SYNCHRONIZATION CONTROLLER TO SOLVE THE MISMATCHED SAMPLING RATES FOR ACOUSTIC ECHO CANCELLATION

by

ADEL NADHEM NAEEM

Thesis submitted in fulfillment of the requirements

for the degree of

Doctor of Philosophy

February 2015

ACKNOWLEDGEMENTS

First, the favour, above all, before all and after all, is entirely Allah's, to whom my never-ending thanks and praise are humbly due.

I would like to take this opportunity to convey my sincere thanks and deepest gratitude to my supervisors Prof. Dr. Sureswaran Ramadass for giving me his support, motivation, time, and guidance during the preparation of this thesis.

Moreover, I would like to thank those who are always in my heart; my father for his endless and continuous encouragement and constant support, my mother for her continuous prayers and inspiration, my wife for her patience and encouragement, my brothers and sisters, my uncles and aunts.

Many thanks to USM and IPS to give me the opportunity to get "USM Fellowship" which helped me with my study, and many thanks to all NAv6 center staff and students.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

A/D	Analog to digital converter
AEC	Acoustic echo cancellation
APA	Affine projection algorithm
ASRC	Arbitrary sampling rate conversion
CD	Compact Disc
CPU	Central processing unit
D/A	Digital to analog converter
dB	Decibel
DCAF	Drift-compensated adaptive filtering
DFT	Discrete Fourier transform
DSP	Digital signal processors
DTD	Double talk detection
ERLE	Echo return loss enhancement
Fast-LMS	Fast least mean square
FDAF	Frequency domain adaptive filter
FFT	Fast Fourier transform
FIR	Finite impulse response
FOE	Frequency offset estimation
GB	Giga byte
GCD	Greatest common divisor
H_L	Low-pass filter
Hz	The hertz
I/O	Input/output

IFFT	Inverse fast Fourier transform
IIR	Infinite impulse response
IP	Internet protocol
LMS	Least mean square
ms	millisecond
MSE	Mean-squared error
NLMS	Normalized least mean square
NR	Noise reduction
PC	Personal computer
PSTN	Public switched telephone network
RAM	Random-access memory
Ratchet FAP	Ratchet fast affine projection
RIR	Room Impulse Response
RLS	Recursive least square
RNG	Random Number Generator
RSO	Relative sample offset
s	Second
SISO	Single-input, single-output
SRC	Sampling Rate Conversion
VLSI	Very large scale integration processors
VoIP	Voice over IP

PENGAWAL PENYELARASAN UNTUK MENYELESAIKAN KETIDAKSEPADANAN KADAR PENSAMPELAN BAGI PEMBATALAN GEMA AKUSTIK

ABSTRAK

Aplikasi-aplikasi Suara melalui IP (VoIP) yang menggunakan set komunikasi bebas tangan semakin meluas digunakan. Masalah gema akustik yang boleh terjadi dalam sistem-sistem komunikasi bebas tangandapat diselesaikan dengan menggunakan pembatalan gema akustik (AEC). Walau bagaimanapun, sebelum ini AEC hanya boleh dicapai melalui pemproses penyepaduan skala sangat besar (VLSI) atau pemproses-pemproses isyarat digital yang direka khas untuk AEC. Namun begitu, dengan peningkatan kuasa komputasi komputer peribadi, pemprosesan isyarat masa nyata dan AEC menggunakan komputer peribadi boleh dilakukan pada masa sekarang. Malangnya, ketidakpadanan kadar pensampelan boleh berlaku antara komponen-kompenen komputer peribadi dan ia akan menyebabkan kegagalan sistem AEC tersebut. Tanpa pengubahsuaian, sistem AEC tidak boleh berfungsi dengan sempurna tanpa kadar pensampelan isyarat-isyarat input yang sepadan. Sistem yang dicadangkan akan menyelesaikan masalah ini dengan menggunakan dua algoritma penukaran kadar pensampelan untuk mengubah kadar-kadar pensampelan kedua-dua isyarat input ke sistem AEC tersebut. Sistem AEC yang dicadangkan akan menyegerakkan dan mengimbangi isyarat-isyarat input supaya penapis adatif akan berfungsi dan dapat dimanfaatkan oleh kesemua pengguna sesi persidangan VoIP. Tesis ini menerangkan sistem AEC yang dicadangkan di mana sistem inimenggunakan penapis adaptif "Fast-LMS" untuk menganggarkan dan

membatalkan gema akustik daripada dua isyarat input yang berlainan.. Beberapa eksperimen simulasi telah dijalankan untuk menguji keberkesanan empat sistem AEC yang berbeza untuk menyerlahkan masalah-masalah yang timbul akibat ketidakpadanan kadar pensampelan dan kesannya terhadap sistem-sistem AEC. Faktor penapis adaptif yang digunakan dalam tesis ini adalah saiz langkah = 0.3 dan saiz blok = 2048. Berbanding dengan sistem AEC yang telah wujud, hasil eksperimen menunjukkan bahawa sistem AEC yang dicadangkan di mana sistem tersebut memberikan kadar pensampelan input dan output (8000Hz) yang tetap dapat mengatasi penukaran kadar pensampelan input.

SYNCHRONIZATION CONTROLLER TO SOLVE THE MISMATCHED SAMPLING RATES FOR ACOUSTIC ECHO CANCELLATION

ABSTRACT

Voice over Internet Protocol (VoIP) applications are extensively used for hands-free communication (audio conferencing and video conferencing). Although hands-free communication systems may encounter acoustic echo problems, such problems can be solved using acoustic echo cancellation (AEC). Previously, AEC had been achieved only through customized large-scale integration processors or digital signal processors that were specially designed for AEC. However, the computational power of personal computers (PCs) has grown over time, and real-time signal processing and AEC in PC environments are now possible. Nevertheless, sampling rate mismatch between personal computer components may occur and induce AEC system failure. An AEC system cannot work properly without matching the sampling rates of the input signals. Therefore, the system proposed in this paper addresses this issue using two different sampling rate conversions to modify the sampling rates of both input signals to the AEC system and fixing the signals to the desired sampling rate of the VoIP sessions. The proposed AEC system uses a synchronization controller to control feeds and synchronize the input signals to the AEC system. Hence, the adaptive filter works and all of the users of the VoIP conference session can benefit from having the fixed sampling rate signal match the desired sampling rate of the VoIP systems. This paper describes the proposed AEC system that uses a fast least mean squares (fast-LMS) adaptive filter to estimate and eliminate acoustic echo from two input signals. Several simulation experiments were conducted to test

the effectiveness of four different AEC systems to highlight the problems caused by sampling rate mismatch and their effects on AEC systems. The adaptive filter factors used in this paper are step size = 0.3 and block size = 2048. Compared with other existing AEC systems, the experiment results in this reserach indicated that the proposed AEC system that provides fixed sampling rate inputs and output (8000 Hz) can handle the changes in input sampling rates.

