16

Speech Technologies

Speech technology covers applications such as the recognition or synthesis of speech, speaker recognition, and optimized coding and enhancement of speech. Major research efforts have been invested in speech technology both in academia and in industry, which have led to several breakthroughs. After several decades of research, coding, synthesis, and recognition of speech are used extensively in several applications. The widespread adoption of speech technologies has been enabled by the advent of sufficiently powerful and relatively inexpensive digital processors.

Many user interfaces based on speech exist. A computer may take commands by recording the speech of the user, thus requiring *speech recognition* abilities, and it may deliver messages to the user by producing intelligible speech, which requires *speech synthesis* technology. This interaction was shown conceptually in Figure I.5 on page 5. In mobile communication, the goal is often to present speech at as low a bit rate as possible, which has led to many methods for *speech coding*, where the special characteristics of speech signals have been taken into account.

This chapter provides a very brief overview of different technologies in speech coding, synthesis, and recognition. The focus is very much on acoustics, signal processing, and audio, and the linguistic and statistical aspects are, in many places, treated superficially. Overall, the aim of this chapter is to give a general description of the main fields in speech technology and to guide interested readers to more comprehensive sources.

Speech enhancement is a wide area, which covers such topics as *error concealment*, *noise reduction*, *bandwidth extension*, and *echo control* in the context of speech communication (Vary and Martin, 2006). The goal of noise reduction and echo control should be clear to the reader. Bandwidth extension refers to techniques used to extend the bandwidth of transmitted narrowband speech to deliver better speech quality to the user. Error concealment refers to applications where the speech decoder is forced to 'fill in' missing data from the received speech signal. For example, when some frames of speech are lost in communication, the content of missing frames has to be guessed through some intelligent solutions to maximize speech quality. These techniques are not discussed in detail in this book.

Communication Acoustics: An Introduction to Speech, Audio, and Psychoacoustics, First Edition.
Ville Pulkki and Matti Karjalainen.

16.1 Speech Coding

The idea in *speech coding* is to transmit or store and replay speech signals using minimal information capacity (number of bits) and with the best possible sound quality. In reality, this implies optimization, which is a trade-off between quality and cost. For an essentially unrestricted channel capacity, the speech signal can be conveyed with such high quality that no degradation is perceived, and there is no need to further improve quality. The transmission in this case is said to be *transparent*. Unfortunately, digital transmission (and storage) of speech requires, in practice, that the information capacity of the channel is limited. The number of bits per second to be transmitted, the bit rate, can be reduced by using proper coding techniques – the proper representation of the speech signal and quantization to a lower rate of bits. In wireless speech communication in particular, the channel capacity is limited, and cost vs. quality optimization is important.

Early analogue telephony transmitted speech as an electrical signal over a wired network. Although this seems a relatively straightforward approach, the system had some technical challenges, and seminal psychoacoustic testing was performed to find the smallest frequency region of speech that would still produce good enough speech quality for the receiver. The first frequency region suggested for telephones ranged from 250 Hz to 2750 Hz (Martin, 1930), but this was later altered to the range between 300 Hz and 3400 Hz, known as the *telephone band*. The telephone band was found to be a good compromise between technical constraints and the intelligibility of speech, and it is still in use today.

A basic technology for digital speech coding is the standard G.711, where the speech signal in the telephone band is sampled at 8 kHz using pulse-code modulation (PCM) with logarithmic quantization using 8 bits (ITU, 1988; Paez and Glisson, 1972). The first version of G.711 was finalized in 1972, after which it has been used widely in communication networks.

The high levels of a signal are thus represented with larger quantization steps than the lower levels. Logarithmic quantization generates unacceptable distortion in a music signal. However, the artefacts are not prominent with speech signals, and logarithmic quantization, in practice, gives a dynamic range of about 12 bits. Thus, when each sample is quantized using 8 bits, and sampled at the rate of 8 kHz, the resulting rate of transmission is 64 kbit/s, which is acceptable in wired communication, but relatively high for wireless telephony.

The demand for further reducing the data rate emerged when the technologies for digital mobile phones were developed in the 1980s. A more effective speech coding principle was developed, which takes advantage of the knowledge of speech production mechanisms. In principle, the method finds the parameters of a simple *source–filter model* of speech production and transmits only the model parameters, possibly with a residual signal not produced by the model (Kleijn and Paliwal, 1995). The filter parameters are often modelled using linear predictive coding, and the source is modelled as noise (unvoiced phonation) or an impulse train (voiced phonation). The linear predictive coding fits an all-pole filter, or an IIR filter, to the signal, as explained in Section 3.3.5 on page 59, and in parallel to this the parameters of the source signal are found, such as pitch, gain, and voicing of the excitation. The parameters of the source and the filter are adapted by measuring how well the system output fits the signal. The parameters are then transmitted and are used at the receiving site to synthesize speech, as shown in Figure 16.1.

