MAS.632 Conversational Computer Systems
Fall 2008

5

Speech Synthesis

The previous two chapters discussed techniques for digital encoding of speech and how different classes of applications utilize this recorded speech. In some instances, such as a voice mail system, the computer treats recording as opaque data; it captures what one person says and relays it to another without knowledge of its lexical content. In other applications, such as telephone-based access to flight schedules or announcements of the approaching station on a subway, the application knows the content of prerecorded snippets of speech and pieces them together to generate spoken output. But although the computer "talks" in such applications, its repertoire is limited to a few prerecorded phrases. Speech synthesis is the process of generating speech from text; for any string of words a speech synthesizer can approximate the way a human would read these same words.

Although synthesized speech cannot be expected to achieve the quality of the human speech that it mimics, the flexibility of synthesis makes it extremely useful for a variety of applications. Synthesis allows voice output in discourse systems in which the computer has a wide variety of things to say, and it can easily be driven by language generating programs that produce text. Synthesis is helpful for voice access to databases because databases may have a very large number of entries to pronounce (such as names of people, streets, or businesses). Finally for some applications the computer may need to speak human-authored text such as an electronic mail message or a text file.

Transforming text into speech is done by a process called **text-to-speech synthesis** or **synthesis-by-rule.** These terms are chosen to contrast some speech coding methods such as Linear Predictive Coding (see Chapter 3), which

are sometimes called **analysis-synthesis** algorithms. Such coders first analyze the speech in terms of parameters to control an underlying vocal tract model, and then "resynthesize" it for playback through a digital system that implements this model. Speech coders can only store spoken speech; speech synthesizers accept arbitrary text as input and generate speech. Throughout this book, the term "speech synthesis" is used to refer exclusively to text-to-speech generation.

To begin to appreciate the difficulty of synthesizing speech, let us consider an apparently simple approach to synthesis that is actually impractical although quite suitable as a substitute to synthesis in certain limited situations. Suppose we simply record a person speaking every possible word and then string these audio segments together for speech output? This approach is impractical for several reasons.

First, there are simply too many words. Not only do we need to digitize the root forms of nouns and verbs, we also need their forms in combinations. For "play," for example, we might want "plays," "playful," "playing," "played," etc. or compounds such as "horseplay."[1] Even if we somehow coped with storing all these word forms, it would be even less practical to store all the proper nouns that might be required to speak from some databases [Spiegel 1985]. Finally, languages change and new words and acronyms ("DRAM," "ISDN," "downsize") keep appearing in English or any language, necessitating recurring updates of the recorded speech.

Second, even if we *could* store every word which might be spoken, they would sound very awkward when strung together. Words change pronunciation in a spoken context. Sentences have a melody and rhythm that changes the pitch and duration of syllables. Phonemes at the beginnings and endings of words are spoken with variations that harmonize with surrounding words. A word spoken in isolation is said to be in **citation form,** but this is not how we speak in normal conversation. Consider the word "the"; when asked to say this word by itself, many people instinctively pronounce it as "thee," but we actually speak it this way only when the following word begins with a vowel. Many words exhibit this phenomenon. "To" is usually spoken more like "ta," and the vowel in "for" practically disappears in fluent speech.

Although these observations are meant to argue that conversion of text to speech is more difficult than patching together digitized words, in some situations it may suffice to record a number of short phrases and concatenate them; this technique is used in many telephone-based applications. For example, the caller speaks with a human directory assistance operator, but then a computer voice recites the number. What actually transpires is that a number of short recorded segments are played starting with "The number is . . ." followed by a separate recording of each digit.

Although the phone number is spoken by concatenating recordings of each digit, some sophistication is required to make the number sound natural. All dig-

[1]Note that we know not to pronounce the "e" in such a compound word.

its in a telephone number are not pronounced the same; intonation is used to group them. North American numbers consist of three digits (area code), three digits (exchange), and four more digits. Each group is pronounced separately, with a falling pitch indicating phrasing (see Figure 5.1). The last digit in each of the first two groups has a rising pitch, which serves as a continuation marker or cue that more information will follow the pause. For each digit, three (or sometimes four) recordings are made, and the appropriate one is chosen depending on whether the digit is in the initial, medial, or terminal position in the number.

SYNTHESIZING SPEECH FROM TEXT

How can text be transformed into speech? As just discussed, it is impractical to simply record each word because of pronunciation problems as well as the amount of storage required for the lexicon. We cannot go directly from text to sound by pasting words together; instead, some smaller unit of representation is required for sound generation. For efficient synthesis, this unit should be significant either from the perspective of the written language (words or letters) or from the spoken language (syllables or phonemes).

English is a difficult language to synthesize largely because its orthography (the written version of the language) is highly irregular; more specifically, the mapping from letters to sounds is not one to one. The same sound may be produced from various letters (e.g., the initial phoneme in "kid" and "cat"). Worse still, the same letters can be pronounced in different ways (e.g., the final four letters in "tough" or "through").

