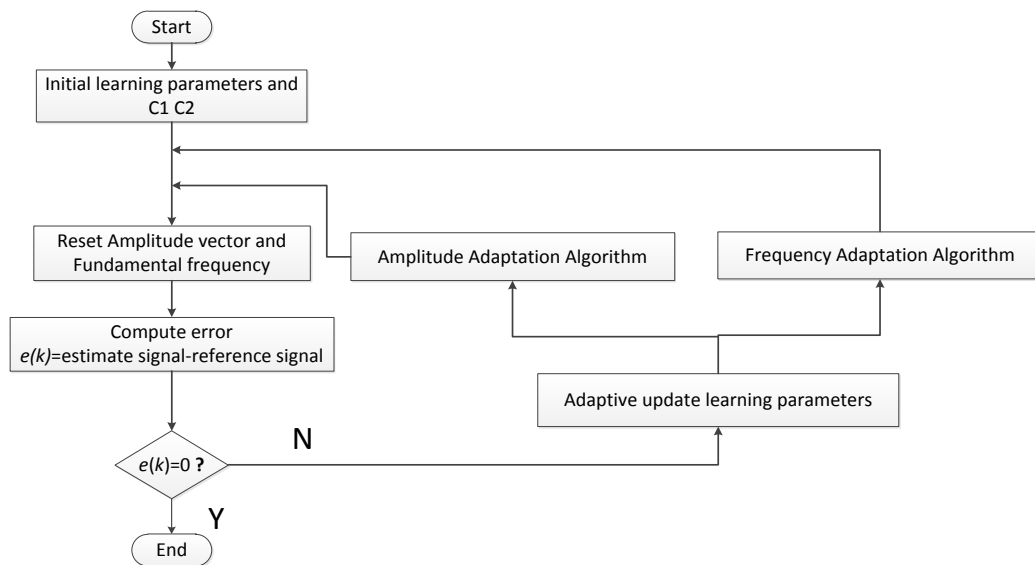


Figure 3- 1 Flowchart of research methodology.

We verified the learning parameters by simulated the proposed algorithm on MATLAB, then the design has been implemented and simulated in Xilinx System Generator running in Simulink environments. Computer simulation experiments showed that the proposed algorithm can estimate the signal faster and more accurately. Then, we implemented the method on a Xilinx Spartan3E-1200 FPGA (Field Programmable Gate Array). The hardware-in-the loop verification results show that the proposed ADALINE adaptive filter can accurately estimate the dynamic harmonic

amplitudes and phases, and adaptively track the dynamic fundamental frequency in the present AC power system. Moreover, the adaptive learning parameters bring the faster and more stable convergence.

3.1 ADALINE

In recent years, a modified ADALINE structure has been proposed in which the fundamental frequency has been treated as unknown parameter and simultaneously estimates it with the tracking of the harmonic components.

ADALINE is a two-layer neural network with n inputs and a single output, which is a linear combination of the inputs. The most common power system is 3-phase at 50 Hz. Then the algorithm has been modified to estimate the 3-phase voltages or currents simultaneously using ADALINE which consists of three neurons (one neuron per phase) [6] as shown in Figure 3-2. The output from the neural estimator for phase-a is:

$$\hat{V}_a(t) = W_a^T X, \quad (3.1)$$

where W_a denotes the weight vector for the phase-a voltage or current and X is the input vector given by :

$$X = [\sin\omega_k t_k \quad \cos\omega_k t_k \quad \sin N\omega_k t_k \quad \cos N\omega_k t_k, \quad \dots, \sin N\omega_k t_k \quad \cos N\omega_k t_k \quad 1 - kT_s]^T. \quad (3.2)$$

After final convergence is reached, the three phase Fourier coefficients for the estimated signals are computed as:

$$A_{l,j} = \sqrt{w_{o,j}^2(2l-1) + w_{o,j}^2(2l)}, \quad (3.3)$$

$$\varphi_{l,j} = \arctan \left[\frac{w_{o,j}(2l)}{w_{o,j}(2l-1)} \right] \quad \text{for } j = a, b, c. \quad (3.4)$$

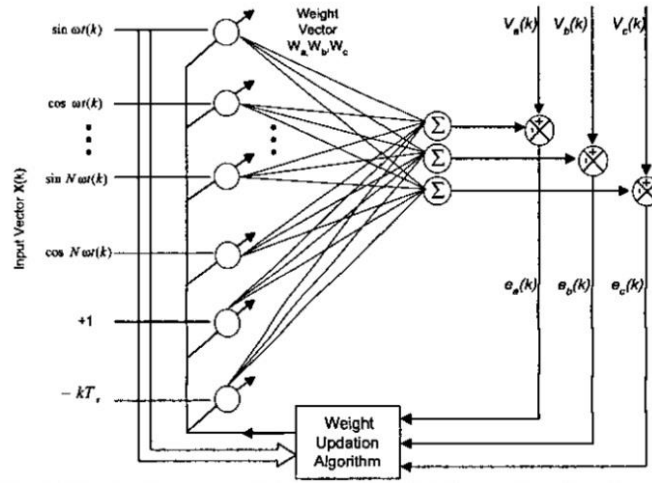


Figure 3- 2 Block diagram of the ADALINE for estimating 3-phase voltages or currents (3-Phase ADALINE) [6].

Let us assume that the power system is distorted by the presence of higher harmonics of unknown magnitudes and phases. The general form of the line voltage is predicated as

$$x(t) = \sum_{l=1}^N X_l \sin(l\omega_k t_k + \varphi_l) + \epsilon(k), \quad (3.5)$$

where X_l and φ_l are the magnitude and the phase of l th harmonic component ($l = 1, 2, 3, \dots, N$). Equation (3.5) may be substituted by the following description which is more suitable for our purposes:

$$x(t) = \sum_{l=1}^N (A_l \sin l\omega_k t_k + B_l \cos l\omega_k t_k) + \epsilon(k), \quad (3.6)$$

where $A_l = X_l \cos \varphi_l$ and $B_l = X_l \sin \varphi_l$. Knowledge of A_l and B_l determine both the magnitude X_l and phase φ_l of the l^{th} harmonic, according to

$$X_l = \sqrt{A_l^2 + B_l^2}, \quad (3.7)$$

$$\varphi_l = \arctan \frac{B_l}{A_l} \quad (3.8)$$

3.2 Tracking of the Amplitudes of the Fundamental Signal and Harmonics

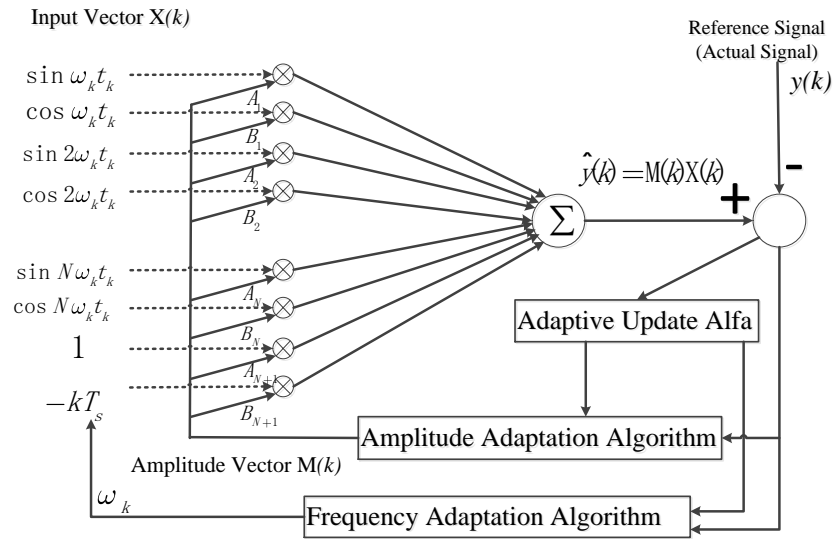


Figure 3- 3 Block diagram of the ADALINE for estimating harmonic components and fundamental frequency.

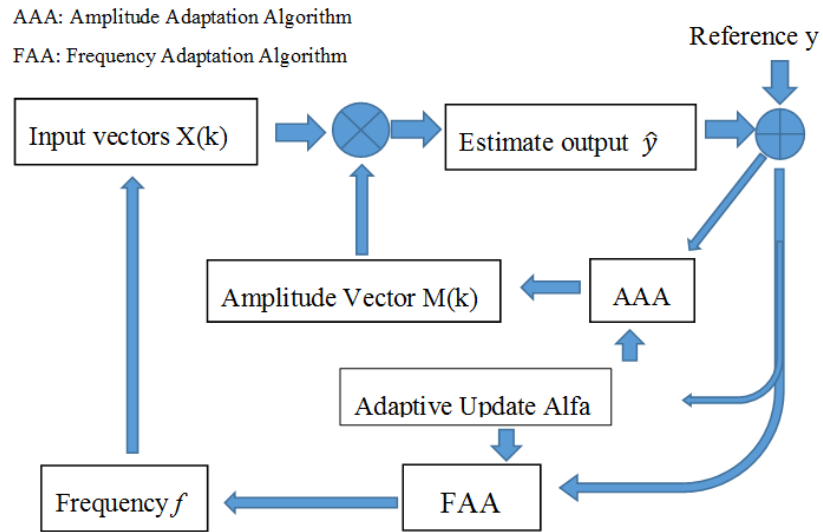


Figure 3- 4 Data flow of the proposed ADALINE.

To obtain the solution for on-line estimation of the harmonics, we proposed the neural approach to the problem. The block diagram are shown in Figure 3-3. Flowchart in Figure 3-4 is drawn according to Figure 3-3. Let us assume that the measurements are made a time window and that the sample $y(k)$ are taken at time instance $t_k(k=1, 2, \dots, n)$ where n is the number of samples in the window. The samples may be equally distributed within the window, and the width of the window is

dependent on the designer. According to the neural concept, we define the performance index function as

$$\xi = e^2 = \left[\sum_{l=1}^N (A_l \sin l \omega_k t_k + B_l \cos l \omega_k t_k) + A_{N+1} - kT_s B_{N+1} - y_k \right]^2. \quad (3.9)$$

If we assume that ω is known and the time instants t_k are fixed, the terms $\sin l \omega_k t_k$ and $\cos l \omega_k t_k$ form the coefficients of the unknown optimized variables A_l, B_l ($l = 1, 2, 3, \dots, N$) and the terms 1 and $-kT_s$ form the coefficients of DC components and DC decaying components are denoted here in vector notation as

$$X(k) = [\sin \omega_k t_k \quad \cos \omega_k t_k \quad \sin 2 \omega_k t_k \quad \cos 2 \omega_k t_k \quad \dots \quad \sin N \omega_k t_k \quad \cos N \omega_k t_k \quad 1 \quad -kT_s]^T. \quad (3.10)$$

We also define the variables A_l, B_l ($l = 1, 2, 3, \dots, N$) in vector notation as

$$M(k) = [A_1(k) \quad B_1(k) \quad A_2(k) \quad B_2(k) \quad \dots \quad A_N(k) \quad B_N(k) \quad A_{N+1}(k) \quad B_{N+1}(k)]^T. \quad (3.11)$$

The optimization problem is quadratic and its minimum can be easily found. This provides the Hessian corresponding to the objective function equation (3.9) is at least positive semidefinite. The approximation of $\nabla \xi(k)$ can be used in the steepest descent approach [8] as

$$M(k+1) = M(k) - \alpha \nabla \xi(k), \quad (3.12)$$

where α denotes the constant of integration and $\nabla \xi$ is the gradient of the objective function ξ with respect to amplitude as

$$\nabla \xi(k)_M = \frac{\partial [e^2(k)]}{\partial M} = 2e(k) \frac{\partial e(k)}{\partial M} = 2e(k)X(k), \quad (3.13)$$

where the error between the estimate signal and the actual signal is:

$$e(k) = \hat{y}(k) - y(k) = M^T(k)X(k) - y(k). \quad (3.14)$$

Combining equation (3.12) with equation (3.13), we can get the equation for update the weight vector

$$M(k+1) = M(k) - 2\alpha e(k)X(k). \quad (3.15)$$

3.3 Tracking of the Fundamental Frequency

We also can use the method of steepest descent method to iterative calculate the frequency as:

$$\omega(k+1) = \omega(k) - \alpha_\omega \nabla \xi(k)_\omega. \quad (3.16)$$

The gradient of the mean square error with respect to phase frequency is:

$$\begin{aligned} \nabla \xi(k)_\omega &= 2e(k) \frac{\partial e(k)}{\partial \omega} \\ &= 2e(k) \sum_{l=1}^N [lt_k A_l \cos(l\omega_k t_k) - lt_k B_l \sin(l\omega_k t_k)]. \end{aligned} \quad (3.17)$$

If we also want to take into account the deviation of frequency from the nominal value, we can apply a similar steepest descent towards to ω . Then, we have the equation for update the fundamental frequency as follow:

$$\begin{aligned} &\omega(k+1) - \omega(k) \\ &= -2\alpha_\omega e(k) \sum_{l=1}^N [lt_k A_l \cos(l\omega_k t_k) - lt_k B_l \sin(l\omega_k t_k)]. \end{aligned} \quad (3.18)$$

The initial value of the angular frequency should be set to the nominal one. We know that the actual ω value is a little departure from the nominal value. In the further stage of adaptation, when both A_l and B_l , as well as ω , are close to the optimum values. There is a fine tuning process of these parameters leading to the global minimum of the objective function.