There are many kinds of speech codecs designed for different purposes, such as the 13-kbit/s codec with regular pulse excitation with long-term prediction (RPE-LTP) for the first generation GSM mobile phones (ETSI, 1992). RPE-LTP uses an 8th-order linear prediction filter to

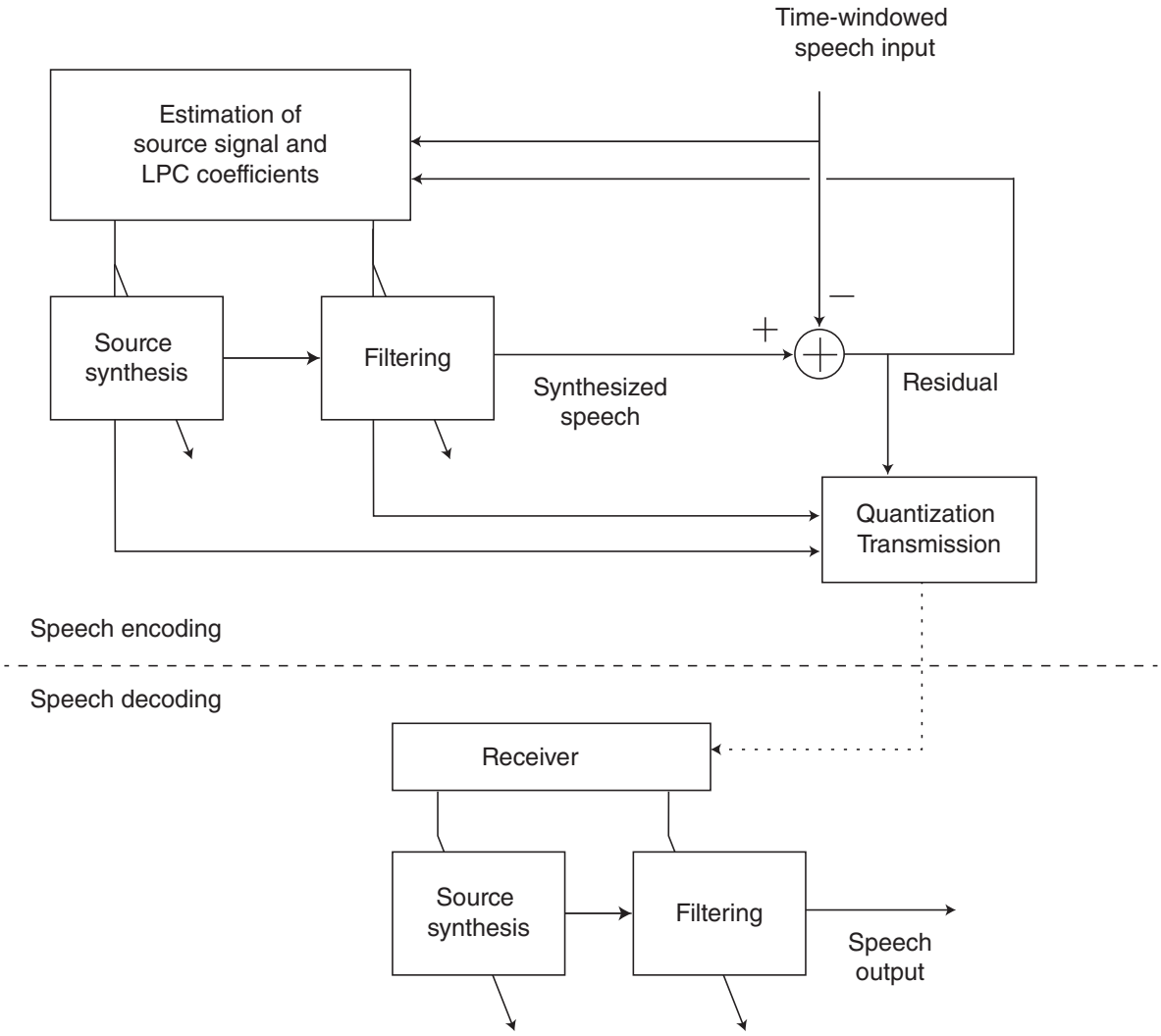

Figure 16.1 A schematic diagram of speech encoding and decoding. The parameters of the source–filter model are estimated iteratively, and the residual-error signal can be weighted perceptually to adapt the system to the most relevant features of speech. The transmitted data consist of filter and source parameters, and possibly the residual signal as well.

model the formant structure of speech, and a single-tap pitch predictor is used to model the periodic structure of voiced speech. Speech compression is achieved by 3:1 downsampling of the residual. Currently, different variants of the code-excited linear prediction (CELP) codec are used widely. In CELP, the residual signal is not sent as a signal, only a few parameters describing the residual are transmitted, which gives a significantly better speech quality at low bit rates. In VoIP applications, the codec commonly used is the conjugate-structure algebraic code-excited linear prediction (CS-ACELP) codec specified in standard G.729 (Salami *et al.*, 1998). The adaptive multi-rate (AMR) codec is commonly used in mobile phones (ETSI, 2011). It was standardized by the European Telecommunications Standards Institute (ETSI) in 1999. It is based on the ACELP technique, and is also able to use in transmission eight different bit rates ranging from 4.75 kbps up to 12.2 kbps. This enables adaptive allocation of bit stream resources to maximize the quality of experience.

The 64-kbit/s data rate obtained with the PCM codec described above can be reduced to rates of 8–12 kbit/s with parametric coding of speech, which is a tolerable data rate in wireless communication. The perceptual quality of speech does not drop noticeably when no strong background sounds are present (Kleijn and Paliwal, 1995).

A trend in speech coding is *wideband coding*, where 'wideband' typically refers to the acoustic bandwidth from 50 Hz to 7000 Hz. This bandwidth gives a substantially better speech quality with 'brighter' and 'fuller' sound compared to narrowband speech. Wideband coding requires, naturally, a higher transmission rate than narrowband codecs, and the use of such codecs is made possible by faster VoIP connections and more powerful mobile networks. In addition to wideband coding, superwideband, with a 14-kHz bandwidth, and fullband, with a 20-kHz bandwidth, have also been introduced. A review of codecs with these wider frequency ranges is given by Cox *et al.* (2009).

