CANADIAN THESES

THÈSES CANADIENNES

THE UNIVERSITY OF ALBERTA

A Study of the Algorithmic Generation of Synthetic Speech

by

R. Scott Stacey

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science

Computing Science

EDMONTON, ALBERTA

Spring, 1986

ISBN 0-315-30286-0

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR R. Scott Stacey

TITLE OF THESIS A Study of the Algorithmic Generation of Synthetic Speech

DEGREE FOR WHICH THESIS WAS PRESENTED Master of Science

YEAR THIS DEGREE GRANTED Spring, 1986

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) R. Scott Stacey

PERMANENT ADDRESS:

#7, 10468 - 76th Avenue

Edmonton, Alberta

T6E - 1L1

DATED Nov 8 1985

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled A Study of the Algorithmic Generation of Synthetic Speech submitted by R. Scott Stacey in partial fulfilment of the requirements for the degree of Master of Science.

L.K. Schubert

Supervisor

....

....

....

Date Nov 8/85

Abstract

This research concerns the algorithmic generation of high quality synthetic speech. Prior research efforts in the field and current trends in synthetic speech software and hardware are reviewed. The implementation and design of "Talker" is elaborated upon. "Talker" is the speech system of the "Kato Heron" robot in the Department of Computing Science at the University of Alberta. The system functions in the following manner: Unadorned orthographic input is first mapped into its English language phonetic equivalent through the utilization of a set of translation rules. This phonemic translation is then augmented prosodically with respect to the duration, amplitude, and pitch of each phoneme. The resulting output is capable of driving a virtual synthetic speech output device. This, in turn, produces highly intelligible synthetic speech. The output device chosen for use in the implementation of "Talker" is a SSI-263A mounted on the robot. Taped recordings of the output of "Talker" and comparable systems are included with the thesis.

Preface

Laws of Computer Speech (After Cater, 1983)

1. If a synthetic speech system can say something at the wrong time, it will.
2. If you repeatedly demonstrate your speech system to the same people, then they will expect an improvement in the quality of speech each time they hear the system talk.
3. A short, unexpected burst of synthetic speech will be lost in the clamor of normal silence.
4. The listener will concentrate on the novel characteristics of a synthetic voice rather than the content of the speech.

"... I know that _____ provides only small refinements over what is available in other systems. Yet several dozen small refinements add to something that is important to me, and I think such refinements might prove important to other people as well."

Donald E. Knuth.

Mathematical Typography, 1979.

Table of Contents

List of Tables

List of Figures

1. Speech Synthesis with reference to Text-to-Speech Systems.

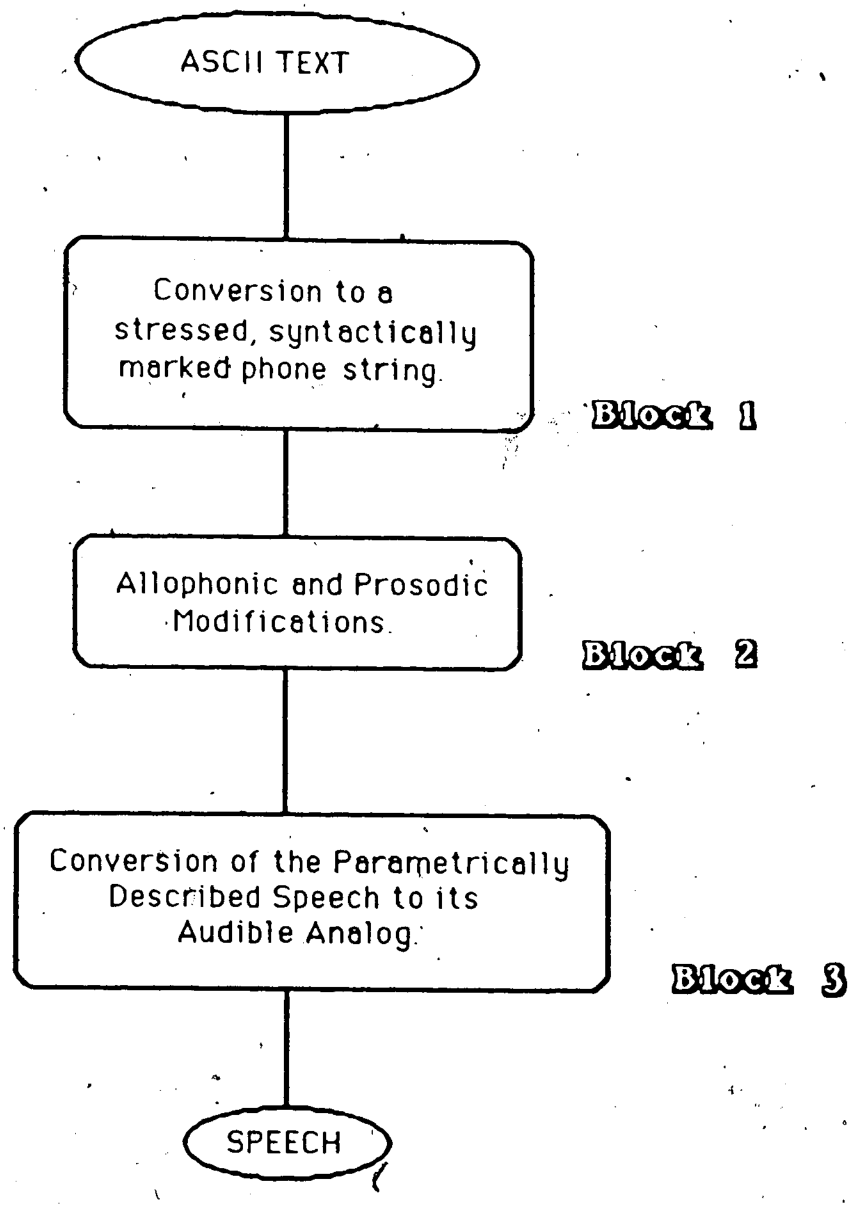
The subject of this thesis is the design and implementation of a system capable of delivering synthetic speech. Historically, speech synthesis has been interpreted as applying to a large range of activities. In the broadest sense, it refers to speech generated by means other than the human vocal apparatus. This includes speech produced by von Kempelen's synthesizer in 1769, Dudley's Voder in the late 1940's, and even the digitally recorded speech produced by a watch marketed by Seiko.

This research will focus on the area of speech synthesis that deals with the algorithmic generation of totally synthetic speech from unadorned orthographic input. This precludes speech formed from portions of a recorded human voice as well as speech formed from any means that was hand-edited. This type of synthetic speech generation is more commonly known as a text-to-speech system. Figure 1.1 is a generalized block diagram of the synthetic speech generation process. The order in which the blocks occur is relatively constant across most systems. Each block has many alternative implementations including elimination of that block. Figure 1.2 superimposes a generalized block diagram upon the modular flow diagram offered by Gilblom (1984). Gilblom (1984) specifies the component modules which make up a typical text-to-speech system in the order that they are most often implemented currently. The choices made by the system's designers greatly affect the quality of speech generated by the complete system, the speed of the system, and its overall cost.

Conversion of text to a phonemic string

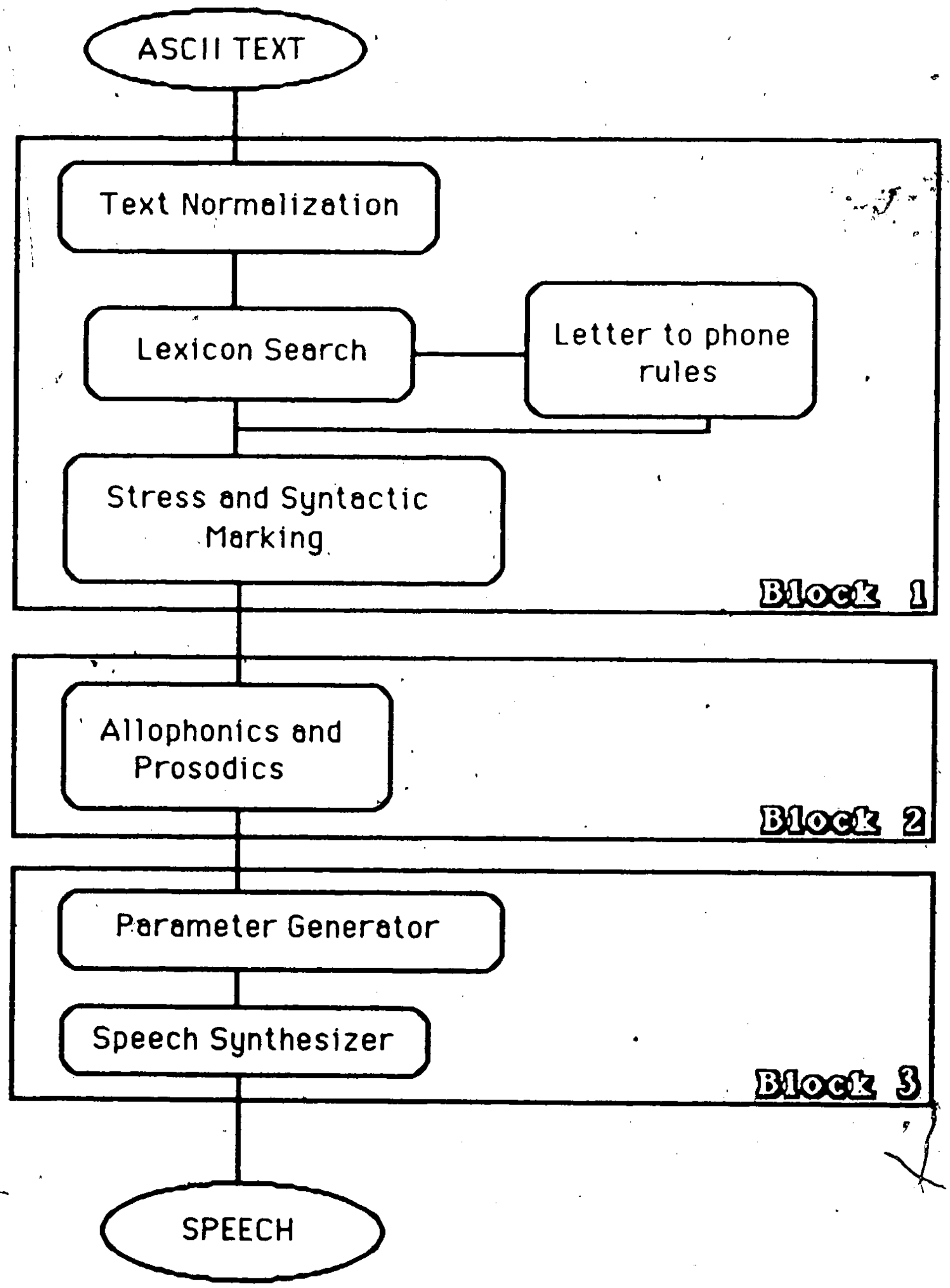
Text Normalizer

The text normalizer module attempts to ensure that a text-to-speech system is able to handle running text adequately (text as it is found in newspapers, magazines, and phonebooks). A



General Block Diagram of
Synthetic Speech Generation

Figure 1.1



Detailed Block Diagram of
Synthetic Speech Generation

Figure 1.2

string such as *1,234* should be spoken as *one thousand two hundred and thirty-four* as opposed to *one comma two three four* . This points out that it is not enough to correctly analyze the input in isolation, rather the context must be considered. This also holds for constructions such as *$12.34, twelve dollars and thirty-four cents; $.45, forty-five cents; 22nd, twenty-second; 12:00, twelve o'clock; Dr. Denton Dr., Doctor Denton Drive.* The idea of contextual consideration is very important and can be handled for these simplistic constructions. Problems arise for more complex tasks such as deciding whether to pronounce *1934* as *one thousand nine hundred and thirty-four, nineteen hundred and thirty-four* , or *nineteen thirty-four* . Currently, all systems which even recognize and attempt to deal with these problems, require the user to specify the correct decision.

Letter to Sound Conversion

Accurate translation from normalized orthographic text to its audible counterpart, is generally thought to require some combination of a set of rules and a lexicon of exceptions. However, systems in limited vocabulary domains often do not use rules [1] and low cost systems in unlimited vocabulary domains often do not make explicit use of an exceptions lexicon [2]. The size and content of the exceptions lexicon and the number as well as the generality of the rules used, are important implementation considerations for this module.

Stress and Syntactic Marking

Gilblom (1984) states that

"Very natural speech can only be produced if the converter *knows* what it is saying." This statement implies a level of artificial intelligence not currently achievable. Given that we cannot attain the ideal, the next best thing that can be done is to take the first steps towards developing a converter which '"... knows what it is saying.". One of these first steps, is to

[1] For example: "Speak-n-Spell" from Texas Instruments Inc.
[2] Text to Speech system from Sweet Micro Systems Inc.

develop a converter with the ability to analyze its input syntactically. The more syntactic analysis performed by the system, the greater the potential disambiguation abilities of the system. Appropriate levels of syntactic analysis will allow the system to resolve the inherent ambiguities associated with pronouncing *lead, (noun vs. verb); wind, (noun vs. verb); read, (past vs. present); bow, (noun vs. verb); close, (noun vs. verb); separate, (noun vs. verb)*. Appropriate syntactic analysis will also allow the proper stress placement in words such as *present, (noun vs. verb); desert, (noun vs. verb); insert, (noun vs. verb)*. For additional examples, see table 3.2. Syntactic analysis is not the panacea that it may seem to be at this juncture. It offers no assistance in disambiguating the pronunciation of words such as *row, (noun vs. noun); bow, (noun vs. noun)* and the construction *1984* . To achieve success in this task, semantic analysis is required.

Obviously, the extent of syntactic and semantic analysis included in the implementaion of a text-to-speech system is a very important consideration. Currently, only the highest quality text-to-speech systems include even a limited amount of *ad-hoc* syntactic analysis. It is often considered a subtlety that is not essential to the understandability of synthetic speech. No system known to this author (at the time of writing, November 1985) even attempts semantic analysis.

Allophonics and Prosodics

One of the least investigated and most interesting areas of speech synthesis is the systematic algorithmic augmentation of synthetic speech with the prosodic features normally found in human speech. Current systems which fail to address this issue have been explored extensively in the past and will not be discussed here [3]. Other current systems do address the problem of prosodics in synthetic speech but do so in an unstructured and unextendable manner [4]. The results of this type of research have been heavily commercialized, and for the most part,

---

[3] For example: "Speak-n-Spell" from Texas Instruments Inc.
[4] "SAM" (Software Automatic Mouth) from Don't Ask Inc., the Text to Speech

the information regarding these devices is proprietary in nature.

"Prosodics" is a catch-all category for the suprasegmental aspects of speech which convey additional information over and above the semantic content of the sentence. This additional information occurs on three logically distinct levels. The sentence "The frog jumped into the lake" is semantically clear. Yet by manipulating the major prosodic parameters (pitch, amplitude, and duration) with reference to high level concepts (Such as: perceived audience, emotion of the speaker, and intent of the speech) the speaker can cause the listener to infer many things. The sentence can even be turned into a question. Thus, the speaker can use prosodic information to convey to the listener additional information to clarify or even at times to confuse the semantic content of the sentence. This is not an easy task to accomplish algorithmically. In the simplest case, at least a syntactic (and probably semantic) analysis of the sentence in a context would be required. The size of the context (sentence, paragraph, or even some measure of context in terms of real time) is not clearly defined. In the general case, the task appears to be quite impossible unless the synthetic speech was generated from a conceptual base rather than through an augmented translation of unadorned orthographic text.

At the second level, the same prosodic parameters (amplitude, pitch, and duration) can be manipulated in the context of phonemes, words, and phrases. At this level, the most important prosodic features are in order of importance the pitch or fundamental frequency (also known as F0), the rhythm (governed by the duration of individual segments), and the amplitude or intensity of the voice signal (Gillot, 1984; Hill, 1980; Bolinger, 1958; Fry, 1955).

At the third level, "segmental" features are found. These are the features which affect the articulation of individual phones or syllable segments. In general, each sound or phone generated by the human vocal tract is contextually dependent on its surrounding phones. A natural-sounding text-to-speech system must model this process, so that pauses are in the correct places, pitch contours conform to the "accepted" norm, the vowels have the proper

---

[4](cont'd) system from Sweet Micro Systems Inc., and Dectalk from Digital Equipment Corporation.

quality (in context), and the phones of a word have the expected durations. Faulty pitch, amplitude, or timing patterns are distracting and hence impair comprehension.

An example of the importance of generating the correct low-level prosodic information can be seen in yelled or intentionally fast speech. In the former case, it is the altered duration of the phonemes, the rate of change of pitch, and the spectral tilt that occurs rather than the volume of the sentence that indicates yelling. In the latter case, the process of deleting certain phonemes, called ellipsis, is present as well. In both cases, not all phones in a word are equally shortened, rather some are shortened, some remain the same, and some are totally eliminated.

Conversion of Phoneme String to its audible analog

Parameter Generator

Speech may be described parametrically in terms of phonemes which have their pitch, duration, amplitude, breathiness, etc. annotated in some fashion. However the speech is specified, the task of the parameter generator module is to translate the phonetic description of the original text into parameters which may be used to drive a speech synthesizer. The format and content of the parameters generated depends on exactly what speech synthesizer is available. Most decisions governing the implementation of this module are dictated by the decisions made when selecting the synthesizer hardware.

Speech Synthesizer

This module may be realized in software, firmware, hardware, mechanically or even acoustically. Most current systems use a digital or analog filter formant synthesizer which is realized through some combination of hardware and software driving a speaker.

1.1 Applications

A text-to-speech system has many potential areas of application [5]. These applications fall into many categories, but two useful divisions are based on the interactive/batch dichotomy and the size of the vocabulary to be spoken.

An unlimited vocabulary synthetic speech system could serve as a output device for any number of interactive systems (including expert systems) where the human component would benefit from audio output. Klatt (1982) points out that in applications which require a limited and finite number of responses, real speech may be digitized and encoded. This allows the responses to be decoded and played back on demand in either a batch or interactive environment. Cater (1983) offers a list of applications of this type which include:

1. talking appliances (clothes washer, microwave oven, television, clocks).
2. talking transportation (cars, elevators).
3. talking entertainment devices (arcade games, cameras, phonographs).
4. talking tools (electronic multimeters, language translators, calculators).

Because of the novelty and low cost of this type of synthetic speech technology, this list is really only limited by the collective imagination of manufacturers. Consumer products may be grouped into two classes: those which require vision to be used for operating the product (i.e. automobiles, televisions, and cameras), and those which do not. Items in the former class equipped with a synthetic voice are often seen as annoyance because most of the information they can deliver is redundant. Conversely, talking elevators and household appliances can be a great boon to a blind homemaker.

Applications which require very large vocabularies or the ability to deal with arbitrary combinations of words in the formation of sentences, are not suited to the stored speech approach. A system which converts unadorned English orthographic input into speech is ideally suited to applications of this type.

----

[5] Gilblom (1983) offers a review of the potential areas and types of benefits that may be realized.

The ideal interactive system would allow speech to be generated in real time. The system could selectably pronounce punctuation, speak letters, whole words or sentences. Klatt (1982) offers a number of examples of this type of application.

1. Reading machines for the blind:

   A reading machine for the blind is an interesting application for a speech synthesis system that has never approached its full potential. The idea has been well discussed in the literature and many systems have been devised, each meeting with limited success (Lee, 1969; Allen, 1973). The Kurzweil reading machine for the blind (currently the most advanced commercial device available) is reputed to have such low quality speech that extended listening is impractical (Witten, 1982). Similarily, a text-to-speech system incorporated in a workstation of a computer installation, could potentially allow visually impaired people access to a device which was designed exclusively for people with normal vision. Work has been done in this area by T. Vincent of the *Open University* in the U.K. (Vincent, 1982a; 1982b).

2. Talking instrument panels:

   This example concerns situations where response time is critical. (eg. chemical plants, or operating theatres). The operator/doctor may not be in front of his instrument console constantly. In the event of a dangerous condition, an alarm usually sounds recalling the individual to the console to determine the origin of the problem. Valuable time could be saved if the system could replace the alarm with a verbalized warning which the individual could respond to and understand.

3. Remote access to information over the phone:

   There exists a need for remote access to interactive systems through a very primitive terminal - the telephone. Here, the advantage could be the system's ability to deliver its output "advice" over a phone line through a telephone receiver to an operator in the field.

This "advice over a phone" may take another form. It is not uncommon for people to be involved in an activity which fully occupies their tactile and visual senses but does not utilize their acoustic sense. The example of a circuit repairman, or a wirewrap technician is appropriate. Here, the instructions necessary to complete the task could be given by a synthetic speech system, thus allowing the operator to concentrate on the task at hand.

4. Speaking aids for the acoustically handicapped:

   Deaf and dumb people are often forced to communicate with others via conventional long distance communication devices such as telephones, which are geared to acoustically complete individuals. Many of the inherent handicaps of the telephone system could be alleviated if the deaf or dumb person could key his request or reply into a synthetic speech system. This would facilitate the use of a device which was never designed for people with those sorts of disabilities.

5. Educational Research Tools:

   An interactive text-to-speech-system could be used to study the acquisition of reading skills. This could be done by experimentally removing portions of the system's rule base until the system ceased to function properly. If the resulting behavior of the system could be considered analogous to the behavior of a learning-disabled child then some parallel might be drawn between the missing portion of the rule base and the missing item(s) in the child's repetoire of reading skills.

   By way of further example, consider the acquisition of phonic skills. This intuitively seems to be a gradual stepwise process for a child. This same stepwise process could be modelled by observing the behavior of the system when it was only working with a subset of its phonic rules. By introducing phonic rules back into the rule set in a stepwise fashion, the researcher could study acquisition of specific phonic skills and the process of forming general rules by modelling a child's skill acquisition process.

The batch mode of operation would be ideal for verbal presentation of instructional/recreational materials, where human presentation is unavailable or too costly. Another application could be to produce a "speech file" from a "text file". This would allow editing and proofreading of orthographic material by ear. Blind people could "read" whole texts of printed matter available in an "on-line" form. A batch text-to-speech-system could be used in computer assisted instruction (CAI) work to facilitate the course author's use of audio output in his courseware. This would allow the author to specify an auditory component to the lesson plan and have it compiled in concert with the visual component. It is a well established fact that audio and visual stimulation is much more effective than visual presentation alone.

1.2 Objectives

The major goals of this research are twofold. The first is to design a text-to-speech system that is capable of addressing the basic problems of synthetic speech. The second goal is to implement a portion of the design so that a user may enter unadorned English text into the system and to be able to listen to the synthetically generated speech. The ideal text-to-speech system should be able to produce output indiscernable from that of a person reading text aloud. This implies certain surface features and secondary goals of the system.

Firstly, the system should take no more time to process the input than an accomplished reader would take. An adequately fast text-to-speech system can be achieved through an appropriate combination of a "powerful enough" computer, and well designed algorithms and data structures.

Secondly, the speech should be appropriately divided into breath groups and the speech rate should be in the range perceived as "normal" by the untrained listener.

Further, the system should be "uncrashable" and not balk at misspellings, ungrammatical constructions, or unknown words [6]. The desired robustness is possible if the

---

[6] Consider the master of ceremonies. He will always mispronounce a performer's name in the interest of continuity rather than taking the time to stop and ask the

basic design of the system allows correctness of pronunciation to be traded off for the dependability of always generating a pronunciation.

Finally, the system should be able to convey to the listener an impression of an understanding of the text. The phrase "convey an impression of an understanding of the text" is used in this situation so that the listener will perceive the synthesizer as having "understood" more than just how to pronounce an isolated series of words. As Witten (1982) states,

> "the intonation patterns used by a reader depend not only on the text itself, but also on his interpretation of it, and also on his expectation of the listeners' interpretation of it. For example:
> 1. **He had a *red* car.** (I think you thought it was black).
> 2. **He had a red *car*.** (I think you thought it was a bicycle)."

A system will be able to "convey an impression of an understanding" only if the prosodic features of the output correctly reflect the syntax and semantics of the input text. This latter question of prosodics is by far the most difficult one to addresss.

The text-to-speech system will be designed in a manner which allows it to be used as a research tool. This implies that the design will allow for systematic experimentation and subsequent incorporation of new ideas, a virtual speech synthesizer output device, and a top-down implementation. This design must allow experimental decisions to be made as to how to divide the system's knowledge base between translation rules and lexical entries. The intention here is through experiments to allow a compromise to be reached which requires the system to "look-up" only those words which are ambiguous out of context [7] or which are mispronounced using generally accepted English pronunciation procedures [8]. Use of rules in this system are desirable because they allow a very robust design. These rules must conform to a number of objectives.

1. The rules should generate the most reasonable phonetic representation of a word.
2. The rules should be maximally general in their scope of application, and minimal in

---

[6](cont'd) performer for the correct pronunciation. The author feels that a text-to-speech-system should behave in an analogous manner.

[7] read, bow, row.

[8] draught, jojoba

number.

3. The rules should take the form of data which is extrinsic to the system which interprets that data.

The system's lexicon must not be merely a look up table but rather a repository of pronunciation information for truly anomalous or ambiguous words. The justification for this is that no speech synthesis system that purports to be correct in its pronunciations with a confidence level approaching 100%, can ever rely <u>only</u> on its translation rule base, or <u>only</u> on its lexical entries. Some sort of compromise situation must be reached. This is due to the diverse multilingual origins of the English language, the sheer massiveness of the English vocabulary, the phonetic misspellings of existing words and the constant evolution and innovation found in "living" languages.

By designing a system which drives a virtual speech synthesizer, the use of a wide variety of actual synthetic speech hardware is facilitated. For this reason, the instructions for the virtual speech synthesizer must be general and comply with certain assumptions about the output device. One assumption is that the output device takes its input in terms of some analog of English phones. This suggests that the phone set used by the system to describe its output to the speech synthesizer should be able to represent the international phonetic alphabet (IPA) as a subset. Further assumptions are that each phoneme can be given an amplitude, pitch, and duration value. These characteristics can probably be most productively given in abstract terms. This allows the specific device driver to construct the most felicitous mapping for the application being considered [9].

Finally, given that the system's instructions were correct, the quality of synthetic speech would be directly related to how readily the instructions could be translated into the code needed to drive the actual output device. This would imply that high quality synthetic speech

---

[9] Pitch values could be given on an integer scale of 0 to 100 where 0 is low and 100 is high. Amplitude values could be given on an integer scale of 0 to 10 where 0 is silent and 10 is loud.

hardware must at least have a tight control on pitch, amplitude, and duration of phonemes regardless of how they are generated.

Implementation

Although, an entire text-to-speech system as outlined in figure 1.1 will be considered theoretically, only the first and third blocks will be implemented. The second block, corresponding to the allophonic and prosodic modifications to a phoneme string, will not be implemented; however, provisions will be made for its later inclusion [10].

The implementation is written in Pascal code compiled and running under a Unix Operating System on a Digital Equipment Corporation's VAX 11/780. The speech synthesizer hardware is a Silicon Systems Inc. SSI-263. This is a single C-MOS chip phoneme speech synthesizer. The chip is located on a Heathkit Hero, a small independently mobile robot. The robot is attached to the VAX 11/780 via a serial link [11]. The basic acceptance criterion for the implementation is that it will be able to acceptably pronounce all syntactically unambiguous constructions found in Canadian English [12]. No statistical results, only pronunciation error rates in selected data will be reported.

----

[10] The rationale for this decision is that the problems of allophonics and prosodics are so vase and poorly understood that a project of this nature could not effectively offer an implementation worth undertaking. Implementation efforts were instead turned to areas where greater benefits could be realized for the time invested.

[11] These choices were made based on the materials and funds available to the author.

[12] Acceptability of pronunciation is to be determined by the author's supervisory committee.

2. Historic and Contemporary Approaches to Speech Synthesis

This section is divided into five main parts. Firstly, proprioceptive feedback loops and the acoustic theory of speech production are discussed. Secondly, with regard to speech production, it is instructive to examine how natural speech is produced in Homo Sapiens. Thirdly, the development over time of the different devices which have been engineered to achieve the goal of generation of speech synthetically are discussed. Fourthly, the various methods which have been used to control these devices will be discussed. Finally, the contemporary approaches to the generation of synthetic speech will be reviewed from a systems point of view. Both the currently available hardware and software will be discussed and when they cannot be separated, the system as a unit will be discussed.

2.1 Acoustic Theory of Speech Production

The production of speech by a person is a very complex task which utilizes feedback loops involving many of the senses, proprioception, and the speech generation center. The primary sensory channels involved are audition (unless deaf, one hears onseself speak), taction (one can feel one's tongue and lips move), and proprioception [13].

