1

Micro-D

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

AN EXPLORATION OF THE USE OF NONLINEAR FUNCTIONS FOR
THE SYNTHESIS OF MUSICAL INSTRUMENTS

by

Anthony A. Armstrong, Beng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of the
requirements for the degree of
Master of Engineering

Department of Electronics

Faculty of Engineering

Carleton University

Ottawa, Canada

November 1986

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-46280-9

The undersigned recommend to the Faculty of Graduate Studies
and Research, the acceptance of the thesis:

'AN EXPLORATION OF THE USE OF NONLINEAR FUNCTIONS FOR
THE SYNTHESIS OF MUSICAL INSTRUMENTS'

submitted by Anthony A. Armstrong in partial fulfillment of the
requirements for the degree of Master of Engineering.

Thesis Supervisor
J. P. Knight

Chairman
Department of Electronics
M. A. Copeland

The undersigned recommend to the Faculty of Graduate Studies
and Research, the acceptance of the thesis:

AN EXPLORATION OF THE USE OF NONLINEAR FUNCTIONS FOR
THE SYNTHESIS OF MUSICAL INSTRUMENTS

submitted by Anthony A. Armstrong in partial fulfillment of the
requirements for the degree of Master of Engineering.

Thesis Supervisor
J. P. Knight

Chairman
Department of Electronics
M. A. Copeland

ABSTRACT

This thesis explores the use of nonlinear functions, in digital table look-up form, for the synthesis of musical instruments. Specifically a 311 hertz tone for Oboe and Clarinet is analyzed and converted into a nonlinear function form. Polynomial coefficients are determined as well as post and pre distortion multiplying envelopes for timbre synthesis.

A simulation of the synthesizer in Fortran with Log and Linear sampling systems helps determine the best sampling structure.,

An extensive overview of synthesis theory makes possible the evaluation of the Nonlinear method in relation to other forms of music synthesis.

Construction of a nonlinear synthesizer with a 37 note monophonic keyboard allowed listeners to appreciate the use of the Nonlinear waveshaping process in basic form.

Comments are forwarded on the feasibility of the nonlinear method, based on the sounds produced from the constructed synthesizer and more sophisticated proposed approaches.

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor, Professor John Knight for his support and valuable feedback. His suggestions and comments were very much appreciated and allowed me to overcome all obstacles.

I thank also Nagui Mikhail for his technical support, without which I would never have assembled a working synthesizer.

I thank also Duncan Glendenning for his advice on CP6 software.

I thank Karen Brown for her attentive listening while evaluating the output sounds of the constructed synthesizer.

Finally I wish to thank my family and friends for their continued support as I slowly but assuredly made my way through the various levels of accomplishment with this thesis.

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION	1-1
	1.1 Overview	1-1
	1.2 Summary of Material	1-2
CHAPTER 2	BACKGROUND THEORY	2-1
	2.1 The Theory of Sound and Music	2-1
	2.1.0 Introduction	2-1
	2.1.1 Pitch and the Theory of Scales	2-3
	2.1.2 Loudness and Hearing	2-7
	2.1.3 The Sound Envelope	2-11
	2.1.4 Timbre (Tone Quality)	2-15
	2.2 Synthesizers and the Theory of Music Synthesis	2-25
	2.2.0 Introduction and Overview	2-25
	2.2.1 Additive and Subtractive Synthesis	2-26
	2.2.2 The Analog Approach	2-33
	2.2.3 The Digital Approach to Music Synthesis	2-36
	2.2.4 The Hybrid Approach	2-43
	2.2.5 Keyboard Interfacing	2-43
	2.2.6 Modulation Techniques	2-46
	2.3 Nonlinear Theory and Applications	2-51
	2.3.1 Introduction to the Nonlinear Technique	2-51
	2.3.2 The Polynomial Distortion (Equations and Significance)	2-52
	2.3.3 The use of Matrix Transformations	2-55
	2.3.4 Tone Synthesis using the Nonlinear Technique with a Predistortion Envelope	2-59
	2.3.5 Nonlinear System Layouts	2-63
CHAPTER 3	TIMBRE DATA ANALYSIS	3-1
	3.0 Introduction	3-1
	3.1 Polynomial Curves	3-3
	3.2 Nonlinearly Generated Waveforms	3-12
	3.3 Extracting Timbre fit Information	3-22
CHAPTER 4	SYSTEM PROPOSALS AND OVERVIEWS	4-1
	4.0 Introduction	4-1
	4.1 The Phase Generator Approach	4-4
	4.2 The A to D Approach	4-8
	4.3 Sine Table with Increment Approach	4-9

CHAPTER 5	SIMULATION OF A NONLINEAR SYSTEM	5-1
	5.0 Introduction	5-1
	5.1 Simulation Flow Chart with Program Summary	5-2
	5.2 Test Sequence Results on Log-Linear System Comparison	5-20
	5.3 Discussion of Simulation	5-22
CHAPTER 6	NONLINEAR SYSTEM HARDWARE AND SOFTWARE	6-1
	6.0 Introduction	6-1
	6.1 An Overview of Signal Generation Hardware	6-2
	6.2 Chromatic Scale Keyboard Interfacing	6-9
	6.3 EPROM Programming for Timbral Generation in the Signal Generation Hardware	6-15
CHAPTER 7	TESTING AND EVALUATION	7-1
	7.0 Introduction	7-1
	7.1 Waveforms and Spectra	7-2
	7.2 Sound Evaluation	7-5
	7.3 Further Description of Tones	7-6
CHAPTER 8	CONCLUDING MATERIAL	8-1
	8.0 Introduction	8-1
	8.1 Analysis and Simulation Routines	8-1
	8.2 Sinewave Generator...General Comments	8-5
	8.3 A Future Enhanced Approach	8-6
	8.4 Overall Comments on Nonlinear Analysis and Synthesis	8-14
REFERENCES		
APPENDIX A:	Polynomial Coefficients and Oboe Curves	A.1
APPENDIX B:	Least Squares Fit Routine	A.11

NOMENCLATURE AND SYMBOLS

EPROM	Erasable Programmable Read only Memory
Mhz	Megahertz
db	Decibel
SR	Sample Rate
f_{clock}	Clock frequency
TL	Table length
DAC	Digital to Analog Convertor
MDAC	Multiplying Digital to Analog Convertor
VCO	Voltage Controlled Oscillator
SPST	Single pole, Single throw Switch
θ	theta: fundamental phase, designating a fundamental frequency $= \omega_{\text{fund}} t$
a or a(t)	Amplitude of fundamental cosine or sine function
2θ	Twice the fundamental phase, designating twice the fund. frequency
PS1, PS2	Polynomial Structure at time frames 1, and 2
DFT	Discrete Fourier Transform
TF1, TF2	Time Frame 1, and Time Frame 2.
NSAMP	Number of Samples
A to D	Analog to Digital Conversion
T_s	Sample Period
$f_{\text{low}}, f_{\text{high}}$	Low and High Fundamental frequencies on a keyboard
ϕ_1, ϕ_2	Two nonoverlapping phases of a two phase clock
hex	Hexadecimal number system
TDM	Time Division Multiplexed

CHAPTER 1: INTRODUCTION

1.1 Overview

The purpose of this thesis was to investigate the use of the nonlinear function to synthesize tones normally produced by conventional musical instruments. It was intended this synthesis be done over a range of instruments but due to limits of time and resources only two were synthesized, namely the Oboe and the Clarinet.

The reason for synthesizing standard musical instruments is so that a keyboard artist can produce a wide range of musical tones. Also, once an instrument has been defined by synthesis, its timbre characteristics can be slightly altered, causing a new yet pleasing type of tone.

The nonlinearity was used in digital table form and further enhanced with a predistortion envelope multiplication as well as a postdistortion envelope multiplication. The predistortion envelope caused changes in spectral content to occur during a single tone while the post distortion envelope was responsible for overall amplitude changes. The envelope curves were determined by a least squares fit analysis. The nonlinear function was derived from the desired harmonic structure with the use of Chebyshev functions.

The use of nonlinear functions and envelope multipliers is a viable alternative to the simpler additive synthesis since it allows vast compaction of data while retaining tonal integrity. For example, if 14 harmonics are retained in the synthesis, then with the additive method, 14 envelope curves and oscillators are required, whereas with nonlinear synthesis only one oscillator and three curves are necessary. (one nonlinear curve and two

envelopes)

The timbre data used to derive the nonlinear curves and envelopes originated from papers published in 'The Computer Music Journal'.

To ensure that proper sampling structure was used, with regards to the sampling number system, a simulation of the entire digital nonlinear synthesis was completed on the CP6 mainframe. This simulation, which ended with a discrete fourier transform to analyze a 256 point sample string, used some 30 subroutines and turned out to be a few thousand lines of code. To cover all possibilities for number systems the simulation included an option with log sampling. After running a number of tests it was decided log sampling would not be used in the final design.

When assembling hardware, a wire-wrap process was used, on two large circuit boards. Off the shelf digital hardware (TTL and CMOS) was used for all assembled circuitry. No specialized music type I.C.'s were used. Interface circuitry was breadboarded and used in that form. The Chromatic keyboard scanner was a kit, assembled according to detailed instructions. EPROM blowing was completed with a special EPROM blower integrated into an Apple 2 computer. Hardware assembly and testing took about 6 months after the final design work had been completed. A 6809, 8 bit microprocessor, was used to regulate data transfer to Signal Generation Hardware. The most expensive hardware besides the microprocessor was the TRW 12 bit multipliers which cost about 200 dollars each.

1.2 Summary of Material

The whole process of initiating the thesis began with a careful survey of a wide range of literature in the fields of Computer Music and Music Synthesis.

To understand more about the musical nature of synthesizers, a large amount of theory is presented in Chapter 2. This theory begins with basic definitions (pitch, scales, loudness, sound envelope) and moves into an outline of the properties of musical instrument timbre. The outline of timbre is fundamental in understanding what makes a synthesized tone distinguishable from an original instrumental tone. This theory proves useful in evaluating and explaining opinions on the accuracy of the nonlinear synthesis which is completed towards the later part of the thesis.

A section of theory follows which explains a variety of synthesis techniques. It also explains the method used to extract harmonic curve data which in fact was the way tonal data was derived for the Clarinet and Oboe tones.

A brief outline is made of the analog and digital approaches to music synthesis. It should be known that an analog approach was originally intended to have been used for the working portion of the thesis but it was discovered that diodes, the most practical analog nonlinear element, were limited.

After a brief look at Hybrid synthesis the theory converges on Modulation Techniques including Frequency Modulation and Nonlinear Waveshaping. Waveshaping is discussed in detail in the final section of theory.

Chapter 3 deals with a look at extraction of functions and data from a 311 hertz Clarinet and Oboe tone. A large number of curves are presented to illustrate in detail the form that the data takes before and after analysis.

Chapter 4 gives us three possible approaches to generating a sine wave to be used with a nonlinearity:

1. Nonlinear Timbre Generation with divide-by-N circuitry to produce the desired fundamental frequency. (Phase Generator Approach)
2. Nonlinear Timbre Generation with an Analog Sinewave source.
3. Nonlinear Timbre Generation with a Variable Increment Phase-to look-up through a Sine Table.

The third approach was favored and is explored in detail in a Simulation which follows.

Chapter 5 deals with the CP6 mainframe FORTRAN simulation of a synthesizer system with nonlinear harmonic generation, pre and post distortion multiplication and incremental phase stepping through a sine table.

Chapter 6 explains the hardware as it was developed based on prior analysis and a knowledge of digital circuitry. Signal Generation Circuitry is outlined in this section along with the interfacing circuitry and EPROM programming. The heart of the system is in fact a series of counters, latches, multipliers, adders and combinatorial logic set up to realize the block diagram presented. The Signal Generation Hardware was wirewrapped while the interface portion of the system was only breadboarded. Extensive testing was completed during development to ensure proper signal

generation was occurring.

A complete series of tests is outlined in Chapter 7.

This leads to the 'Concluding Material' of chapter 8 which is a collection of comments made on various aspects of the thesis, giving further insight into the approaches taken and the nonlinear system in general.

2.0 BACKGROUND THEORY

2.1 The Theory of Sound and Music

2.1.0 Introduction

It is necessary to describe some basics of music theory before we are able to account for the range of sound effects the non-linear synthesizer is capable of producing.

The characteristics of music may be classified with respect to tone, dynamics, tempo, and quality. These characteristics are used to form the basis of all music. A music composition consists of a series of expressive or intelligible tones of definite structure and significance according to the laws of melody, harmony and rhythm. (Olson, 67) In its simplest form, a musical composition consists of a melody and its accompaniment.

A melody is "a succession of notes varying in pitch, which have an organized and recognizable shape". (Kennedy, 1985) Melody is 'horizontal'. This means the notes producing it are consecutive whereas harmony is 'vertical', meaning the notes producing it are simultaneous. Accompaniment implies a principal performer with a subservient, supplied performer or group of performers, the subservient performers providing accompaniment. With a modern synthesizer one performer may provide both melody and various levels of accompaniment through a record-playback-overlay record sequence. Even a monophonic synthesizer (producing one tone at a time) may produce melody and accompaniment if a series of tapes are made and overlaid properly.

The basic element of melody, the tone, has several components. Two of these components are pitch and timbre. Basically timbre is a sound

sensation having pitch in all its various complex and simple forms. A series of tones graded in pitch from low to high is referred to as a scale. Section 2.1.1 defines and describes tonal scales.

The merits of a good synthesizer are based on the range of tones it can produce as well as how well it can simulate other standard tones which conventional instruments already produce.

Music prepared from recorded sounds either in nature (birds, water in river etc.) or manmade (traffic, machines etc.) is termed Musique Concrete. Strictly speaking concrete music should not be modified electronically but the distinction between it and electronically synthesized sound has been increasingly blurred to the point that the term electronic music covers the whole process.

The dynamics of music are the graduations in intensity of its sounds. Dynamics depend simply on loudness. Loudness is largely a function of hearing. Section 2.1.2 describes the concepts of loudness as they are related to human hearing.

The temporal characteristics of music involve duration and tempo. In an analytic perspective the duration of a tone depends on its sound envelope. Tempo is the rate of speed at which a musical composition is played or sung. Tempo can affect duration. Meter is the recurrence or repetition of stress, beat, or sound accent, occurring in a regular pattern. Meter and tempo are partly functions of the way the musician plays his instrument. In the case of a music synthesizer the instrument 'played' is primarily an electronic keyboard. We cannot really analyze tempo or meter from an instrumental viewpoint, unless of course there is a special rhythm section built into the synthesizer. We will therefore look simply at the sound envelope (section 2.1.3) to understand temporal characteristics. The

envelope however is also a fundamental element of timbre as well, and as such is an important factor in discriminating between the tones of different instruments.

Finally we state that the qualitative aspects of music involve timbre, or the harmonic constitution of a tone. Section 2.1.4 introduces some important aspects of timbre, necessary for tone evaluation.

2.1.1 Pitch and the Theory of Scales

Pitch is an auditory sensation which may be ordered on a scale extending from low to high, such as a musical scale. Pitch is dependent upon the frequency of the sound stimulus, the more the vibrations per second of the radiating body the higher the pitch.

The lowest limit of pitch is that frequency which gives us the lowest sensation of tone. This usually corresponds to about 16 hz but varies with a number of factors. One of these is intensity; if low frequency tones are to be heard they must be fairly loud. Under the most favourable conditions a good listener can get tonal fusion at as low as 12 hz.

The upper limit of pitch is the highest frequency which can be heard. This depends on the individual's hearing efficiency but for most people under the age of forty (the upper limit decreases with increased age), the upper limit is 16,000 cycles per second or 16khz. The smallest difference in pitch between two tones which an individual can detect is termed 'pitch discrimination'. Pitch discrimination is tested by sounding two frequencies in rapid succession and gradually reducing their frequency difference until the observer claims they are identical. (Olson, 67) Figure 2-1 illustrates pitch discrimination as a function of frequency and intensity. Between 1khz and 10khz a .5% change in frequency goes undetected. It can be seen that the ear is most sensitive to high frequency changes. Between 50hz and 100 hz a 20 db signal has an undetected frequency change at up to 5% difference. Typically the total number of 'just noticeable differences' in pitch which an individual can hear throughout the hearing range is about 1400. This is enough to define a large series of scales of notes with highly perceptible

Figure 2-1 The Noticeable variation of Δf in relation to Frequency. Note that the Δf is the perceptible change in frequency. Numbers on the curves indicate phons above the threshold of hearing. (Olson, 67)

changes in pitch between defined tones.

It should be noted that a tone must persist for a certain length of time in order to establish pitch. If the tone duration is very short, say one millisecond, it sounds like a click. At a somewhat longer duration it may sound like a noise with some attributes of pitch. Finally as duration is further increased the tone establishes a definite pitch.

Scales

Around 600 BC Pythagoras discovered that strings under equal tension sounded harmonious if their lengths were in the ratio of whole numbers like $2/1$, $3/2$, $4/3$, $5/3$, and so on. Since that time experiments have confirmed that for pleasing sound combinations it is the ratios of frequencies of notes that is important, not the absolute frequency.

With this understanding, musical scales were developed. When two or more tones are sounded simultaneously and the result is pleasing to the ear, the resultant sound is termed consonant. Alternatively a combination of tones which is not pleasing to the ear is termed dissonant.

Scales are comprised of intervals. Intervals are the ratios of the frequencies of the scale's tones, or the 'distance' between two notes. Intervals are so important that they have been assigned names. For example a $2/1$ ratio is called an octave, $3/2$ is called a perfect fifth, $4/3$ is called a perfect fourth, $5/3$ is called a major sixth. These intervals are actually used to form the 'Pure Diatonic Scale', also known as the 'Scale of Just Intonation'. Table 2-1 depicts the intervals in this scale and their frequency ratios. (Olson) Note that a 'cent' is the interval between two tones whose basic frequency ratio is the twelve-hundredth root of 2. One octave is 1200

cents, 12 semitones or a frequency ratio of $1/2$. The cent is important in defining the 'Equally Tempered Diatonic Scale'.

Pianos, electric organs, and most other musical instruments are tuned to a scale slightly different than the pure diatonic scale, namely the equally tempered diatonic scale. This scale involves intervals, or frequency ratios, which are not integers, but rather the tones within the octave are all equally spaced. This equal spacing is referred to as equal temperament. Equal temperament, introduced in the 16th century allows music to be played in any key or key signature without changing the tuning of the instrument. The key or key signature describes the specific scale or tone/semitone structure which will be used. It was partly to prove the value of equal temperament that J.S. Bach wrote his 'Well-tempered Clavier', a compilation of fugues and preludes in all keys. These pieces were written for keyboard instruments tuned in equal temperament so that all key signatures were equally out of tune or equally tempered. (Lloyd, 68)

In the equally tempered scale the ratio of frequencies of any two adjacent notes turns out to be the twelfth root of two. (approximately 1.0594631) Musicians use this ratio where there are twelve half steps per octave and the octave represents the $2/1$ ratio. The octave is the only purely harmonic interval in this scale. The perfect fifth is .11% low (compared to the interval of the pure diatonic scale), the perfect fourth is .11% high and so on. Since the most discriminating ear can only perceive differences in frequency of more than .2% (fig. 2-1), for a given note, the most harmonious intervals (octave, fifth, and fourth) are indistinguishable between the two scales. Figure 2-2 illustrates the key of C Major and the corresponding frequencies of the pure and equally tempered diatonic intervals. Note that in the pure scale a half step or semitone up in pitch is

an increase of $16/15$ in frequency and a whole step is an increase of $9/8$ or $10/9$. In the tempered scale all half steps are an increase in frequency by the twelfth root of two, and all whole steps up in pitch increase the frequency by the sixth root of two, ($2^{1/6}$). It may also be noticed that the difference between the pure and tempered notes is imperceptible for four of the eight illustrated notes. Also the .91% difference between the pure and tempered 'A' notes refers to their interval ratio relative to C and not their absolute frequency ratio. (which is obviously 440 hz for both) By convention, scales are tuned in North America to the A above middle C or 440 hz. The difference in the 'A' note intervals is explained by the fact that C is 264 hz for the Pure scale and 261.6 hz for the tempered scale. Figure 2-3 illustrates how with changing key signatures the tempered scale retains its frequencies while the frequencies of the pure scale change.