Speech transmitted by parametric methods is rated to be intelligible to a high degree, although the listeners may characterize it as being a bit 'synthetic'. Relatively often, the obtained quality depends on the speaker. With some codecs the quality obtained with male speakers is better than with females (Vary and Martin, 2006, p. 285). Since the encoder makes a strong assumption that the encoded sound is a human voice, a natural consequence is that the quality of reproduction degrades for some other signals, such as with music. This also leads to a situation where, in conditions of strong background noise, the encoder may make wrong assumptions with regard to the parameters of the source–filter model, and the speech reproduced may turn out to be unintelligible. However, some codecs are designed to use source–filter models of speech only when the sound evidently is speech; the codec changes the mode of operation to coding of a waveform if the input signal is not single-source speech (Neuendorf *et al.*, 2013).

Speech coding methods thus use the principle behind the *vocoder* (short for voice encoder), which was developed in the 1930s (Dudley, 1940). In the original implementation, the input speech signal was passed through a filter bank, and the signal in each band was passed through an envelope follower. The control signals from the envelope followers was communicated to the decoder. The decoder applied these (amplitude) control signals to corresponding filters in the synthesizer. Since the control signals changed only slowly compared to the original speech waveform, the bandwidth required to transmit speech was reduced, and the parameters could be modelled statistically. Nowadays, the term vocoder is used to mean the principle where speech is dissolved into relatively simple parameters, and synthesized back to a perceptually similar speech signal.

16.2 Text-to-Speech Synthesis

Synthetic speech, in the sense of artificially created speech signals, has a relatively long history, including the acoustic–mechanical speech production by Kratzenstein and Von Kempelen (Schroeder, 1993) in the late 18th century, a mechanically controlled electronic synthesizer called the *Voder* by Dudley in 1939, and more advanced electronic synthesizers since the 1950s (Karjalainen and Laine, 1977; Karjalainen *et al.*, 1980; Klatt, 1987; Schroeder, 1993). These electronic models of human voice generation then evolved to be controlled by computer, enabling intelligible *text-to-speech synthesis*.

The main principle of speech synthesis is to use a signal model of speech production, such as the source–filter method discussed in the previous section, and to control the parameters of the model to create speech, realizing the text input of the system as understandable and providing as natural speech as possible. The success of the source–filter approach, or vocoder approach, in speech synthesis proves that good-quality speech can be synthesized if the parameters are

derived from natural speech. Unfortunately, deriving the parameters from text input is far from an easy task, as will be shown below.

This section first reviews the early knowledge-based synthesis methods, where complex rule-based systems were built for speech synthesis. Although such methods have been abandoned, they might still be of interest for historical and educational purposes, and for this reason they are discussed here.

Data-based synthesis methods have largely been adopted in academia and in industry since the 1990s. The increase in power and resources of computer technology has enabled the building of natural-sounding synthetic voices based on the utilization of large, single-speaker databases of natural speech (Zen *et al.*, 2009). In contrast to knowledge-based synthesis, each phonetic unit need not be crafted for each applicable context anymore, but the information in the database is used to train the system on how to produce correct-sounding speech.

Data-based methods can be divided into two subclasses: *unit-selection synthesis* and *statistical parametric synthesis*. Unit selection synthesis does not perform actual 'synthesis' of sound, but the output signal is composed of audio samples taken from the database (Beutnagel *et al.*, 1999). Statistical parametric synthesis, on the other hand, synthesizes the speech signal using a model, the parameters of which are controlled using a system trained with real speech samples.

16.2.1 Early Knowledge-Based Text-to-Speech (TTS) Synthesis

The source–filter signal models of human voice were discussed in Section 5.3 on page 90. In principle, any static voice produced by the human speech organs can be produced with such source–filter models. The goal of text-to-speech synthesis is actually much more complicated, as the system should be able to produce words, sentences, and complete utterances that are not only intelligible, but also indistinguishable from human speech in the ears of a listener. The first efforts to synthesize speech created sets of rules on how to go from text to control parameters of the signal model of voice and finally to intelligible speech.

An overall structure of knowledge-based *text-to-speech synthesis* is characterized in Figure 16.2. The first phase is to read text in and parse it into an internal representation in terms of phonetics and linguistics of the language to be synthesized. Various methods are applied here depending on the complexity and degree of linguistic processing. A simple case might have only *text normalization*, such as expanding abbreviations, numbers, and possibly some application-specific rules. *Letter-to-phoneme mapping* is always needed, which is simple in some languages (Finnish is an example) and complex in others (like English or French).

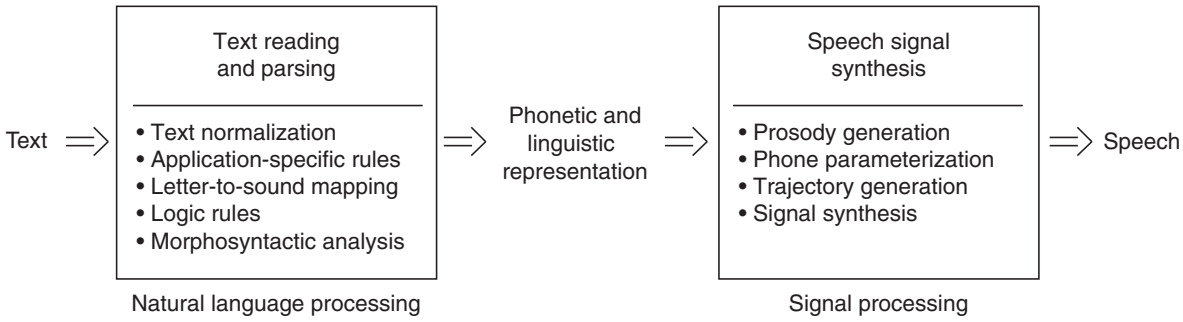

Figure 16.2 A general description of knowledge-based text-to-speech synthesis (TTS).

