國立交通大學

工學院聲音與音樂創意科技碩士學位學程

碩士論文

DSP Implementation of Audio Effects for a Singing Robot

唱歌機器人音效之DSP實現

指導教授：白明憲

研 究 生：張濬閣

中華民國99年6月

唱歌機器人音效之DSP實現

DSP Implementation of Audio effects for Singing Robot

研究生： 張濬閣　　Student ：Chun- Ge Chang

指導教授 ： 白明憲　　Advisor ：Ming-sian Bai

國立交通大學

工學院聲音與音樂創意科技碩士學位學程

碩士論文

A thesis
Submitted to Master Program of Sound and Music Innovative Technologies
Collage of Engineering
National Chiao Tung University
In Partial Fulfillment of Requirements
for the Degree of Master of Science
in
Master Program of Sound and Music Innovative Technologies
June 2010
HsinChu, Taiwan, Republic of China

中華民國九十九年六月

唱歌機器人音效之DSP實現

研 究 生：張濬閣　　　　　　　　　　　　指導教授：白明憲

國立交通大學工學院聲音與音樂創意科技碩士學位學程

摘要

在卡拉ok機與多媒體系統中，音效往往伴演著舉足輕重的角色；現今在許多3C產品上，亦可看見許多音效的應用，例如卡拉ok機中的迴響(Echo)，電話及電視媒體常使用的變聲器(Pitch-shifter)等等。早期的各種音效主要是利用訊號延遲的原理去產生，而後隨著各種處理器運算速度的提升，現今的音效不但要求娛樂性高，擬真性好，同時也需要運算量少這方面的考量。在本論文中，主要在探討目前卡拉ok娛樂系統上常看見的各種音樂特殊效果，從物理原理及數學方程式上出發，將之實現至CPU與一特定DSP內，並試著在運算量上最佳化，成為較具新意的各種音效演算法。此外，將這些音效實現在娛樂型機器人上。

DSP Implementation of Audio effects for a Singing Robot

Student: Chun-Ko Chang **Advisor: Mingsian Bai**

Master Program of Sound and Music Innovative Technologies

National Chiao-Tung University

1001 Ta-Hsueh Road, Hsin-Chu 30050

Taiwan, Republic of China

Abstract

Audio effects, such as delay, echo, flanging, chorusing, pitch shifting, vibrato, tremolo and etc…, and virtual bass are indispensable in music productions and performance. Some are also available for home entertainments and car audio systems. Most of these effects are implemented using digital signal processors, which may reside in separate modules or may be built into keyboard workstations and tone generators. In this thesis, DSP implementation of various digital audio algorithms will be demonstrated, with theoretical equations as well as actual programming implementations shown wherever possible. These include basic audio signal manipulation, filter techniques, waveform synthesis techniques, digital audio effects theories and so on. Finally, these audio effects will implement on a singing robot.

誌謝

時光飛逝，短短兩年的研究生生涯轉眼就過去了。首先感謝指導教授白明憲博士的諄諄指導與教誨，使我順利完成學業與論文，在此致上最誠摯的謝意。而老師指導學生時豐富的專業知識，嚴謹的治學態度以及待人處事方面，亦是身為學生的我學習與景仰的典範。

此外也感謝本學程的鄭泗東教授及冀泰石教授在百忙中撥冗閱讀，並依據他們的經驗提出了寶貴的意見與指導，使得本文內容更完善與充實，在此也致上學生最無限的感激。

回顧這兩年的日子，承蒙同實驗室的博士班、林家鴻學長、嚴坤龍學長、蔡耀坤學長、曾瑞宏學長、李雨容學姊以及大我一屆的碩班的艾學安學長、何克男學長、劉冠良學長、王俊仁學長與郭育志學長們在研究與學業上的適時指點，並有幸與廖國志、曾智文、桂振益、陳俊宏，劉櫻婷及廖士涵同學互相切磋討論，每在烏雲蔽空之時，得以撥雲見日，獲益甚多。此外學弟衛帝安、許書豪、許偉智、王俊凱、吳俊慶、及馬瑞彬在生活上的朝夕相處與砥礪磨練，都是我得以完成研究的一大助因，在此獻上我最由衷地感激。

能有此刻，我也要感謝所有在精神上給我鼓舞支持的人，謝謝你們在我失落及沮喪時給我鼓勵。最後僅以此篇論文，獻給我摯愛的雙親張直樹先生、劉月香女士，姐姐佳音與绮婇。今天我能順利取得碩士學位，要感謝的人很多，上述名單恐有疏漏，在此也一一致上我最深的謝意。

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Introduction

Audio effects, such as delay, echo, flanging, chorusing, pitch shifting, vibrato, tremolo, and etc…, are indispensable in music production and performance. Some are also available for home and car audio systems. They are used by all individuals involved in the generation of music signals and start with special playing techniques by musicians, merge to the use of special microphone techniques and migrate to effect processors for synthesizing, recording, production and broadcasting of musical signals. Most of these effects are implemented using digital signal processors, which may reside in separate modules or may be built into keyboard workstations and tone generators.

The aim of this thesis is the study and implementation of digital audio effects. In Chapter 2, the physical and acoustical phenomena of digital audio effects will be presented at the beginning of each effect description, follow by an explanation of the signal processing techniques to achieve the effect and some musical application and the control of effect parameters. Also, DSP implementation of various digital audio algorithms will be demonstrated, with theoretical equations as well as actual programming implementations shown wherever possible. It will cover several categories of sound or audio effects and their impact on sound modifications, including basic audio signal manipulation, filter techniques, waveform synthesis techniques, digital audio effects theories and so on. In chapter 3, the result of effects simulation in offline and real-time processors. Also, a singing robot implementation of all audio effects will be demonstrated. Finally, a short discussion of these results is described in conclusions.

2. Audio Effect Algorithms

2.1. Delays and Echoes

Delays can be experienced in acoustical spaces. A sound wave reflected by a wall will be superimposed on the sound wave at the source. If the wall is far-away, such as a cliff, we will hear an echo. If the wall is close to us, we will notice the reflections through a modification of the sound color. Repeated reflections can appear between parallel boundaries. In a room, such reflections will be called flatter echo. The distance between the boundaries determines the delay that is imposed to each reflected sound wave. In a cylinder, successive reflections will develop at both ends. If the cylinder is long, we will hear a pitched tone. Equivalents of these acoustical phenomena have been implemented as signal processing units.

When a sound source is reflected from a distant, hard surface, a delayed version of the original signal is heard at a later time. Before the introduction of DSP in audio, the first delay units were created by using tape delay with multiple moving recording heads, while other units then produced the delay with analog circuitry. To recreate this reflection digitally, DSP delay effects units encode the input signal and store it digitally in a delay-line buffer until it is required at the later time where it is decoded back to analog form. The DSP can produce delays in a variety of ways. Delay units can produce stereo results and multiple-tapped delayed results [1]. Many effects processors implement a delay and use it as a basis for producing multi-tap and reverb effects. Multi-tapped signals can be panned to the right, left or mixed together to give the listener the impression of the stereo echo bouncing from one side to the other.

2.1.1. Single Reflection Delay

To create a single reflection (See Figure 1) of an input signal, the implementation shown above is represented in the following difference equation [2]:

$$y(n) = x(n) + ax(n-D) \quad , \qquad \textbf{(1)}$$

and it's transfer function is : $H(Z) = 1 + aZ^{-D}$; **(2)**

Notice that the input $x(n)$ is added to a delayed copy of the input. The signal can be attenuated by a factor that is less than 1, because reflecting surfaces, as well as air, contain a loss constant *a* due to absorption of the energy of the source wave. The delay *D* represents the total time it takes for the signal to return from a reflecting wall. *D* is created by using a delay-line buffer of a specified length in DSP memory. The frequency response results in a FIR comb filter where peaks in the frequency response occur at multiples of the fundamental frequency [2]. Comb filters result whenever a direct input signal is combined with delayed copies of the direct input.

The DSP can subtract the delay instead of adding it:

$$y(n) = x(n) - ax(n-D); \quad \textbf{(3)}$$

An example implementation for adding an input to a delayed replica is:

$$y(n) = \frac{1}{2}x(n) + \frac{1}{2}x(n-D); \quad \textbf{(4)}$$

2.1.2 Echo in FIR Filter

Multiple delayed values of an input signal can be combined easily to produce multiple reflections of the input. This can be done by having multiple taps pointing to different previous inputs stored into the delay line, or by having separate memory buffers at different sizes where input samples are stored.

The difference equation is a simple modification of the single delay case. To 5 delays of the input, the DSP processing algorithm would perform the following difference equation operation:

$$y(n) = x(n) + a_1x(n-D) + a_2x(n-2D) + a_3x(n-3D) + a_4x(n-4D) + a_5x(n-5D); \quad \textbf{(5)}$$

The structure (see Figure 2) uses 5 delay-line tap points for fetching samples. In addition, feedback can be used to take the output of the system delay and feed it back to the input.

2.2. Delay Modulation Effects

Delay-based Effects are some of the more interesting type of audio effects but are not computationally complex. The technique used is often called *Delay-Line Interpolation* [3], where the delay-line center tap is modified, usually by some low frequency waveform. Figure 3 summarize some common types of modulators used for moving the center tap of a delay-line [4].

Consider the FIR comb filter. If the delay is in the range 10 to 25 ms, we will hear a quick repetition named *slapback* or *doubling*. If the delay is greater than 50 ms we will hear an *echo*. If the time delay is short (less than 15 ms) and if this delay time is continuously varied with a low frequency such as 5 Hz, we will hear the *flanging* effect. If several copies of the input signal are delayed in the range 10 to 25 ms with small and random variations in the delay times, we will hear the *chorus* effect, which is a combination of the *vibrato* effect with direct signal [5]. These effects can also be implemented as IIR comb filters.

The general structure (Figure 4) described by J. Dattorro [3] will allow the creation of many different types of delay modulation effects. Each input sample is stored into the delay line, while the moving output tap will retrieved from a different location in the buffer rotating from the tap center. When the small delay variations are mixed with the direct sound, a *time-varying comb filter* results [2, 3].

The general delay line equation for the structure is:

$$y(n) = a_1 x(n) + a_2 x(n - d(n)) - a_f x(n - D_{fixed}); \qquad \textbf{(6)}$$

N = variable delay $d(n)$;

and, $d(n)$ rotates around tap center of delay line *D*.

As we will see, the above general structure will allow the creation of many different types of delay modulation effects. Each input sample is stored into the delay line, while the moving output tap will retrieved from a different location in the buffer rotating from the tap center. If a delay of an input signal is very small (around 10 ms), the echo mixed with the direct sound will cause certain frequencies to be enhanced or canceled (due to the comb filtering). This will cause the output frequency response to change. By varying the amount of delay time when mixing the direct and delayed signals together, the variable delay lines create some amazing sound effects such as chorusing and flanging.

2.2.1. Flanging Effect

Flanging was coined by the way it was accidentally discovered. As legend has it, a recording engineer was recording a signal onto 2 reel-to-reel tape decks and monitored from both playback heads of the 2 tape decks at the same time. While trying to simulate the ADT or doubling effect, it was discovered that small changes in the tape speed between the 2 decks created a “swooshing” jet sound. This effect was further enhanced by repeatedly leaning on the flanges of one of the tape reels slightly to slow the taped down. Thus the flanging effect was born.

It is very easy to recreate this effect using a DSP. Flanging can be implemented in a DSP by varying the input signal with a small, variable time delay at a very low frequency and adding the delayed replica with the original input signal (Figure 5). When the time delay offset is varied by rotating the delay-line center tap, the in-phase and out-of-phase frequencies as a result of the comb filtering sweep up and down the frequency spectrum. The “swooshing” jet engine effect created as a result is referred to as flanging.

By modifying the single reflection echo equation, the flanging can be implemented as follows:

$$y(n) = a_1 x(n) + a_2 x(x - d(n)) - a_f x(n - D); \qquad \textbf{(7)}$$

$d(n)$ rotates around tap center of delay line *D*, and notice that it must scale each signal by a constant to prevent overflow:

Flanging is created by periodically varying delay *d*(*n*). The variations of the delay time (or delay buffer size) can easily be controlled in the DSP using a low-frequency oscillator sine wave lookup table that calculates the variation of the delay time, and the update of the delay is determined on a sample basis or by the DSP's on-chip timer.

2.2.2. Chorus Effect

Chorusing is used to "thicken" sounds. This time delay algorithm (between 10 and 35 milliseconds) is designed to duplicate the effect that occurs when many musicians play the same instrument and same music part simultaneously. Musicians are usually synchronized with one another, but there are always slight differences in timing, volume, and pitch between each instrument playing the same musical notes. This chorus effect can be re-created digitally with a variable delay line rotating around the tap center, adding the time-varying delayed result together with the input signal.

Using this digitally recreated effect, a 6-string guitar can also be chorused to sound more like a 12-string guitar. Vocals can be thickened to sound like more than one musician is singing.

The chorus algorithm is similar to flange, using the same difference equation, except the delay time is longer. With a longer delay-line, the comb filtering is brought down to the fundamental frequency and lower order harmonics. Figure 6 shows the structure of a chorus effect simulating 3 instruments [2, 3]. To implement a chorus of 3 instruments, 2 variable delay lines can be used. Use a scaling factor of a constant to prevent overflow with fixed point math while mixing all three signals with equivalent gain.

$$y(n) = a_1 x(n) + a_2 x(n - d_1(n)) + a_3 x(n - d_2(n)) - (a_{f1} + a_{f2}) x(n - D) \;; \qquad \textbf{(8)}$$

2.2.3. Vibrato Effect

The vibrato effect duplicates vibrato in a singer's voice while sustaining a note, a musician bending a stringed instrument, or a guitarist using the guitars whammy bar. This effect is achieved by evenly modulating the pitch of the signal. The sound that is produced can vary from a slight enhancement to a more extreme variation. It is similar to a guitarist moving the whammy bar, or a violinist creating vibrato with cyclical movement of the playing hand. Some effects units offered vibrato as well as a tremolo. However, the effect is more often seen on chorus effects units [6].

The slight change in pitch can be achieved (with a modified version of the chorus effect) by varying the depth with enough modulation to produce a pitch oscillation. This is accomplished by changing the modify value of the delay-line pointer on-the-fly, and the value chosen is determined by a lookup table. This results in the interpolation/decimation of the stored samples via rotating the center tap of the delay line. The stored history of samples are thus played back at a slower, or faster rate, causing a slight change in pitch .