3.4 Adaptive Variation of the Learning Parameters α

As we mentioned in the previous part, it should determine the learning rate α and α_ω to find the balance between the convergence speed and the state-steady-misadjustment error. An alternative method for tuning the learning parameter α is derived here by using a Lyapunov function. The Lyapunov energy function is widely used in control system for providing stability and fast convergence of the system oscillations to their steady state values. A Lyapunov function $V(k)$ is thus defined as

$$V(k) = e^2(k). \quad (3.19)$$

Here at each instant, the Lyapunov function depends on error and its derivative, which are dependent on the distorted signals superimposed over the actual signal to be estimated. The change in the Lyapunov function $V(k)$ is calculated as

$$\Delta V(k) = V(k) - V(k - 1) . \quad (3.20)$$

The value of the learning parameter $\alpha(k)$ is chosen as

$$\alpha(k) = \alpha_0 + c_1 * \text{SGN}(\Delta V(k)) * V(k), \quad (c_1 > 0). \quad (3.21)$$

Then we used it again for update learning parameter $\alpha_\omega(k)$:

$$\alpha_\omega(k) = \alpha_1 + c_2 * \text{SGN}(\Delta V(k)) * V(k), \quad (c_2 > 0). \quad (3.22)$$

Also constant values of c_1, c_2 are chosen according to amplitude of current or voltage in the actual situation. To find the change in the fundamental frequency, ω is initially set to the nominal value [7].

3.5 MATLAB Code

List 3- 1 Using adaptive Learning parameters in MATLAB.

Using Adaptive Learning Parameters	
1	t=0.00001:0.0002:0.3;
2	for i=1:length(t)
3	y(i)=1.5*sin(100*pi.*t(i)+pi*29.3/180)+0.068*sin(5*100*pi.*t(i)
4	+pi*141.6/180) +0.024*sin(7*100*pi.*t(i)+pi*66.2/180)
5	+0.02*sin(11*100*pi.*t(i)+pi*(-99.4)/180);
6	end
7	c=[1 -1 5 -5 7 -7 11 -11];
8	a=0.0179; a0=a;
9	aw=12.5; a1=aw;
10	m=50.05*pi*2;
11	c1=20.33; c2=3.70;
12	w=[1.2 0.7 -0.045 0.04 0.006 0.02 -0.001 -0.010]';
13	s=size(t); s=s(2);
14	v=w; f=0;l=0;
15	for i=1:s
16	X=[sin(m*t(i)) cos(m*t(i)) sin(5*m*t(i)) cos(5*m*t(i)) sin(7*m*t(i))
17	cos(7*m*t(i)) sin(11*m*t(i)) cos(11*m*t(i))];
18	e=X*w-y(i);
19	w=w-a0*e*X';
20	x=[cos(m*t(i)) sin(m*t(i)) cos(5*m*t(i)) sin(5*m*t(i)) cos(7*m*t(i))
21	sin(7*m*t(i)) cos(11*m*t(i)) sin(11*m*t(i))];
22	q=t(i)*((x.*(c))*v);
23	m=m-a1*e*q;
24	L=e^2;
25	T=L-f^2;
26	a0=a+sign(T)*c1* L;
27	a1=aw+sign(T)*c2* L;
28	g(:,i)=w; F(i)=m; E(i)=e; v=w; f=e; l=L;
29	end

In the List 3-1, line 1~6 is for generating the reference signal. The sample time of the signal is 0.0002s. Line 7 is the order numbers of the harmonics. Then, we initialized the learning parameters for amplitude tracking and fundamental frequency

tracking in line 8~9. Line 10 is for initializing the fundamental frequency. Line 11 is for setting the constant parameters for adaptive update the learning parameters in equation (3.21) and equation (3.22). Line 12 is for initializing the amplitude vectors. Line 18 is for computing the error between the estimate signal and the reference signal. Line 19 is according to equation (3.15) for tracking the amplitudes. Line 23 is according to equation (3.18) for tracking the fundamental frequency. Line 24~27 is for adaptive updating the learning parameters. Line 26 is for getting the adaptive learning parameter for amplitude tracking, it is according to the equation (3.21). Line 27 is for getting the adaptive learning parameter for fundamental frequency tracking, it is according to the equation (3.22). Vectors in line 28 is used for store the data. 'g' is for storing the amplitudes. 'F' is for storing the fundamental frequency. 'E' is for storing the error.

List 3- 2 Using constant Learning parameters in MATLAB.

Using Constant Learning Parameters	
1	t=0.00001:0.0002:0.3;
2	for i=1:length(t)
3	y(i)=1.5*sin(100*pi.*t(i)+pi*29.3/180)+0.068*sin(5*100*pi.*t(i)
4	+pi*141.6/180) +0.024*sin(7*100*pi.*t(i)+pi*66.2/180)
5	+0.02*sin(11*100*pi.*t(i)+pi*(-99.4)/180);
6	end
7	c=[1 -1 5 -5 7 -7 11 -11];
8	a=0.0179; aw=12.5; m=50.05*pi*2;
9	w=[1.2 0.7 -0.045 0.04 0.006 0.02 -0.001 -0.010]';
10	s=size(t); s=s(2);
11	v=w; f=0;l=0;
12	for i=1:s
13	X=[sin(m*t(i)) cos(m*t(i)) sin(5*m*t(i)) cos(5*m*t(i)) sin(7*m*t(i))
14	cos(7*m*t(i)) sin(11*m*t(i)) cos(11*m*t(i))];
15	e=X*w-y(i);
16	w=w-a0*e*X';
17	x=[cos(m*t(i)) sin(m*t(i)) cos(5*m*t(i)) sin(5*m*t(i)) cos(7*m*t(i))
18	sin(7*m*t(i)) cos(11*m*t(i)) sin(11*m*t(i))];
19	q=t(i)*((x.*(c))*v);
20	m=m-a1*e*q;
21	g(:,i)=w; F(i)=m; E(i)=e; v=w; f=e;
22	end

In the List 3-2, line 1~6 is also for generating the reference signal. Line 7 is the order numbers of the harmonics. Then, we initialized the learning parameters for amplitude tracking and fundamental frequency tracking and the fundamental frequency in line 8. Line 9 is for initializing the amplitude vectors. Line 17 is for computing the error. Line 18 is according to equation (3.15) for tracking the amplitudes. Line 22 is according to equation (3.18) for tracking the fundamental frequency. Vectors in line 23 is used for store the data. 'g' is for storing the amplitudes. 'F' is for storing the fundamental frequency. 'E' is for storing the error.

3.6 FPGA Implementation

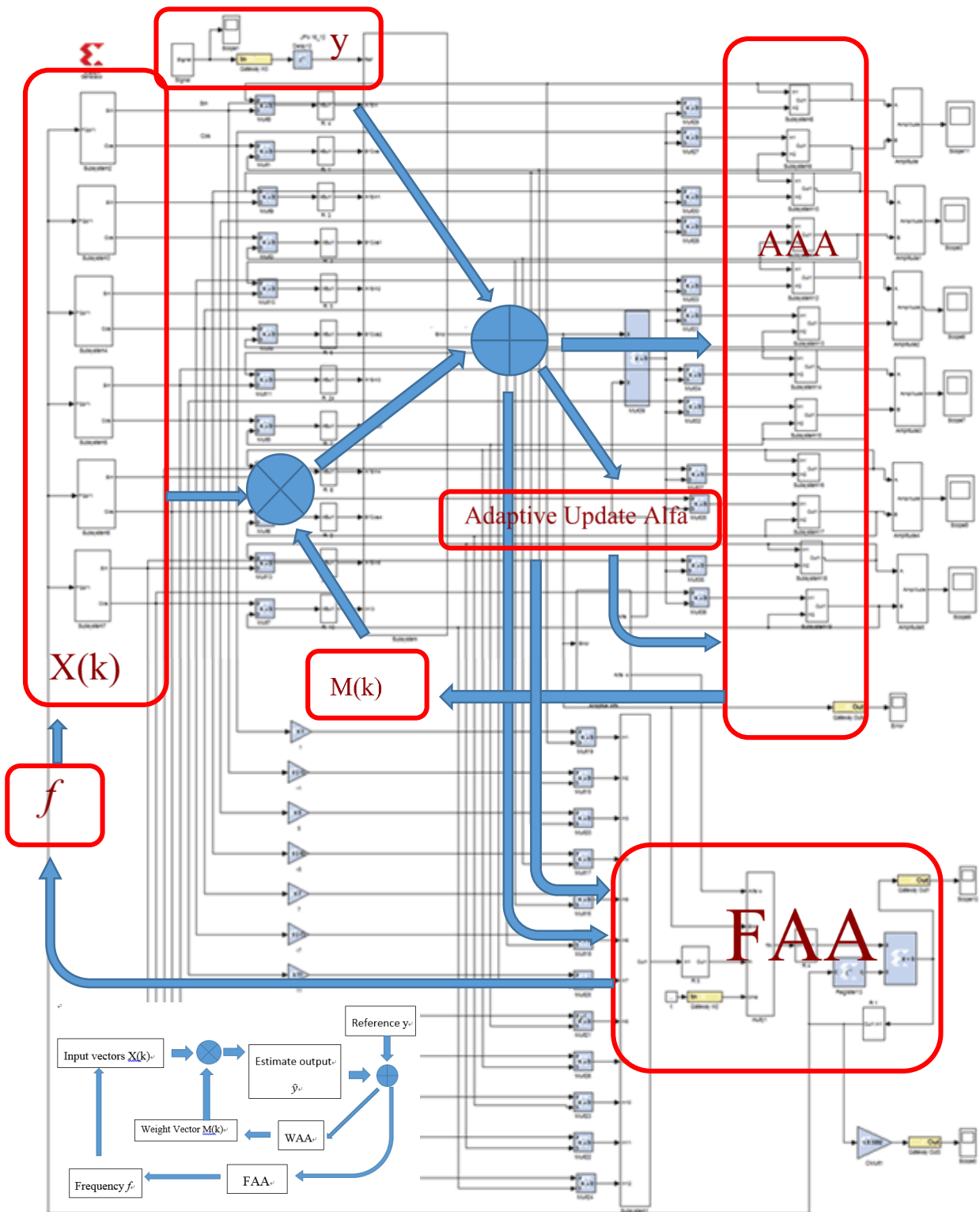


Figure 3- 5 Total Xilinx design for tracking fundamental and harmonics.

According to the Figure 3-3, we can apply every part into FPGA implementation as shown in Figure 3-5. The details of Xilinx model design of each part are as follow.

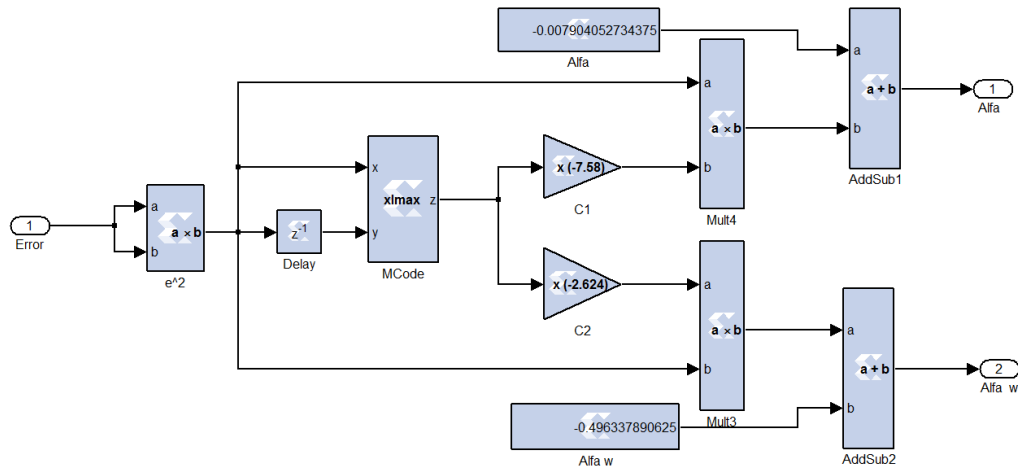


Figure 3- 6 Adaptive model for update learning parameters.