In a more advanced case, the parsing may include morphological and syntactic analysis of the language being synthesized. These functions are examples of *natural language processing*.

The second phase of knowledge-based text-to-speech synthesis, depicted in Figure 16.2, converts the phonetic and linguistic representation to a speech signal. This part, consisting largely of signal processing, gives a parameterized representation of the phonemes (phones) and attaches prosodic features to them. Continuous-time parameters are produced which control the final speech signal synthesis that may be based on any synthesis model, such as those discussed in Section 5.3.

Figure 16.3 illustrates an example of the organization of the synthesis, and shows the multilevel structure of information and data in speech generation. In this specific case, the linguistic pre-processing is simple, resulting in a phoneme string representation using letter-to-phoneme rules. The linguistic analysis also contains a simple analysis of the syllabic and sentence structure of the message. Based on these representations, the synthesis process proceeds to prosodic feature and segmental parameter computation for the segments of the phonemes. The next step is to convert the segmental parameters into continuous-time trajectories for synthesis model control.

The largest problem with rule-based text-to-speech synthesizers is the quality of speech. When unlimited text is automatically transformed into speech, the synthesized speech almost always sounds 'robotic' and is barely intelligible. These methods have been largely abandoned after the introduction of data-based speech synthesis, which will be discussed below. However, in some special cases, knowledge-based speech synthesizers have their uses. For example, in a situation where data-based speech synthesis would require far more computational power than is available and where a lower quality of speech could be tolerated, the rule-based solutions could still be used.

16.2.2 Unit-Selection Synthesis

A database consisting typically of tens of hours of recorded speech has first to be created for unit-selection synthesis. During the creation of the database, each recorded utterance is segmented into some or all of the following: individual phones, diphones, half-phones, syllables, morphemes, words, phrases, and sentences. The segmentation can be done using automatic systems, although often the segments have to be manually fine-tuned.

The units in the speech database are then indexed, and the segments are associated with both phonetic and prosodic information. The phonetic information labels the phone and also describes its *phonetic context* as the position of the phone in the syllable, word, and sentence. The prosodic parameters are then the acoustic parameters analysed from the phone, such as the fundamental frequency (pitch), voicing, and spectral structure. The *prosodic context* then describes the prosodic parameters in neighbouring segments.

At run time, the text to be synthesized is given, and a set of required units is formed based on the text. The desired target utterance is created by determining the best chain of candidate units from the database, which is the *unit selection* process. An important concept is the *target cost*, which measures how well a candidate unit from the database matches the required unit. It is basically a distance measure. Similarly, the *concatenation cost* defines how well two units combine when presented successively. This evaluation can be performed using a specially weighted decision tree, where the cost is to be minimized. The best-matching units are then concatenated, as shown in Figure 16.4.