CHAPTER ONE

INTRODUCTION

Video conferencing, teleconferencing, and hands-free telephone systems are important communication tools because of their effect on people's personal lives and on business communication. The increased use of hands-free telephone systems has prompted researchers to improve voice quality by reducing signal noise, delays, and echoing. Echo, defined as a delayed and distorted version of an original sound that is reflected back to the source, is one of the important challenges facing such improvements in hands-free telephone systems (Fukui, Shimauchi, Kobayashi, Hioka, & Ohmuro, 2014; Furui & Sondhi, 1992). Two types of echo can occur: electrical and acoustic. The search for improved voice quality has led researchers to study the causes of acoustic echo and potential methods for removing it (Mondol & Zhou, 2014; Raghavendran, 2003). One solution to the echo problem is acoustic echo cancellation (AEC), which uses an adaptive filter to model room acoustics and identify the acoustics from microphone and speaker signals (acoustic echo). The filter calculates an estimated microphone signal from the speaker signal. This estimated microphone signal is then subtracted from the real microphone signal and, with further feedback, the resulting signal no longer contains the speaker signal (Adrian, 2004; Bispo & Freitas, 2013; Storn, 1996; Talagala, Zhang, & Abhayapala, 2013).

1.1 Background

Today, everyone uses the internet to communicate with each other. One method of voice communication using the internet is called Voice over IP (VoIP). Hands-free systems are one way to use VoIP like conference systems (or regular loudspeaker and microphone voice chat) to allow multiple users at the same location to participate in a VoIP session. Another advantage of using a hands-free system is that it allows a user to have both hands free and to move freely about the room.

Echo is a known problem that negatively impacts telephone communication systems (Storn, 1996). Both electrical and acoustic echo can be encountered in telephone communication systems. Electrical echo occurs due to the impedance mismatch at various points along the transmission medium. This type of echo can occur in the Public Switched Telephone Network (PSTN), mobile, and IP phone systems. The electrical echo is created at the hybrid connections where the subscriber's two-wire lines are connected to four-wire conversion points (Lu, 2007). This type of echo is not included in the scope of this thesis. Acoustic echo usually occurs in hands-free telephone communication systems because of coupling between the loudspeaker and the microphone. The presence of strong acoustic coupling between the loudspeaker and microphone can produce an echo that makes conversation difficult or at least less intelligible. Furthermore, the acoustic system can become unstable and produce a loud howling noise at specific frequencies where the time delay in the coupling provides positive feedback in the system (Raghavendran, 2003). Figure 1.1 provides a general explanation of the acoustic echo problem: Signal (x) leaves the speakers and is then reflected back and captured by the microphone, which causes acoustic echo.

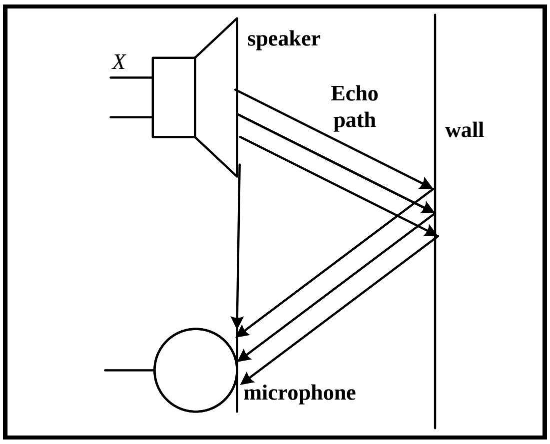

Figure 1.1: Illustration of how acoustic echo occurs

Several methods can be used to eliminate or reduce acoustic echo in VoIP systems (Adrian, 2004; Raghavendran, 2003):

- **Headsets** are the most powerful and simplest tools used to avoid acoustic echo, as they do not use external speakers that can acoustically couple to the microphone. However, the increasing use of hands-free gadgets that require speakers with a separate microphone has made this solution ineffective (Storn, 1996).
- **Negative feedback** reduces the overall signal amplification. If the system amplification is less than one, the howling will fade, but this solution can result in low volume (Adrian, 2004).
- **Some forms of echo suppression** can be used with a half-duplex system, in which only one side can talk at a time. The echo suppressor works by detecting signals on one side and shutting down the microphone on the other side. The speaker signal on the non-active side does not travel back to the

active side; thus, there is no echo. However, it causes significant problems in conversations when people try to talk at the same time; therefore this type suppressor is not used with full-duplex telephone systems (Adrian, 2004; Storn, 1996).

- The concept of **Acoustic Echo Cancellation (AEC)** is shown in Figure 1.2. The signal (*x*) comes from the far-end (the second party of the VoIP session) and is played out from the speaker. By direct coupling, or after being reflected back by different surfaces, the acoustic echo represented by the signal (*d*) is captured by the microphone, along with the near-end (the first party of the VoIP session) signal (*s*) and the noise signal (*n*). Therefore, an adaptive filter that models the room acoustics is needed to remove the echo signal (*d*) from the near-end signal (*s*). The adaptive filter should identify the acoustic echo signal from given microphone and speaker signals; the adaptive filter will then calculate an estimated echo signal from the returned speaker signal. This estimated echo signal is subtracted from the real microphone signal and the result (*e*) is fed back to the adaptive filter so that the resulting signal (*e*) no longer contains the speaker signal (acoustic echo) (Adrian, 2004; Schmidt, 2004; Shi, 2008; Storn, 1996). A double talk detector (DTD) is used with an AEC system to sense when far-end speech is corrupted by near-end speech; the role of this system is to freeze adaptation of the adaptive filter when near-end speech is present; this action prevents divergence of the adaptive filter (Raghavendran, 2003).