To obtain an even variation in the pitch modulation, the delay line is modified using a sine wavetable. Note that this is a stripped down of the chorus effect, in that the direct signal is not mixed with the delay-line output. This effect is often confused with tremolo, where the amplitude is varied by a LFO waveform. The tremolo and vibrato can both be combined together with a time-varying LPF to produce the effect produced by a rotating speaker (commonly referred to a 'Leslie' Rotating Speaker Emulation). The figure 7 shows the implementation of the vibrato effect.

2.3. Amplitude-Based Audio Effects

2.3.1. Tremolo Effect

Tremolo consists of panning the output result between the left and right output stereo channels at a slow periodic rate. This is achieved by allowing the output panning to vary in time periodically with a low frequency sinusoid. This example pans the output to the left speaker for positive sine values and pans the output to the right speaker for negative sine values (Figure 8). The analog version of this effect was used frequently on guitar and keyboard amplifiers manufactured in the '70s. A mono version of this effect can be done easily by modifying the code to place the tremolo result to both speakers instead of periodically panning the result. The I/O difference equation is as follows [10]:

$$y(n) = x(n) * \sin\left(2\pi f_{cycle} t\right) \; ; \qquad \textbf{(9)}$$

$$f_{cycle} = f \,/\, (sampling\ rate).$$

2.3.2. Rotary Speaker Effect

The rotary speaker effect was first used for the electronic reproduction of organ instrument. A combination of modulation and delay line can be used for a rotary speaker effect simulation, as shown in figure 9. The simulation makes use of a modulated (or fixed) delay line and amplitude modulation for intensity modifications [11]. A directional sound characteristic similar to rotate speakers can be achieved by amplitude modulating the output signal of the delay lines. A stereo rotary speaker effect is perceived due to unequal mixing of the two delay lines to the left and right channel output. The directional characteristic of the opposite horn arrangement performs an intensity variation in the listener’s ear.

2.3.3. Wah-wah Effect

The wah-wah effect was first used for electronic guitar. It produced mostly by foot-controller signal processors containing a bandpass filter with variable center/resonant frequency and a small bandwidth. Moving the pedal back and forth changes the bandpass cut-off/center frequency. The wah-wah effect is then mixed with the direct signal as shown in Figure 10. This effect leads to a spectrum shaping similar to speech and produces a speech like “wah-wah” sound. If the variation of the center frequency is controlled by the input signal, a low frequency oscillator is used to change the center frequency. Such an effect is call an auto-wah filter.

2.4. Phase Vocoder Basics

A very interesting (and intuitive) way of modifying a sound is to make a two-dimensional representation of it, modify this representation in some or other way and reconstruct a new signal from this representation. Consequently a digital audio effect based on time-frequency representations requires three steps: an analysis (sound to representation), a transformation (of the representation) and a re-synthesis (getting back to a sound). The analysis/synthesis scheme is termed the *phase vocoder*. That means the input signal $x(n)$ is multiplied by a sliding window of finite length N, which yields successive windowed signal segments. These are transformed to the spectral domain by FFTs. In this way, a time-varying spectrum $X(n,k) = |X(n,k)|e^{j\varphi(n,k)}$ with $k = 0,1,\ldots\ldots\ldots, N - 1$ is computed for each windowed segment. The short-time spectra can be modified or transformed for a digital audio effect. Then each modified spectrum is applied to an IFFT and windowed in the time domain. The windowed output segments are then overlapped and added yielding the output signal [12].

The short-time Fourier transform (STFT) of the signal $x(n)$ is given by:

$$X(n,k) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)W_N^{mk} \quad , k = 0,\ 1,\ \ldots .,\ N\text{-}1\ , \quad W_N = e^{-2\pi/N} \quad , \qquad \mathbf{(10)}$$

$$= X_R(n,k) + jX_I(n,k) = |X(n,k)|e^{j\varphi(n,k)}. \qquad \textbf{(11)}$$

$X(n,k)$ is a complex number and represents the magnitude $|X(n,k)|$ and phase $\varphi(n,k)$ of a time-varying spectrum with the frequency index $0 \le k \le N-1$ and time index n. The summation index is m. At each time index n the signal $x(m)$ is weighted by a finite length window $h(n\text{-}m)$. Thus the computation of (10) can be performed by a finite sum over m with an FFT of length N.

2.4.1. Filter Bank Summation Model

The computation of the time-varying spectrum of an input signal can also be interpreted as a parallel bank of N bandpass filters, as shown in Figure 11, with impulse responses and Fourier transform given by

$$h_k(n) = h(n)W_N^{-nk}, k = 0,1,.....N-1, \qquad \textbf{(12)}$$

$$H_k(e^{j\Omega}) = H(e^{j(\Omega-\Omega_k)}), \Omega_k = \frac{2\pi}{N}k \qquad \textbf{(13)}$$

Each bandpass signal $y_k(n)$ is obtained by filtering the input signal $x(n)$ with the corresponding bandpass filter $h_k(n)$. Since the bandpass filters are complex-valued, we get complex-valued output signals $y_k(n)$, which will be denoted by

$$y_k(n) = \tilde{X}(n,k) = |X(n,k)| \cdot e^{j\tilde{\varphi}(n,k)} \qquad \textbf{(14)}$$

These filter operations are performed by the convolutions

$$\begin{aligned} y_k(n) &= \sum_{m=-\infty}^{\infty} x(m)h_k(n-m) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)W_N^{-(n-m)k} \\ &= W_N^{-nk} \sum_{m=-\infty}^{\infty} x(m)W_N^{mk}h(n-m) = W_N^{-nk}X(n,k) \end{aligned} \qquad \textbf{(15)}$$

From (15) and (16) it is important to notice that

$$\tilde{X}(n,k) = W_N^{-nk}X(n,k) = W_N^{-nk}|X(n,k)|e^{j\varphi(n,k)} \qquad \textbf{(16)}$$

$$\tilde{\varphi}(n,k) = \frac{2\pi k}{N}n + \varphi(n,k) \qquad \textbf{(17)}$$

Based on equation (15) and two different implementations are possible, as shown in figure 11. The first implementation is the so-called complex baseband implementation. The baseband signals $X(n,k)$ (short-time Fourier transform) are computed by modulation of $x(n)$ with W_N^{nk} and lowpass filtering for each channel k. The modulation of $X(n,k)$ by W_N^{-nk} yields the bandpass implementation, which filters the input signal with $h_k(n)$ given by (12), as shown in the lower part of figure 11. This implementation leads directly to the complex-valued bandpass signals $\tilde{X}(n,k)$. If the equivalent baseband signals $X(n,k)$ are necessary, they can be computed by multiplication with W_N^{nk}. The operations for the modulation by W_N^{nk} yielding $X(n,k)$ and back modulation by W_N^{-nk} (lower part in figure 11) are only shown to point out the equivalence of both implementations.

The output sequence $y(n)$ is the sum of the bandpass signals according to

$$y(n) = \sum_{k=0}^{N-1} y_k(n) = \sum_{k=0}^{N-1} \tilde{X}(n,k) = \sum_{k=0}^{N-1} X(n,k) W_N^{-nk} \qquad \textbf{(18)}$$

The output signals $y_k(n)$ are complex-valued sequences $\tilde{X}(n,k)$. For a real-valued input signal $x(n)$ the bandpass signals satisfy the property $y_k(n) = \tilde{X}(n,k) = \tilde{X}*(n,N-k) = y_{N-k}^*(n)$. For a channel stacking with $\Omega_k = \dfrac{2\pi}{N}k$ we get the frequency bands shown in the upper part of figure 11. The property $\tilde{X}(n,k) = \tilde{X}*(n,N-k)$ together with the channel stacking can be used for the formulation of real-valued bandpass signals (real-valued kth channel)

$$\begin{aligned} \hat{y}_k(n) &= \tilde{X}(n,k) + \tilde{X}(n,N-k) = \tilde{X}(n,k) + \tilde{X}*(n,k) \\ &= |X(n,k)| \cdot \left[e^{j\tilde{\varphi}(n,k)} + e^{-j\tilde{\varphi}(n,k)}\right] \\ &= 2|X(n,k)| \cdot \cos[\tilde{\varphi}(n,k)] \end{aligned}$$

for $k = 1,\ldots,N/2-1$. **(19)**

This leads to

$$\hat{y}_0(n) = y_0(n), k = 0, \quad \text{dc channel.}$$

$$\hat{y}_k(n) = 2|X(n,k)| \cdot \cos[\Omega_k n + \varphi(n,k)], k = 1,....N/2-1,$$

bandpass channel.

$$\hat{y}_{N/2}(n) = y_{N/2}(n), k = N/2, \quad \text{highpass channel.} \qquad \textbf{(20)}$$

Besides a dc and a highpass channel we have *N*/2-1 cosine signals with fixed frequencies Ω_k and time-varying amplitude and phase. This means that we can add real-valued output signals $\hat{y}_k(n)$ to yield the output signal

$$y(n) = \sum_{k=0}^{N/2} \hat{y}_k(n) \qquad \textbf{(21)}$$

This interpretation offers analysis of a signal by a filter bank, modification of the short-time spectrum $\tilde{X}(n,k)$ on a sample-by-sample basis and synthesis by a summation of the bandpass signals $y_k(n)$[13]**.** Due to the fact that the baseband signals are bandlimited by the lowpass filter *h*(*n*), a sampling rate reduction can be performed in each channel to yield $X(sR,k)$**,** where only every *R*th sample is taken and *s* denotes the new time index. This leads to a short-time transform $X(sR,k)$ with a hop size of *R* samples. So the analysis algorithm is given by

$$X(sR_a,k) = \sum_{m=-\infty}^{\infty} x(m)h(sR_a - m)W_N^{mk}$$

$$= W_N^{sR_a k} \sum_{m=-\infty}^{\infty} x(m)h(sR_a - m)W_N^{-(sR_a - m)k}$$

$$= W_N^{sR_a k} \cdot \tilde{X}(sR_a,k) = X_R(sR_a,k) + jX_I(sR_a,k)$$

$$= |X(sR_a,k)| \cdot e^{j\varphi(sR_a,k)}, \qquad \text{k=0,1,….,}N\text{-1} \qquad \textbf{(22)}$$

It means that the time index is now *n*=sR_a, where R_a denotes the analysis hop size. The analysis window is denoted by *h*(*n*). Spectral modifications in the time-frequency plane can now be done which yields $Y(sR_s,k)$**,** where R_s is the synthesis hop size. The synthesis algorithm is given by

$$y(n)=\sum_{s=-\infty}^{\infty} f(n-sR_s)y_s(n-sR_s) \tag{23}$$

with $y_s(n)=\frac{1}{N}\sum_{k=0}^{N-1}\left[W_N^{-sR_sk}Y(sR_s,k)\right]\cdot W_N^{-nk}$,

where $f(n)$ denotes the synthesis window. Finite length signals $y_s(n)$ are derived from inverse transforms of short-time spectra $Y(sR_s,k)$. These short-time segments are weighted by the synthesis window $f(n)$ and then added by the overlap-add procedure.

2.4.2. Filter Bank Approach

From a musician's point of view the idea behind this technique is to represent a sound signal as a sum of sinusoids. Each of these sinusoids is modulated in amplitude and frequency. These sinusoids represent filtered versions of the original signal. The manipulation of the amplitudes and frequencies of these individual signals will produce a digital effect including pitch shifting or time stretching.

One can use a filter bank to split the audio signal into several filtered versions. The sum of these filtered versions reproduces the original signal. For a perfect reconstruction the sum of the filter frequency responses should be unity. In order to produce a digital audio effect, one needs to alter the intermediate signals that are analytical signals consisting of real and imaginary parts. The implementation of each filter can be performed by a heterodyne filter, as shown in figure 12.

The implementation of a stage of a heterodyne filter consists of a complex-valued oscillator with a fixed frequency Ω_k, a multiplier and an FIR filter. The multiplication shifts the spectrum of the sound, and the FIR filter limits the width of the frequency shifted spectrum. This heterodyne filtering can be used to obtain intermediate analytic signals, which can be put in the form

$$X(n,k)=\left[x(n)\cdot e^{-j\Omega_k n}\right]*h(n)=X_R(n,k)+jX_I(n,k)=|X(n,k)|e^{j\varphi(n,k)}$$

$$=\left|X(n,k)\right|\cos\left(\varphi(n,k)\right)+\left|X(n,k)\right|\sin\left(\varphi(n,k)\right) \qquad \textbf{(24)}$$

The difference from classical bandpass filtering is that here the output signal is located in the base band. This representation leads to a slowly varying phase $\varphi(n,k)$ and the derivation of the phase is a measure of the frequency deviation from the center frequency Ω_k. A sinusoid $x(n)=\cos[\Omega_k n+\varphi_0]$ with frequency Ω_k can be written as $x(n)=\cos[\tilde{\varphi}(n)]$. The derivation of $\tilde{\varphi}(n)$ gives the frequency $\Omega_k=\frac{d\tilde{\varphi}(n)}{dn}$. The derivation of the phase $\tilde{\varphi}(n,k)$ at the output of a bandpass filter is termed the instantaneous frequency given by

$$\begin{aligned}\Omega_i(n,k)=w_i(n,k)T=2\pi f_i(n,k)/f_s&=\frac{d\tilde{\varphi}(n,k)}{dn}=\Omega_k+\frac{d\varphi(n,k)}{dn}\\&=\Omega_k+\varphi(n,k)-\varphi(n-1,k)\end{aligned} \qquad \textbf{(25)}$$

$$f_i(n,k)=\left(\frac{k}{N}+\frac{\varphi(n,k)-\varphi(n-1,k)}{2\pi}\right)\cdot f_s \qquad \textbf{(26)}$$

As soon as we have the instantaneous frequencies, we can build an oscillator bank and eventually change the amplitudes and frequencies of this bank to build a digital audio effect. The recalculation of the phase from a modified instantaneous frequency is done by computing the phase according to

$$\tilde{\varphi}(n,k)=\tilde{\varphi}(0,k)+\int_0^{nT}2\pi f_i(\tau,k)d\tau \qquad \textbf{(27)}$$

The result of the magnitude and phase processing can be written as $Y_k(n,k)=\left|Y(n,k)\right|e^{j\varphi_y(n,k)}$, which is then used as the magnitude and phase for the complex-valued oscillator running with frequency Ω_k. The output signal is then given by

$$\begin{aligned}\tilde{Y}(n,k)=\left|Y(n,k)\right|e^{j\varphi_y(n,k)}\cdot e^{j\Omega_k n}&=Y(n,k)\cdot e^{j\Omega_k n}\\&=\left[Y_R(n,k)+jY_I(n,k)\right]\cdot e^{j\Omega_k n}\end{aligned} \qquad \textbf{(28)}$$

The re-synthesis of the output signal can then be performed by summing all the individual back shifted signals according to

$$y(n)=\sum_{k=0}^{N-1}\tilde{Y}(n,k)=\sum_{k=0}^{N/2}A(n,k)\cos\left[\Omega_k n+\varphi_y(n,k)\right] \qquad \textbf{(29)}$$

2.5. Phase Vocoder Effects

2.5.1. Time Stretching

Time-frequency scaling is one of the most interesting and difficult tasks that can be assigned to time-frequency representations: changing the time scale independently of the “frequency content”. For example, one can change the rhythm of a song without changing its pitch, or conversely transpose a song without any time change. Time stretching is not a problem that can be stated outside of the perception: we know, for example, that a sum of two sinusoids is equivalent to a product of a carrier and a modulator.