Let us examine how the word "Hello" is read aloud. The eyes recognize the word and signal for the retrieval of the information describing how to say "Hello". The vocal apparatus generates the speech while the ears, lips, tongue, palate, and teeth function as the receptors of the input feedback signal. This feedback input is used to "fine tune" the vocal apparatus so that the emitted sound and the movement of the vocal apparatus evince agreement with the stored information retrieved earlier. This feedback loop allows a person to say "Hello" in a recognizable fashion in a number of situations such as when a person is chewing gum, has had laryngitis, or has just returned from the dentist with an anesthetized mouth.

[13] One could imagine the visual sense serving in a proprioceptive manner if one was reading his own lips in a mirror.

Cater (1983) offers an introduction to this feedback loop and a description of the sensory systems involved. Unfortunately, there is no analog to the use of proprioceptive sensory systems and the general feedback loop principle in current speech synthesis technology. The problem is that a feedback loop of this nature presupposes the ability of an organism (speech generator) to recognize speech. Speech recognition by machines is an area which is poorly understood but currently under research. However, the vocal apparatus, which is the biological analog of the electronic hardware used in speech synthesis, and the acoustic theory of speech production without the elements of the feedback loop will be discussed.

The acoustic theory of speech production is based on a source and filter model. The origin of voiced and unvoiced sounds can be regared as the "source". This corresponds to phonation. The articulation of the mouth can be viewed as performing a filter function. Therefore, any speech sound can be regarded as the filtered output of a network into which a sound source was input (Fant, 1973).

> "The characteristics of any quasi-stationary sound segment thus contains the characteristics of the source and those of the network, the latter referred to as the vocal tract transfer function or filter function. In terms of Laplace transforms
> $$P(s) = S(s)T(s)$$
> where **P(s)** pertains to the radiated sound, **S(s)** to the source, **T(s)** to the vocal tract transfer function, and s to the complex frequency variable."
> (Fant, 1973)

2.2 Human Speech

The human vocal tract consists of an air-filled tube of approximately 17 centimeters in length. The actual organ of voice is the larynx which is situated in the distal portion of the vocal tract contiguous and below the base of the tongue. The volume as well as the length of the larynx varies according to the sex and the age of the individual (Gray, 1974). The average diameter in an adult male is 4.4 centimeters as reported by Gray (1974). The remainder of the vocal tract is composed of the air passages in the pharynx, oral tract and nasal tract which are of a deformable nature causing great variance in the resonant characteristics of the vocal tract as a whole. Figure 2.1 labels the features of the vocal tract (after D.L. Rice, 1976b).

Voiced speech begins with the vocal cords and the glottis (space between the vocal cords) where the flow of air from the lungs is cyclically broken into what is termed a glottal pulse. The cycle starts with the opening of the glottis enabling the flow of air past the vocal cords. The vocal cords start to vibrate causing the glottis to rapidly open and close. As the glottis snaps shut, ending the driving pulse with a rapidly falling edge, the air present in the tract vibrates for a few milleseconds [14]. The glottal pulse shape is represented in figure 2.2. When the glottal pulse is examined in terms of frequency, the result is a graph similar to figure 2.3. This graph indicates that the glottal pulse frequency can run from 0 to approximately 500 Hz. but the distribution is highly biased towards the lower frequencies. Cater (1983) explains this.

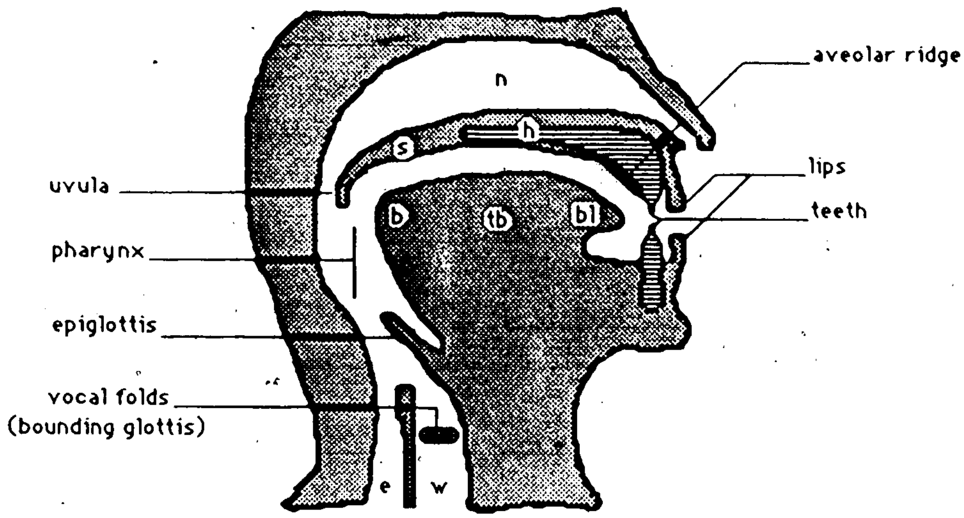
> "As the vocal cord muscles are tightened during speech, the fundamental frequency, or primary frequencies, of this distribution curve will rise in frequency to produce a rising change in voice pitch. Typical pitch frequencies for male voices range from 130 Hz. to 146 Hz. with an average frequency of around 141 Hz. The voice pitch of a female, on the other hand, has a range of approximately 188 Hz. to 295 Hz. with a median frequency of approximately 233 Hz. Under certain extremes of voice frequency extension during very inflective speech, the human glottal oscillation may reach a pitch as high as 480 Hz."

If, for simplicity, we ignore the resonant effect of the nasal tract, the length of the vibrating column of air is determined by the distance from the closed glottis to the lips where the speech is emitted. This simplification is justifiable because it allows the vocal tract to be viewed as a reasonable approximation to a pipe closed at one end.

Now consider, the frequency response of an ideal column of air. It will possess resonant frequencies (formants) corresponding to odd integral multiples of the source signal's quarter wavelength. There are strong energy peaks at odd multiples of the quarter wavelength. The equation which describes this behavior is:

---

[14] This is an explanation of only voiced speech which ignores the other types of speech (e.g. unvoiced speech).



Key

e - esophagus
b - back of tongue
bl - blade of tongue
h - hard palate
n - nasal cavity
s - soft palate (velum)
tb - tongue body
w - windpipe

Sagital Section of Human
Vocal Tract

Figure 2.1 (After D. R. Hill, 1980)

The Glottal Pulse Shape

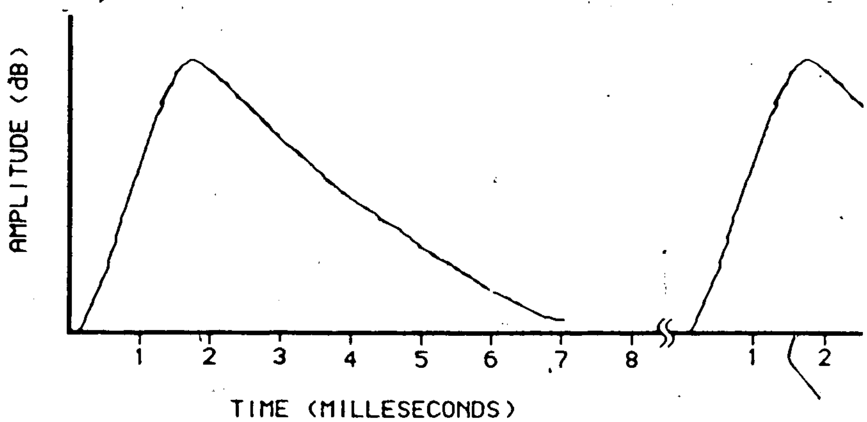


Figure 2.2

Glottal Pulse Frequency Spectrum

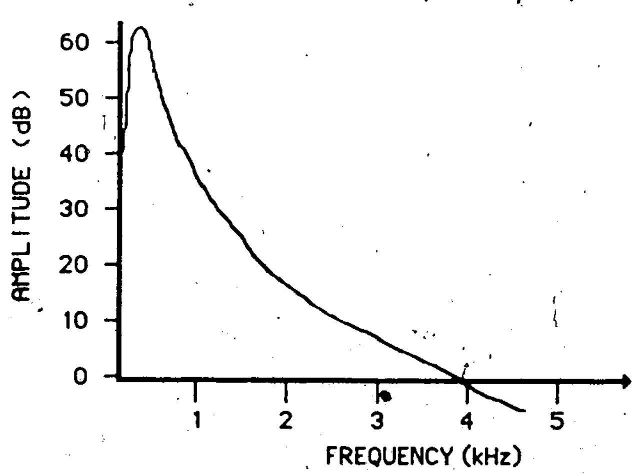


Figure 2.3

$$F_n = (V/(4L)) \quad n = 1, 3, 5, \ldots$$

Where F is the frequency (Hz.) of the resonance, V is the speed of sound in normal air (340 m/s) [15] , L is the length of the pipe, and n is the odd formant being considered. Assuming our ideal hypothetical tube is of constant diameter and 17.5 cm. long, the odd resonant energy peaks would have frequencies of 500 Hz., 1500 Hz., 2500 Hz., etc. These are known as the formant frequencies; F1, F2, and F3 respectively.

The resonances of the human vocal tract are not fixed at 1000 Hz. intervals and may be swept to higher or lower frequencies depending on the shape of the tube.

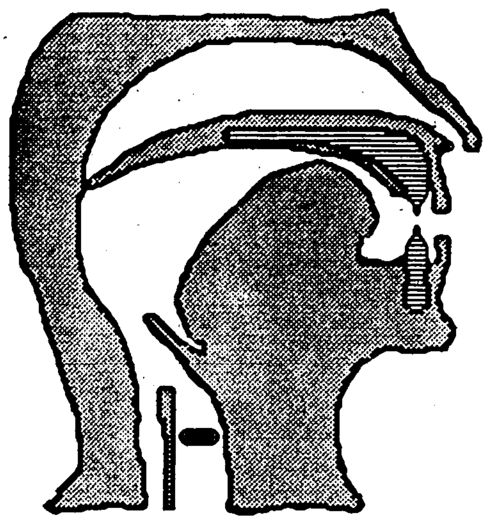
> "The average spacing withing the frequency scale of these resonances is of the order of 1000 c/s or more specifically $c/2l$ where l is the effective length of the vocal tract and c the velocityof sound. This inverse dependency of formant frequencies on vocal cavity length dimensions explains the higher formant frequencies of females compared to males, and of childern compared to adults."
> (Fant, 1973)

For example, moving the tongue forward and upward to pronounce "ee", as in figure 2.4, causes the shape of the tube to change so that there is a large resonant cavity in the back of the mouth where the tongue has been pulled away from the walls of the throat. The size of the tube just behind the teeth is greatly reduced. This new shape results in F1 dropping to as low as 200 Hz. and F2 rising to as high as 2300 Hz. (D.L. Rice, 1976).

This is the way in which one synthesizes one's speech on a moment by moment basis. To make a trite but important point, it should be noted that people have been doing this effectively for an exceedingly long time in a very unconcious manner. Over time, this unconcious phenomenon has been modelled by many different people who have approached the problem from many different perspectives. The following section discusses some of the more meaningful historical work in the area.

---

[15] Temperature directly affects the speed of sound. At room temperature sound is considered to travel 330 m/s however this is inside the vocal tract which is at an elevated temperature.



Sagital Section of Vocal Tract
Generating "ee"

Figure 2.4
(After D.L. Rice, 1976)

2.3 Historical attempts at Synthetic Speech

The current work in speech synthesis, and to a lesser degree, the work described in the following sections, is a direct extension of earlier investigative studies that have been going on for centuries. A full history of speech synthesis will not be attempted here. [16] However, some of the earlier attempts at connected speech synthesis by rule which were significant in their contribution to the understanding and advancement of the subject of speech synthesis as a whole will be discussed [17].

The conception of an artificial speaking device dates back to Gerbert (d. 1103) and Albert Magnus (1198-1280) who are both purported to have constructed "speaking heads". Robert Greene's <u>The Honorable Historie of frier Bacon and frier Bongay</u> (1594) acquaints the reader with the "myth of the brazen head" whose construction is attributed to Roger Bacon (1214-1294) (Mattingly, 1968).

Contemporary speech synthesis is considered by Mattingly (1968) to begin with von Kempelen's investigations of 1769. The final version of von Kempelen's device (as seen in figure 2.5) was operated by a person who manipulated the bellows and tubes with his right hand while his left hand effected the different resonances required by altering its position in front of the "mouth". Mattingly (1968) relates von Kempelen's claim that one could learn to operate the synthesizer in 3 weeks and synthesize phrases such as "Romonarum imperator" or "Vous êtes mon ami".

----

[16] For a more complete description, please see: "Reed Organ-Pipes, Speaking Machines, etc.," *The Scientific Papers of Sir Charles Wheatstone* . (London and New York, 1879), pp. 348-367, or *London and Westminster Review*, 6 and 23 (1837), 27-41; H. Dudley and T. H. Tarnoczy, "The Speaking Machine of Wolfgang von Kempelen," *Jour. Acoust. Soc. Amer.*, (1950), 22:151-166; C. G. M. Fant, "Modern Instruments and Methods of Acoustic Studies of Speech," *Proc. Eighth Int. Cong. Linguistics.* (Oslo 1958), pp. 282-358; F. S. Cooper, "Speech Synthesizers," *Proc. Fourth Int. Cong. Phonetic Sciences* (The Hague, 1962), pp. 3-13; J. L. Flanagan, *Speech Analysis Synthesis and Perception* . (New York, 1965) pp. 167-191. (References obtained from Mattingly 1968).

[17] The adjective "connected" is important in this context as there exist significant early works which dealt with speech as an unconnected phenomenon which will not be discussed here. For further information please see the prior footnote.

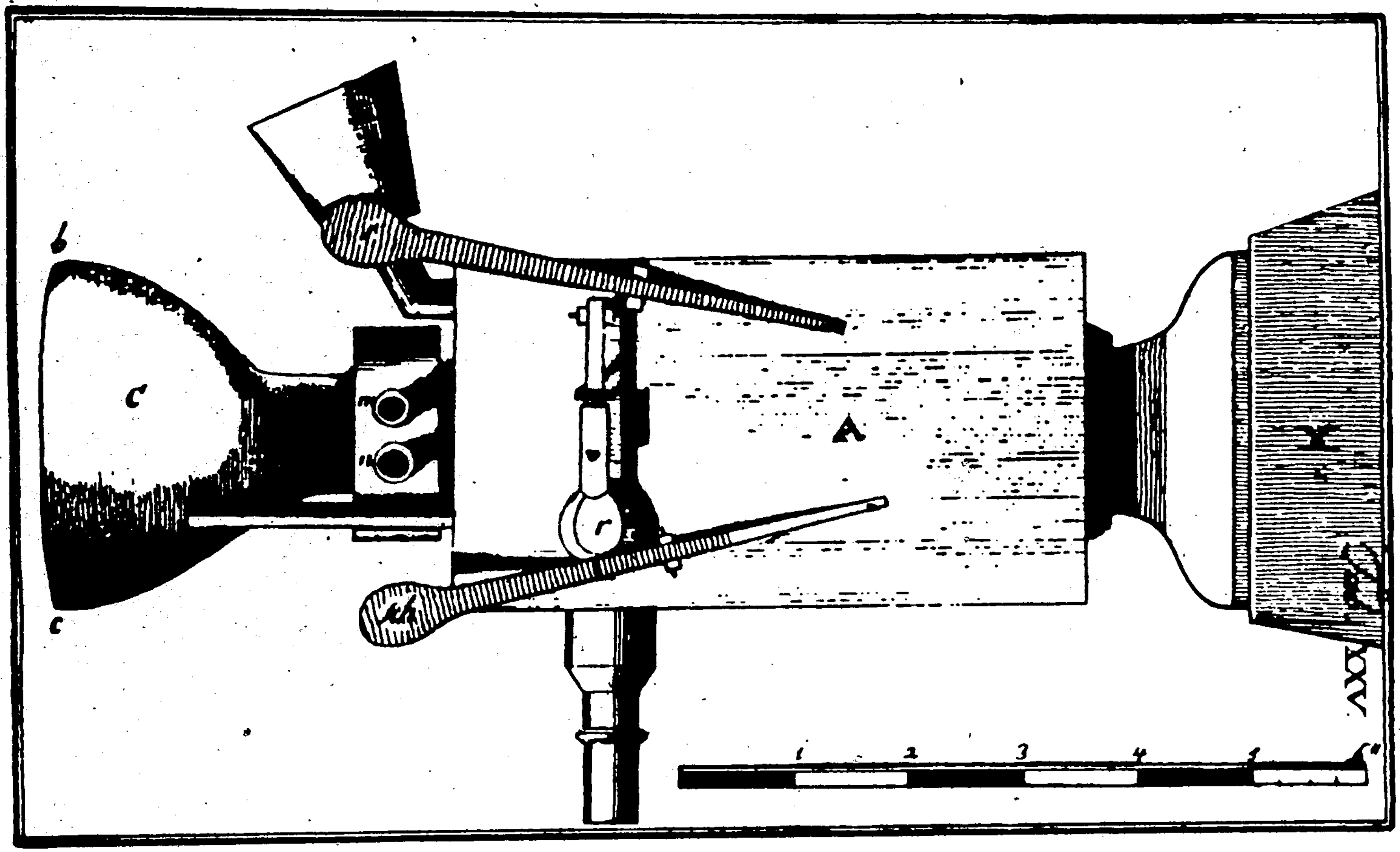


Figure 2.5
von Kempelen's synthesizer, reproduced from his
Mechanismus der menschlichen Sprache (1791).

It is instructive to note that von Kempelen apparently regarded speech as merely an articulatory phenomenon [18]. Further, many of the articulatory facts of which von Kempelen was aware could not be incorporated into the design of the synthesizer. The only means of dynamic control over the synthesizer was via a human operator and while the operator was synthesizing speech by rule, the rules could not readily be made explicit.

In the century following von Kempelen, many manually operated, mechanical speech synthesizers were created. There was Wheatstone's copy of von Kempelen's device, Faber's "Euphonia", and Paget's artificial larynx called a cheirophone (Mattingly, 1968).

An interesting anecdote is that in the late 1800's, a young experimenter from Edinburgh Scotland had a chance to view the copy of von Kempelen's device as constructed by Wheatstone. This prompted young Alexander Graham Bell to construct his own model. It was based on an actual mold of a human skull with the vocal apparatus modelled using soft cotton batting and rubber. The necessarily movable portions of the vocal apparatus were controlled by levers. The vocal cords were simulated by passing air through a slotted rubber membrane. Cater (1983) states that

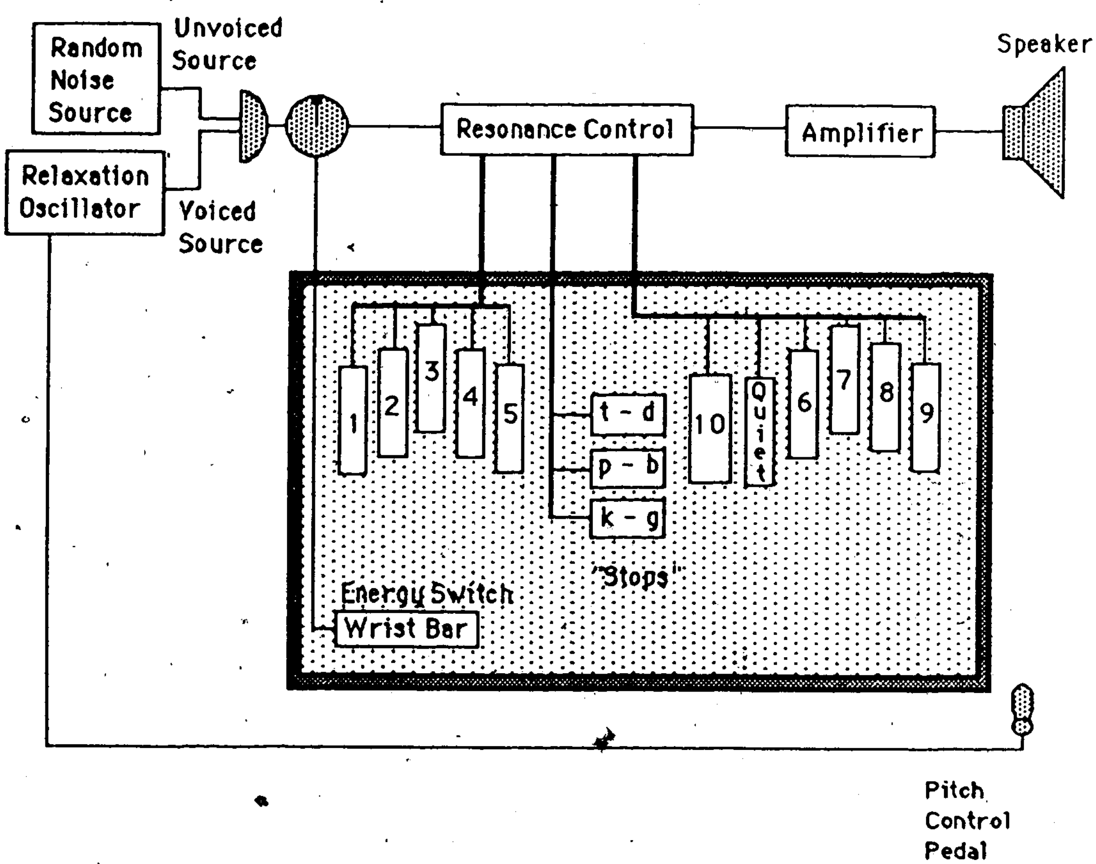
"Mr. Bell claimed that the device could speak vowels, nasals and even as he gained more experience, simple connected phrases."

The first electrical analog to human speech was created in 1922 by J. Q. Stewart, but it could not produce connected speech. For a more complete description please see Cater (1983).

The next significant speech synthesis device, according to Mattingly (1968) was the electrical Voder which was built by Dudley during the period 1937 to 1938 while in the employ of Bell Laboratories. It is described by Mattingly (1968) as being essentially a Vocoder [19] modified to facilitate manual operation. (See figure 2.6, after Mattingly, 1968). The Voder was capable of adjusting the amplitude of the output through the "quiet key". With the three stop-consonant keys for the three voiced/unvoiced pairs (b/p), (d/t), and (g/k) the correct

---

[18] Speech has been regarded in the past and present as a cognitive, auditory, acoustic, and neuromotor phenomenon as well.
[19] See H. Dudley, 1939. "The Vocoder", *Bell Labs. Record* 18:122-126.



Block Diagram of Dudley's Voder

Figure 2.6

(After Mattingly, 1968)

sequence of acoustic events for stops (abrupt excitation cut-off, silence, burst, and resumption of excitation) could be generated automatically (Mattingly, 1968). Dudley and his collegues demonstrated the Voder at the 1940 World's Fair in San Francisco. Listeners were cued by a human speaker who "conversed" with the Voder. Dudley noted that the cueing was an important precaution due to the unusual quality and unavoidable imperfections of the voice which the audience would be hearing for the first time.

Dudley (1939) made two observations which are worth noting because they are very relevant to even today's methods of speech synthesis. Firstly, on discovering the best way to synthesize a sound, he said,

> "the most fruitful method of attack was to search for the desired sounds by manipulation guided by the ear;"

and secondly, regarding the ability of the listener to extract meaning from synthetically formed speech after prolonged exposure to it,

> "the listener becomes expert at interpreting badly formed words and ceases to be critical."

The systems of von Kempelen and Dudley both required a human operator who had to be trained in the operation of the device. In both cases, the operator's training implicitly included rules for synthesizing speech which were never made explicit. Thus both systems produced speech-by-rule but there were some fundamental differences between the two of them.

The first difference is basically due to the differing technologies available to each researcher. Dudley's Voder was primarily electrical whereas von Kempelen's work was acoustical. This electrical implementation had certain consequences. Changes in design were more readily and quickly implemented. Further, many of the articulatory facts pertaining to synthetic speech which von Kempelen was forced to ignore, could now be incorporated into the design of the Voder as a result of using inherently well determined electrical circuits with easily modifed properties.

The second difference is due to each researcher's frame of reference rather than technology. Dudley, a telephone engineer, tended to regard speech as an acoustic as opposed to an articulatory phenomenon. This led to an acoustic model which embodied only the most obvious speech signal characteristics. Both the model and its control strategy were relatively simple; consequently the implicit synthesis strategy and synthesis rules became increasingly complex. The result was that a year or more of training with the Voder was required before intelligible speech could be produced by the operator.

The first speech synthesis-by-rule device that made its rules explicit was a direct result of the invention of the spectrograph in the 1940's by Potter of Bell Laboratories (Koenig, Dunn, Lacey; 1946). The spectrograph is an instrument which disperses sound waves into a spectrum which may be mapped as a spectrogram. The spectrogram is a graphical representation of speech in which the horizontal dimension represents time, the vertical dimension, frequency; and the density of inking of the picture, energy. Here the vocal tract formants appear as dark bars which rise and fall in frequency over time. It is reasonable to view the spectrograph as a visual tape recorder of speech. It is also reasonable to conceive of a device which would play back the "stored" speech. Potter designed and built such a device to achieve a means of demonstrating that his spectrograms preserved the essential speech information. This speech synthesis device was a landmark in the sense that for the first time, the synthesis of conjoined speech was not a transient event, but rather a controlled result of a spectrogram. Cooper of Haskins Laboratories saw the experimental value of such a device and built a research version called the "Pattern Playback".

A set of explicit synthesis rules was developed by Frances Ingemann (1957) which could be used to edit or produce an original spectrogram which was used as input to the Pattern Playback. The refinement of the complex and sophisticated rule statements proved difficult and their application to produce a precisely specified utterance proved laborious. In spite of these difficulties, and the inherent limitations of the Pattern Playback.[20] for the first time an explicit set of rules was available so that an utterance could be synthesized and stored; [21] a spectrogram created by rule could be checked; and speech created by different versions of the same rule could be compared before it was heard. The development of Ingemann's rules also clearly defined the concept of speech synthesis-by-rule as a research objective.

The idea of "Compiled Speech" was examined in the 1950's and the 1960's by a host of researchers. This concept was prompted by the advent of the magnetic tape recorder. The compilation process consisted of taking prerecorded segments of natural speech and conjoining them to form an utterance. Harris (1953) used segments corresponding to phonemes and syllables but the resulting speech was of disappointingly poor quality. One of the reasons for this poor performance is that acoustic cues for phonemes overlap in time and any attempts to build an uttereance from isolated segments are forced to ignore this reality. In order to deal with this inherent problem more effectively, segments consisting of the last half of one phone and the first half of another were used in the dyadic synthesis attempt of Peterson, Wong, and Silvertsen (1953). There are two major downfalls with this approach. The first problem is that the larger the segments, the better the results, and to retain the ability to synthesize arbritrary passages, the inventory of segments now becomes combinatorially large (Gaitenby, 1961). The second problem is that the manner in which prosodics are to be handled remains totally unresolved (Mattingly, 1968).

A promising variation on the concept of compiled speech involved the use of synthesized, rather than natural speech segments. The ability to control the production of the segments in a very explicit fashion opened up a way of eliminating some of the difficulties mentioned earlier. This was first examined by Estes and his associates (1964) and later by Gaitenby (1967). Unfortunately, either the synthetic segments required a large number of

---

[20] There was no means to specify hiss excitation or to vary the fundamental frequency of the buzz excitation.
[21] The rules described how to paint a spectrogram.

parameters to be specified for each moment of speech, making the task of synthesis highly complex, or a great deal of simplification was made and the auditory quality of the utterance suffered.

Subsequent work took two general directions. If a person wished to study the lowest levels of speech synthesis or retain control over all aspects of synthetic speech production, he tended toward building his own speech synthesizer. Alternately, if a person wished to study the higher level aspects of speech synthesis such as prosodics and the use of stress, then he tended toward the use of the newly available commercial devices, despite their unsatisfactory enunciation.

2.4 Contemporary Speech Synthesis Systems

There are many text-to-speech systems in existence, both in research and commercial environments. However, virtually all existing systems use a speech generation technique which can be categorized as one of three basic types.[22] The methods of conversion of parametrically described speech to its audible analog are, in order of complexity of implementation:

1. Waveform encoding/decoding for direct speech reconstruction (WED).
2. Phoneme specified formant speech synthesis (FS).
3. Linear predictive coding for mathematical speech reconstruction (LPC).

Each method is capable of generating understandable speech. Exactly which method is used, depends on many factors which have been mentioned earlier in this thesis.