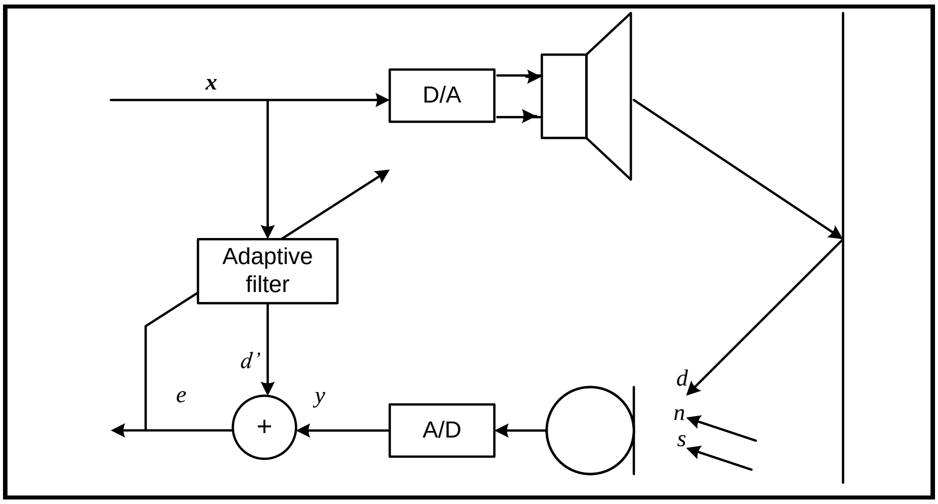

Figure 1.2: General concept of the AEC process

Real-time implementation of an AEC may be performed by utilizing both a very large-scale integration (VLSI) processor and a digital signal processor (DSP). These processors are specially designed for AEC and signal processing tasks. Because there has been a revolution in the field of personal computers (PCs), it now is possible to implement an AEC in PCs. However, many problems remain when running an AEC system as software on a PC. One problem is that the sound I/O device of the PC may have different clock sources for input and output (Carôt & Werner, 2009; Robjohns, 2003b). Thus, each pathway may have a different sampling rate. A difference in sampling rate affects how the AEC system works by causing a change in the echo path. Thus, the adaptive filter will not be able to properly calculate the estimated far-end contribution to the near-end signal, as the latter does not match the sampling rate of the input (far-end) signal. Sampling rate mismatch may occur in two ways (Ding & Havelock, 2010; Frunze, 2003; Robledo-Arnuncio, Wada, & Juang, 2007). First, AEC is affected when playing CD-quality music or any other media file with sound when the playback sampling rate is higher than the capture rate (Stokes & Malvar, 2004). Second, the different sampling rates of the D/A (Digital to Analog) converter and the A/D (Analog to Digital) converter of low-cost PC audio hardware may increase or decrease delay, thereby causing lost or repeated samples (Ding & Havelock, 2010; Pawig, Enzner, & Vary, 2010).

1.2 Research Problem

Researchers have examined several means of providing good quality and clear voice reproduction. However, acoustic echo remains a crucial challenge in advancing hands-free VoIP systems (Sugiyama, 2004). The key to maintaining and improving the perceived voice quality of a call is effectively removing acoustic echoes, which are inherent within the telecommunications network infrastructure (Raghavendran, 2003).

Running an AEC system in a PC to cancel the reflected acoustic echo during a hands-free VoIP session or video conference may be affected by the digital sound card and played media during the session. The major problem that affects the AEC system is the mismatch in the sampling rates of input and output signals, which occurs when the sound I/O devices in a PC have different clock sources with different sampling rates. This feature induces AEC system failure because of the mismatch in the sampling rate of the input signals. Besides, playing CD-quality music or any other sound media file concurrently with a hands-free voice chat or video conferencing session affects the session users because of the sound distortion that results from the reflection of the played media echo. AEC systems cannot remove the played media echo because the playback sampling rate of sound media files is typically higher than the sampling rate of the VoIP session. Therefore, AEC systems should be improved by investigating the effect of the mismatched sampling rates of input signals on system performance in VoIP applications (Ding & Havelock, 2010; Pawig et al., 2010). Different sampling rates of the input signals to the AEC system during a hands-free session will induce a synchronization problem. Moreover, hands-free VoIP session users commonly use different PCs; thus, each user will have a different sampling rate as an output from his/her PC, which will subsequently cause problems in the output signal sampling rate.

1.3 Research Objective

This research primarily aims to enhance the performance of AEC systems in hands-free VoIP applications and video conferencing systems using a new AEC system design that removes the reflected acoustic echo signal from PC speakers that is captured by the PC microphone. A synchronization controller is used to solve the mismatched sampling rates issue for acoustic echo cancellation system, by fixing the sampling rates of the input and output signals.

1.4 Research Contribution

The research presented in this thesis contributes a new design for acoustic echo cancelation system. The proposed AEC system design incorporates three models, including the two existing models, namely, sample rate conversion (SRC), which fixes the sampling rate of the played media during a VoIP session, and arbitrary sample rate conversion (ASRC), which fixes the signal sampling rate of the microphone signal. The third model is the synchronization controller model that corrects the misalignment of input signals after resampling and feeds the adaptive filter with the fixed sampling rate signals.

1.5 Research Steps

To achieve the objective this research, the following research stages were conducted: literature review, defining the research problems, proposing the new AEC system, designing and simulating the system, and evaluating the proposed AEC system (see Figure 1.3).