There are two implementations for time-frequency scaling by phase vocoder. The first one uses a bank of oscillators, whose amplitudes and frequencies vary over time. If we can manage to model a sound by the sum of sinusoids, time stretching and pitch shifting can be performed by expanding the amplitude and frequency functions. The second implementation uses the sliding Fourier transform as the model for re-synthesis: if we can manage to spread the image of a sliding FFT over time and calculate new phases, then we can reconstruct a new sound with the help of inverse FFTs. Both of these techniques rely on phase interpolation, which need an unwrapping algorithm at the analysis stage, or equivalently an instantaneous frequency calculation. The time stretching algorithm mainly consists of providing a synthesis grid which is different from analysis grid, and to find a way to reconstruct a signal from the values on this grid. The classical way of using a phase vocoder for time stretching is to keep the magnitude unchanged and to modify the phase in such a way that the instantaneous frequencies are preserved.

In the FFT analysis (sum of sinusoids synthesis) approach, we calculate the instantaneous frequency for each bin and integrate the corresponding phase increment in order to reconstruct a signal as the weighted sum of cosines of the phases. However, here the hope size for the re-synthesis is different form the analysis. Therefore the following steps are necessary:

1. Calculate the phase increment per sample by $d\varphi(k) = \Delta\varphi(k)/R_a$.
2. For the output samples of the re-synthesis integrate this value according to $\tilde{\varphi}(n+1,k) = \tilde{\varphi}(n,k) + d\varphi(k)$.
3. Sum the intermediate signals which yields $y(n) = \sum_{k=0}^{N/2} A(n,k)\cos(\tilde{\varphi}(n,k))$. (see figure 13).

2.5.2. Pitch Shifting

Pitch shifting is different from frequency shifting: a frequency shift is an addition to every frequency, while pitch shifting is the multiplication of every frequency by a transposition factor. Pitch shifting can be directly linked to time stretching. Resampling a time-stretched signal with the inverse of the time stretching ratio performs pitch shifting and going back to the initial duration of the signal (see figure 14). There are, however, alternative solutions which allow the direct calculation of a pitch shifted version of a sound.

In the time stretching algorithm using the sum of sinusoids we have an evaluation of instantaneous frequencies. As a matter of fact transposing all the instantaneous frequencies can lead to an efficient pitch shifting algorithm. Therefore the following steps have to be performed:

1. Calculate the phase increment per sample by $d\varphi(k) = \Delta\varphi(k)/R_a$.
2. Multiply the phase increment by the transposition factor ***transpo*** and integrate

the modified phase increment according to

$$\tilde{\psi}(n+1,k)=\tilde{\psi}(n,k)+transpo\cdot d\varphi(k)\ .$$

3. Calculate the sum of sinusoids: when the transposition factor is greater than one, keep only frequencies under the Nyquist frequency bin N/2. This can be done by taking only the $N/(2*transpo)$ frequency bins.

2.5.3. Robotization

The effect applies a fixed pitch onto a sound. Moreover, as it forces the sound to be periodic, many erratic and random variations are converted into robotic sounds. The sliding FFT of pulses where the analysis is taken at the time of these pulses will give a zero phase value for the phase of the FFT. This is a clear indication that putting a zero phase before an IFFT re-synthesis will give a fixed pitch sound. So zeroing the phase can be viewed from two points of view:

1. The result of an IFFT is a pulse-like sound and summing such grains at regular intervals gives a fixed pitch.
2. Due to fact that the time-frequency representation now shows a succession of vertical lines with zero values in between, this will lead to a comb filter effect during re-synthesis.

3 Virtual Bass Synthesis

3.1 Bandwidth Extension(BWE)

There is increasing popularity of 3C products (computer, communication, and consumer electronics) in nowadays human life. From the marketplace, more audio and video functions are required for 3C products than ever. For instance, a third-generation handset has been designed to serve not only for communications but also for entertainments like a TV, a camera and a MP3 player. Therefore, it is significant that creating new technology of audio systems to address the needs of 3C products.

Bandwidth extension (BWE) refers to methods that increase the frequency bandwidth of signals. It is desired when the frequency content of the signal at some point should be enhanced to improve audio effects or if the bandwidth of signal has been reduced because of some economical constraints. An obvious way to categorize various BWE methods is based on the frequency range of interest (high frequency or low frequency) and where the signal bandwidth is actually extended (physical or psycho-acoustical extension) [21]. The psycho-acoustical extension, different to physical BWE, use no practical implementation to contain the frequency range of interest, but exploit the property of human hearing to achieve bandwidth extension.

For example, small loudspeakers can not radiate very low frequency components because their fundamental resonant frequencies are too high, so the psychoacoustic extension takes part in the auditory system instead of extending the actual physical bandwidth of the signal.

3.2 Nonlinear Processing

In this chapter we review some efficient nonlinear operations to extend frequency bandwidth. These algorithms are convenient ways for generating harmonics signals with odd or even harmonics. They have their own spectral characteristics and can create different kinds of audio effects.

Multiplier

Figure 15 shows an implementation structure of multiplier whose input signal is repeatedly multiplied with itself, producing a harmonic series. It has the advantage that one can control at the outset, the number of harmonics created, and their relative amplitudes.

We begin the analysis by assuming a pure tone input signal $x(t)$ of frequency f_0 at the reference level (defined to be 1):

$$x(t) = \sin(2\pi f_0 t)\,. \tag{30}$$

After multiplying a scaled x with a scaled replica, we get $x_2(t)$, as

$$x_2(t) = G_1 G_2 \sin^2(2\pi f_0 t) = \frac{G_1 G_2}{2}[1 - \cos(4\pi f_0 t)]\,. \tag{31}$$

The frequency doubling is apparent: G_1, G_2 are the scaling factors. By multiplying x_2 with another scaled replica of x, the third harmonic x_3 can be created, and so on.

The frequency doubling is apparent: G_1, G_2 are the scaling factors. By multiplying x_2 with another scaled replica of x, the third harmonic x_3 can be created, and so on.

The output signal $y(t)$, assuming that three harmonics above f_0 are generated, will become

$$y(t) = h_0 + \sum_{i=1}^{2}[h_{2i-1}\sin((2i-1)\times 2\pi f_0 t) + h_{2i}\cos(2i \times 2\pi f_0 t)]\,. \tag{32}$$

The h_i being the scale factors given as

$$h_0 = \frac{G_1 G_2}{2}[1 + \frac{1}{4}G_3 G_4]\,, \tag{33}$$

$$h_1 = G_1[1+\frac{3}{4}G_2G_3],$$
$$h_2 = -\frac{G_1G_2}{2}[1+G_3G_4],$$
$$h_3 = -\frac{G_1G_2G_3}{4},$$
$$h_4 = \frac{G_1G_2G_3G_4}{8},$$

If we specify what the amplitudes h_i should be, then we must choose the scaling factors G_i such that

$$G_1 = h_1 + 3h_3, \tag{34}$$
$$G_2 = -2\frac{h_2+4h_3}{h_1+3h_3}, h_1 \neq -3h_3,$$
$$G_3 = \frac{2h_3}{h_2+4h_4}, h_2 \neq -4h_4,$$
$$G_4 = -2\frac{h_4}{h_3}, h_3 \neq 0.$$

Figure 16 shows that a sine wave 100Hz is created the 3rd, 4th, and 5th harmonics by a multiplier, and their relative magnitudes of the 3rd, 4th, and 5th harmonics are designed equally to the fundamental frequency.

Rectifier

A very efficient method of harmonics generation is by rectification, it includes half-wave or full-wave. As a whole system is nonlinear, the resulting spectrum consists of only the even harmonics of f_0, which implies that fundamental frequency of the output signal is now 2 f_0. Perceptually, this means that the synthetic bass sounds an octave high, compared to the input signal. Another disadvantage is that a rectifier cannot control the magnitude of harmonics like a multiplier. However, the method can still be attractive mainly because of the efficient implementation. Figure 17 shows that a sine wave 100Hz in Fig. 16(a) is created only even harmonics by a half-wave rectifier. We can notice that the fundamental frequency 100Hz is eliminated.

Clipper

Another convenient way to generate a harmonics signal with only odd harmonics is by means of a clipper. The clipper output signal g_c in response to an input f is

$$g_c(t) = \begin{cases} f(t) & if \quad |f(t)| \leq l_c \\ l_c & if \quad f(t) > l_c \\ \text{-}l_c & if \quad f(t) < \text{-}l_c, \end{cases} \tag{35}$$

where l_c is the threshold. The clipper in Fig. 18 demonstrates very good subjective results in the low-frequency psychoacoustic BWE application, which will be validated using subjective listening tests in Section 2.6.6. This effect due to clipper sounds low-pitched and saturated enough, so the method is applicable to the realization of virtual bass. We can get information from Fig. 19 that only odd harmonics are created by clipper and the fundamental frequency is still preserved. The differences between clipper and rectifier are not only positions of harmonics generated but also preservation of the fundamental frequency. Another disadvantage for a clipper, like a rectifier, cannot control the magnitude of harmonics.

Hyperbolic tangent

Unlike the clipper that is a “hard” clipper, the hyperbolic tangent function shown belongs to a “soft” clipper. Fig. 20 shows the transform of a sine wave 100 Hz on the time and frequency domains by the methods of hyperbolic tangent. We can notice that the waveform modified by the hyperbolic tangent in Fig. 20(a) seems to be compressed. It uses a function that has a gain at low and moderate signal levels, but attenuation at high signal levels. It is different form ordinary compressors in that it is memoryless. That is, it is an instantaneous compressor. During experiments, it appeared that the function where $x(t)$ is the input time signal, and $y(t)$ is the modified output signal by hyperbolic tangent.

$$y(t) = c_1 \tanh(c_2 x(t)) \tag{36}$$

The constant c_1 determines the maximum output level and c_2 determines the gain at low signal levels.

3.3 Virtual Bass

The motivation of the VB synthesis is that the performance of low frequency in a general speaker is poor; we try to realize the bass frequency components as well as possible. Without using an extra sub-woofer or damaging the speaker, the unique Virtual Bass synthesis can be completed by means of generating the harmonics of the fundamental of the low frequency, which is badly performed in a common loundspeaker. Having generated the harmonics, the fundamental parts of the low frequency components as well as the generated harmonics are determined according to the Timbre Loudness Control in order to acquire an appropriate weighting for harmonics. Finally, the original signal and the Virtual Bass signal are combined together, and the merged signal can be reproduced in an ordinary speaker, for the reason that the bass signal is replaced by the Virtual Bass.

3.4 Psychoacoustic Bandwidth Extension for Low Frequencies

Because it is generally difficult to have a good low frequency loudspeaker response with small loudspeakers, it is pertinent to ask whether other options are available. One option is to use BWE, with the ‘extension’ taking part in the auditory system, instead of extending the actual physical bandwidth of the signal. This approach is to make use of the ‘missing fundamental’ effect: a special case of residue pitch, also known as virtual pitch. We can substitute an $f < f_1$ by a series $kf, k > 2$, to evoke the residue pitch of f, while the loudspeaker does not radiate energy at frequency f.

3.5 Realization of Virtual Bass

With good properties of spectral characteristics, temporal characteristics and inter-modulation distortion [22], we choose the clipper as the method of creating harmonics because of the best low-frequency psychoacoustic performance. Fig. 21 shows the whole process of VB realization. There are two paths, the first path is the main structure performing virtual bass, and another path just contains delay. In the other hand, we use clipper described in Section 3.2 for creating harmonics, and then an adjustable gain control will be used. If this procedure is not performed, the signal modified by clipper is not amplified or attenuated as a desired output. Instead of adjusting the loudness by equal loudness contour, we use timbre loudness control to obtain a suitable spectrum and timbre. It not only controls the loudness but avoids the distortion of timbre. As the equalizer is concerned, it is similar to adjust again control over the whole bandwidth. Now the work we do is the same purpose as the equalizer to design frequency curves which fit with different requirements. Each step is introduced as follows.

First, we use the white noise as the input signal, and pass it along clipper and also creates a long bandwidth of harmonics. Next, we try to design a frequency curve in frequency domain, and let the input signal be filtered off the frequency curve. After try and error, if the output sounds like the white noise in loudness and timbre, this frequency curve is the optimal design. Finally, after combination of the first and second path, a high pass filter should be design for avoiding reduction of high frequency components.

Virtual bass can also be performed on a cell phone whose loudspeaker size is much smaller than the common one, but we must redesign the range of band pass filter. The fundamental resonant frequency of a cell phone is almost 1000Hz when the fundamental resonant frequency of ordinary speakers is 200Hz~300Hz approximately. That is why I want to shift the range of band pass filter to 500Hz~1000Hz instead of 50Hz~200Hz.

3.6 Subjective Listening Test

Subjective listening tests were conducted to evaluate the performance of the aforementioned virtual bass. The loudspeaker arrangement and listening room follows the standard, ITU-R BS 1116. Multi Stimuli with Hidden Reference and Anchor (MUSHRA) of ITU-R BS 1534-1 procedure was employed as the test procedure. Four subjective indices were employed for assessing the performance of these virtual techniques. These indices are defined as follows:

(1) Total Preference: The global impression and preference.