The Chromatic scale (see table 2-2), the most widely used tempered scale, is so called because it has tones that are not in the key of the composition and as such adds 'colour' (the Greek meaning of the word chroma) to the composition. (Lloyd, 68)

2.1.2 Loudness and Hearing

As stated in the introduction, the dynamic aspects of music depend primarily upon the intensity. The loudness of a sound is the magnitude of the auditory sensation produced. The loudness level of a sound in phons (a common loudness unit) is numerically equal to the sound level in decibels of a free progressive sound wave of 1000 hz which is judged to be of equal loudness. The decibel rating of the free progressive wave is taken relative

Interval	Frequency ratio from starting point	Cents from starting point
Unison	1:1	
Semitone	16:15	111 731
Minor tone (second)	10:9	182 404
Major tone (second)	9:8	203 910
Minor third	6:5	315 841
Major third	5:4	386 314
Perfect fourth	4:3	498 045
Augmented fourth	45:32	590 224
Diminished fifth	64:45	609 777
Perfect fifth	3:2	701 955
Minor sixth	8:5	813 687
Major sixth	5:3	884 359
Harmonic minor seventh	7:4	968 826
Grave minor seventh	16:9	996 091
Minor seventh	9:5	1,017 697
Major seventh	15:8	1,068 269
Octave	2:1	1,200 000

Table 2-1 : The Scale of Just Intonation. (Olson, 67)

Key of C Major									
Musical Terms	Interval From C	Unison	Major Second	Major Third	Perfect Fourth	Perfect Fifth	Major Sixth	Major Seventh	Octave
	Sung	do	re	mi	fa	sol	la	ti	do
	Called	C	D	E	F	G	A	B	C
	Written
Physicist's or Mathematician's Terms	Frequency								
	Pure	264	297	330	352	396	440	495	528
	Tempered	261.6	293.7	329.6	349.2	392.0	440	493.9	523.3
	Frequency Ratio to C for								
Musical Terms	Pure	1/1	9/8	5/4	4/3	3/2	5/3	15/8	2/1
	Tempered	1	2 ^{1/6}	2 ^{1/3}	2 ^{5/12}	2 ^{7/12}	2 ^{3/4}	2 ^{11/12}	2
	Tempered/ Pure	0	-0.23	+0.79	+0.11	-0.11	+0.91	+0.68	0
	Step Size		Whole	Whole	Half	Whole	Whole	Whole	Half

Figure 2-2 : The Key of C Major. Note that :

$$\text{Tempered/Pure} =$$

$$100 * ((\text{Pure ratio} - \text{Tempered ratio}) / \text{Pure ratio})$$

(Struve, 79)

Figure 2-3 : The Key of A minor. Note that the minor scale has a different whole step/half step structure than the major scale. (see figure 1-2) Comparing the pure and tempered scales we can see why musicians may prefer the non-changing frequencies of the even tempered scale. (Struve, 79)

Interval	C Major	A Minor	Tempered Ratio
1 Unison	C-C	A-A	2 ^{0/12}
2 Minor Second	C-D ^b	A-B ^b	2 ^{1/12}
3 Major Second	C-D	A-B	2 ^{2/12}
4 Minor Third	C-E ^b	A-C	2 ^{3/12}
5 Major Third	C-E	A-D ^b	2 ^{4/12}
6 Perfect Fourth	C-F	A-D	2 ^{5/12}
7 Tritone	C-G ^b	A-E ^b	2 ^{6/12}
8 Perfect Fifth	C-G	A-E	2 ^{7/12}
9 Minor Sixth	C-A ^b	A-F	2 ^{8/12}
10 Major Sixth	C-A	A-G ^b	2 ^{9/12}
11 Minor Seventh	C-B ^b	A-G	2 ^{10/12}
12 Major Seventh	C-B	A-A ^b	2 ^{11/12}
13 Octave	C-C	A-A	2 ^{12/12}

Table 2-2 : The Chromatic Scale. Note: the scale is even tempered. Also the intervals in boxes are specifically non-chromatic. All others are 'chromatic' or add colour to the scale. (Struve, 79)

Figure 2-4 : Contour lines of equal loudness for normal ears. Numbers on curves indicate loudness level in phons. 0-db = .000204 dynes per square centimeter. (Olson,67)

Figure 2-5 : Frequency and Intensity ranges of Speech and Music. The solid line depicts the boundaries of normal hearing, that is, the upper and lower limits of intensity and frequency. (Olson,67)

to a sound pressure of .000204 dynes per square centimeter.

The ear is most sensitive to the region between 3,000 and 4,000 hz. Contours of equal loudness are illustrated in figure 2-4. The limits of audible sound are illustrated in figure 2-5. The lowest contour of figures 2-4 and 2-5 is the threshold of hearing. The average normal ear cannot hear a tone below this intensity. The upper contour is the threshold of feeling. A tone with an intensity greater than this level will be painful.(Olson,67)

A good dynamic range for a synthesizer would be about 90 db. This allows a 50 db variation in intensity with a worst case signal to noise ratio of 40 db for the quietest signal.

In a Concert Hall environment soft (pianissimo) playing of a weaker orchestral instrument, for example a violin, flute, or bassoon, produces a typical sound level of 55 to 60 db. Fortissimo (loud) playing on the same instruments raises the db level to about 70 to 75. Louder instruments, for example trumpet or tuba, range in loudness from 75 db at pianissimo to about 90 db at fortissimo.

2.1.3 The Sound Envelope

Theoretically the growth and decay characteristics exhibited by most instruments can be approximated with exponential type functions. These functions affect the duration of a played note. Duration is the length of time that a tone persists or lasts. In music notation, duration is indicated by the type of note played (for example: whole, half, quarter) and the tempo. (the rate of movement of the music)

The growth and decay of tones depends on the type of generator or

Figure 2-6 : The Growth and Decay Envelopes for a Typical Organ, Piano, Guitar, and the sound 'Ah'. (Olson, 67)

instrument. In some instruments, such as the piano for example, the build-up time of a tone is short, while the decay time is long. In an organ, the growth and decay times of a tone are relatively long. (see Figure 2-6)

Growth and decay curves are part of the sound envelope which can be further broken down to include sustain and release components. Release refers to the point at which a note is released from play. Sustain refers to the portion of constant intensity persisting after the initial attack/decay until the point of release. Figure 2-7 illustrates cello and clarinet tones with their attack decay curves, time waveforms and frequency spectrums. Note the exponential nature of the envelope shapes.

For crude music synthesis a tone's fundamental component may be produced without all its harmonic components. (higher-multiple frequencies) If a proper sound envelope is used on these approximated components, then it will sound roughly like the instrument it is meant to represent. However a more impressive and accurate synthesis is accomplished by applying separate envelopes (determined by analysis) not only to a tone's fundamental component, but all its harmonic components as well. The envelopes depicted in figure 2-7 are actually a summation of all the envelopes of the given tone. (fundamental and harmonics) Accurate production of a tone, with its component envelopes is critical in synthesizers for full quality, or timbre. Nonlinear synthesis is a useful method for generating all the harmonic envelopes of a tone without recalling enormous amounts of data.

Figure 2-7 : Sound Envelopes, Waveforms and Spectra for a Cello (A) and Clarinet (B) tone. The upper plot shows the waveform of the entire note. The second plot shows a segment of the same waveform containing about three periods. The lower plot is the Discrete Fourier transform of an 80 ms segment in the middle of the tone. Note that the upper plot is divided into the attack, steady state (sustain), and decay segments. (Moorer, 77)

2.1.4 Timbre (Tone Quality)

Timbre is the characteristic of tone which depends on harmonic or partial structure. Timbre for all practical considerations is the dynamic tonal spectrum. Spectrum characteristics range from a pure tone, to pitchless sounds such as thermal noise, to the harmonic rich sounds of mechanical organ pipes. In all of these, timbre is the characteristic which enables the listener to recognize the kind of instrument which produces the tone. This is illustrated in Figure 2-8. We can see that the timbre spectrum varies with the note played, (see piano, guitar spectrums for different fundamentals) even on a single instrument.

Timbre also depends on the intensity of the played note. This is in part due to the fact that the ear is nonlinear and produces new overtones or alters the existing ones. For example when a pure tone of sufficient intensity is detected by the ear, a series of harmonics or overtones of the original frequency are heard. The sensation levels of the fundamental at which the harmonics generated by the ear first become detectable are shown in Figure 2-9. (Olson, 67)

It should be noted that:

'A partial is a harmonic if it has a frequency which is equal to the fundamental frequency (usually related to the pitch) multiplied by an integer. (1, 2, 3, 4,) Partial may be inharmonic if their frequency equals the fundamental frequency multiplied by a noninteger.' (Snell, 77)

Similarly an overtone is an acoustical frequency which is higher in frequency than the fundamental, but related to the fundamental in the same way 2nd and higher harmonics are. (Snell, 77)

A complete picture of timbre involves all the spectral components of the given tone, and their variations over the duration of play. This kind of

(A) Flute.

(B) Clarinet

(C) Guitar

(D) Piano

Figure 2-8 : Typical Spectra for tones of (A) Flute, (B) Clarinet, (C) Guitar, and (D) Piano. Frequencies indicated represent the fundamental pitch. (Olson,67)

Figure 2-9 : The level above threshold at which harmonics are generated in the ear at various frequencies. The numbers on the curves indicate the order of the harmonic. (Olson,67)

Figure 2-10 : Spectral (3-dimensional) plot of the Partial of a Clarinet tone. Note: partial 1 is the fundamental which in this case is 311 hz. This tone is actually e-flat above middle C. (Grey,75)

information is represented in Figure 2-10, which illustrates the spectrum of data for a 311 hz fundamental clarinet tone (E flat above middle C). The partials are observed to have a wide range of envelopes and may vary in frequency slightly over the duration of the tone.

It should be noticed that for a greater intensity of a given tone, the greater the number of partials generated by the instrument.

Timbre by itself is a very general term. Although it has been stated that timbre depends on the variations of partial content of a tone, nothing descriptive about this content is related by the word 'timbre'. Therefore because of the non-specific nature of the term 'timbre', several studies have been done to discover the components of timbre responsible for tone quality. The results of these studies are of fundamental importance since they reflect an understanding of how we perceive the differences and similarities between different complex instrumental tones.

A benchmark study in the field of timbre research, is the work completed by John M. Grey at Stanford University (Grey, 1975). This work explores the aspects of timbre in a quantitative fashion. Unfortunately Grey's results may not be universal because although several instruments were tested, only a single tone, at a single intensity and pitch E flat, was used to evaluate the properties of timbre. It is known that timbre can vary not only from instrument to instrument, but also to a certain degree from note to note for the same instrument.

For the purpose of ascertaining some facts about timbre, we will now review the highlights of Grey's work.

Grey summarized studies completed in the fifties and sixties, which attempted to apply verbal descriptions to timbre. Helmholtz (1954) applied

verbal labels to complex tones to describe such perceptual attributes as brightness, richness, and sweetness. Other studies confirmed the acceptance of three basic attributes for complex tones apart from pitch and loudness, namely, brightness, fullness, and roughness.

Brightness was defined to be a function of the location of the frequency scale midpoint of the energy distribution. Fullness was interpreted as being a function of the relative presence of odd or even harmonics. Roughness was found to be present in tones consisting of harmonics above the 6th, and was also a function of the location of these harmonics, in the whole sequence of higher partials.

In reference to the envelope characteristics of timbre, it has generally been found that the attack segment is the most important portion of the tone for its identification. In one experiment reviewed by Grey, better identification of a tone was possible with just 60 milliseconds of attack than for 150 milliseconds of steady-state. Further studies showed the durations of tone attack transients, varied with both the pitch and player for any single musical instrument, as well as varying from instrument to instrument.

As for timbre in relation to loudness, some studies have concluded that timbre is a very weak function of tone amplitude while others have concluded there is a close relation between timbre and amplitude. In general it can be observed that the harmonic richness of a tone varies with its overall amplitude level.

Early synthesis studies were done in the mid-sixties. A computer analysis and synthesis of the time-variant properties of trumpet tones was completed by Risset in 1966.

Grey summarizes:

"By using synthesis based on various models for simplifying the complex analyzed parameters of the sounds, Risset concluded three particular features were aurally important:

1. The relationships of the attack times of the harmonics, whereby successively higher harmonics take longer to appear and grow more slowly.
2. The fluctuation of the frequency, which is of small amplitude, fast, and quasi-random.
3. The harmonic content of the tone, which becomes richer in high-frequencies when the overall intensity increases."

Grey goes on to explain the significance of the analysis/synthesis method of timbre research, which is what he uses himself. He then summarizes four phases of his research. The only one of which we are concerned with is phase one, which gives us clues about data reduction and critical features of synthesized tones.

The analysis portion of Grey's study was completed on 16 different instruments, each played near the pitch of E-flat above middle C, approximately 311 hz. The tones were performed and recorded in an acoustical isolation chamber, which is a very dry environment. The tones were recorded and then digitized through a 14 bit analog to digital converter, at a sampling rate of 25.6 khz. Complex amplitude and frequency functions were derived with a heterodyne filter.

Resynthesis in phase one of Grey's work, was completed under four different conditions. Each was compared to the original tone:

- A) Direct resynthesis, from the digitized signal (referred to as 'Complex Synthesis') (Figure 2-11 A)
- B) Synthesis by line segment approximation of amplitude and frequency functions. (Figure 2-11 B)
- C) Synthesis using line segment approximation as well as exclusion of initial low-amplitude inharmonicity in the attack. This approximation was labelled 'cut attack'. (Figure 2-11 C)
- D) Synthesis using line segment approximation for envelope amplitude and constant frequencies for the harmonic frequency functions. This is referred to as the 'constant frequency' approximation. (Figure 2-11 D)

Grey explains that 16 'musically sophisticated' listeners were used for

(A) Complex Synthesis

(B) Line Segment Approximation

(C) Cut Attack

(D) Constant Frequency

Figure 2-11 : Grey's Resynthesis forms in Spectrographic plots. Note all plots represent the same tone which is that of the Clarinet at 311 hertz. The y-axis represents frequency (fundamental at the bottom), the x-axis represents time, while the bar thickness depicts loudness. The significance of the resynthesis terms is described in the text. (Grey,75)

phase 1 of his experiment. They were placed 10 feet from the speaker, for two ten hour long sessions, giving a total of 573 trials. The trials were structured in an AAAB discrimination paradigm, three tones of four being identical. The position of the different tone was randomly established. The listener was asked to distinguish tones in the discrimination paradigm as being either the same or different.

The results of this experiment are indicated in Figure 2-12. Note: "The collected data were averaged over the four repeated measurements per trial combination for each listener. The averaged discrimination score equalled the number of times the different pair (discrimination paradigm presented in pairs) was correctly identified plus $1/2$ the number of times there was no answer on a trial; this sum divided by 4".

Grey's conclusions were as follows:

1. The difference between complex synthesis and line segment approximation in terms of discrimination averaged about .63, indicating that line segment approximation is a hard case to distinguish. This infers that data reduction through line segment approximation is valid, although it does not include the complete microstructure of time variant amplitude and frequency functions.

2. The discriminability between the constant frequency approximation and both the complex synthesis and the line segment approximation suggest this degree of simplification is highly discriminable in many cases. (see figure 2-12 for detail)

3. The discriminability of the cut-attack condition is essentially 'parallel' and similar to that of constant frequency.

In nonlinear synthesis we primarily have constant harmonic frequencies throughout the duration of the tone. According to Grey such a simplification in tone synthesis would result in about an 86% discrimination probability. (discriminable from the original tone) It is now suggested that without a discrimination paradigm approach, the change in timbre may not be as noticable. However in evaluating the accuracy of the nonlinear

.675

.772

.627

.856

.769

.682

.845

.733

.650

.760

DISCRIMINATION

Complex Synth				
Line Segments				
Constant Freq				
Cut Attack				
	O	C	L	F
	r	o	i	r
	i	m	n	e
	q	p	e	q
		l	s	s
		e		
		x		

Figure 2-12 : Overall Discrimination Scores for various resynthesized tonal conditions, averaged over 16 instrument tones. The form of the matrices in terms of the pairs of the tonal conditions judged in the discrimination paradigm is given in the grid immediately above. (Grey, 75)

approach such a factor would have to be accounted for.

Although the individual harmonic frequencies may not be varied with nonlinear synthesis it is possible to vary all the tone frequencies in tandem by amplitude modulation, after introducing the nonlinearity. The difference in discriminability such a modification would cause when applied to nonlinear synthesis is unknown, and is suggested as a future topic of investigation.

Grey states also that certain instruments have inherently constant harmonics (such as the french horn) and one may conclude that such instruments would be the best candidates for nonlinear synthesis.

In this thesis we mostly investigate the use of nonlinearities to synthesize the oboe and clarinet, primarily because the spectral data for these instruments was readily available.

2.2 Synthesizers and the Theory of Music

Synthesis

2.2.0 Introduction and Overview

In evaluating the merits of music synthesizers there are two approaches to take. One is to consider how well they synthesize accepted and standard orchestral instruments. The other approach is to consider how well they produce new and stimulating tones. We are primarily concerned with the first approach, since the data structures that apply to such are already defined by the required harmonic content.

Most of the early analog synthesizers were quite capable of producing a range of unique and expressive tones but were incapable of duplicating already existing musical tones. However with today's more sophisticated synthesizer techniques it is feasible to attempt impersonation of fundamental musical instruments. Once this has been accomplished it is then possible to extend the simulation parameters to produce new and expressive tones.

In the sections of theory which follow we will begin exploring briefly the additive and subtractive synthesis techniques. This will be followed by a discussion of analog and then digital synthesis forms. These forms can use subtractive and additive techniques and as such the discussion may appear to be a little repetitive. It should be realized however that form and technique are distinctly separate categories, thus they are presented independently.

The section then deals with hybrid synthesis, a form which uses both

analog and digital circuitry. Finally modulation techniques are briefly introduced, in particular the use of FM and Nonlinear Functions.

The main concern of this thesis is to deal with the nonlinear synthesis technique. To evaluate its merits it will be compared with additive, subtractive and FM synthesis.

2.2.1 Additive and Subtractive Synthesis

The conceptually simplest approach, to music synthesis, is that of additive synthesis, developed in the 1950's and 1960's and elaborated on by James A. Moorer. In additive synthesis a musical tone is taken to be the sum of a set of harmonics, each of which varies in amplitude over time. Using this technique to mimic for instance the sound of a clarinet, the musician begins by recording a single note, say E above middle C. Then with a series of filters the amplitudes of the fundamental and harmonics over time, are extracted. A phase analysis of these harmonics reveals variation in frequency over time as well.

This leads to a general model in the additive technique which may be expressed for both the digital and analog domain:

$$\text{For the analog domain- } X(t) = \sum_{k=1}^M A_k(t) \sin(t(kw + 2\pi F_k(t)))$$

where

w = fundamental frequency in radians

k = harmonic number

$A_k(t)$ = amplitude of harmonic k at time t

$X(t)$ = the signal at time t

$F_k(t)$ = the frequency deviation of harmonic k
at time t .

M = number of harmonics

For the digital domain- $X(n) = \sum_{k=1}^M A_k(n) \sin(nT(kw + 2\pi F_k(n)))$
 where

$X(n)$ = the signal at time nT

n = the sample number

T = the time between consecutive samples

w = the angular fundamental frequency

k = the harmonic number

$A_k(n)$ = the amplitude of the harmonic k at
 time nT

$F_k(n)$ = the frequency deviation of harmonic k
 at time nT (assumed to be slowly
 varying)

Figure 2-13 illustrates a block diagram model of an additive synthesis type synthesizer. (Moorer, 1977) The numbers for additive functions can be made up or obtained through analysis. One way of estimating the partials amplitudes and frequencies is as follows:

$$a_k(n) = \sum_{r=n}^{n+N-1} X(r) \sin(rkw_0 T)$$

$$b_k(n) = \sum_{r=n}^{n+N-1} X(r) \cos(rkw_0 T)$$

$$\hat{A}_k(n) = \sqrt{a_k^2(n) + b_k^2(n)}$$

$$\hat{\theta}_k(n) = \text{atan} \left[\frac{a_k(n)}{b_k(n)} \right]$$

$$\hat{F}_k(n) = \frac{1}{2\pi} \frac{d\hat{\theta}_k(n)}{dt}$$

Figure 2-13 : Block Diagram of Additive Synthesis

Figure 2-14 : Extraction of Additive Synthesis Parameters

where:

ω_0 = angular frequency (should be set to $2\pi/NT$)

k = harmonic number.