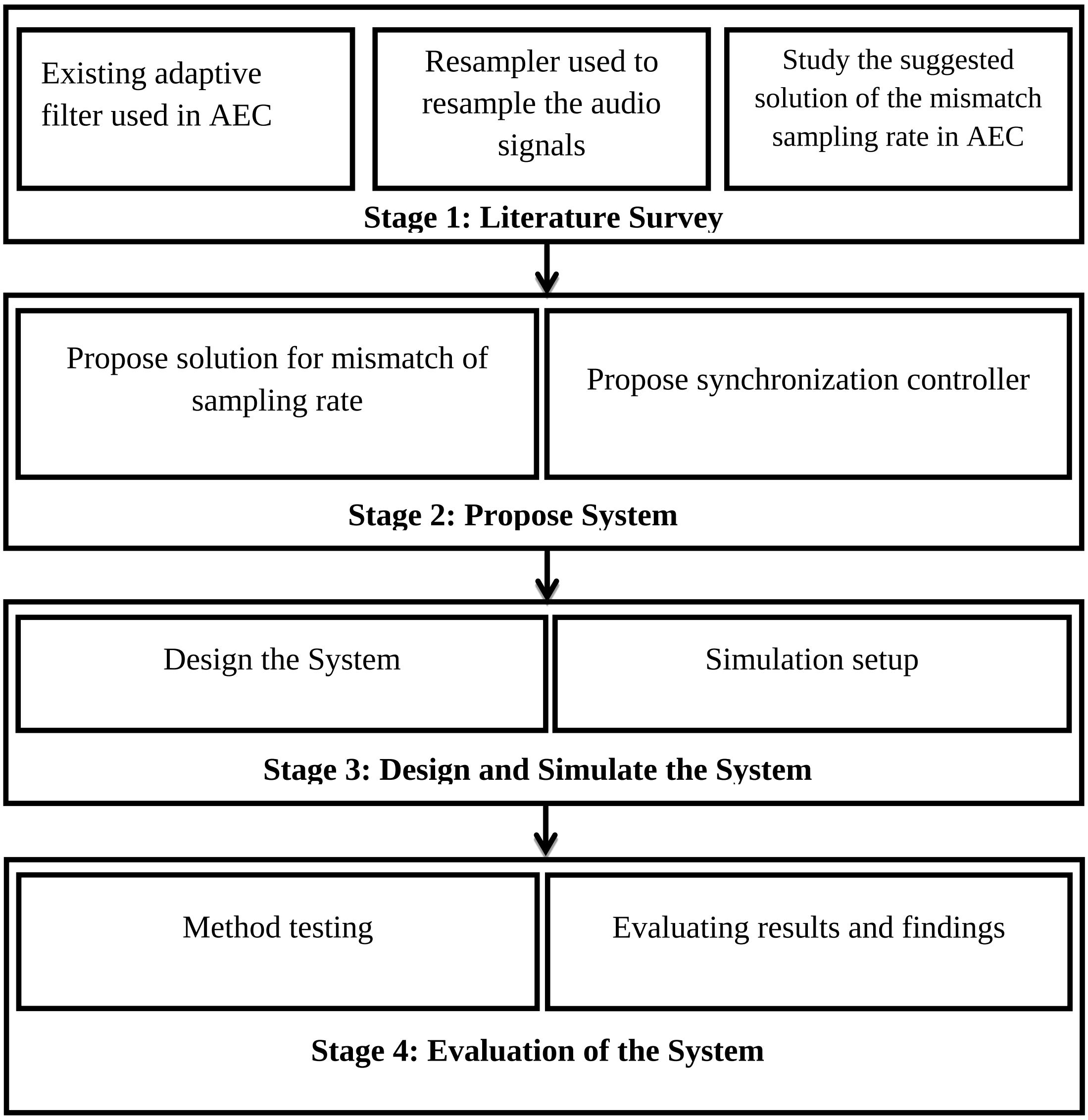

Figure 1.3: Stages of the research.

1.6 Thesis Organization

This thesis is organized into five chapters. **Chapter 1** was already presented. **Chapter 2** reviews the fundamental concepts related to AEC, the adaptive filters used, and the sampling techniques employed. The related studies are discussed and compared at the end of the chapter.

Chapter 3 describes the proposed solution to improve the AEC system, the methodology, the sampling rate conversion used, the synchronization method employed to provide proper input signals to the adaptive filter of the AEC system, and the means through which the proposed AEC method provides fixed sampling rates.

Chapter 4 discusses the test results for the proposed AEC system, as well as the research findings and their comparison to the results from other studies.

Chapter 5 presents the conclusions and recommendations for future research.

CHAPTER TWO

LITERATURE REVIEW

To operate, Voice over internet protocol (VoIP) applications, digital sound card, microphone, speakers, and an operating system require for its operation. However, in an age when complex systems are composed of interchangeable subsystems, these interacting elements, while functionally appropriate and operational, may not be fully compatible. Most electronic incompatibilities are eliminated by proper design. Nevertheless, some requirements are subtle and do not become apparent until applications beyond the original functionality are implemented (Deng, Bao, & Li, 2013). One such problem is the sound I/O device on a PC, in which the input and output may have different clock sources (Carôt & Werner, 2009; Robjohns, 2003a) and thus have different sampling rates. Such deficiencies may cause echo path changes in two ways with acoustic echo cancellation (AEC) (Ding & Havelock, 2010; Frunze, 2003). First, AEC will be affected when playing CD-quality music or any other media with sound for which the playback sampling rate of the sound file is higher than the capture rate at the microphone (Stokes & Malvar, 2004). Second, the different sampling rates of the D/A converter and the A/D converter of low-cost PC audio hardware may increase or decrease delay, thereby causing lost or repeated samples (Pawig et al., 2010). This chapter provides a review of adaptive filters, methods of sampling and resampling (rational, irrational) of digital signals, and interpolation and decimation methods. At the end of this chapter, related works are discussed and compared.

2.1 Adaptive Filters Used in Acoustic Echo Cancellation

One of the important way to remove the acoustic echo in the hands free system during the VoIP session is using AEC system, where is depend on using an adaptive filter to estimate the echoed signal and removed from the microphone signal. The main types of digital filtering that use for digital signal processing are finite impulse response (FIR) and infinite impulse response (IIR). IIR and FIR achieve the same performance with different coefficients and computation, as the complexity of the filter grows, the order of the IIR filter increases a lot and that result increases of the computational cost. However, FIR filters used in AEC because of the instability of IIR (Adapa & Bollu, 2013; Lo, 2009; Lu, 2007; Pushpalatha & Kumar, 2014).

The main function of the adaptive filter of the AEC system is to estimate the echo path of the room in order to obtain a signal similar to that of the echo signal. For echo path estimation, an adaptive update is needed to identify any environmental changes, such as the movement of people. Convergence speed is crucial in obtaining the best echo path estimate, as convergence speed indicates how quickly the adaptive filter of the AEC system models the room and adapts to echo path changes (Hutson, 2003). Several kinds of adaptive filters are used in AEC, including least mean square (LMS), recursive least square (RLS), affine projection algorithm (APA), and frequency domain adaptive filter (FDAF) (Lu, 2007; Pushpalatha & Kumar, 2014). The choice of adaptive filter depends on the criteria presented in Table 2.1.

Table 2.1: Adaptive filter criteria (Lo, 2009)

Criteria	**Description**
Convergence speed	Number of iterations needed

Misadjustment	Amount by which the final converged value differs from the true value
Robustness	Convergence behavior in the presence of noise
Computational requirements	Complexity, number of operations needed
Structure	How information flows in the adaptive filter, useful for hardware implementations
Numerical properties	Stability and accuracy

In general, the goal of using the adaptive filter is to adjust the coefficients of the adaptive filter, w[n] (see Figure 2.1). It is also used to minimize the error $e[n]$ in the mean squared sense by keeping feeding $e[n]$ back to the adaptive filter, where $e[n] = d[n] - y[n]$, $d[n]$ represents the system output signal (desired signal) and $y[n]$ is the output from the adaptive filter. The adaptive filter essentially identifies a vector of coefficients w[n] that minimizes the following quadratic equation:

$$\xi[n] = E\{|\ e[n]\ |^2\} \tag{2.1}$$

where $E\{\cdot\}$ denotes the expected value and $\xi[n]$ is the mean squared error (Goldfinger, 2005; Lo, 2009).

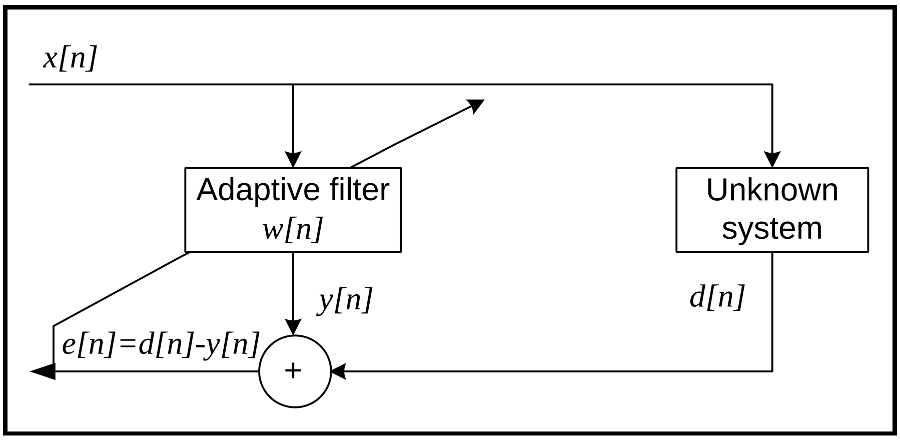

Figure 2.1: Adaptive filter adjusts filter coefficients w[n] (Lo, 2009)

Adaptive filter algorithms work in four steps: filtering, computing the error, calculating the coefficient updates, and updating the coefficients. Adaptive filter differ in how they perform the coefficient update calculation step (Lo, 2009; McLoughlin, 2009; Pushpalatha & Kumar, 2014).