As shown in Figure 3-6, the Xilinx model is for adaptive update the learning parameters. We use the error between the estimate output from neural network and the actual signal as the input. Then we compare the error with the error in the last one instant by using the MCode block. According to equation (3.21) and equation (3.22), the function in the Mblock is below

```

function z = xlmaz(x, y)
if x > y
z = 1;
elseif x < y
z = -1;
else
z = 0;
end

```

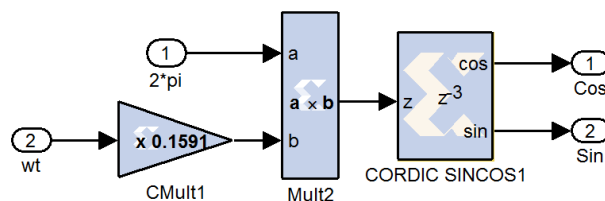


Figure 3- 7 Model for generating dynamic input vectors (Sin, Cos).

Figure 3-7 is for generate dynamic input vectors (Sin, Cos). At the first multiply, one input is time. So the results from first multiply will be very big number

finally. And as we know, $\sin\varphi = \sin(\varphi + 2k\pi)$, $\cos(\varphi) = \cos(\varphi + 2k\pi)$. So we use the result from the first multiply to divide 2π , then we keep the fractional part left. Use the fractional part times 2π , then we will get the number smaller than 2π . Then we can get the target keep the number in a certain bits.

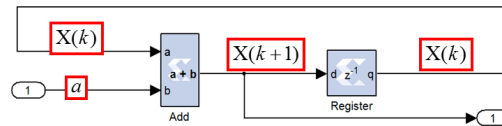


Figure 3- 8 Model for accumulating weight vectors.

$$x(k+1) = x(k) + a. \quad (3.23)$$

In the design, we will use equation (3.15) and equation (3.18) to adaptive tracking the amplitude and fundamental frequency. We have equation (3.23) similar as equation (3.15) and equation (3.18). Using a as the input b and using $x(k)$ as the input a , then we will get the output $a + b$ i.e. $x(k+1)$. Due to we will set the initial value $x(0)$, so we use the register with latency is 1. So the output from register is $x(k)$ i.e. a . Then we can make the accumulate compute come true.

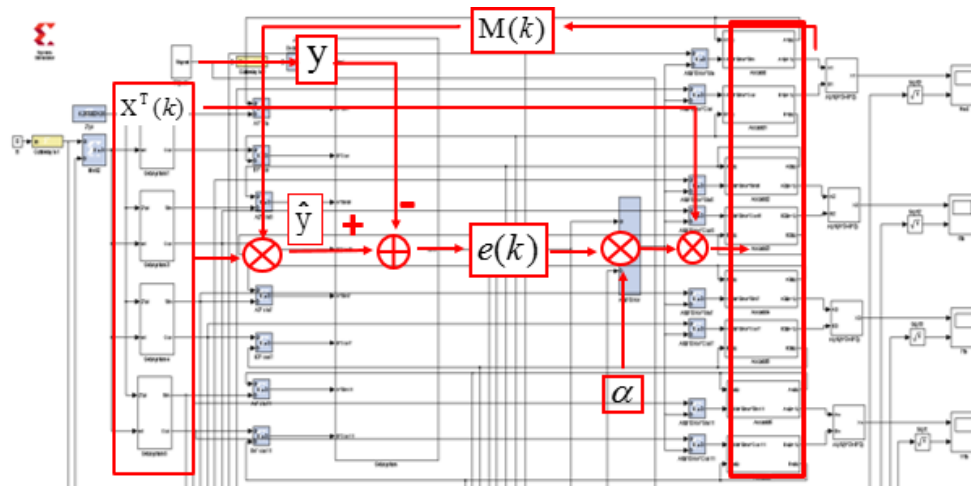


Figure 3- 9 Xilinx design for tracking the amplitude.

The design in Figure 3-9 is according to equation (3.15). This part is for tracking the amplitudes of the fundamental signal and harmonics.

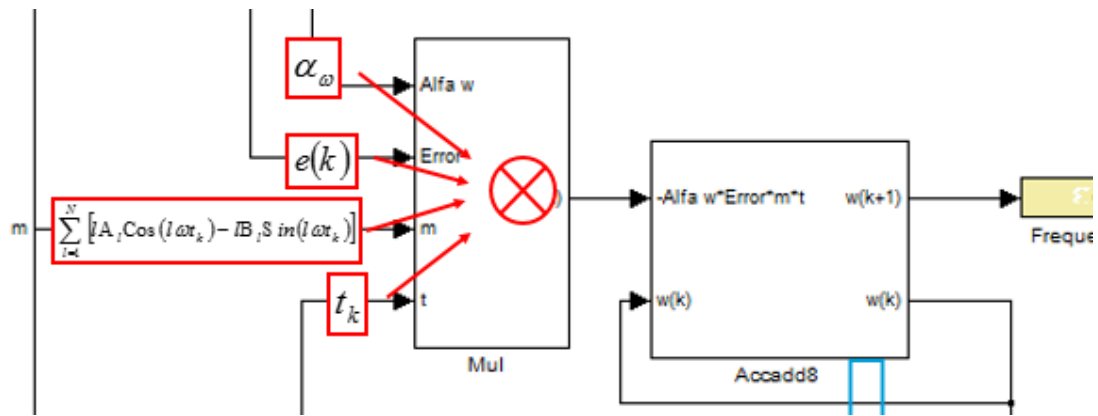


Figure 3-10 Xilinx design for tracking the fundamental frequency.

The design in Figure 3-10 is according to equation (3.18). This part is for tracking the fundamental frequency.

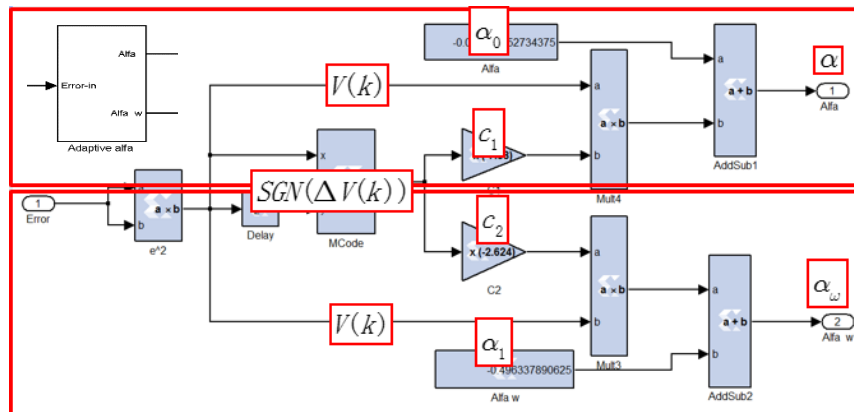


Figure 3-11 Xilinx design for tracking the fundamental frequency.

The design in Figure 3-11 is according to equation (3.21) and equation (3.22). This part is for adaptively get the learning parameters for the amplitude tracking and fundamental frequency tracking.

Chapter 4. Results and Discussions

Figure 4-1 shows the flowchart of experiments to validate the proposed method. We used constant learning parameters and adaptive learning parameters for the comparison objects. In order to evaluate the performance of the ADALINE in estimating amplitudes, the phase of the harmonics and the fundamental frequency, numerical experiments using MATLAB software have been performed. At the same time, the design has been implemented on Xilinx Spartan3E-1200 FPGA (Field Programmable Gate Array). The hardware-in-the-loop verification results show that the proposed ADALINE adaptive filter can not only accurately estimate the harmonic amplitudes and phases but also adaptively track the dynamic fundamental frequency in the present AC power system.

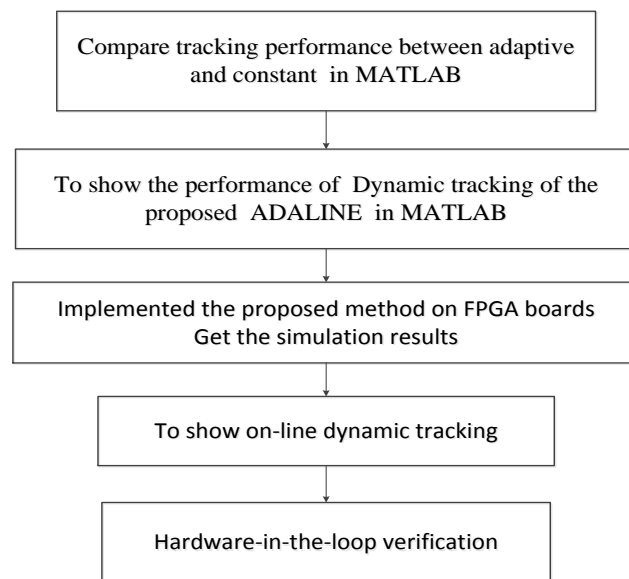


Figure 4- 1 Flowchart of experiments.

Here, we are presenting some illustrative results to show the accurate tracking capability of the neural estimator in MATLAB simulation and on FPGA fashion.

We assume that there is a nominal distorted signal as

$$y(t) = 1.5 \sin(100\pi t + 29.3^\circ) + 0.068 \sin(5 \times 100\pi t + 141.6^\circ) + 0.024 \sin(7 \times 100\pi t + 66.2^\circ) + 0.02 \sin(11 \times 100\pi t - 99.4^\circ), \quad (4.1)$$

and the signal waveforms are shown in Figure 4-2.

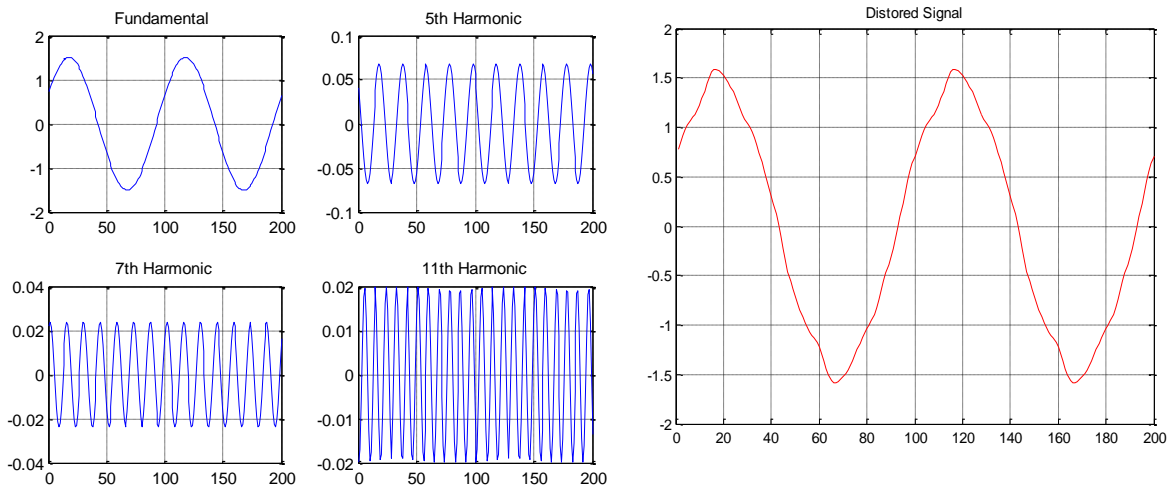
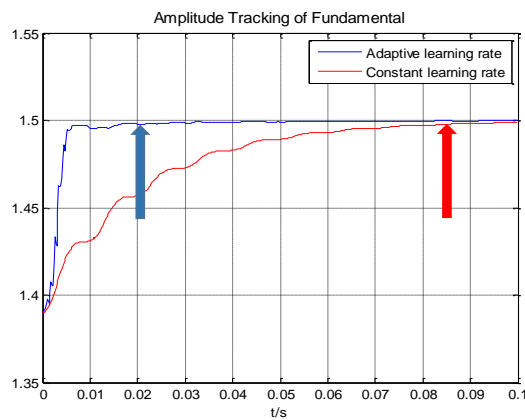
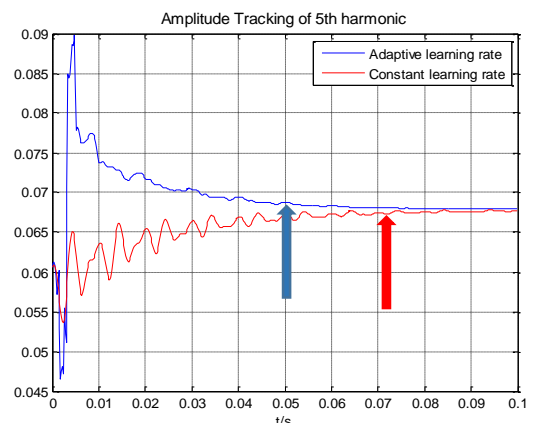


Figure 4- 2 Distorted Signal (Fundamental,5th, 7th, 11th)(time=2T).