$X(r)$ = input waveform at time rT

N = integer no. of samples in one period of the input waveform.

Note: hats on amplitude, angle, and frequency variables indicate these are estimates of the actual values and are not necessarily equal to the exact values that may have generated the tone. (see Figure 2-14 for diagram of this approach)

The forementioned equations are similar to those of Fourier Analysis. Note the use of differentiation of the sampled data function (digital domain). Also the fundamental pitch of the tone must be known and nearly constant over the interval of analysis. The usefulness of the equation relies on summation over one period to place a zero of transmission at the frequencies of the harmonics other than the one under analysis. If the fundamental frequency is chosen incorrectly, the frequencies of the zeros will not necessarily correspond to the frequencies of adjacent harmonics, thus care must be taken in selecting the fundamental. Studies show that this estimation method can tolerate no more than a 2% deviation error in the fundamental of the input signal. (Moore, 77)

The summation formulae for the $a_k(n)$ and $b_k(n)$ amount to filtering by averaging over one period.

The frequency of the harmonic is obtained by first getting the angle between the two sequences as computed by the arctan function, producing the PRINCIPAL VALUE of the phase angle. We can then differentiate the phase angle to get the frequency deviation. This deviation is the difference between the actual frequency and the harmonic $k\omega_0$.

Figure 2-15 : Piecewise Linear Fit of Timbre to Approximate a Clarinet tone at 311 hz. Note: the initial attack segments are still present. (Grey,75)

Figure 2-16 : Additive Synthesis Analysis data in Spectrographic form.

Note: same as the type of plot shown in figure 1-11.

x=time, with 1/10th second lines; y=frequency, with khz lines; width of bars= db relative to -40.

This tone is again a 311 hz clarinet tone digitized from a tape recording. The random dot pattern at the beginning and end of the tone are made by tape hiss. (Moorer,77)

The above process is used M times on the digitized sample tone, once for every harmonic.

The analyzed data may be displayed as illustrated in Figure 2-10 (Moorer, 1977). An alternative would be a spectrographic-line representation where each harmonic is represented as a bar. The thickness of each bar is proportional to the log of the amplitude of that harmonic and the center of each bar is located at the frequency of the harmonic at that time. (see Fig. 2-16)

The data as collected could be used to synthesize tones directly but when using an actual composition, one finds that the amplitude and frequency functions present a prohibitive amount of data. We then must find some means of performing data reduction. While doing this we are constrained by a few important factors. The generated tones must retain fidelity and must be very easy to modify.

One data reduction technique which has been quite successful is to fit amplitude and frequency functions with piecewise-linear functions. (Figure 2-15 shows the results of such a fit) Grey analyzed 16 instruments using this technique (see Timbre section 2.1.4) and used the results in psychoacoustic experiments. The results indicated that the synthesized tones represented by piecewise-linear functions were judged to be very close to the originals. (Grey, 75)

This analysis technique, as described, gives high quality but is only useful for isolated tones of near constant frequency. It is also required that the fundamental frequency be known a priori. (Moorer, 77) The analysis-based additive synthesis technique is however a powerful tool for computer music and psychoacoustics. By altering piecewise-linear parameters one can produce a wide variety of new and unusual tones. An

efficient data and control structure could be used with a microprocessor to implement this technique, controlling several oscillators in real time to produce the desired notes.

Snell (1977) investigated the use of sine look-up tables to generate 256 low distortion sine waves which could be used to synthesize 16 tones; each with 16 partials. This approach required a great deal of data for the individual partial envelopes and also required a great deal of dedicated hardware. The technicalities of the sin look-up table method (with table increment) will be covered in section 2.2.3. It is brought to the attention of the reader at this point that the nonlinear technique may be a simpler approach for synthesis since it uses much smaller tables of data and may even require less hardware.

Subtractive Synthesis

The subtractive ~~Synthesis~~ technique is closely related to the way many acoustic instruments work. It also mimics the way a human voice works. It starts with a generated harmonic rich periodic signal, containing energy at every frequency that is required for the specified tone. Filters then subtract out the unwanted frequencies. The filter's characteristics can be changed dynamically to cut off high frequency components as desired, or enhance any given set of frequencies with a high Q. An envelope generator is usually included before or after the filter to attenuate and control overall amplitude. In an analog system this necessitates using a voltage controlled amplifier fed with the desired amplitude envelope, typically lasting from .2 to 2.5 seconds. The greatest attraction of subtractive synthesis is that it is analogous to actual acoustic instruments and thus the

physics of the instrument can serve as a model for the synthesis structure. For example horns and woodwinds use the lips or a vibrating reed to generate the harmonic rich periodic driving signal. Various chambers and cavities in the horn and the horn shape itself forms the filter network. Similarly in speech, in the human vocal tract, the glottis vibrates to generate the basic pitch and the throat, mouth, and nose form the various filter cavities.

Digital implementation of subtractive synthesis begins with an excitation function, which is sent through a time varying filter. This process is similar to that used to synthesize speech.

Unfortunately, the standards by which music is judged tend to be much more rigid than those used for speech. The subtractive synthesis model has shortcomings which prohibit exact duplication of the original instruments. The problem is that although the spectrum can be modelled accurately with high order filters, there are inadequate tools for modelling the excitation function in a dynamic fashion. If a method could be found for either modelling more precisely the excitation or modifying the pitch and timing of the excitation, then it would be possible to achieve high fidelity synthesis. (Moorer, 1977)

2.2.2 The Analog Approach

Electronically generated music dates back three quarters of a century ago to the Telharmonium, an instrument developed by Thaddeus Cahill. The Telharmonium sounds were generated by over one-hundred alternators,

controlled by a keyboard and fed to subscriber's speakers over leased telephone lines. Also, early in the 20th century, experiments in electronic music were made in Germany by Fischinger and later in the 1930's, in the USSR, electronic music was produced by photoelectronic techniques, rather than by oscillators. In fact the development of electronic music has proceeded step by step with the development of technology. (telephone, loudspeaker, microphone, tape film sound track, etc.) (Kennedy,1985)

In the 1950's the composition of electronic works was more prevalent but slow because of the primitive equipment in the early studios. (Kennedy,1985) A composition consisting of a variety of prerecorded sounds could take hours to assemble on a final tape but would only last a few minutes. The complexity and novelty of the studio fabricated composition depends on the pattern of tape splicing, as much as the sounds originally recorded.

Synthesizer music did not become common until the late 1960's when the Moog synthesizer was developed. This version used analog circuits such as oscillators and filters and could be 'patched' in a variety of configurations. In today's analog synthesizers circuit elements have either voltage control or potentiometer control. Many voltage controlled elements use transconductance methods to allow control and variability in their configuration. The components which comprise a basic analog synthesizer are:

- a.the voltage controlled filters(VCF)
- b.the envelope generators(ADSR)
- c.the voltage controlled oscillators(VCO)
- d.the voltage controlled amplifiers(VCA).

Patching Configurations

The elements just listed can be 'patched' in a variety of configurations. Patching may be done with shielded cable and shielded jacks for audio signals, and banana plugs and non-shielded cable for control voltages. An alternative to pre-wired fixed patching is to feed the signals, ie audio inputs and outputs as well as control voltages, into a switching matrix which is composed of reed relays controlled by a digital control signal.

Figure 2-17 illustrates a patch configuration for nonlinear synthesis. Further details about the significance of the nonlinear configuration will be given in section 2.3.

Figure 2-17 : Patching Configuration for Analog Nonlinear Synthesis

2.2.3. The Digital Approach to Music Synthesis

Digital music synthesis techniques may be divided into two broad categories; hardware and software. Software generated signals are based on discrete time samples which may or may not be generated in real time, depending on the processor speed and processing complexity. According to sampling theory the generated signal must be sampled at least twice as often as the highest frequency being sampled. This assumes a mathematically perfect system but for practical applications a 20 khz signal (for example) should be represented by about 50,000 samples per second. Another consideration of software synthesis is dynamic range. If the quietest signal needs to have a signal to noise ratio of 40 db (corresponding to 7 bits) along with a 50 db dynamic range (corresponding to 9 bits) then the total dynamic range representation needs to be at least 90 db or 15 to 16 bits of data for each sample. (Alles, 1980) We now have the requirement of generating a 16 bit number every 20 microseconds. With suitable processor speed this requirement could be met for real time music generation. (Alles, 1980)

One may ask what the advantages of digital synthesis are that make it worth the effort. The answer is precision and control. The various synthesis parameters can be absolutely and repeatedly set to precise values.

We will now briefly look at some of the basics of digital music generation. The sinewave oscillator, the heart of the synthesizer, is usually comprised of a clock run phase generator and a sine look-up table. Figure 2-18(A) illustrates this technique. The following should be noted about the system in figure 2-18(A):

1. With 2^n samples in the sine look-up table, clock pulses are fed into the summer every $1/(2^n * f)$ seconds where f is the frequency of the desired sinewave.

2. The data in waveform memory can be a simple sinewave or any arbitrary combination of sinewaves.

3. By varying the input clock rate, the table output frequency can be varied. In general $f_{\text{sine(out)}} = f_{\text{clock}} / 2^n$.

A similar and probably more efficient technique is to use a sine look-up table with an address increment that steps through it. The bigger the increment, the smaller the frequency since fewer samples will represent the sine wave. For fractional increments interpolation between sin-table values may be used. However without interpolation the fractional part of the sum may be retained for successive summations to reduce error. The increment technique is further explored in section 4.3. Using the increment technique we find:

$$\text{Frequency out} = \frac{\text{Increment through Sinetable} * \text{Sample Rate}}{\text{Table Length}}$$

Note: the Sampling Rate is a constant, usually 32 khz or higher.

The increment/sinetable system is illustrated in Figure 18(B). The finite table length results in a slight error referred to as phase jitter. This jitter combined with quantization error accounts for the total signal error in the system. At the Stanford Center for Computer Research in Music and Acoustics, F. Richard Moore analyzed the jitter/quantization noise and found with a truncating oscillator (truncating a fractional increment), a Signal-to-Error Noise Ratio of 60.4 db was possible using a 4096 size sine

A. Digital Sinewave Oscillator with Phase Register (Alles, 80)

B. Digital Sinewave Oscillator with Variable Phase Increment (Olson, 67)

Figure 2-18 : Forms of Digital Sinewave Oscillators

Note: 1. $\text{Increment} = (\text{Freq. out} \times \text{Table length}) / \text{Sample rate (SR)}$

2. f_{clock} would have to be a two phase nonoverlapping clock.

3. For increment of 3, as illustrated for the sine look-up table frequency out of table = $(3 \times \text{SR}) / 2048$

Figure 2-18 (C) An expounded Illustration of a Digital Sinewave Oscillator with Variable Phase Increment.

table and 12 bits of quantization on each output sample. (Moore, 77)

Such digital sine oscillators may be realized with either software or dedicated hardware. Attenuation or amplification can also be realized in software using simple multiplication routines. Signal summing is accomplished by simply adding samples carefully so as to give proper attention to synchronization. Filtering in the digital domain is a little more complicated and amounts to calculating the results of difference equations using consecutive samples. The most popular filter is the finite impulse response type or 'transversal' filter. This structure is useful for producing transmission zeroes, to null out particular frequencies. The transversal filter can be realized in either hardware or software.

Further software manipulation of data samples can be done with a wide range of procedures or subroutines. For example frequency modulation could be accomplished using the FM equation. (see section 2.2.6; Modulation techniques) A sound envelope can likewise be produced using exponential equations with the appropriate time constants to yield attack, decay, and release curves. This along with a multiply routine would produce the envelopes for the tone harmonics or the overall tone itself. For faster routine execution the sound envelope could be antecedently generated and stored in memory.

As portrayed in section 2.2.1, additive and subtractive synthesis techniques are often used in the digital domain to generate music. With these and all other digital synthesis techniques the end product is passed to the analog world through a digital to analog converter. If the music is not generated in real time vast amounts of buffer memory may be required at this stage.

Specialized Digital Hardware Techniques

Digital hardware synthesis may be similar to the software approach (as with the use of sine look-up tables) but can differ as well.

The top octave generator exemplifies a different and unique hardware technique. A high frequency clock is divided down in frequency to generate square wave pulses at 12 different frequencies, corresponding to those spanned by an even tempered scale octave at the high end of the Chromatic scale. These twelve frequencies are in turn divided down by multiples of two to produce a full range of tones. With this technique only square waves are usually generated. However if the selected output is multiplied in frequency using a phase-locked loop then the resulting multiple frequency can be sent into a function look-up table resulting in a waveform at the necessary frequency. The phase-locked loop would generate a series of pulses which would access a function table such that the function would be output at the required frequency. The frequency from the PLL would have to be equal to the length of the function table, times the desired output frequency, assuming the functional table stores one period of the desired waveform. The only problem with this method is that a variable sample rate results from using the top octave generator to step through the function table and necessitates the use of special tracking filters to eliminate undesired sampling replication resulting from low sampling frequencies.

When the top octave generator is used, it requires a digital multiplexer to select the output desired. (see figure 2-19)

Figure 2-19 : Top Octave Generator, Application Diagram
(Gen. Inst. Cat.,1980)

NOTE : 2 CHANNELS FOR STEREO EFFECT

Figure 2-20 : Hybrid Music Synthesis; a Nonlinear Example

2.2.4 The Hybrid Approach

The hybrid approach represents a technique similar to that used in hybrid computing. A computer controller is interfaced with analog circuits to produce the desired synthesis. To generate the music a number of DAC interfaces are used, providing control voltages for analog building blocks. Sequences of music may be stored digitally through an Analog to Digital interface or the control voltages alone may be stored in memory. Amplifiers and attenuators may be made with Multiplying Digital to Analog converters (MDAC).

A simple hybrid system is illustrated in Figure 2-20. Two channels are illustrated in this figure to indicate a stereo output result or multiple (polyphonic) note capability.

2.2.5 Keyboard Interfacing

Digital techniques are most useful for good quality keyboard interfaces. Digital interfacing assumes that the keyboard is being interfaced to either a computer or hardware which will respond to digital signals.

A simple analog interface employs a voltage divider which generates a unique voltage for each note on the keyboard. These voltages are scaled in such a manner as to produce the appropriate frequency when fed into a VCO.

Problems arise however when more than one note is played at a time. Also the temperature dependent characteristics of the resistors in the

divider network can cause frequency drift.

Obviously a digitally encoded keyboard does not experience these problems. There are a number of ways digitally encoding may be accomplished but only one will be discussed here. The first is used primarily in monophonic systems, but can be incorporated in polyphonic systems with a few modifications. The keyboard consists of a matrix of SPST switches overlaid in usually an 8×8 pattern. The matrix is scanned with a high frequency clock, counter, and set of analog multiplexers as illustrated in Figure 2-21. The binary counter (4024) simultaneously addresses two 8 channel multiplexers which feed onto the 8×8 matrix of keyboard switches. When a matrix position is closed (by a down-stroke on the note playing keyboard) a positive voltage is transferred to resistor R4; if the corresponding position in the matrix is addressed. By scanning for downstrokes at a high speed, no notes are missed. The positive voltage at R4 causes the 'strobe' line to be active and may be used as an interrupt to signal a controlling processor that a note has been detected. A 'SCAN' line can be used to freeze clock action till the note address is picked up by the processor. The note data may be fed directly into a digital to analog converter to control a VCO or may otherwise be manipulated by a processor to generate a tone.

A more advanced keyboard may record the velocity or transition time of the downstroke key action, thus providing information on the intensity of the key-strike. This information may also be used by a processor to construct the desired output tone. Several notes may easily be played simultaneously via computer control. The tones may be generated in software or dedicated hardware, with the aid of additional circuitry.

Note:

1. Clock should be stopped when strobe line is activated, until keyboard data is 'collected'.
2. Counter (4024) should be reset to zero at end of count. Count ranges from zero to sixty-four.

Figure 2-21 : Scanning Keyboard Encoder (Simonton Jr., 1980)

2.2.6 Modulation Techniques

Most digital synthesis techniques which are neither subtractive nor additive, involve modulation in one form or another and are called modulation techniques or sometimes 'contrived' techniques. These techniques use some mathematical transformations which makes it computationally efficient to produce complex signals with varying partial frequency content. The most widely used modulation technique is Chowning's Frequency Modulation technique.

The frequency modulation model is summarized by the following equation.

$$S(t) = A(t)\sin(w_c t + I(t)\sin(w_m t)) \quad \text{--eq'n 2-1}$$

where-

w_c = carrier frequency

w_m = modulation frequency

$A(t)$ = time dependent envelope function

$I(t)$ = time dependent envelope function; specifying the modulation index

$S(t)$ = signal at time t

A typical FM system is illustrated in Figure 2-22.

In the FM system, if the carrier and modulation frequency are set equal, then the result is a series of harmonics at multiple frequencies of the carrier, which in effect is the fundamental. The harmonics have amplitudes specified by Bessel functions of the first kind and n_{th} order; $J_n(I(t))$, the argument of which is the modulation index. Using the expansion of equation 'A' we find:

Figure 2-22 : F.M. System; Block Diagram of Oscillator Structure (Alles,80)

Figure 2-23 : Bessel Function Plot of the first kind, taken with respect to Modulation Index (Stremmler,77)

$$\begin{aligned}
 S(t) = & (J_0(I(t))\sin w_c t \\
 & + J_1(I(t))(\sin(w_c t + w_m t) - \sin(w_c t - w_m t)) \\
 & + J_2(I(t))(\sin(w_c t + 2w_m t) - \sin(w_c t - 2w_m t)) \\
 & + \dots\dots\dots
 \end{aligned}$$

—eq'n 2-2

A plot of the Bessel function of the first kind taken with respect to the modulation index is shown in Figure 2-23. Note that inharmonic spectra can be produced if w_c/w_m is irrational. Unfortunately, because the partials can only vary according to the Bessel function as illustrated in figure 2-23, completely arbitrary synthesis is not possible. However by using carefully contrived modulation index envelopes, as well as overall amplitude envelopes and filters, closely approximated synthesis is possible. Chowning investigated primarily synthesis of Brass, Woodwind, and Percussive instruments.

Since there is no easy way to fit natural tones to a frequency modulation structure, FM synthesis requires empirical trying of time varying modulation indexes to fit the desired tone.

Nonlinear Distortion

The bulk of this thesis is concerned with the nonlinear distortion technique. The nonlinear technique involves distorting a fundamental frequency sine wave to create higher order harmonics, which will account for the timbre of a complex tone.

The nonlinear distortion is described by a polynomial equation which 'maps' a pure input sine wave onto a distorted output. The polynomial may be represented by a power series expansion incorporating multipliers and

Figure 2-24 : Distortion of a Cosine Wave with a Nonlinear Polynomial

adders, or more practically may be stored in the form of a memory look-up table which stores the appropriated distortion function. The general form of the polynomial distortion function is as follows:

$$y = f(x(t)) = a_0 + a_1 x(t) + a_2 x(t)^2 + a_3 x(t)^3 + \dots a_n x(t)^n$$

eq'n 2-3

A simple polynomial may be shaped as illustrated in Figure 2-24.

The nonlinear distortion technique is also referred to as waveshaping.

The significance and consequences of the nonlinear theory will be more fully explored in the section which follows.

2.3 Nonlinear Theory and Applications

2.3.1 Introduction to the Nonlinear Technique

We have seen in the previous section that the basic concept of the nonlinear technique is very simple. A predetermined distortion of a pure sine wave produces a required harmonic structure. However harmonics structures are usually time variant for natural instrument tones. It would be impractical to continuously change the nonlinear function over the duration of the tone to synthesize complex spectral changes. However the nonlinear synthesis method provides an alternative, specifically the use of the predistortion envelope which alters the amplitude of the initial sine/cos wave. This results in a wide range of spectral changes, occurring for a single polynomial function. A post distortion envelope completes the synthesis.

Historically waveshaping (the other common term for nonlinear distortion synthesis) has been around since the late 60's. In 1970, R. W. Schaefer presented the technique in an analog implementation (with diodes) (Schaefer,1970) . Also in 1970, Suen introduced the mathematical techniques which could be used to obtain the desired spectra, produced by nonlinear waveshaping.(Suen,1970)

Hutchin presented a clear and concise exposition of nonlinear theory in an analog context in 1976. Finally Arfib (1979), Lebrun (1979), and Beauchamp (1979), individually provided further insights into the usefulness of the distortion technique during the late seventies.