2.1.1 Least Mean Square (LMS)

The LMS is the simplest and most widely used FIR adaptive filtering. Its computational complexity is low (*O(L)*, where *L* is the length of the adaptive filter), and it is suitable for most applications. Table 2.2 summarizes the LMS process (Haykin, Widrow, & Wiley, 2003; Lo, 2009; Pushpalatha & Kumar, 2014).

Table 2.2: Least Mean Square

Operation	**Computation**
Filtering	$y[n] = \mathrm{w}^T[n]\mathrm{x}[n]$
Compute Error	$e[n] = d[n] - y[n]$
Update Calculation	$\mathrm{w\hat{}}[n] = \mu e[n]\mathrm{x}[n]$
Coefficient Update	$\mathrm{w}[n] = \mathrm{w}[n-1] + \mathrm{w\hat{}}[n]$

Here, w[n] and x[n] are vectors of the size L of the coefficients and input samples, respectively, T denotes transpose of a vector or a matrix (Huang & Benesty, 2004), and μ is the step size that controls the speed of adaptation. One of the important disadvantages of the LMS that make it unusable in the AEC systems is including the sensitivity of the adaptation to the power of the input varies with time and a gradient noise amplification problem, so the step size between two adjacent filter coefficients will vary as well, and that lead also to change of the convergence speed. The convergence speed will slow down with small signals, and for the loud ones the over-shoot error would increase (Lee, Gan, & Kuo, 2009; Lo, 2009).

2.1.2 Normalized Least Mean Square (NLMS)

The NLMS solves the sensitivity of the LMS to the inputs power. The idea is to continuously adjust the step size parameter with the input power. Therefore, the step size is normalized by the current input power, and the complexity is as low as that of the LMS $O(L)$. The NLMS process is summarized in Table 2.3 (Goldfinger, 2005; Lee et al., 2009; Lo, 2009; Pushpalatha & Kumar, 2014; Shah, Lewis, Grant, & Angrignon, 2013; Ted S Wada & Juang, 2009).

Table 2.3: Normalized Least Mean Square

Operation	**Computation**
Filtering	$y[n] = \mathrm{w}^T[n]\mathrm{x}[n]$
Compute Error	$e[n] = d[n] - y[n]$
Update Calculation	$\hat{\mathrm{w}}[n] = \mu e[n]\frac{\mathrm{x}[n]}{\|\|\mathrm{x}[n]\|\|^2}$
Coefficient Update	$\mathrm{w}[n] = \mathrm{w}[n-1] + \hat{\mathrm{w}}[n]$

2.1.3 Recursive Least Squares (RLS)

Compared with the LMS and NLMS, the RLS has the fastest convergence speed because it is dependent on the input signals themselves instead of the statistics of the signals. Although the RLS converges very rapidly, its computational complexity is very high $O(L^2)$. Thus, it is too expensive for many applications because it requires long filter lengths and that mean it need more memory and more calculation process. Table 2.4 summarizes the RLS process (Farhang-Boroujeny, 1998; Goldfinger, 2005;

Lee et al., 2009; Lo, 2009; Munjal, Aggarwal, & Singh, 2008; Pushpalatha & Kumar, 2014).

Table 2.4: Recursive Least Squares

Operation	**Computation**
Filtering	$y[n] = \mathrm{w}^T[n]\mathrm{x}[n]$
Compute Error	$e[n] = d[n] - y[n]$
Update Calculation	$\hat{\mathrm{w}}[n] = k[n]\,\alpha[n]$
Coefficient Update	$\mathrm{w}[n] = \mathrm{w}[n-1] + \hat{\mathrm{w}}[n]$

The coefficient update of the RLS comes from the fallowing equations:

$$\alpha[n] = d[n] - \mathrm{w}^T[n\text{-}1]\mathrm{x}\,[n] \tag{2.2}$$

$$k[n] = \frac{\pi[n]}{\mathrm{P}[n-1]} \tag{2.3}$$

$$\pi[n] = \mathrm{x}^T[n]\mathrm{P}[n\text{-}1] \tag{2.4}$$

$$\mathrm{P}[n] = \frac{1}{\lambda}\,\mathrm{P}[n\text{-}1] - \frac{1}{\lambda}k[n]\,\mathrm{x}^T[n]\mathrm{P}[n\text{-}1] \tag{2.5}$$

In these equations, P is the filter order (Borisagar & Kulkarni, 2010; Vaseghi, 2009). λ is the forgetting factor, which increases the weight of new data and enhances the filter adaptability to non-stationary signals. The adaptive filter can respond quickly to the characteristics of the changes in the process of input when $0 \leq \lambda \leq 1$ (Paleologu, Benesty, & Ciochina, 2008).

2.1.4 Affine Projection Algorithm (APA)

The APA was proposed to generalize the NLMS and to offer a faster convergence rate for correlated signals. The computational complexity of the APA falls between those of the NLMS and RLS [$2Lp + O(p^2)$]. The APA process is summarized in Table 2.5 (Douglas, 1995; Goldfinger, 2005; Lee et al., 2009; Lo, 2009).

Table 2.5: Affine Projection Algorithm

Operation	**Computation**
Filtering	$y[n] = \mathrm{w}[n]\mathrm{X}^T[n]$
Compute Error	$e[n] = d[n] - y[n]$
Update Calculation	$\hat{\mathrm{w}}[n] = \mu e[n]\mathrm{X}[n](\mathrm{X}^T[n]\,\mathrm{X}[n]+\delta\mathrm{I})^{-1}$
Coefficient Update	$\mathrm{w}[n] = \mathrm{w}[n-1] + \hat{\mathrm{w}}[n]$

Here, X[n] is a $L \times p$ matrix containing the input samples x[n], p and L are the projection order and adaptive filter length, respectively, and δ is the regularization variable for stability purposes and is typically very small (Lo, 2009).