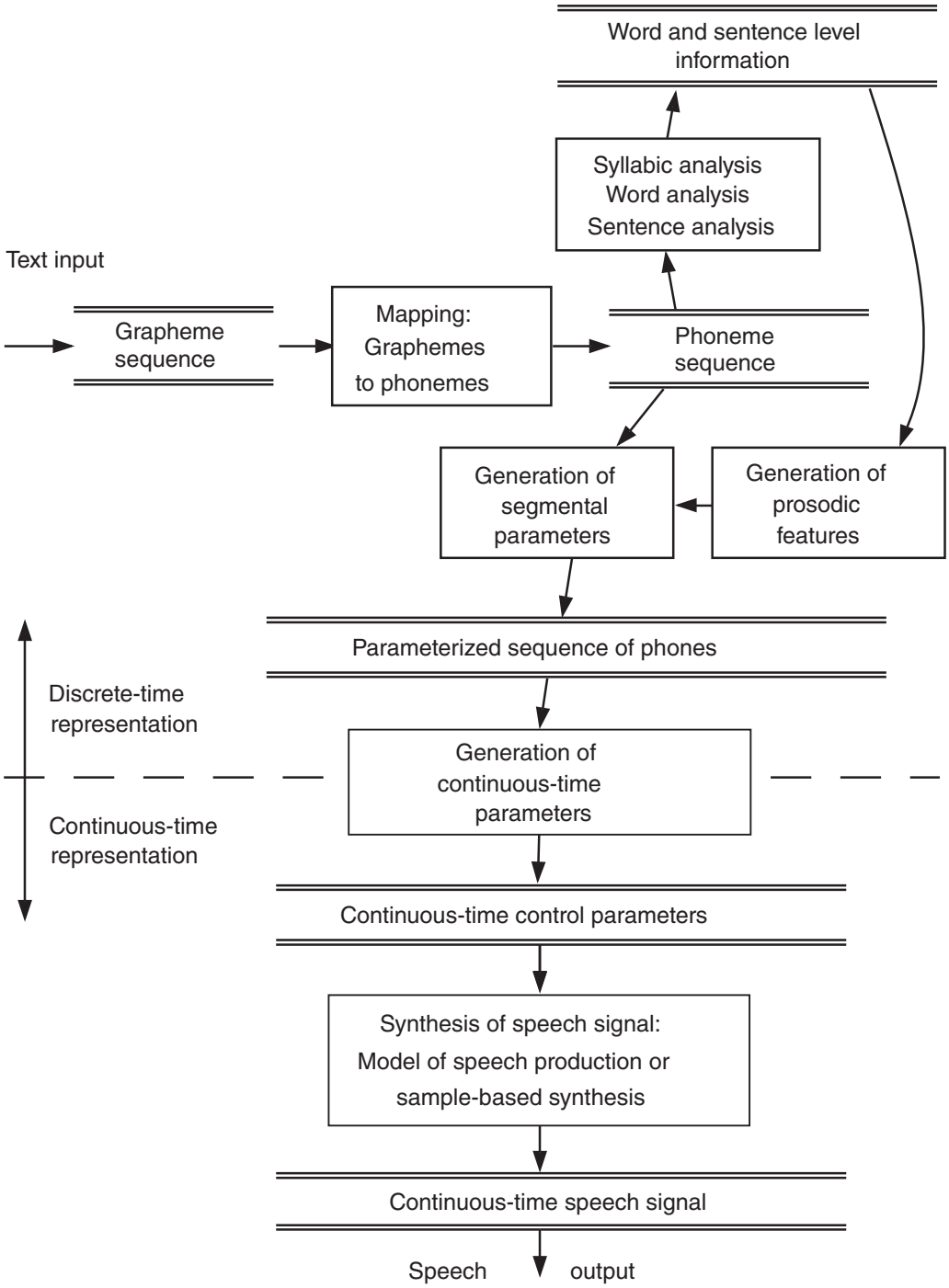

Figure 16.3 Knowledge-based text-to-speech synthesis (TTS) as multi-level information processing.

When designing an implementation of unit-selection synthesis, the size of the units may be set to be shorter or longer. If they are made shorter, more joining points will be available to select the next unit. Since there are more possibilities to select the joining point, there will be a smaller concatenation cost, which leads to better quality. However, too short a length for the units may also cause problems. Kishore and Black (2003) tested the optimal length for the Hindi language, and the best result was obtained with the unit length equalling that of syllables.

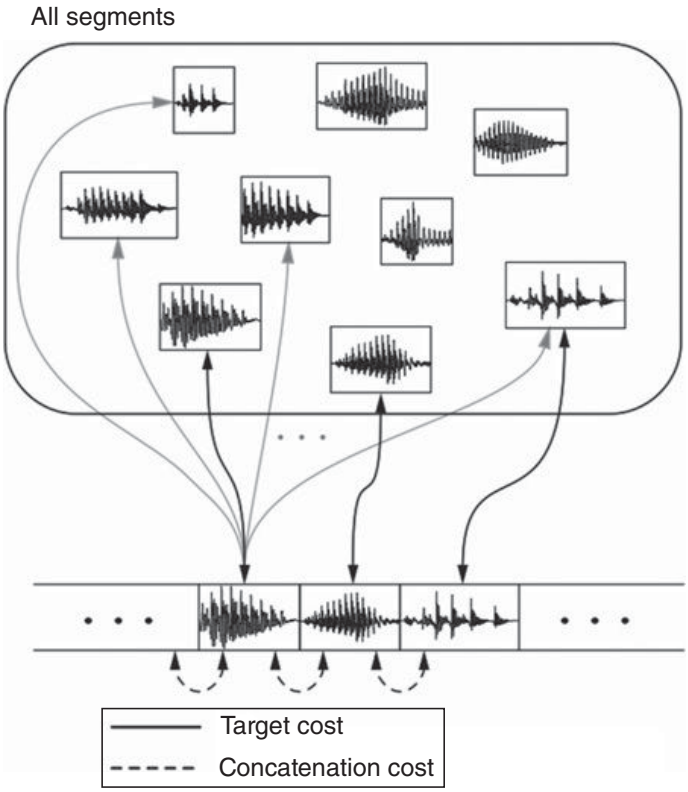

Figure 16.4 An overview of the general unit-selection scheme. The solid lines represent target costs and the dashed lines concatenation costs. Adapted from (Zen *et al.*, 2009), and reprinted with permission from Elsevier.

The units can also be made to have variable lengths, which may at least partly mitigate this issue (Segi *et al.*, 2004).

In principle, unit selection produces a very natural result, because it does not change significantly the recorded speech. The output from the best unit-selection systems can be indistinguishable from real human voices, especially in applications for which the system has been tuned (Beutnagel *et al.*, 1999). Unfortunately, achieving the most natural results typically requires unit-selection speech databases to be very large, representing tens of hours of speech, which can be a limiting factor in some applications. The method does not itself provide strong means to modify the type of speaker nor to change prosodic features dynamically, since in principle it is based on the playback of catenated speech samples. Also, the quality of the system may severely deteriorate if the phonetic and prosodic contexts required in a sentence are under-represented in a database (Zen *et al.*, 2009), and the errors caused by mismatches between concatenated phones reduce the intelligibility of speech.

16.2.3 Statistical Parametric Synthesis

Statistical parametric synthesis methods try to utilize the best properties of signal models and statistical models of natural speech. The source–filter models of speech are often used in statistical parametric synthesis, and they will be used to illustrate the techniques in this section.