(2) Artifact: Any extraneous disturbances to the signal.

(3) Fullness: Dominance of low-frequency sound.

(4) Brightness: Dominance of high-frequency sound.

Sixteen subjects participated in these tests and they were instructed with definitions of the subjective indices before the tests. In the listening tests, the subjects were asked to respond on a questionnaire with the subjective indices placed on the scale from -3 to +3 shown in Table 1. The results of the tests were processed by using the MANOVA. Not only the mean grades but also the significance levels were shown in the analysis for different methods. Cases with significance levels below 0.05 indicate that statistically significant difference exists among methods.

This subjective listening test is separated into three parts, but they all refer to virtual bass. The first test aims at comparing the difference of virtual bass between our methods, WinDVD and Media player using a common loundspeaker. The second test aims at comparing the difference of VB filtered by a high pass 200Hz filter. The purpose of second test is to verify the existence of psychoacoustic bandwidth extension for low frequency. The procedure of last test is the same with first test except using a cell phone instead of a ordinary speaker. In the first and second listening test, a multimedia two-channel stereo loudspeaker with approximately 120Hz cut-off frequency was employed as the rendering device, as shown in Fig.22(a) and the experimental arrangement of the last test is shown in Fig.22(b). In these tests, we define the unprocessed signal as the hidden reference, and the highpass filtered signal as the anchor.

The results of the tests were processed by using the MANOVA. Not only the mean grades but also the significance levels were shown in the analysis for different methods. Cases with significance levels below 0.05 indicate that statistically significant difference exists among methods.

It is noted that the scores obtained for the hidden reference and anchor were used for evaluating each listener's inconsistency. Therefore, it was decided to exclude from this analysis the scores obtained for these two items since this is a relatively easy task in terms of inconsistency.

Table 2 shows p-values which below 0.05 indicate that statistically significant difference exists among methods. Fig.23 shows respectively subjective grade of the first, second and third test. Figure 24 shows the unprocessed signal, and the signal modified by our approach respectively on the time-frequency domains.

According to the first and the second subjective listing tests and p-values, the effect of psychoacoustic BWE has been realized. Beside the fullness index, our approach earns more total preference than that of reference and creates no artifacts. The subjective result of third test indicates that it is practicable for small loudspeakers to use this technology.

4. Simulations and Experiments

4.1. Offline Simulation

We can use MATLAB to simulate every effect offline.

4.2. Real-Time Implementation

We use ADI-BF533 to simulate effects real-time. It has four inputs and six outputs. VisualDSP++ provides the following features.

• **Extensive editing capabilities.** Create and modify source files by using multiple language syntax highlighting, drag-and-drop, bookmarks, and other standard editing operations.

• **Flexible project management.** Specify a project definition that identifies the files, dependencies, and tools that we will use to build projects. Create this project definition once or modify it to meet changing development needs.

• **Easy access to code development tools.** Analog Devices provides these code development tools: C/C++ compiler, assembler, linker, splitter, and loader. Specify options for these tools by using dialog boxes instead of complicated command line scripts. Options that control how the tools process inputs and generate outputs have a one-to-one correspondence to command line switches. Define options for a single file or for an entire project. Define these options once or modify them as necessary.

• **Flexible project build options.** Control builds at the file or project level. VisualDSP++ enables you to build files or projects selectively, update project dependencies, or incrementally build only the files that have changed since the previous build.

• **Flexible workspace management.** Create up to ten workspaces and quickly switch between them. Assigning a different project to each workspace enables you to build and debug multiple projects in a single session.

• **Easy movement between debug and build activities.** We start the debug session and move freely between editing, build, and debug activities.

Figure 21 shows the Integrated Development and Debugging Environment.

4.3. Implementation on a Singing Robot

We use LEGO NXT robot to simulate every audio effect, the LEGO NXT robot is shown in Fig25. The experiment has two steps, the first step is command recognition that utilize to recognize voice command of the singer name and audio effect mode. Besides, we implemented all algorithms on the LabView system. After the command recognition, the robot will sing the song of the singer with audio effect.

5. Conclusions

In general, most audio algorithms fall under one of two classes: professional and consumer audio. For professional audio, the applications are targeted to a specific consumer base that consists of professional musicians, producers, audio engineers and technicians, such as electronic music keyboards, digital audio effect processors (reverb, chorus, vibrato, pitch shifting……), graphic and parametric equalizers, digital mixing/recording consoles. Consumer audio applications target a high volume customer base through consumer electronic retailers, such as home theater systems, karaoke machines, digital graphic equalizers, CD/DVD players, computer audio multimedia systems.

In the first part of this thesis, the offline and real-time (DSP) algorithms which can combine to a new audio effect module, that can be used in all two markets segments, while others are used only in the professional or consumer space.

In the second part of this thesis, we accomplish the virtual bass effect exploiting the psycho-acoustic property. According to the subjective listening tests, this technology is practicable for ordinary speakers and also micro speakers.

In the final part of this thesis, we implemented all audio effects on a LEGO NXT robot, which can recognize voice command and sing the song with audio effects, and achieved the entertainment of the robot.

References

[1] Scott Lehman, *Harmony Central Effects Explained,* Internet: Http://www.harmony-central.com/Effects/Articcles, (1996)

[2] S. J. Orfanidis, *Introduction to Signal Processing*, Chapter 8, Sec 8.2, pp. 355-383, Prentice Hall, Englewood Cliffs, NJ, (1996).

[3] Jon Dattorro, "Effect Design, Part 2: Delay-Line Modulation and Chorus," *J. Audio Engineering Society*, 10, pp. 764-788, October 1997.

[4] C.Anderton, *Home Recording for Musicians*, Amsco Publications, New York, NY, (1996)

[5] C.Anderton, *Multieffects for Musicians*, Amsco Publications, 1995

[6] P. White. *L'enregistrement creative, Effects et processeurs, Tomas 1 et 2*. Les cahiers de l'ACME, 1993.

[7] Udo Zolzer, "*DAFX, Digital Audio Effects,*" chapter 3, Sec3.3, pp. 68, John Wiley & Sons, LTD, 2002.

[8] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, (1999).

[9] John Tomarakos, Dan Ledger, *Using The Low-Cost, High Performance ADSP-21065L Digital Signal Processor For Digital Audio Applications*, October 1999.

[10] V. Pulkki, "Virtual Sound Source Positioning Using Vector Base Amplitude Panning", *J. Audio Engineering Soc.*, Vol. 45, No. 6, pp. 456-466, June 1997.

[11] S.Disch and U.Zolzer. Modulation and delay line based digital audio effects. In *Proc. DAFX-99 Digital Audio Effects Workshop*, pp. 5-8, Trondheim, 1999.

[12] R.E. Crochiere and L.R. Rabiner. *Multirate Digital Signal Processing*.

Pretice-Hall,1983.

[13] M.R. Portnoff. Implementation of the digital phase vocoder using the FFT. *IEEE Transactions on Acoustics, Speech, and Signal Processing*,24(3):243-248,1976.

[14] Udo Zolzer, "*Digital Audio Signal Processing*," John Wiley & Sons, LTD, 1997.

[15] Richard Chinn, "*Equalizers and Constant Q*," 1989.

[16] 陳亙志, 2002, "Multi-Band Room Effect Emulator for 5.1 Channel Sound System" 國立交通大學碩士論文.

[17] Jim Coates, "*TAS3103 Equalization Filter,*" March 2003.

[18] S. J. Orfanidis, "Digital Parametric Equalizer Design with Prescribed Nyquist-Frequency Gain," *J. Audio Engineering Soc.*,Vol. 45, No. 6, pp. 444 - 455, June 1997.

[19] J. Eargle. *Sound Recording*. Van Nostrand, 1976.

[20] Udo Zolzer, "*DAFX, Digital Audio Effects*," chapter 5, John Wiley & Sons, LTD, 2002.

[21] E. Larsen and R. M. Aarts, and O.Ouweltjes. "A unified approach to low- and high-frequency bandwidth." the 115th AES convention, New York, Audio Engineering Society, 2003.

[22] E. Larsen and R. M. Aarts, *Audio Bandwidth Extension*, John Wiley, West Sussex, England, 2004.

Tables

Table 1 The grading scale used in the MUSHRA process.

Performance	**Grade**
Better	**3**
Slightly better	**2**
Perceptible better	**1**
Same as	**0**
Perceptible worse	**-1**
Slightly worse	**-2**
Worse	**-3**

Table 2 The MANOVA results of VB subjective listening tests.

Index Case	Total Preference	Artifacts	Fullness	Brightness
1st	0.025373	0.654854	0.000000	0.000001
2nd	0.566472	0.906800	0.292582	0.048493
3rd	0.038846	0.330565	0.000027	0.087174

Figures

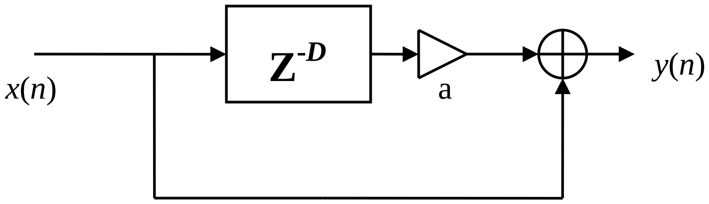


Figure 1. (a)

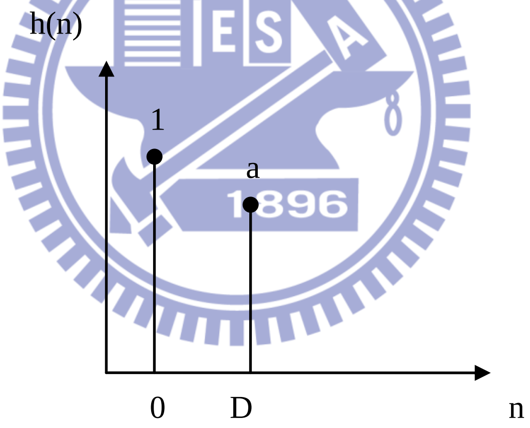


Figure 1. (b)

Figure 1. Single reflection delay.

(a) Implementation of a Digital Delay with a Single Tap.

(b) Impulse Response of a Single Reflection.

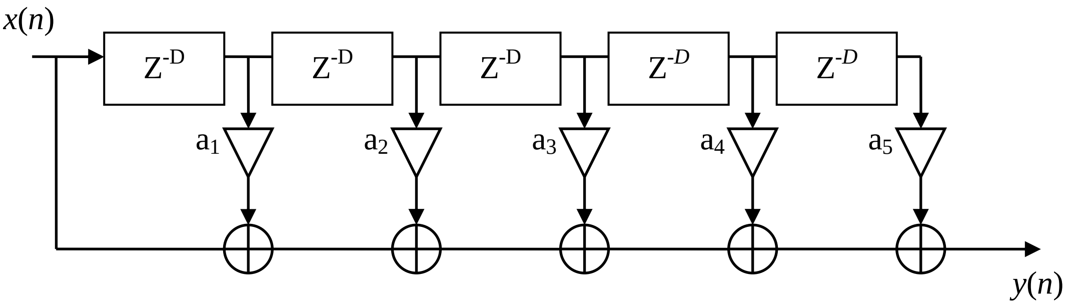


Figure 2.(a)

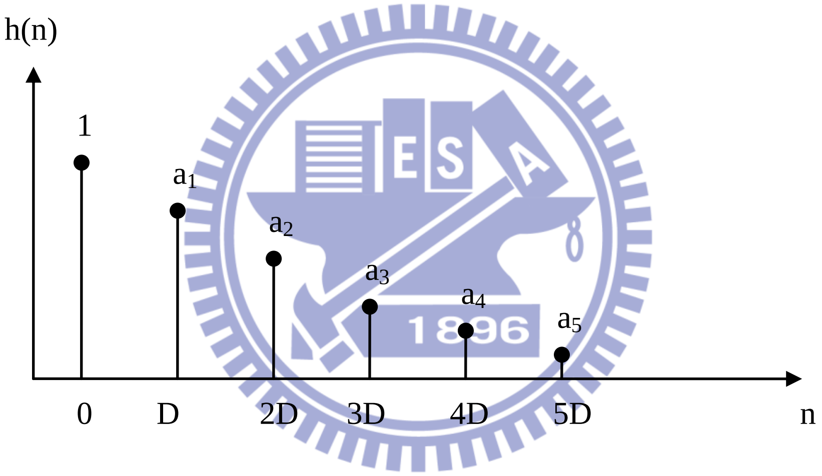


Figure 2.(b)

Figure 2. Multiple Delays (5-Tap) Example.

(a) The structure of multiple Delays.