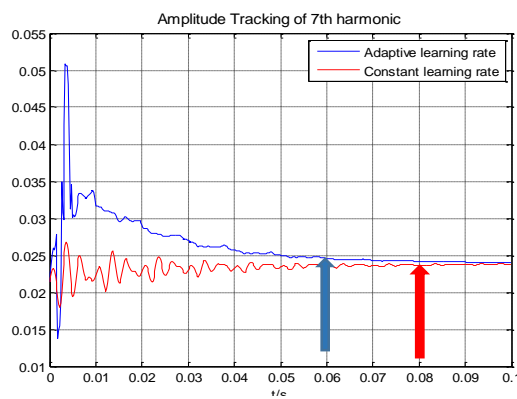
4.1 MATLAB Simulation Results



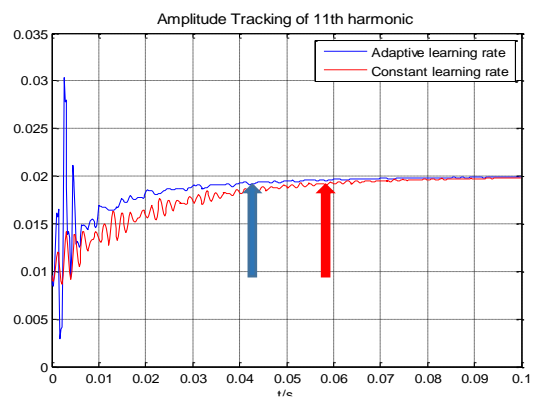
(a) Amplitude Tracking of Fundamental signal.



(b) Amplitude Tracking of 5th harmonic.



(c) Amplitude Tracking of 7th harmonic.



(d) Amplitude Tracking of 11th harmonic.

Figure 4- 3 Amplitude Tracking of Fundamental signal and harmonics (5th, 7th, 11th) in MATLAB simulation (comparison case).

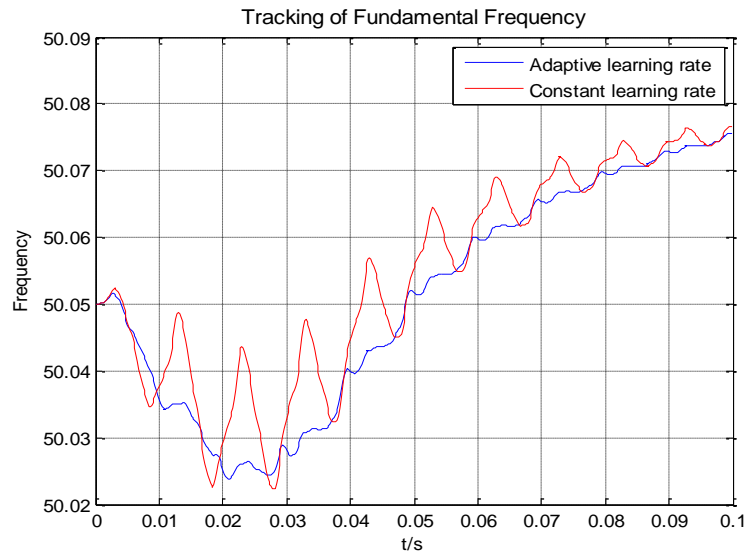


Figure 4- 4 Tracking of Fundamental Frequency in MATLAB simulation (comparison case).

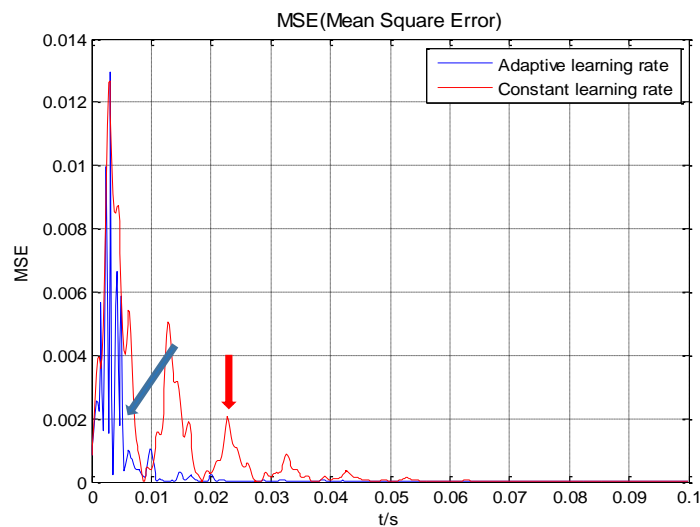


Figure 4- 5 Tracking of Mean Square Error in MATLAB simulation (comparison case).

Figure 4-3, Figure 4-4 and Figure 4-5 show the performance using ADALINE with adaptive step parameters (blue line), whereas and the performance using ADALINE with constant step parameters (red line). They show the tracking results when the fundamental frequency and amplitude change. We initialized the amplitude vectors to some values, set the initial fundamental angle frequency as 50.05Hz. When the fundamental frequency increased, the proposed ADALINE can track the change of

fundamental frequency. Even though, both of the two situations can track the fundamental frequency and amplitude of the harmonics. But if we compare the results, the ADALINE with adaptive learning parameters brings more stable and faster convergence, especially in the situation of tracking the amplitudes. We summary the convergence time of amplitudes and mean square error in details in the Table 4-1.

Table 4-1 Convergence Time.

Amplitude of Signal and Harmonics	Convergence Time	
	Adaptive (Blue Line)	Constant (Red Line)
Fundamental	0.02 s	0.085 s
5 th	0.05 s	0.072 s
7 th	0.06 s	0.08 s
11 th	0.043 s	0.058 s
Mean Square Error(0-0.002)	0.005 s	0.023 s

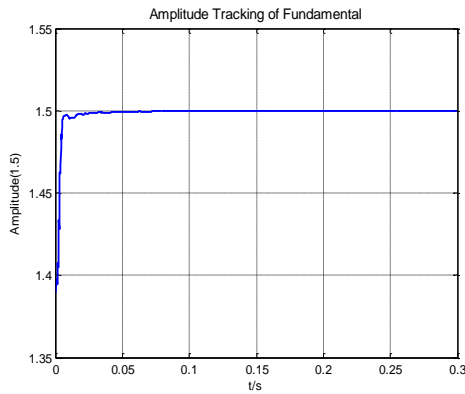
From the Table 4-1, we macroscopically know that the convergence speed of the amplitude when using adaptive learning parameters is faster. On the other hand, the mean square error is the real performance index in the application of ADALINE. If we assume that we get convergence when the mean square error within 0.0002. Then from the convergence speed of mean square error, we can make sure the condition using adaptive learning parameters is much better than the condition using constant learning parameters. From Figure 4-4, we can see obviously that the convergence change of the frequency tracking with using adaptive learning parameters (blue) is more stable than the condition using constant learning parameters.

We did the MATLAB simulation in two cases. Both of the two cases used the same initial fundamental frequency as 50.05 Hz. In case 1, the fundamental frequency suddenly changed to 50.1 Hz. Set the initial amplitude vector as $w=[1.2 \quad 0.7 \quad -0.045 \quad 0.04 \quad 0.006 \quad 0.02 \quad -0.001 \quad -0.010]'$ which different from the reference signal.

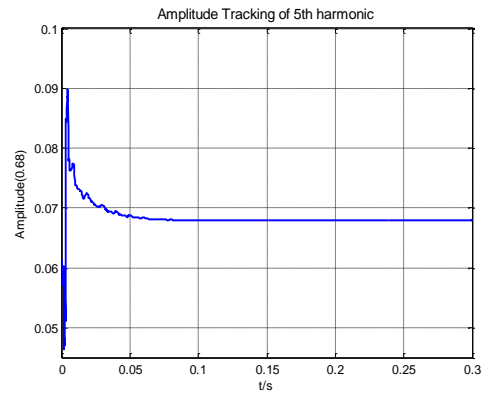
Case 1 in MATLAB:

Set the initial fundamental frequency ($f=50.05$ Hz) , the reference fundamental frequency ($f=50.1$ Hz),

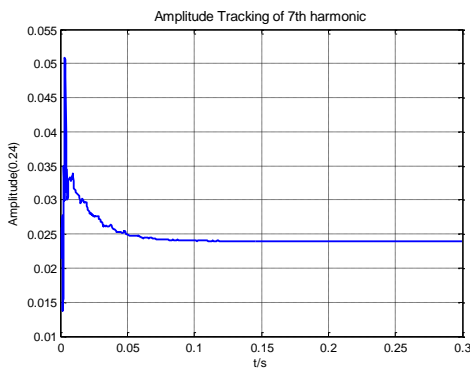
$$w=[1.2 \ 0.7 \ -0.045 \ 0.04 \ 0.006 \ 0.02 \ -0.001 \ -0.010]'$$



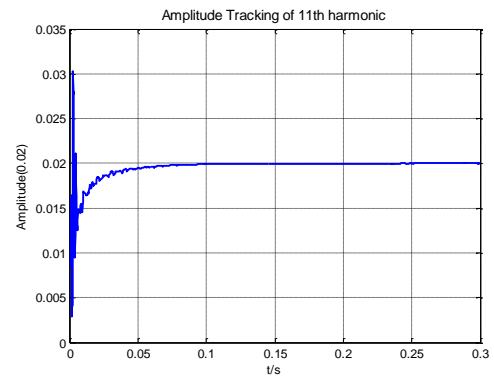
(a) Amplitude Tracking of Fundamental signal.



(b) Amplitude Tracking of 5th harmonic.



(c) Amplitude Tracking of 7th harmonic.



(d) Amplitude Tracking of 11th harmonic.

Figure 4- 6 Amplitude Tracking of Fundamental signal and harmonics (5th, 7th, 11th).

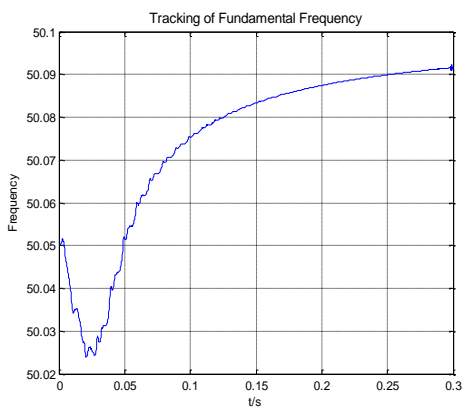


Figure 4-7 Tracking of Fundamental Frequency.

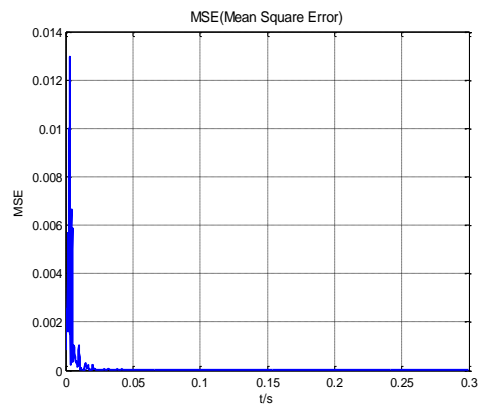


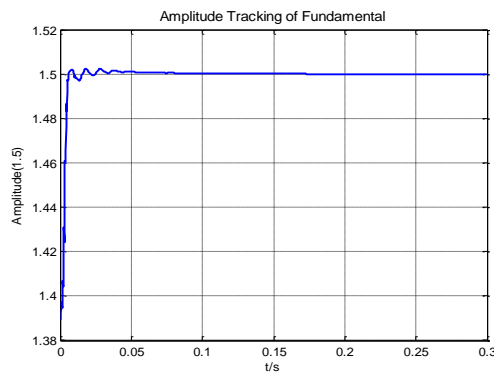
Figure 4-8 Tracking of Mean Square Error.