2.1.5 Frequency Domain Adaptive Filter (FDAF)

The FDAF (or Fast-LMS) was proposed to use the frequency domain to deal with the signals instead of keeping it in the time domain (Shynk, 1992). The frequency domain can also use discrete transforms, which reduces the processing required in signal processing applications. The Fast Fourier Transform (FFT) and the Inverse Fast Fourier Transform (IFFT) are used because they are able to adequately represent the signal data, even for short input data strings, and components will not be distorted when transmitted over linear systems (Ferrara, 1980; Gunale, Motade, Nalbalwar, & Deosarkar, 2010; Lee et al., 2009; Rao & Farhang-Boroujeny, 2009).

Figure 2.2 illustrates implementation of the Fast-LMS filter, where the input signals x(n) (far-end signal) and d(n) (near-end signal) are transformed in the frequency domain using FFT. The signals are processed using the block format; x(n) and d(n) are sequenced into blocks of length m, where m = 2n. Thus,

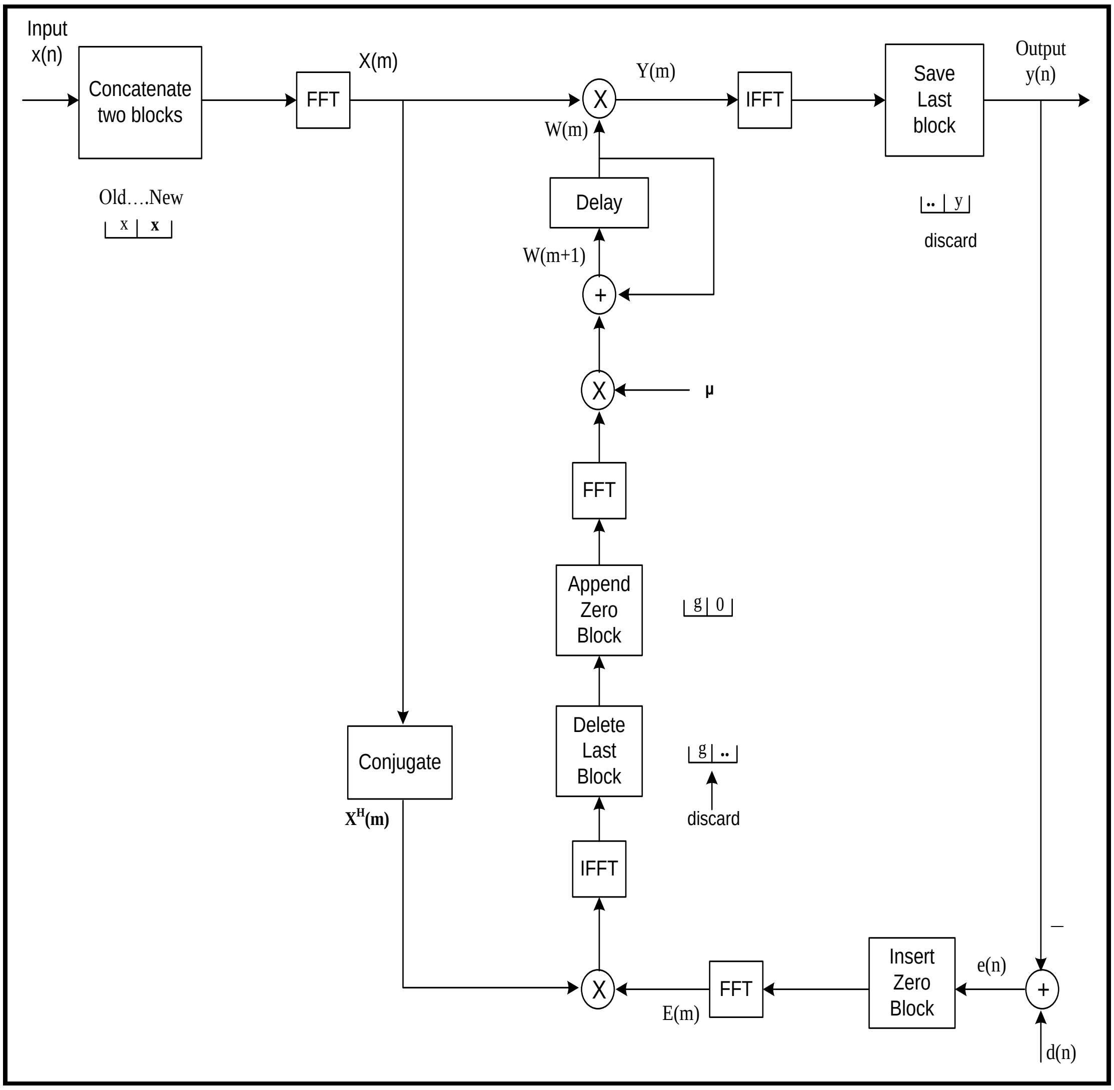

Figure 2.2: Block diagram of the FDAF (Gunale et al., 2010; Shynk, 1992)

- An input block of size m is taken from the input array, X; the FFT of this block is calculated:

$$X(m) = FFT\{x_{old}(n), x_{new}(n)\} \tag{2.6}$$

- Filter output can be computed by multiplying the FFT of the input block, X(m), by the updated filter coefficients:

$$Y(m) = X(m)W(m) \tag{2.7}$$

- The output is transformed into the time domain by computing the IFFT of the above result. The first half of this result is discarded due to circular convolution and the second half represents the output of the adaptive filter:

$$y(n) = \text{second half } (IFFT\{Y(m)\}) \tag{2.8}$$

- The error signal is calculated by the difference between the desired and the actual response:

$$e(n) \ = \ d(n) - y(n) \tag{2.9}$$

- The error signal has to be transformed into the frequency domain and needs to be 2n in size. Thus, it can be calculated by adding n zeros to the start of e(n) and performing FFT:

$$E(m) = FFT\{zeros, e(n)\} \tag{2.10}$$

- The conjugate of X(m) is calculated X^H(m). X^H(m) is multiplied by E(m) then process by IFFT to get the result. The second half of the IFFT result is discarded:

$$g(n) = \text{first half}(\ IFFT\{E(m)X^H(m)\}) \tag{2.11}$$

- Subsequently, n zeros are added to the end of g(n). The m point FFT of the resulting sequence is calculated and multiplied by μ (the step size parameter):

$$W^{\wedge}(m) = \mu \bullet FFT\{g(n)\} \tag{2.12}$$

- This filter coefficient update factor, $W^{\wedge}$(m), is added to W(m):

$$W(m+1) = W(m) + W^{\wedge}(m) \tag{2.13}$$

- The updated W(m + 1) is used as filter coefficient for the next block of input.

The Fast-LMS process is summarized in Table 2.6.