The three methods of speech generation differ primarily in two areas. The first area is the amount and rate of information (baud rate or bits per second) required for the method to convert a described word to its audible analog. Speech is a complex event. To describe such an

---

[22] Cater (1983) states that other possible techniques exist for speech synthesis such as Walsh function synthesis, direct Fourier synthesis, and signal correlation and partial autocorrelation (PARCOR). These techniques are not used in any text-to-speech system that is of interest with reference to this thesis, therefore they have not been discussed.

event adequately, a great deal of information is needed. Generally speaking, when more information is used to form a word, the quality of the output is correspondingly higher (more human like). This is because the more information that is provided for a synthesis technique, the more accurate the description of the acoustic event can become. (See Table 2.1) Each method of speech generation requires input within a certain range of baud rates.

The second manner in which these speech generation techniques differ is the source of the information used to create the synthetic speech output. Synthesizers of both the waveform encoding type and the linear predictive type are based ultimately on human speech. The speech is compressed and encoded through a variety of methods. While these two techniques differ greatly in their implementations, the important point is that they both generally require a prespoken vocabulary for speech generation. [23] On the other hand, formant synthesis systems need not (but can and do) use processed human speech directly as their information source. This total independence of formant synthesis systems from human speech makes them much more popular in a text-to-speech environment where truly synthetic speech is the goal.

Given the objectives of this research project, as stated earlier in Chapter 1, the most acceptable method of truly synthetic speech generation is one grounded on the ideas of phoneme specified formant synthesis. The technique of waveform encoding is not acceptable on at least two grounds. Firstly, the technique does not synthesize speech, but rather reconstructs it from prerecorded natural speech. Secondly, this technique is not grounded on the concept of generalizable pronunciation rules. The same criticisms apply to linear predictive coding systems given that they are based on natural speech. [24] A similar view is taken by Cater (1983), who concludes that only speech generated using a system which does not depend directly on prespoken human speech, can be truly called a synthetic speech generation system.

----

[23] It is possible for systems categorized as LPC synthesizers to operate in a text-to-speech environment (Cater, 1983). An example of the type of hardware used is described by Caldwell (1979, 1980).

[24] The optimal functioning of a LPC system requires prerecorded human speech as the medium from which to build its output. There are however LPC systems which synthetically composed signals as the medium from which to build speech.

Synthesis Technique Summary Data

Synthesis Technique	Bit Rate per Second	Storage Required in Bytes for "Hello"
FS	100-800	4 to 30
LPC	1200-1500	45 to 188
WED	16,000-120,000	600 to 4500

Table 2.1 (After Cater, 1983)

Information Encoding Techniques

A complete description of the theory behind these three speech synthesis techniques will not be attempted here. [25] However, it would be remiss not to introduce the basic concepts of linear predictive coding in addition to discussing the ideas behind formant synthesis.

Formant synthesis is normally regarded as residing in the frequency control domain (Ciarcia, 1981). Frequency domain synthesis is considered to be the classic approach to the problem of speech generation. It has been actively researched over the past several decades (Costello and Mozer, 1984). Synthesis schemes of this sort model speech as a combination of two types of source excitations (i.e. turbulent air and vocal cord vibrations) together with a substantially larger number of output filter states wich represent the resonant states of the vocal tract. Compression of data is achieved by storing the filter and vocal excitation parameters instead of the original waveform. An algorithm of relatively high complexity utilizing a multi-pole digital or analog filter is usually used to transform the stored frequency

[25] For an introduction to these techniques, please refer to: Koehler and Mackey, 1984; Ciarcia, 1981, 1982, 1983; Smith, 1984; Kaplan and Lerner, 1985. For a more complete review of these techniques and their potential uses and general viability as research tools, please see: Witten, 1982; Cater, 1983; Bristow, 1984.

domain information into an audio signal in the time domain to approximate the original waveform. This algorithm is usually realized in terms of integrated circuitry (Rabiner, *et. al.*, 1971) however, a software implementation of a multi-pole digital filter does exist and has been used successfully (Klatt, 1980).

In contrast, WED is normally considered to be a time domain synthesis technique (Witten, 1982). Methods of this type store sound segments as compressed representations of speech waveforms viewed as functions of time. Because the stored information is already in the time domain, no filter is necessary, and the synthesizer merely unpacks the information and sends it to the hardware to produce the output speech signal. The advantange of these speech waveform compression techniques is not their limited hardware requirements,

> "...but rather the analysis that enables the speech waveforms to be stored in such a highly compressed form." (Costello and Mozer, 1984).

Linear Predictive Coding Synthesis

LPC was developed and popularized in the early 1970's (Makhoul, 1984; Witten, 1982). The technique is a method of compressing the storage requirements of digitized speech. It is based on the assumption that the sound generated at given time *T* is a continuation of the sound generated at time *T-1*. The speech sample at time *T* is predictable based on a weighted average (linear combination) of speech samples at a small number of prior instants. By removing the natural redundancies present in speech, LPC reduces the required number of bits to record a second of speech by as much as 98.5% when compared to purely digitized speech [26].

The LPC analysis starts with an actual recording of the words or sound segments to be reproduced. This recording is first sampled at a fixed rate to convert the recorded waveform into digitized data. The data is then compressed to extract source information, amplitude, and

[26] To reproduce one second of speech, digitization techniques require 96000 bits for storage (no compression). LPC requires only 1200 bits for storage (Koehler and Mackey, 1984).

multi-stage lattice filter parameters. The amplitude is a measure of the energy or the loudness of the utterance. The source information indicates the state of the vocal cords (vibrating or still) and if appropriate, the pitch or frequency at which they are vibrating. The filter parameters relate the relative placement of the teeth, lips, and tongue in the vocal tract. This information is used to reconstruct the utterance based on a mathematical model of the human vocal tract.

The technique's mathematical model represents the vocal tract as an acoustic wave guide comprised of between 10 and 16 uniform tube sections which have their cross sectional areas change dynamically during speech. These "conceptual tubes" are represented by the programmed activity of the multistage lattice filters which are the heart of the mathematical model of the vocal tract (Ciarcia, 1981) [27]. Digitally represented sound sources excite these conceptual tubes creating pressure waves which advance and retreat within the tube (Kaplan and Lerner, 1985). See table 2.2 for current examples of systems which use this technique.

LPC can be seen as being akin to FS in that LPC can operate in the frequency control domain and use similar hardware to emulate the human vocal tract (Ciarcia, 1981). Although LPC is primarily a method of coding information in the time domain, it can be used to generate frequency domain parameters such as formant frequencies, amplitude, and bandwidth (Witten, 1982).

The differences arise in that the parameters for LPC are stored as multi-pole digital or analog filter parameters, amplitude or gain settings, and excitation frequencies (source information).

Formant Synthesis

Formant synthesis consists basically of modelling the natural resonances of the human vocal

----

[27] The lattice filters are responsible for the generation of the required formants used in actual audio output. On the basis of this, some authors in the popular literature have classified LPC based speech generation systems as formant synthesizers.

Currently available LPC type Speech Synthesizers.

Device	Synthesis Type	Approx. Cost
The Texas Instruments Inc.(TI) Speak and Spell.	10-Pole LPC	$60
The Texas Instruments Inc.(TI) TMS 5200	10-Pole LPC	$80
Hitachi HD61885 and HD38880	10-Pole PARCOR	N/A
Speech Technology Corp. M410	12-Pole LPC	$185
Street Electronics Echo II	10-Pole LPC	$200
Speech Technology Corp. VR/S100	12-Pole LPC	$325
Street Electronics Echo-GP	10-Pole LPC	$370
Telesensory Speech Systems Speech 1000	12-Pole LPC	$1200
Telesensory Speech Systems SP1020	12-Pole LPC	$2500
The Texas Instruments Inc.(TI) PASS.	LPC-Encoder	$15,000

Table 2.2 (After Cater, 1983)

tract called formants. This technique depends upon analysis of specified speech segments to define them in terms of at least the three lowest formants (i.e. F1, F2, F3). Speech segments of any size (phrases, words, or phones) may be analyzed. In addition to the three formants, information regarding the pitch, amplitude, duration, and respective bandwidths of the three formants, may be extracted. This information is then stored so that it may be accessed in terms of the original sound segment. A library of sound segments is thus formed.

This technique may be used in a system which concatenates the information regarding a number of sound segments to form a meaningful unit (e.g. a word, or words concatenated to form a phrase) which is then used to drive a hardware speech synthesizer. Exactly what type of information is stored clearly depends on what information is needed to synthesize a sound segment using the selected hardware.

The most common example of the formant synthesis technique is a phoneme specified synthesis system. This type of system makes the assumption that phonemes constitute the basic sound units of human speech. This assumption is based on the fact that a phoneme is defined as being a member of the set of the smallest units of speech that serve to distinguish one utterance from another in a language or dialect (i.e. the *p* of *pin* and the *b* of *bin* are two different phonemes). Theoretically, if all of the linguistically acknowledged phonemes are defined, then by linking them appropriately, unlimited, natural, intelligible vocabularies may be achieved [28].

The result of speech through phoneme synthesis is an electronic voice which varies in quality and intelligibility according to the extra parameters (pitch; duration; amplitude; higher order formants; dynamics of change of pitch, duration, and amplitude) which are used to shape each phoneme. The data rate associated with this technique also varies according to the extra information which is provided. However, Ciarcia (1981) states that

> "In most cases, the electronic voice generated is quite intelligible, but it may have a mechanical quality ... with a data rate less than 400 BPS (bits per second)".

The reason for this is that phonemes are conceptual objects which are never realized in human speech. What actually exists are allophones which are defined as one of the variant sounds forming a phoneme (i.e. the aspirated *p* of *pin* and the unaspirated *p* of *spin* are allophones of the phoneme *p*.) It is enlightening to view allophones as variants on a theme, or as approximations to a target. The manner in which the allophones of a phoneme vary is related to their context [29]. Thus if only phonemes are specified, the resulting speech is unsatisfactory. To

----

[28] While this is theoretically possible, it has yet to be fully realized in practice. Phrases such as "linking them appropriately" and words such as "natural" and "intelligible" imply a great many things.

[29] Context here refers to the phones surrounding the allophone in question. For detailed discussion, the reader is referred to the linguistic literature. Alternatively, Jassem and Nolan (1984) offer an introduction for the non-linguist.

achieve higher quality speech, it becomes necessary to specify the correct allophone in the correct context. There is no phoneme synthesis system known to this author which allows all known allophones to be specified.

See table 2.3 for current examples of formant sysnthesis systems. The Votrax type'n talk, Intex-talker, and Microvox text-to-speech synthesizer are based on the Votrax SC-01A voice synthesis chip. The Text to Speech system from Sweet Micro Systems Inc. is based on the SSI-263A chip. The Dectalk and the Prose 2000 units use a more advanced phoneme formant synthesis technique. Both of these commercial systems are based on the MI talk-79 research system of Allen's (1979) and all will be discussed in more detail later in the thesis. The Votrax Division of Federal Screw Works SC-O1A chip, and the Silicon Systems Incorporated SSI-263A chip both use a table look up procedure (as described earlier) to generate the information for a series of predefined sound segments. The write ups for these systems in the popular literature call these sound segments phonemes. This use of the term phoneme only loosely corresponds to its normal use in a linguistic sense.

In the opinion of the author, with respect to what is being investigated here, the SSI-263A chip is the output device of choice in this comparison. This is the device currently being used by this research project as its primary output device. The reasons behind this choice will be described in greater detail later.

Parameter Generation

The parameter generation module (PGM) of a speech synthesis system serves as the interface between information encoding techniques and synthesizer hardware. The PGM receives a phonetic version of the original input string. This phonetic string has most often been syntactically disambiguated and marked for segmental and suprasegmental features. The task of the PGM is to take this information as input and decode or translate it so that it can be re-expressed in terms of the parameters needed to drive the particular synthesizer hardware that

Phoneme specifiable, text-to-speech formant synthesis systems.

Device	Approx. Cost
The Sweet Micro Systems Inc. "Text-to-Speech System".	$100
The Intex "Talker".	$150
The Microvox "Text-to-Speech synthesizer".	$150
Votrax "Type and Talk".	$375
The Speech Plus "Prose 2000".	$3,500
The Digital Equipment Corporations "Dectalk".	$4500
Kurzweil Reading Machine (KRM).	$30,000

Table 2.3 (After Cater, 1983)

is being used as the output device.

The algorithm used to decode or translate the phonetic string to synthesizer parameters can vary from trivial to complex. The complexity of the algorithm is directly related to the number of changes that must be made to the phonetic input string to have it become acceptable as output to a speech synthesizer. The number and type of these changes could be quantified to give a measure of the complexity of the PGM of each speech synthesis system. This however, is not the point. Even in cases where the complexity of different algorithms was similar, the algorithms would not be interchangeable. The simple reason for this is that there is no way (or reason) to standardize the characteristics of either the input or the output to the PGM. This makes each implementation of a PGM application specific. Additionally, in the applications that are relevant to the topic of this thesis (e.g. those using high level synthesis hardware), the PGMs can generally be implemented very easily based on the idea of translation tables. These two reasons suggest to the author that there is no need to discuss PGMs in isolation in any further detail.

Synthesizer Hardware

A large number of speech synthesizers have been described in the last five decades [30]. Klatt (1980) divides these recent synthesizers into two broad categories. These are articulatory synthesizers and electronic resonance synthesizers.

Articulatory Synthesizers

Articulatory synthesizers attempt to model the mechanical motions of the human vocal tract faithfully. The motions of articulators, the resultant volume velocity distributions and sound pressure in the lungs, larynx, vocal and nasal tracts are all kept track of (Flanagan, Ishizaka, and Shipley, 1975). Witten (1982) notes that although articulatory synthesis has the potential for very high quality speech, owing to the inherent difficulty of modelling the coarticulation effects caused by tongue and jaw inertia, it has yet to be realized. Since this type of synthesizer is not readily usable in a text-to-speech environment, it will not be discussed further.

Electronic Resonance Synthesizers

Electronic resonance synthesizers operate by modelling speech in the acoustic domain. The acoustic view of speech may be summarized by stating that:

> "... a simple, reasonable and approximate model of speech generation includes a time-varying filter, whose resonances and antiresonances can change continuously to simulate the vocal tract transmission, and whose excitation is derived from two kinds of signal sources: a periodic pulse generator of variable period to simulate voiced

----

[30] Dudley, Riesz, and Watkins, 1939; Cooper, Liberman, and Borst, 1951; Lawrence, 1953; Stevens, Bastide, and Smith, 1955; Fant, 1959; Fant and Martony, 1962; Flanagan, Coker, and Bird, 1962; Holmes, Mattingly, and Shearme, 1964; Epstein, 1965; Scott, Glace, and Mattingly, 1966; Liljencrants, 1968; Rabiner, *et. al.*, 1971; Klatt, 1972; Holmes, 1973; Klatt, 1980; (From Klatt, 1980).

sounds, and a broad band noise generator to simulate voiceless sounds." (Flanagan, 1984).

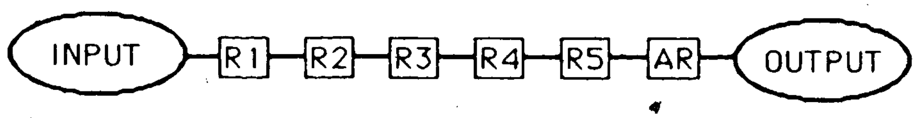
The channel vocoder was the first method to ever take advantage of the source and filter model for speech coding. The first example of the method was implemented by Dudley in 1939. The word *vocoder* is actually a contraction of the phrase *voice coder* . This method employs a bank of fixed bandpass resonance filters whose amplitudes are controlled in an effort to model speech by amplifying only those resonances which are found in the target speech. The use of fixed, as opposed to variable, bandpass resonance filters tends to limit the accuracy of speech reproduction attempts. The greatest limitation of device is the fact that it is driven by spectrographs. Spectrographs are a very difficult medium in which to specify controlling parameters well and consistently. Witten (1982) reports that this type of synthesizer is not normally used in current text-to-speech systems because of its inherently poor quality speech and its spectrograph input medium. Channel vocoders use the same type of sound sources as formant synthesizers which are discussed subsequently.

Recently, the most successful electronic resonance synthesizers have been formant synthesizers. Formant synthesizers are normally only concerned with the production of the first 3 formants (i.e. F1, F2, F3). The formants are created by passing a broad source signal generated by an excitation source through a few parametrically controlled filters. Periodic signals are used for voiced sounds and aperiodic signals are used for unvoiced sounds (Kaplan and Lerner, 1985). This approach was pioneered by Fant in 1960 (Fant, 1960).

Two general configurations of formant synthesizers are common. One is called a cascade formant synthesizer (see Figure 2.7) and the other is called a parallel formant synthesizer (see Figure 2.8) They differ in the arrangement of the formant resonators. An amalgamation or "best of both worlds" synthesizer has been proposed by Klatt (1980). He calls this a cascade/parallel formant synthesizer (see Figure 2.9). Each approach has is advantages and disadvantages [31].

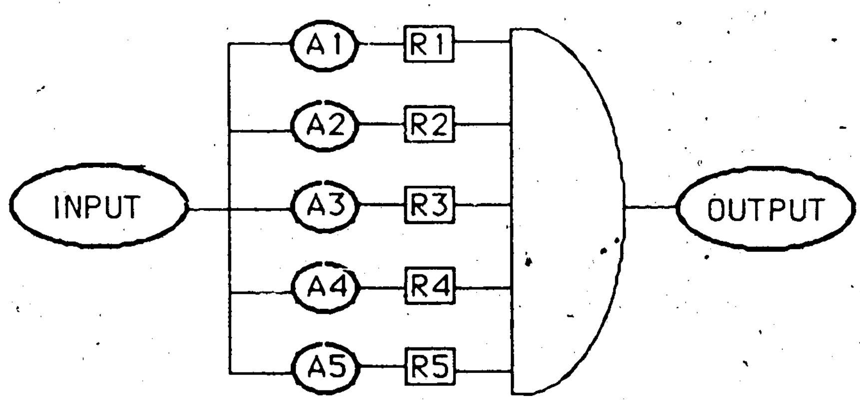
---

[31] See Witten (1982) for a mathematical discussion of the merits of each approach.



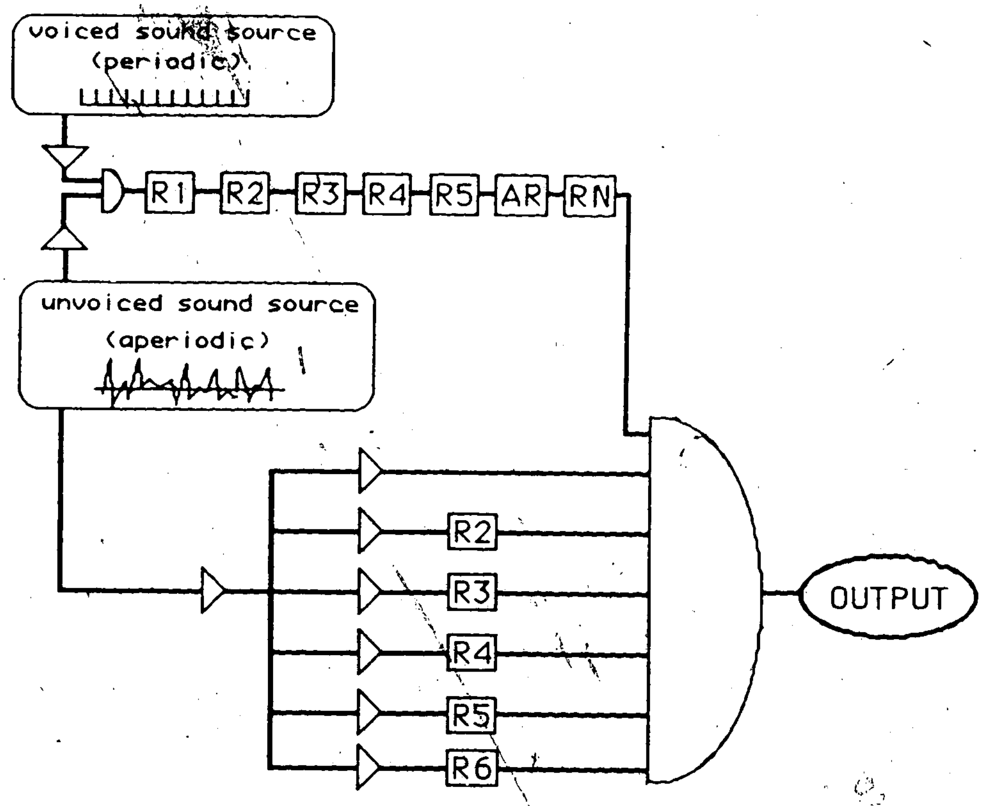
Cascade Synthesizer

Figure 2.7



Parallel Synthesizer

Figure 2.8



Combined Parallel/Cascade
Speech Synthesizer

Figure 2.9

(After Kaplan & Lerner, 1985)

Parallel formant synthesizers

The parallel formant synthesizer connects in parallel the formant resonators which simulate the transfer function of the vocal tract. Normally, each resonator is preceded by a gain modulator which determines the relative amplitude of a particular formant in output spectrum of both the voiceless and voiced speech sounds. This type of synthesizer is considered particularly useful by Klatt (1980)

> "... for generating stimuli which violate the normal amplitude relations between formants or if one wishes to generate, e.g., single-formant patterns."

Klatt goes on to say that the parallel configuration also facilitates the generation of fricatives and plosive bursts as their sound source is above the larynx.

Cascade formant synthesizers

The cascade formant synthesizer produces sonorants through a series of cascade-linked formant resonators. However, some parallel path must be included because of the need for fricatives and plosives. This results in a conceptually more complex implementation. The advantage of the cascade configuration is that individual amplitude controls are not needed to control the relative amplitudes of formant peaks. Additionally, cascade-linked resonators provide a superior model of the vocal tract transfer function during the production of non-nasal sonorants (Flanagan, 1957).

Parallel/cascade formant synthesizers

Klatt's parallel/cascade synthesizer utilizes a cascaded as well as a parallel bank of resonators to simulate the resonances and antiresonances of the mouth, nose and throat. Both periodic (voiced) and aperiodic (voiceless) sources may be fed into the cascaded resonators. These are used to produce such voiced sounds as those produced by the vowels, and the phonemes *v* and *z*. Only the aperiodic or "white noise" sound source may be fed into the parallel resonators.

They are responsible for the production of the unvoiced speech sounds such as the phonemes *f* and *s* (Kaplan and Lerner, 1985).

3. A Modular Systems Approach to Speech Synthesis

The speech synthesis system designed by this author, "Talker", was conceived as a research oriented text-to-speech system The design ideas are based on the conventional control blocks presented in figure 1.1. Figure 3.1 presents the modular grouping of routines chosen by this author. The following sections first discuss Talker as a whole and then each of its components in relation to the design goals and the implementation decisions of the system.

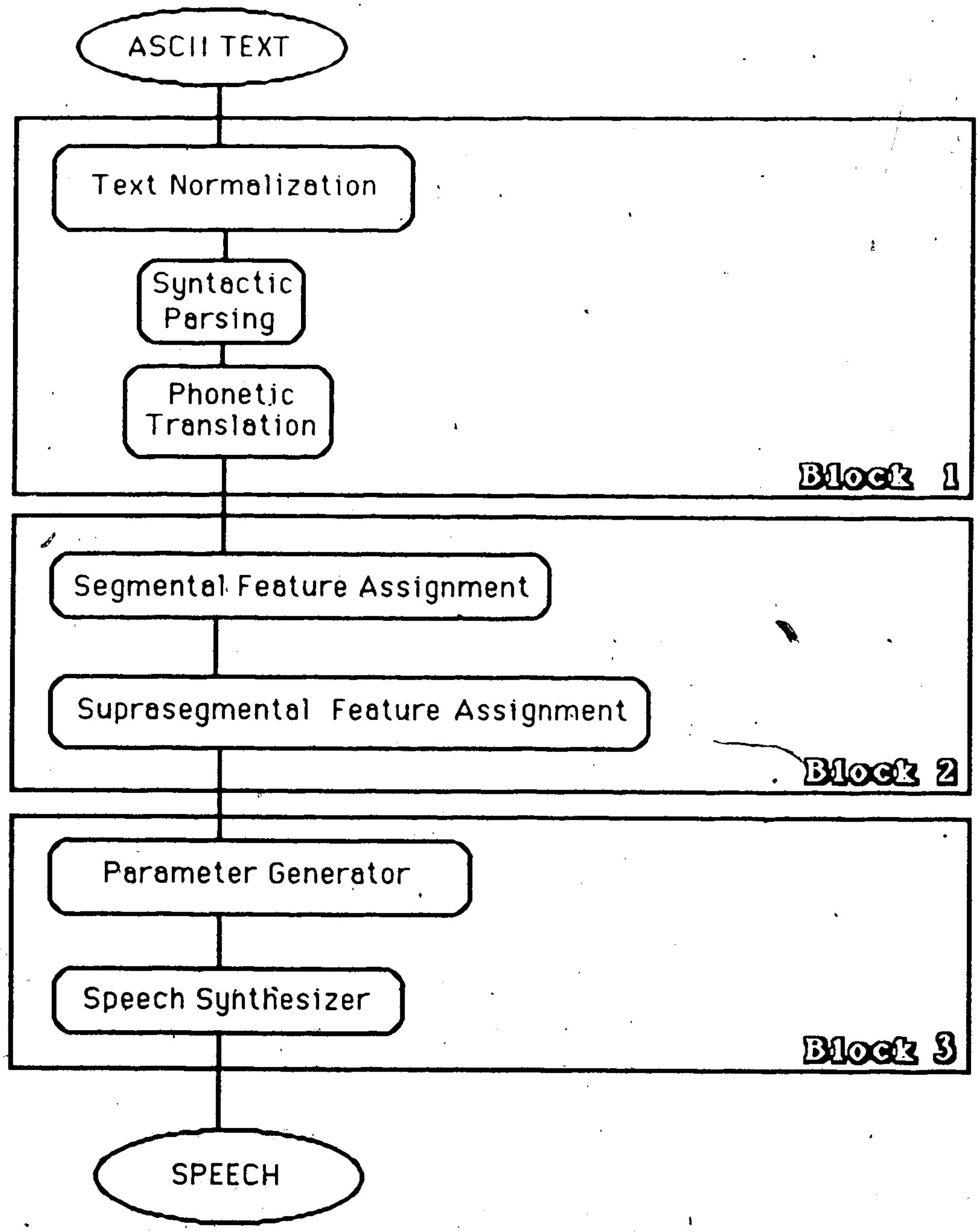
In the design of a text-to-speech system, many decisions must be made. The design consideration space is so large as to preclude an optimal solution. Fortunately, every design option does not require detailed consideration once criteria have been established. The sequential nature of the top-down design process causes lower level decisions to fall out naturally once higher level choices are resolved. Often the basis for judging the acceptability of a decision is unclear. For example, commercial text-to-speech systems are designed for fast operation and inexpensive implementation. Systems such as these tend to be found in expensive dedicated hardware, peripheral to the host system. Conversely, research systems are concerned with cleanly interfaced logical components and the retention/display of information calculated or gathered in the synthesis procedure. They tend to be more expensive specialized hardware and software applications integral to the host system. In each type of system, different design criteria result in different text-to-speech systems.

The design alternatives of Talker were considered under the following criteria. The system should be minimally restricted, fast, and accurate. Levinson (1980) makes the point that a system of this nature

> "... should be modular and simple to minimize the work involved in its construction, debugging, and tuning. The task also should initially be no more complex than necessary to present an interesting set of problems."

There was a need to complete this system in a finite time span. A top level concern was the need to allow for the incorporation of a syntactic parser of the English language [32]. The

[32] This parser was to be developed concurrently by Dr. L. K. Schubert and was not available at the time of implementation.



Block Diagram of the
Operation of Talker

Figure 3.1

examination of the areas of speech synthesis which are the least explored and most interesting such as prosodics and semantic understanding was considered desirable. The testing and tuning of the system was important so that it could be evaluated in a quantitative manner to allow further comparison with other existing systems and future efforts. A further consideration, relating to the potential accuracy and relative speed of the system, is that possibly relevant information should not be discarded. This information retention principle is especially important in an experimental system like Talker. Finally, this entire system had to be implemented under very tight budgetary restrictions making expensive hardware and software purchases out of the question. It was plain to see from the outset that these criteria are not wholly compatible and that sizable tradeoffs had to be made.