In case 2 we also set the initial fundamental frequency as 50.05 Hz, and the fundamental frequency suddenly change to 49.9 Hz. We also set the initial amplitude vector as $w = [1.2 \quad 0.7 \quad -0.045 \quad 0.04 \quad 0.006 \quad 0.02 \quad -0.001 \quad -0.010]'$ which is same the situation in case 1.

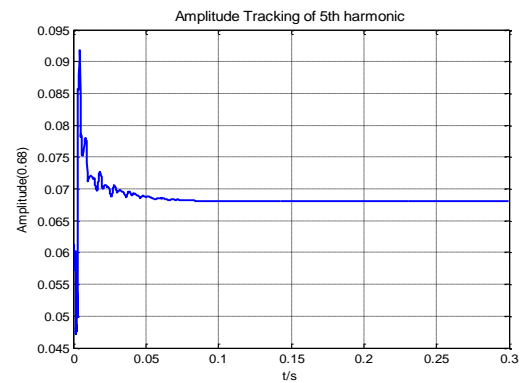
Case 2 in MATLAB:

Set the initial fundamental frequency ($f=50.05$ Hz) , the reference fundamental frequency ($f=49.9$ Hz),

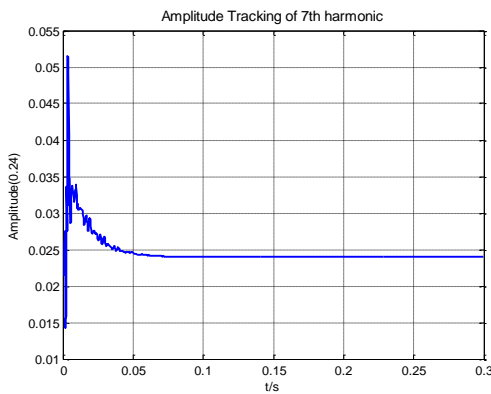
$$w=[1.2 \ 0.7 \ -0.045 \ 0.04 \ 0.006 \ 0.02 \ -0.001 \ -0.010]'$$



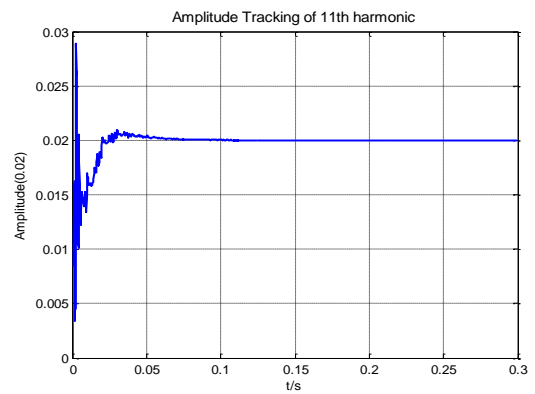
(a) Amplitude Tracking of Fundamental signal.



(b) Amplitude Tracking of 5th harmonic.



(c) Amplitude Tracking of 7th harmonic.



(d) Amplitude Tracking of 11th harmonic.

Figure 4- 9 Amplitude Tracking of Fundamental signal and harmonics (5th, 7th, 11th).

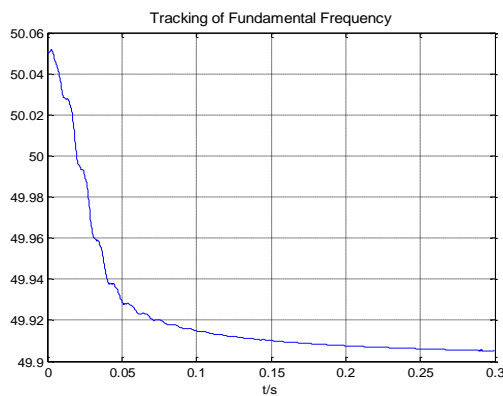


Figure 4-10 Tracking of Fundamental Frequency.

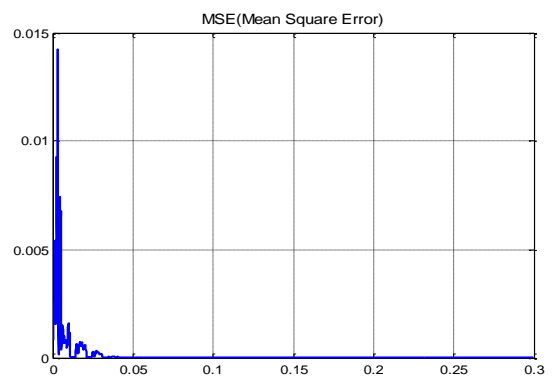


Figure 4-11 Tracking of Mean Square Error.

Table 4-2 Convergence Time in case 1 and case 2.

Amplitude of Signal and Harmonics	Convergence Time	
	Case1	Case2
Fundamental	0.02 s	0.02s
5 th	0.05 s	0.048s
7 th	0.06 s	0.05s
11 th	0.043 s	0.055 s
Mean Square Error(0-0.002)	0.005s	0.007s
Fundamental Frequency (0-0.01 Hz)	0.25s	0.15s

From the results in the Table 4-2, we can know that the proposed method can track not only the amplitude of signal and harmonics but also the fundamental frequency of the distorted signal.

4.2 FPGA Implementation results

After we verified the method in the MATLAB. We implemented it on the FPGA fashion. Figure 4-12 shows the total Xilinx design for tracking fundamental and harmonics components.

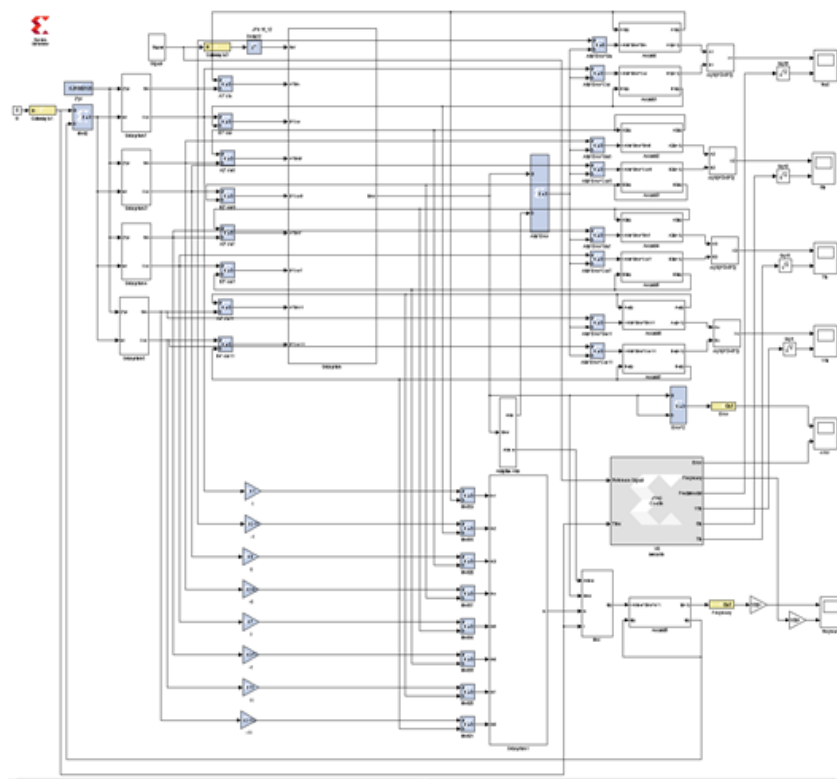


Figure 4- 12 Total Xilinx design for tracking fundamental and harmonics.

Table 4-3 Device Utilization Summary (Xilinx Spartan3E-1200 FPGA).

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	588	17,344	3%
Number of 4 input LUTs	12,442	17,344	71%
Number of occupied	6,529	8,672	75%
Total Number of 4 input LUTs	12,831	17,344	73%
Number of bonded IOBs	166	250	66%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	24	28	85%
Average Fanout of Non-Clock Nets	2.23		

Table 4-4 Static Timing Analysis Report (Xilinx Spartan3E-1200 FPGA).

Source Clock	Src: Rise Dest: Rise	Src: Fall Dest: Rise	Src: Rise Dest: Fall	Src: Fall Dest: Fall
clk	95.609			
Design statistics: Minimum period:95.609 ns{1} (Maximum frequency:10.459 MHz)				

Table 4-5 Device Utilization Summary (xc6slx100-3fgg484 FPGA).

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	3016	126,576	2%
Number of Slice LUTs	12,342	63,288	19%
Number used as logic	12,278	63,288	19%
Number used as Memory	4	15,616	1%
Number used exclusively as route-thrus	60		
Number of occupied Slices	4,258	15,822	26%
Number of MUXCYs used	9,200	31,644	29%
Number of LUT Flip Flop pairs used	12,344		
Number of bonded IOBs	166	326	50%
Number of BUFG/BUFGMUXs	1	16	1%
Number of ILOGIC2/ISERDES2s	16	506	3%
Number of DSP48A1s	24	180	13%
Average Fanout of Non-Clock Nets	2.89		

Table 4-6 Static Timing Analysis Report (xc6slx100-3fgg484 FPGA).

Source Clock	Src: Rise Dest: Rise	Src: Fall Dest: Rise	Src: Rise Dest: Fall	Src: Fall Dest: Fall
clk_1	75.590			
Design statistics: Minimum period:95.609 ns{1} (Maximum frequency:13.229 MHz)				

From the resource Table 4-3 and Table 4-5 of the two FPGA boards, we can know that the Spartan6 can help us save the resources. Then we can track the signal with

more kinds of harmonics. As shown in Table 4-4 and Table 4-6, we know that the minimum period in Spartan6 is shorter, then we can get the convergence faster.

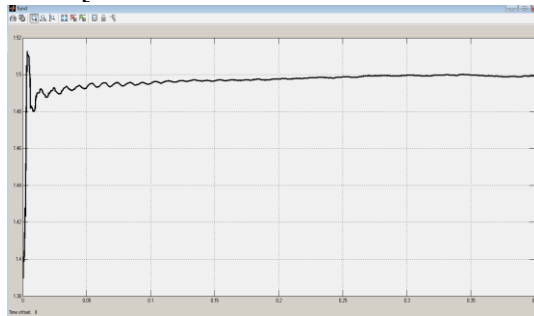
Case 1 in Xilinx System Simulation:

Set the initial fundamental frequency ($f=50$ Hz).

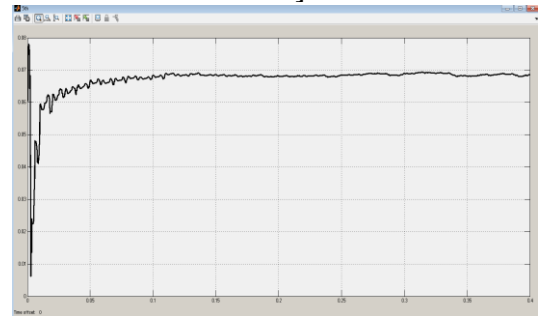
Set the reference fundamental frequency ($f=50.1$ Hz).

Initial amplitude vector:

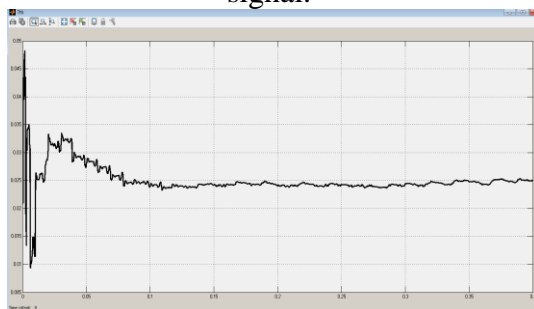
$w=[1.2 \quad 0.7 \quad -0.045 \quad 0.04 \quad 0.006 \quad 0.02 \quad -0.001 \quad -0.010]'$.