Table 2.6: Frequency Domain Adaptive Filter

Operation	**Computation**
Filtering	$y(n) = IFFT\{X(m)\, W(m)\}$
Compute Error	$e(n) = d(n) - y(n)$
Update Calculation	$W^{\wedge}(m) = \mu\, FFT\{g(n)\}$
Coefficient Update	$W(m+1) = W(m) + W^{\wedge}(m)$

2.2 Sampling Rate Conversion (SRC)

The AEC system can be affected by sampling rate mismatch, due to media played during the VoIP session or to different types of sound cards, many types of sample rate conversion (SRC) can be used to solve the mismatch in the sampling rate and help the AEC system to work without any problems. The SRC operation takes one audio signal with a specific sampling rate and changes it to another sampling rate. It converts a continuous time signal $x(t)$ into a discrete time signal $x[k]$ by taking repeated measurements defined by a fixed interval to obtain a specific time. The interval is called T_s. The sampling rate is $F_s = 1/T_s$, as shown in Figure 2.3 (Franz, 2001; Kappeler & Grünert, 2004; LaValley, 2004; Lehtinen & Renfors, 2009).

$$x[k] = x(kT_s) \tag{2.14}$$

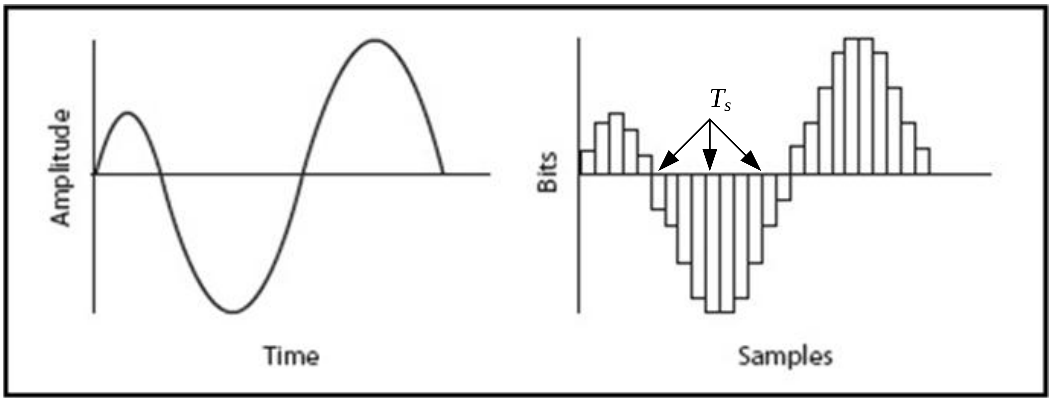

Figure 2.3: Conversion of a continuous time signal *x(t)* into a discrete time signal *x[k]* (Franz, 2001)

SRC is used to manage the spectrum of the signal. It can be calculated by using the discrete Fourier transform and limiting the frequency spectrum of the sampled signal to half the sampling rate (Nyquist frequency, which is $F_n = F_s/2$) (Parker & ScienceDirect, 2010; Redmon, 2007; Tao, Deng, Zhang, & Wang, 2008).

$$\left(e^{i2\pi F_s}\right) = a_0 + \sum_{k\in Z}^{\infty} x[k]e^{i2\pi F_s} \tag{2.15}$$

SRC can be used in two ways: for rational factors or for arbitrary ratios.

2.2.1 SRC for Rational Factors

SRC for rational factors can be performed by downsampling (decimation by factor *M*), upsampling (interpolation by factor *L*), or resampling *L/M* (Kappeler & Grünert, 2004; Rothacher, 1995; Wang, 2008) :

2.2.1.1 Decimation

Downsampling (decimation) decreases the samples in an audio signal. Both the frequency domain representation of the signal and the time domain must be considered. Figure 2.4 shows the old sampling rate and the new sampling rate after decimation by 2 (decreasing the samples by 2) (Parker & ScienceDirect, 2010).

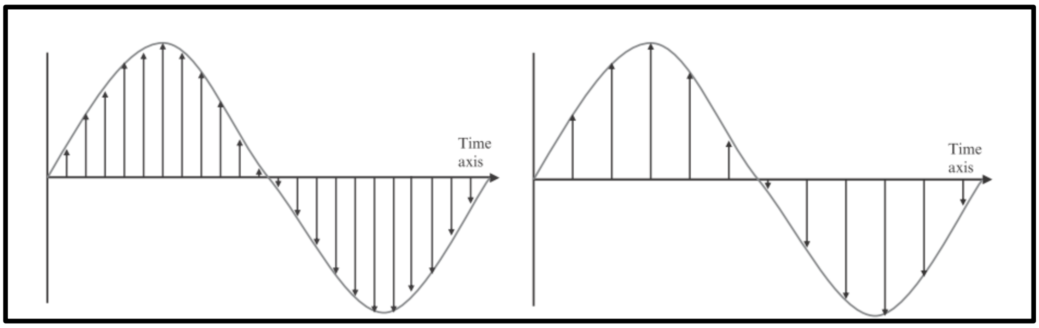

Figure 2.4: Decimation of the digital signal by 2 (Parker & ScienceDirect, 2010)

The sample is taken from an audio signal with sampling rate Fs and converted into the new signal by removing half of the signal samples. The signal itself does not change; only the sampling rate frequency and corresponding Nyquist rate frequency change. If the new Nyquist frequency is larger than the signal frequency, aliasing does not occur (aliasing is the phenomenon of sinusoids changing frequency during sampling) (Smith, 1997).

Decimation is simpler than interpolation because it does not require finding the unknown position of the new inserted samples. Instead, the digital signal will just drop the number of samples that need to be decimated (Figure 2.5) using the following calculation:

$$y_k = \begin{cases} y_1 \,, & k = 1 \\ y_{kM+1} \,, & k = 2.. \end{cases} \tag{2.16}$$

where y_k is the new sample point, k is the number of samples for the new signal after decimation, and M represents the number of decimated points.

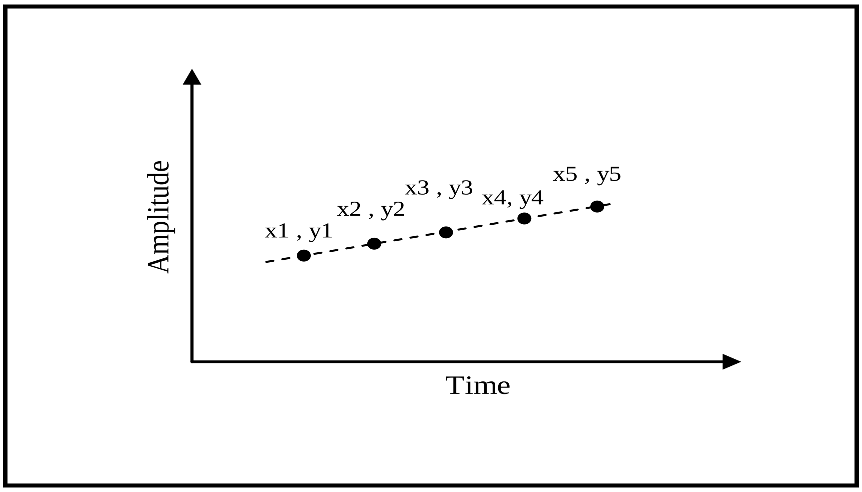

Figure 2.5: Illustration of decimation

For example, say x,y is (1,1), (2,2), (3,3), (4,4), (5,5) and M = 4. The result of decimation will keep the first point and remove the points M-1 that located after the first point that kept (i.e., {(2,2), (3,3), (4,4)}), and the final points will be {(1,1), (5,5)}.