In some languages, the stronger relationship among letters and sounds would suggest that letters be used as a unit of synthesis. An example is the Turkish language; the Turks replaced their former Arabic script with a Roman alphabet earlier this century as part of the "modernization" of the country. The new alphabet was created specifically for the language from symbols used in other languages,

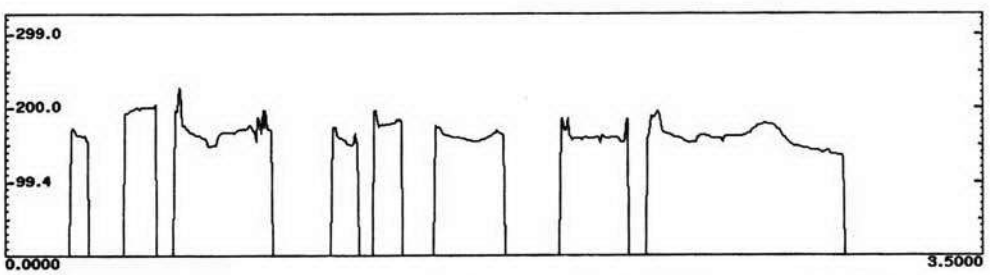

Figure 5.1. Pitch track of a telephone number including area code. Note the phrasing of the three groups of numbers. The first two number groups show a small terminal rise, which is a cue to the listener that there is more to come. There is no such rise in the final number group. The vertical axis is pitch in Hertz, the horizontal axis shows time in seconds.

and it is conveniently phonetic. Equally important, spoken languages continually change but only slowly. So spoken Turkish has not yet diverged from its recently specified written form. In the new Turkish alphabet, not all the letter symbols are recognizable in any single European language; for example, both "o" and "u" exist in an umlaut and nonumlaut form as in the German "u" and "ü," but "i" also has a dotted and undotted form ("ı" and "i"), which German does not. However, once the single pronunciation of each letter is learned, it is easy for someone who does not speak Turkish to read it aloud. This is exactly the task of a speech synthesizer.

Since English is not this simple, it is necessary to first convert the letters to a less ambiguous representation and then apply sound realization rules to that representation. The most common alternative form to the letter is the phoneme as there are a fairly small number of phonemes in any language, simplifying the sound generation rules.[2] Synthesis is composed of two steps; the first converts the text to a string of phonemes (with intonational markers), and the second realizes the output of the first as a speech waveform. This is depicted in Figure 5.2. Note that in the Turkish example, since each letter maps to exactly one phoneme, we can simply skip the first step in this model except that intonational markers must still be added to each syllable.

FROM TEXT TO PHONEMES

So how can text, an ordered series of letters, be converted to the equivalent string of phonemes? Two general approaches may be identified: a **pronunciation lexicon** (dictionary) or a set of **rules** similar to what we learn in elementary school. Both are useful and a hybrid approach is optimal.

The dictionary approach is simple. A string of phonemes is stored for each word, and conversion of text to phonemes is accomplished by lookup. Several problems prevent a lexicon-only solution, however. First, the dictionary grows rapidly as a result of the sizable vocabularies required for many applications. Second, the dictionary must be stored in nonvolatile memory in a stand-alone synthesizer. Third, at some point in the dictionary lookup a "morphological decomposition" analysis of the text must occur, if only to identify simple forms such as plurals and past tense; if both "cat" and "cats" are to exist as distinct

[2]The number of phonemes in a language varies from a low of 13 (Hawaiian) to a high of about 75. English is in the middle with approximately 40 phonemes.

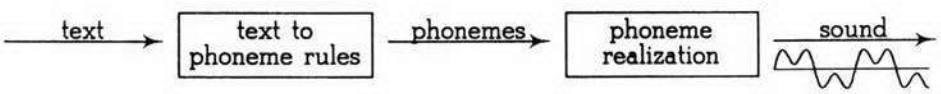

Figure 5.2. A basic model of speech synthesis.

entries, the dictionary must be much larger. Fourth, sometimes correct pronunciation cannot be gained from looking at a word in isolation, but rather, sentence context must be employed ("read" has two pronunciations depending on tense; some other examples are discussed below on the topic of lexical stress). Finally, no matter how large the dictionary is, there will always be words it does not contain such as proper nouns, jargon, and new words.

The rule-based approach, in contrast, uses knowledge of spelling rules to derive pronunciation from the text. There are several advantages to such techniques. First, the set of rules is much more concise than a lexicon of pronunciations for each word. Second, it is more flexible; effective rules can generate a plausible pronunciation for almost any word including names. Third, rules represent succinct knowledge about a language. Rules may be portable across languages, and the rules for a language can generate pronunciations for names from that nationality even while synthesizing a different language.

Although a completely rule-based synthesizer is attractive, this is quite difficult and not the easiest path to the goal of accurate synthesis of commonly used words. Since a dictionary may more easily embody correct pronunciation than some of the more general spelling rules, the usual approach is to store common words in the lexicon and apply rules to the remaining words. It is best to put in the lexicon only those common words that are known *not* to be pronounced well by the rules. If lookup fails, a rule is invoked which is either correct (or else the word would be in the lexicon) or at least a reasonable guess for an unlisted and probably unusual word (which would not be in the lexicon under any circumstances).

Even for a completely rule-based synthesizer, it is quite useful to include a user-definable phonetic exceptions dictionary to deal with important words that are not pronounced correctly such as names or acronyms common in a particular organization. Although these words may be uncommon in the language as a whole, local use may be frequent; if they are mispronounced, the result will be greater listener dissatisfaction.