(b) Typical Impulse Response of Multiple Delay Effect

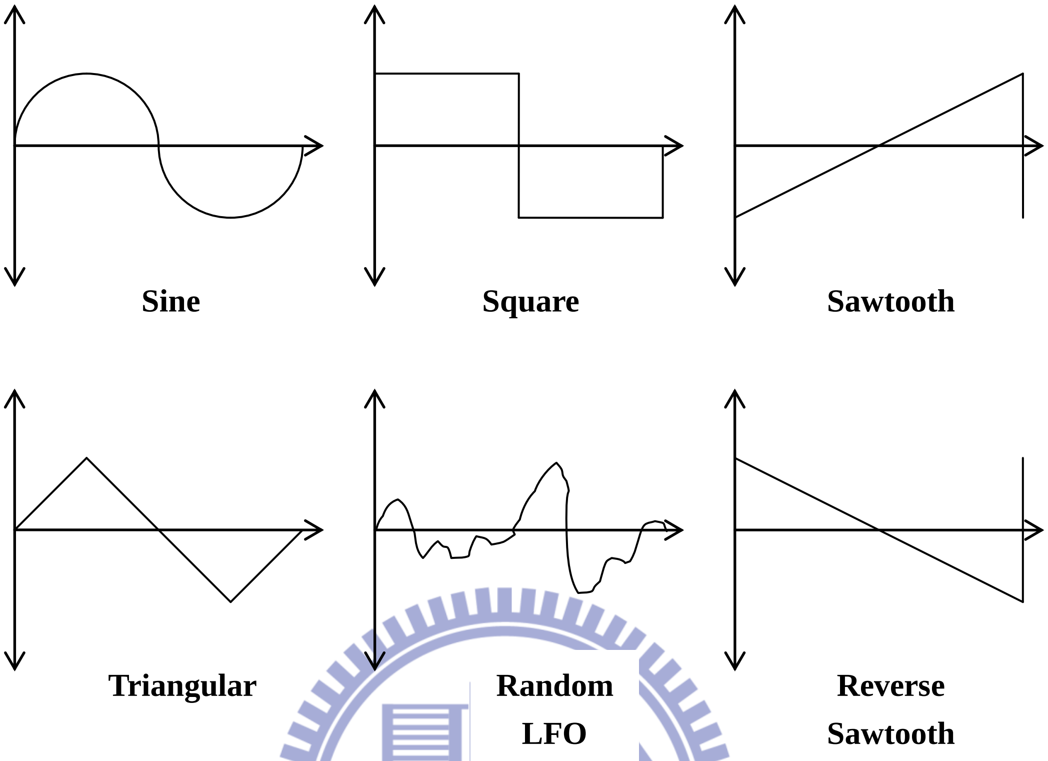


Figure 3. Common Methods of Modulation

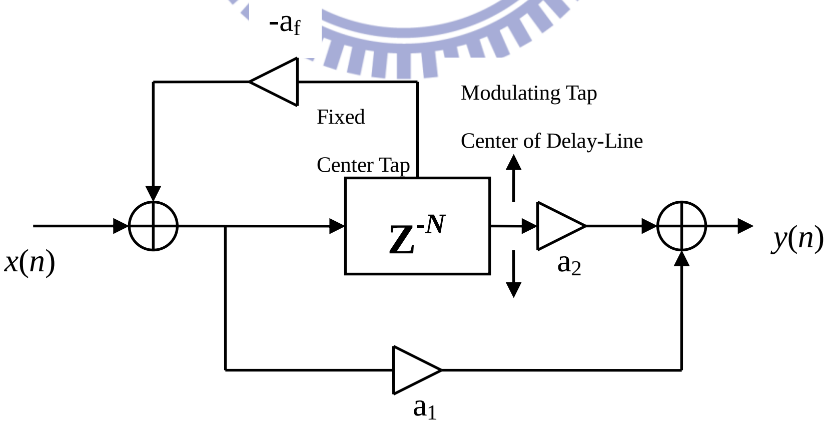


Figure 4. Delay-Line Modulation General Structure.

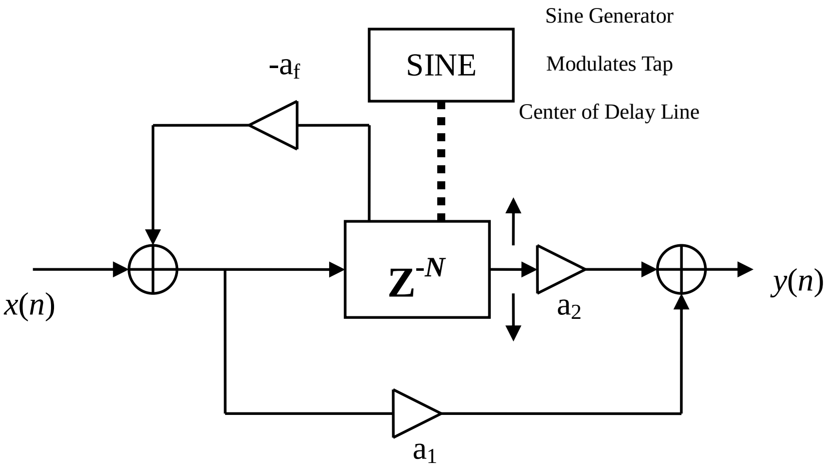


Figure 5.(a)

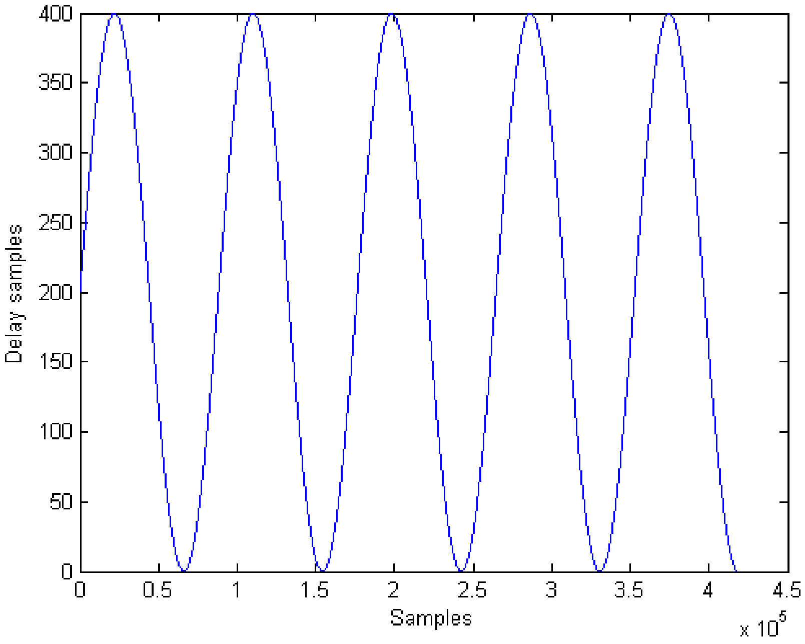


Figure 5.(b)

Figure 5. Implementation of a Flanging Effect.

(a) Block diagram.

(b) Sine wave delay points.

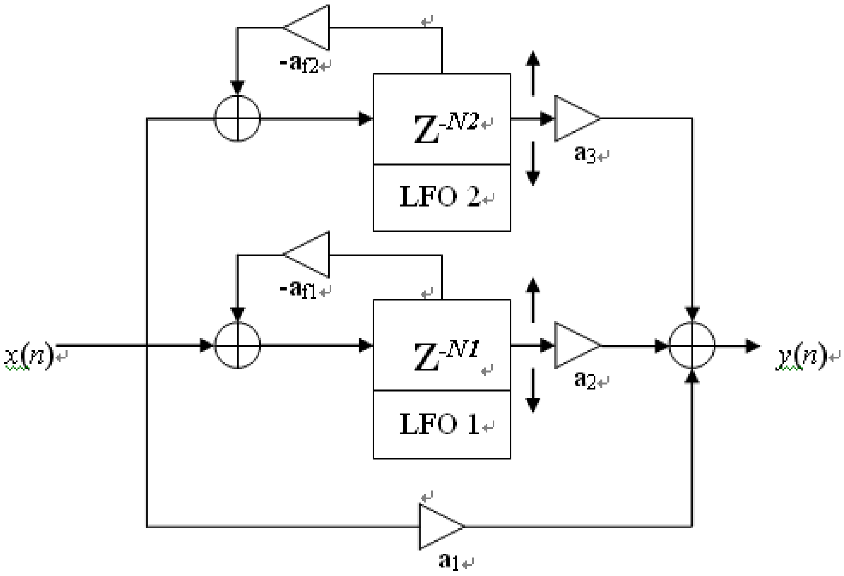


Figure 6.(a)

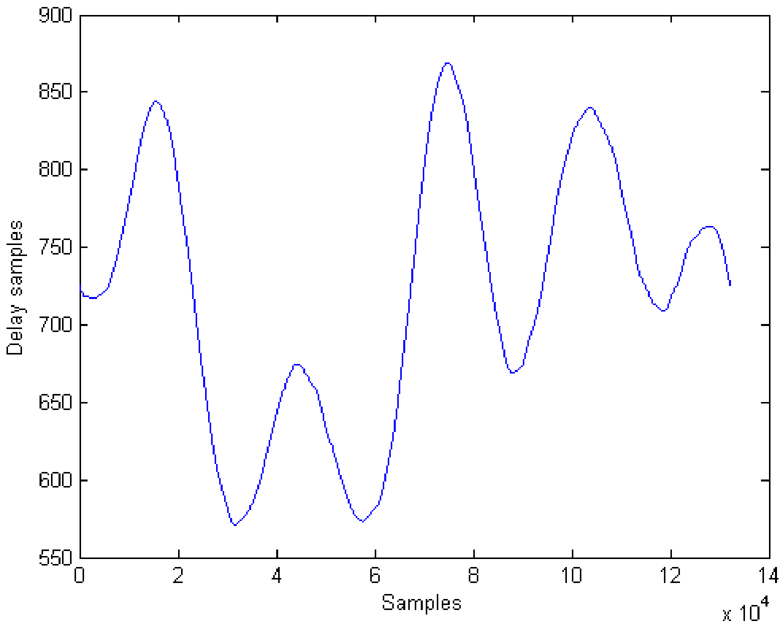


Figure 6.(b)

Figure 6. Chorus Effect Simulating 3 Instruments.

(a) Block diagram.

(b) An example of LFO wavetable.

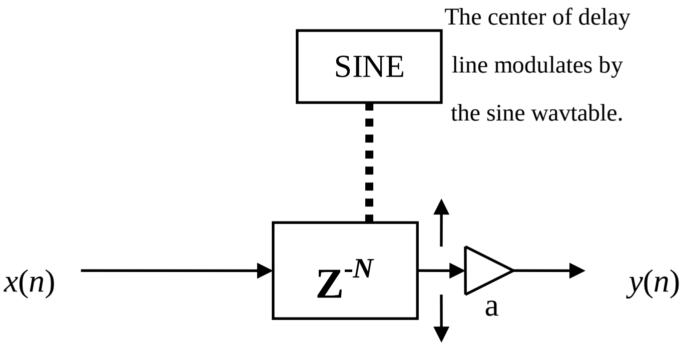


Figure 7. (a)

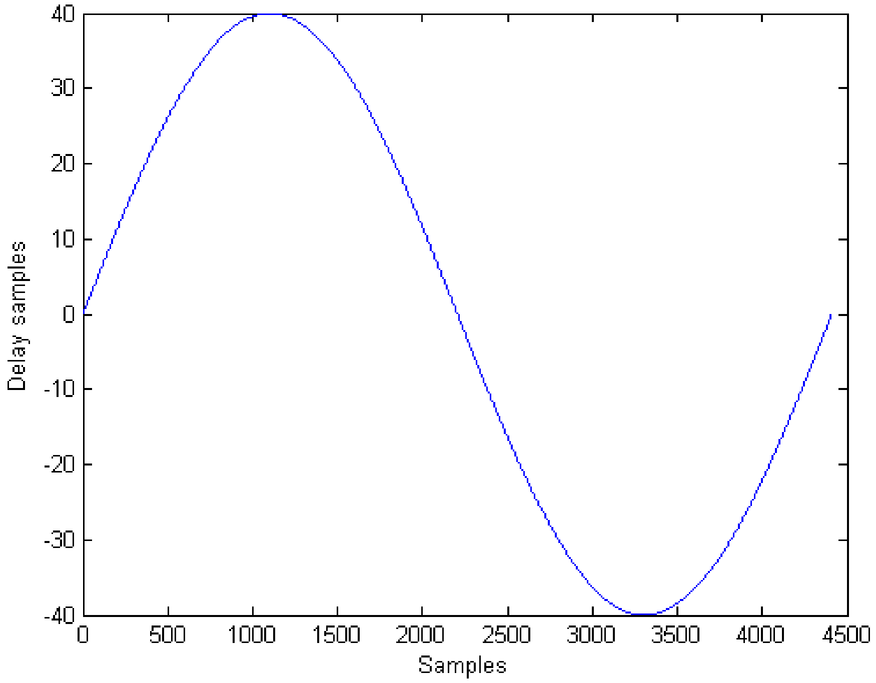


Figure 7. (b)

Figure 7. Implementation of the Vibrato Effect.

(a) Block diagram.

(b) The delay line modulates by the sine wave.

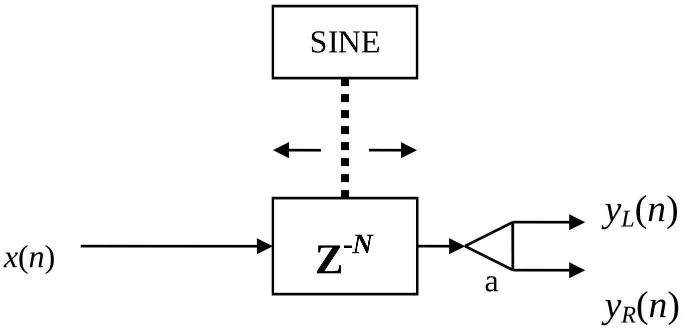


Figure 8. Stereo Implementation of the Tremolo Effect.

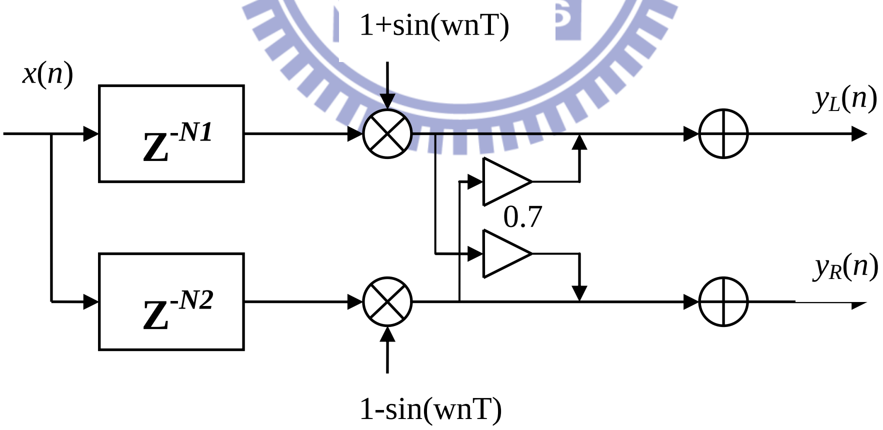


Figure 9. Rotary Speaker Simulation.