The tradeoffs involved in this research took many forms. For example, the prosodic aspects of synthetic speech were examined in a theoretical framework only, as only a rudimentary prosodic processing control block was implemented. The rule base of Talker was not optimized towards any criteria other than correctness (such as maximum generality, or minimum number) due to time constraints. Kaplan and Lerner sum up the problems of tuning in synthetic speech as follows:

> "Cecil Coker, speech synthesis researcher at Bell Labs, reported that one could easily spend a month on tuning or optimizing the vocal tract parameters to make just one phoneme sound natural ... -only to discover later an additional phoneme combination in which the phoneme in question sounded artificial." (Kaplan and Lerner, 1985).

Coker's observation applies equally well to interdependent letter-to-phoneme rules. For reasons of time, the quantitative testing and tuning were sacrificed in favor of qualitative judgements. A decision was made to maximize the opportunities for utilizing the knowledge gathered in the synthesis process by minimally "throwing away" the information. Levinson (1980) makes the point that this information retention principle

> "... is essentially a negative principle; one which warns about potentially restrictive choices, rather than guiding the designer to a particular decision".

It is this principle which led to the current data structures used in the system because it suggested that information such as the original orthographic input, its syntactic description and its phonemic equivalent might all be useful reference information to the prosodic routines. Unfortunately the retention of information, and the concomitant complex data structures, is in direct conflict with an earlier mentioned criterion of simplicity. Nevertheless, the decision was made to retain information whenever the complexity that it engendered was not excessive.

The first block of Figure 3.1 is composed of three parts. These are, in order, the normalization modules, the syntactic parser, and the phonetic translation module. These three modules together serve to generate the basic phonetic string equivalent to the orthographic input.

3.1 Normalization

Normalization is the process which causes the input string to conform to certain expected characteristics. While the input string can be any string of ASCII [33] characters, only a subset of the entire ASCII set makes sense in this particular application. It is therefore, part of the job of the normalization process to remove any of the meaningless characters from the input stream [34]. Further, normalization forces the alphabetic characters to a single case (i.e. upper or lower case). This reduces the number of possibilities the remainder of the program must investigate. Some text-to-speech systems (Prose 2000) include the expansion of abbreviations (and possibly contractions) in the normalization process. By way of example, see Table 3.1 for a list of possible input strings and their normalized equivalents.

Conventional text-to-speech systems deal with normalization either by ignoring it or treating it as a separate process which may be considered as a form of pre-processing. If normalization is ignored, then a great deal of the robustness of the system is lost. If normalization is approached as a pre-processing task, then it cannot be incorporated as tightly into the system design as is possible.

---

[33] American Standard Code for Information Interchange
[34] Form feeds, line feeds, and miscellaneous control characters.

Examples of Normalized Text

Before Normalization	After Normalization
Mrs.	MISSUS
Ms.	MIZ
$17.23	SEVENTEEN DOLLARS AND TWENTY THREE CENTS
St. Boniface St.	SAINT BONIFACE STREET

Table 3.1

3.1.1 Design and Implementation of the Normalization Module

A case will be made for the consideration of the normalization process right from the inception of the system design. This is because decisions regarding normalization have consequences for the structure of the lexicon and the rule base of the completed system.

The pre-processing of non-conforming input would normally be viewed as a string substitution procedure characterized by parsing, lexical look-up, and the replacement of a string by its normalized equivalent. This approach is logical in isolation, but unreasonable when viewed as part of a text-to-speech system. The problem is that the lexicon is accessed first for the normalized equivalent of the abbreviated "word", and later for the phonemic description of the normalized text. Similarily, this approach can lead to multiple parses of the input string, at least once for abbreviations, a second time for the case conversion, and possibly a third time for the syntax analysis of the input string. Also, if only lexical entries are used to expand abbreviations, then how may the potentially infinite numeric expressions be handled?

This suggests that the expansion of abbreviations and numeric expressions should be dealt with through the same mechanism (discussed later in the paper) which deals with "regular" text strings, namely the phonetic translation module. This allows the trade off between lexical entries and letter-to-phone translation rules to be exploited and avoids potential redundancy.

This leaves conversion to upper case as the only preprocessing operation. Unlike expansion of abbreviations and numerics, this character-by-character operation leads to no redundancies in subsequent processing. One further wrinkle which was added to the implementation was the rearrangement of strings representing monetary amounts so that they will translate properly later (e.g., conversion of *$50.07* to *50$&7¢*). This latter wrinkle was added because it was possible to do with only forward parsing.

3.2 Syntactic Parsing

The syntactic parsing module (SPM) was conceived much before Talker itself. Its design and implementation was the responsibility of L.K. Schubert. It is viewed as a separate process which takes as input character strings, and returns the most reasonable parse tree of the string based on the generalized phrase structure grammar (GPSG) of Gazdar *et. al.* (1985). This type of syntactic information is very useful in the disambiguation of both the pronunciation and the selection of the stressed syllables in a word (see Table 3.2). Syntactic information is also very relevant and important to the optimal functioning of the prosodics section of the system (to be discussed later).

The design of a suitable parser is a very ambitious undertaking, since the parser must be error-tolerant, allow for an unlimited vocabulary (the categories of unknown words should be guessed), and choose among multiple alternative parses in a human-like fashion.

Examples of Word Stress Disambiguated through Syntactic Categorization

Word	Noun	Verb
subject	sub'-ject	sub-ject'
project	pro'-ject	pro-ject'
reject	re'-ject	re-ject'
present	pres'-ent	pre-sent'
rebel	reb'-el	re'-bel
affix	af'-fix	af-fix'
object	ob'-ject	ob-ject'
refuse	ref'-use	re-fuse'
survey	sur'-vey	sur-vey'

Table 3.2

3.2.1 Design of the Syntactic Parsing Module

The idea is that the input would have to be English but could be as varied as poetry, text, elliptical constructions, notes, or even a computer program. The parser proceeds in a bottom-up fashion utilizing its own lexicon to establish word classifications, tenses, etc. The parser is capable of morphological analysis of the words it investigates so that a small lexicon can be used. Also, the SPM can use the information gleaned from the affixes stripped from the words, to make conclusions with regard to the syntactic category of the word. This information, placed in context, with other words can resolve many of the ambiguities mentioned earlier in Chapter 1 and listed in Table 3.2.

The parser chooses amoung alternative word categorizations and phrase attachments using lexical preferences (e.g., a preference for categorizing *fat* as an adjective rather than a noun), a variant of Kimball's principle of Right Attachment (e.g., a preference for attaching the final prepositional phrase to the last verb in a sentene like *John bought the book which I had selected for Mary*), as well as, potentially, semantic and pragmatic principles (Schubert, 1984).

The output from the SPM would be a parse tree of the input string and/or possibly a summary of the relevant information taken from the tree. This information could take the form of syntactic (and, where appropriate, tense) categories of each word, the boundaries of phrases in the string (noun, verb, prepositional, etc.) as well as the overall tense of the sentence.

3.2.2 Implementation of the Syntactic Parsing Module

At the time of this writing (May, 1985), none of the SPM has been incorporated into Talker. It is currently being written in Pascal.

3.3 Phonetic Translation

The phonetic translation module (PTM) transforms a normalized orthographic string into its phonemic equivalent in a process described as *Constructive Synthesis* by Dilts (1984). There are myriad ways to produce a phonetic translation of a text string. Which way is "best" depends on the desired properties of the translation. Given the criteria defined earlier for Talker, there are three basic ways in which to construct a PTM. A PTM can be lexically based, rule based or based on the morphological composition of the words. These approaches are not totally distinct. Rather, they differ in their conceptions of what the basis of a PTM should be.

A lexically based PTM implies that the translation information is primarily stored in a lexicon. As was pointed out in Chapter 1, a PTM based exclusively on a lexicon will have a limited vocabulary. To ameliorate this problem, morphological analysis is used to decompose the many variant forms that a word may take. The advantage is that only the root word and not all of its derivatives must be stored in the lexicon. General pronunciation rules may be introduced to assist in properly pronouncing the whole word given its root pronunciation. This lexicon-based scheme tends to have the most severly limited vocabulary of the three approaches discussed here. It also requires the largest amount of storage. The advantage of this strategy is that when a word is translated, it is virtually always perfect (given the limitations of the device). Further, if it is not, the location which must be accessed to correct the problem is obvious.

A rule-based PTM is very much like a lexicon-based PTM with regard to the components which comprise each. The two approaches differ in the relative importance given to each component. The rule based PTM strategy is based on the conception that an extensive enough list of explicitly stated pronunciation rules will allow all words in English to be pronounced. This idea allows for a potentially unlimited vocabulary. Invariably, an exceptions lexicon must be included with such a rule base. It may explicitly take a form similar to the lexicon of the lexical based PTM. Alternately, if a rule exists which is applicable to only one word, then the word may be viewed as an entry into an implicit exceptions lexicon. A rule based PTM strategy requires the least storage of any of the three approaches. It also has the advantage of being able to provide a translation for virtually all words. The corollary of this is that proper pronunciation is not guaranteed. If the pronunciation is incorrect it is not an easy task to find the source of the error unless specific provisions have been made for such debugging information. Further, the tuning of such a set of rules is not straightforward because their scope of applicability is not always clear.

Allen (1976) put forth the idea that phonetic translations of English text could be best produced using a morph based approach. Allen allows that there are approximately 12,000 morphs in the English language and on average there are slightly less than 2 morphs per word. These morphs include prefixes (*con-*, *be-*, *mini-*), derivational suffixes which affect the meaning of the word (*-dom, -ship, -ness, -al*) and inflectional suffixes, which affect the grammatical role of the word (*-s, -ed, -ing*). Additionally, there are two kinds of root morphs:

1. free morphs which can stand alone, (*snow, boat, house*)
2. bound morphs which must combine with an adjacent morph (*-turb, -ceive, crimin-, -pel*).

Allen points to the stability of the number of these morphs over time for choosing this strategy.

Compound words exemplify the benefit of knowing the morph constituents of words (i.e. *assembly* vs. *houseboat; snowman* vs. *woman;*). Most of the words that fall in this category are composed of compounded free morphs and the biggest problem is caused by:

1. the incorporation of the silent final "e" into the compound word (e.g. *houseboat*)
2. the deletion of the final silent "e" without the concomitant change in the compounds pronunciation (as in *scarcity*).

There is a set of rules (Lee, 1968) for decomposing words into their constituent morphs. The rules recursively choose the longest first match from the right end of the word. The primary problem with Lee's rules is improper affix decomposition. Allen feels that Lee's rules should be augmented by a set of selection rules which choose the "best/correct" decomposition. Table 3.3 gives two examples of the results of applying various decomposition strategies to two words. Affixation is preferred to compounding so "scarce-ity" is chosen over "scar-city" when pronouncing "scarcity". Further, inflectional affixation is preferred to derivational affixation so "rest-ing" is chosen over "re-sting" when pronouncing resting.

One design solution is to assume that Allen's morph based approach is not the answer because it is too computationally expensive and can be replaced by some combination of a rule based or lexical based PTM. There is evidence to suggest that this idea is a viable applications environment solution. Bernstein and Pisoni (1980) examined two systems, the Telesensory Systems Inc. (TSI) text-to-speech system, and the MITalk-79 system from Massachusetts Institute of Technology (MIT). The TSI system is the production system "offspring" of MITalk-79. MITalk-79 makes extensive use of Allen's ideas of morphological analysis and does

Morph Decomposition Strategies

Word	Method of Decomposition	Decomposition
scarcity	compounding	scar-city
	affixation	scarce-ity
	compound & affixation	scar-cite-y
resting	inflectional affixation	rest-ing
	derivational affixation	re-sting

Table 3.3

not have an exceptions lexicon. Conversely, the TSI system does not use morphological analysis but does have an exceptions lexicon. A more complete specification of characteristics of the two systems may be found in Table 3.4. Bernstein and Pisoni compared the two systems and found that after simplifying modifications, the TSI system pronounced 97% of its words correctly as compared to 99% correct pronunciation with MITalk-79. Bernstein and Pisoni went on to report that this difference is not significant [35].

An alternate solution is a compromise between Allen's morph based approach and the more conventional letter to phoneme rules supplemented with an exceptions lexicon. Assume that the PTM follows the SPM, then in the process of parsing the string for syntax and lexical look-up, the morphological analysis will already have been done. In keeping with the information retention principle mentioned earlier, a record of this information could be attached to the word in the data structure. In the case of compound words such as 'snowman' and 'houseboat', this could be very effective if the letter-to-phone rules could make use of this information. This was the type of design initially considered for Talker.

[35] The reader of their paper is left to assume that they are speaking of a statistical significance, however they do not report the levels at which the results are significant.

TSI Text-to-Speech system and MITalk-79

TSI Text-to-Speech	MITalk-79
2000 word exceptions lexicon	12,000 morpheme lexicon
no parser	phrase parser
two parts of speech recognized	26 parts of speech recognized
punctuation breaks marked	phrase and clause ends marked

Table 3.4 (After Bernstein and Pisoni, 1980)

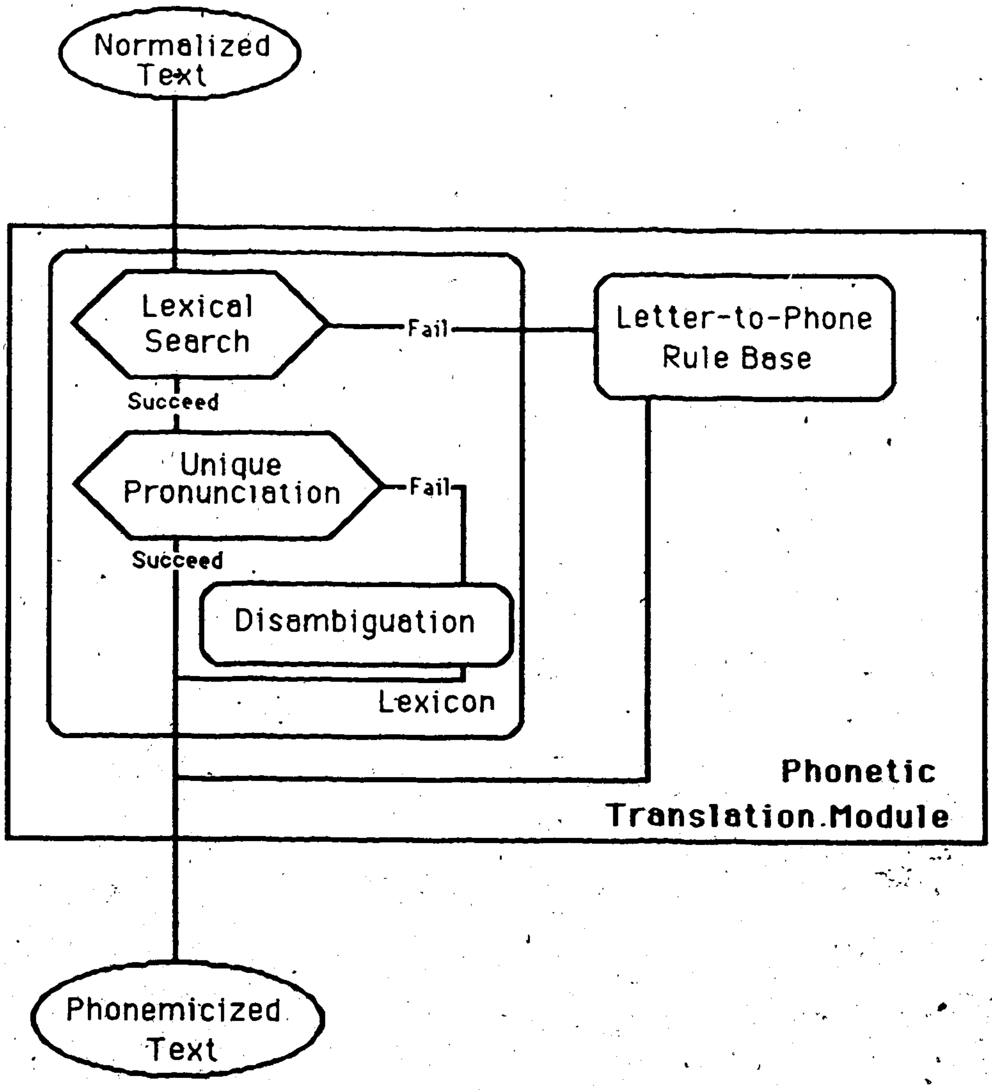
3.3.1 Design of the Phonetic Translation Module

The sequential placement of the PTM within the second block (see Figure 3.1) is not critical because the translation process may precede, follow, or even occur in parallel with the syntactic analysis process. The internal units of the PTM (see figure 3.2) will be discussed separately and then as they relate to each other and the design of Talker's PTM.

3.3.1.1 Letter-to-Phone Rules (LTPR's)

These rules are language specific and function to translate English orthography into its phonemic equivalent. There is no generally accepted set of LTPR's for the English language. The most compelling reason is the number of dialects of English and the variety of pronunciations of English orthography. LTPR's are based on the phonic rules of a dialect, thus each LTPR set constructed by a researcher is different.

A second reason for the lack of a generally accepted LTPR set is that the phones into which strings are translated are not a constant set. Many of the sets that are used are a combination of sound segments not representable in the International Phonetic Alphabet



Detail of Hypothetical Phonetic Translation Module

Figure 3.2

(IPA). This type of phone translation set has been used by researchers to implement a "one pass approach" to translating the input text (Text-to-Speech system of Sweet Micro Systems Inc.; Ciarcia, 1982; Allen, 1973). This approach combines in one rule the task of interpreting English orthography on a phonemic level with the task of choosing the proper allophone of that phoneme given the current context. The reasoning behind this approach is that if only one translation step is made, then a great deal of information must be stored in the rules and this must be representable in the translation phone set. This is a realistic approach if the text-to-speech system is being implemented under resource-limited circumstances or if the system is designed as only a first approximation to the problem's ultimate solution.

Other systems use a multiple-pass approach to their rule base, and even divide their rules up into subsets to be applied sequentially (Hertz, 1982; Olive, 1974; Carlson & Granstrom, 1974). This allows the phonemic rules and the allophonic rules to be put in separate sets. A good example of a subset of such a rules base is given by Dilts (1984) [36].

The advantage of the multipass approach is its logical hierarchical structure. The English language is very context dependent in most aspects of pronunciation, and written English is highly structured and sequential. Recognition of an orthographic symbol gives a clue as to pronunciation but it is the sequential nature of the symbol's context that refines the pronunciation of the target symbol. The multipass approach utilizes the structured nature of pronunciation knowledge as a basis for organizing its rules set.

When the knowledge contained in the rules base is logically split into subsets then each rule in each subset has less knowledge embedded in it and there are fewer rules in total. This is because the pronunciation knowledge is distributed between the logic used to access the subsets of the rules, the logical order of the application of the subsets, the order

----

[36] It should be recognized that the one pass approach and the mulitpass approach lie on a continuum of ideas of how to process information. These two representative points are discussed here as a dichotomy for simplicity.

of the rules themselves within each subset, as well as the rules themselves.

The translation rules of a one pass system are kept in one non-structured set. That is to say that the rules are unordered and unsegregated according to any criteria. This lack of structure is the primary advantage of a one pass rule set. This implies that the knowledge needed to pronounce English strings is embedded in only the rules and not their access methods or grouping strategies; thus the knowledge is in the data not the control structure. This allows a researcher to state and revise his theory of English pronunciation separately from the program which operates on that knowledge.

Disadvantages to a LTPR base are that a large number of unorganized rules are difficult to check for attributes such as redundancy, scope of applicability, and even correctness of application. These problems are more acute in a one pass approach but the multipass approach is not without disadvantages as well. If a researcher is starting from scratch, a multipass rule base is more difficult and time consuming to construct. This is because it requires an underlying logical structure before one takes a "try this and see if it works" approach to refining the rules. A problem that both approaches suffer from is that the rules are not normally algorithmically ordered within the sets. The ordering is something that is laboriously done by hand.

LTPR's translate letters in the context of the word or phrase and from their English to their phonetic equivalent. In other words, LTPR's in conjunction with this interpretation system, function as language transducers (Denning, Dennis, & Qualitz, 1978). They deterministically translate each English language input into a specific phonetic language output.

3.3.1.2 Lexicon

The data comprising a lexicon tends to vary with the application. However, in a text-to-speech system which utilizes a syntactic parsing module, the data could be expected to include the potential syntactic categories the string can assume, and the pronunciation and semantic description of the string that corresponds to each syntactic category. The data could be stored in a directed graph. The maximum height of the graph would be the length of the longest word. Each level of the graph would be an alphabetical list arranged in a most-recently-used-first manner. Each node corresponds to a letter in one of the lists. The entry of a word into the structure would consist of constructing a path from the root (which represents the first letter in the word) through each level, connecting the appropriate nodes in order to spell the word. The structure is searched by parsing the target string and simultaneously traversing the tree. If the string is there, the last letter/node will point to the information available about that word. If it is not there, then the last letter will point at some null location.

3.3.1.3 Disambiguation Process

Ambiguity of pronunciation is a major problem in a text-to-speech system. This problem has been mentioned earlier and examples of certain aspects of the problem were displayed in Table 3.2.

There are two basic types of ambiguity, that related to pronunciation and that related to syllabic stress patterns. These problems usually arise due to syntactic, tense, or semantic ambiguity. Two observations can be made regarding these problems and current text-to-speech systems.

The first observation is that many words are ambiguous if their syntactic category is not known. Some words have multiple pronunciations associated with the same spelling. Other words are basically pronounced the same but have different stress patterns when used differently syntactically (see Table 3.2). In each case, if the syntactic category of the word is known, and the word has been located in an exceptions lexicon, then the correct pronunciation or stress pattern may be retrieved.

Secondly, some words only have an ambiguous pronunciation if they are semantically unclear. For example, consider the word *bow* in the sentence: "*That is the* *bow*." . Even though it is clear that *bow* functions as a noun in that sentence, its pronunciation is ambiguous.

Due to the nature of the LTPR base, the rules cannot generate more than one pronunciation for a word. Therefore, if the word's phonemic analog has been generated through the rule base, there is no problem with ambiguity, although there may be pronunciation errors. It will be part of the tuning process to ensure that words with multiple pronunciations are stored in the exceptions lexicon.

In a text-to-speech system, the only way to deal with ambiguous pronunciations is through multiple entries in the lexicon. If every pronunciation is associated with the correct syntactic category of that word, then most of the ambiguous pronunciations can be resolved given that the syntactic categorizations are available.

With regard to the problem of semantic ambiguity, perhaps future research will shed enough light on semantics and context that this problem can be solved. Presently, only *ad-hoc* solutions or heuristics such as a use count can be proposed [37].

The design of the phonetic translation module attempts to achieve an effective compromise between the three basic approaches introduced earlier.

It was assumed that a syntactic module would precede the PTM. Therefore, the syntactic information could allow disambiguation to be done when the string was sought out in the lexicon. If the string was not in the lexicon, the letter-to-phone rules would be used to translate the string. The lexicon is searched prior to the application of the rule base because lexical search can fail while the LTPR cannot. The rule base was designed so that the rules could be applied using a one pass approach. Rules were introduced that acted in a manner analogous to morph decomposition. This allows some of the advantages of

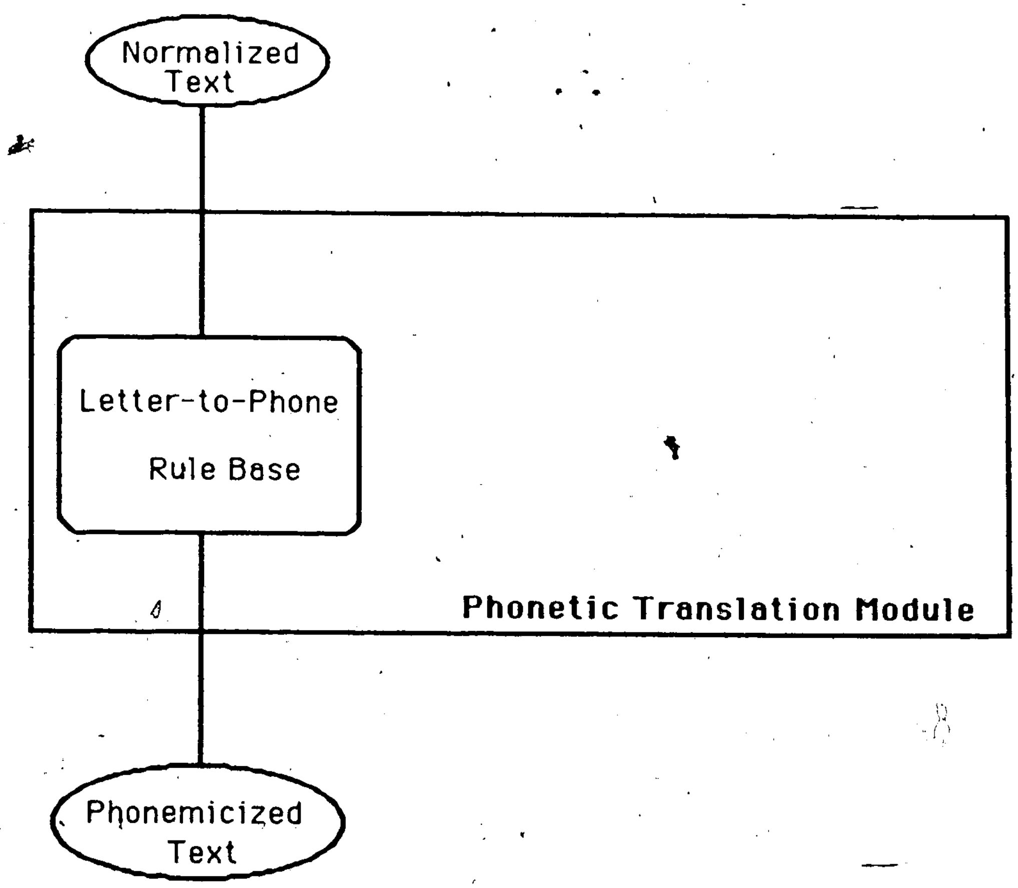
[37] At the start of each new conceptual text entity (a paragraph perhaps), the use count for each potentially semantically ambiguous word is set to zero. Then, when a word can be pronounced unambiguously, the pronunciation used has its use count incremented. In a situation where ambiguity arises, the most highly used variant pronunciation will be used. The idea is that in normal conversation/text, the speaker/writer will not attempt to confuse the listener/reader.

morphological analysis to be used with compound words. The rules were modified so that they did not have to be ordered by hand. The direct benefit of this is that the introduction of a new rule only requires consideration of the rule itself, not its placement.

3.3.2 Implementation

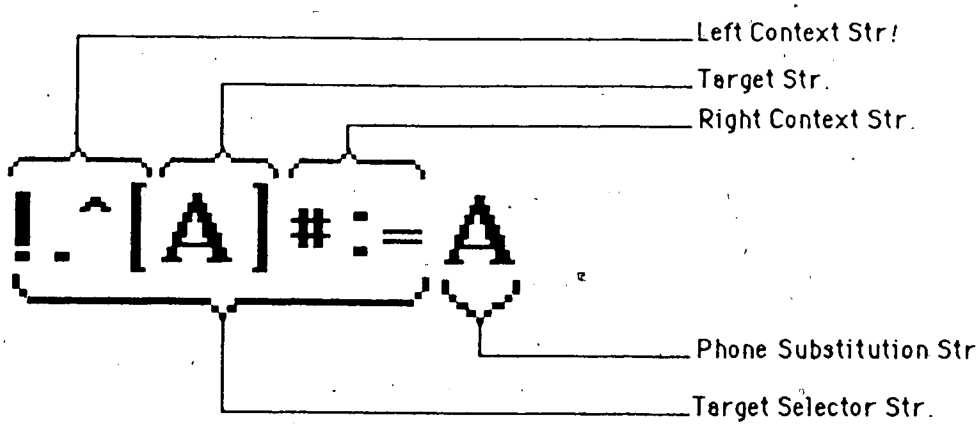
The actual implementation of the phonetic translation module is diagrammed in Figure 3.3. Because the SPM was never implemented, there would be no syntactic information available. This obviates the need for multiple pronunciation entries in a lexicon for a given word. The need for disambiguation also went by the wayside when only LTPR's were to be used. The lexicon was implemented implicitly as described earlier. The rule set of Talker is based on a combination of the Naval Research Laboratory Letter-to-Sound rules developed by Elovitz *et. al.* (1976) and the letter-to-phoneme rules used in the text-to-speech system of Sweet Micro Systems.