The process of text-to-phoneme translation is performed in several steps. The first step is text preprocessing or **normalization** to convert symbols and abbreviations into their full-text spellings such as converting the symbol "$" to "dollar." Normalization is not quite as simple as it may first appear; the best full-text spelling of "$4.35" is "four dollars and thirty-five cents," but "$2.00" is simply "two dollars." Similarly, some abbreviations, such as "Dr." and "St.," can represent one of several possible words. "Dr." is usually "doctor" when the word following is a proper noun ("Dr. Jones"); it is pronounced "drive" when a proper noun precedes it ("Jones Dr.").

The next step is **morphological analysis,** which copes with words composed of several root parts (morphemes) including plurals and compound words such as "baseball" and "backpack." Many words in English are built by combining root forms with **affixes.** An affix may be a prefix such as "un" in "unusual," or a suffix such as the "ing" in "buying." A word may have multiple affixes at either end, such as "usability," from "use" + "able" + "y." This last example also shows how spelling may change as affixes are added; these regular spelling changes must be included in the morphological decomposition.

Since words are built from morphemes, there will be far fewer morphemes than words, making storage in a lexicon more practical. Text is broken down into morphological units, pronunciation is looked up for each, and rules are applied to those units for which lookup fails. This process of producing phonemes from text is summarized in Figure 5.3.

Additional Factors for Pronunciation

Unfortunately, simply converting text to a stream of phonemes is inadequate for the next stage of synthesis: generating sound. Several confounding factors complicate the model shown in Figure 5.2; additional information, such as **lexical stress, coarticulation,** and **intonation** must accompany the phonemes. Lexical stress is the pattern of syllabic emphasis within a word. Coarticulation is the change in pronunciation of phonemes as a function of their phonemic environment, i.e., the sounds preceding and succeeding them. And overall sentence intonation, or **prosody,** requires adjustments to the output of the text-to-phoneme processing.

Lexical Stress

In English, as in many languages, not all syllables are created equal. We learned in elementary school that every word has one syllable carrying primary stress and possibly others carrying secondary stress. How do we generate stress acoustically? Although the intuitive answer is the volume (amplitude) of the speech, this is only a secondary factor. As Fry demonstrated in some key experiments [Fry 1958], stress is carried primarily as variations in pitch and duration of the appropriate syllables. A stressed syllable is somewhat longer than an unstressed syllable and will also carry higher (or lower) pitch than it would otherwise as shown in Figure 5.4.

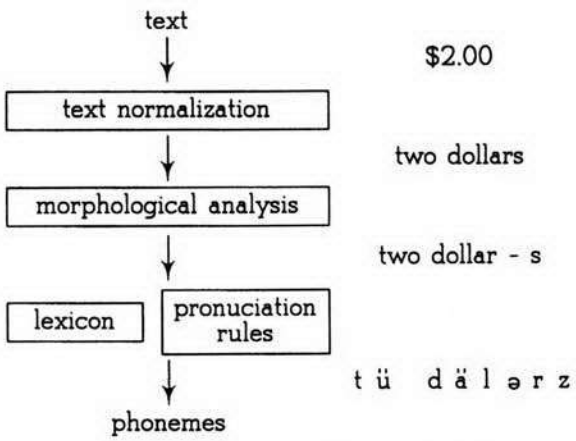

Figure 5.3. Steps in converting text to phonemes.

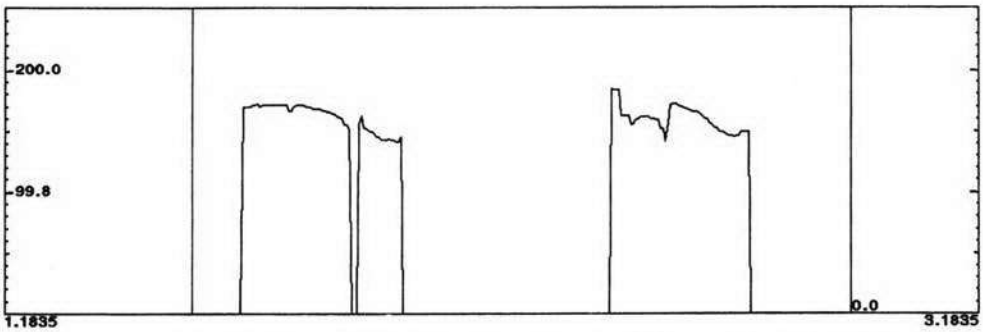

Figure 5.4. Pitch tracks showing lexical stress. The word on the left is CON-duct. The word on the right is con-DUCT. Note that the stressed syllable is longer and higher pitched.

Lexical stress is part of a complex interplay between the intonation and rhythm of a whole sentence. Discussion of lexical stress is limited to that which can be detected in a word spoken in isolation, in its citation form. Stress covers many phenomena, all realized as variations in rhythm and meter; lexical stress is what differentiates syllabic "accent" as opposed to emphasizing one word over another in a sentence. Lexical stress is important for understanding spoken language; incorrectly assigned stress may make a word unintelligible. For some English words the assignment of stress differentiates noun and verb forms, e.g., "convert" and "conflict" (see Figure 5.4). Determining which syllable to stress requires a reliable syntactic analysis of the sentence.

Stress confounds text-to-phoneme conversion because stress is not directly phonemic. In addition to producing the proper phonemes, the syllables containing those phonemes also must be marked for stress. This is done by assigning pitch and duration for each voiced phoneme during the conversion to phonemes.