This section will begin with a look at basic nonlinear theory. Of

particular interest is the Chebyshev polynomial and how it relates the nonlinear function to the desired harmonic structure. This, along with extraction of partial envelope data, comprises the analysis portion of the analysis/synthesis succession. The discussion of Chebyshev polynomials leads to an overview of related matrix transformations (section 2.3.3).

The variable sine/cos amplitude approach or predistortion effect is examined in section 2.3.4. Finally nonlinear system layouts are presented in section 2.3.5.

This last section (2.3) completes the background theory. This theory will be used in nonlinear synthesis system design, data analysis, and system evaluation. A discussion towards the end of the thesis will also draw freely from this chapter.

2.3.2 Polynomial Distortion (Equations and Significance)

The following relation holds for the Chebyshev polynomial:

$$T_k(\cos \theta) = \cos k\theta \quad \text{---eq'n 2-3.}$$

where

T_k = Chebyshev polynomial of order k

$\cos \theta$ = input cosine waveform

Now suppose we have a required harmonic structure as follows:

$S(t)$ = signal at time ' t '

$$= h_0 + h_1 \cos \theta + h_2 \cos 2\theta + h_3 \cos 3\theta + \dots h_m \cos m\theta \quad \text{---eq'n 2-4}$$

where

$$\theta = \omega t$$

= fundamental angular frequency * time

Now we can represent equation 2-4 in terms of the relationship of equation 2-3, as follows:

$$S(t) = h_0 + h_1 T_1(\cos \theta) + h_2 T_2(\cos 2\theta) + h_3 T_3(\cos 3\theta) + \dots h_m T_m(\cos m\theta)$$

.....where $T_1, T_2, T_3, \dots$ are the Chebyshev

polynomials.:

$$T_0 = 1$$

$$T_5 = 16x^5 - 20x^3 + 5x$$

$$T_1 = x$$

$$T_6 = 32x^6 - 48x^4 + 18x^2 - 1$$

$$T_2 = 2x^2 - 1$$

$$T_7 = 64x^7 - 112x^5 + 56x^3 - 7x$$

$$T_3 = 4x^3 - 3x$$

.....and so on

$$T_4 = 8x^4 - 8x^2 + 1$$

We will now proceed with an example, taken from LeBrun (LeBrun, 79). Suppose $h_1=9, h_2=3, h_3=5, h_4=7$, and $h_5=1$. Then the following polynomial will be necessary to generate the specified harmonic structure from the fundamental:

$$\begin{array}{rcl} 9T_1(x) & = & 9x \\ 3T_2(x) & = & 6x^2 - 3 \\ 5T_3(x) & = & 20x^3 - 15x \\ 7T_4(x) & = & 56x^4 - 56x^2 + 7 \\ +T_5(x) & = & 16x^5 - 20x^3 + 5x \end{array}$$

$$\begin{aligned} f_1(x) &= 16x^5 + 56x^4 - 50x^2 - x + 4 \\ &= 16(\cos \theta)^5 + 56(\cos \theta)^4 - 50(\cos \theta)^2 - \cos \theta + 4. \end{aligned}$$

Scaled by a factor of 25, this nonlinear function appears as shown in Figure 2-25. Note that the order of the highest harmonic required, dictates the order of the polynomial. (in preceeding example the order is 5)

The example just presented assumes that the input cosine to the

Figure 2-25 : A Polynomial Curve.

The equation for this curve is given as:

$$f_1(x) = 16x^5 + 56x^4 - 50x^2 - x + 7$$

See example, section 2.3.2 (LeBrun, 79)

nonlinear function has a constant maximum amplitude of 1. If however it does not, then a variety of changes occur in the amplitudes of the harmonics. This phenomenon is discussed further in section 2.3.4.

2.3.3 The Use of Matrix Transformations

The harmonic to polynomial transition may be done most efficiently with a matrix transformation. The reverse process (polynomial to harmonics), may also be accomplished with matrix transforms.

Arfib (Arfib,79) gives us the form of the forward transform, (harmonic to polynomial), as follows.

One may define:

$$S(t) = f(x(t)) \quad ; x(t) = a(t) \cos wt$$

$$S(t) = h_0 + a(t)h_1 \cos wt + a(t)h_2 \cos 2wt + \dots a(t)h_n \cos nwt$$

$$f(x) = d_0 + d_1 x + d_2 x^2 + d_3 x^3 + \dots d_n x^n$$

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} (2/a)^0 \\ (2/a)^1 \\ (2/a)^2 \\ \vdots \\ (2/a)^n \end{bmatrix} * B * \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} \quad \text{eq'n 2-5}$$

(Arfib,79)

where B is derived from the definition of the Chebyshev polynomials and can be generated in the following way:

The first line of B =

$$1 \quad 0 \quad -2 \quad 0 \quad +2 \quad 0 \quad -2 \quad 0 \dots\dots\dots$$

and the other lines are computed according to the recursive relationship-

$$B(i,j) = B(i-1, j-1) - B(i, j-2)$$

Using this definition we find that for a matrix of rank 10 B is :

$$\begin{array}{cccccccccc}
 1 & 0 & -2 & 0 & +2 & 0 & -2 & 0 & 2 & 0 \\
 & 1 & 0 & -3 & 0 & 5 & 0 & -7 & 0 & 9 \\
 & & 1 & 0 & -4 & 0 & 9 & 0 & -16 & 0 \\
 & & & 1 & 0 & -5 & 0 & 14 & 0 & -30 \\
 & & & & 1 & 0 & -6 & 0 & 20 & 0 \\
 & & & & & 1 & 0 & -7 & 0 & 27 \\
 & & & & & & 1 & 0 & -8 & 0 \\
 & & & & & & & 1 & 0 & -9 \\
 & & & & & & & & 1 & 0 \\
 & & & & & & & & & 1
 \end{array}$$

All values to the left of the diagonal are equal to zero.

For the reverse process, ie. finding harmonic amplitudes given polynomial coefficients, the following is true:

$$\text{given } x(t) = a(t) \cos \omega t$$

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} = 2 * A * \begin{bmatrix} d_0 * (a/2)^0 \\ d_1 * (a/2)^1 \\ d_2 * (a/2)^2 \\ \vdots \\ d_n * (1/2)^n \end{bmatrix} \quad \text{--- eq'n 2-6}$$

where A is defined as follows: $A(1, 1) = 1$

$A(i, j) = A(i-1, j-1) + A(i+1, j-1)$, for i not equal to 1

$A(i, j) = 2 * A(2, j-1)$, for i equal to 1

From the matrix definition on the preceding page, we would find an A matrix of rank 10 would be as follows:

Figure 2-26 : Dynamic Timbre Analysis: Fitting with Nonlinear Function

1	0	2	0	6	0	20	0	70	0
	1	0	3	0	10	0	35	0	126
		1	0	4	0	15	0	56	0
			1	0	5	0	21	0	84
				1	0	6	0	28	0
					1	0	7	0	36
						1	0	8	0
							1	0	9
								1	0
									1

(Arfib,79)

Using the transforms of equations 2-5 and 2-6 one can do the following:

1. Find the nonlinear polynomial structure (PS1) for the harmonic spectra at time t_1 . (using equation 2-5)
2. Try to fit 'PS1' to the harmonic spectra at time t_2 , by varying the 'a' or cosine amplitude (this would be done with a predistortion envelope)

This procedure is illustrated in Figure 2-26. The timbre spectra must be determined in advance in the same way they were determined for the additive synthesis technique. (see section 2.2.1) This requires digitizing the tones under analysis, and extracting their harmonic amplitudes using digital filter techniques.

2.3.4 Tone Synthesis using the Nonlinear Technique with a Predistortion

Envelope

The waveshaping synthesis technique can be enhanced by using a variable amplitude cos function, determined as outlined in Figure 2-26. Additional enhancement is possible by using amplitude modulation on the output to generate inharmonic partials. Such effects are required for natural percussive sounds such as bells or drums. (Lebrun, 79)

Now to mathematically determine how the amplitude of the initial cos function affected the harmonic amplitudes. Knowing that:

$$f(x) = h_0 / 2 + \sum_{k=1}^{\infty} h_k T_k(x)$$

where $x = \cos \omega t$

Expressing the shaping function in terms of the cosine amplitude:

$$f(ax) = h_0(a) + \sum_{k=1}^{\infty} h_k(a) T_k(x)$$

Using the specific example of : $f_2(x) = 2x^2 + x - 1$ —eq'n 2-7

Therefore: $f_2(ax) = 2a^2x^2 + ax - 1$ —eq'n 2-8

Now one can reverse the Chebyshev construction process by subtracting out successive harmonics. We note that $f_2(x)$ contains no harmonics higher than the second because the expression contains no power of x greater than the second.

Now expressing our shaping function as a power series in x :

$$f(ax) = \sum_{p=0}^{\infty} d_p(a) x^p \quad \text{—eq'n 2-9}$$

In the case of the specific values of equation 2-7;

$$d_2(a) = 2a^2, d_1(a) = a, \text{ and } d_0(a) = -1.$$

Expressed in this way, one can proceed to convert the polynomial of degree M into a sum of Chebyshev polynomials. (Lebrun, 79). The algorithm is as follows. First we initialize a variable 'Q' to $f(ax)$, then subtract from this Q the expression $h_m(a)T_m(x)$ where:

$$h_m(a) = q_m(a) / 2^{m-1}$$

and

m = current maximum exponent in Q

$q_m(a)$ = corresponding coefficient of m .

If $m > 0$, we continue the process with the resulting Q, where successive Q's are determined as follows;

$$Q_{\text{next}} = Q_{\text{previous}} - h_m(a)T_m(x)$$

By unwinding the successive subtractions we can see that the $h_m(a)$ are in fact the correct harmonic amplitudes in the Chebyshev expansion of $f(ax)$. (Lebrun, 79)

Now proceeding with the example of equation 2-8:

$$Q = f(ax) = 2a^2x^2 + ax - 1$$

$$h_2(a) = q_2(a) / 2^{2-1} = 2a^2 / 2 = a^2$$

$$\begin{aligned} Q &= Q - a^2 T_2(x) = (2a^2x^2 + ax - 1) - (2a^2x^2 - a^2) \\ &= ax + (a^2 - 1) \end{aligned}$$

Now $m = 1$, $q_1(a) = a$, hence:

$$h_1(a) = q_1(a) / 2^{1-1} = a / 1 = a$$

$$Q = Q - a T_1(x) = (ax + a^2 - 1) - ax = a^2 - 1$$

.....and finally:

$$h_0(a) = q_0(a) / 2^{0-1} = (a^2 - 1) / (1/2) = 2(a^2 - 1)$$

Checking the expansion of f_2 by adding the terms which have just been calculated:

Figure 2-27 : Harmonic Amplitudes as a Function of $\cos$ Amplitude for:

$$E_2(ax) = 2a^2 x^2 + a - 1; x = \cos \theta$$

(LeBrun, 79)

Figure 2-28 (A) : Block Diagram of the use of the Nonlinear Function Table in an Analog System. Note: System needs several latches and Sample/Hold's to work properly.

Figure 2-28 (B) : Block Diagram of a total Digital Nonlinear System. Note: f_{c2} is envelope regulating clock. Also, several latches are required in this system too.

$$\begin{aligned}
 f_2(x) &= h_0(a)/2 + \sum_{k=1}^{\infty} h_k(a)T_k(x) \\
 &= 2(a^2-1)/2 + aT_1(x) + a^2T_2(x) \\
 &= a^2 - 1 + ax + 2a^2x^2 - a^2 \\
 &= 2a^2x^2 + ax - 1
 \end{aligned}$$

The harmonic amplitudes, if plotted as a function of 'a', would appear as shown in figure 2-27.

A complex algorithm may be devised to determine the $h_m(a)$ functions for a given polynomial, and the results can be plotted. A much easier method however, for plotting the $h_m(a)$'s for any polynomial, would be to use the transformation of equation 2-6, for a range of 'a' values. If however one wishes to use an algorithm which determines the mathematical expressions for the $h_m(a)$'s, one may refer to: LeBrun, 1979, published in the Journal of Audio Engineering.

2.3.5 Nonlinear System Layouts

The form of the nonlinear system has is presented in the functional block diagrams of figures 2-28.

Typically the nonlinear block is in reality a look-up table. The input to the table is scaled to range from -1 to +1 while the table itself consists of anywhere from 32 to 4096 memory addresses which 'map' a nondistorted sinusoidal wave onto a distorted waveform. The more the number of memory words in the table, the more precise the distortion and resulting harmonic structure.

Since the look-up table is stored in digital form, an analog system

Figure 2-29 : Restrictions of the Diode in the Nonlinear Application

Note: System may be realized in Analog or Digital Form.

Figure 2-30 : Nonlinear Approach Using Polynomial Expansion

would require an Analog to Digital conversion, followed by a memory look-up and Digital to Analog conversion. (see Figure 2-28(A)) A system starting with an already digitized sine wave can be interfaced directly to the waveshaping table. (see Figure 2-28(B))

An all analog system is possible but highly impractical since it would most likely incorporate diode functions and as such does not have the capacity for negative line slopes, as is often required in the distorting polynomial curve. (see Figure 2-29)

We note that it is possible to generate the nonlinearity by using the expanded form of the polynomial. This requires a large number of multipliers and is inherently inaccurate because of the magnitude of the polynomial coefficients which may vary over several orders of magnitude for polynomials of order 5 or more. The expanded polynomial technique is illustrated in Figure 2-30. Note that the system may be analog but would be much more accurate if its components were digital.

If more than one sine wave is to be distorted at one time it can be done with one waveshaping table if Time Division Multiplexing is used. Otherwise, if sinewave signals are summed and then run through the nonlinear function, intermodulation distortion will result.

CHAPTER 3.0 TIMBRE DATA ANALYSIS

3.0 Introduction

This chapter discusses the methods used to extract data from timbre curves given in the literature for the Oboe and Clarinet. The timbre curves for these instruments were from a 311 hertz fundamental, which is the tone e-flat above middle C.

The first task was to determine sets of polynomials that could generate the harmonic amplitude curves provided. Polynomials were extracted in 26 time frames across a time base which ran from 0 to approximately 3.4 seconds. Curves were fitted with various degrees of polynomials ranging from 5th order to 10th and 14th order. A FORTRAN routine SEQGEN was implemented to test the validity of these polynomials. In this routine the polynomial was used to transform a single period 311 hz cosine wave with 128 to 512 samples. A Discrete Fourier Transform (DFT) calculated the frequency content of the wave after it had been transformed.

The results of this program indicated significant differences in spectra resulted when different numeric resolutions were used. It was concern over these differences that led to an even more sophisticated nonlinear simulation, namely the program called 'LLSIMULATE'.

LLSIMULATE has a main routine and 30 subroutines. It simulates the phase increment stepping through a cosine table as well as nonlinear lookup with pre and post distortion multiplication. The details of this fortran routine are covered in Chapter 5.

Data reduction as introduced in section 2.3.3 is done by a program LSTSQFTT. Given 26 polynomial equations, one for each time frame of data,

LSTFIT determined which of the 26 time frames fit the other 25 with the least error. The fit was accomplished by optimising the pre and post multiplier curves to give the best results. From this analysis, 26 time frames of polynomials (some 364 numbers) could be reduced to a predistortion curve, a postdistortion curve and a single nonlinear curve (66 numbers of curve data). Using these rough approximations one can appreciate how much the data can be reduced with only one nonlinearity to work with. The rest of this chapter discusses in greater detail the analysis just briefly overviewed. This leads into a look at specific timbre generating systems which will be covered in chapter 4.

3.1 Polynomial Curves

The original LSTSQFIT routine was used to plot polynomial data for a visual representation of a nonlinearity. The polynomials were extracted using the theory presented in Chapter 2.3. This theory implements a matrix transformation technique based on Chebyshev polynomials. The transformation is 2-way meaning one can transform from harmonic to polynomial data or vice versa.

To simplify the process of graphical representation of data 6 roughly evenly spaced time frames were chosen for polynomial analysis. These were:

column 2: at .6 seconds into envelope
 column 5: at .9 seconds into envelope
 column 10: at 1.4 seconds into envelope
 column 15: at 1.9 seconds into envelope
 column 20: at 2.4 seconds into envelope
 column 25: at 2.9 seconds into envelope
 (see Fig. 3-1)

Each of these frames had a corresponding polynomial plotted with 14 originating harmonics, 9 originating harmonics and 4 originating harmonics. Consequently the orders of the resulting polynomials were 14, 9 and 4. The plots of these polynomials are shown in Figure 3-2. These plots all correspond to a portion of the timbre of a 311 hz fundamental clarinet tone.

Some general observations may be made upon inspection of these curves. Typically the higher order curves (14th order) have much more detail, showing more 'valleys' and 'hills' in their outline. The curves appear symmetrical to a large degree. (about the origin) Also one can see that one may fit a single 9th order curve to other 9th order curves with pre and

Figure 3-1 : Clarinet Timbre for 311 Hertz Fundamental

Figure 3-2 (A) : Nonlinear Curves fit to Clarinet Tone at 311. Hertz with time from 0 to 3.0 seconds. Note: an order 14 polynomial is fit to the various harmonic structures at the times indicated.

Figure 3-2 (B) : Nonlinear Curves fit to Clarinet Tone at 311 Hertz with time from 0 to 3.0 seconds. Note: an order 9 polynomial is fit to the various harmonic structures at the times indicated.

Figure 3-2 (C) : Nonlinear Curves fit to Clarinet Tone at 311 Hertz with time from 0 to 3.0 seconds. Note: an order 4 polynomial is fit to the various harmonic structures at the times indicated.

post multiplication and with less error than the same fittings for 14th order curves. This is observed by the fact that columns 10, 15 and 20 (times 1.4, 1.9 and 2.4 seconds into envelope) have similar shapes (see fig. 3-2), and the 9th order curves are closer in form than the 14th order curves. However since 14th order polynomials generate 14 harmonics as compared to only 9 harmonics with 9th order curves we restrict the bulk of our analysis to higher order polynomials as these will produce a richer sound.

The oboe curves generated have a much different shape. They are not symmetrical in any way. This is probably because the oboe is a double reed instrument with a different resonating cavity than the single reed clarinet. A comparison of the harmonic envelopes of the clarinet and oboe shows their timbre structures to be quite different. (see figs. 3-1 and 3-4) Notice that unlike the clarinet, the oboe has even harmonics almost as large as the odd harmonics. The data is normalized so the largest amplitude corresponds to one unit. The time base spreads from 0 to 3 seconds in both plots. The curves in figures 3-1 and 3-4 are generated by straight line interpolation over 26 time frames.

The polynomial of figure 3-3 was taken .9 seconds into the oboe envelope. This roughly occurs at halfway down the decay portion of the harmonics just after the initial attack.

More discussion on the significance of the nonlinear curves and their corresponding harmonic structures will follow in the rest of this chapter, specifically the waveforms generated by polynomial transforms and the spectrum resulting from these waveforms will be considered. Ideally a generated spectrum would be identical to the originating harmonic structure.

The extracted polynomial coefficients for 14th order clarinet and

Figure 3-3 : Nonlinear curve fit to Oboe Tone at .9 seconds into Envelope. Note: this is a fit to an E-flat tone above middle C.(311 hertz fundamental)

Figure 3-4 : Oboe Timbre for a 311 hz (E-flat above middle C) Tone.

oboe are presented in the Appendix. Although the coefficients have large values the overall polynomial value rarely exceeds 4, for a normalized harmonic data structure. This is because successive positive and negative coefficients cancel each other.

3.2 Nonlinearly Generated Waveforms

When one generates a multi-harmonic waveform by using a nonlinear look-up table with a large number of points one gets a smooth wave as illustrated in figure 3-5. This wave was generated from the nonlinearity illustrated in figure 3-2, part A, at .9 seconds into the tone. It was plotted by the SEQGEN (sequence generator) routine. The waveform originated from a 1024 sample cosine wave distorted by the polynomial of figure 3-2. In a nonlinear system realization, the nonlinearity is a lookup table and the input is sinusoidal in nature with probably no more than 12 bits of precision.