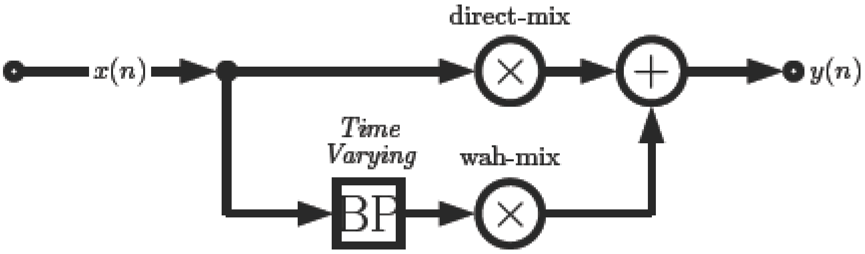


Figure 10. Wah-wah : time-varying bandpass filter

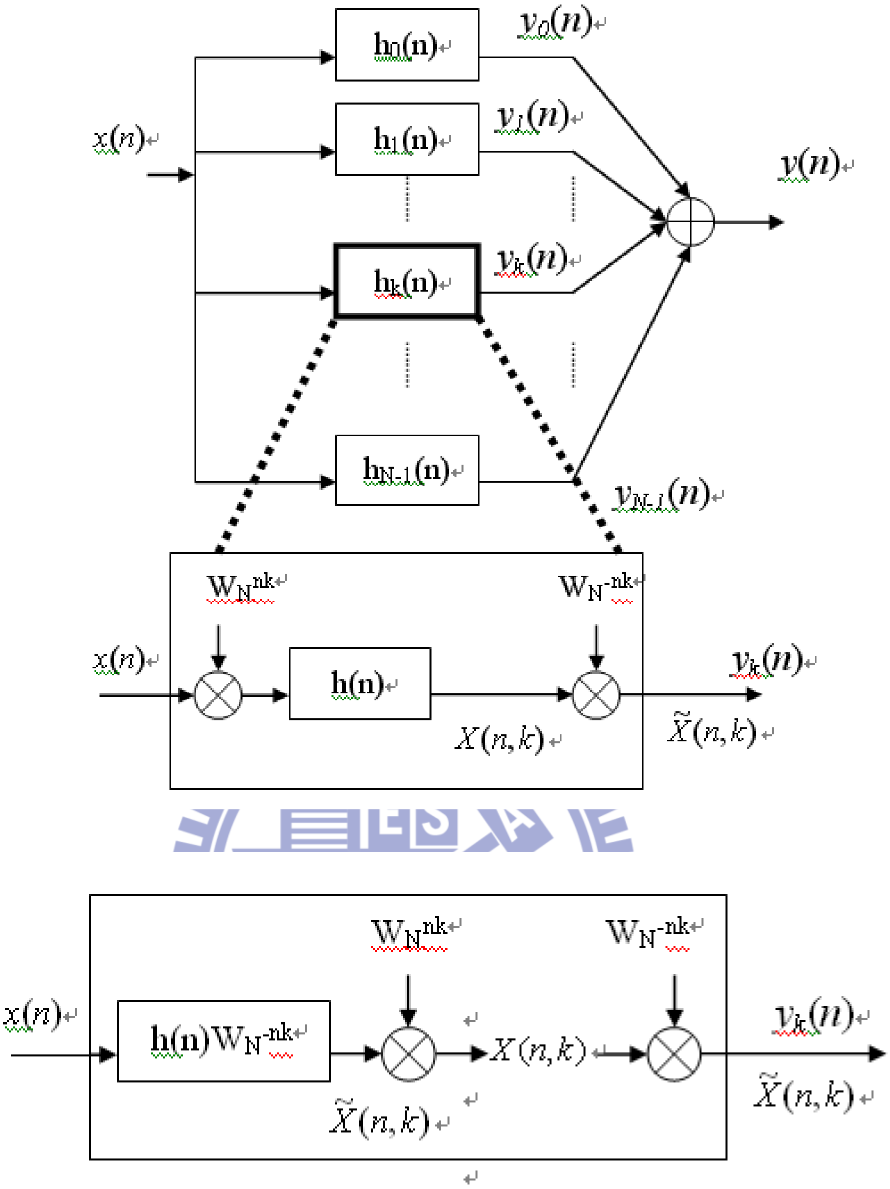


Figure 11.Two Implementation of the *k*th Channel.

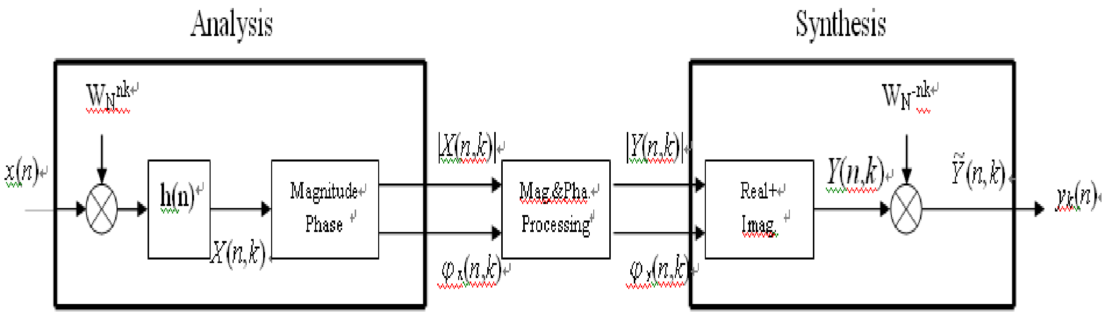


Figure 12. Heterodyne Filter Implementation.

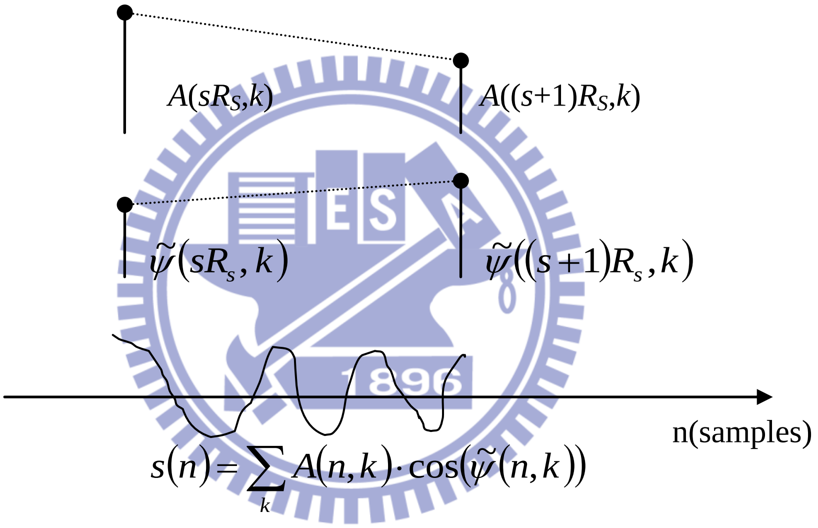

$A(sR_S,k)$
$A((s+1)R_S,k)$
$\tilde{\psi}(sR_s,k)$
$\tilde{\psi}((s+1)R_s,k)$
$s(n)=\sum_k A(n,k)\cdot\cos(\tilde{\psi}(n,k))$
n(samples)

Figure 13. Calculation of Time-Frequency samples

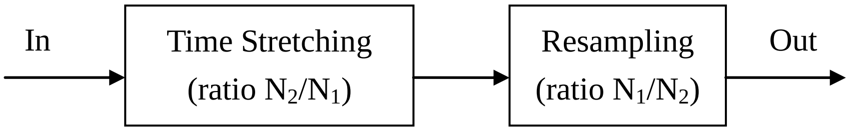


Figure 14. Resampling of a Time Stretching Algorithm.

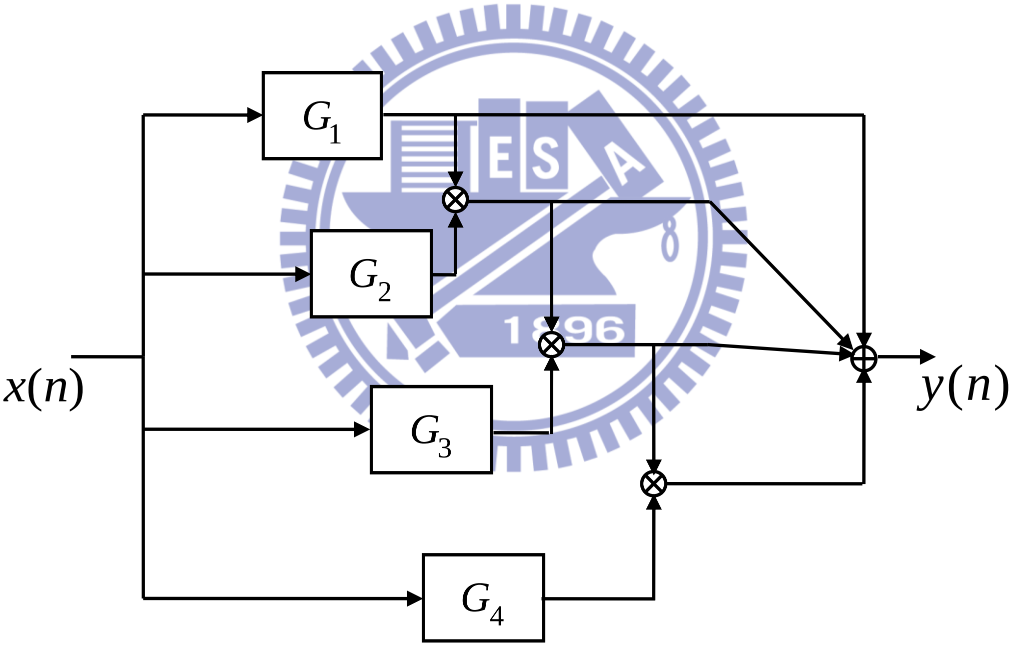
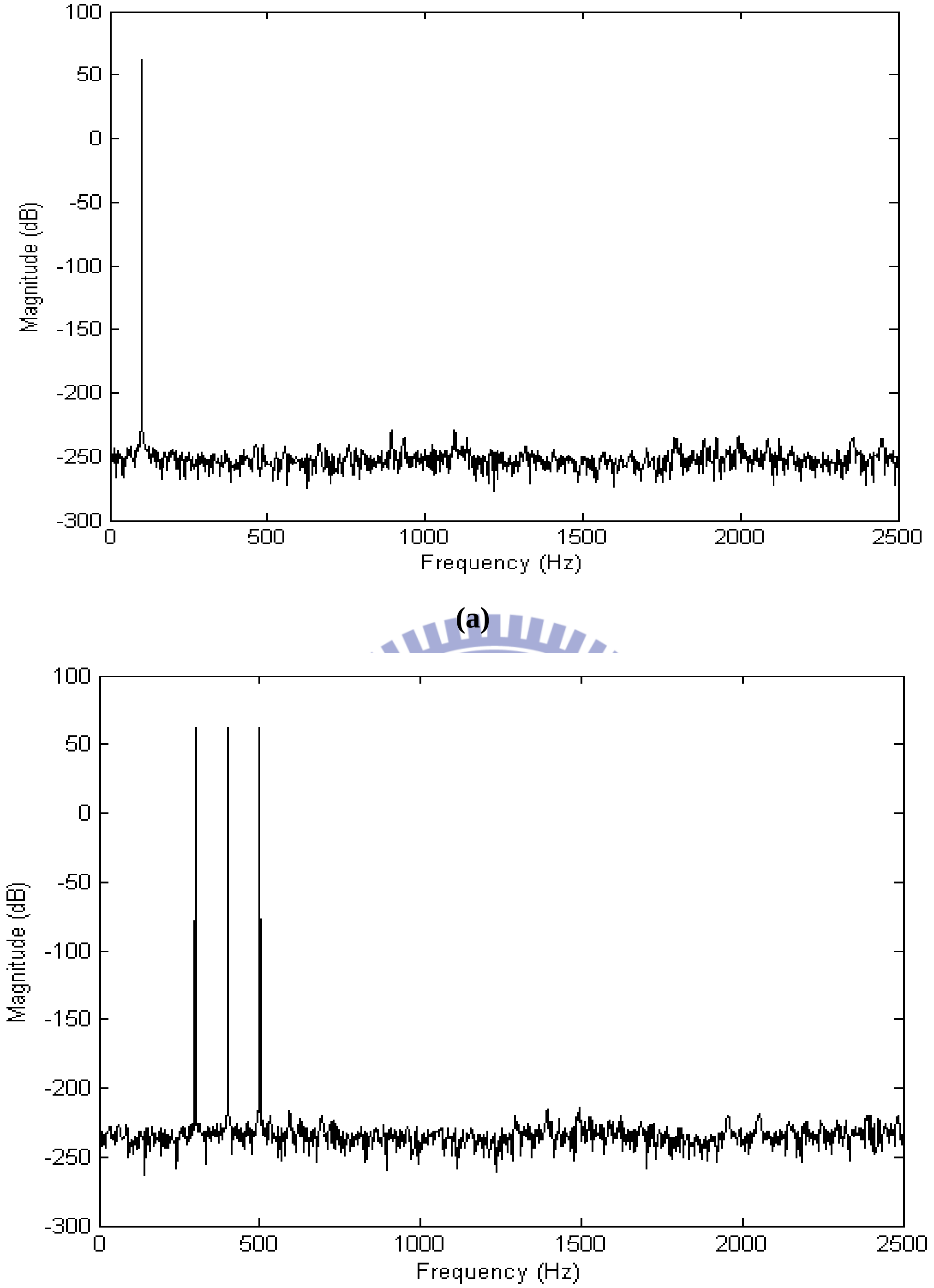


Fig. 15 Harmonics generation by multiplication.



(a)

(b)

Fig. 16 The frequency spectrums of a pure-tone signal and its harmonics by a multiplier (a) a sine wave 100Hz (b) the 3rd , 4th, and 5th harmonics of Fig. 16(a).