The LTPR's used by Talker consist of a pair of strings, namely a target selector string, and a substitution string. The target selector string is made of optional variable length context strings surrounding a target string. The context strings may be null, but the target is required to be non-null. The substitution string is an optional length (possibly null) string of either hexadecimal codes which serve to identify the phonetic code terminal symbols in the case of this one pass approach. An example rule is given in Figure 3.4. Table 3.5 lists the meaning of the symbols used in stating the rules. The entire set of the rules used by Talker is listed in Appendix B. These rules were chosen because they were originally designed to work with the SSI-263A chip which is the same speech synthesizer used by Talker. The original one pass approach was retained because the set of phones that the input is translated into directly corresponds to those available with the SSI-263. Further, the SSI-263A phone codes can represent all other potential speech synthesizer codes as a subset. The IPA and the phonetic code used by most dictionaries are subsets of the SSI-263A phone codes. The table demonstrating the equivalence relationship



Detail of Actual Phonetic Translation Module

Figure 3.3



Example Letter-to-Phone-Rule

Figure 3.4

<table>
<tr><th>Symbol</th><th>Function</th></tr>
<tr><td>!</td><td>Represents any nonalphabetic character in the input string.</td></tr>
<tr><td>*</td><td>Represents one or more vowels.</td></tr>
<tr><td>:</td><td>Represents zero or more consonants.</td></tr>
<tr><td>+</td><td>Represents a front vowel (E,I,Y)</td></tr>
<tr><td>^</td><td>Represents one consonant.</td></tr>
<tr><td>.</td><td>Represents a voiced consonant (B, D, G, J, L, M, N, R, V, W, Z)</td></tr>
<tr><td>|</td><td>Represents a digit (1,2,3,4,5,6,7,8,9,0)</td></tr>
<tr><td>[</td><td>Left context delimiter</td></tr>
<tr><td>]</td><td>Target string delimiter</td></tr>
<tr><td>=</td><td>Right context delimiter</td></tr>
<tr><td>eoln</td><td>Phone substitution string delimiter</td></tr>
</table>

Letter-to-Phone-Rule Symbol
Interpretation Table

Table 3.5

between these different phone code sets is presented in Appendix A.

The rules reside in a separate file which is input at the start of each experiment. The rules are then placed in order within their data structure and then written out to the original data file to ensure that their internal order is known. Insights gained in structuring the rule data and implementing the search strategy allowed the rules to be algorithmically ordered, the "lexicon" to be searched first, and effective analysis data to be generated.

The rules are stored in a structure quite similar to that proposed earlier for a lexicon. The root "node" is an array of 30 trees corresponding to individual letters (26), plus symbols, brackets, punctuation, and number categories (4). Any permissible input letter falls into one of these categories. Each rule is categorized according to the first character in its target string. Thus we have a subset of LTPR's for numbers, each letter ('A' - 'Z'), punctuation, etc., with each subset being one of the trees. Each node in a tree corresponds to a position of a character in a rule starting with the leftmost character of the target string. Additionally, each node contains information regarding applicable left contexts, pointers to subsequent nodes, and whether this node corresponds to the last character position in a rule.

Each LTPR subset tree is searched in breadth first fashion. This leads to multiple paths being explored. So long as each path can be extended, it is retained. If one path cannot be extended and another can, the unextendable path is dropped from the search space. If no paths may be extended, then incomplete paths are dropped from the active search space but are retained in case of backtracking. Each complete unextendable path found will have its left context string examined. If a successful match is found, then the substitution phone string is appended to the phone string being built for the current input text. This search behavior was developed out of the following observation. For all rules applying to a given target string in a given context, only the rule with the longest and most highly specified target selector string should be selected. This observation also suggested that there was an underlying complete ordering which could be imposed on each subset of LTPR's. This ordering was based on the scope of applicability of each of the symbols used in the rules. Each rule symbol first takes on a value equal to the number of characters which it may represent. This results in a partial ordering. In addition, all those symbols which represent the same number of characters are sorted into their ASCII alpha-numeric order. The rules are then sorted left to right on the target string and the right context string. Since it is possible for two rules to have the same target string and right context (thus being represented by exactly the same path in a subset LTPR tree), the left context is used to complete the ordering. Because the left context string is most reasonably considered in a right to left direction, it is sorted in the same direction.

The ultimate product of this data structure, its search strategy, and the algorithmic ordering of the rules, is that the implicit "lexicon" (i.e. the rules which supply phone strings for complete words) is effectively searched first before any of the more general LTPR's are tried.

Finally, in implementing the PTM, the information retention principle had to be considered. For debugging purposes, there is code within Talker which allows virtually a complete record of the paths which were investigated to be output. Since this volume of output is both expensive to store and expensive to generate in terms of performance forgone, it is not normally generated. What is generated is an echo of the input line with a record of the rules used to translate it, their phone substitution strings and the portion of the input string to which each rule applies. This feature simplifies the problem of tracking down any mispronounciations which occur.

Speech is much more than a series of properly pronounced words concatenated to form an utterance. Duration, amplitude, and pitch are all characteristics which have their effect on utterances as a whole.

3.4 Segmental Feature Module

Segmental features are relevant at the phone or symbol level. Each phone's characteristics must be considered in the context of its neighboring phones because of the need to model the limited postures of the human vocal apparatus and its transitional information. The limited context of consideration affects the relevance and manifestation of segmental acoustic cues such that they are comparatively localized (Hill, 1980). Witten (1982) notes that:

> "The distinction between prosodic and segmental effects is a traditional one, but it becomes rather fuzzy when examined in detail"

The problem is that prosodics cannot exist without segmental features functioning as building blocks. This leads to the recognition that the prosodic characteristics of pitch, rhythm, and timing may be productively examined on a segmental level for it is here where they have their realization.

A prosodic unit's timing is a direct result of the durations of the segments which compose it. The length of a given consonant within a given word is inversely related to its positional distance from the beginning of the word. Consonants are generally not affected by local stress constraints or weight (Church, 1985). The converse is true for vowel length (Cater, 1983; Klatt, 1975). Also, the vowel in the stressed (prominent or salient) syllable of a word tends to be longer than the same vowel would be in a non-stressed syllable. Additionally, the phonetic quality of the vowel may be altered (reduced versus non-reduced).

On a suprasegmental level, the pitch contours of a prosodic unit seem to be relatively independent of segmental influence. There is, however, the phenomenon of "microintonation". This is seen in the pitch changes on the transition in and out of certain consonants (Witten, 1982; Haggard et. al., 1970). Ladefoged (1967) explained this in terms of a variation of air pressure from the lungs on the vocal cords. This occurs because the brief increase in the supraglottal pressure associated with some consonants (voiced fricatives, and voiced stops) cause a reduction in air flow across the vocal cords. This results in a concomitant reduction in both the cord's vibrational frequency and the pitch produced.

The third prosodic characteristic of amplitude or loudness also finds expression as a segmental feature. Although loudness is a very weak prosodic characteristic, it is very important to segments of synthetic speech, particularily phonemes. Table 3.6 summarizes the relative amplitudes of the various phonemes.

Other segmental features deal more with how segments are combined and the interactions which result from such combinations. For example, consonants in word-initial or syllable initial positions may have increased aspiration. Further, salient syllables may be articulated more clearly than less prominent ones hence the target values of their formant transitions are more likely to be reached (Witten, 1982). This last class of segmental interaction features is well addressed in the linguistic literature and is therefore not addressed here. Further, this latter class of features is considered to be at a level of abstraction below that which this project is aimed.

3.4.1 Design of the Segmental Feature Module

The available time and hardware imposed stringent constraints on the segmental features that could be considered in the design. The synthesizer hardware chosen for this project does not allow for the any control over the "shape" of individual phonemes so that the interesting segmental features could not be investigated. The only real segmental feature which can be addressed using this particular hardware is the duration of individual phones. As mentioned earlier, this segmental feature is seen at the suprasegmental level as a timing characteristic of the utterance. The decision was made to design Talker with the idea of allowing for the future incorporation of a subsystem that could be used to experiment with phone durations based on their type, position in the word, and other applicable information.

Relative Power of Speech Sounds

Rank	Phoneme	Example	Relative Power (p^2)
1	AW	talk	680
2	AH1	top	600
3	UH	ton	510
4	AE	tap	490
5	O	tone	470
6	OO	took	460
7	A	tape	370
8	EH	ten	350
9	Y1-U	tool	310
10	I	tip	260
11	E	peek	220
12	R	rare	210
13	L	lilly	100
14	SCH	sugar	80
15	NG	sing	73
16	M	mama	52
17	T-SCH	chuch	42
18	N	nancy	36
19	J	judge	23
20	E-J	azure	20
21	Z	zoo	16
22	S	sister	16
23	T	tot	15
24	KV	go	15
25	K	cook	13
26	V	vote	12
27	THV	that	11
28	B	bob	7
29	D	dad	7
30	P	paper	6
31	F	fluffy	5
32	TH	thick	1

Table 3.6

3.4:2 Implementation of Segmental Feature Module

While the segmental feature module is part of the conceptual design, there was little implemented in terms of identifiable code to perform this function. The consideration of segmental features led to the current design of the data structure holding the phonemic description of the utterance. Each phoneme can be given a duration other than that specified by the translation rule used to generate the phonetic translation. This durational information remains easily accessible throughout the chain of subroutines which comprise Talker [38].

3.5 Prosodic Feature Module

Prosodic or suprasegmental features characterize the utterance as a whole. Here, "utterance" refers to a unit of speech which may encompass several words, a phrase, a clause, or a sentence. These divisions form the bounds of the natural prosodic units (Witten, 1982).

Dilts (1984) describes the his personal view of prosodics in English by saying,

> "in this particular language, prosodic features are extrinsic to information content and serve primarily to allow speakers to express emotion or indicate the relative importance of individual words."

This may or may not be so, but for synthetic speech to improve, it must become possible to both embed and recognize prosodic information in an utterance. To do this, three key areas must be investigated. Firstly, with respect to the text-to-speech translation process, what are the syntactic and/or semantic cues which indicate prosodic information? Secondly, what exactly are the aural signposts which the listener interprets as being prosodically meaningful? And thirdly, what is the mapping from the textual cues to the aural expressions of prosodic information? A great deal of research has been done in all of these areas and a review of that literature will not be attempted here. Some of the more significant research and definitive points of view will however be mentioned [39].

----

[38] The implementation of Talker included a dummy subroutine call which was designed to allow the easy incorporation later of a module built separately to perform the segmental feature application process.

[39] Witten (1982) details further reading sources as including Abercrombie, D. (1965); Bolinger, D. (1972); Crystal, D. (1969); Bimson, A.C. (1966); Lehiste, I. (1970); Pike, K.L. (1945). Also suggested are Witten, I. (1982) and Bristow (1984).

Basic to many discussions of prosodic features is the idea of stress. Witten (1982) points out that

"Stress is an everday notion, and when listening to natural speech people can usually agree on which syllables are stressed. But it is difficult to characterize in acoustic terms. From the speaker's point of view, a stressed syllable is produced by pushing more air out of the lungs. For a listener, the points of stress are *obvious*. ...However, it is a rather subtle feature and does not correspond simply to duration increases or pitch rises. It seems that listeners unconsciously put together all the clues that are present in an utterance in order to deduce which syllables are stressed. It may be that speech is perceived by a listener with reference to how he would have produced it himself, and that this is how he detects which syllables were given greater vocal effort."

There are two widely separated points of view as whether or not the reader (text-to-speech system) may be able to extract prosodic features from textual material. There is the school of thought which feels this can be done (Chomsky and Halle, 1968; Bresnan, 1971,1972; Klatt, 1975; Culicover and Rochemont, 1981; Gillot, 1985; Church, 1985). Much of this analysis depends on the nuclear stress rule as described by Chomsky and the idea of surface and deep structure. There are other people, notably D. Bolinger, who feel that "Accent is predictable (if you're a mind reader)" (Bolinger, 1972). Others besides Bolinger, who subscribe to this sort of view are Berman and Szamosi (1972), and Lakoff (1972). This author is greatly influenced by Church (1985) and Gillot (1985) and is of the opinion that a great deal of prosodic information may be extracted from the text of an utterance. Unfortunately, this presupposes a syntactic and/or semantic analysis of the utterance. It is felt that this analysis is best handled in the GPSG framework discussed earlier than by Chomsky's methods.

A source of prosodic information other than straight text is the concept upon which that text is based. This is referred to as synthetic speech from concept (Young and Fallside, 1979). While this topic is tangential to text-to-speech per se, it is mentioned because it has great potential for very effective expression of prosodic information in synthetic speech.

With regard to the question of what a listener interprets as prosodically meaningful, a great deal of research has been done. It is known that stress (prominence, accent, or salience) is acoustically manifested in terms of duration, pitch and amplitude. There are studies involving the pitch contours of utterances (Witten, 1979; Hill and Reid, 1977; Pierrehumbert, 1981) and the rhythm or timing of utterances (Jassem et. al., 1984; Klatt, 1975). There are essentially no prosodic studies directed at controlling amplitude because of its negligible level of importance.

Assuming that prosodic information is available and that we know the acoustic parameters that must be controlled, the question arises as to how to control the parameters to get across the intended message. Intuitively, the non-linguistic researcher often describes a stressed syllable as being louder than others proximate to it. Lehiste and Peterson (1959) have shown that this is not necessarily or even usually the case. The rate of change of pitch tends to be greater across a stressed syllable (Witten, 1982). Stressed syllables often have a longer vowel sound than both the same syllable in an unstressed situation and other nearby syllables (Witten, 1982). This is not universally true, however, as Morton and Jassem (1965) have conducted experiments which used bisyllabic nonsense words which led them to conclude that some people consistently judge the shorter syllable to be stressed in the absence of other clues.

Unfortunately, the absolute frequency, the direction of pitch change, and the shape of the associated pitch contour are also involved in whether or not a syllable is perceived as being stressed. Witten (1982) notes that prosodic stress,

> "...is confused by the fact that certain syllables in words are often said in ordinary language to be *stressed* on account of their pos tion in the word"

irrespective of the role of the word in the utterance.

3.5.1 Design of the Prosodic Feature Module

The prosodic (suprasegmental) feature module (PFM) of Talker was designed to allow the manipulation of the characteristics of an utterance's pitch only. This was done because pitch is generally accepted as the most prominent acoustic realization of prosodic information. Rhythm was ignored in an effort to simplify the problem and also because segmental duration (and indirectly rhythm) was considered in an earlier module. Amplitude was disregarded because of its neglible import to prosodics in general.

The PFM was conceived of as the tool that would select preset pitch contours, modify them in the context of the utterance, then apply them. The genesis of this idea lies in the work of Witten (1979) with regard to the transference of original pitch contours to synthetic speech. Witten's work would provide the basis for the algorithm used to map the prespecified pitch contour to the utterance. The original part of this whole idea was that this process should be iterative and constructive. That is, first a contour would be selected for a word and applied. Then a contour would be selected for the next highest syntactic category that included the word (e.g. noun phrase), and that would be applied with reference to the contour of the word selected earlier. This process would continue until the highest syntactic classification (a sentence) was considered. The result would be a unique contour for the sentence based on its constituents.

The strengths of this constructive synthesis of pitch contour hypothesis are that one can include Halliday's (1970) ideas about typical pitch contours of sentence types but not limit oneself to only those types. Also, this idea allows the inclusion of a pitch contour for one word, possibly specifiable in a dictionary [40]. The disadvantages of this hypothesis are that it requires a very complete syntactic analysis of the utterance as data and that the idea itself is not grounded on experimental evidence found in empirical studies of natural speech.

----

[40] This could assist in clarifying the differences between different pronunciations of words based on their syntactic categorization.

3.5.2 Implementation of the Prosodic Feature Module

The implementation of the PFM as designed was not possible due to the lack of the syntactic analyzer component of the system. The data structures used and the principle of information retention followed through this project would, however, allow its eventual inclusion. What was implemented in its place was a method of applying a static pitch contour to each word.

Each word has its first phone start on a base pitch which is user selectable. The pitch of each phone is then altered in a stepwise manner which approximates a sine curve for the duration of the word. The size of the pitch change from phoneme to phoneme is constant, therefore a short word tends to have a rising pitch contour and a longer word has a rise/fall pitch contour. The result is that the pitch contour of the utterance as a whole, (at least from the point of view of the listener) is unpredictable.

The rate of change of pitch is also controllable, in an indirect manner. By altering the duration of a particular phone, the relative rate of change of pitch across that phone, automatically changes. Unfortunately, both the actual proximity of the realized pitch of a phone to its specified target pitch and the smoothing of pitch transitions are handled by the synthesizer hardware and are therefore out of control of the researcher.

The third block of Figure 3.1 contains two modules, the parameter generator and the speech synthesizer itself. These two modules perform a function analogous to the vocal apparatus of a person. That is to say that up to this point, the earlier modules were concerned with the generation and detailed description of the utterance in abstract terms. This third block forms the effector apparatus of Talker.

3.6 Parameter Generator Module

A parameter generator takes the fully described phonetic string as input and produces the actual parameters needed for the speech synthesizer to function. The PGM functions in a manner analogous to a device driver and is therefore synthesizer specific. This is why both the PGM and the speech synthesizer are located together in one block of Figure 3.1.

3.6.1 Design of the Parameter Generator Module

The PGM is a very straightforward module to design as it is entirely based on translation tables. This module could be designed to consume the phonetic string description synchronously with its generation (on a phoneme by phoneme basis). Alternatively, the module could run asynchronously with respect to the generation of the phonetic string description. This has the advantage of allowing the PGM to be responsive to real time interrupts. This type of asynchronous design is particularily suited to applications types of text-to-speech systems where it allows the system to respond to time delays and higher priority interrupts without requiring a complete input string.

3.6.2 Implementation of the Parameter Generator Module

The PGM was implemented in a synchronous fashion. This choice was made because of the research nature of Talker. It was not considered crucial for Talker to respond to real time delays. Asynchronous design leads to more cleanly interfaced modules and simpler sharing of data between modules.

A completely translated phonetic string is generated before the PGM is called. This has the effect of slowing down the synthesis process relative to asynchronous implementation. The phonetic string is treated as a first-in, first-out (FIFO) queue as it is dismantled and the parameter string is built up.

3.7 Synthesizer

The synthesizer that was chosen to be used in the implementation of Talker was the Silicon Systems Incorporated 263A chip (SSI-263A) designed by a group headed by D.G. Maeding[41].

The SSI-263A is a 24 pin VLSI chip implemented as a single monolithic C-MOS integrated circuit (SSI-263A Data Sheet, 1984). The chip's design is based on the ideas discussed earlier in this paper. This design allows it to turn the 50 to 500 bits/second data rate it receives (instructions) into a 10 Kilobit/second data rate in its vocal tract section (Electronics, 1984).

The human vocal tract is emulated through the use of a set of switched capacitor filters. Separate glottal and fricative sources drive the multiple filter elements in the simulated vocal tract. Fricatives are generated through the addition of pseudo-random noise. The combined signal is then filtered to create the appropriate spectral shape. There is a separate section on the chip to control speech dynamics (see Fig 3.5).

The SSI-263A is clocked at 1 MHz in this application. Internally, the chip contains five eight-bit registers which provide the information needed to produce one phone. These registers allow 256 phones to be specified, four modes of handshaking, 4096 levels of pitch or 32 levels with eight different speeds of inflection movement, 16 overall rate or speed settings, 16 levels of amplitude, 8 rates of articulation and 255 level settings of the vocal tract filter frequency response. This last feature allows complete sound effects capabilities (Design Specification SSI-263A, 1984).

---

[41] This chip has also been incorporated in the commercially available Sweetalker II built by Steve Ciarcia (Ciarcia, 1984).

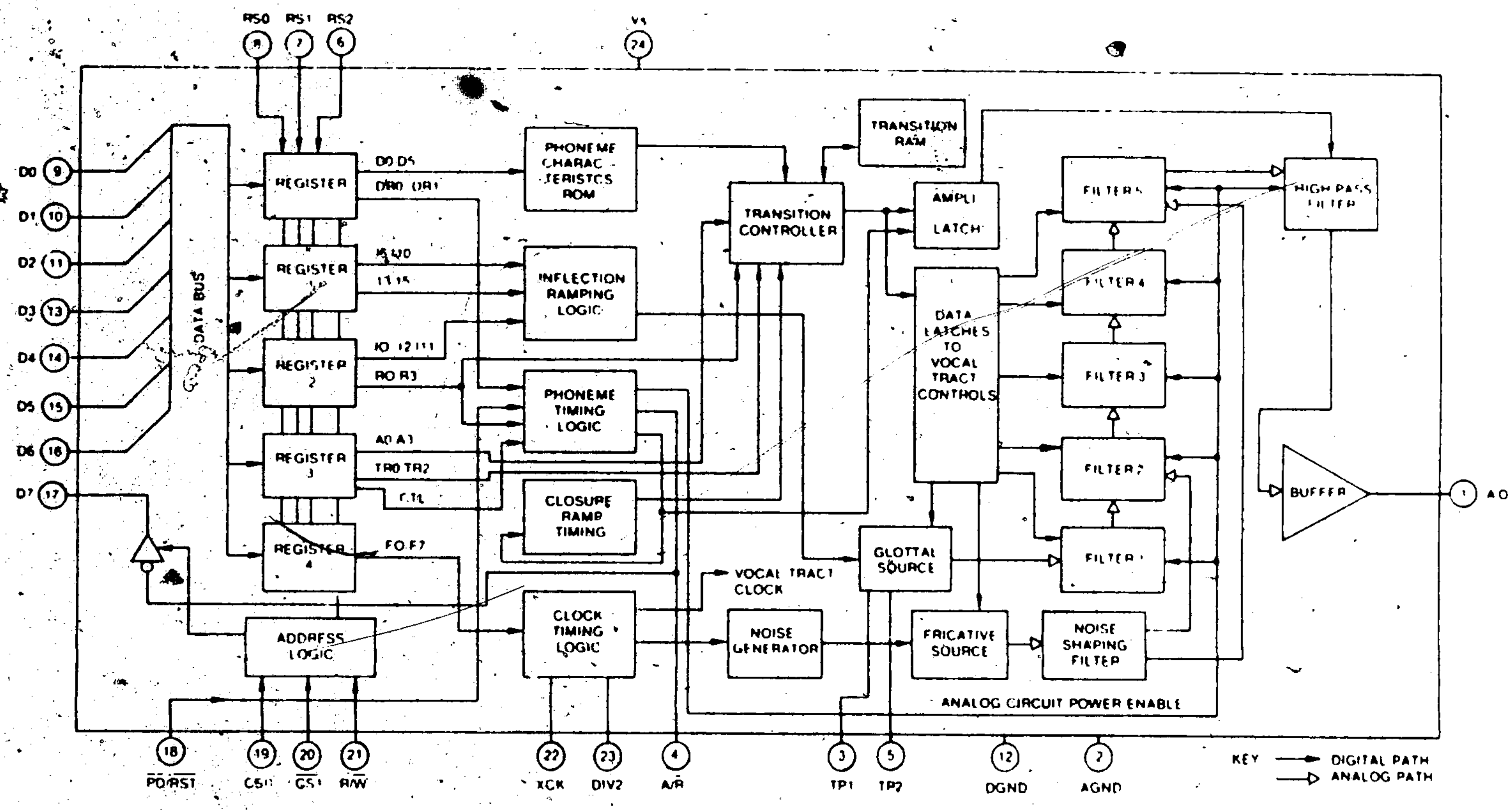


Figure 3.5

Block Diagram of SSI-263A Logic
(After Electronics, 1984)

4. Results

The results of this research project are presented in this thesis in the form of an audio cassette tape included at the end of the thesis.

Side A of the tape is a recording of synthetic speech as generated by Talker. The contents of the tape are as follows:

1. Table 3.1
2. Table 3.2
3. Table 4.1
4. Table 4.3
5. A sample of connected speech: "The subject of this thesis is the design and implementation of a system capable of delivering synthetic speech."
6. The phrase "Please get off my cord, thank you".

Tables 3.1, 3.2, 4.1, and 4.3 are all recorded with the inflection, speech rate, and filter frequency of the SSI-263 chip set at 8, 8, and 232 respectively. The sample of connected speech is repeatedly recorded at various inflection, speech rate, and filter frequency settings. These settings are, in order of occurrence: [42]

Inflection set at 1, 10, 20;

Speech rate set at 0.5, 1.0, 1.5, 3.0 and 5.0 times "normal" speed;

Filter frequency set at 220, 230, and 240.

Side B contains the same material as A, however the recording is of the output generated by the Text to speech system manufactured by Sweet Micro Systems Inc. This sytem is the one from which the LTPR base of Talker is derived. The side B recording is included for comparison purposes only.

---

[42] These settings do not correspond directly to any accepted units of measurement. The "inflection", "Speech rate", and "Filter frequency" are merely aspects of the SSI-263A which may be reset to produce different qualities of output.

Additionally, the LTPR base used by Talker is included in Appendix B and a sample of the debugging output generated during the recording of the tape is included in Appendix C.

4.0.1 Evaluation

Text-to-speech systems cannot be improved without evaluations being made. Both the test data and the type of evaluation performed on the test results are important areas. Consider a text-to-speech system which mispronounces only one word. If it is the least frequent word in the English language, it may never be detected. If it is the most frequent word, the error is intolerable. To detect these types of problems in a systematic manner, Cater (1983) proposes to test the systems using a list of the most frequently used words (see Table 4.1) [43]. Testing a system in this manner can reveal inadequacies in the stored lexicon or generalized rules. Using this same principle, testing could be done on the phonemes produced by the system in order of frequency of usage in running text (see Table 4.2). One could use a type of articulation drill (Table 4.3) to pinpoint a deficiency in a specific phoneme. The examination of individual sound segments should clearly be performed prior to testing the system on words or phrases as a sound segment test can point out inherent and potentially uncorrectable limitations in the speech synthesizer that is being used. Most importantly, test data should be recognized by experts in the field and have been used in a prior system evaluation.

Qualitative evaluations are useful when the subject lends itself to comparison with a generally known standard. In the case of synthetic speech, the most common comparison is with human speech. The majority of reviews and evaluative articles published recently, analyze synthetic speech research topics in a qualitative manner (Klatt, 1980; Witten, 1982; Miastkowski, 1982; Cater, 1983; Bristow, 1984; Smith, 1984; Kaplan and Lerner, 1985). Qualitative evaluation was the method that was chosen for this project. Talker is evaluated with

---

[43] Exactly how many words should be tested and the confidence level predictable from tests of this type, are not the subject of this paper. That area is best left to the field of statistics which specializes in such types of measurement.

meaning of the word (*-dom, -ship, -ness, -al*) and inflectional suffixes, which affect the grammatical role of the word (*-s, -ed, -ing*). Additionally, there are two kinds of root morphs:

1. free morphs which can stand alone, (*snow, boat, house*)
2. bound morphs which must combine with an adjacent morph (*-turb, -ceive, crimin-, -pet*).

Allen points to the stability of the number of these morphs over time for choosing this strategy.

Compound words exemplify the benefit of knowing the morph constituents of words (i.e. *assembly* vs. *houseboat; snowman* vs. *woman;*). Most of the words that fall in this category are composed of compounded free morphs and the biggest problem is caused by:

1. the incorporation of the silent final "e" into the compound word (e.g. *houseboat*).
2. the deletion of the final silent "e" without the concomitant change in the compounds pronunciation (as in *scarcity*).

There is a set of rules (Lee, 1968) for decomposing words into their constituent morphs. The rules recursively choose the longest first match from the right end of the word. The primary problem with Lee's rules is improper affix decomposition. Allen feels that Lee's rules should be augmented by a set of selection rules which choose the "best/correct" decomposition. Table 3.3 gives two examples of the results of applying various decomposition strategies to two words. Affixation is preferred to compounding so "scarce-ity" is chosen over "scar-city" when pronouncing "scarcity". Further, inflectional affixation is preferred to derivational affixation so "rest-ing" is chosen over "re-sting" when pronouncing resting.