Calculation of Sample Rate

If one has a constant sample rate then the number of samples per period is:

$$\text{No. of Samples} = \frac{\text{SR(Sample Rate)}}{\text{Frequency}}$$

The piano keyboard dictates the frequencies of the fundamental waveforms. If the highest note on the keyboard is taken as the highest fundamental frequency in the synthesizer, one could calculate a reasonable sample rate as follows:

$$\text{Highest fund. frequency} = 4186$$

$$\text{No. of samples} = 30 \text{ (for reproduction of 15 harmonics)}$$

Figure 3-5 : The Generation of a smooth Nonlinearly formed waveform from 1024 samples. (at .9 seconds into Clarinet 311 hertz tone; 6-flat above middle C)

$$\text{Sample rate} = 30 * 4186 = 125,580 \text{ hz.}$$

Aliasing

Note the cutoff frequency for hearing is around 18 kilohertz. Thus the number of samples could be limited so that only the first 4 or 5 harmonics were produced. With 5 harmonics we would need 10 samples per period or:

$$\text{Sample Rate (SR)} = 10 * 4186 = 41,860$$

Unfortunately higher harmonics would replicate back to distort the frequency content below 18 khz, in the example just given. Figure 3-6 shows how distortion results from higher harmonics aliasing back into lower frequencies.

To illustrate the problem of having too few samples in a waveform for the number of harmonics we are trying to reproduce consider the computer runs shown in table 3-1, using 16 and 32 samples per fundamental period. The DFT results for these sample sizes are presented. The polynomial of figure 3-2 (at .9 seconds) was used with a fundamental of 16 and then 32 samples per period. The original table for the cosine wave had 1024 intervals as does the nonlinearity. Note that there is a lot of error in the 16 sample result, but little with 32 samples. This 16 sample result is due to aliasing with a polynomial that was generated from a 14 harmonic tone. The waveforms generated from 16 and 32 samples are illustrated in figure 3-7.

The 311 hertz frequency was chosen for the above testing because

Table 3-1

Dft Results: 311 Hz Waveform, 10 Bit Lookup Nonlinearity, 16 and 32 samples per Period.....9 seconds into Tone

Harmonic	Frequency (Hertz)	16 Samples per Period	32 Samples per Period	Orig. Harmonic Structure
0	0	.0091525	.019553	0
1	311	1.0011	1.0033	1
2	622	.15769 *	.015222	.025
3	933	.688641 *	.26834	.2637
4	1244	.14561 *	.071629	.079
5	1555	.89654 *	.8469	.842
6	1866	.10357 *	.04894	.04736
7	2177	.25934 *	.20867	.2105
8	2488	.32034 *	.16017	.1579
9	2799	#	.05067	.0526
10	3110	#	.05472	.0526
11	3421	#	.04960	.0526
12	3732	#	.07398	.0711
13	4043	#	.4203	.421
14	4354	#	.1427	.1316
15	4665	#	.00282	.01400
16	4976	#	0.00	0.00

Note:

* - too big because of replication
- calculated as reflection of first
8 samples with 16 bit dft.

2

STROMBERG-CARRISON
MICROD
A DIVISION OF STROMBERG-CARRISON

Figure 3-6 : Harmonic Distortion due to Replication from low Frequency Sampling.

Figure 3-7 : Effect of a Small Number of Samples on Waveform Shape.
311 Hertz Clarinet Tone at .9 seconds. 16 samples (above)
32 samples (below).

the harmonic data was taken from a 311 hz fundamental tone. The frequency results are in fact relative and would apply to any fundamental with the same harmonic structure.

Spectral Testing of Waveforms

Further Spectral testing was done completed with LLSIMULATE. This Fortran routine tested different number systems, used to generate the multi-harmonic waveforms. The test frequency was 220 hz and the sample rate was 131,072 hz. This means each period of the test waveform had 595 points. Oboe spectral plots and waveforms are illustrated in Appendix A. Clarinet spectral plots and waveforms are illustrated in Figure 3-8. These waveforms were derived using a cos lookup table with 4096 samples and a 10 bit number system. The DFT analysis was done with 256 points. These points were aligned with the 595 sample points for a legitimate 256 point transform. LLSIMULATE is further used (as seen in chapter 5) to compare the signal-to-error noise results from linear and log numeric sampling.

Figure 3-8 (A) Dft results with 256 points, for waveform as illustrated below. (Clarinet tone, 311 Hertz, .5 seconds into envelope)

Figure 3-8 (B) Dft results with 256 points, for waveform as illustrated below. (Clarinet tone, 311 hertz, .9 seconds into envelope)

Figure 3-8 (C) Dft results with 256 points, for waveform as illustrated below. (Clarinet tone, 311 hertz, 1.4 seconds into envelope)

Clarinet at 1.4 seconds into tone.

3.3 Extracting Timbre fit Information

The program used to extract post and pre-distortion envelope data is called "LSTSQFIT". The purpose of LSTSQFIT is to fit a timbre structure with only one polynomial, extracted from a single time frame of harmonic data. By varying the amplitude of the cosine or sine wave entering the nonlinear look-up, we can vary widely the harmonic structure and with proper care can approximate a close timbre fit to the desired tone over all time frames. The predistortion multiplier is the cos amplitude while the postdistortion multiplier is the overall output amplitude function. Both pre and post distortion values take the shape of a continuous envelope typically starting at zero, taking on some determined shape, and then ending near zero.

Figure 3-9 shows how, with a varying "a" (cos amplitude) value, one can get a wide range of varying harmonics when reforming the timbre using a fixed polynomial structure. The equation for Fig. 3-9 is:

$$\begin{aligned}
 f(ax) = & -.06835 + 2.9975(\text{acos}\theta) + 8.0861(\text{acos}\theta)^2 - 26.1165(\text{acos}\theta)^3 \\
 & - 147.97(\text{acos}\theta)^4 + 185.88(\text{acos}\theta)^5 + 966.64(\text{acos}\theta)^6 - 697.68(\text{acos}\theta)^7 \\
 & - 2946.92(\text{acos}\theta)^8 + 1265.742(\text{acos}\theta)^9 + 4562.63(\text{acos}\theta)^{10} \\
 & - 1081.033(\text{acos}\theta)^{11} + 3479.0(\text{acos}\theta)^{12} + 352.567(\text{acos}\theta)^{13} \\
 & + 1036.962(\text{acos}\theta)^{14}
 \end{aligned}$$

$\cos\theta = x$

(see table A-1, in Appendix)

After choosing columns 2,5,10,15,20 and 25 of the 26 columns (time

Figure 3-9 : Harmonics as a Function of Cos Amplitude. (Polynomial values chosen from 1.4 seconds into original envelopes)

frames) of data we have, each of these columns was fitted to the other to see which produced the smallest overall error. This was completed by the following procedure which was incorporated into LSTSQFIT:

1. Calculate polynomials for 26 time frames and the selected number of harmonics. (up to 14)
2. Select one time frame (TF1) which will be used with a pre and post distortion multiplier to fit the other time frames. (TF2)
3. Vary the cos predistortion multiplier on TF1 and regenerate approximated fits to other time frames by the varying cosine amplitude.. "a". (both source, TF1, and target, TF2, harmonics are normalized)
4. Compare the normalized results of TF1 with varying "a" with other normalized (original) harmonic time frames (TF2)
Calculate the least square error between corresponding harmonics and sum for the time frame.
5. The time frame which fits other time frames with the least total square error, is chosen for the synthesis process and using normalizing data pre and post distortion multiplier curves are generated. The best fitting time frame is chosen by comparing fits of time frames 2,5,10,15,20, and 25 to each other.

Whereas the pre-distortion multiplier varies harmonic content, the post-distortion multiplier varies the overall amplitudes of all harmonics together, thus giving a timbre its overall amplitude envelope. The above process results are summarized in tables 3-2 and 3-3, which are based on the 311 hz tone of the Clarinet. Once the best fit is found for the initial 6

Table 3-2 : Crossfitting polynomials to several time frames using best 'a' value as found by successive incrementation and least square fitting.

IPIT.(TF2) Destination	ICOL.(TF1) Source	Least Square Sum (Sum of all Harm.)	Cos Amplitude (Optimized)
5	2	.4358083	.42
10	2	.1688318	.40
15	2	.1454327	.40
20	2	.1091742	.42
25	2	.03111943	.58
SUM = .8900			
2	5	.1064627	.12
10	5	.1734265	1.00
15	5	.1706426	.46
20	5	.1292018	.46
25	5	.04713948	.22
SUM = .626873			
2	15	.0692325	.12
5	15	.1946734	.92
10	15	.0052922	1.0
20	15	.030358	1.0
25	15	.01500824	.52
SUM = .31451			
2	20	.0974689	.36
5	20	.3106217	1.00
10	20	.0378349	1.00
15	20	.030354	1.00
25	20	.0405971	.54
SUM = .51687			

Table 3-3 Fitting of time frame 15 to all other 26 time frames. Note the program LSTSQFIT determines the best "a" value for lowest least squares sum in each time frame. The post distortion multiplier values are relative and would probably be scaled to lower values in the final realization.

Time Frame	1	2	3	4	5
"a" value	.96	.12	.34	.94	.92
Least Square Sum	.1762806	.06923	.072309	.153147	.19467
Post Dist. Mult.	.1635028	3.36154	3.198	5.725563	8.81918

Time Frame	6	7	8	9	10
"a" value	.92	.92	1.00	1.00	1.00
Least Square Sum	.0576	.036566	.02134	.09942	.00529
Post Dist. Mult.	9.747515	9.63147	4.1002	4.000212	3.95023

Time Frame	11	12	13	14	15
"a" value	1.00	1.00	1.00	1.00	1.00
Least Square Sum	.0037838	.003115	.00114	.000246	---
Post Dist. Mult.	3.950229	3.9002	3.9002	3.8502	3.8002

Time Frame	16	17	18	19	20
"a" value	1.00	1.00	1.00	1.00	1.00
Least Square Sum	.000548	.003039	.00569	.0203	.03036
Post Dist. Mult.	3.8002	3.6502	3.4502	3.400197	3.2002

Time Frame	21	22	23	24	25
"a" value	1.00	1.00	.54	.54	.52
Least Square Sum	.04166	.06760	.060244	.024308	.015008
Post Dist. Mult.	3.10018	2.80016	3.50367	2.9697	2.6212

Time Frame	26				
"a" value	.50				
Least Square Sum	.015899				
Post Dist. Mult.	2.30315

Figure 3-10 : Least Squares Fit Analysis Results: 26 time frames over 3 seconds, 311 Hertz Clarinet Tone. Data from Table 3-3. Note: broken lines are areas of interpolation. Fitting is done with column 15 (1.9 seconds into envelope) polynomial data.

columns of data, the result is generalized to all 26 time frames. Table 3-3 and Figure 3-10 illustrate the fitting of time frame 15 to all other time frames. From Figure 3-10 one sees that the best fit occurs during the sustain portion of the envelope. Also the worst error in fit occurs near the beginning and peak of the attack. The extrapolation on the two upper curves (end of pre and post distortion curves) is an approximation entered because the data was only available for the first 3 seconds into the envelopes. The first point on the predistortion curve was interpolated to a near zero value because it would otherwise cause discontinuity problems which could affect the attack envelope integrity. The high error for this point indicates that a different value such as that interpolated (near zero) may not cause too much concern.

The next chapter will discuss an overview of three harmonic generating systems with nonlinearity, only one of which will be chosen because of its accuracy and practicality.

CHAPTER 4: SYSTEM PROPOSALS AND OVERVIEWS

4.0 Introduction

Any nonlinear synthesis system begins with a sinusoidal wave which should be distortion free and pure in nature. Since this thesis generates harmonics using a digital lookup table in digital memory and not with analog components then one must either start with a digitally represented sine wave or use an analog wave with an analog-to-digital conversion.

There are three possibilities for generating the sinusoidal wave for the nonlinear approach:

1. Use a frequency divider to generate a pulsed frequency which feeds a counter which in turns steps through a function table or sine table (see Fig. 4-1A, Phase Generator Approach)
2. Generate an analog sine wave and use an analog to digital convertor to get a sample train which can be fed into a nonlinear look-up table. Pre and post distortion envelope multiplication may be done with a multiplying digital-to-analog convertor. (MDAC) (see Figure 2-28 A)
3. Use a phase counter to step through a sine table, the output of which is digitally multiplied by a predistortion envelope. This feeds into a digital nonlinear table and then a post distortion multiplication. (see Figs. 2-28B and 4-1B)

The details of operation were worked out for all three above approaches. A choice was eventually made to implement approach 3 because

7

Figure 4-1 (A) : Nonlinear Timbre Generation with divide-by-N circuitry to produce desired frequency.

Disadvantages:

- requires tracking filter to cut out low freq. overtones from low frequency sampling
- requires a lot of circuitry to generate phase count.

Figure 4-1 (B) : Nonlinear Timbre Generation with variable increment phase through sine table.

Disadvantages

- requires large sine table for accurate freq. generation by variable phase increment. (however memory is relatively cheap)
- lots of circuitry is required to generate phase sum with round-off. (however these are simple components with straightforward layout)

it had a constant sample rate and did not involve much analog circuitry.

4.1 The Phase Generator Approach

This approach relies on taking a high frequency clock at 4.25216 Mhz and dividing it down to the 12 semitones of the high octave end of the piano keyboard. These 12 frequencies are in turn divided down by factors of two, producing all the lower octaves. This process is shown in Figure 4-3.

Once a pulse is generated at the desired frequency it is multiplied by the number of samples in the sine table it looks into and then fed into a counter which counts through the sine table.

Since a variable phase interval is used, low frequency overtones which are undesired will result in some cases. For example, choosing to generate a 27.5 hz signal with the number of samples (NSAMP) equal to 32, our sample rate becomes 880 hz. Assuming a sample and hold action on this sample rate, it would be necessary to filter out replicated samples from 880 hz down to 495 hz. To avoid replication at low frequencies one can use a sine table with more entries or use a high order low pass tracking filter. Considering the first option, ie more samples, if 1024 samples per period are chosen then the new sample rate would be:

$$\begin{aligned}\text{Sample Rate (SR)} &= \text{NSAMP} * \text{Frequency} \\ &= 1024 * 27.5 \\ &= 28,160\end{aligned}$$

Replication would not occur in the audio range with a sample rate of 28,160 so there would not be any need for filters. This is illustrated in figure 4-2 A.

Continuing to step through a sine table with 1024 entries, ascending

the tonal scale, there is limitation from high frequency sampling. For example, if the frequency is 440 hertz:

$$\begin{aligned} \text{SR} &= \text{NSAMP} * \text{frequency} ; \text{let frequency} = 440 \text{ hz.} \\ &= 1024 * 440 \\ &= 450 \text{ khz.} \end{aligned}$$

.....which is approaching the limit of the sampling frequency, due to timing restrictions on memory access, multiplication and digital to analog conversion. However there can be a second sine table with only 256 samples for frequencies at 440 hz and above:

$$\text{SR} = \text{NSAMP} * \text{frequency} = 256 * 440 = 112,640$$

(see Fig. 4-2 B)

By increasing the frequency to 1760 hz our sample rate becomes:

$$\begin{aligned} \text{SR} &= \text{NSAMP} * \text{freq.} = 256 * 1760 \\ &= 450.56 \text{ khz.} \end{aligned}$$

(see Fig. 4-2 C)

By using three sine tables of decreasing size, one can obtain a tolerable sample rate and still avoid replication distortion without the use of elaborate filters.

A system which uses this approach of separate sine tables is depicted in figure 4-3. Note that a phase-lock loop multiplies the square wave fundamental by the number of samples in the sine table so one has a resulting clock with output feeding a counter. The counter, feeds into the sine table, yielding a sampled sine wave at the desired fundamental frequency instead of simply an on and off pulse. The output of the phase-lock loops could also be used to step through a function table which would be a sine and nonlinear table combined. The disadvantage of this

Figure 4-2 (A) : Spectra with 440 hz Fundamental, 1024 Samples and 14 harmonics.

Figure 4-2 (B) : Spectra with 440 hz Fundamental, 256 Samples and 14 harmonics.

Figure 4-2 (C) : Spectra with 1760 hz Fundamental, 256 Samples and 14 harmonics.

Figure 4-3 : Nonlinear Timbre Generation with divide-by-N circuitry for producing desired frequency. Note: multiple table lookup can be used to prevent aliasing and other error.

system is that it does not have a predistortion envelope multiplication. It would thus not be able to generate a wide range of timbre spectra.

In the final analysis the phase generator approach was abandoned because of the complication of using either switched capacitor tracking filters or multiple table look-up. These elaborate convolutions were a direct result of the sampling rate being variable.

4.2 The A to D Approach

In this approach, one uses a constant sample rate and a VCO to produce the sine wave input. Actually any analog sine wave source could be used. However the sine wave must be distortion free and of a predetermined and constant amplitude. A VCO with log frequency scaling can take a linearly stepped input voltage and output a 12 semitone scale with several octaves if tuned properly.

When two or more sine waves are summed and fed through the nonlinear table via an analog to digital convertor, a fair amount of intermodulation distortion occurs. One should thus remember that for the analog sinewave approach, only a single pure sinewave can be used by the nonlinear table unless it is time division multiplexed.

One can now determine what the sampling rate would have to be for proper analog to digital conversion:

Highest fundamental frequency = 4186 hz

No. of harmonics = 14

Highest frequency of interest = $14 * 4186 = 58,604$

Thus to avoid replication one would have to chose a sampling frequency as follows:

$$SR = 20\text{khz} + 58,604 \text{ hz} = 78,604 \text{ hz}$$

$$T_g (\text{Sampling Period}) = 12.7 \text{ microseconds}$$

If one is to time share a look-up table amongst various sine waves one finds:

$$T_g \text{ minimum} = 500 \text{ ns}$$

$$\text{Number of units time sharing table} = \frac{12.7 \text{ usec}}{.500 \text{ usec}} = 25$$

.....thus the system would be feasible with a number of sine waves being sampled and time-division multiplexed through the nonlinear table. This system however was not selected because it was believed that an all digital system would be simpler to implement and would provide a more accurate form of nonlinear synthesis.

4.3 Sine Table with Increment Approach

The most feasible method, eventually realized, was the sinetable with increment approach. This method stems from the use of a precise increment value which is used to step through a long sine table.

When stepping through the sine table the fractional part of the increment is not seen but is retained in the overall phase sum to reduce the cumulative phase error which would have resulted from integer round-off.

The 'phase sum', attained from consecutive addition of the

increment, is the value addressing sine table memory. This sum must be able to reach 4096 (size of the sine table as derived later in this section). Therefore it must have 12 integer places as well as a number of fractional places.

Using the equation introduced in section 2.2.3 one finds:

$$\text{Increment through the sine table} = \frac{\text{Frequency desired} * \text{Table length}}{\text{Sample Rate}}$$

Or.....

$$\text{Frequency out} = \frac{\text{Increment} * \text{Sample Rate}}{\text{Table Length}}$$

NSAMP = number of samples per period, previously calculated to be
a minimum of 30.(section 3.2)

$$\begin{aligned} &= \frac{\text{Sample Rate (SR)}}{\text{Frequency Out (FO)}} \\ &= \frac{\text{Table Length (TL)}}{\text{Increment (Inc)}} \end{aligned}$$

The highest fundamental frequency out is 4186 hz. With this:

$$\text{NSAMP} = 30 = \frac{\text{SR}}{4186}$$

$$\text{SR} = 125,580$$

(for practicality a value divided down from a 20 Mhz clock..156,250 hz
is chosen)

Now having determined the sample rate one can determine the table
length required:

$$30 = \frac{\text{Table Length}}{\text{Increment}}$$

Using 7 integer bits for the increment..... $2^7 = 128$ (maximum)

$$\text{TL} = 30 * 128 = 3840$$

Rounding up to a power of 2 : $TL = 4096$ or 2^{12}

On an 88 key piano :

$$f_{low} = 27.5 \text{ hz}$$

$$f_{high} = 4186 \text{ hz}$$

which gives:

$$Inc_{low} = .720896$$

$$Inc_{high} = 4186 * 4096 / 156,250 = 109.733$$

With 16 bits used for the increment value, one can use 9 for fractional increment and 7 for integer increment. This means one can get fine resolution on the increment; ie. as small as $2^{-9} = +/- .0019531$

The sine table is 4096 long which infers that we must have a phase sum with 12 integer bits of address. Thus the total number of bits on the phase sum is $12 + 9$ (fractional part retained) giving 21 bits in all.