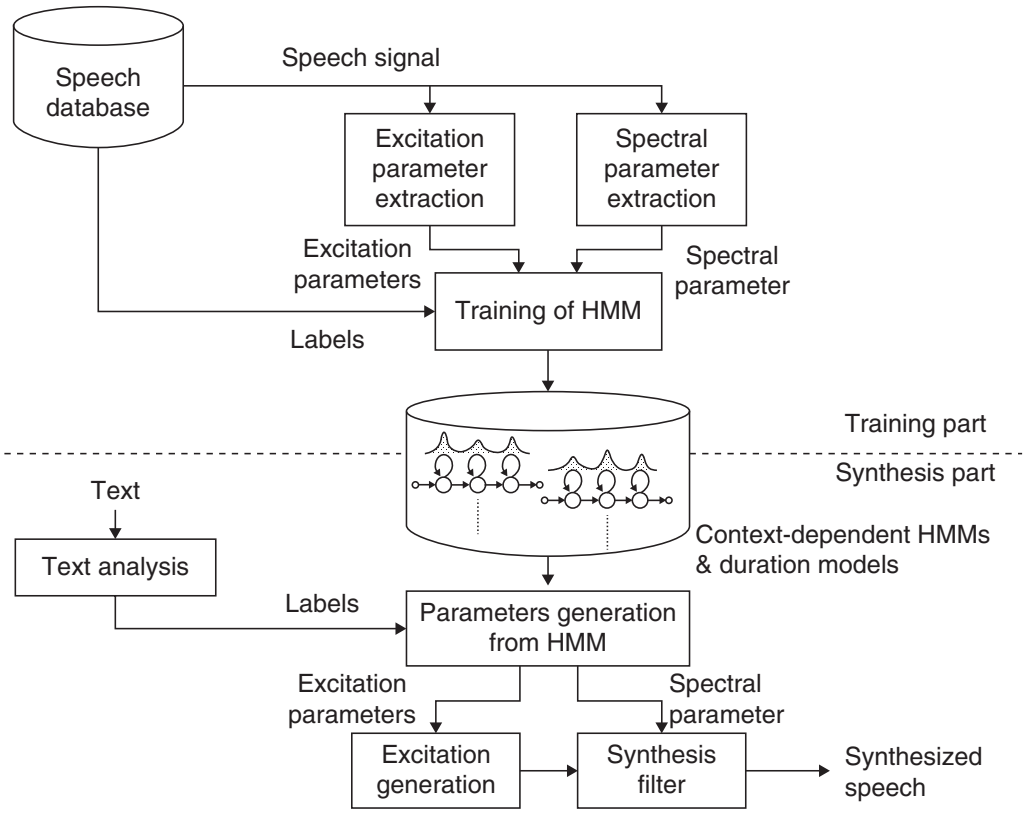

Figure 16.5 A block diagram of statistical parametric speech synthesis implemented with hidden Markov models (HMM). Adapted from Zen *et al.* (2009), and reprinted with permission from Elsevier.

The basic idea behind this type of synthesis does not, in principle, limit the choice of modelling technique; for example, sinusoidal modelling of speech has also been used in the same context (Erro *et al.*, 2014; Masuko *et al.*, 1996). The driving idea is to derive the parameters for source–filter synthesis by training the system using large speech databases. The promise is that the speech synthesized from text should have better quality than with knowledge-based synthesis, but in addition, the flexibility and adaptivity to modify the characteristics of voice obtainable with source–filter synthesis should be gained.

A block diagram of the statistical parametric system based on hidden Markov models (HMM) is shown in Figure 16.5. Before the training part, a large speech database is collected, where, again, a data structure of phonemic and prosodic context is associated with each segment of speech. The size of the speech database is typically much smaller than with unit-selection synthesis, as, with just one hour of speech data, decent synthesis results can be obtained (Yamagishi *et al.*, 2009). The structure contains information such as the current phoneme and its position, adjacent phonemes, the current syllable and its position, adjacent syllables, and information on nearby syllables. The structure may also contain similar data about words, phrases, and utterances in the context of each segment.

In the HMM-based speech synthesis framework, each context-dependent phoneme is modelled as a sequence of HMM states (commonly 3 or 5). Each state models the vocoder parameters using a single Gaussian distribution for each parameter. In the training of the system, the principle of the vocoder is utilized: the most suitable parameters for excitation and

spectral filtering are computed that are assumed to produce an output perceptually similar to the original segment when applied to the source–filter model in synthesis. This is done by first decomposing the speech data into vocoder parameters, and then estimating the Gaussian statistics (mean and variance) of the vocoder parameters and the duration for each state for all context-dependent phonemes. Using multi-stream HMM training (Gales and Young, 2008), the system can model the spectrum, excitation, and duration of each segment in a unified framework.

The parameters utilized in the source and filter depend on their actual implementation. For example, the spectral parameters may be represented as mel frequency cepstral coefficients (Fukada *et al.*, 1992) or as line spectral frequencies (Soong and Juang, 1984), and the fundamental frequency of the voiced excitation is often presented as the logarithm of f_0. The computation of mel cepstra was discussed in Section 13.1.1. The logarithm of f_0, in turn, corresponds to the frequency scale used in music (see Section 11.6.2). Commonly, a simple impulse train is used for exciting voiced speech, but the waveform of a voiced glottal excitation, inverse filtered from natural speech, may also be used to improve naturalness (Raitio *et al.*, 2011). The STRAIGHT method is nowadays commonly used as a speech vocoder (Kawahara, 2006), and in other speech processing besides data-based speech synthesis. In STRAIGHT synthesis, the source signal is generated using mixed excitation consisting of impulses and a noise component acting as the aperiodic component of voiced speech. Finally, the pitch-synchronous overlap add (PSOLA) method (Moulines and Charpentier, 1990) is used to reconstruct the excitation signal, which is then applied to excite the filter.

The synthesis part performs an operation that resembles the inverse of speech recognition (Zen *et al.*, 2009). A given word sequence is first converted into a sequence of context-dependent labels. After this, the HMM of an utterance is constructed by concatenating the context-dependent HMMs following the label sequence. Then, smooth sequences of spectral and excitation parameters are generated from the utterance HMMs using the mean and variance values of each state. Finally, a speech signal is synthesized using excitation generation and a speech synthesis filter.