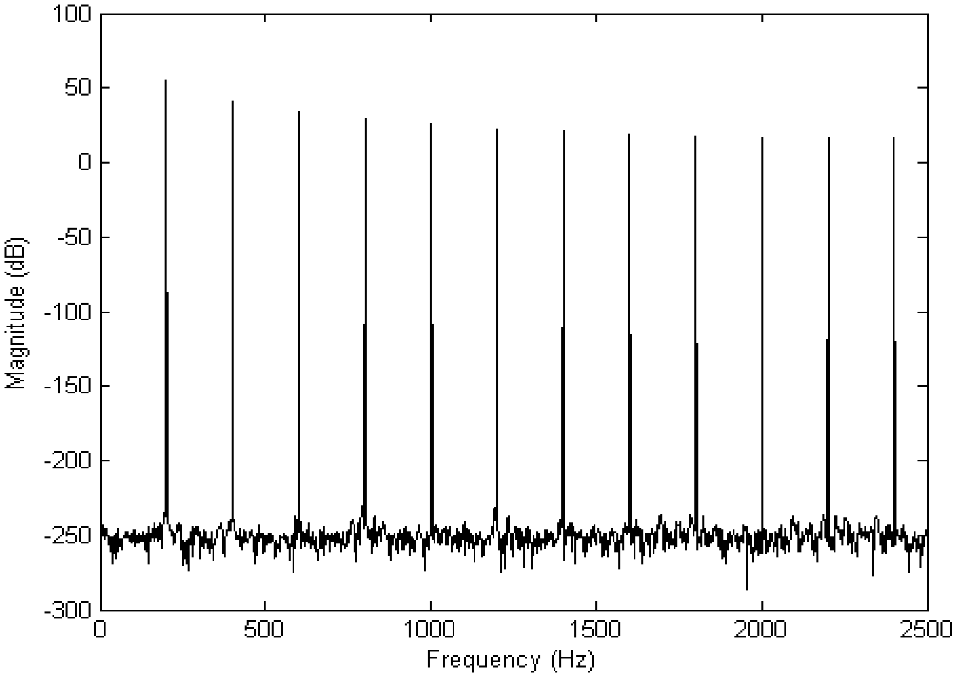
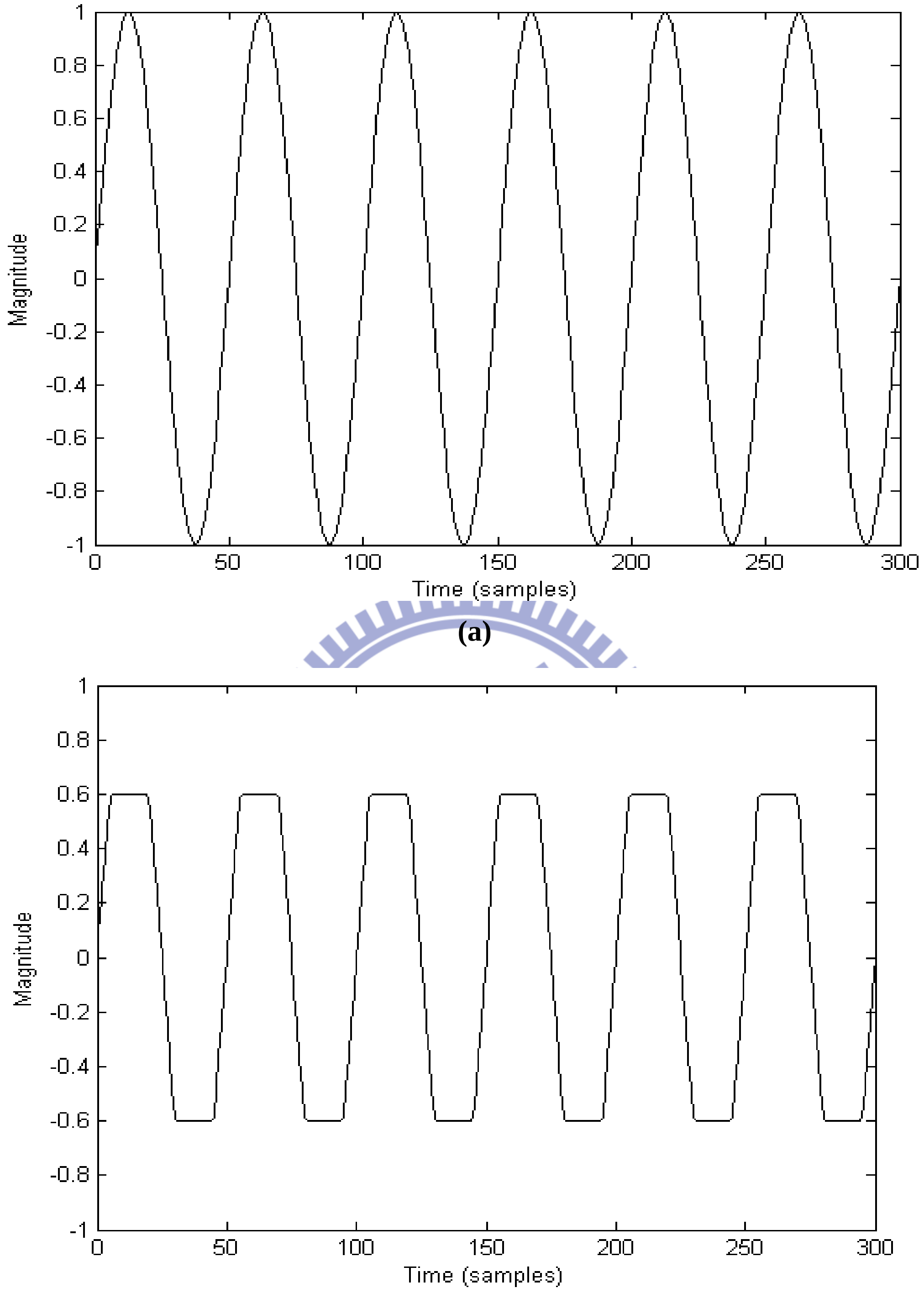


Fig. 17 The frequency spectrum of only even harmonics generated by a rectifier using a sine wave 100Hz in Fig. 16(a). The fundamental frequency is eliminated.



(a)

(b)

Fig. 18 The waveforms of a pure-tone (a) and its truncated signal by a clipper (b).

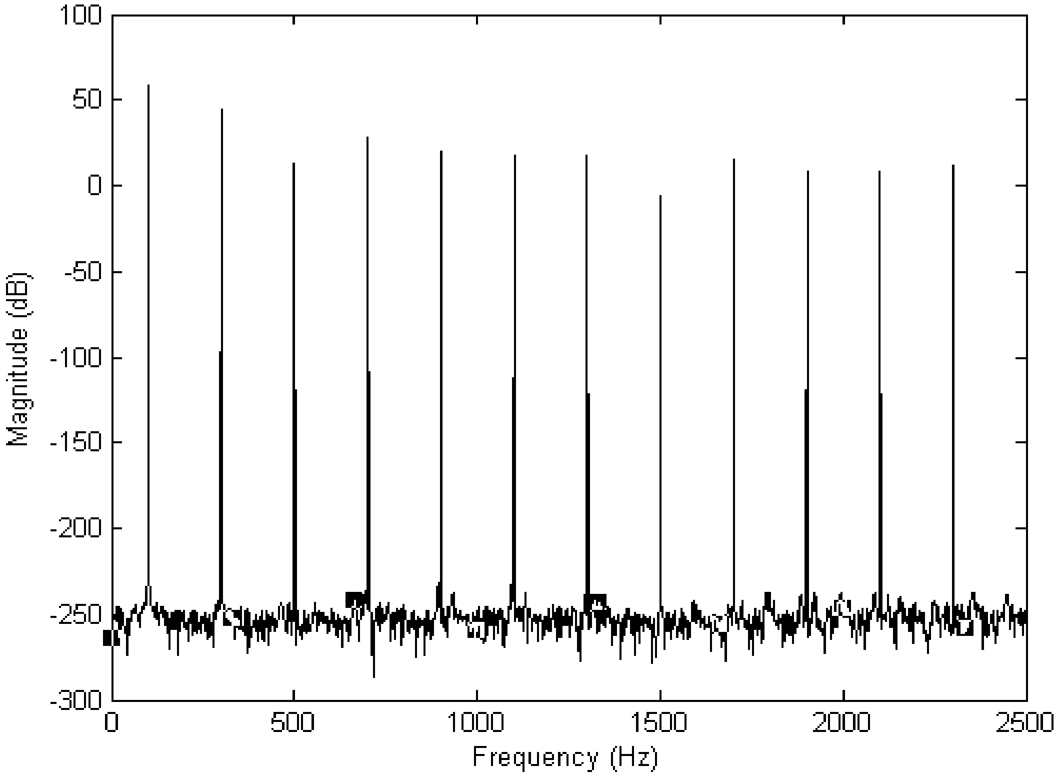
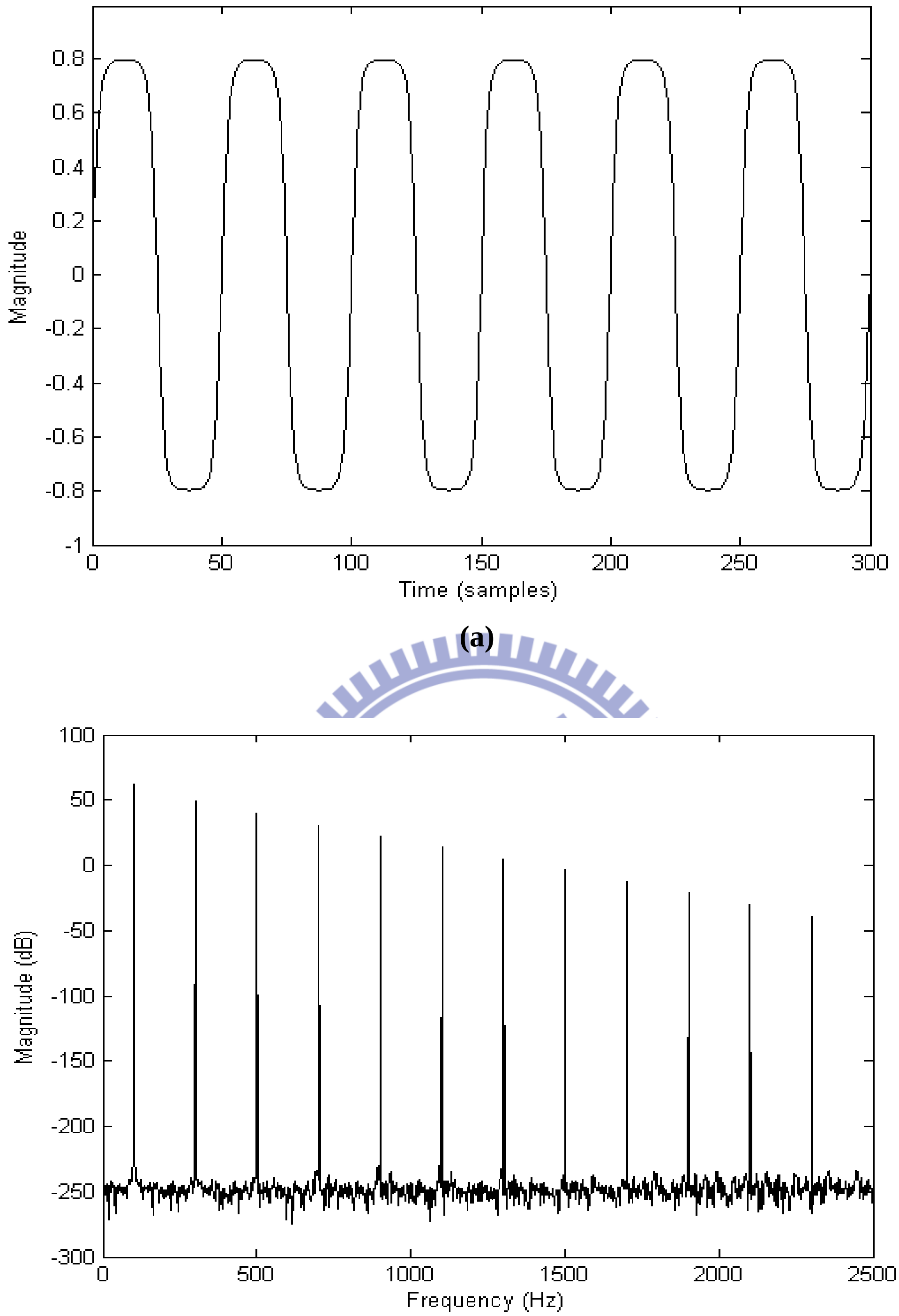


Fig. 19 The frequency spectrum of only odd harmonics generated by a clipper using a sine wave 100Hz in Fig. 16(a).



(a)

(b)

Fig. 20 The transform of a sine wave 100 Hz on the time and frequency domains by the methods of hyperbolic tangent. (a) on the time domain (b) on the frequency domain.

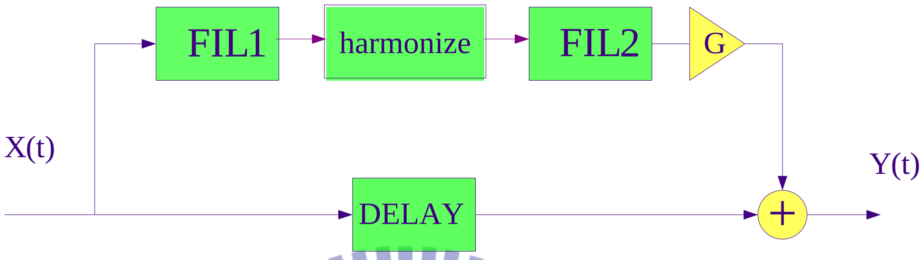


Fig. 21 The structure of virtual bass realization.