This increment system is depicted in figure 4-4.

The Sine Table with Increment method proved the best choice for a nonlinear music system because of its all digital form and the use of a constant sample rate. In the next chapter a simulation of the complete nonlinear system is covered with both logarithmic and linear sampling, along with appropriate analysis. This leads to the final design parameters for a system realized in hardware.

Figure 4-4 : Phase addition for phase step through sine lookup table.
 Note: the number of bits in each stage are selected for least error in frequency. The sine table in the next stage would be 4096 words long.

CHAPTER 5: SIMULATION OF A NONLINEAR SYSTEM

5.0 Introduction

The nonlinear look-up table, sine/cosine lookup table along with pre and post distortion multiplier was thoroughly simulated using log and linear sampling, in the mainframe computer routine "LLSIMULATE". This fortran program contained some 1800 lines of code and comments consisting of the following:

1. Chromatic scale and fundamental frequency calculations, with computed phase increments to generate a range of frequencies by stepping through a sine table.
2. Cos table definition in log or linear form, with a variety of bit patterns to test accuracies of log and linear sampling approaches.
3. Nonlinear table generation (after reading nonlinear table coefficients) with input range from plus one to minus one. Specified number systems are used for both input and output of table data.
4. System test, beginning with generation of one period of the frequency being tested (Must be above or equal to 220 hz). After one period of a cos wave is produced it is predistortion multiplied by a multiplier constant, fed into the polynomial then aligned with Dft sample points. (128, 256 or 512) The number systems defined in 2 and 3 are used.
5. A Dft is computed on the test string of data and the results are presented in table or plot form.
6. The program interacts with the user to determine if a repetition of any analysis is required with different data. If so, branching

to the appropriate program code is done.

The simulation program has some 30 subroutines which cover some 58 pages. It is unique from other routines developed (SEQGEN and COMPROUT) in that it defines the number systems used first, then later steps through the lookup tables, defined with these number systems, to generate a string of data for Dft testing. It is also unique in that it can be used to test logarithmic sampling with 6 to 11 bits in integer and fractional form. The log bits are assumed to have a negative sign since in all cases they represent numbers from 0 to 1.

By testing linear and log number systems with the same number of bits it is possible to determine which is most efficient for use in the nonlinear system. Testing was done with a 220 hz tone which was fed into a quadratic polynomial doubling the fundamental frequency. By completing Dfts on sample strings it was possible to determine which systems had the best signal-to-error noise ratio.

The log system had an advantage in that pre and post distortion multiplication was possible with an addition. However to get the final sampling in linear form requires either a large log to linear lookup table or a sophisticated D-to-A conversion with antilog circuitry.

5.1 Simulation flow chart with Program Summary

The flow chart in figure 5-1 is complicated by the names of all the subroutines called upon to complete its execution. The purpose of the subroutines is indicated in detail in the listing included in an internal

Figure 5-1 : Simulation Program Flow Chart.

Simulation flow chart cont.

Simulation flow chart (cont.)

departmental report from Carleton U. Dept. of Electronics (Armstrong, 86). Each subroutine is one to three pages long and may call on other subroutines to complete its designated task.

Worth special mention are the scaling routines used in both log and linear systems. They calculate, bit position by bit position, the on and off bits in say an 'n' bit representation. The log algorithm for calculating on and off bits is much more complicated than the linear approach. In a linear system we take a number to be scaled and compare its magnitude to that of the most significant bit to determine if the most significant bit should be set. This process is completed with consecutively smaller bits, subtracting bits from the total when they are set.

In log scaling, we compare products instead of sums. The bits set are used in a base two exponent, a two being raised by the power of the bit-on being tested. A running multiplier is kept which must be lower than the error on the least significant exponential value. When the multiplier product total is less than ~~this~~ error value then the bit of the exponent being tested is set on. The multiplier product is the value being scaled multiplied by the exponentiation of the logarithm bit being tested. Thus if our value was .7 and the first on bit was the $1/2$ place in the exponent then the multiplier product would be $1.414 \times .7 = .99$ which is less than our maximum limit of say 1.09 ($2^{1/8}$). Therefore the new multiplier product becomes .99 and further least significant bits are tested to see if they should be set on or off. This process is further explained by the code given in routine 5, in the LLSIMULATE listing, given in the internal report mentioned above. (Armstrong, 86)

Note that a large portion of programming code is devoted to plotting in the simulation routine. Plotting is possible for the following functions:

1. Cosine table in log or linear form.
2. Normalized Nonlinear curve in linear form.
3. Logarithmic nonlinear curve in log.log, linear.log, or
linear. linear form.
4. Sampled data string before alignment with Dft points.
5. Sampled data string as represented by Dft points.
6. Dft results in the following linear and log plots.

Figure 5-2 (A) : Nonlinear Function from logarithmic number system.
 The scales are linear and the log number system is 5,4
 (5 integer, 4 fractional) before and after the nonlinearity.
 Note the step size is large at the extremities of the function
 so accuracy and resolution is compromised.

Figure 5-2 (B) : Sample String from quadratic of 5-2(A). Samples are represented by a 5,4 log system. This illustrates a theta to two-theta mapping through a quadratic nonlinearity. Note the large step sizes present near a magnitude of one.

Figure 5-2 (C) : Discrete Fourier Transform results from a 256 point dft on the sample string of Figure 5-2(B). Note that this results in about a 39 db signal to noise error ratio.

Figure 5-2 (D) : Discrete Fourier Transform results from a 9 bit linear sampling system. With a theta to two-theta mapping we have about 60 db signal to noise error ratio, as seen by the spectral lines of this illustration.

8 BIT SYSTEM*				10 BIT SYSTEM*				
NO. SYS.	3,3	2,4	LINEAR 7 PLACES	4,4	3,5	2,6	LINEAR 9 PLACES	
DYNAMIC RANGE	20x LOG $\left(\frac{1}{2^{-7 \frac{1}{8}}}\right)$ = 47.41dB	20x LOG $\left(\frac{1}{2^{-3 \frac{1}{16}}}\right)$ = 23.71dB	42.14dB	20x LOG $\left(\frac{1}{2^{-15 \frac{1}{16}}}\right)$ = 95.95 dB	20x LOG $\left(\frac{1}{2^{-7 \frac{3}{32}}}\right)$ = 47.98dB	20x LOG $\left(\frac{1}{2^{-3 \frac{63}{64}}}\right)$ = 24 dB	54.19 dB	
12 BIT SYSTEM*				16 BIT SYSTEM*				
NO. SYS.	4,6	3,7	2,8	LINEAR 11 PLCS.	5,9	4,10	3,11	LINEAR 15 BITS
DYNAMIC RANGE	20x LOG $\left(\frac{1}{2^{-15 \frac{63}{64}}}\right)$ = 92.24 dB	20x LOG $\left(\frac{1}{2^{-7 \frac{127}{128}}}\right)$ = 48.12 dB	20x LOG $\left(\frac{1}{2^{-3 \frac{255}{256}}}\right)$ = 24.06 dB	66.23 dB	20x LOG $\left(\frac{1}{2^{-31 \frac{511}{512}}}\right)$ = 192.65 dB	20x LOG $\left(\frac{1}{2^{-15 \frac{1023}{1024}}}\right)$ = 96.23 dB	20x LOG $\left(\frac{1}{2^{-7 \frac{2047}{2048}}}\right)$ = 48.16dB	90.31 dB
8 BIT SYSTEM				10 BIT SYSTEM				
NO. SYS.	3,4	4,3	LINEAR 7 BITS	4,5	5,4	3,6	LINEAR 9 BITS	
DYNAMIC RANGE	20x LOG $\left(\frac{1}{2^{-7 \frac{1}{16}}}\right)$ = 47.79 dB	20x LOG $\left(\frac{1}{2^{-16 \frac{1}{8}}}\right)$ = 95.58 dB	42.14 dB	20x LOG $\left(\frac{1}{2^{-15 \frac{1}{32}}}\right)$ = 96.14 dB	20x LOG $\left(\frac{1}{2^{-31 \frac{1}{16}}}\right)$ = 192.28 dB	20x LOG $\left(\frac{1}{2^{-7 \frac{63}{64}}}\right)$ = 48.07dB	54.19 dB	
12 BIT SYSTEM				16 BIT SYSTEM				
NO. SYS.	4,7	5,6	6,5	LINEAR 11 BITS	6,9	5,10	7,8	LINEAR 15 BITS
DYNAMIC RANGE	20x LOG $\left(\frac{1}{2^{-15 \frac{127}{128}}}\right)$ = 96.28 dB	20x LOG $\left(\frac{1}{2^{-31 \frac{63}{64}}}\right)$ = 192.56 dB	20x LOG $\left(\frac{1}{2^{-63 \frac{31}{32}}}\right)$ = 385.13 dB	66.23 dB	20x LOG $\left(\frac{1}{2^{-63 \frac{511}{512}}}\right)$ = 385.30 dB	20x LOG $\left(\frac{1}{2^{-31 \frac{1023}{1024}}}\right)$ = 192.65 dB	20x LOG $\left(\frac{1}{2^{-127 \frac{255}{256}}}\right)$ = 770.61 dB	90.31 dB

Table 5-1 (A) : Dynamic Ranges of Log and Linear Binary Number Systems.

Note: maximum value for all systems is one.

* WITH BIT POSITION RESERVED FOR UNDERFLOW.

NOTE: ALL NUMBER SYSTEMS HAVE ONE BIT POSITION RESERVED FOR SIGN.

Table 5-1 (B) : Absolute Accuracies of Logarithmic and Linear Binary Systems

Number System Type		Zero Step	Accuracy Near Zero	Accuracy Near One
Gen'l	Specific			
8 bit	3,4 Log	.00408	$\pm .00018$	$\pm .042$
8 bit	4,3 Log	$1.66 \times 10^{*-5}$	$\pm 1.5 \times 10^{*-6}$	$\pm .083$
8 bit	Linear 7	.0078	$\pm .0078$	$\pm .0078$
10 bit	4,5 Log	.0000156	$\pm 3.0 \times 10^{*-7}$	$\pm .021$
10 bit	5,4 Log	$2.43138 \times 10^{*-10}$	$\pm 1.076 \times 10^{*-11}$	$\pm .042$
10 bit	3,6 Log	.0039	$\pm .000043$	$\pm .011$
10 bit	Linear 9	.00195	$\pm .00195$	$\pm .00195$
12 bit	4,7 Log	$1.53 \times 10^{*-5}$	$\pm 8.33 \times 10^{*-8}$	$\pm .0054$
12 bit	5,6 Log	$2.3537 \times 10^{*-10}$	$\pm 2.563 \times 10^{*-12}$	$\pm .011$
12 bit	6,5 Log	$5.5397 \times 10^{*-20}$	$\pm 1.21 \times 10^{*-21}$	$\pm .021$
12 bit	Linear 11	.000488	$\pm .000488$	$\pm .00048$

Note:

1. Single star means multiplication
Double star means exponential
2. All log systems retain an implied negative sign indicating negative exponent. The absolute value will be between zero and one.

eg.: 3,6 Log would be as follows:

$$2^{-111.111111}$$

The bit position to indicate negative overflow is not retained so no zero value can be represented.

Table 5-2 : Distortion test Results for Log and Linear Number Systems

Note: tests 1 to 11 for θ to 2θ nonlinear transform
 test 12 for θ to $0,1,2,3,4,5,6,7,8,9\theta$ transform
 test 13... θ to 2θ nonlinear transform

Test No.	Test Number System	Signal to Error Noise Ratio
1	7 bits linear ✓	34 db
2	log: 3,3 before and after nonlin.	34 db
3	log: 4,4 before NL, 3,3 after NL	39 db
4	log: 4,4 before and after nonlin	39 db
5	log: 4,3 before and after nonlin	34 db
6	9 bits linear	59 db
7	log: 5,4 before and after nonlin	39 db
8	log: 5,5 before and after nonlin	47 db
9	log: 5,6 before and after nonlin	52 db
10	log: 5,6 before; 7,4 after nonlin	49 db
11	log: 5,6 before; 7,8 after nonlin	52 db
12	11 bits linear (θ to $0, \theta, 2\theta, \dots, 9\theta$)	50 db
13	11 bits linear (θ to 2θ)	65 db

Note also Signal to Error noise ratio derived from
 256 point dft.

Figure 5-3 (A) : Nonlinear Transform plot of 0 to $\cos 2\theta + 2\theta + \dots$
 θ -theta, all of equal magnitude. Note the horizontal axis is
input and the vertical axis is output. Magnitudes are scaled
so both axes have a maximum of one.

Figure 5-3 (B) : Waveform after passing a single period 220 hz signal through the nonlinearity of 5-3 (A). Note the amplitude is shown to lie between +1 and -1. The number system used for sampling is 11 bits linear.

Figure 5-3 (C) : Discrete Fourier Transform results after passing a period of 220 Hz signal through the nonlinearity of 5-3 (A). Notice about a 48 dB signal to error noise ratio occurs. An 11 bit linear sampling system was used in this waveshaping test.

Figure 5-4 : Discrete Fourier Transform results.
 Theta to 2-theta transform, 200 hz
 fundamental, 11 bits sampling, 256
 point Dft.

5.2 Test Sequence Results on Log-Linear System Comparison

To assess the accuracy of the log system there are a few calculations initially completed which summarize a variety of bit system resolutions near zero and near one. Resolutions at these quantities were established since they define the values of most interest in the sample system we are using. Dynamic range values for different sampling systems were also established. (tables 5-1A and 5-1B)

The logarithmic number systems were tested with LLSIMULATE. The resulting accuracies were represented by plotted Dft results. The test results are summarized in ~~table 5-2~~.

Simulation; Summary of Results..

Tests 1 to 12 use a 256 point Dft, a test frequency of 220 hertz and 1191 samples. The sample rate was chosen to be 262,144 hz for these tests. The signal-to-error noise ratio is the displacement spectral components other than generated 2ω (meaning twice the original frequency) have in relation to the 2ω component. The 0 to 2ω nonlinear mapping was completed with a standard quadratic polynomial. Specifically $2X^2 + X - 1$. Distortions beyond that expected occurred in the log representation of the quadratic because of the step size near +1 and -1 and awkward perturbations in the nonlinear function which occurred as a result of the large step size. This is illustrated by the nonlinear plot of figure 5-2(A) and the resulting sample string for one period of a 220 hz signal.(fig. 5-2(B)) The Dft results for this test are illustrated in figure 5-2(C) and show a signal-to-error noise ratio

of about 40 db. The 5,4 log system results just described (5 integer bits, and 4 fractional bits) can be compared to a linear system with 9 bits, which gives a signal-to-error ratio of about 60 db.(fig. 5-2(D))

Looking at table 5-2 one sees that a large number of log systems were tested and they did not produce very good results. Some of the tests used a different log system for before and after the nonlinearity. Some tests used just one number system throughout. Clearly for the same number of bits it appears that the linear system has better signal-to-error ratio. The second last test on table 5-2 was test #12. This transformation generated harmonics from dc through to 90 from a single 0 to test resolution for complex nonlinearities. The nonlinearity for this mapping is displayed in figure 5-3A while the waveform and spectral results are shown in figs. 5-3B and 5-3C .

A final test, shown in table 5-2 (no. 13) was a 0 to 20 transformation, using 11 bits sampling with a resulting Signal-to-error noise ratio of about 65 db. (see fig. 5-4). This was in fact the number system chosen for hardware system design since it had an excellent noise rejection for only 11 bits per sample.(10 fractional and 1 sign)

5.3 Discussion of Simulation

The LLSIMULATE routine was large and costly to run. Some 900 resource units in all were consumed during development and testing of this routine. The later part of the simulation, the output of the nonlinear table, had double-precision variable storage which for large arrays, consumed a lot of memory. In the whole routine many arrays were used because the simulation used tables in memory, the same way a table would be used in hardware implementation. The table input addresses were calculated in lookup based on the data value being used for table input. The Discrete Fourier Transform used double precision-arithmetic.

At least half of the simulation program dealt with log values although in conclusion it was decided log values should not be used for the sampling number system. Of special mention is the fact that if log values were implemented an antilog table or antilog hardware would have to be used to convert the output back to linear form. This would have been acceptable if the log sample system displayed good signal to error noise ratio, but as seen earlier there is an unavoidable problem with sample steps and resolution.

The dynamic range is much greater when more integer log bits are used. This is clearly the case since more integer bits in the exponent cause a smaller value to be obtained, enabling greater dynamic range. (maximum value for all cases is one) Note that in both the log and linear bit systems, one bit is reserved for the sign. Also in the exponential representation of a number, it may be necessary to reserve a bit position for negative overflow, since there is no other representation for a zero value. Hence if for

instance we are using an 8 bit system we may signify the exponent with 6 of these 8 bits or:

- A. have 6 bits for exponent representation, say 3,3 type
such as 111.111
- B. have 1 bit for the sign of the number represented,
assuming the exponent sign to always be negative since
we are working with numbers of magnitude one or less.
- C. have 1 bit for signifying a zero value from negative
overflow.

In the final analysis, the number systems tested did not allow for the last bit described above. (C. value) This was thought to be acceptable because the difference between zero and the zero step (value closest to zero) ~~was~~ almost negligible as seen by table 5-1A. Hence a zero value would be represented by the smallest amplitude the log number system could represent.

As shown in table 5-1A, if we have a log system with accuracy near zero, we trade off fractional exponential bits to lose accuracy near the maximum amplitude of one.

We note that had a greater accuracy near zero been required then the log system would have been more effective but it was decided that because of the complex nature of the waveforms and their generation through a nonlinear table, logarithmic representation would not be feasible. This is a result of the logarithmic nonlinearity function not having small enough step sizes near one. Normally this would be acceptable since the error near one would be proportionally constant to the larger magnitude value near one. This companding effect however does not work very well when one has to distort through a nonlinearity. The resulting waveform

would have samples near zero which could transform to larger output function values but now with a large signal error. This process could work in reverse to. Looking at figure 5-2A we see that for a value close to zero on input we have a output transform step of about .042. Even though the input values may be small in step size the amplitude of the corresponding output resolution makes fine accuracy throughout, prohibitive. This is further illustrated by the large step size in the middle of the waveform of figure 5-2B. The upper half of the waveform has much fewer sampling steps than the lower half. This causes severe reductions in the signal-to-error noise ratio in the final output.

It may have been worth while however to mix log and linear systems by having a linear input to the nonlinear table and a logarithmic output. This would allow addition instead of multiplication for post distortion multiplication. However an antilog conversion would again be necessary.

CHAPTER 6: NONLINEAR SYSTEM HARDWARE AND SOFTWARE

6.0 Introduction

The hardware used to implement a nonlinear waveshaper has already been partially specified. Chapter 3 gave data for the pre and post distortion envelope curves, for the simulation of a clarinet tone. Chapter 4 gave the specifications of bit lengths for the increment and phase registers. Chapter 5 explained the 50 to 60 db signal-to-error noise ratio obtained if one used a sine-table and nonlinear-table with 11 linear bits.

This chapter will explain how all the above factors are integrated into the design of the nonlinear system. It begins with an overview of the system components in block diagram form and then explains in detail the timing and synchronization, which is controlled by specialized hardware. This leads to an explanation of the chromatic scaled keyboard interface as well as the output interface, which delivers the signal to an audio amplifier. This chapter also describes the EPROM programming required for the following:

- A) increment memory values for all fundamental frequencies
(64 words)
- B) sine wave table values (2048 words)
- C) nonlinear table values (2048 words)
- D) predistortion envelope multiplier values (256 words)
- E) postdistortion envelope multiplier values (256 words)

All EPROMs used were type 2716 or equivalent. All the above values were programmed using two EPROMs each, since 10 to 16 bit words were required.

6.1 An Overview of Signal Generation Hardware

The Signal Generation Hardware is the circuit which generates a complex tone dynamic waveform, given a specified fundamental frequency address. The fundamental frequency address is taken from a keyboard scanner built into the chromatic scale keyboard. The scanner data includes a signal indicating a new note has been played along with the address of that note.

The address of the note being played was used by the 6809 microprocessor to determine the increment memory value which would increment through the sine table giving us the desired fundamental frequency. Since the address of the scanner did not correspond directly to the note addresses in increment memory, the 6809 microcomputer calculated the appropriate address which was in turn sent out to the Signal Generation Hardware.