With regard to the quality of speech obtained with statistical parametric synthesis, Zen *et al.* (2009) conclude with the words, 'Although even the proponents of statistical parametric synthesis feel that the best examples of unit-selection synthesis are better than the best examples of statistical parametric synthesis, overall it appears that the quality of statistical parametric synthesis has already reached a level where it can stand in its own right.' The speech quality obtained with statistical parametric synthesis is thus not on the same level with unit-selection synthesis, but better quality is obtained than with knowledge-based speech synthesis.

One benefit of statistical parametric synthesis is that the source–filter-type synthesis of speech signals provides possibilities for changing the characteristics of voice, such as speaking styles and prosodic features, and possibly even provides multilingual support (Zen *et al.*, 2009). With the unit-selection method, such modifications are a bit complicated, as voice conversion techniques have to be utilized (Stylianou *et al.*, 1998), which may degrade the quality of the speech.

However, although understandable speech is obtained with statistical parametric synthesis, its major drawback is still in the quality of speech. The speech is perceived as less authentic than with unit-selection synthesis, and several possible factors affecting this and various suggested refinements are reviewed by Zen *et al.* (2009).

16.3 Speech Recognition

The goal of *speech recognition* (Rabiner and Juang, 1993) is to capture the acoustic human speech signal and process the spoken language into text. It is one of the biggest challenges in speech technology. Evolution and culture have built speech into a fast and robust communication channel over adverse acoustic channels, tolerant to relatively high background noise levels. Speech is not only a rich container of information added to words and non-speech voices, it also carries the identity, age, and gender about the speaker and prosodic features carrying emotions and other meanings. Additionally, the speed of speech may vary substantially, and the vast number of languages, dialects, and speaking styles and disorders make the problem even more challenging.

In principle, speech recognizers should simply mimic brain functions to perform the task as well as humans do. Note that the goal of processing is different in brains and in speech recognizers. Brains turn speech into neural code and speech recognizers to written text. Speech recognition conducted by humans is based strongly on the high-level functions of the brain, such as language, awareness of environment, and assumptions about the content of speech based on previous utterances. Auditory models and models of the brain are clearly not at the level where such functions can be emulated. Thus, the most successful speech recognition techniques are based on generic principles of pattern recognition and data processing.

The complexity of a speech recognition task depends heavily on the definition of the task. The simplest task is to recognize speech from one known speaker uttering temporally separated words from a small, known vocabulary. The current methods perform well in such tasks. The task becomes harder when the vocabulary is made larger or not restricted at all, when words are not separated by silences, when multi-language speech is allowed, and when the content of speech is not limited to any specific topic. Even harder tasks are those involving multiple speakers, and when the level of background noise is high.

The speech recognition process can be divided into different phases. The first phase is to pre-process the speech signal to remove unnecessary redundancy and to describe the signal with *features*. Given the acoustic models of speech sounds, the statistical language models, and the lexicon of words, the next phase is a *pattern recognition* task where the decoder turns the sequence of features directly into the most likely sequence of words in the language.

The most common speech recognition systems are based on HMMs (Rabiner, 1989). The early techniques in speech recognition were dynamic time warping (Vintsyuk, 1968) and neural networks (Kohonen, 1988), but they were largely abandoned in the late 1990s. However, recently, deep neural networks have been proposed for use in speech recognition (Dahl *et al.*, 2012).

The main parts of a typical speech recognizer are shown in Figure 16.6. The speech input is turned into feature vector sequences first. After this, the probability of the vectors or vector sequences belonging to linguistic classes is computed using HMMs. The probability sequences are then decoded into text using a vocabulary and a model of language. Prior to using the recognizer, the HMM has to be trained with natural speech, and the vocabulary and model of language have to be constructed.

Speech recognizers commonly window the input signal at a rate of about 100 Hz and utilize window lengths of about 20 ms. The features computed from the signal segment are typically mel-frequency cepstral coefficients. Differentials between temporally subsequent features are also commonly utilized and have been found to improve the accuracy of recognition.

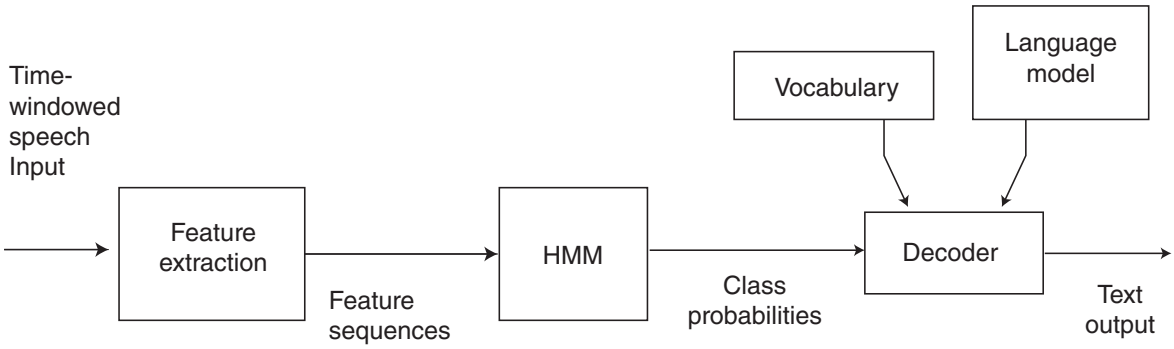

Figure 16.6 A simplified block diagram of speech recognition.