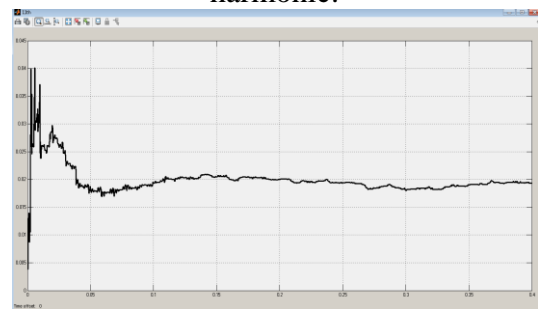
(a) Amplitude Tracking of Fundamental signal.



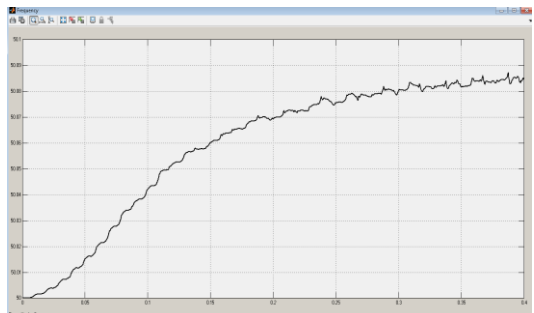
(b) Amplitude Tracking of 5th harmonic.



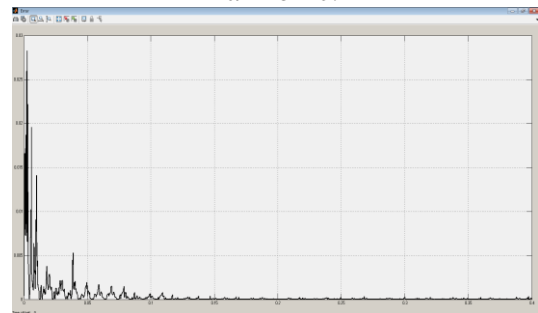
(c) Amplitude Tracking of 7th harmonic.



(d) Amplitude Tracking of 11th harmonic.



(e) Tracking of Fundamental Frequency.



(f) Tracking of Mean Square Error.

Figure 4- 13 Xilinx Simulation in Case 1.

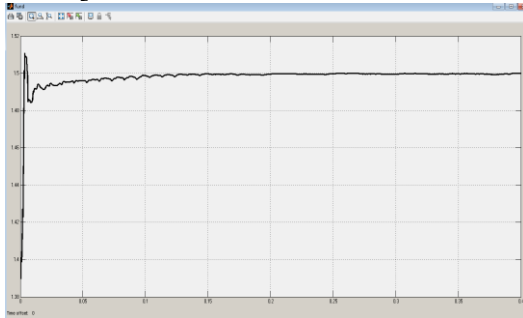
Case 2 in Xilinx System Simulation:

Set the initial fundamental frequency ($f=50$ Hz).

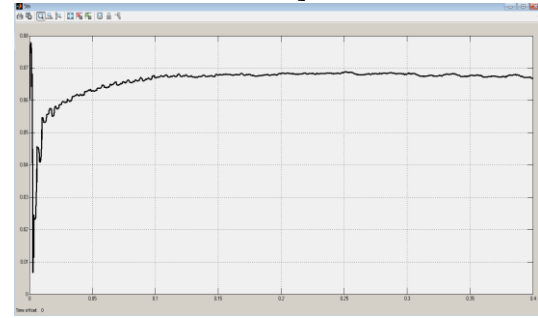
Set the reference fundamental frequency ($f=49.9$ Hz).

Initial amplitude vector:

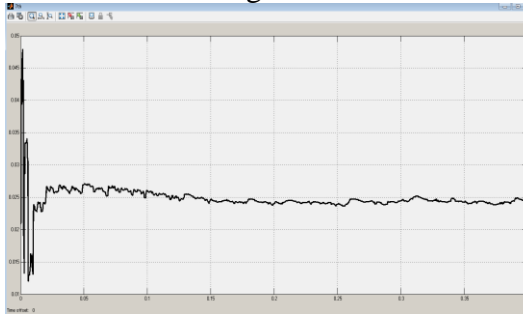
$$w=[1.2 \quad 0.7 \quad -0.045 \quad 0.04 \quad 0.006 \quad 0.02 \quad -0.001 \quad -0.010]'$$



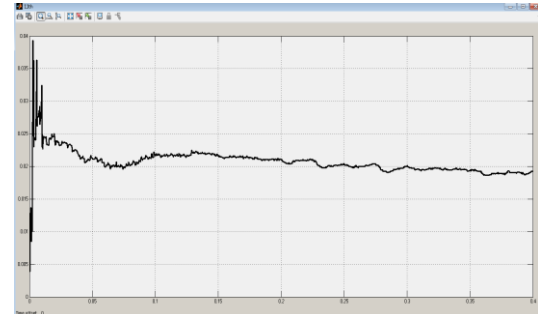
(a) Amplitude Tracking of Fundamental signal.



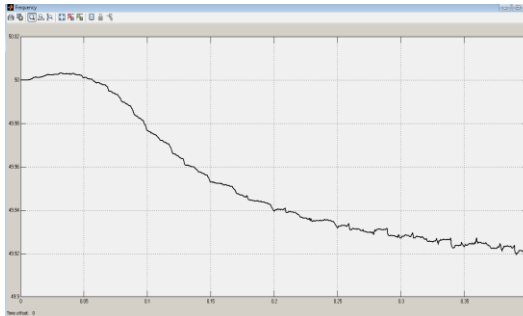
(b) Amplitude Tracking of 5th harmonic.



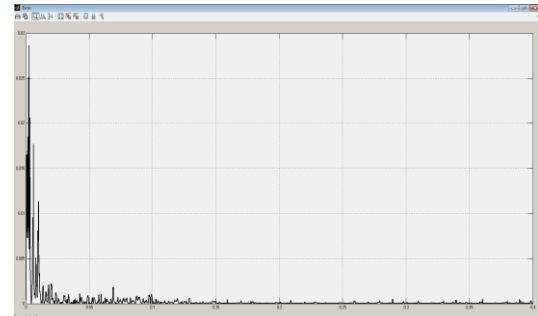
(c) Amplitude Tracking of 7th harmonic.



(d) Amplitude Tracking of 11th harmonic.



(e) Tracking of Fundamental Frequency.



(f) Tracking of Mean Square Error.

Figure 4- 14 Xilinx Simulation in Case 2.

In this thesis, we implemented the method on Xilinx Spartan3E-1200 FPGA. From the results from Xilinx system, they show the tracking results when the fundamental frequency and amplitude changed. In case 1 when the fundamental frequency increased, the proposed ADALINE can track the change of frequency. In case 2 when the fundamental frequency reduced, the proposed ADALINE also can track the change of frequency.

From the results from MATLAB code simulation and Xilinx system simulation, the results are almost same. So we know that the method proposed can work well not only in the MATLAB simulation but also in the Xilinx simulation. Next step, in order to make sure it can work in the real system, we will do the hardware in-the loop in next step.

4.3 On-line Dynamic Tracking

The learning parameter determine the convergence speed in the LMS algorithm. The learning parameter larger, we can get convergence faster. However, if we make the learning parameter too large, the algorithm will become unstable. Then, we would like to use a big learning parameter during the convergence stage. On the other hand, we would like to use a small learning parameter during the tracking stage. In order to get different learning parameter in different stage, we used equation (3.21) and equation (3.22) for adaptively update learning parameter in this thesis. For example in equation (3.21), we can make the α_0 small for convergence and make the c_1 very big for tracking. However, if c_1 is too big, even $V(k)$ is very small, $c_1 * V(k)$ will be very big, it's very hard to keep the stable in the tracking stage. In the present power system, the fundamental frequency is dynamic. So, in order to dynamic track the fundamental frequency and make sure that we can get a good ability of tracking, we will use registers with reset port in Figure 4-15.

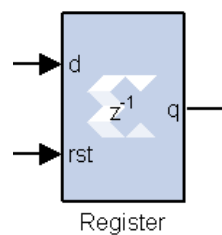


Figure 4- 15 Register with reset port.

We also get the results for tracking different fundamental frequency in Figure 4-16 to Figure 4-23. Here, we set the same initial values of amplitudes and fundamental frequency in every case.

Case a: $f=50.2\text{Hz}$

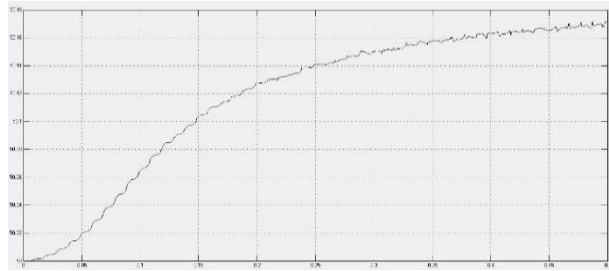


Figure 4- 16 Case a (Reference fundamental frequency is 50.2 HZ).

Case b: $f=50.15\text{Hz}$

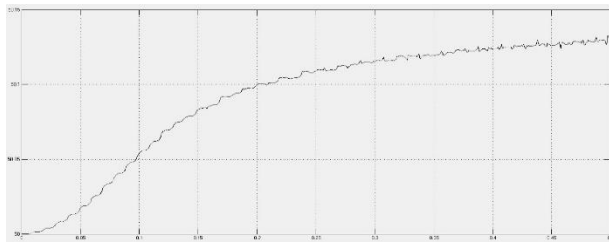


Figure 4- 17 Case b (Reference fundamental frequency is 50.15 HZ).

Case c: $f=50.1\text{Hz}$

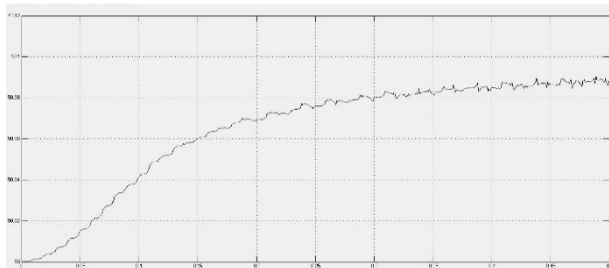


Figure 4- 18 Case c (Reference fundamental frequency is 50.1 HZ).

Case d: $f=50.05\text{Hz}$

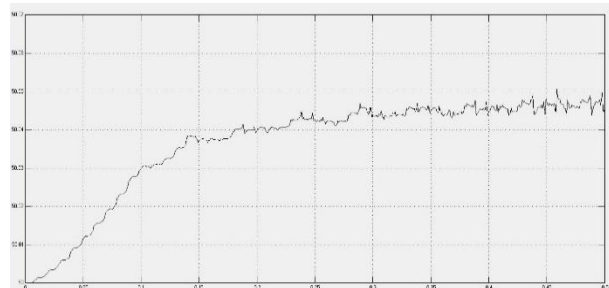


Figure 4- 19 Case d (Reference fundamental frequency is 50.05 HZ).

Case e: $f=49.95\text{Hz}$

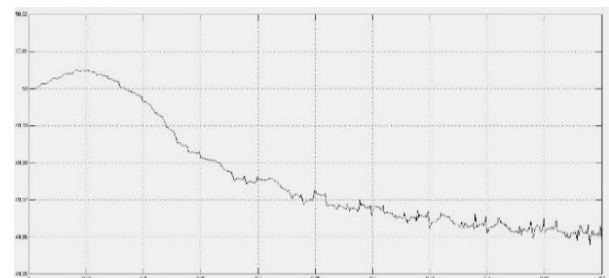


Figure 4- 20 Case e (Reference fundamental frequency is 49.95 HZ).

Case f: $f=49.9\text{Hz}$

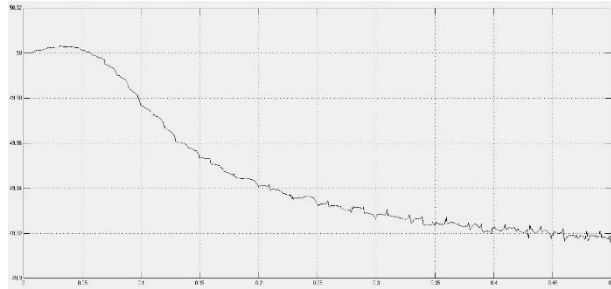


Figure 4- 21 Case f (Reference fundamental frequency is 49.9 HZ).