Coarticulation

Coarticulation is the process whereby the pronunciation of a phoneme changes as a function of its surrounding phonemes. This change may be an allophonic variation, that is, substitution of an acoustically distinct version of the same phoneme. An example of this is the "t" sound in "but" versus "butler"; terminal stop consonants tend to be released. Alternatively, the substitution may result in an entirely different phoneme from the citation form. Consider the second "d" in "Did you . . . ," which can be pronounced much closer to a "j" without loss of intelligibility.

Coarticulation occurs because the vocal tract is a physical system. The articulators (tongue, lips, etc.) have mass and make continuous motions that are not instantaneous, due to inertia and the physical processes controlling the various muscle groups. Although for each phoneme there is a desired target configuration for the articulators (such as tongue touching the palate), in practice the articula-

tors may never fully realize that configuration. Instead, they simply make a gesture in its direction, which is sufficient to change the sound emanating from the vocal tract just enough to cue the listener to the identity of the intended phoneme. Pronunciation is sloppy. This is not a defect in speech; rather it is an accommodation to the speech production process that results in an ability to communicate faster than we could otherwise.

In some cases these coarticulation effects are localized within a word. The consonant "r" is a good example; the sound of "r" is realized largely through modification of the preceding vowel. In other cases these effects occur across word boundaries. Consider the fricatives in "the cats went" and "the cats sat"; the pause required to distinguish the plural form of cats in the latter introduces more aspiration on the first "s." Speech synthesized without considering these coarticulation effects will be stilted and perhaps less intelligible. Fortunately, many coarticulatory effects can be predicted as they follow rules [Oshika *et al.* 1975], although a complete set of such rules has not yet been compiled for English.

There is also interaction between coarticulation and lexical stress; in general, the unstressed syllables are much more subject to change. An example is **vowel reduction,** in which the vowel of an unstressed syllable is shortened and turned into a schwa.[3] In the extreme, the reduced vowel may simply vanish; consider how many syllables are pronounced in "chocolate" (two or three?). Note what happens to the "e" between "manager" and "managerial"; in "managerial" the "e" is stressed and hence not reduced.

Intonation

A final consideration, before phonemes can be realized as sounds, is the overall melody or intonation of the sentence. Intonation refers to the pitch contour of the sentence, both in terms of which words receive greater emphasis as well as the general slope (rising or falling) of the pitch contour. Intonation differentiates questions from statements even when the words themselves may be identical, and it also reveals which words in a sentence are particularly emphasized.

Consider the intonation of the sentences "She went to Paris." (simple statement of fact) and "She went to Paris?" (expression of disbelief, or request for clarification). The first sentence (see Figure 5.5) has overall falling pitch typical of simple declaratives. The second sentence (see Figure 5.6) has pitch that rises on the last word (more precisely, on the last stressed syllable) because it is a question. But a so-called "Wh-question"[4] has its stress on the "wh-" word (see Figure 5.7).

The sentences just used as examples are deceptively simple since it is easy to deduce correct intonation from the words and punctuation. Intonation is much

[3]The *schwa,* often represented as an upside-down "e" in dictionary pronunciation guides, is approximately the unstressed initial vowel in "abut" and is the "generic" vowel in English.

[4]"Wh-"questions begin with "Who," "What," "Where," "When," etc. They are in contrast to the questions that expect a yes-or-no answer.

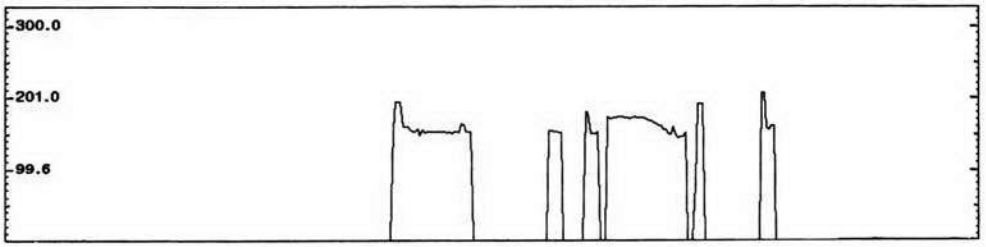

Figure 5.5. Pitch track of "*She went to Paris.*"

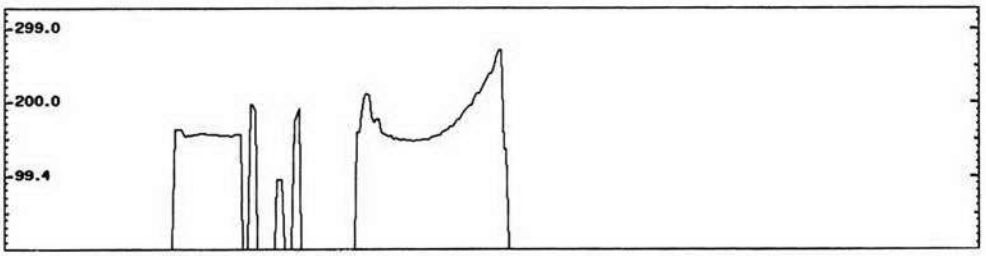

Figure 5.6. Pitch track of "*She went to Paris?*"

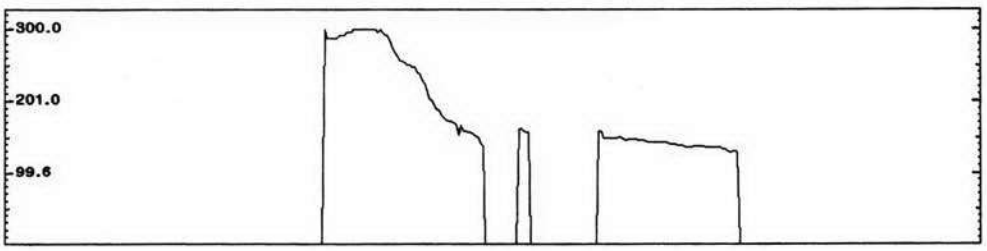

Figure 5.7. Pitch track of "*Who went to Paris?*"

harder to predict for longer sentences that are syntactically more complex; each phrase contributes its own intonation. Questions may be embedded in statements, and we cannot determine from orthography alone whether "*She went to Paris?*" is asking about the traveler (Who went to Paris?), the destination (Paris or Rome?), or verb tense (Has she gone there already?). In short, intonation is difficult.