The Signal Generation Hardware had the following signals going into it:

- start envelope signal
- stop envelope signal
- address of the played note, to be looked up in increment memory.

The following signals come out of the Signal Generation Hardware:

- sign bit (one means negative analog quantity)
- analog sample value at 156250 hertz

The whole of the signal generation system is illustrated in diagrams

6-1,A,B,C. The bulk of the hardware is shown in figure 6-1 A, with the exception of the phase generator (figure 4-4). The block diagram of figure 6-1A illustrates the clock phase timing that was used to generate the complex waveform signal. A two phase nonoverlapping clock was the principal timing element. It ran at a frequency of 156250 hz ($20\text{Mhz}/2^7$). In addition to the phi 1 and phi 2 clocks were two shorter pulses which occurred during phi 1. These pulses were designated phi 1A and phi 1B and were used to regulate timing in the multiplier on-chip latches. These pulses are present but not illustrated in figure 6-1A.

The master clock as illustrated in figure 6-1B (20 Mhz) is the source of all the timing pulses. Four counters divide down the 20 Mhz to provide the envelope timing as well as the pipeline timing. The pulse timing for one sample cycle is illustrated in fig.6-1B in the upper right hand corner. The gates in the lower right hand corner of 6-1B illustrate generation of the basic pipeline timing signals from the counter A output.

The 76.2939 hz signal was obtained by division from the 20Mhz master clock and was used to regulate the envelope output for both pre and post distortion. The output of the envelope address counter was synchronized with the phi 1, phi 2 timing by use of a large nand gate as illustrated in figure 6-1C. Hence the large nand gate 74LS133 allowed a count to only occur in counter 'D' on a phi 2 edge at a 76.2939 hz frequency. The 'D' counter also feeds into a large nand gate which shuts off the envelope counting, once a single envelope sequence had been generated. In addition an outside envelope stop signal also shuts down tone generation by zeroing the envelope count and disabling the counter input.

Note that for pipelining latches are inserted between all the elements of the circuit. The pipelining is accomplished by using phi 1 and phi 2

Figure 6-1 (B) : Timing pulses for Pipeline
Clocking and Envelope Synchronization

Figure 6-1 (C) : Timing pulses for Pipeline Clocking and Envelope Synchronization

Table 6-1 : Hardware requirements for Signal Generation Hardware of the Digital Nonlinear Monophonic Synthesizer.

ITEM NO.	INTEGRATED CIRCUIT REQUIREMENTS
1	2; TRW Digital Multipliers....type MPY012HJ1
2	2; Nontransparent Latches....type 74LS374
3	16; Transparent Latches....type 74LS373
4	10; EPROMS...type 2716 or 2715 (2k * 8)
5	9; Adders.....type 7483
6	1; D type flip flop....type 74LS175
7	4; Binary counters.....type 74393
8	1; Dual 4 input and gate....type 7421
9	1; Quad 2 input and gate....type 7408
10	1; Hex inverter.....type 7404
11	1; 13 Input Nand gate....type 74133
12	1; 8 Input positive Nand gate....type 7430
13	1; Quad 2 input positive or gate....type 7432
14	Several 14, 16, 20, 24, and 64 pin sockets

Note: D-to-A Conversion is completed with AD565A and CMOS 4053.

Figure 6-2 : Digital to Analog Conversion with Sign Bit ON as Negative.

A'on X1 transmitted
A.off: X2 transmitted
4053 ANALOG MUX

BIPOLAR OUTPUT
to: headphones
or: Amplifier
or other instrumentation

alternatively in a latch enable configuration. When two binary quantities are loaded at the same time, such as with the multiplier, they both have the same latch control signal. Note also that as illustrated in figure 6-1A the sign bit bypasses some elements of the circuit such as the multiplier and the digital-to-analog convertor. In the case of the multiplier this is legitimate since the envelope values are always positive. In the case of the D-to-A convertor this is legitimate since a positive unipolar output comes out of the D-to-A, and conversion to a bipolar signal is completed by op-amp inversion and analog multiplexing of both positive and negative signals. (see figure 6-2)

A complete list of Signal Generation Hardware (up to the D-to-A, which is an AD565A DAC) is given in table 6-1.

6.2 Chromatic Scale Keyboard Interfacing

In order to get a signal transferred from the chromatic scale keyboard to the Signal Generation Hardware, additional circuitry and software is required.

An overview of the whole system showing the circuitry required for interfacing is illustrated in figure 6-3. The digital keyboard interface hardware was essentially the same as that shown in figure 2-21, Chapter 2. This keyboard circuitry is illustrated again in figure 6-4 with a few more NOR gates. The last NOR gate illustrated (pins 11,12,13) was fed into a combination of one shots which permit one interrupt only per keyboard scan. These one shots are illustrated in figure 6-5, and as illustrated in the timing diagrams permit only one note to be played per keyboard scan. This

Figure 6-3 : Interface of the Chromatic Scale Keyboard to Signal Generation Hardware.

Figure 6-4 : Digital Keyboard Hardware used for Keyboard Interface. (Simonton Jr., 1980)

NOTE timing regulated by external capacitors and potentiometers.

TIMING DIAGRAM

Note: only 1st note in a scan generates interrupt because of masking effect of 'B' pulse

Figure 6-5 : Interfacing of Interrupt Signal to signify Note Play Status.

Figure 6-6 : 6809 Interrupt Service Routine Algorithm

was necessary because the Signal Generation Hardware enables only one note at a time to be played. As shown by the K1 timing signal, as soon as a key is released, an interrupt signal is generated at the end of the next scan to turn off the released note.

The flowchart of figure 6-6 illustrates the 6809 assembler routine implemented to handle all possible conditions. A note left on will play to the end of its envelope. Otherwise a new note will cut off the last note played and the new notes signal will begin play. A note released will stop sounding immediately upon release. A tone takes a full duration time of about 2.5 seconds. If held down long enough its full 2.5 to 3.5 second envelope will play. However rarely is a tone held on this long so we rely on its attack time of about .5 seconds (or less) to allow timbral identification.

The 6809 assembler routine which was used to run the keyboard interface is listed in a departmental internal report (Armstrong,86)

6.3 EPROM Programming for Timbral Generation in the Signal Generation Hardware

EPROMs were used for all digital tables implemented in the nonlinear circuit. Programs were written to generate these tables which were then transferred to read only memory, hardwired into the Signal Generation Hardware.

There were four programs written in Basic on the Apple 2 microprocessor. Each of these programs was used to write data to a block of the Apples's memory which in turn was copied to the EPROM being programmed.

One of the card slots in the Apple contained the EPROM programmer. The four programs used to complete EPROM programming were called:

INCROM- for programming increment memory which regulated the frequency being used in the signal generation hardware.(64 different values..see fig.6-7)

TBLSIN- for programming the sine values in a 2048 memory table. (see fig. 6-8)

NONLIN- for programming the 2048 nonlinear values which comprise the nonlinear look-up table. (see fig. 6-9)

ENVFIL- variations of this program produced the 256 memory locations for the envelope tables. Two of these tables were required for each instrument synthesized, one for predistortion multiplication and one for post distortion multiplication. (see fig. 6-10)

Figure 6-7 : INCROM..Program flowchart for programming Increment ROM, used to increment through the sine table. The increment is actually an address step size which regulates sine table output frequency.

Figure 6-8 : TBLSIN flowchart. This program generates a 2048 place sine table with phase ranging from 0 to π .

Figure 6-9 : NONLIN..Routine for programming a 2048 word EPROM to provide a polynomial in nonlinear form which will produce a desired waveform. The polynomial coefficients are assigned at the beginning of the program and are used to calculate the nonlinearity.

Figure 6-10 : ENVFIL...Envelope filling program. This program takes 35 Envelope values and interpolates between them to get 255 points in a more highly resolved envelope.

All the above programs actually were used to 'blow' two, 2716 EPROMs each. This was because each word required 10 to 16 bits of memory and a single 2716 only provides eight bits. In each routine the calculation of the required values was done. For example in TBLSIN each sine value was first calculated; if $y = \sin(\pi/2048)$ this value of 'Y' would then be converted to a 10 bit representation, two bits for the upper byte and eight bits for the lower byte.

Similarly in INCROM, after the increment value was determined, two 8 bit bytes were programmed to hold the complete 16 bit representation. Note that although the increment register is 16 bits long, the phase register which holds the full phase look-up value is 21 bits wide and is rounded to the 12 most significant bits (9 fractional bits dropped) to look-up in the sine table.

The nonlinear look-up table was the most difficult to program. It was divided not only into lower and upper byte portions but also positive and negative input regions. This is illustrated in figure 6-11. Notice that four zero values were used and because of this there were actually 1022 values in each array between 6000 (hex) to 6400, 6400 to 6800, 6800 to 6C00, and 6C00 to 7000.

The most difficult EPROM programming software was for 'ENVFIL'. It used a linear interpolation routine to fill an envelope with 256 points. The envelope previously had only 30 to 35 points across a 3.4 second spread. This 3.4 seconds was the maximum envelope on-time used. Shorter on-times were used later in the development.

An internal report (Armstrong,86) contains listings of the BASIC programs that were used on the Apple. All programs began storing values in location 6000 (hex) in the Apple's memory. Both the nonlinear program

Table 6-2 Nonlinear Coefficients for Simulation of Clarinet and Oboe

Coefficient Number	Clarinet Coefficient	Oboe Coefficient
1	-.026316	-.260869
2	2.6342	1.93044
3	4.5632	11.875
4	-15.484	.96695
5	-89.326	-94.02435
6	97.642	-5.3426
7	595.71	227.561
8	-387.03	-265.9601
9	-1828.38	-369.9757
10	735.663	930.5043
11	2837.56	690.0869
12	-646.7369	-1086.3
13	-2166.568	-819.2
14	215.579	427.4087
15	646.7368	356.1739

Figure 6-11 : Nonlinear look-up table programmed in Apple Memory.
 Example using a quadratic function to program nonlinearity.

NOTE: NUMBER SYSTEM IS SIGNED
 MAGNITUDE.

(NONLIN) and the sine table program TBLSIN filled 4k of memory.

EPROMs that were being used for a 2nd or 3rd time were erased for 1 hour under an ultraviolet light before being programmed again. After an EPROM had data read into it, its contents were automatically verified by an Apple software routine which read the contents of the EPROM and compared this with the memory just written from. If there were any discrepancies Apple data memory was examined for errors, and a new EPROM was used.

The actual data used in the EPROM programming were as follows:

INCROM- used a Chromatic scale as defined by $2^{1/12}$ spacing in frequency. The scale starts as 110 hertz and proceeds to 64 tones above this. The values required to generate these frequencies by stepping through a 4096 sine table, with a 156250 sampling rate, were calculated. Increment values were resolved to 9 fractional binary places.

TBLSIN- by dividing the sine function phase into 2048 value from 0 to pi, half the sine wave table was generated. The other half was created by allowing one additional bit to account for a negative sign. The sine value was resolved to 10 fractional binary places. The 11th bit, the sign bit, was wired around the sine table. (see figure 6-1A)

NONLIN- this was the nonlinear look-up routine which used data from column 15 of the clarinet 311 hz tone. A similar selection was made to, obtain a time frame of Oboe nonlinear data. The nonlinear coefficients used for both the Oboe and Clarinet are presented in table 6-2.

ENVFIL- This routine was actually referred to by four different names in Apple software. Each of these four programs used different data but executed the exact same algorithm. The envelope data set for the clarinet is illustrated in figure 3-10. The envelope data for the

Figure 6-12: Envelope data curves for the Oroc.
 Note: these are extracted as per least squares fit analysis.

Oboe is shown in figure 6-12 and was extracted with a least squares fit analysis, as was that of figure 3-10. The data points of these illustrations were the basis of the linear interpolation which extracted a full 256 points for each envelope.

CHAPTER 7 : TESTING AND EVALUATION

7.0 Introduction

This chapter proceeds by presenting waveform and spectral results then elaborates on sound tests with verbal description. The sound tests describe in detail the timbre of the clarinet synthesis. This is followed by a terser description of Oboe synthesis.

Note that the process of testing and evaluation was ongoing throughout the development of the hardware and software. It began with Dft analysis of simulated waveforms and ended with a listening evaluation of instrument synthesis.

The Discrete Fourier Transform (Dft) analysis verified simulation waveforms were of proper shape. Similarly a spectrum analyzer was used on hardware waveforms to verify their content. A cross comparison of simulated and actual waveforms indicated expected consistency.

Growth and decay was observed in oscilloscope traces, of synthesized tones, verifying envelope functionability.

Listening tests were completed by a musically trained individual, Karen Brown (BMus, Ottawa U.), to obtain a constructive and descriptive evaluation of the tone synthesis.

7.1 Waveforms and Spectra

The most objective way to analyze how correct a tone synthesis has been, is by observing its waveform and determining the spectral content. The two waveforms illustrated in figure 7-1 proved correct as indicated by their simulation counterparts. These waveforms were observed and photographed as their envelopes were held at unity. They reflect the shape of the nonlinear curve.

Note that the envelopes used with the instrument waveforms were of fundamental importance in completing the synthesis. The presence of the envelopes was illustrated by the changes that took place in the signal waveform throughout the duration of the tone. Observation of this verified that wave-shape and wave-amplitude varied as expected.

Spectra

Synthesized spectra were observed to be close to required spectra by testing with a Hewlett Packard Spectrum Analyzer. All harmonics required were present with no more than about a 20% error. Harmonics past the 14th were essentially not present, indicating the accurate implementation of a 14th order polynomial.

Figure 7-1 (A) : Clarinet Waveform.
Compare this waveform to figure 3-8. This waveform is from 1.9 seconds into the tone while that of figure 3-8 is from 1.4 seconds into the tone. The two time frames are close and basically have the same shape. Their differences are illustrated by the mild variations in nonlinear curvature that each of their polynomials exhibits. This is illustrated by figure 3-2 (A), parts iii and iv.

Figure 7-1 (B) : The expected shape for this waveform can be verified by careful observation of the corresponding nonlinearity as illustrated by figure 3-3.

7.2 Sound Evaluation

Since the same nonlinear function was used throughout the full three octaves of the 37 note keyboard it would be of use to evaluate the accuracy of synthesis by considering each octave of play individually.

With the clarinet synthesis, the following descriptions were given by the test subject:

1st octave- 'full rich fog-horn sound'

2nd octave- 'more buzz and less hollow sound'

3rd octave- 'more whine and less hollow sound'

The waveform and envelopes were derived from a 311 hertz tone, or E-flat above middle C. The three octaves used on the synthesis keyboard were:

octave #1: 130.81 hz(low C) to 261.63 hz(middle C)

octave #2: 261.63 hz(middle C) to 523.25 hz(upper C)

octave #3: 523.25 hz(upper C) to 1046.5 hz(high C)

Since the waveform and envelope data were extracted from a 311 hertz tone one would expect the best synthesis to occur between middle C and Upper C. This theory was concurred by the test subject.

The following conditions describe how the synthesized tone evolves as more factors of the nonlinear synthesis are implemented. This description applies to a 311 hertz tone played on a Clarinet. Again a musically trained subject was used for the evaluation:

A. Waveform without envelopes....sounds like a bank of oscillators

or an organ.

B. Waveform with post-distortion envelope....sounds like an organ
with gradual attack and release.

C. Waveform with pre and post distortion envelopes....full Clarinet
sound apparent...clearly a reed instrument.

7.3 Further Description of Tones

Further verbal descriptions are completed in this section to describe the similarities and differences between tones tested. The descriptions are listed in point form to simplify their expression. Again the descriptions were given by the same musically trained test subject.

1. Clarinet: -'fuller in bottom octave'

-'squeekier in higher octaves, but with more of a reed sound'

-'tubular in lower octave, also hollow and smooth'

-'high octaves have funny attack, like sound is being
squeezed out.

2. Oboe: -'more twang sound, not like buzz of Clarinet'

-'better fit over three octaves than Clarinet'

-'fuller sound than Clarinet in lower octave'

-'no squeek factor as with Clarinet'

Further discussion of the quality of nonlinear synthesis is presented near the end of chapter 8.

CHAPTER 8 : CONCLUDING MATERIAL

Introduction

This chapter comments on the preceeding material. A set of concluding remarks are made about data analysis and nonlinear methods. A set of improvements and enhancements are presented.

The last section of this chapter presents overall comments about the work in this report, indicates nonlinear system limitations, and closes with general conclusions about the merits of Nonlinear Synthesis.

8.1 Analysis and Simulation Routines

It is known that every instrument has timbre which varies in time. Also at any given time a spectrum can be generated by a polynomial curve. This polynomial curve changes in time, as the timbre harmonics change. The predistortion curve has the effect of changing a constant polynomial in time in such a way that it will produce a timbre that has changes very similar to the original timbre. In other words the predistortion multiplier successfully eliminates the need for a changing polynomial to accurately represent a given tone's timbre.

As shown in the Timbre Data Analysis chapter a wide range of polynomial curve shapes can result when trying to fit a nonlinearity to the full timbre of a tone. However a visual inspection can aid in determining if a single curve can fit other curves accurately, with the aid of the predistortion multiplier. By observing, for example, the shape of a curve from $x=1/2$ to $x=-1/2$ we can imagine how, if this section could be

Figure 8-1: Fitting of curves by visual inspection.
 Note: curve 'A' is the source curve and curve 'D' is the curve we are trying to fit.

stretched, it would fit another curve. The stretching on the y-axis of the polynomial is completed with the post-distortion multiplier, while the sectioning or fitting of smaller portions of the curve is completed with a pre-distortion multiplier. (see figure 8-1)

Note that for both the Oboe and Clarinet synthesis only one set of timbre was used. This was the E-flat above middle C tone, played at average intensity. Since the synthesizer keyboard could only play one intensity of a tone, analysis with one loudness level was all right. However to account for timbre variation, from note to note, several tones would have to be analyzed and fitted to curves which would be recalled selectively as a particular note is played. This is a level of difficulty beyond the single timbre analysis used in this thesis.

Least Squares Fitting

The LSTSQFIT Fortran routine was effective in fitting one nonlinear curve to another by varying the pre and post distortion multipliers. The Least-squares-fit procedure was a novel approach for multiplier envelope determination with a nonlinear system. In this process only 26 time frames of the nonlinear curves were dealt with. The number, 26, was arbitrarily chosen (using 1/10th second intervals in the envelope) and further investigation may show that even smaller intervals are required, for better synthesis. Note that the 26 points of the predistortion and postdistortion envelope are further expanded to 256 points using a linear interpolation routine. Even with 256 points, rapid attacks caused a problem. For the Cornet the first 10 of 256 points were entered directly instead of by interpolation, to ensure an accurate attack sequence. Even with this

precaution envelope samples made large discrete jumps for the attack. It is suggested that for more precise envelope shaping a 512 point curve be used. This would only be necessary for instruments with a very sharp attack, say one less than 30 milliseconds.

Basically the least squares fit method was involved, requiring a lot of computer resources and lots of data. It did however work well as seen in the comparison of tone synthesis with and without least-square-fitting. Specifically the Oboe and Clarinet synthesis, using least square fitting was much more accurate than the Cornet synthesis which used estimated envelopes. However the Cornet was also a problem for digital synthesis because of its rapid attack.

Simulation Routine

The simulation routine introduced a novel type of analysis. Different log and linear systems were tested to determine: a) if log systems were more accurate than linear systems and; b) which number system would produce a 50 to 60 db signal-to-error noise without requiring too many bits.

The calculations in the simulation were based on a nonlinear and sine-table lookup.(with variable phase increment) Both log and linear number systems were signed magnitude and not two's complement. This factor simplified lookup implementation in both the simulation and consequent hardware.

It is known that log number systems are ideal for sampling as used in a digital phone system, but when one introduces a nonlinearity in the system, the log sampling loses its constant relative error thus becoming ineffective. As seen in chapter 5, the linear system has better

signal-to-error noise ratios when using the same number of bits as logarithmic sampling system. An effective compromise would be to use linear input to the nonlinear table and logarithmic output. This would allow post multiplication envelopes to be added instead of multiplied. A logarithmic digital-to-analog conversion (as in telephones) would have to be used.

Finally, concerning number systems it should be noted that when higher order harmonics are used, the nonlinear curve becomes more complex and more sampling bits are needed to maintain a high signal-to-error noise ratio.