One design solution is to assume that Allen's morph based approach is not the answer because it is too computationally expensive and can be replaced by some combination of a rule based or lexical based PTM. There is evidence to suggest that this idea is a viable applications environment solution. Bernstein and Pisoni (1980) examined two systems, the Telesensory Systems Inc. (TSI) text-to-speech system, and the MITalk-79 system from Massachusetts Institute of Technology (MIT). The TSI system is the production system "offspring" of MITalk-79. MITalk-79 makes extensive use of Allen's ideas of morphological analysis and does

100 most frequently used words as compiled by Godfrey Duey

Rank	Word	Frequency	Rank	Word	Frequency
1	the	7.31	51	when	0.23
2	of	3.99	52	him	0.23
3	and	3.28	53	them	0.22
4	to	2.92	54	her	0.22
5	a	2.12	55	am	0.21
6	in	2.11	56	your	0.21
7	that	1.34	57	any	0.21
8	it	1.21	58	more	0.21
9	is	1.21	59	now	0.21
10	I	1.15	60	its	0.20
11	for	1.03	61	time	0.20
12	be	0.84	62	up	0.20
13	was	0.83	63	do	0.20
14	as	0.78	64	out	0.20
15	you	0.77	65	can	0.19
16	with	0.72	66	than	0.19
17	he	0.68	67	only	0.18
18	on	0.64	68	she	0.18
19	have	0.61	69	made	0.17
20	by	0.60	70	other	0.16
21	not	0.58	71	into	0.16
22	at	0.58	72	men	0.16
23	this	0.57	73	must	0.16
24	are	0.54	74	people	0.16
25	we	0.52	75	said	0.16
26	his	0.51	76	may	0.16
27	but	0.50	77	man	0.15
28	they	0.47	78	about	0.15
29	all	0.46	79	over	0.15
30	or	0.45	80	some	0.15
31	which	0.45	81	these	0.15
32	will	0.44	82	two	0.14
33	from	0.43	83	very	0.14
34	had	0.41	84	before	0.13
35	has	0.39	85	great	0.13
36	one	0.36	86	could	0.13
37	~~our~~	0.33	87	such	0.13
38	an	0.33	88	first	0.13
39	been	0.32	89	upon	0.12
40	no	0.32	90	every	0.12
41	their	0.31	91	how	0.12
42	there	0.30	92	come	0.12
43	were	0.30	93	us	0.12
44	so	0.30	94	shall	0.12
45	my	0.29	95	should	0.11
46	if	0.26	96	then	0.11
47	me	0.25	97	like	0.11
48	what	0.25	98	will	0.11
49	would	0.25	99	little	0.11
50	who	0.24	100	say	0.11

Table 4.1 (After Cater, 1983)

Frequency of Speech Sound Segments.

Rank	Phoneme	Frequency	Rank	Phoneme	Frequency
1	I	7.94	21	F	1.84
2	N	7.24	22	HF	1.81
3	T	7.13	23	B	1.81
4	R	6.88	24	O	1.63
5	UH	5.02	25	U	1.60
6	S	4.55	26	AH2-E	1.59
7	D	4.31	27	AW	1.26
8	AE	4.17	28	NG	0.96
9	E	3.89	29	SCH	0.82
10	L	3.74	30	KV	0.74
11	EH	3.44	31	OO	0.69
12	THV	3.43	32	YI	0.60
13	AH	3.33	33	OU	0.59 (Dipthong)
14	Z	2.97	34	T-SCH	0.52
15	M	2.78	35	J	0.44
16	K	2.71	36	TH	0.37
17	A	2.35	37	E-U	0.31 (Dipthong)
18	V	2.28	38	O-E	0.09 (Dipthong)
19	W	2.08	39	E-J	0.05
20	P	2.04			

Table 4.2 (After Cater, 1983)

Articulation Drill with specified Phoneme.

Test Words	Specified Phoneme
saw, horse, horn, ball, talk	AW
yard, clock, top, block, star, arm	AH
gloves, rug, truck, tub, button, ton	UH
tap, hat, can, black, grass, basket	AE
tone, boat, coat, snow, stove, comb	O
book, cook, foot, look, took	OO
tape, cake, grapes, table, lady, tail	A
ten, bed, dress, red, steps, feather, sled	EH
tool, blue, moon, tooth, shoe	U
tip, chicken, fish, pillow, pig	I
peek, cheese, meet, sleep, trees, green, feet	E
radio, rake, barrel, car, tire, rabbit, red	R
ladder, lease, leg, letter, ball, bottle, look	L
sheep, shelf, dish, fish, brush, push, shoulder, shake	SCH
finger, sing, swinging, ring, tongue, blanket	NG
move, music, memory, most, more, meek, mimic, movie	M
chair, cheese, chicken, watch, catch, matches, teacher, speech	T-SCH
nasal, know, knife, candle, woman, nancy, spoon, man	N
juice, engine, orange, soldier, bridge, joke, jump	J
glacier, azure, measure, television	E-J
music, zoo, roses, ears, nose, zebra, scissors	Z
seven, see, saw, sleep, spoon, basket, glasses, face	S
table, tire, butter, tot, letter, white	T
gloves, grass, gun, golf, digging, wagon, rug, flag	KV
crack, pocket, black, clock, cook, fake	K
vase, violet, vivacious, cover, drive, river, stove	V
thimble, three, thin, thick, mouth, teeth	TH
bed, boat, rabbit, ribbon, umbrella, table, bob	B
dog, drink, indian, radio, dud, bed, wood	D
paper, pencil, airplane, apple, pop, cap, rope, sleep	P
feather, finger, fire, fluffy, elephant, laugh, roof, knife	F
these, those, brother, then, father, feather, loathe	THV

Table 4.3 (After Cater, 1983)

respect to the general intelligibility of the speech as opposed to the naturalness of the speech.

Quantitative evaluation is appropriate for complete or stable systems. A quantitative evaluation has been done comparing the outputs of MITalk and the Telesensory text-to-speech system (precursor of the Prose 2000). Confidence intervals and error rates were used to report statistical tests conducted on results generated from recognized data (Bernstein and Pisoni, 1980). Kaplan (1985) reports that H.C. Nusbaum and D.B. Pisoni presented results to the *Fourth Voice Data Entry System Applications Conference* in Arlington Va. which quantitatively evaluated Digital Equipment Corporation's DECTalk along with MITalk-79, the Prose 2000, and the Votrax Corporation's Type-N-Talk.

Quantitative evaluations report results in a systematic manner against which other system's results may be validated or compared without actually having to have all the systems under comparison present. Pisoni's work has generally used the Modified Rhyme Test, a one hundred sentence subset of the Harvard Phonemically Balanced Sentences, one hundred anomolous sentences (like sequences used at Haskins Laboratories), and passages of connected text from Pisoni's own Hybrid Reading Comprehension Test (Bernstein and Pisoni, 1980).

A primarily qualitative evaluation of Talker was chosen for two reasons. Primarily, the current implementation has not been optimized and Talker can be considered an incomplete system. The second reason is the time consideration. It is more beneficial to spend time improving the current implementation than it is to spend time analyzing quantitatively the current output. This is because there are many obvious system improvements to be [illegible]ore any rigorous testing is carried out.

The quantitative analysis that was performed on Talker consisted of two parts. The first part of the analysis considered the 100 most commonly used words in the English language, henceforth referred to as the "100 MCW" (Table 4.1).

Twenty native speakers of Canadian English were used as subjects to evaluate the pronunciation of the "100 MCW". The criteria of evaluation was the understandability of the spoken word. The subjects were required to make a binary decision. A recording of the words was played to the subjects while they viewed the list of the "100 MCW".

The results indicated that in a sample of text composed only of these "100 MCW", the error rate could be expected to be 1.30%. This result is of limited interest however, as the 100 most common words (based on their frequency reported by Cater) only constitute 53.89% of a typical sample of English text.

The second part of the quantitative analysis involved selecting a sample of English text and calculating the error rate of pronunciation of words. The text sample contained 105 words in total. Of that 105 words, there were 73 different words and the "100 MCW" constituted 39.99% of the sample. The text of the sample is given below with each error underlined and footnoted where each footnote serves to relate the actual pronunciation of the word as generated by Talker. The phonetic description is in keeping with SSI standard as described in Appendix A. It should be noted that the operation of Talker and the LTPR was in no way optimized towards this sample of text on the following page.

"The Computer Revolution [44] . In a little more than three decades [45] , computer technology [46] has come a very long [47] way. The first commercial [48] computer was large [49] enough to fill a gymnasium and was considered [50] too expensive for all but the largest [51] companies [52] . Today [53] , millions [54] of people own "personal" [55] computers [56] and use them for all kinds of domestic and business applications [57] . Personal [58] computers [59] are thousands [60] of times [61] faster and more powerful [62] than the first commercial [63] computers [64] were, but they are no larger or much more expensive than a typewriter [65] . If the automobile [66] industry [67] had experienced similar progress, a new car would cost less than a gallon [68] of gas." (Long, 1984).

---

[44] ĕ
[45] ŭ
[46] aspirated 'zh'
[47] soft 'j'
[48] 'air'
[49] 'air'
[50] ĕ
[51] 'air'
[52] ă
[53] ŏ
[54] ĭ, i
[55] 'air', ŏ
[56] 'air'
[57] 'pull'
[58] 'air', ŏ
[59] 'air'
[60] 'd' is too prominent
[61] i
[62] 'air'
[63] 'air'
[64] 'air'
[65] i
[66] ŏ, ĭ
[67] ĭ
[68] ă

The error rate on the pronunciation of the preceding sample of text was 26.67% when all of the words were considered. When repetitive words were removed, the error rate was 30.14%. If only the first instance of a series of repetitions of the same mistake was considered, the error rate was 26.03%.

At first glance, these error rates might be considered high. What must be examined is the criteria which was used to judge whether the pronunciation of a word was in error. Improper pronunciation rather than understandability was the guiding criteria. As a result, all words which were flagged as errors were improperly pronounced based on the pronunciation guide found in *Webster's New Collegiate Dictionary* (1977). In no case, did Talker mangle the pronunciation of a word to the point where the word was not understandable.

An extensive quantitative analysis of Talker seems to be a good candidate for future consideration. It would provide a firm measure of the quality of the output of the system upon which to compare future changes or other systems. Extensive quantitative analyses can also serve to direct improvements in the system when the path to greater performance was not clear.

5. Conclusions

This research project may be extended in a variety of directions. Extensions to a system of this nature would generally fall into four areas:

1. Ease of Use.
2. Portability.
3. Speed.
4. Correctness of output.

5.0.1 Ease of Use

Modifications in this area generally concern the man-machine interface and are most appropriate if the system is to have many naive users. This is not the case with Talker. Were it to become important in the future, the ergonomics of Talker would have to be investigated as they were designed and implemented with only the sophisticated researcher in mind.

5.0.2 Portability

Talker is written almost entirely in ANSI Pascal which can be brought up on other systems quite easily.[69] The portion of Talker not written in Pascal is that associated directly with controlling the speech synthesizer hardware. One area of improvement in portability would be to completely dissociate the third block of Figure 3.1 (the PGM and the Synthesizer) from the heart of Talker. This would remove the only portion of the internal code written in another language. A second possible improvement to Talker would be to increase the abstraction of the phonetic string description. Greater abstraction implies that the output of Talker becomes less suited to any one particular device and the PGM must perform a more complex translation function. While this would increase the portability of the system, there would be a concomitant

---

[69] Any extensions of Pascal that are used, are those which appear in the Unix environment. Any usage of a non-standard extension is well documented.

drop in speed.

5.0.3 Speed

Generally, any improvements in the speed or response time of the system would come through efforts to pipeline the processing of the phonetic description of the input stream. The problem associated with this is that the modular interface becomes much more complex and the simplicity of design is sacrificed. The possibility exists that fine tuning the translation rule base might speed up the system by eliminating part of the the potential space associated with phonetic translation.

5.0.4 Correctness

The implementation of the syntactic parser as a separate module merits discussion. Earlier, with regard to text normalization, the point was made that if possible, redundancy in string parsing should be eliminated. The SPM is one instance where this is a tradeoff. By leaving the SPM separate, redundancy in encouraged. This should be seen as being currently beneficial for the system. This is because both the SPM and Talker do not deal with well-understood problems. As such, the solutions they propose to the problems should not be regarded as error free. Simply put, the separateness of the two systems, Talker and the SPM, enables large logic changes to be made more quickly and cleanly. In the view of the author, mating the logic aspect of the two systems should be reserved for time when, if ever, Talker is to be used in a production environment. There are other aspects of the two systems which would only be helped by an early combination of efforts. These aspects include the question of text normalization, the SPM's separate lexicon as well as the concomitant redundant parsing involved. The parsing issue is self-explanatory in that it does not make sense to look up the same text string twice, once for its syntactic category, and once for its phonemic translation.

The SPM seems like the logical place to put any text normalization which cannot be accomplished on a character by character basis. The justification is that the parser accesses the string in a bottom up fashion and must deal with the words (abbreviations, numerals, etc.) as tokens. For only a little extra effort, the normalization could be completed at this level.

The intention of reducing redundancy while maintaining logical modularity suggests that there should be only one lexicon to be shared by all processes. Currently, the SPM and Talker use two different lexicons. The reason is that each contains quite different information in quite different forms. Ideally, these two lexicons should be coalesced and a more optimal form found for storing the combined information. This new lexicon should store the information in a reasonably malleable form because it is not clear exactly what type of information is ultimately to be stored. It seems as though this problem could be productively viewed as a data base management problem and treated accordingly.

6. Bibliography and References

6.0.1 Bibliography

Ciarcia, S. 1978. *Add a voice to your computer for $35 ; Talk to me*. Byte Vol.6:142-151.

Data Sheet. 1984. *SSI 263A Phoneme Speech Synthesizer*. Silicon Systems Inc. 14351 Myford Road, Tustin CA. 92680

Design Specification - SSI 263. 1984. *Specification Number 21.19C 138W*. Silicon Systems Inc. 14351 Myford Road, Tustin CA. 92680

Duker, S. 1974. *Time-compressed speech*. Volumes I, II, & III The Scarecrow Press, Inc. Metuchen, N.J.

Flanagan, J.L.; Rabiner, L.R. 1973. *Speech Synthesis*. Dowden, Hutchison, & Ross, Inc. Stroudsburg, Pennsylvania. pp.511

Haggard, M.P.; Ambler, S.; Callow, M. 1970. *Pitch as a voicing cue*. J. of Acoust. Soc. Amer. 47:613-617.

McPeters, D.L.; Tharp, A.L. 1984. *The influence of rule-generated stress on computer-synthesized speech*. Int. J. Man-Machine Studies 20:215-226.

Morgan, N. 1984. *Talking chips*. McGraw-Hill Book Co. N.Y., N.Y. pp.178

Teja, E.R.; Gonnela, G. 1983. *Voice technology*. Reston Publ. Co. pp. 212

Umeda, N. 1976. *Phonological Rules for a text-to-speech system*. American Jour. of Computational Linguistics (microfiche)57.

Users Guide. 1984. *Phonetic programming using the SSI 263A*. Silicon Systems Inc. 14351 Myford Road, Tustin CA. 92680

Young, S.J.; Fallside, F. 1980. *Synthesis by rule of prosodic features in word concatenation synthesis*. Int. J. Man-Machine Studies 12:241-258.

6.0.2 References

Abercrombie, D. 1965. *Studies in phonetics and Linguistics*. Oxford Univ. Press, London. England.

Ainsworth, W.A. 1973. *A system for converting english text in speech*. IEEE. Transactions on Audio & Elecroacoustics Vol.21(3):288-290.

Allen, J. 1973. *Reading machines for the blind: The technical problems and methods adopted for their solution*. IEEE, Transactions on Audio & Elecroacoustics 21(3):259-264.

Allen, J. 1973. *Speech synthesis from unrestricted text*. in *Speech synthesis*. J.L. Flanagan, L.R. Rabiner (eds.) Stroudsbourg, Pa., Dowden, Hutchison & Ross.

Allen, J. 1976. *Synthesis of speech from unrestricted text*. Proceedings of the IEEE. 64(4):433-442.

Allen, J.; Hunnicutt, S.; Carlson, R.; Granstrom, B. 1979. *MI talk-79: The 1979 MIT Text-to-Speech System*.in *Speech Communication Papers presented at the 97th meeting of the Acous. Soc Am.*, J.J. Wolf and D.H. Klatt (eds.), The Acoustical Society of America, NY. NY. pp. 507-510.

Allen, J. 1981. *Linguistic-based algorithms offer practical text-to-speech systems*. Speech Technology Vol.1(1)12-16.

Berman, A.; Szamosi, M. 1972. *Observations on sentential stress*. Language 48(2):304-325.

Bernstein, J.; Pisoni, D.B. 1980. *Unlimited Text-to-Speech System: Description and Evaluation of a Microprocessor Based Device*. Proceedings of IEEE. ICASSP. Denver, pg. 576.

Bolinger, D.L. 1958. *A theory of Pitch Accent in English*. Word.

Bolinger, D.L. 1972. *Accent is predictable (if you are a mind-reader)*. Language 48(3):633-644.

Bolinger, D.L. (ed.) 1972. *Intonation*. Penguin, Middlesex, England.

Bresnan, J.W. 1971. *Sentence stress and syntactic transformations*. Language 47(2):257-281.

Bresnan, J.W. 1972. *Stress and syntax: a reply*. Language 48(2):326-342.

Bristow, G. 1984. *Electronic Speech Synthesis.Techniques, Technology, and Applications.*: Granada Publ. Ltd. pp. 346.

Caldwell, J.L. 1979. *Flexible, High-Performance Speech Synthesizer Using Custom NMOS Circuitry*. J. Acoust. Soc. Am. Suppl 64:S72(A)

Caldwell, J.L. 1980. *Programmable Synthesis using a new Speech Microprocessor..* IEEE.(4):868-871.

Carlson, R.; Granstrom, B. 1974. *A phonetically oriented programming language for rule description of speech*. Proc. of the Speech Communication Seminar, Stockholm, Sweden.

Cater, J.P. 1983. *Electronically Speaking: Computer Speech Generation*. Howard W. Sams & Co. Inc., Indianapolis, Indiana, USA.

Chomsky, N.; Halle, M. 1968. *The Sound Patterns of English*. New York, Harper and Row.

Church, K. 1985. *Stress Assignment in Letter to Sound Rules for Speech Synthesis*. Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics. Published by Ass. Comp. Ling.

Ciarcia, S. 1981. *Build a low-cost speech synthesizer interface*. Byte Vol.6:46-68.

Ciarcia, S. 1981. *Build an unlimited-vocabulary speech synthesizer*. Byte Vol.8:38-50.

Ciarcia, S. 1982. *Build the Microvox text-to-speech synthesizer; part 1: hardware*. Byte Vol.8:64-88.

Ciarcia, S. 1982. *Build the Microvox text-to-speech synthesizer; part 2: software*. Byte Vol.9:40-64.

Ciarcia, S. 1983. *Use ADPCM for highly intelligible speech synthesis*. Byte Vol.6:35-49.

Costello, J.; Mozer, F. 1984. Chapter 9 *Chips Using Time Domain Synthesis*in *Electronic Speech Synthesis. Techniques, Technology, and Applications* Bristow (ed.) Granada Publ. Ltd. pp. 346.

Cooper, F.S.; Liberman, A.M.; Borst, J.M. 1951. *The Interconversion of Audible and Visible Patterns as a Basis for Research in the Perception of Speech*. Proc. Natl. Acad. Sci. (U.S.) 37:318-325.

Crystal, D. 1969. *Prosodic systems and intonation in English*. Cambridge University Press.

Culicover, P.W.; Rochemont, M. 1983. *Stress and focus in English*. Language 59(1):123-165.

Denning, P.J.; Dennis, J.B.; Qualitz, J.E.; 1978. *Machines, Languages, and Computation* Prentice Hall of Canada Ltd. Toronto.

Dudley, H. 1939. *The Vocoder*. Bell Labs. Record. 18:122-126.

Dilts, M., 1984. Chapter 6 in *Electronic Speech Synthesis. Techniques, Technology, and Applications* Bristow (ed.) Granada Publ. Ltd. pp. 346.

Dudley, H.; Riesz, R.R.; Watkins, S.A. 1939. *A Synthetic Speaker*. Jour. Franklin Inst. 227:739-764.

Electronics (Feb. 23) 1984. *VLSI voices*. Mcgraw Hill Publ., pp.134-136.

Elovitz, H.S.; Johnson, R.; McHugh, A.; Shore, J.E. 1976. *Letter-to-sound rules for automatic translation of English text to phonemes*. IEEE. Transactions on Acoustics, Speech, and Signal Processing (ASSP) Vol.24(6):446-459.

Epstein, R. 1965. *A Transistorized Formant-Type Synthesizer*. Status Report on Speech Research SR-1, part 7, Haskins Labs.

Estes, S.E.; Kerby, H.R.; Maxy, H.D.; Walker, R.M. 1964. *Speech Synthesis from Stored Data*. IBM Jour., 8:2-12.

Fant, G. 1959. *Acoustic Analysis and Synthesis of Speech with Applications to Swedish*. Ericsson Technics 1:1-106. (from Klatt, 1980).

Fant, G. 1960. *Acoustic Theory of Speech Production*. 's-Gravenhage: Mouton & Co., The Hague. (from Bristow, 1984)

Fant, G.; Martony, J.; 1962. *The Instrumentaion for Parametric Synthesis (OVE II)*. Speech Trans. Labs QPSR 18-24 Royal Inst. of Tech., Stockholm (from Klatt, 1980).

Fant, G. 1973. *Speech Sounds & Features*. The MIT Press Cambridge, Mass.

Flanagan, J.L., 1957. *Note on the Design of Terminal Analog Speech Synthesizers*. J. Acoust. Soc. Am. 29:306-310.

Flanagan, J.L.; Coker, C.H.; Bird, C.M. 1962. *Computer Simulation of a Formant Vocoder Synthesizer*. J. Acoust. Soc. Am. 35:2003(A).

Flanagan, J.L.; Ishizaka, K.; Shipley, K.L. 1975. *Synthesis of Speech from a Dynamic Model of the Vocal Cords and Vocal tract*. Bell System Tech. J. 54:485-506.

Flanagan, J.L., 1972. *Speech Analysis Synthesis and Perception*. Second Ed. Springer-Verlag, New York.

Flanagan, J.L., 1984. Chapter 4 in *Electronic Speech Synthesis. Techniques, Technology, and Applications* Bristow (ed.) Granada Publ. Ltd. pp. 346.

Fry, D. 1955. *Duration and Intensity as Physical correlates of Linguistic Stress*. Jour. of the Acoustic Soc. of America 27:765-768.

Gaitenby, J.H. 1961. *Word Reading Device: Experiments on the Transposability of Spoken Words*. (abstract), Jour. Acoust. Soc. Amer. 33:1664.

Gaitenby, J.H. 1967. *Rules for Word Stress Analysis for Conversion of Print to Synthetic Speech*. (abstract), Jour. Acoust. Soc. Amer. 42:1182.

Gazdar, G.; Klein, E.; Pullum, G.; and Sag, I. 1985. *Phrase Structure Grammar*. Blackwell.

Gilblom, D.L. 1982. *A high-quality real-time text-to-speech converter*. in Proc. Electro '82 Session 11, May, 1982.

Gilblom, D.L. 1983. *Applications of text-to-speech conversion*. in Proc. Electro '83 Session 23, April, 1983.

Gilblom, D.L. 1984. Chapter 12.4 in *Electronic Speech Synthesis. Techniques, Technology, and Applications* Bristow (ed.) Granada Publ. Ltd. pp. 346.

Gillot, T. 1985. *The Simulation of Stress Patterns in Synthetic Speech - a Two-level Problem*. Interim report, Dept of Artificial Intelligence, University of Edinburgh.

Gimson, A.C. 1966. *The linguistic relevance of stress in English* in *Phonetics and Linguistics*. Jones W.E.; Laver, J.(ed.) Longmans, London. pp. 94-102

Gray, H. 1974. *Anatomy, descriptive and surgical*. T.P. Pick, and R. Howden.(eds.) Running Press, Philadelphia, Penn.

Halliday, M.A.K. 1970. *A Course in spoken English: Intonation*. Oxford University Press, London England.

Harris, C.M. 1953. *A Study of Building Blocks in Speech*. Jour. Acoust. Soc. Amer., 25:962-969.

Harris, C.M. 1953. *A Speech Synthesier*. Jour. Acoust. Soc. Amer., 25:970-975.

Hertz, S.R 1982. *From text to speech with SRS*. J. Acoust. Soc. Am. 72(4):1155-1170.

Hess, W. 1983. *Pitch Determination of Speech Signals. Algorithms and Devices*. Springer-Verlag. New York.

Hill, D.R. 1972. *A Basis for Model Building and Learning in Automatic Speech Pattern Discrimination.* in *Machine Perception of Patterns and Pictures. Proc. Inst. Physics/Inst. E.E/National Physical Laboratory Conf. At Teddington (NPL) April* pp. 151-160.

Hill, D.R.; Reid, N.A. 1977. *An experiment on the Perception of Intonational Features*. Int. J. Man-Machine Studies (9):337-347.

Hill, D.R.; Witten, I.H.; Jassem, W 1978. *Some Results from a Preliminary Study of British English Speech Rhythm*. Man-Machine Systems Laboratory Report no. 78/26/5 Univ. of Calgary.

Hill, D.R. 1980. *Spoken Language Generation and Understanding by a Machine: A Problems and Applications Oriented Overview.* in *Spoken Language Generation and Understanding* J. C. Simon. (ed.) pp. 3-38.

Holmes, J.N.; Mattingly, I.; Shearme, J. 1964. *Speech Synthesis by Rule*. Language Speech 7:127-143.

Holmes, J.N. 1973. *The Influence of the Glottal Waveform on the Naturalness of Speech From a Parallel Formant Synthesizer*. IEEE Trans. Audio Electroacoust. 298-305.

Ingemann, F. 1957. *Speech Synthesis by Rule*. (abstract) Jour. Acoust. Soc. Amer., 29:1255.

Jassem, W.; Hill, D.R.; Witten, I.H. 1984. *Isochrony in English Speech: its Statistical Validity and Linguistic Relevance.* in *Intonation, Accent and Rhythm; Studies in Discourse Phonology.* D. Gibbon and H. Richter. (eds.) Walter de Gruyter, Berlin.

Jassem, W.; Nolan, F. 1984. Chapter 3 in *Electronic Speech Synthesis. Techniques, Technology, and Applications* Bristow (ed.) Granada Publ. Ltd. pp. 346.

Kaplan, G.; Lerner, E.J. 1985. *Realism in Synthetic Speech*. IEEE. Spectrum (Apr):32-37.

Klatt, D.H. 1972. *Acoustic Theory of Terminal Analog Speech Synthesis*. Proc. 1972 Int. Conf. on Speech Comm. and Proc. IEEE Catalog No. 72 CH0567-7 AE, 131-135.

Klatt, D.H. 1975. *Vowel Lengthening is syntactically Determined in a connected discourse*. Jour. of Phonetics 3:129-140.

Klatt, D.H. 1976. *A Cascade-Parallel Formant Synthesizer and a Control Strategy for Consonant vowel Synthesis*. J. Acoust. Soc. Am. Supplement 1, Vol. 61:S68.

Klatt, D.H. 1976. *Linguistic uses of segmental duration in English: Acoustic and perceptual evidence*. J. Acoust. Soc. Am. Vol. 59:1208-1221.

Klatt, D.H. 1976. *Structure of a phonological rule component for a synthesis-by-rule program*. IEEE. Transactions on Acoustics, Speech, and Signal Processing (ASSP) Vol.24(5):391-398.

Klatt, D.H. 1982. *The Klattalk Text-to-Speech Conversion System*. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) Vol.3:1589-1592.

Koehler, L.M.; Mackey, T.C. 1984. *Speech Output for HP-Series 80 Personal Computers*. Hewlett-Packard Journal. (Jan):29-36.

Koenig, W.H.; Dunn, H.K.; Lacey, L.Y. 1946. *The Sound Spectrograph*. Jour. Acoust. Soc. Am. Vol. 18:19-49.

Ladefoged, P. 1967. *Three areas of experimental phonetics*. Oxford University Press, London.

Lakoff, G. 1972. *The global nature of the nuclear stress rule*. Language 48(2):285-303.

Lawrence, W. 1953. *The Synthesis of Speech from Signals which have a Low Information Rate*. in Communication Theory Ed. by W. Jackson Butterworths, London. pp. 460-469.

Lee, F.F. 1969. *Reading machine: from text to speech*. IEEE. Transactions on Audio & Elecroacoustics Vol.17(4):275-282.

Lehiste, I.; Peterson, G.E. 1959. *Vowel amplitudes and phonemic stress in American English*. Jour. Acoustical Soc. of Amer. 54:1228-1234.

Lehiste, I. 1970. *Suprasegmentals*. MIT Press, Cambridge Massachusetts.

Levinson, D.C. 1982. *The Well-Tempered Speech Recognizer*. Doctoral Thesis. University of Calgary, Calgary Alberta.

Liljencrants, J. 1968. *The OVE-III Speech Synthesizer*. IEEE Trans. Audio Electroacoust. AU-16, 137-140.

Long, L. 1968. *Introduction to Computers and Information Processing*. Prentice-Hall Inc., Englewood Cliffs, New Jersey. pg. 567

Makhoul, J. 1984. Chapter 5 Linear Predictive Coding in *Electronic Speech Synthesis. Techniques, Technology, and Applications* Bristow (ed.) Granada Publ. Ltd. pp. 346.