2.2.1.2 Linear interpolation

Unlike decimation, **upsampling** (interpolation**)** increases the samples in the audio signal. Figure 2.6 shows an example of an old sampling rate and the new sampling rate after interpolation by 2 (increasing the samples by 2) (Parker & ScienceDirect, 2010).

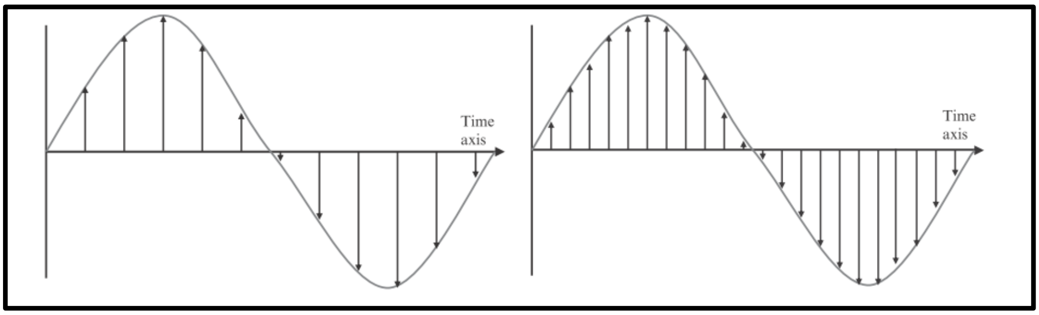

Figure 2.6: Interpolation of the digital signal by 2 (Parker & ScienceDirect, 2010)

The sample is taken from the audio signal with sampling rate F_s and converted into the new signal by doubling the samples in the signal. The signal itself does not change. Only the sampling rate frequency and corresponding Nyquist rate frequency change. As long as the new Nyquist frequency is larger than the signal frequency, aliasing does not occur. Many interpolation methods are used with digital signals, including the following:

- Linear interpolation is the simplest interpolation method that can be used for signal resampling (Babaeizadeh, 2003; Crochiere & Rabiner, 1981; Jiří Schimmel, 1999; Meijering, 2002). With this method, interpolation points (L) are placed in a straight line between two known points (xa,ya) and (xb,yb), as shown in Figure 2.7.

$$y = ya + (x - xa)\frac{yb - ya}{xb - xa} \tag{2.17}$$

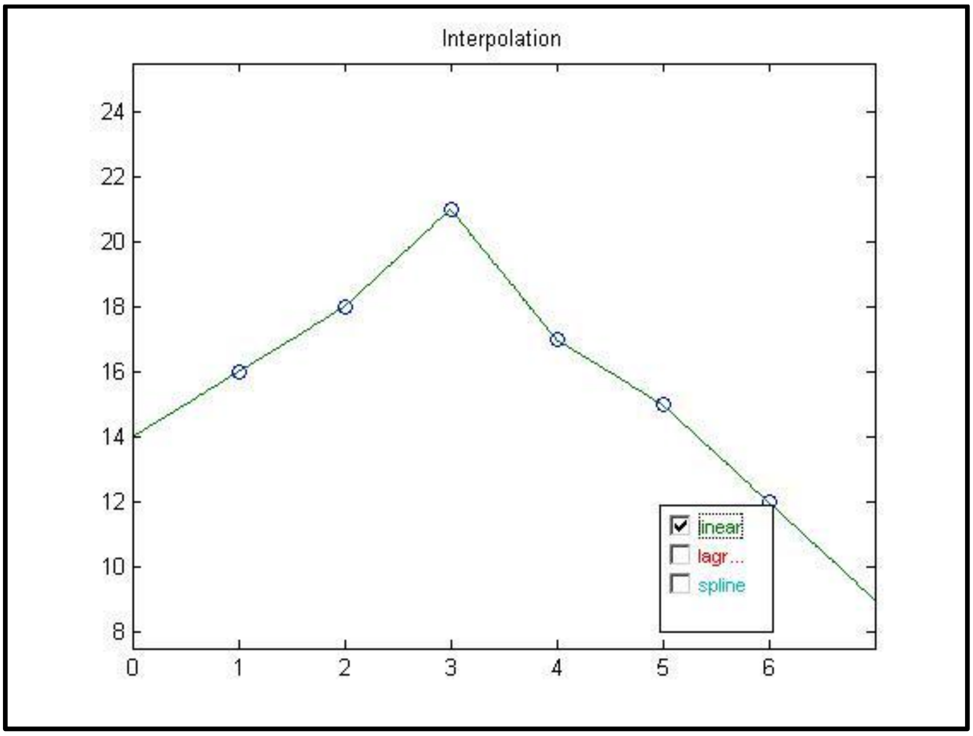

Figure 2.7: Linear interpolation

Interpolation of a digital signal is used to increase the number of samples. To find the point (x,y), which is located between (xa,ya) and (xb,yb) shown in Figure 2.8, using linear interpolation, the equation 2.17 is used:

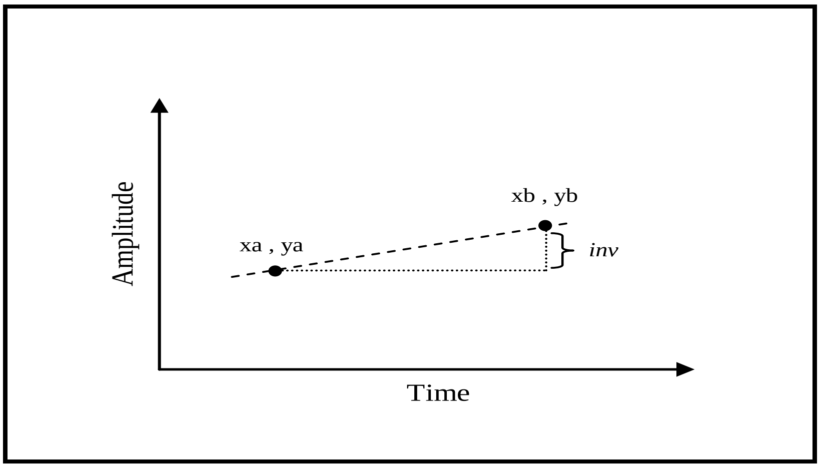

Figure 2.8: Linear Interpolation between two points