8.2 Sinewave Generator...General Comments

Three approaches for making a nonlinear synthesizer were given in chapter 4. The first approach, the Phase Generator method, used a top octave generator and phase lock loops to produce a digital sampled sine wave. Unfortunately this technique proved too cumbersome to implement since varying sample rates resulted and this required either special filtering or multiple look-up tables to eliminate aliasing.

The second alternative using an analog-to-digital conversion, was also eliminated as a design choice because it relied on analog sinewave signals as a source input. The chromatic scaling of these sine signals required special tuning of temperature dependent circuitry.

The Sine Table with increment approach was not much simpler than the first two alternatives, but it was more reliable being all digital, and did not require any special filtering. In a polyphonic system the sine table could

be shared in a time division multiplex configuration. This enhancement would enable several notes to be played at once. Such a system is discussed in the next section.

8.3 A Future Enhanced Approach

It should be noted that a nonlinear system was assembled as detailed in chapter 6 to demonstrate circuit functionability and the nonlinear concept. For greater synthesizer flexibility and accuracy one may use larger EPROMS with multiple instrument capability or more exactly, multiple timbre capability. A polyphonic effect is accomplished primarily with time division multiplexing. Filters can also be implemented for further dynamic control.

A polyphonic, or simultaneous note playing effect, would require among other things, a more sophisticated keyboard interface and improved note handling software. The hardware interface would have to use dedicated memory to compare present note status with previous status as the Chromatic keyboard is scanned. If the scanned note is in the 'on' state an interrupt will only be generated if memory indicates the previous position as 'off'. Similarly if the scanned note position is in an 'off' state there will only be an interrupt if the previous position was 'on'. (see figure 8-2) The software interrupt handler for the polyphonic system would update a table of notes which are played by using output signals to control phase increment circuitry. The phase increment circuits are in turn TDM'd through the rest of the signal generation hardware. (see figure 8-3 A,B)

To implement different envelopes for different notes on the keyboard

Figure 8-2 : A Polyphonic Keyboard Interface. Note: memory is used to store the last position of a key. A change in key status initiates an interrupt. Interrupt acknowledge restarts scan process.

Figure 8-3 (A) : Time Division Multiplexed Notes.
6 Notes played simultaneously with clock speed near 1 Mhz.

Figure 8-3 (B) : Time Division Multiplexing Signal Generation Hardware Output configuration.

would require either a 'split' keyboard or a different envelope assignment for different notes. In a split keyboard a range of notes, say the lowest octave, is assigned one set of envelopes while the upper octaves are assigned an alternate set (pre and post distortion) of envelopes. To apply different envelopes to a range of notes being played simultaneously, one would have to rely on time division multiplexing again as illustrated in figure 8-4. In this circuit one uses the same tristate control signals as illustrated in figure 8-3(A) to again control a multiplexing latch configuration. A separate counter circuit is required for each envelope, since the starting times of the envelopes do not necessarily coincide. A double latch, after multiplexing, is required to synchronize the envelope signal with the appropriate sine wave signal. A similar multiplex circuit would be constructed for post distortion multiplication as well.

Multiple Table Blocks

If a separate look-up table is required for each note's envelope, then one may think that with a 37 note keyboard, 37 envelopes have to be TDM'd. However this is not the case since one may instead use 37 or more blocks of memory consecutively stored. If one implements 32K of memory one can separately address 64 different envelope tables using a control signal. This 'selection control signal' would be held in a table in the interface processor. Upon playing a tone the processor would output an envelope select code just before sending out a start envelope signal. (see figure 8-5)

The process of using larger EPROMs with multiple table capability is also applicable to nonlinear look-up as well. With 2K allowed for each

nonlinear table, a 64K memory could store 32 different nonlinear functions. Each function could define a range of notes on a single instrument or alternatively define a different instrument altogether. Again the interface processor would control which nonlinear function is applicable, as indicated by split keyboard definition or external switches.

Filters

If the synthesizer user wanted to implement filters, for further dynamic control, a whole new circuit with software control would have to be defined. A filter, with real time parameter changes, could be implemented to further change timbre over the interval of a tones duration. It is unknown however if such a process would enhance the integrity of the timbre of a tone more than the use of envelopes alone. A most practical application for a filter may be to control the intensity of a played note. Since normalized data is used throughout the signal generation hardware, intensity control has been ignored. As noted in chapter 2, changes in intensity cause changes in 'harmonic richness' as well as loudness. Both loudness and harmonic richness could be altered with filters. However implementing a dynamically controlled filter will not be discussed here because it is could become somewhat involved. It may be considered as a future topic of investigation.

8.4 Overall Comments on Nonlinear Analysis and Synthesis

The evaluation presented in chapter 7 has demonstrated the effectiveness of nonlinear synthesis, at least in the case of reed instruments. The key however to proper nonlinear synthesis is the efficiency with which one polynomial time frame can be fit to others, within the same timbre, by using pre and post distortion envelopes.

In this body of work, reed instruments have been the primary concern. Further investigation is recommended to understand how effectively brass and string instruments could be synthesized with nonlinearities.

As with Grey's work (outlined in section 2.1.4) one may choose to evaluate the effectiveness of synthesis by using discrimination listening tests.

One may conclude that the nonlinear method, although not as accurate as additive synthesis, as would be seen by discrimination tests, still offers a viable alternative, with reduced quantities of data. The trade off between reduced data and accurate synthesis is common to all music synthesis systems. It has been demonstrated, based on the stand alone listening tests of chapter 7, that timbre synthesis with a nonlinear system is a workable method.

Comparing the Nonlinear method to the Frequency Modulation method we find that there are again trade offs. The variation in modulation index for FM synthesis proves a powerful tool for altering harmonic structures as required by complex timbre. However the change in harmonic structure must

be found by trial and error with FM, whereas the Nonlinear method can use a more methodical approach, specifically the least square fit routine as shown in chapter 3 of this report. Both methods (Nonlinear and FM) are effective in reducing the data storage requirements and as such prove superior, with respect to data storage, to additive synthesis.

Nonlinear Synthesis Limitations

If one observes that the polynomial curves of a particular timbre vary widely over the duration of the tone of interest then one may conclude that a predistortion envelope will provide a fit only with a large error. The larger the error of the fit the greater the compromise of the synthesis.

Another weakness of nonlinearities is that they rely on constant frequency sounding for the harmonics of a tone. As indicated by Grey's work (seen in 2.1.4) frequencies vary somewhat throughout the duration of a single tone. Ignoring this fact did not seem to totally invalidate our synthesis, although there was no discrimination testing from which the accuracy of the nonlinear technique could be thoroughly assessed. Another factor which Grey presents as important in discrimination tests is the initial attack segment on a tone. This small amplitude, inharmonic portion of the attack could have been included in the synthesis but was not because it was not present in the timbre curves from which data was extracted.

Concluding Comments

This body of work has explored new concepts in Nonlinear Synthesizing. These are:

- a. Least squares fitting of envelopes.
- b. Logarithmic number sampling.
- c. Visual polynomial inspection to determine the appropriateness of single polynomial fitting to a tone.

A novel digital circuit has demonstrated the potential of nonlinearities in relation to reed instruments.

A lengthy synthesis simulation was useful for determining a good digital number system for hardware implementation. The simulation was effective at illustrating polynomial effects in a nonlinear system.

The least squares fit method was successful as a data reduction tool. Clarinet and Oboe synthesis was accurate. The phase increment with sine table lookup method worked effectively. Also the pipeline and timing circuit designs were functional and operated without error.

The only disappointment encountered was with Cornet synthesis results. The cornet synthesized sound was too crude to deserve merit as a brass replacement. This is attributed to the Cornet's sharp attack envelopes. Furthermore because of limitations with computer resources least-square envelope fitting was not done. Instead envelopes were estimated based on observation of harmonic curves.

The hardware circuitry was spread over two large wire-wrap boards and two teflon breadboards. If a permanent system was to be constructed it would have been confined to an amplifier sized enclosure with appropriate component and input/output access.

To broaden the usefulness of the nonlinear method, one would have to compile timbre data for a wide range of tones and instruments. Having done such, multiple instrument definition would be possible using large memory blocks and in the case of polyphony, time division multiplexing.

It seems that there is a credible use for nonlinear synthesis in the vast domain of music synthesis. However to realize the multi-instrument potential, more tones would have to be analyzed as prescribed in chapter two and three. This would require digitizing single tones from instruments and extracting their harmonic envelopes from which the polynomials would be found.

The use of nonlinearities is relatively simple to realize since the harmonic structure is related to polynomial coefficients as defined by the Chebyshev functions. The hardware design implemented in test circuitry was constructed with few debugging setbacks. Enhancement of this hardware as presented in this chapter indicates a reasonable synthesis potential for serious users.

3 of/de 3

MicroD

REFERENCES

1. Alles, Harold G. , "Music Synthesis Using Real Time Digital Techniques", Proceedings of the I.E.E.E., Vol. 68, No. 4, April 1980.
2. Arfib, Daniel , "Digital Synthesis of Complex Spectra by Means of Multiplication of Nonlinear Distorted Sinewaves", Journal of the Audio Engineering Society, October 1979, Vol. 27, No. 10.
3. Beauchamps, James, "Brass Tone Synthesis by Spectrum Evolution Matching with Nonlinear Functions", Computer Music Journal, Volume 3, No.2.
4. Chowning, John M., "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation", Computer Music Journal, April 1977, pp 46-54.
5. Devarahi, "The Complete Guide to Synthesizers", Prentice-Hall Inc., New Jersey, Copyright 1982.
6. Fink, Donald G. ; Editor-in-Chief , "Electronics Engineers Handbook", McGraw Hill Book Co., Copyright 1975.
7. General Instruments, Applications specs for Music Related IC's, 1980
8. Grey, J.M., "An Exploration of Musical Timbre", PhD Dissertation, Dept. of Music, Stanford Centre for Research in Music and Acoustics, Stanford University, Rep. STAN-M-2, Feb. 1975
9. Lebrun, Marc, "Digital Waveshapping Synthesis", Journal of Audio Engineering Society, April 1979, Vol.27, No.4.
10. Lerner, Eric J., "Electronically Synthesized Music", I.E.E.E. Spectrum, June 1983.
11. Moore, F. Richard, "Table Lookup Noise for Digital Oscillators", Computer Music Journal, April 1977, pp 26-29.
12. Moorer, James A., "Signal Processing Aspects of Computer Music, A Survey", Proceedings of the I.E.E.E., vol.65, No.8, August 1977
13. Moorer, James A., "The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulae", Journal of Audio Engineering Society, Vol.24, No.9, pp 717-727, 1976.
14. Moorer, James A. and Grey, John, "Lexicon of Analyzed Tones", Computer Music Journal, June 1977.
15. Motorola Inc., "Motorola Microprocessor Manual", MOS Integrated Circuits Group, Microprocessor Division, Motorola, Austin, Texas.

16. Olson, Harry F., "Music, Physics and Engineering", General Publishing Company Ltd., Copyright 1967.

17. Roads, C., "A Tutorial on Nonlinear Distortion or Waveshaping Synthesis", Computer Music Journal, Volume 3, No.2.

18. Roberts, Steven K., "Polyphony Made Easy", Published in: The Byte Book of Computer Music, Byte Publications Inc., pp 117-120, copyright 1979.

19. Schaefer, Richard A., "Electronic Musical Tone Production by Nonlinear Waveshaping", Journal of Audio Engineering Society, Volume 18, No.4, August 1970.

20. Seashore, Carl E., "Psychology of Music", Dover Publications Incorporated, New York, 1967.

21. Simonton Jr., "Friendly Stories About Computers/Synthesizers", Published by PAIA Electronics Inc., Jan. 1980.

22. Snell, John., "Design of a Digital Oscillator which will Generate up to 256 Low Distortion Sine Waves in Real Time", Computer Music Journal, April 77.

23. Stremler, Ferrel G., "Introduction to Communication Systems", Addison-Wesley Publishing Company, Copyright 1977.

24. Struve, Bill, "A \$19 Music Interface", published in :The Byte Book of Computer Music, Byte Publications Inc., pp 33-38, Copyright 1979

25. Suen, C.Y., "Derivation of Harmonic Equations in Nonlinear Circuits", Journal of the Audio Engineering Society, December 1970, Vol.18, No.6.

26. "6800 Assembler Manual", Technical Systems Consultants Inc. and Smoke Signal Broadcasting, Copyright 1977.

27. "Flex User's Manual", Technical Systems Consultants, Copyright 1978, West Lafayette, Indiana.

28. Interfacing of the M6800., Data Systems 68, 2316 Diversified Way, Orlando Florida.

29. "TRW VLSI Data Book", TRW Electronic Components Group, P.O. Box 2472, LaJolla, CA 92038.

30. Kennedy, Michael, "The Oxford Dictionary of Music", Oxford University Press, Oxford, New York, 1985.

APPENDIX A

POLYNOMIAL COEFFICIENTS AND OBOE CURVES

APPENDIX B

LEAST SQUARES FIT ROUTINE

APPENDIX A: POLYNOMIAL COEFFICIENTS AND OBOE CURVES

Table A1 Polynomials for 6 time frames of Clarinet Timbre Structure:

Polynomial Order	Time :.6 Frame 2	Time :.9 Frame 5	Time :1.4 Frame 10	Time :1.9 Frame 15	Time :2.4 Frame 20	Time :2.9 Frame 25
0	.07	.05	-.06835	-.026316	-.115625	-.00555
1	3.05	8.31579	2.9975	2.6342	2.3125	.84444
2	-4.72	5.6316	8.0861	4.5632	12.89125	-.6333
3	-66.12	-152.0	-26.1165	-15.484	-10.5375	-3.8889
4	47.12	-144.063	-147.9696	-89.326	-216.275	8.7111
5	556.0	1173.895	185.8835	97.642	49.60	28.9778
6	-152.32	1003.284	966.643	595.71	1345.00	-26.489
7	-1945.6	-4072.421	-697.681	-387.03	-185.0	-99.911
8	130.56	-3092.884	-2946.92	-1828.38	-3954.00	4.2667
9	3276.8	6871.579	1265.742	735.663	350.4	172.089
10	184.32	4777.77	4562.63	2837.56	5936.0	82.489
11	-2641.92	-5551.158	-1081.033	-646.737	-307.2	-142.222
12	-368.64	-3627.116	-3479.01	-2166.57	-4403.2	-113.778
13	819.2	1724.632	352.5671	215.58	102.4	45.5111
14	163.84	1077.895	1036.962	646.737	1280.0	45.5111

Table A2 Polynomial Coefficients for 6 time frames of an Oboe Tone (311 hz)

Polynomial Order	Time :.6 Frame 2	Time :.9 Frame 5	Time :1.4 Frame 10	Time :1.9 Frame 15	Time :2.4 Frame 20	Time :2.9 Frame 25
0	-.125	-.38334	-.260869	-.461165	-.42449	-.47619
1	-1.4	1.90333	1.93044	1.4733	1.30612	.09524
2	1.6	13.78	11.875	13.20388	4.40816	-.152381
3	-11.0	-.7933339	.9669561	7.86408	-4.66938	-2.74286
4	-48.0	-105.3733	-94.02435	-96.657	-46.106	1.600
5	127.2	-12.82669	-5.3426	-64.27188	-45714	6.55238
6	261.2	256.4266	227.5617	180.6602	164.9633	-1.21905
7	-493.6	-225.8133	-265.9061	-28.738	-42.5796	-4.2667
8	-731.2	-396.8	-369.9757	-68.3495	-336.457	1.21904
9	905.6	853.333	930.5043	484.6601	191.2163	2.4381
10	1158.4	678.4	690.0869	23.61161	466.0245	0.000
11	-780.8	-1024.00	-1086.33	-695.9223	-242.4163	0.000
12	-947.0	-785.067	-819.20	-188.8932	-384.5225	0.000
13	256.0	409.0	-427.4087	298.2524	100.3102	0.000
14	307	341.3333	356.1739	139.1845	133.7469	0.000

Figure A-1 : (A) Polynomial, (B) waveform and (C) resulting Spectra
Generated by the LLSIMULATE program.
(.6 seconds into Oboe timbre, 311 hertz tone)

(A) Polynomial : 1024 points, from .6 seconds into timbre envelopes.
Y axis is normalized. The corresponding waveform is
illustrated in part (B) and the spectra is illustrated
in part (C).

fig. A-1

(B) Waveform : 220 hertz, with Sample Rate of 131,072 and number of samples at 595.

Original Data : (.6 seconds into timbre)

harmonic	amplitude	harmonic	amplitude
0	0	8	3
1	49	9	2
2	50	10	10
3	80	11	4
4	50	12	5
5	15	13	5
6	5	14	3
7	5	15	0

Fig. A-1

(C) Spectra, 256 point dft results from waveform simulation with 10 bits of accuracy. (.6 seconds into timbre)
 Note: amplitude scale on plotted spectra relatively accurate, not absolutely.

Figure A-2 : (A) Polynomial, (B) Waveform and (C) resulting Spectra
Generated by the LLSIMULATE program.
(.9 seconds into Oboe Timbre, 311 hertz tone)

(A) Polynomial : 1024 points, from .9 seconds into timbre envelopes.
Y axis is normalized. The corresponding waveform is
illustrated in part (B) and the spectra is illustrated
in part (C).

Fig. A-2

(E) Waveform: 220 hertz, with Sample Rate of 131,072 and number of samples at 595.

Original Data : (.9 seconds into timbre)

harmonic	amplitude	harmonic	amplitude
0	0	8	10
1	354	9	80
2	347	10	310
3	477	11	180
4	465	12	120
5	600	13	60
6	128	14	25
7	43	15	0

RELATIVE AMPLITUDE

HARMONIC

Fig. A-2

(C) Spectra : 256 point dft results from waveform simulation with 10 bits of accuracy. (.9 seconds into timbre)

Note: amplitude scale on plotted spectra relatively accurate, not absolutely.

Figure A-3 : (A) Polynomial, (B) waveform and (C) resulting spectra
Generated by LLSIMULAE program.
(From 1.4 seconds into Oooe timbre, 311 Hertz tone)

fig. A-3

(B) Waveform : 220 hertz, with Sample Rate of 131,072 and number of samples at 595.

Original Data : (.6 seconds into timbre)

harmonic	amplitude	harmonic	amplitude
0	0	8	10
1	412	9	40
2	285	10	200
3	380	11	110
4	300	12	60
5	380	13	30
6	68	14	7
7	15	15	0

RELATIVE AMPLITUDE

Fig.
A-3

(C) Spectra : 256 point fft results from waveform simulation with 10 bits of accuracy. (1.4 seconds into timbre)

Note : amplitude scale on plotted spectra relatively accurate, not absolutely.

APPENDIX B: LEAST SQUARES FIR ROUTINE:

Algorithm for Determining Least Squares Fit

1. Normalize Harmonics Column by Column
 Normalization Factor: "HNORM(IFIT)"
 Normalization Array: "HN(IR,IFIT)" ; IFIT= selected column
 2. Generate Polynomials in 'Source' Column.
 3. Regenerate Harmonics in 'Source' Column with up to 250 cos amplitudes. (from zero to one, each one labelled by 'IX')
 4. Normalize 'Source' Column Harmonics.
 Normalization Factor: "XNORM(IX)"
 Normalization Array: "HNFUNCA(IR,IX)"
 5. Compare 'Source' to 'Target' Harmonics. Square difference and sum for each column of 'HNFUNCA'. Smallest square error indicates which cos amplitude is appropriate. (column 'IDO')
 6. Predistortion Multiplier = cos amplitude with smallest least square fit error.
 Postdistortion Multiplier = HNORM(IFIT)/XNORM(IDO)
- $$\text{Least Squares Sum} = \sum_{IX=0}^{IDIV} \text{ABS}(\text{HNFUNCA}(\text{IR}, \text{IX})) - \text{ABS}(\text{HN}(\text{IR}, \text{IFIT}))$$
- IDIV = no. of divisions of cos amp.

Selecting a Polynomial to fit a Tone's Timbre

1. Select Columns 2,5,10,15,20, and 25 and least squares fit each one to each other one.
2. Total least squares sum for cross fitting. For example if column 2 is selected, cross fit to columns 5,10,15,20, and 25. Add least squares sums of these fits.
3. Smallest sum from cross fitting indicates which column will be selected to fit all others.

END

06·07·89

FIN