Mattingly, I.G. 1968. *Synthesis by Rule of General American Enlgish*. Supplement to: Status Report on Speech Research. Haskins Laboratories, New York.

McIlroy, M.D. 1974. *Synthetic English Speech by Rule*. Bell Telephone Laboratories Inc. Murray Hill, New Jersey.

Miastkowski, S. 1982. *Add a voice to your computer*. Popular Computing Vol.6:81-86.

Olive, J.P. 1974. *Speech synthesis by rule*. Proc. of the Speech Communication Seminar, Stockholm, Sweden.

Peterson, G.E.; Wang, W.S.-Y.; Silvertsen E. 1953. *Segmentation Technique in Speech Synthesis*. Jour. Acoust. Soc. Am. Vol. 30:739-742.

Pierrehumbert, J. 1981. *Synthesizing intonation*. J. Acoust. Soc. Am. 70(4):985-995.

Pike, K.L. 1945. *The intonation of American English*. University of Michigan Press, Ann Arbor, Michigan.

Rabiner, L.R.; Jackson, L.B.; Schafer, R. W.; Coker, C.H. 1971. *A Hardware Realization of a Digital Formant Speech Synthesizer*. IEEE Trans. Comm. Tech. COM-19 1016-1070.

Rice, D.L. 1976a.(*Hardware and software for speech synthesis*. Dr. Dobb's Journal of Computer Calisthentics and Orthodontia, Vol.4:6-8.

Rice, D.L. 1976b. *Friends, humans, and countryrobots: lend me your ears*. Byte Vol.12.

Rice, D.L. 1977. *Speech synthesis by a set of rules (or can a set of rules speak English?)*. in Proc. of the First West Coast Computer Faire, San Francisco, 1977.

Schubert, L.K.; Pelletier, F.J. 1982. *From English to Logic: Context-free computation of conventional logical translations*. Proc. of the International Joint Conference on Artificial Intelligence.

Schubert, L.K. 1984. *On Parsing Preferences*. Proc. of COLING-84, Stanford; pg.247-250.

Sclater, N. 1983. *Introduction To Electronic Speech Synthesis*. Howard W. Sams & Co. Inc. Indianapolis, Indianna.

Scott, R.J.; Glace, E.M.; Mattingly, I. 1966. *A Computer-Controlled On-Line Speech Synthesizer System*. 1966 IEEE. Int. Comm. Conf., Degest of Tech. Papers, Philadelphia, 104-105.

Sherwood, B.A. 1978. *Fast text-to-speech algorithms for Esperanto, Spanish, Italian, Russian and English*. Int. J. Man-Machine Studies Vol.10:669-692.

Smith, E.H. 1984. *Five Voice Synthesizers*. Byte Vol.9:337-346.

Stevens, K.N.; Bastide, R.P.; Smith, C.P. 1955. *Electrical Synthesizer of Continuous Speech*. J. Acoust. Soc. Am. 27:207(A).

Vincent, A.T. 1982a. *Computer assisted suport for blind students - the use of a microcomupter linked voice synthesizer*. Computers and Education 6:55-60.

Vincent, A.T. 1982b. *CAL for blind students: some recent developments*. Microcomputers and Teaching Project Report, Milton Keynes, England: Open University. (from Bristow, 1984).

Witten, I.H.; Abess J. 1979. *A microcomputer based speech synthesis-by-rule system*. Int. J. Man-Machine Studies 11:585-620.

Witten, I.H.; 1979. *On transferring fundatmental-frequency contours from one utterance to another*. J. Acoust. Soc. Am. 65(6):1576-1579.

Witten, I.H.; 1982. *Principles of Computer Speech*. Academic Press, Toronto. 286 pp.

Young, S.J.; Fallside, F. 1979. *Speech Synthesis from concept: A method for speech output from information systems*. J. Acoust. Soc. Am. 66(3):685-695.

Silicon Systems Inc: Phone Codes by Hex Value

\	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	¬1	E1	E11	y1	Y11	ay1	ie1	I1	A1	A11	EH1	EH11	AE1	AE11	AH1	AH11
1	AW1	O1	OU1	OO1	iu1	iu11	U1	U11	UH1	UH11	UH21	UH31	ER1	R1	r11	r21
2	L1	L11	LF1	W1	B1	D1	KV1	P1	T1	K1	HV1	HVC1	HF1	HFC1	HN1	Z1
3	S1	J1	SCH1	V1	F1	THV1	TH1	M1	N1	NG1	a1	oh1	u1	uh1	e21	lb1
4	¬2	E2	E12	y2	Y12	ay2	ie2	I2	A2	A12	EH2	EH12	AE2	AE12	AH2	AH12
5	AW2	O2	OU2	OO2	iu2	iu12	U2	U12	UH2	UH12	UH22	UH32	ER2	R2	r12	r22
6	L2	L12	LF2	W2	B2	D2	KV2	P2	T2	K2	HV2	HVC2	HF2	HFC2	HN2	Z2
7	S2	J2	SCH2	V2	F2	THV2	TH2	M2	N2	NG2	a2	oh2	u2	uh2	e22	lb2
8	¬3	E3	E13	y3	Y13	ay3	ie3	I3	A3	A13	EH3	EH13	AE3	AE13	AH3	AH13
9	AW3	O3	OU3	OO3	iu3	iu13	U3	U13	UH3	UH13	UH23	UH33	ER3	R3	r13	r23
A	L3	L13	LF3	W3	B3	D3	KV3	P3	T3	K3	HV3	HVC3	HF3	HFC3	HN3	Z3
B	S3	J3	SCH3	V3	F3	THV3	TH3	M3	N3	NG3	a3	oh3	u3	uh3	e23	lb3
C	¬4	E4	E14	y4	Y14	ay4	ie4	I4	A4	A14	EH4	EH14	AE4	AE14	AH4	AH14
D	AW4	O4	OU4	OO4	iu4	iu14	U4	U14	UH4	UH14	UH24	UH34	ER4	R4	r14	r24
E	L4	L14	LF4	W4	B4	D4	KV4	P4	T4	K4	HV4	HVC4	HF4	HFC4	HN4	Z4
F	S4	J4	SCH4	V4	F4	THV4	TH4	M4	N4	NG4	a4	oh4	u4	uh4	e24	lb4

Table 7.1; Numerically Organized SSI Phoneme Codes

Phone Code Equivalence Chart

Dectalk	Webster's New Collegiate Dictionary	IPA	VOTRAX	SAM	NRL	SSI	Sample Words (American English)
b	b	b	B	B	B	B	bat, tab
ch	ch	t•2	CH	CH	CH	T-SCH	church, char
d	d	d	D	D	D	D	dub, bud
f	f	f	F	F	F	F	fat, ruff, photo, laugh
						HV	eh
						HVC	d(h)ouble
hx	h	h	H	HH	/X, /H	HF	hat, home
						HFC	p(h)ad, fluff(h)
					/H	HN	hnh-hnh
jh	j	dz	J	JH	J	J	job, rage
k	k	k	K	K	K	K	kit, tick
g	g	g	G	G	G	KV	big, gag
l	l	l	L	L	L	L	lab, ball
						L1	plan, club, slave
el	•ll	l	L		UL	LF	bottle, channel
m	m	m	M	M	M	M	mad, dam
n	n	n	N	N	N	N	not, ton
nx	nj	nj	NG	NX	NX	NG	ring, rang
p	p	p	P	P	P	P	pat, tap
r	r	r	R	R	R	R	rat
s	s	s	S	S	S	S	sat, lass
sh	sh	•2	SH	SH	SH	SCH	shop, push
t	t	t	DT, T	T	T	T	tap, pat
dh	th	ð	THV	DH	DH	THV	bathe, the
th	th	•3	TH	TH	TH	TH	bath, theory
v	v	v	V	V	V	V	vow, pave
w	w	w	W	W	W	W	why, quake(kwake)
w	w	hw	H-W	WH	WH	W	where, which
y	y	j	Y1	Y	Y	Y1	you
z	z	z	Z	Z	Z	Z	zap, maze
zh	zh	z	ZH	ZH	ZH	E-J	leisure
			PA0, PA1		PA0, PA1	blank	[pause]
ey	ā	e	A1, A2, A, AY	EY	EY	A	day
ah, ix	•1	e				A1	care
ae	a	æ	AE	AE	AE	AE	laugh, dad, advent
ae	a	æ	AE1	AE	AE	AE1	ask
ax	•1	•1	AH, AH1, AH2	AA	AA	AH	about
aa	ä	a	UH2	AX	AX	AH1	father, top
ao	ȯ	•4	AW, AW1, AW2	AO	AO	AW	saw, caught
aw	aů	aU	AH-O1	AW	AW	AW-U	how, growl
iy	ē	i	E, E1, Y	IY	IY, IX	E	beet, be
eh	e	Σ			EH, EH1	E1	advent
eh	e	Σ	EH2, EH3	EH	EH	EH.	leg, said
						EH1	silent
		•5	ER	ER	ER	ER	third, urn, heard
ih	i	I	I, I1, I2, I3	IH	IH	I	sit, bid
ay	ī	aI	AH-E1	AY	AY	AH2-E	bite, silent
ow	ō	o	O, O1, O2	OW	OW, OH	Θ	boat, abode
uh	ů	U	OO, OO1	UH	UH	OO	put, pull, look
yu	yü		Y			Y1-U	cute
						OU	orb

oy	ɔi	*41	O1-E1	OY	OY	O-E	bo*y*, bofl
uw	u	u	U, IU	UW	UX, UW	U	b*oo*t, *you*, f*oo*l
			U1			U1	p*oo*r
ah, ix	*1	*5	UH, UH1	AH	AH	UH	c*u*p
				UM		U-M	astrono*my*
en	*1n						butt*on*
			UH3		UN	UH1	circ*us*
			UH3		UN	UH2	nation(naesh*u*n)
						ay	franc*ais* (French)
						a	*e*´tre (French)
						e2	sh*o*n (German)
						ie	*i*l (French)
						iu	p*eu*t (French)
						iu1	G*oe*the (German)
						oh	men*u*, t*u* (French)
						u	f*uh*len (German)
						uh	men*u*, t*u*` (French)
						y	*y* (French)
						lb	i*l* (French)
						r1	*r*eponse (French)
						r2	*r*ichtig (German)
						YX	DIPTHONG ENDING
						WX	DIPTHONG ENDING
						RX	'r' after a vowel
						LX	'l' after a vowel
						DX	flap as in pity

Table 7.2; Alphabetic list of Phoneme codes and IPA equivalents [70]

[70]

1 - schwa (upside-down and backwards 'e').
2 - integral sign.
3 - theta.
4 - backwards 'c'.
5 - backwards small epsilon.

8. Appendix B

This Appendix contains the entire letter to phoneme rule set that is used by Talker. The rules are divided up into tables according to the same categories that are used internally to Talker.

Table 3.5 in chapter 3 provides a complete explanation of the symbols used in this appendix.

Each rule translates its target string into a set of hexadecimal code numbers. There are 2 hexadecimal digits for each phonetic code. Table 7.1 in Appendix A provides a numerically organized way of determining which hexadecimal code applies to which phonetic code. Table 7.2 in Appendix A is a phonetic code equivalence chart which provides sample words for each phonetic code, and the equivalent code for the same sound in other systems.

8.0.1 "A" Rules

Rule No.	LTPR	Rule No.	LTPR
1	![A]!=0804	2	[A]!=1A
3	[AA]=0E	4	![A]BO=18
5	[AGAIN]=1AA9A60A38	6	#:[AG]E=072531
7	OUNT[AI]N=0A	8	#^:[AI]N=0A
9	[A]IR=08	10	[AI].=484804
11	[AI]=0804	12	[AL]F!=8C4C
13	[AL]K=5050	14	#:[A]LLY=
15	#:[AL]!=20	16	#:[ALS]!=202F
17	[AL]M=5050	18	[AL]V=4C4C
19	[A]L.=5050	20	[A]LL=5050
21	[A]L^=10	22	![A]L#=1A
23	[A]MENT=4E	24	![A]ND=0A
25	#:[A]NCE!=0A	26	[ANG]+=08043831
27	#^[A]NT!=47	28	~[A]NT=4A
29	!:[A]NY=0A	30	[A]QUA:=10
31	!:[AR]!=4E4E5C	32	[AR]!=1C
33	^[ARA]=0B5C8C	34	[AR]D=4E4E5C
35	![ARE]!=4E4E5C	36	![ARE]N'T!=0F9C
37	![A]RO=1A	38	![A]RRO=0C
39	![A]RR=1A	40	[A]RR=0C
41	[AR]#=085C	42	[AR].=4E4E5C
43	[AR]:=0E5C	44	![A]SS#=0E
45	!^[A]S#=0804	46	[ATEAU]=0C281163
47	[ATE]=088428C0	48	[ATH]E!=484835
49	[A]TIN=0C	50	!^[A]TIONAL=0C
51	[A]TORY=58	52	[A]TURE=18
53	[AUSE]=50502F	54	[AU].=5050
55	[AU]=10	56	![AVE]!=0C33EC4A4A780416
57	[A]WA=1A	58	[AW]=5050
59	![A]X=0C	60	[AY]=484804
61	[A]^ACE=0E	62	[A]^AT=4C4C
63	[A]^EFUL=0804	64	[A]^EMENT=0804
65	[A]^ET=4C4C	66	^[A]^IC=8C4C
67	[A]^ID=4C4C	68	[A]^ISH=4C4C
69	[A]^IT=4C4C	70	[A]^LE=1B
71	!:[A]^L#=0804	72	[A]^OON=0C
73	[A]^OROUS=4C4C	74	[A]^OT=4C4C
75	[A]^RE!=4848	76	[A]^RIC=0C
77	!:[A]^R#=0804	78	![A]^T#=0C
79	[A]^UT=4C4C	80	![A]^:#=0E
81	[A]^+#=0804	82	[A]^+:#=0C
83	:[A]^#=0804	84	[A].=4C0C
85	[A]=0C		

8.0.2 "B" Rules

Rule No.	LTPR	Rule No.	LTPR
86	![B]!=2401	87	![BED]:=244A4A25
88	![BENE]=240A3807	89	![BE]ˆ:#=2441
90	[BE]ING=2401	91	![BIG]=2407E6C0
92	![BI]ˆ#=244E4E84	93	[BIO]=24010E
94	![BI]#=240E84	95	![BO]TH!=245151
96	[BOUGH]!=244E4E16	97	[BOW]L=245151
98	[BOW]=241016	99	![BREA]K=241D08
100	[BROW]=241D1016	101	[BT]=28
102	![BUS]#=24072F	103	![B]L=24
104	![B]R=24	105	![B]ˆ:=
106	M[B]ING!=	107	M[B]S!=
108	M[B]!=	109	[BB]=24
110	[B]=64		

8.0.3 "C" Rules

Rule No.	LTPR	Rule No.	LTPR
111	![C]!=3001	112	![CAPI]=290C2747
113	[CC]+=2930	114	[CEA]N=3218
115	[CES]!=304B2F	116	![CERTAIN]=301C680A38
117	EX[C]E=	118	![CH]ˆ=29
119	![CH]ARACTER=29	120	![CHOIR]=29630E845C
121	![CH]OR=29	122	ˆ+[CH]=29
123	[CH]!=283280	124	[CH]=2832
125	!S[CI]#=304E4E84	126	[CI]AL=32
127	[CIA]N!=3218	128	[CI]A=3201
129	[CI]O=32	130	[CI]EN=32
131	[C]+=30	132	[CKG]=E6C0
133	[CK]=29ADC0	134	[CO]M=29AD5A5A
135	[CO]ST=29ADC05050	136	[COU]NTR+=29AD18
137	[COUR]SE=29AD111C	138	[COUS]IN=29182F
139	[COW]=294F63	140	[CRACY]=291D183001
141	[CREA]TU=291D01	142	[CREA]T#=291D010804
143	![CZ]#=2F	144	![CZ]+=2832
145	![C]=292D	146	[CC]=69
147	[C]=29		

8.0.4 "D" Rules

Rule No.	LTPR	Rule No.	LTPR
148	![D]!=2501	149	[DDED]!=250725
150	#:[DED]!=250725	~~151~~	E[D]!=25
152	#ˆ:E[D]!=28	153	~~![DEA]TH=250A~~
154	![DELI]=250A2007	155	![DE]ˆ#=2507
156	![DIA]L=250E8458	157	![DIAR]=250E449C
158	:[DIER]=25311C	159	![DIO]=25010E
160	![DIS]=250730	161	[DG]=2531
162	[DJ]=2531	163	![DO]!=65AB16A3
164	[DOCTRI]N=255829281D	165	![DO]ING!=65AB16A3
166	![DOES]=A5AB182F	167	![DO]NE=A5AB18

168	[DOUGH]=251256	169	![DOW]=250F17
170	![DR]!=250E29281C	171	![DR]=65715C
172	![DU]A=255656	173	[DU]A=3116
174	#[DUR]=B11C	175	![D]=A5AB
176	[D]!=65C0	177	[DD]=65
178	[D]=65		

8.0.5 "E" Rules

Rule No.	LTPR	Rule No.	LTPR
179	![E]!=4141	180	#:[E]!=
181	[E]MENT=	182	!:[E]!=01
183	#:[EA]!=011A	184	[ED]!=25
185	#:[ELY]!=2001	186	[EA]D=4A4A
187	[EAR]N=5C5C	188	![EAR]^=1C
189	[EA]RT=0E	190	[EA]TURE=01
191	[EAU]!=1163	192	[EAU]=8416
193	![EA].=4141	194	![EA]=0181
195	R[EA]LI^=018D	196	[EA]:E!=01
197	[EA]:ED!=01	198	[EA]:EE=01
199	[EA]:ES!=01	200	[EA]:ING=01
201	[EA].ON=4141	202	[EA]^ON=01
203	[EA]^TH!=0A	204	[EA]:#=0A
205	[EA]=C201	206	[EE].=0141
207	[EE]=01	208	[EF]UL=34
209	[EH]#=01	210	[EIGH]=0804
211	[EIN]=084438	212	[EI]=01
213	![ELA]B=01200E	214	![ELE]=0A204A
215	[EL].=0A4A60	216	[EL]^=0A60
217	[ENE]=0138	218	#:EM[E]NT=0A
219	!:[EN]!=0A38	220	[EN]!=7878
221	[EN]S!=7878	222	[E]O=01
223	[E]QU=02	224	^[ERE]!=021C
225	[ERE]LY=411C	226	[ERI]#=011D01
227	[ERI]=0A1D07	228	#:[ER]#=0A1D
229	![ERR]=0A5C	230	![ER]:#^=0B1D
231	[ER]#=0A1D	232	[ER].=1C5C
233	[ER]=5C	234	H[ES]!=0A2F
235	[E]SIVE=01	236	#:[E]S!=
237	L[EU]=16	238	R[EU]=16
239	[EU]=8416	240	[EVE]NING=0133
241	![E]VEN=01	242	[EVER]=0A331C
243	#:[E]W=	244	L[EW]=5656
245	R[EW]=5656	246	[EW]=845656
247	![EXTRA]=0A2930281D4E	248	![EX.]!=4A29304C0C372760
249	![EYE]=4E4E84	250	[EY]=0884
251	^[E]^AL=01	252	^[E]^EA:=01
253	^[E]^EOU=01	254	^[E]^EO=01
255	^[E]^ET=0A	256	[E]:E:!=4141
257	[E]^E:!=01	258	^[E]^EU=01
259	^[E]^IA:=01	260	^[E]^IE=01
261	^[E]^IOU=01	262	^[E]^IO:=01
263	^[E]^ISH=0A	264	^[E]^IU=01
265	!:[E]^RIC=0A	266	!:[E]^L#=01
267	!:[E]^R#=01	268	[E].=4A4A
269	[E]=4A		

8.0.6 "F" Rules

Rule No.	LTPR	Rule No.	LTPR
270	![F]!=4B34	271	[FA]THER=344F4F
272	![FING]ER=340739E6C0	273	[FOO]D=345663
274	![FORE]=34111D	275	[FOUR]=34111C
276	![FUL]=745860	277	[FUL]=7420
278	![F]=342C	279	[FF]=74AC
280	[F]=74AC		

8.0.7 "G" Rules

Rule No.	LTPR	Rule No.	LTPR
281	![G]!=3101	282	B#[G]=E6C0
283	![GE]N=25310A	284	[GE]OUS=656531
285	![GEO]=2531010E	286	![GER]=25311C
287	[G]ER!=E6E9	288	![GES]=25310A30
289	[GE]T=E6C00A	290	SU[GGES]=2531710A30
291	[GG]=E6C0	292	![GH]=E6E9
293	#[GH]=	294	![GI]N=3107
295	[GI]V=E9E607	296	![G]I'=E9E6
297	[G]N!=	298	#[G]N#=65B1
299	![G]N=	300	![GO]!=E9E61163
301	![GO]NE=E9E64E4E	302	#[GRA]PHY=E6E91E8F
303	[GREA]T=E9E65C4884	304	![GROW]N=E9E65D1163
305	[GUE]!=E6C0	306	![GUI]D=E9E6585884
307	[GUIS]=E9E65858842F	308	[GURE]!=E6E9445C
309	![GYM]=B1078737	310	![G]YN=E9E6
311	![GYR]'=B101411D	312	![GYR]=65310E041D
313	[G]+=31	314	[G]!=E6C0
315	![G]=E9E6	316	[G]=E6E9

8.0.8 "H" Rules

Rule No.	LTPR	Rule No.	LTPR
317	![H]!=09042832	318	![HABI]=AC0C2447
319	![HA]STE=AC4242	320	![HAVE]=AC4C4C33C0
321	![HEAR]D=AC1C5C	322	[HEARS]#=AC5C5C30
323	![HE]IGHT=AC	324	![HEIR]=0A1C
325	![HEMA]=AC01374E	326	![HER]!=AC5C5C
327	![HERE]#=AC011C	328	![HERE]!=AC011C
329	![HOLI]=AC0E2047	330	[HOME]=AC51516337
331	![HONE]ST=4E4E384A	332	![HOSPI]=AC0E302707
333	![HOUR]=0F161C	334	[HOW]=AC0F1680
335	![HY]DR=AC0E04	336	!EX[H]=
337	[H]#=AC	338	[H]=

8.0.9 "I" Rules

Rule No.	LTPR	Rule No.	LTPR
339	[I'M]=4E4E8437	340	![I]!=4E4E84

341	[I]A!=01	342	[IA]L=4118
343	[IA]N=018C	344	[IARY]=1C01
345	[IA]TE=01840884	346	[I]A=4E4E84
347	[IC]AL!=4769	348	#:[ICE]!=0A30
349	![ICE]=0E8430	350	[I]CY=0E84
351	![IDIO]=47254151	352	![I]DI=07
353	![I]DL=4E4E84	354	[IE]!=4E4E84
355	[IE]D!=4E4E	356	[IE]ND=0B
357	[IE]N^=040B	358	!:[I]ER!=4E4E84
359	[IE]S!=4E4E84	360	[IE]T=4E4E840A
361	[IEU]=16	362	RR[IE]^=01
363	[IE].=4141	364	[IE]=01
365	[IGH]T=0E84	366	[IGH]!=4E4E84
367	[IGM]:=0E0437	368	[IGN]!=4E4E8438
369	[IGNING]=4E4E04384739	370	[IGN]^=4E4E8438
371	[I]LDER=07	372	[I]LDR=4747
373	[I]LD=4E4E84	374	[I]L.=4747
375	[I]L^=07	376	[I]MENT=47
377	![I]M=07	378	[I]NDL=07
379	[I]ND=4E4E84	380	G[I]NE=07
381	[I]NG=07	382	![I]N=47
383	#:[IO]T=041B	384	[IO]T=0E840A
385	![I]O=4E4E84	386	[I]PHE=0E84
387	[I]QUE=01	388	[IRES]!=4E445C2F
389	[IRE]!=4E445C	390	[IRE]MENT=4E445C
391	![IRO]N=4E845C	392	[IR].=1C5C
393	[IR]=5C5C	394	![IS]L=4E4E84
395	![ISO]=0E843051	396	![ITER]=07285C
397	![I]TI=4E4E84	398	[IT'S]=472830
399	[I]VEN!=07	400	[I]VER=07
401	[I]ZE=4E4E84	402	[I]^ACY=47
403	[I]^ANT=47	404	[I]^ARY=47
405	[I]^AT+=47	406	[I]^A=4E4E84
407	[I]^EFUL=4E4E84	408	[I]^EMENT=0E84
409	[I]^ENT=47	410	[I]^ET=07
411	^T[I]^E=0141	412	[I]^E=8E4E84
413	#^:[I]^LE=47	414	[I]^L#=0E04
415	[I]^R#=0E04	416	[I]^+:#=47
417	![I]^#=4E4E84	418	[I]!=0E04
419	[I].=4747	420	[I]=47

8.0.10 "J" Rules

Rule No.	LTPR	Rule No.	LTPR
421	![J]!=25310904	422	![JULY]!=3114204E4E84
423	[J]=B1		

8.0.11 "K" Rules

Rule No.	LTPR	Rule No.	LTPR
424	![K]!=290904	425	[KEY]=290184
426	![KH]=29	427	[K]K=
428	![KNOW]N=381163	429	![K]N=
430	[K]!=29ADC0	431	N[K]=A9

432	[K]=29

8.0.12 "L" Rules

Rule No.	LTPR	Rule No.	LTPR
433	![L]!=0B20	434	![LABI]=20022401
435	#[LACE]=201830	436	![LAUGH]=204D4D34
437	[LEA]D=2001	438	![LEGI]^=200A253147
439	![LENS]=200A382F	440	![LIAR]=200E041C
441	[LIAR]=20441C	442	![LIFE]=200E0434
443	![LIKE]=204E8429	444	[L]L=
445	[LO]C#=2011	446	#[LOG]+=6058F1
447	[LOP]E=201127	448	[LOP]=201927
449	![LOVE]=201833	450	#^:[L]=1A20
451	[L]=60		

8.0.13 "M" Rules

Rule No.	LTPR	Rule No.	LTPR
452	![M]!=0A37	453	![MACHI]NE=370E324141
454	![MADA]M=370C2558	455	![MALA]=370C2018
456	![MALE]!=37424220	457	![MALE]S=37424220
458	![MALE]=370C204A	459	![MANE]=37424238
460	![MANI]=370C3847	461	![MAN]=370C38
462	![MATE]!=37424228	463	![MATE]=374E280A
464	![MAYBE]!=3702444424	465	![MEA]NT=370A
466	![MEN]!=370A38	467	![META]=370A2818
468	^[MINE]=374738	469	![MINU]TE=37073818
470	![MODE]!=3751516325	471	![MOD]=370E25
472	![MONEY]=37193801	473	![MONK]=37183829C0
474	![MON]=370E38	475	[MOV]=371633
476	![MI]C=774E84	477	![MILI]=37072047
478	!![MM]=3737	479	[M]M=
480	![M]N=	481	[M]N!=
482	![MR]!=370730281C	483	![MRS]=370730472F
484	[M]=37		

8.0.14 "N" Rules

Rule No.	LTPR	Rule No.	LTPR
485	![N]!=0A38	486	![NA]TURE!=3808
487	+[NG]+=3831	488	#[NG]=39
489	[NN]=38	490	![NOMI]=380E3747
491	![NO]NE!=385858	492	[NO]N=380E
493	[NO]THING=385858	494	![NOWHERE]!=3811230A
495	![NOW]=381016	496	[NQU]=78A963
497	![NUIS]=381630	498	[N T]=3828C0
499	[N]=78		