Case g: $f=49.85\text{Hz}$

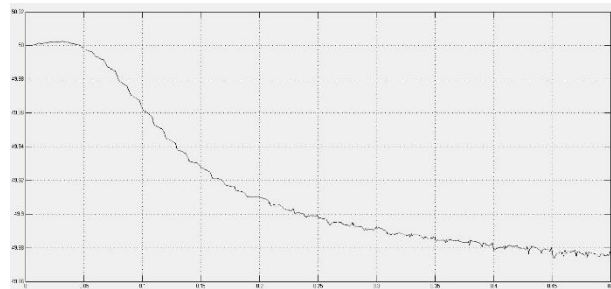


Figure 4- 22 Case g (Reference fundamental frequency is 49.85 HZ).

Case h: $f=49.8\text{Hz}$

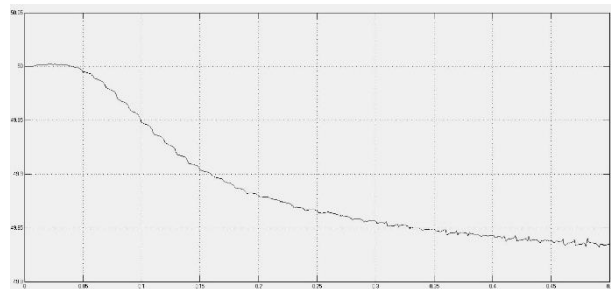


Figure 4- 23 Case h (Reference fundamental frequency is 49.8 HZ).

From the figures, we know that the amplitudes and fundamental frequency both get the convergence in 0.4s. And we used the registers with reset port, so we can reset the register every 0.4s. Then we can track the distorted signal with dynamic fundamental frequency.

4.4 Hardware-in-the-Loop Verification

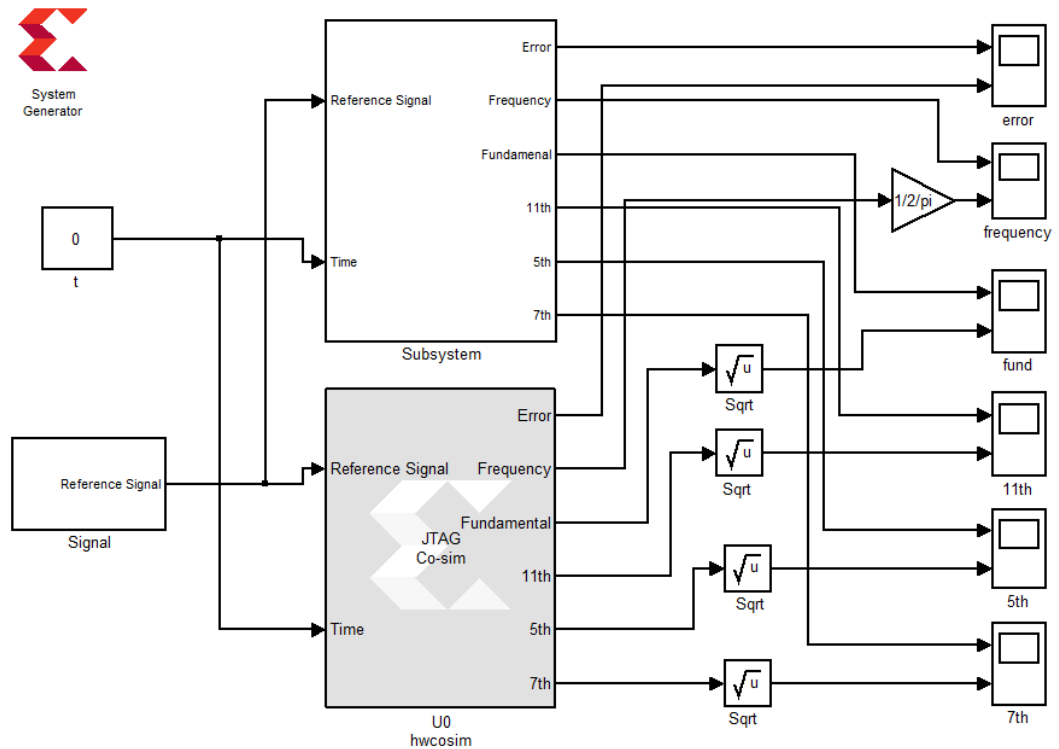


Figure 4- 24 Complete Design Ready for the Hardware-in-the-Loop Simulation.

In order to make sure that the proposed algorithm can work on the FPGA board. We also did the hardware-in-the-loop verification. The design of hardware-in-the-loop has been shown in the Figure 4-24.

In the FPGA experiments, we use Simulink model built the distorted signal show in Figure 4-25.

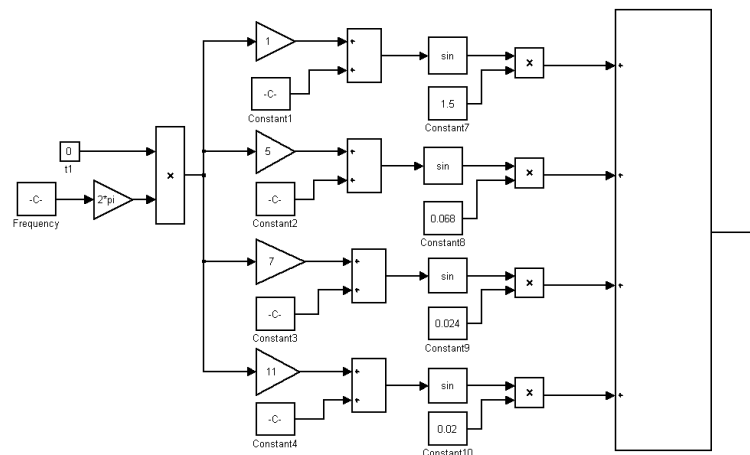


Figure 4- 25 Model design of the distorted signal.

In the signal model, we use the sampling time is 0.0002s, and the sampling time is same as the FPGA simulation time.

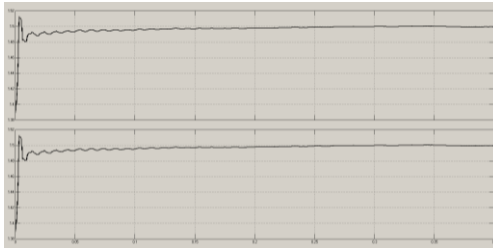
Case1:

Set the initial fundamental frequency ($f=50$ Hz).

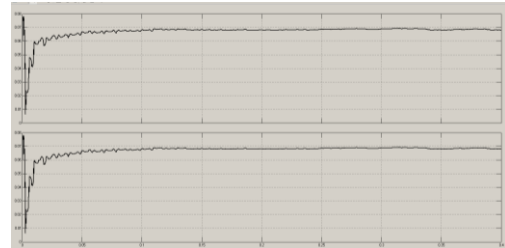
Set the reference fundamental frequency ($f=50.1$ Hz).

Initial amplitude vector:

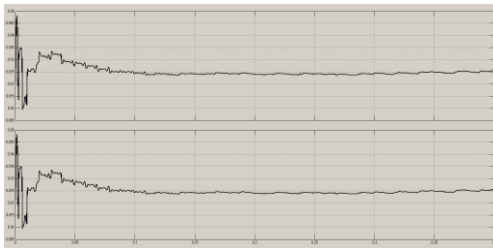
$w=[1.2 \ 0.7 \ -0.045 \ 0.04 \ 0.006 \ 0.02 \ -0.001 \ -0.010 \]'$.



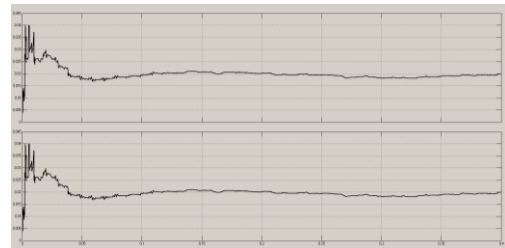
(a) Amplitude Tracking of Fundamental signal



(b) Amplitude Tracking of 5th harmonic



(c) Amplitude Tracking of 7th harmonic



(d) Amplitude Tracking of 11th harmonic

Figure 4- 26 Simulation result showing simulator output on top and hardware output at the bottom (Amplitude Tracking).

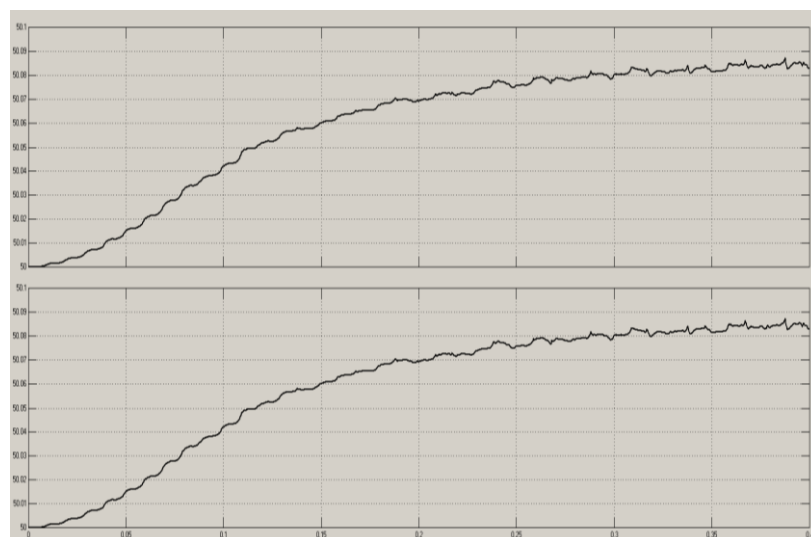


Figure 4- 27 Simulation result showing simulator output on top and hardware output at the bottom (Frequency Tracking).

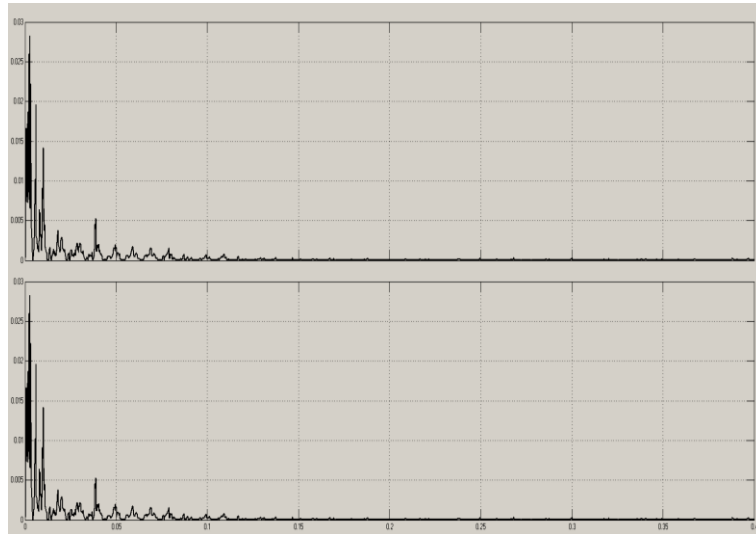


Figure 4- 28 Simulation result showing simulator output on top and hardware output at the bottom (Error).

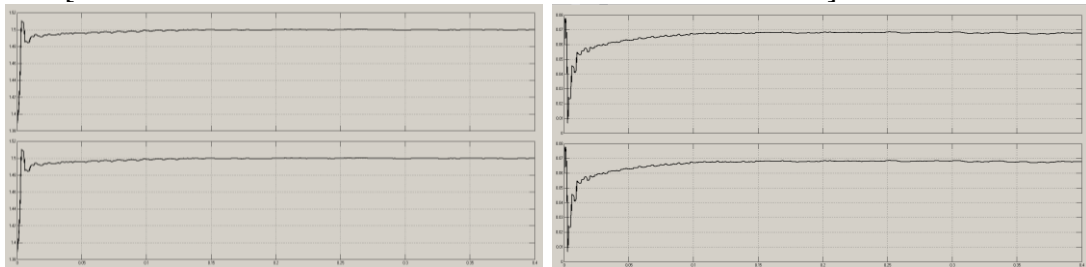
Case2:

Set the initial fundamental frequency ($f=50$ Hz).

Set the reference fundamental frequency ($f=49.9$ Hz).