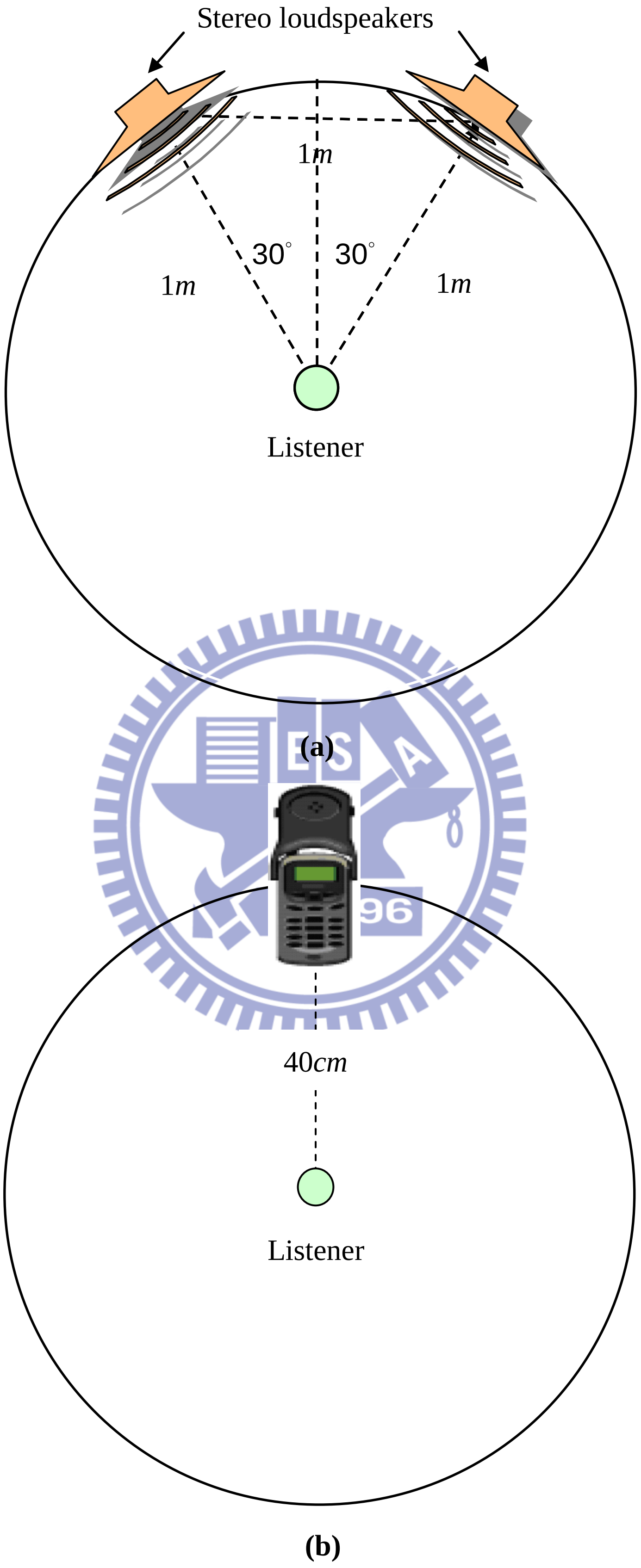
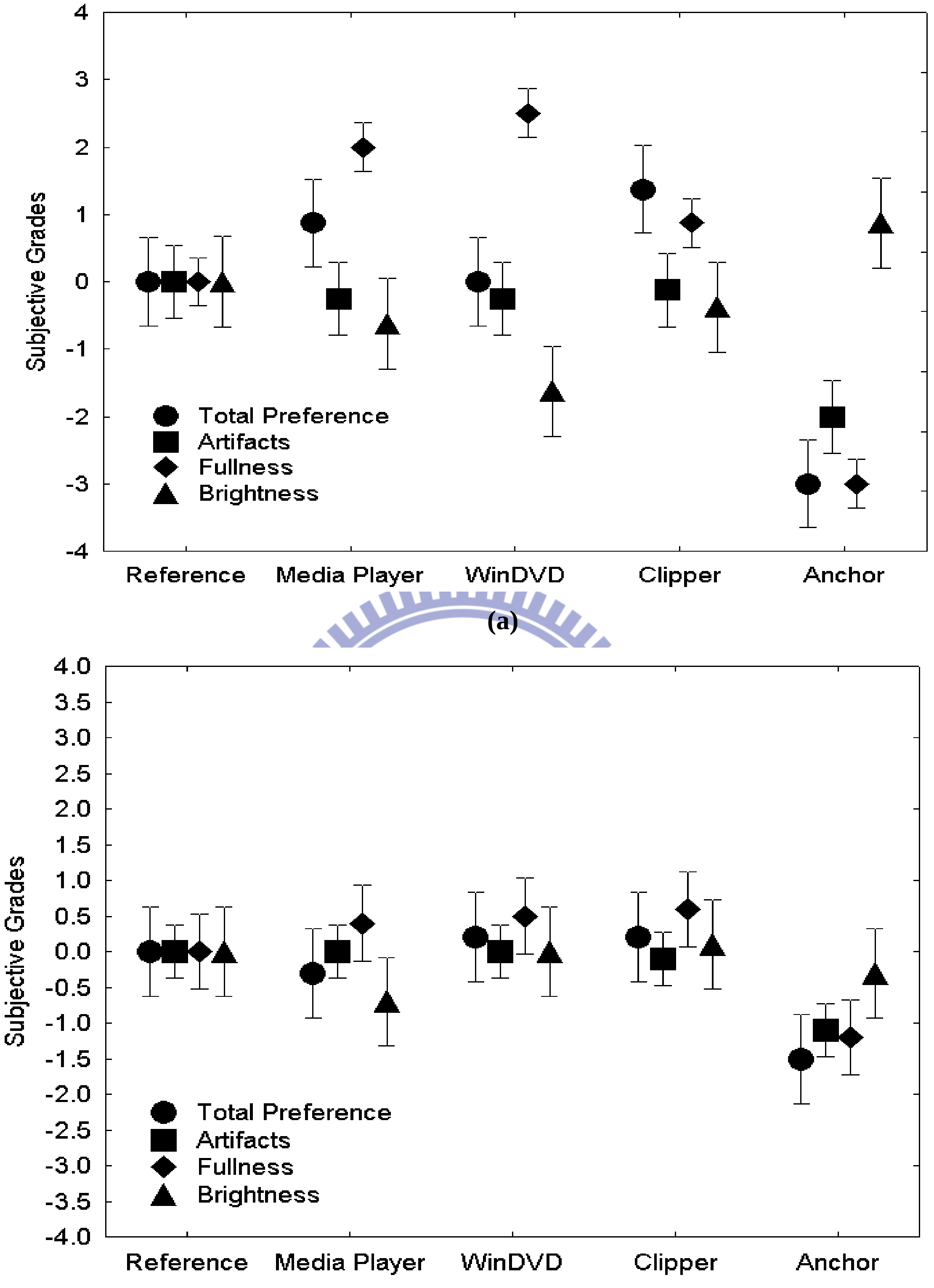
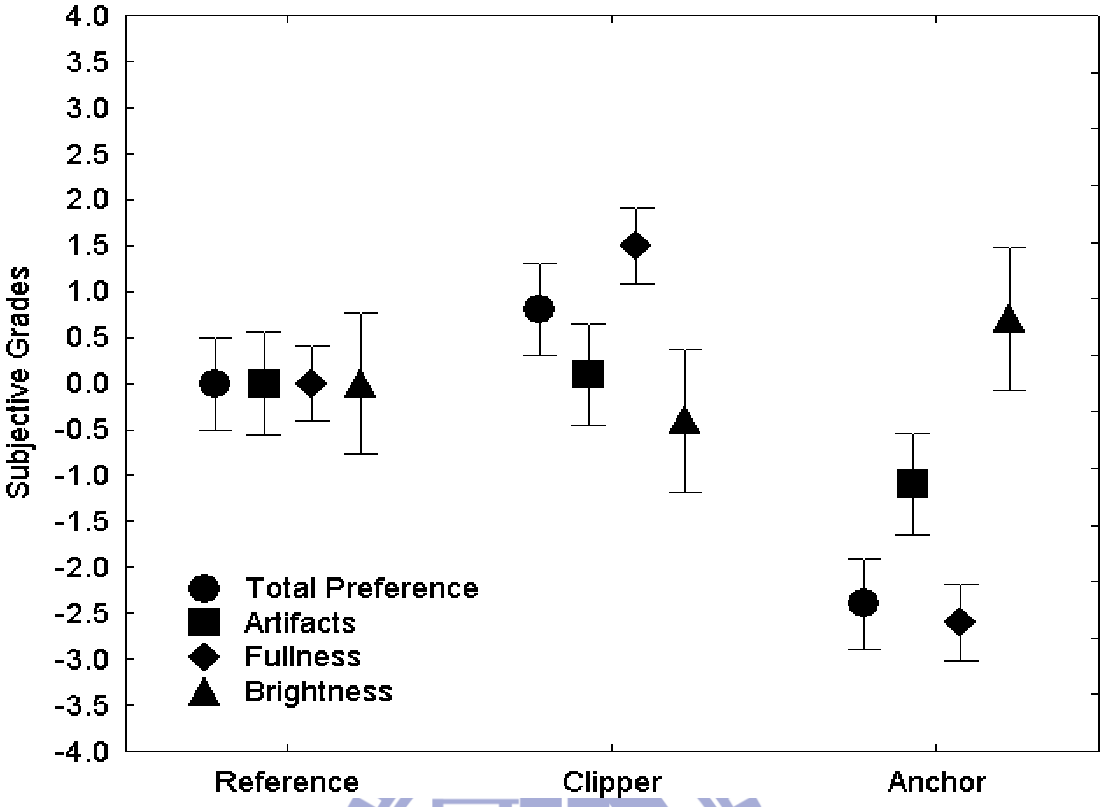


Fig. 22 (a) Arrangement for the multimedia stereo loudspeakers. (b) Arrangement for a cell phone.


4
3
2
1
0
-1
-2
-3
-4
Subjective Grades
● Total Preference
■ Artifacts
◆ Fullness
▲ Brightness
Reference
Media Player
WinDVD
Clipper
Anchor
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
-2.5
-3.0
-3.5
-4.0
Subjective Grades
● Total Preference
■ Artifacts
◆ Fullness
▲ Brightness
Reference
Media Player
WinDVD
Clipper
Anchor

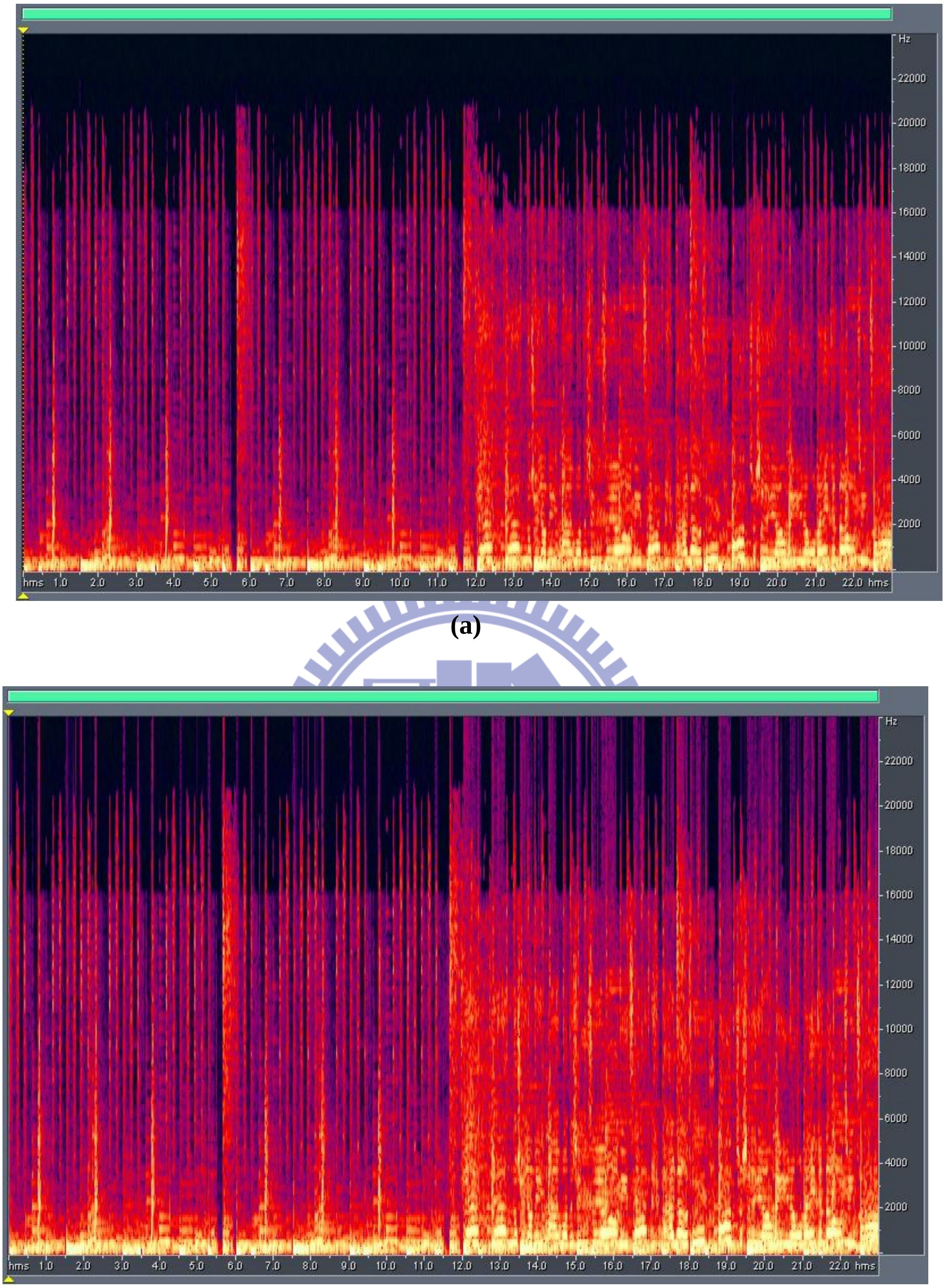
(a)

(b)



(c)

Fig. 23 Grades of the subjective listening tests of virtual bass.



(a)

(b)

Fig. 24 The time-frequency diagrams (a) an unprocessed signal (b) the result of virtual bass.

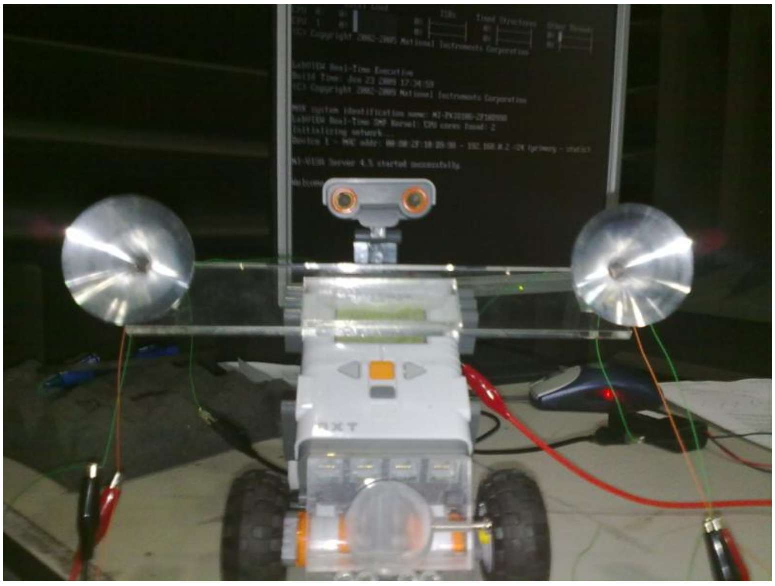


Fig. 25 The LEGO NXT Robot