8.0.15 "O" Rules

Rule No.	LTPR	Rule No.	LTPR
500	![O]!=1117	501	[O]!=1163
502	[OA]=1163	503	![O]DD=4E4E
504	[O]E=11A3	505	[OF]!=1A33C0
506	![OFFI]=103447	507	[O]F^=10
508	.[O]G=0E0E	509	![OH]!=1163
510	[OI].=515101	511	[OI]=1101
512	![OK]!=116329484804	513	[O]LD=5151
514	[OLK]=116329	515	~~[O]LOO=0E8E~~
516	^#:[O]L=18	517	[O]LL=11
518	[OL]M=5151	519	[O]L.=5151
520	#^:[O]M=18	521	![ON]!=0E38
522	![O]NCE=2318	523	![O]NE=2318
524	![O]NLY=11	525	I[O]N=5A
526	[O]N'T=11	527	!^:[O]N=0E
528	[ON]S!=7878	529	#:[O]N!=5A
530	#^[O]N=5A	531	[ON]!=7878
532	L[OO]D=5858	533	[OO]D=5252
534	[OO]F=5656	535	[OO]K=13
536	[OOR]=115C	537	[OO]=5663
538	[OROUGH]=5C1E5163	539	#:[OR]!=1C
540	#:[ORS]!=1C2F	541	[O]RR=11
542	![ORDI]=515C2547	543	[OR].=51515C
544	[OR]=115C	545	[O]S+=1163
546	[O]SS!=10	547	[O]ST!=11
548	[O]THER=18	549	![O]VER=11
550	R[O]V=11	551	T[O]V=11
552	[O]V=18	553	!ALL[OW]=0E63
554	![OW]L=1016	555	![OW]=1163
556	[OW]!=11E3	557	[OW]=1016
558	![OXY]=4E293047	559	[O]^AGE=0E
560	[O]^A=1163	561	[O]^E=11A3
562	[O]^I#=1163	563	[O]^ICE=51A3
564	[O]^L#=11	565	[O]^R#=11
566	[O]^U=11	567	[O]^Y=11
568	[OUGHT]=1028	569	[OUGH]=505034
570	[OULDER]=116360255C	571	[OULD]=955565
572	[OU]N^#=0E63	573	[OU]PL=18
574	[OU]P=16	575	![OUR]=0E635C
576	[OUR]:#=1C	577	[OUR]^=111C
578	[OU]S#=0E12	579	[OU]S=1A
580	![OUT]=101628	581	^[OU]^L=18
582	[OU]:#=16	583	[OU]=0E63
584	[OY]=1101	585	[O]=0E

8.0.16 "P" Rules

Rule No.	LTPR	Rule No.	LTPR
586	![P]!=2701	587	[P]!=27ECC0
588	![PARLIA]=27181C2018	589	[PA]STE=270804
590	![PATE]=270E2858	591	#:[PB]#=24
592	![PECU]=2707290416	593	![PENE]=270A384A
594	![PHOTO]=341123285163	595	![PHYS]=3447472F

596	[PH]=34	597	[PPH]=34
598	[PEOP]=270127	599	![PN]=38
600	![POE]T=27110A	601	[PO]P+=2711
602	[PO]P=270E	603	![POSSE]=2711232F0A
604	![POSSI]=270E3047	605	[POUL]=271120
606	[POUR]=27111C	607	[POW]=271016
608	[PP]=27	609	![PRETT]=271D4728
610	[PRO]VE=271D16	611	[PROO]F=271D16
612	[PRO]'=271D11	613	[PSEUDO]=3057576591A3
614	[PSYCH]=304E4E8429	615	![PS]=30
616	![PT]=28	617	CEI[PT]=28
618	[PU]BLI=2718	619	![PU]NISH=2718
620	[PUT]!=271328C0	621	[PY]:A=2707
622	[PY]:O=274E4E84	623	![P]=276D
624	[P]=27		

8.0.17 "Q" Rules

Rule No.	LTPR	Rule No.	LTPR
625	![Q]!=290316	626	A[QUAR]+=2963085C
627	[QUAR]=2963119C	628	[QUAI]=29630884
629	[QUA]=29230F	630	#[QUET]!=2908
631	'[QUET]!=294A68	632	![QUEUE]=29445656
633	[QUE]!=29	634	[QUI]V=292307
635	[QU]=2923	636	[Q]=29

8.0.18 "R" Rules

Rule No.	LTPR	Rule No.	LTPR
637	![R]!=0E5C	638	![RE]ACT=1D01
639	![READY]=1D4A4A2501	640	![READ]=1D414125
641	![REC]+=1D0130	642	![REC]=1D0A29
643	![RE]'#=1D01	644	[RE]D=1D0A
645	[RHO]M=1D0E	646	[RHO]=1D1163
647	[RHY]TH=1D07	648	[RH]=1D
649	[RINE]!=1D0138	650	![RI]V=[illegible]
651	TH[ROUGH]=1D16	652	OR[illegible]
653	UR[R]=	654	[RR]=1D
655	![RUN]=1D1838	656	[R]=1D

8.0.19 "S" Rules

Rule No.	LTPR	Rule No.	LTPR
657	![S]!=0A30	658	[SAI]D=304A4A
659	![SAT]U=300C28	660	![SAYS]!=304A4A2F
661	![SCH]=3029	662	[S]C+=
663	#[SED]!=2F25	664	![SEMI]=300A3747
665	![SEW]!=301163	666	![SHOE]=323C5656
667	![SHOW]N=321116	668	[SH]=32
669	[SSI]O=32	670	[SI]O=32
671	[SI]VE=3047	672	#[S]M=2F
673	[SOME]=30585877	674	![SON]=301838

675	[SOU]L=3011	676	[SO]URCE=3011
677	![SPE]CIAL=30270A	678	#[ST]EN=30
679	[STHM]=3037	680	[ST]LE!=30
681	![ST.]!=304848047868C0	682	![ST.]!!=3068725D0168C0
683	![SUB]=301824	684	[SUPER]=3016271C
685	#[SUR]#=311C	686	[SUR]#=726C169C
687	#[SU]#=2F2C16	688	#[SSU]#=3216
689	.[S]!=2F	690	CE[S]!=4A2F
691	SE[S]!=4A2F	692	#:.E[S]!=2F
693	#^:##[S]!=2F	694	#^:#[S]!=30
695	U[S]!=30	696	!:#[S]!=2F
697	[SY]^=3047	698	#[S].=2F
699	[SSH]=3032	700	~~[SS]=30~~
701	[S]=30		

8.0.20 "T" Rules

Rule No.	LTPR	Rule No.	LTPR
702	![T]!=2801	703	[T]!=68C0
704	![TAXI]=280C293041	705	![TEN]!=280A38
706	[THAT'S]=350C2830	707	![TIR]=280E441C
708	![TO]!=281663	709	![TON]!=281838
710	![TOWARD]=2863515C25	711	![TWO]=2816
712	[TA]STE=280804	713	[T]CH=
714	#:[TED]!=280725	715	![TH]AN!=35
716	[TH]AT!=35	717	![THE]!#=3501
718	![THE]!=351A	719	[THEI]R=350A
720	![TH]EM!=35	721	![TH]EN=35
722	![THERA]=360A5D8E	723	![THERE]=350A1D
724	[TH]ERLY=35	725	[THESE]!=3541412F
726	![THEY]=35484884	727	![THIS]!=350730
728	[THOSE]=35112F	729	[THOUGH]!=351163
730	[THS]!=356F	731	![TH]US=35
732	[TH]Y=35	733	[TH]=36
734	S[TI]#N=2832	735	[TI]AL=32
736	[TI]A=3201	737	[TIE]N=321A
738	[TI]O^=72	739	[TI]O=723C
740	[TI]V=280A	741	[TOU]CH=2818
742	[TOUR]=28161C	743	![TRIU]=281D0E0458
744	[TR]=68725D	745	[TT]=28
746	[TU]A=283216	747	![TUES]DAY=28162F
748	[TUR]#=68725C	749	[TT]=28
750	![TWICE]!=28630E0430	751	[TZ]=2830
752	[T]=68		

8.0.21 "U" Rules

Rule No.	LTPR	Rule No.	LTPR
753	![U]!=4416	754	[UH]=98
755	[UI]L.=0747	756	[UI]L^=07
757	R[UI]T=5656	758	[UI]TE=6301
759	[UI]^E=5656	760	S[UI]T=5656
761	[U]L.=5858	762	![ULTRA]=1820281D58
763	[U]LY=16	764	[U]M.=5858

765	[U]NION=4414	766	[U]NITE=445656
767	![UNIN]=18380738	768	![U]NI=4416
769	[U]N.=5858	770	![UN]=1838
771	![UPON]=1A27505038	772	[UR]#=44169C
773	[UR].=5C5C	774	[UR]^=1C
775	[U].!=5858	776	[U]^!=18
777	R[U]^L#=16	778	[U]^L#=16
779	R[U]^R#=16	780	[U]^R#=16
781	[U]~=18	782	~[U]+=16
783	[UY]=4E4E04	784	!G[U]#=
785	G[U]#=23	786	L[U].#=5656
787	L[U]^#=16	788	#N[U]=0416
789	R[U]#=16	790	R[U].#=5656
791	R[U]^#=16	792	![U]=845656
793	[U]=8416		

8.0.22 "V" Rules

Rule No.	LTPR	Rule No.	LTPR
794	![V]!=3301	795	![VAL]=330E20
796	![VEGE]=330A314A	797	![VIB]=330E0424
798	[VIEW]=33845656	799	![VIND]=33073825
800	![VIO]=330E0418	801	[VI]^E!=334E4E84
802	[VOW]=331016	803	[V]=33EC

8.0.23 "W" Rules

Rule No.	LTPR	Rule No.	LTPR
804	![W]!=251B24200416	805	![WERE]=231C
806	[WA]NT=230E	807	[WARE]=23085D
808	[WAR]=23115C	809	[WA]STE=230804
810	[WA]S=2319	811	[WA]T=230F
812	![WEDNES]DAY=634A4A382F	813	S[W]ER!=
814	[WHA]T=6319	815	![WHE]N=630A
816	[WHERE]=230A1D	817	![WHICH]=63072832
818	[WHOL]=2D1220	819	[WHO]=2316
820	[WH]=23	821	![WIZ]=23072F
822	![WOMA]N=23583747	823	![WOMEN]=2347374A38
824	[WON]=235878	825	[WOR].=A31C5C
826	[WOR]^=A35C5C	827	[WOW]!=231016
828	[WR]=1D	829	[W]=63

8.0.24 "X" Rules

Rule No.	LTPR	Rule No.	LTPR
830	![X]!=0B2930	831	![X]=2F
832	[XIOUS]=321830	833	[X]U=2932
834	[X]X=	835	[X]^=2930
836	[X]=2930		

8.0.25 "Y" Rules

Rule No.	LTPR	Rule No.	LTPR
837	![Y]!=234E4E84	838	![YES]=040A30
839	[YOU]NG=0458	840	![YOUR]=04115C
841	![YOU]=0416	842	[Y]PHE=0E84
843	![Y]=04	844	!B[Y]=0E04
845	!C[Y]N=07	846	!C[Y]=0E04
847	F[Y]!=0E84	848	PL[Y]=0E84
849	^[YR]^=5C5C	850	!:[Y]!=4E4E84
851	!:[Y]#=4E4E84	852	!:[Y]^E=0E84
853	!:[Y]^+:=07	854	^[Y]:#=07
855	[Y]^L#=4E4E84	856	[Y]^R#=4E4E84
857	!:[Y]^#=4E4E84	858	!^[Y]:=07
859	[Y]=01		

8.0.26 "Z" Rules

Rule No.	LTPR	Rule No.	LTPR
860	![Z]!=2F0101	861	[ZZ]=2F
862	[Z]=2F		

8.0.27 Punctuation Rules

Rule No.	LTPR	Rule No.	LTPR
863	[ ]=80	864	[?]=0000
865	[...]=0A783011630E38	866	I[.]I=276D5151017868C0
867	[.]=0000	868	[,]=00
869	[:]=40	870	[;]=0080
871	[!]=0000	872	!["]=292311A368
873	["]!=1838292311A368	874	["]=
875	^['S]=30	876	:[']=

8.0.28 Symbol Rules

Rule No.	LTPR	Rule No.	LTPR
877	[]=8C8C28	878	[#]=78585837645C
879	[$]=250E201C	880	[%]=276D5C304A4A7868C0
881	[&]=0A7865C0	882	[+]=27201830
883	[-]=378E4E84781830	884	[I]=115C
885	[<]=604A30354C0C78	886	[>]=E9E65C4884685C354C0C78
887	[=]=0229230F602F	888	[/]=
889	[\]=640C29ADC030600C32	890	[ ]=780E68C0
891	[/]=	892	[^]=
893	I[^]=304A4A786830	894	[•]=
895	[[]=	896	[]]=
897	[(]=	898	[)]=
899	[{]=	900	[}]=

8.0.29 Numeric Rules

Rule No. LTPR

901 |[0]=
902 [0]=2F021C11
903 [1]=230E38
904 [1ST]=342C5C5C3068C0
905 [2]=281663
906 [2ND]=304A295A7865C0
907 [3]=361C01
908 [3RD]=361C5C65C0
909 [4]=34111C
910 [4TH]=342C115C36
911 [5]=340F0133
912 [5TH]=342C4774AC36
913 [6]=30072930
914 [7]=300B330B38
915 [8]=088428
916 [9]=380F0338
917 [10]=280A38
918 [11]=0A204A33EC7878
919 [12]=68630A4A6033EC
920 [13]=365C5C68014178
921 [14]=342C1168014178
922 [15]=342C4774AC68014178
923 [16]=3047293068014178
924 [17]=304A4A33EC4A4A7868014178
925 [18]=080468014178
926 [19]=7807784A68014178
927 [2]|=68634A4A786801
928 [3]|=365C5C6801
929 [4]|=342C115C6801
930 [5]|=342C4774AC6801
931 [6]|=304729306801
932 [7]|=304A4A33EC4A4A786801
933 [8]|=08046801
934 [9]|=7807784A6801

9. Appendix C

9.0.1 Process Listing Debug Data

The following is a sample of the type of process listing generated by Talker. The session listed here corresponds to the input of the following two sentences:

1. "The subject of this thesis is the design and implementation of a system capable of delivering synthetic speech.".
2. "Please get off my cord, thank you.".

```
The sentence is: The subject of this thesis is the design and
                 implementation of a system capable of delivering
                 synthetic speech.

..........................................
Orthographic original      Translation Rule
 T .... 718 .... ![THE]!=351A

..........................................
Orthographic original      Translation Rule
   .... 863 .... [ ]=80

..........................................
Orthographic original      Translation Rule
 S .... 683 .... ![SUB]=301824
 J .... 423 .... [J]=B1
 E .... 269 .... [E]=4A
 C .... 147 .... [C]=29
 T .... 703 .... [T]!=68C0

..........................................
Orthographic original      Translation Rule
   .... 863 .... [ ]=80

..........................................
Orthographic original      Translation Rule
 O .... 505 .... [OF]!=1A33C0

..........................................
Orthographic original      Translation Rule
   .... 863 .... [ ]=80

..........................................
Orthographic original      Translation Rule
 T .... 727 .... ![THIS]!=350730

..........................................
Orthographic original      Translation Rule
   .... 863 .... [ ]=80

..........................................
Orthographic original      Translation Rule
 T .... 733 .... [TH]=36
 E .... 269 .... [E]=4A
 S .... 701 .... [S]=30
```

I .... 420 .... [I]=47
S .... 694 .... #^:#[S]!=30

....
Orthographic original   Translation Rule
.... 863 .... [ ]=80

....
Orthographic original   Translation Rule
I .... 420 .... [I]=47
S .... 696 .... !:#[S]!=2F

....
Orthographic original   Translation Rule
.... 863 .... [ ]=80

....
Orthographic original   Translation Rule
T .... 718 .... ![THE]!=351A

....
Orthographic original   Translation Rule
.... 863 .... [ ]=80

....
Orthographic original   Translation Rule
D .... 155 .... ![DE]^#=2507
S .... 701 .... [S]=30
I .... 368 .... [IGN]!=4E4E8438

....
Orthographic original   Translation Rule
.... 863 .... [ ]=80

....
Orthographic original   Translation Rule
A .... 24 .... ![A]ND=0A
N .... 499 .... [N]=78
D .... 176 .... [D]!=65C0

....
Orthographic original   Translation Rule
.... 863 .... [ ]=80

....
Orthographic original   Translation Rule
I .... 377 .... ![I]M=07
M .... 484 .... [M]=37
P .... 624 .... [P]=27
L .... 450 .... #^:[L]=1A20
E .... 181 .... [E]MENT=
M .... 484 .... [M]=37
E .... 218 .... #:EM[E]NT=0A
N .... 499 .... [N]=78
T .... 752 .... [T]=68
A .... 81 .... [A]^+#=0804
T .... 738 .... [TI]O^=72
O .... 529 .... #:[O]N!=5A

N .... 499 .... [N]=78

....

Orthographic original Translation Rule

.... 863 .... [ ]=80

....

Orthographic original Translation Rule

O .... 505 .... [OF]!=1A33C0

....

Orthographic original Translation Rule

.... 863 .... [ ]=80

....

Orthographic original Translation Rule

A .... 1 .... ![A]!=0804

....

Orthographic original Translation Rule

.... 863 .... [ ]=80

....

Orthographic original Translation Rule

S .... 697 .... [SY]^=3047
S .... 701 .... [S]=30
T .... 752 .... [T]=68
E .... 268 .... [E].=4A4A
M .... 484 .... [M]=37
C .... 147 .... [C]=29
A .... 83 .... [A]^#=0804
P .... 624 .... [P]=27
A .... 70 .... [A]^LE=1B
B .... 110 .... [B]=64
L .... 450 .... #^:[L]=1A20
E .... 180 .... #:[E]!=

....

Orthographic original Translation Rule

.... 863 .... [ ]=80

....

Orthographic original Translation Rule

O .... 505 .... [OF]!=1A33C0

....

Orthographic original Translation Rule

.... 863 .... [ ]=80

....

Orthographic original Translation Rule

D .... 154 .... ![DELI]=250A2007
V .... 803 .... [V]=33EC
E .... 227 .... [ERI]=0A1D07
N .... 488 .... #[NG]=39

....

Orthographic original Translation Rule

.... 863 .... [ ]=80

....

Orthographic original	Translation Rule
S 697	[SY]' =3047
N 499	[N]=78
T 733	[TH]=36
E 269	[E]=4A
T 703	[T]!=68C0
I 420	[I]=47
C 147	[C]=29

....

Orthographic original	Translation Rule
.... 863	[ ]=80

....

Orthographic original	Translation Rule
S 701	[S]=30
P 624	[P]=27
E 207	[EE]=01
C 123	[CH]!=283280

....

Orthographic original	Translation Rule
. .... 867	[.]=0000

....

This is the phonetic translation.

Phon	Pitch	Duration	Speech Rate	Amplitude
THV1	9	98	1.2e+00	12
UH21	8	98	1.2e+00	12
PA1*	9	49	1.2e+00	12
S1**	9	98	1.2e+00	12
UH1*	8	98	1.2e+00	12
B1**	7	98	1.2e+00	12
J1**	6	49	1.2e+00	12
EH1*	5	74	1.2e+00	12
K1**	4	98	1.2e+00	12
T1**	3	74	1.2e+00	12
PA1*	2	25	1.2e+00	12
PA1*	9	49	1.2e+00	12
UH21	9	98	1.2e+00	12
V1**	8	98	1.2e+00	12
PA1*	7	25	1.2e+00	12
PA1*	9	49	1.2e+00	12
THV1	9	98	1.2e+00	12
I1**	8	98	1.2e+00	12
S1**	7	98	1.2e+00	12
PA1*	9	49	1.2e+00	12
TH1*	9	98	1.2e+00	12
EH1*	8	74	1.2e+00	12
S1**	7	98	1.2e+00	12
I1**	6	74	1.2e+00	12
S1**	5	98	1.2e+00	12
PA1*	9	49	1.2e+00	12
I1**	9	74	1.2e+00	12

Z1**	8	98	1.2e+00	12
PA1*	9	49	1.2e+00	12
THV1	9	98	1.2e+00	12
UH21	8	98	1.2e+00	12
PA1*	9	49	1.2e+00	12
D1**	9	98	1.2e+00	12
I1**	8	98	1.2e+00	12
S1**	7	98	1.2e+00	12
AH1*	6	74	1.2e+00	12
AH1*	5	74	1.2e+00	12
Y11*	4	49	1.2e+00	12
N1**	3	98	1.2e+00	12
PA1*	9	49	1.2e+00	12
EH1*	9	98	1.2e+00	12
N1**	8	74	1.2e+00	12
D1**	7	74	1.2e+00	12
PA1*	6	25	1.2e+00	12
PA1*	9	49	1.2e+00	12
I1**	9	98	1.2e+00	12
M1**	8	98	1.2e+00	12
P1**	7	98	1.2e+00	12
UH21	6	98	1.2e+00	12
L1**	5	98	1.2e+00	12
M1**	4	98	1.2e+00	12
EH1*	3	98	1.2e+00	12
N1**	2	74	1.2e+00	12
T1**	1	74	1.2e+00	12
A1**	0	98	1.2e+00	12
Y11*	1	98	1.2e+00	12
SCH1	2	74	1.2e+00	12
UH21	3	74	1.2e+00	12
N1**	4	74	1.2e+00	12
PA1*	9	49	1.2e+00	12
UH21	9	98	1.2e+00	12
V1**	8	98	1.2e+00	12
PA1*	7	25	1.2e+00	12
PA1*	9	49	1.2e+00	12
A1**	9	98	1.2e+00	12
Y11*	8	98	1.2e+00	12
PA1*	9	49	1.2e+00	12
S1**	9	98	1.2e+00	12
I1**	8	74	1.2e+00	12
S1**	7	98	1.2e+00	12
T1**	6	74	1.2e+00	12
EH1*	5	74	1.2e+00	12
EH1*	4	74	1.2e+00	12
M1**	3	98	1.2e+00	12
K1**	2	98	1.2e+00	12
A1**	1	98	1.2e+00	12
Y11*	0	98	1.2e+00	12
P1**	1	98	1.2e+00	12
UH31	2	98	1.2e+00	12
B1**	3	74	1.2e+00	12
UH21	4	98	1.2e+00	12
L1**	5	98	1.2e+00	12
PA1*	9	49	1.2e+00	12
UH21	9	98	1.2e+00	12
V1**	8	98	1.2e+00	12

PA1•	7	25	1.2e+00	12
PA1•	9	49	1.2e+00	12
D1••	9	98	1.2e+00	12
EH1•	8	98	1.2e+00	12
L1••	7	98	1.2e+00	12
I1••	6	98	1.2e+00	12
V1••	5	98	1.2e+00	12
HF1•	4	25	1.2e+00	12
EH1•	3	98	1.2e+00	12
R1••	2	98	1.2e+00	12
I1••	1	98	1.2e+00	12
NG1•	0	98	1.2e+00	12
PA1•	9	49	1.2e+00	12
S1••	9	98	1.2e+00	12
I1••	8	74	1.2e+00	12
N1••	7	74	1.2e+00	12
TH1•	6	98	1.2e+00	12
EH1•	5	74	1.2e+00	12
T1••	4	74	1.2e+00	12
PA1•	3	25	1.2e+00	12
I1••	9	74	1.2e+00	12
K1••	8	98	1.2e+00	12
PA1•	9	49	1.2e+00	12
S1••	9	98	1.2e+00	12
P1••	8	98	1.2e+00	12
E1••	7	98	1.2e+00	12
T1••	6	98	1.2e+00	12
SCH1	5	98	1.2e+00	12
PA1•	4	49	1.2e+00	12
PA1•	9	98	1.2e+00	12
PA1•	8	98	1.2e+00	12

The sentence is: Please get off my cord, thank you.

```
...........................................
Orthographic original     Translation Rule
  P  ....  623 .... ![P]=276D
  L  ....  451 .... [L]=60
  E  ....  196 .... [EA]:E!=01
  S  ....  701 .... [S]=30
  E  ....  180 .... #:[E]!=

...........................................
Orthographic original     Translation Rule
     ....  863 .... [ ]=80

...........................................
Orthographic original     Translation Rule
  G  ....  289 .... [GE]T=E6C00A
  T  ....  703 .... [T]!=68C0

...........................................
Orthographic original     Translation Rule
     ....  863 .... [ ]=80

...........................................
Orthographic original     Translation Rule
  O  ....  507 .... [O]F^=10
  F  ....  279 .... [FF]=74AC

...........................................
Orthographic original     Translation Rule
     ....  863 .... [ ]=80

...........................................
Orthographic original     Translation Rule
  M  ....  484 .... [M]=37
  Y  ....  850 .... !:[Y]!=4E4E84

...........................................
Orthographic original     Translation Rule
     ....  863 .... [ ]=80

...........................................
Orthographic original     Translation Rule
  C  ....  145 .... ![C]=292[illegible]
  O  ....  543 .... [OR].=51[illegible]15C
  D  ....  176 .... [D]!=65C0

...........................................
Orthographic original     Translation Rule
  ,  ....  868 .... [,]=00
     ....  863 .... [ ]=80

...........................................
Orthographic original     Translation Rule
  T  ....  733 .... [TH]=36
  A  ....   84 .... [A].=4C0C
  N  ....  499 .... [N]=78
  K  ....  430 .... [K]!=29ADC0
```

....
Orthographic original Translation Rule
.... 863 .... [ ]=80

....
Orthographic original Translation Rule
Y .... 841 .... ![YOU]=0416

....
Orthographic original Translation Rule
. .... 867 .... [.]=0000

....
This is the phonetic translation.

Phon	Pitch	Duration	Speech Rate	Amplitude
P1**	9	98	1.2e+00	12
HFC1	8	74	1.2e+00	12
L1**	7	74	1.2e+00	12
E1**	6	98	1.2e+00	12
S1**	5	98	1.2e+00	12
PA1*	9	49	1.2e+00	12
KV1*	9	25	1.2e+00	12
PA1*	8	25	1.2e+00	12
EH1*	7	98	1.2e+00	12
T1**	6	74	1.2e+00	12
PA1*	5	25	1.2e+00	12
PA1*	9	49	1.2e+00	12
AW1*	9	98	1.2e+00	12
F1**	8	74	1.2e+00	12
HF1*	7	49	1.2e+00	12
PA1*	9	49	1.2e+00	12
M1**	9	98	1.2e+00	12
AH1*	8	74	1.2e+00	12
AH1*	7	74	1.2e+00	12
Y11*	6	49	1.2e+00	12
PA1*	9	49	1.2e+00	12
K1**	9	98	1.2e+00	12
HFC1	8	98	1.2e+00	12
O1**	7	74	1.2e+00	12
O1**	6	74	1.2e+00	12
ER1*	5	74	1.2e+00	12
D1**	4	74	1.2e+00	12
PA1*	3	25	1.2e+00	12
PA1*	9	98	1.2e+00	12
PA1*	8	49	1.2e+00	12
TH1*	9	98	1.2e+00	12
AE1*	8	74	1.2e+00	12
AE1*	7	98	1.2e+00	12
N1**	6	74	1.2e+00	12
K1**	5	98	1.2e+00	12
HFC1	4	49	1.2e+00	12
PA1*	3	25	1.2e+00	12
PA1*	9	49	1.2e+00	12
Y11*	9	98	1.2e+00	12
U1**	8	98	1.2e+00	12
PA1*	9	98	1.2e+00	12
PA1*	8	98	1.2e+00	12

9.0.2 Rule Usage Debug Data

This section lists all of the rules used in the synthesis session involving the two sentences:

1. "The subject of this thesis is the design and implementation of a system capable of delivering synthetic speech.".
2. "Please get off my cord, thank you.".

LTPR Rule Usage Debug Data

Use Count	Rule Number	Letter-to-Phoneme-Rule
1	1	![A]!=0804
1	24	![A]ND=0A
1	70	[A]^LE=1B
1	81	[A]^+#=0804
1	83	:[A]^#=0804
1	84	[A].=4C0C
1	110	[B]=64
1	123	[CH]!=283280
1	145	![C]=292D
3	147	[C]=29
1	154	![DELI]=250A2007
1	155	![DE]^#=2507
2	176	[D]!=65C0
2	180	#:[E]!=
1	181	[E]MENT=
1	196	[EA]:E!=01
1	207	[EE]=01
1	218	#:EM[E]NT=0A
1	227	[ERI]=0A1D07
1	268	[E].=4A4A
3	269	[E]=4A
1	279	[FF]=74AC
1	289	[GE]T=E6C00A
1	368	[IGN]!=4E4E8438
1	377	![I]M=07
3	420	[I]=47
1	423	[J]=B1
1	430	[K]!=29ADC0
2	450	#^:[L]=1A20
1	451	[L]=60
4	484	[M]=37
1	488	#[NG]=39
5	499	[N]=78
3	505	[OF]!=1A33C0
1	507	[O]F^=10
1	529	#:[O]N!=5A
1	543	[OR].=51515C
1	623	![P]=276D
3	624	[P]=27
1	683	![SUB]=301824

1	694	#^:#[S]!=30
1	696	!:#[S]!=2F
2	697	[SY]^=3047
5	701	[S]=30
3	703	[T]!=68C0
2	718	![THE]!=351A
1	727	![THIS]!=350730
3	733	[TH]=36
1	738	[TI]O^=72
2	752	[T]=68
1	803	[V]=33EC
1	841	![YOU]=0416
1	850	!:[Y]!=4E4E84
22	863	[ ]=80
2	867	[.]=0000
1	868	[,]=00

Table C.1