Consequences for Synthesis

The factors just considered, namely lexical stress, coarticulation, and intonation reveal the limitations of the simplistic model of text-to-phoneme interpretation described earlier. To determine lexical stress, one must examine the underlying

morphological composition of the word. The realization of particular phonemes may change during the composition; for example, consider the "s" in "use" and "useful" as contrasted with "usability," in which the fricative is voiced. Further, identification of the stressed syllable is a prerequisite to application of coarticulation rules as unstressed syllables are more amenable to change.

Text-to-phoneme rules and lexical lookup generate a string of phonemes. Coarticulation may then substitute or remove phonemes or may call for allophonic variations; this suggests that allophones are a better intermediate representation than phonemes. Stress must then be realized as pitch and duration applied to these phonemes. Finally, sentence-level intonation adds an overall pitch contour above and beyond the syllable stress. To convey all this information, the string of phonemes or allophones must be marked with pitch and duration information.[5] Figure 5.8 summarizes the interaction of the components of more complete text-to-phoneme generation.

FROM PHONEMES TO SOUND

After text has been converted to a string of phonemes accompanied by prosodic information, the phonemes must be realized acoustically. Two approaches are used to generate the appropriate sounds to produce speech. The first method controls a digital vocal tract model by modifying internal parameters over time, while the second method pieces together small segments of digitized speech.

Parametric Synthesis

Parametric synthesis, which is also called **terminal analog** or **formant** synthesis, generates speech by varying the parameters that control a software vocal tract model; changing these parameters over time generates speech-like sounds. Vocal tract models for phoneme synthesis are similar to those outlined in the dis-

[5]This process is consistent with the discussion in Chapter 1 about how prosody does not fit cleanly into the layered model of speech communication.

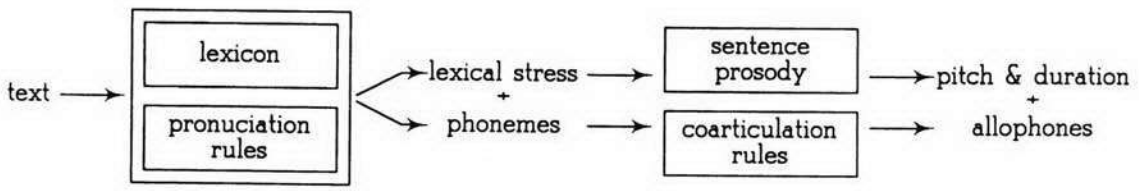

Figure 5.8. A refined diagram of text-to-phoneme reduction with output marked for pitch and duration.

cussion of Linear Predictive Coding in Chapter 3. This model includes a source and a number of filters to implement the transfer function (resonances) of the vocal tract. The source may be voiced (periodic) or unvoiced (noise); if voiced, the pitch (period) must be specified. This signal is then modified by a number of filters that produce the formants in the speech signal. Although the details of such a model are beyond the scope of this book (see the general references, especially [Klatt 1980, Klatt 1987a]), Figure 5.9 shows a partial list of the parameters used to control one of the early experimental synthesizers.

The vocal tract model produces a steady sound for a particular set of control parameters. It produces this sound by taking an input (a periodic or aperiodic pulse train) and filtering it to produce an output value; the parameters control the source and filtering functions. A new output value is produced 8000 or 10,000 times per second. The vocal tract model alone is not enough; its control parameters must be changed over time to mimic speech. The phoneme-to-sound portion of the synthesizer must model the dynamic character of phonemes by updating the parameters frequently (every 5 milliseconds is typical). The phoneme realization model takes account of how the vocal tract is configured to produce the spectra typical of the various speech sounds.

In many commercial synthesizers, the text-to-phoneme rules run in a general purpose microprocessor, and the vocal tract model runs as a continuous process on a digital signal processor. The control parameters are passed across a hardware interface between these two components. Considering the relatively small bandwidth of the control parameters, this is an effective task distribution in the hardware. However, general purpose processors have now achieved speeds where it is practical to implement the entire speech synthesis process on a workstation

```
Amplitude of voicing (dB)
Amplitude of frication (dB)
Amplitude of aspiration (dB)
Fundamental frequency of voicing (Hz)
First formant frequency (Hz)
First formant bandwidth (Hz)
First formant amplitude (dB)
Nasal zero frequency (Hz)
Nasal zero amplitude (Hz)
Nasal pole frequency (Hz)
Nasal pole amplitude (Hz)
Nasal formant amplitude (Hz)
Glottal resonator frequency (Hz)
Glottal resonator bandwidth (Hz)
```

Figure 5.9. Some of the control parameters for the synthesizer described in Klatt [1980]. These parameters control filters in the vocal tract model, e.g., the center frequencies and bandwidth of the various formants.

without any additional signal processor. This will lead soon to a new generation of all-software speech synthesis products.