Initial amplitude vector:

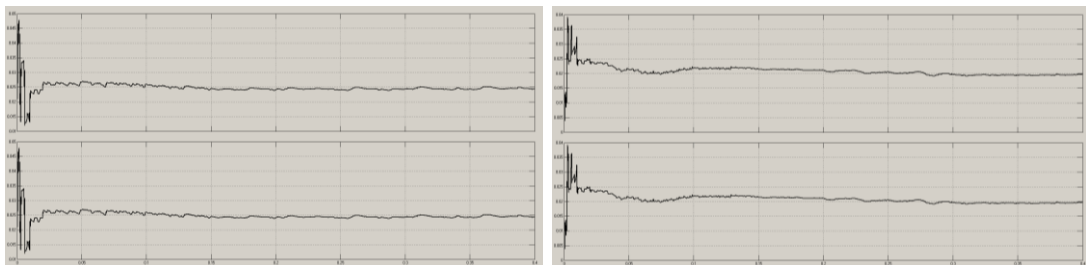
$w=[1.2 \quad 0.7 \quad -0.045 \quad 0.04 \quad 0.006 \quad 0.02 \quad -0.001 \quad -0.010]'$.



(a) Amplitude Tracking of Fundamental

(b) Amplitude Tracking of 5th harmonic

signal



(c) Amplitude Tracking of 7th harmonic

(d) Amplitude Tracking of 11th harmonic

Figure 4- 29 Simulation result showing simulator output on top and hardware output at the bottom (Amplitude Tracking).

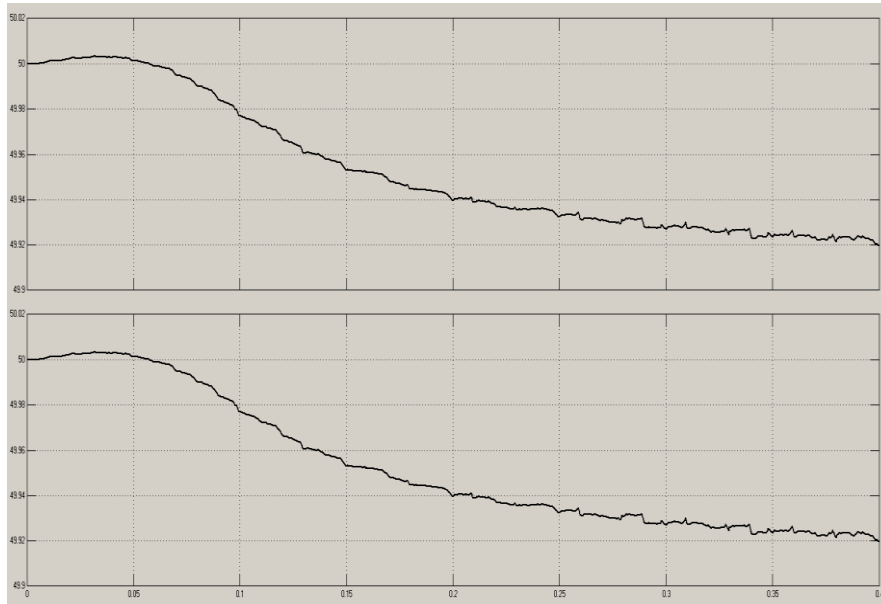


Figure 4- 30 Simulation result showing simulator output on top and hardware output at the bottom (Frequency Tracking).

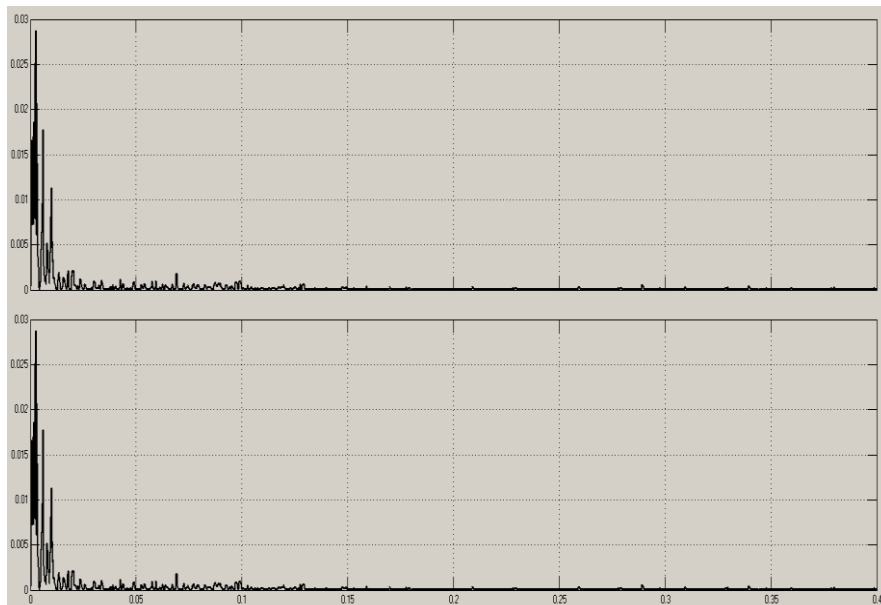


Figure 4- 31 Simulation result showing simulator output on top and hardware output at the bottom (Error).

After we implemented the proposed method on a Xilinx Spartan3E-1200 FPGA (Field Programmable Gate Array). The hardware-in-the loop verification results show that the proposed ADALINE adaptive filter can not only accurately estimate the dynamic harmonic amplitudes and phases but also adaptively track the dynamic fundamental frequency in the present AC power system. Due to the limitation of

resource, we need to reduce the bits of some part. The amplitudes of high order harmonic maybe a little different from the simulation. But the higher order of the harmonics, the amplitude will be smaller. So we can ignore the difference between the results of simulation and the results from hardware-in-the-loop. Then we can get the target of tracking the components of the harmonics.

Chapter 5. Conclusions, Problems and Suggestions

5.1 Conclusions

This thesis has presented the adaptive linear neural network (ADALINE) based on the LMS algorithm which is used for minimizing the mean-square-error for tracking power system harmonics. The learning parameters were verified by simulating the LMS algorithm on MATLAB. Then, the design has been implemented by Xilinx System Generator running in the Simulink environments. The simulation experiments showed that the weight adaptation along with an adaptive algorithm can accurately and simply estimate the signal. The adaptive parameters was also used in bringing a better convergence and in tracking the harmonic components.

Finally, we successfully implemented the proposed method on a Xilinx Spartan3E-1200 FPGA (Field Programmable Gate Array). The hardware-in-the loop verification results show that the proposed ADALINE adaptive filter is not only accurately estimate the dynamic harmonic amplitudes and phases but also adaptively track the dynamic fundamental frequency in the present AC power system. Moreover, the adaptive learning parameters bring the faster and more accurate convergence.

5.2 Problems and Suggestions

- 1) Due to amplitude of the distorted voltage in the real power system is different, so we should define different values of α , α_ω , c_1 , c_2 in the different conditions. For example, we should define different values for the motors with different rated voltage.
- 2) In order to get a faster convergence, we'd better to set the initial values of the amplitudes and the fundamental frequency close the values of the rated voltage. However, the values of α , α_ω , c_1 , c_2 is related to the initial values of the registers. So the values of α , α_ω , c_1 , c_2 would be different in the conditions with different initial values.
- 3) Due to the amplitude of random noise is very small and the amplitudes of the higher order harmonics are also very small. It is very hard to track the random noise. So, in this thesis, we have to ignore the random noise. Then we will get some errors in the real system. However, the proposed method in this thesis can

make sure the error of the voltage within the allowable range. Then, we can make the electronic devices work in the safe situations.

References

- [1] A.A. Girgis and F.M. Ham, "A quantitative study of pitfalls in FFT", *IEEE Trans. Aersoso. Electron.Syst.*, Vol. 16, No. 4, pp. 434-439, 1980.
- [2] S. Liu, "An adaptive Kalman filter for dynamic estimation of harmonic signals", *Proc.IEEE/PES and NTUA*, October 14-16, 1998.
- [3] A.A. Girgis, W.B. Chang, E.B. Mal, "A digital recursive measurement scheme for on-line tracking of power system harmonics", *IEEE Trans. on Power Delivery*, Vol. 6, No. 3, pp. 1153-1160, July 1991.
- [4] P.K. Dash, D.P. Swain, A.C. Liew, and S. Rahman, "An adaptive linear combiner for on-line tracking of power system harmonics", *IEEE Trans. on Power Systems*, Vol. 11, No. 4, pp. 1730-1735, Nov. 1996.
- [5] R.E. Shatshat, M. Kazerani and M.M.A. Salama, "Modular approach to active power-line harmonic filtering", *Proc. IEEE Power Electronics Specialists*, pp. 223-228, 1998.
- [6] R.E. Shatshat, M. Kazerani and M.M.A. Salama, "On-line tracking and mitigation of power system harmonics using ADALINE-based active power filter system", *proc. IEEE Electrical & Computer Engineering*. Vol.4, pp. 2119 - 2124, 2004.
- [7] S. Osowski, "Neural network for estimation of harmonic components in a power system", *Proc. IEE*, Vol.139, No.2, pp. 129-135.
- [8] M.T. Hagan, H.B. Demuth, M.H. Beale. "Neural network design", Boston: Pws Pub., 1996.
- [9] P.K. Dash, S.K. Panda, B. Mishra and D.P. Swain, "Fast estimation of voltage and current phasors in power networks using an adaptive Neural Network", *IEEE Trans. Power Systems*, Vol. 12, No. 4, pp. 1494-1499, Nov. 1997.
- [10] B. Widrow, M. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation", *Proc. IEEE*, Vol. 78, No. 9, pp. 1415-1442, Sep. 1990.

- [11] S.M. Wang, W. Suntiamomtut, N. Jindapetch and J. Qiufan, "Scheduling & resources sharing technique for adaptive LMS filter", *Proc. ECTI-CON*, pp. 114 – 117, 2011.
- [12] S.A. Jimaa, A. Al-Simiri, R.M. Shubair, et al. "Convergence evaluation of variable step-size NLMS algorithm in adaptive channel equalization", *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2009: 145-150.
- [13] P.K. Ray, B. Subudhi. "Ensemble-Kalman-filter-based power system harmonic estimation", *IEEE Transactions on Instrumentation and Measurement*, 2012, 61(12): 3216-3224.
- [14] Z. Yonghui, C. Xi, Z. Xiyuan. "Power quality analysis based on all-phase FFT", 2010 International Conference on Intelligent System Design and Engineering Application (ISDEA). IEEE, 2010, 2: 69-72.
- [15] Y.H. Zhang, X. Chen, X.Y. Zhang, "Power harmonic analysis based on all-phase FFT", *Proc. ICSPS*, pp. 576 – 579, 2010.
- [16] P.K. Dash, K.R. Krishnanand, M. Padhee. "Fast recursive Gauss–Newton adaptive filter for the estimation of power system frequency and harmonics in a noisy environment", *IET generation, transmission & distribution*, 2011, 5(12): 1277-1289.
- [17] B. Widrow, S.D. Stearns. "Adaptive signal processing", Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985, 491 p., 1985, 1.
- [18] N. Jindapetch, S. Chewae, P. Phukpattaranont. "FPGA implementations of an ADALINE adaptive filter for power-line noise cancellation in surface electromyography signals", *Measurement*, 2012, 45(3): 405-414.
- [19] U. Meyer-Baese. "Digital signal processing with field programmable gate arrays", Heidelberg: Springer, 2007.
- [20] B. Vasumathi, S. Moorthi. "Harmonic estimation using Modified ADALINE algorithm with Time-Variant Widrow-Hoff (TVWH) learning rule", *2011 IEEE Symposium on Computers & Informatics (ISCI)*. IEEE, 2011: 113-118.

- [21] W. Hongbo, H. Rong, Q. Qi, et al. "Improved teager energy operator for on-line tracking of power grid voltage", 2014 33rd Chinese Control Conference (CCC). IEEE, 2014: 8234-8238.
- [22] C. Chen, C.Y. Cheng, Y.C. Chen. "Estimation of voltage variation events with adaptive linear filtering", 2014 International Symposium on Computer, Consumer and Control (IS3C). IEEE, 2014: 597-600.
- [23] J.J. Wang, N. Jindapetch, K. Sengchuai, "Adaptive on-line tracking of power system harmonics using ADALINE", Signal Processing of the ECTI International Conference(ECTI-CON 2015), Hua Hin, Thailand, June 2015.

VITAE

Name Miss. Wang Jing Jing

Student ID 5610120092

Educational Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Engineering	Anhui Polytechnic University	2013

List of Publication and Proceeding

- [1] J.J. Wang, N. Jindapetch, K. Sengchuai, "Adaptive on-line tracking of power system harmonics using ADALINE", Signal Processing of the ECTI International Conference(ECTI-CON 2015), pp. 226, Hua Hin, Thailand, June 2015.