Concatenative Synthesis

Consonants are generally more difficult to synthesize than vowels because of their short duration and dynamic behavior. Therefore, it should come as no surprise that the intelligibility of synthesized consonants, especially consonant clusters ("str," "sp," "bl," etc.), is lower than that of vowels [Spiegel 1988]. This problem has spurred an alternate form of phoneme realization called **concatenative synthesis** because it is accomplished by putting together short segments of recorded speech.

A concatenative synthesizer speaks by "gluing" together small units of digitized speech. What segments of speech should be used as the basic units? As stated at the beginning of this chapter, storage of each word in the language is impractical simply because there are too many of them. The next smallest unit is the syllable, but this too is impractical as there are over 10,000 syllables in English.

At the opposite extreme lies the phoneme. Any language uses a limited set of phonemes so storage would be minimal. But there are several problems with the use of phonemes as the base unit of concatenation. First, some of the voiceless consonants cannot be spoken independent of the associated vowel, rendering pure phoneme concatenation impossible. Second, it is exactly the dynamic nature of phonemes that motivates concatenative synthesis.

Since the dynamic and coarticulatory aspects of phonemes are most prevalent at their boundaries, this suggests a compromise consisting of a unit going from midway in one phoneme to midway into the next. This unit is called a **diphone** [Peterson *et al.* 1958]. Related units of speech for synthesis are the **dyad,** or vowel-consonant-vowel segments [Sivertsen 1961], and the **demi-syllable** [Fujimura and Lovins 1978], which is as the name implies half a syllable. Since there are about 40 phonemes in English, there are 1600 possible diphones as a diphone is simply a pair of phonemes. But not all these possible diphones exist in the language; for example, "sl" does occur in English, but "sr" does not. In practice, some extra diphones must be included to allow for allophonic variations and perhaps for differences between stressed and unstressed syllables.

In whatever form these units are encoded, it is important that their boundaries match closely. For example, if "bad" is to be synthesized by concatenating "ba" and "ad," the formants in the vowel portions of each must match or the discontinuity will be audible. Furthermore, it is necessary to be able to modify the pitch and duration of the stored diphones in order to support stress and prosody. LPC speech has often been used for concatenative synthesis because its pitch can be changed by modifying a single coder parameter, and segments can be lengthened by adding frames. Several successful diphone synthesizers have been built, perhaps the most notable (for English) by AT&T Bell Laboratories [Olive 1977]. At Bellcore (Bell Communications Research), Orator is a demi-syllable synthesizer that has been optimized to pronounce names [Spiegel et al. 1991, Spiegel and Macchi 1990].

QUALITY OF SYNTHETIC SPEECH

From the application developer's point of view, recorded and synthesized speech are very different because the latter allows an application to speak arbitrary text instead of a few prerecorded responses. However, from the end user's perspective, the two are functionally similar and share many problems relevant to user interface design; they will be treated jointly in the next chapter. In the user's experience there is one difference, however, that is difficult to disguise—intelligibility. Although some studies have compared synthesized speech favorably with very low bit-rate digital speech, few users are likely to encounter such degraded recorded speech except perhaps in children's toys. Although intelligibility is always a potential issue with speech output, it is especially pernicious with synthesized speech.

There are a number of possible sources of mispronunciation. The text-to-phoneme rules may be inadequate, or the morphological decomposition may break down resulting in an incorrect phoneme string. Lexical stress may be incorrectly assigned to confuse the noun and verb forms of a word or adjective and verb forms (e.g., "elaborate" and "live"). Coarticulation may be omitted or applied incorrectly, or prosody may be inappropriate. Incorrect intonation or duration rules can result in poorer listener comprehension than no prosody at all (monotone) [McPeters and Tharp 1984].

Even if the phoneme string is correct and marked with accurate pitch and duration, the phoneme-to-sound rules may not produce correct output. The consonant clusters are especially difficult to synthesize because of their dynamic aspects. Nasal consonants also pose problems because of the inability of simple vocal tract models to mimic the effect of the nasal cavity's absorption of some frequencies.[6]

Synthetic speech is sometimes likened to noisy recorded speech in terms of intelligibility. But as Nusbaum *et al.* discovered, the listeners make different mistakes when listening to synthetic speech [Nusbaum *et al.* 1984]. Different sets of phonemes are confused for synthetic or noisy human speech. Pisoni attributes the difference to the paucity of distinguishing cues in synthesized phonemes versus the rich set of cues in human speech [Pisoni *et al.* 1985].

Proper names are particularly difficult for text-to-phoneme analysis; this is due in large part to the broad ethnic diversity of names in English (at least in North America). Yet names are one of the more likely types of data found in text databases. Storing names in a lexicon is difficult simply because there are so many of them. Although a small number of common names covers a significant portion of the population, there are a tremendous number of names remaining; Spiegel [Spiegel 1985] estimates that although the top 5% of the names covers

[6]An all-pole model cannot cope with the zeros present when the nasal cavity is connected to the vocal tract.

90% of the population, there is probably a total of 1.5 million names in the United States. In addition, the names popular in one region of the country may be uncommon in another; German names are common in Milwaukee, Chinese and Hispanic names in Los Angeles, and Irish names in Boston.

How can rules be used to pronounce surnames? English pronunciation rules fail for many other languages, resulting in incorrect pronunciation of names derived from those languages. A better strategy is to identify the ethnic origin of a name based on spelling and then apply rules specific to that language. Even this may not be adequate as many foreign names have been Anglicized, and there are often several variants of the Anglicized form. Perhaps a suitable performance criterion is whether the synthesizer chooses a pronunciation of the name that may be plausible if spoken by a person. Vitale details a name pronouncing algorithm similar to that described here in [Vitale 1991].

Measuring Intelligibility

How can intelligibility be measured? There are a variety of methods, none of which reveals the entire picture (a good description plus some experimental results for all methods discussed here is found in [Pisoni *et al.* 1985]). A common test is the Modified Rhyme Test (MRT) [Fairbanks 1958], in which the listener is presented with a synthesized monosyllabic word (Consonant-Vowel-Consonant), and a set of six alternatives from which to choose the word that was spoken. Half of the stimuli vary the initial consonant (e.g., "hit," "wit," "mitt"), and the other half vary the final consonant. Thus, the MRT provides insight into the intelligibility of individual phonemes. With an "open form" of the same test, subjects are *not* given a list of choices, but instead simply indicate which word they heard. Not surprisingly, scores on the open response form are significantly lower than for the closed form. It should be noted, as pointed out by Spiegel [Spiegel 1988], that the MRT does not test intelligibility of consonant clusters, which we would expect would be particularly difficult to synthesize. More recent synthesizers score higher on the MRT than the older and less-developed ones, yet still do not approach natural speech.

However, we rarely listen to single words spoken in isolation; words are parts of a sentence or dialog, and we can often guess the next word even before it is spoken. By comparing listener transcription accuracy for semantically meaningful sentences (the Harvard psychoacoustic sentences [Egan 1948]) and nonsense sentences (e.g., the Haskins syntactic sentences [Nye and Gaitenby 1974]), we can see the contribution of meaning to intelligibility. As would be expected, semantically meaningful sentences are more intelligible than meaningless ones, but the difference is much greater for synthetic speech than for human speech [Pisoni *et al.* 1985]. This suggests that the semantic constraints on word possibilities play a stronger role in decoding the less intelligible synthetic speech.

The tests just mentioned measure the intelligibility of isolated words or single sentences based on the correctness and speed of a listener's response. For many applications, a synthesizer does not speak in isolation, but rather in the context

of a dialog or while reciting a longer passage of text. Listening comprehension tests have been used to quantify performance in such circumstances. Passages are either read by a human or synthesized, and subjects answer standard reading comprehension questions. Such tests have indicated that comprehension scores for synthesized passages may be as high or higher than for natural speech [Pisoni and Hunnicutt 1980], which suggests that general linguistic knowledge is a significant factor in comprehension.

Listener Satisfaction

Intelligibility metrics do not tell the whole story; other indications of listener satisfaction are more subjective and generally negative. Even if intelligible, synthetic speech is generally less pleasant to hear. This implies that it should be used sparingly, and in applications for which it will provide some clearly recognized added value to the user. Speech synthesis is still reserved for limited situations in which its poor sound quality is offset by the specific advantage of speaking from text.

It is important to note the difference between intelligibility and naturalness. Naturalness may be a particularly elusive goal, and, reassuringly, its achievement may not be required for many computer interaction tasks. On the other hand, unnatural speech may be tiring or boring to pay attention to. Poor prosody interferes with word intelligibility and, equally important, makes it difficult to attend to sentence breaks and the introduction of new material in a conversation. These are certainly important aspects of naturalness that impact listener satisfaction.

Performance Factors

Intelligibility and listener satisfaction cannot be summarized in single numbers. Although perhaps useful for comparison, ranking of synthesizers by intelligibility does not tell the entire story. Additional factors may contribute to the judgment on the viability of speech synthesis for a particular application.

One issue in user performance is the increased **cognitive load** associated with listening to synthetic speech. Because it is more difficult to decode the phonemes in synthetic speech, more demand is placed on the listener's short-term memory. If listening to synthetic speech takes short-term memory, this may interfere with whatever else the listener may be doing, such as keeping track of position in a menu hierarchy or remembering the information being spoken. In a series of experiments requiring subjects to memorize a string of numbers while listening to synthetic and then natural speech, Luce, Feustel, and Pisoni [Luce, Feustel and Pisoni 1983] found that subjects were able to recall the natural speech significantly better than the synthetic.

Another, possibly related, issue is **response time.** Because more processing is required to understand synthetic speech it takes longer. Measurements of subjects' response times in a task in which they were required to identify which word had been spoken showed significantly increased time for synthetic speech [Pisoni

1981]. This can also be detected using a method known as "gating," in which subjects are presented with short initial pieces of a previously recorded word (either natural or synthetic) and asked to identify it. A longer portion of a word is required for correct identification of synthetic versus natural speech.

On the positive side, there is strong evidence that understanding synthetic speech is a skill which can be learned and retained once learned. One experiment [Schwab *et al.* 1985] demonstrated that comprehension soared after a half-hour of daily exposure to synthesized speech over a 10 day period; this skill was retained 6 months later. If prospective users have sufficient reason to begin using synthetic speech, they quickly become proficient at its use.

APPLICATIONS OF SYNTHETIC SPEECH

For the reasons discussed in the previous section, synthetic speech is inferior to digitized natural speech as a computer output medium. Even when explicitly "mechanical" speech is desired,[7] digitized speech passed through various signal processing algorithms may be preferred to synthetic speech. For what purposes, then, is synthetic speech useful?

An obvious application is a reading machine for the blind. This concept started much of the original speech synthesis work in the 1970s and was the focus of several early commercial ventures. Such a reading machine relies on optical character recognition for input and synthetic speech for output. Even without a scanner, synthetic speech can be used by the blind and visually impaired to provide computer access. With their experience and increased dependence on the auditory senses, blind users learn to use synthesizers set at a very rapid speaking rate. Blind programmers are capable of writing and maintaining large programs with such an interface. Despite its importance and tremendous potential for the individuals involved, aid for the disabled is a rather small application niche.

Synthetic speech is valuable for prototyping an application using speech output, even though the final product will employ recorded speech. During interactive design, phrasings are liable to change significantly as the designer experiments not only with selecting the most appropriate words but also with how the system should articulate these words. Prototyping these systems with synthetic speech may save much time in the long run, as the speech output can be changed in the program source code with a text editor instead of recording a whole new set of prompts. Recording speech can be a difficult and time-consum-

[7]At the Atlanta airport, an automatic subway system moves passengers between terminals. When someone holds the door open, a voice announces that the train will depart the station when the doors are released. A monotone recorded voice was chosen for this recording to emphasize that no conductor was on board the train to intervene. The desired effect is that the passengers already on board the train glower at the offender and this intimidation generally frees the doors. From personal observation, the theory seems to be effective.

ing task typically requiring multiple attempts before resulting in a recording of agreeable quality.

For larger scale uses of synthetic speech, the indicators of a suitable application should be the size and complexity of the output vocabulary. The first class of appropriate applications for synthetic speech are those where the system has a relatively small repertoire of phrases it can say, but the words that make up these phrases come from a very large vocabulary. An example is an automated system to report an address for a given telephone number; the street name database could be huge. Automatic telephone directory assistance is a related problem due to the size of the name database and its diversity from city to city.

For applications such as these, the primary reason to choose synthetic over recorded speech is the logistics of building and maintaining the required database of voice recordings. Because the set of possible computer utterances is small, a number of whole phrases could be recorded with a few appropriate words substituted such as "The address is" "fifteen" "martin road." But it would be tedious to build the database, and both street and surname databases change over time. By the time the new recording is needed, the person who originally spoke the recordings may no longer be available, and switching voices in the middle of a sentence sounds confusing. With synthesized speech there is no need to record new words in the database, although a phonetic-exception dictionary may be required for those that are not pronounced correctly.

A second class of applications are those which generate natural language to describe some data in response to a query. Examples might include an electronic training manual describing how to diagnose and repair heavy machinery or a context-sensitive computer help system [Nakatani *et al.* 1986]. Such applications might be capable of generating a wide variety of utterances, and it would be difficult to exhaustively list and record them all. Synthesis of such material is sometimes referred to as **synthesis-from-concept** [Witten and Madams 1977].

The two types of applications just described are likely to be very different in terms of underlying software sophistication. The first is typified by simple queries into a very large database. The second type exhibits a much broader range of conversational ability arising in situations where the system is trying to convey in English some derived reasoning about the database. It is ironic that a sophisticated language-generation program may have a well-developed notion of syntax and the salience of particular words. This would be invaluable for synthesis of proper intonation but is lost when the utterance is reduced to a character string to be synthesized.

Finally, a third major area for use of synthesized speech is recitation of human-authored text. Since the text is created by a human, it is hardly predictable and cannot be limited to a small number of prerecorded words or phrases. One use of synthesis in this context is proofreading; it may be easier to hear errors than to see them especially for the author of a document. More widespread applications will come from telephone-based access to electronic mail or such common databases as a calendar or personal address book. Recently consumer products have appeared that use synthesis to speak traffic alerts, which are broadcast as text over pager frequencies.

SUMMARY

This chapter has served as an introduction to the technically difficult subject of synthesis of speech from text. The difficulty is due to the broad scope of the problem, ranging from linguistic analysis (parsing to find syntactic structure and prosody) to digital signal processing (vocal tract models and control parameters). It should be clear to the reader that speech synthesis is far from trivial and many factors influence the effectiveness of synthetic speech for the listener. There is much left to be learned about the fundamentals of speech production and the acoustic characteristics of natural speech. Speech synthesis will continue to improve, although progress may be slow and take place across a number of research areas.

There are some applications for which speech synthesis is superior to recorded speech, or where the latter simply is not adequate. Because of the steep learning curve encountered with synthetic speech, the prospective user must be enticed into initial use. The initial negative reaction to the speech quality can be overcome by an appropriate application with a well-designed user interface. The next chapter discusses in detail some guidelines for the development of such applications and offer examples of working systems as a means of exploring these issues.

FURTHER READING

Much of the pioneering work in speech synthesis in the United States was done by Dennis Klatt and Jonathan Allen at MIT's Research Laboratory in Electronics. In Allen *et al.* they describe along with coauthors the algorithms of MITalk, one of the early complete synthesis schemes. Klatt provides an excellent although slightly dated survey of international speech synthesis research in Klatt [1987b]. A section of Furui and Sondhi 1992 is devoted to speech synthesis with an especially strong overview by Allen. Witten [1982] provides a valuable perspective on speech synthesis in Witten, which is also a bit dated.