If the recognition task is limited to a small vocabulary and to isolated words, it is possible to construct separate HMMs for all the different words without sharing parameters between the models. The number of states in a word HMM is then determined with some heuristics, such as relating it to the number of phonemes or the average number of observations (Rabiner, 1989). As the size of the vocabulary increases, a word-level approach becomes infeasible due to the large number of states and parameters. Instead, a small set of segments, such as phones or diphones, is modelled with unique HMMs. The word models can then be formed by a simple concatenation of these units. A three-state, left-to-right HMM (Schwartz *et al.*, 1985) is commonly used for the model of a phoneme.

The sequences of probabilities of phonemes are then turned into written text using knowledge about the language they were spoken in. Some information about the language is already embedded in the HMMs of the units of speech, either in the form of word models or as the phone inventory. However, an extensive knowledge of language must be embedded in the recognizer. Most commonly, the grammatical constraints are learned as statistical dependences in the context of words from large text databases (Goodman, 2001). When the vocabulary is small, it may be feasible to make a list of allowed words and define their HMMs. However, when the size of the vocabulary is larger, acoustically similar words are hard to distinguish during recognition. Detecting boundaries of words in continuous speech is also difficult, which adds to the challenge. It becomes necessary to utilize either task-specific or general constraints to overcome these problems.

Speech recognition has already been adopted in many applications. For example, some televisions and handheld devices can already be controlled by voice, and telephone customer services may use speech recognition in automated tasks. Speech recognition is effective in browsing the contents of spoken speech databases (Chelba *et al.*, 2008). One application that is also actively being studied is a telephone service that translates the language of a speaker into another language. Although this service has not yet been achieved, translation in limited cases has been possible. Optimally, the characteristics of the voice of the original speaker should be preserved in the translated speech output. The biggest challenges are, however, in the processing of language, and not in the processing of speech signals.

Summary

Speech coding is perhaps the most mature of the speech technologies. Very good speech quality can be obtained with low bit rates when signal models of speech production mechanisms are utilized. The methods are relatively simple, and no large databases are required.

Speech synthesis and recognition techniques have also been developed fairly extensively. When compared with speech coding, a clear difference is that the systems must have access to large databases of natural language and/or speech in order to obtain good results. Speech synthesis and recognition thus require that the computer has knowledge of the recognized or synthesized language at some level, something that is not needed in speech coding. The performance of the synthesis and recognition techniques is still far from that of human performance, although multiple commercial applications already exist.

Further Reading

The synthesis and recognition of speech is described in more detail by Huang *et al.* (2001), Jurafsky and Martin (2008), and Saon and Chien (2012). A public domain toolkit for research in automatic speech recognition is available, and its documentation also serves as a good explanation of speech recognition (Young *et al.*, 2006). Hidden Markov models are explained in detail by Gales and Young (2008). The enhancement, coding, and error concealment in speech transmission are discussed by Vary and Martin (2006), and more information on audio coding can be found in Chu (2004).

References

Beutnagel, M., Conkie, A., Schroeter, J., Stylianou, Y., and Syrdal, A. (1999) The AT&T next-gen TTS system. *Joint meeting of ASA, EAA, and DAGA*, pp. 18–24.

Chelba, C., Hazen, T.J., and Saraçlar, M. (2008) Retrieval and browsing of spoken content. *Signal Proc. Mag., IEEE*, **25**(3), 39–49.

Chu, W.C. (2004) *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*. John Wiley & Sons.

Cox, R.V., De Campos Neto, S.F., Lamblin, C., and Sherif, M.H. (2009) ITU-T coders for wideband, superwideband, and fullband speech communication. *Communications Mag., IEEE*, **47**(10), 106–109.

Dahl, G.E., Yu, D., Deng, L., and Acero, A. (2012) Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio, Speech, and Language Proc.*, **20**(1), 30–42.

Dudley, H. (1940) The carrier nature of speech. *Bell Sys. Tech. J.* **19**(4), 495–515.

Erro, D., Sainz, I., Navas, E., and Hernaez, I. (2014) Harmonics plus noise model based vocoder for statistical parametric speech synthesis. *IEEE J. Selected Topics in Signal Proc.*, **8**(2), 184–194.

ETSI (1992) GSM Full Rate Speech Transcoding. Recommendation ETSI GSM 06.10, European Telecommunication Standards Institute.

ETSI (2011) AMR speech codec, general description. Standard 3GPP TS 26.071, version 10.0.0, 3rd Generation Partnership Project.

Fukada, T., Tokuda, K., Kobayashi, T., and Imai, S. (1992) An adaptive algorithm for mel-cepstral analysis of speech. *IEEE Int. Conf. Acoustics, Speech, and Signal Proc.*, volume 1, pp. 137–140 IEEE.

Gales, M. and Young, S. (2008) The application of hidden Markov models in speech recognition. *Found. Trend. Sig. Proc.*, **1**(3), 195–304.

Goodman, J.T. (2001) A bit of progress in language modeling. *Comp. Speech Lang.*, **15**(4), 403–434.

Huang, X., Acero, A., Hon, H.W., and Foreword By-Reddy, R. (2001) *Spoken Language Processing: A guide to theory, algorithm, and system development*. Prentice Hall.

ITU (1988) *Pulse code modulation (PCM) of voice frequencies*. Recommendation ITU-T G.711, International Telecommunication Union, Geneva, Switzerland.

Jurafsky, D. and Martin, J.H. (2008) *Speech and Language Processing: An Introduction To Natural Language Processing, Computational Linguistics, and Speech*, 2nd edn. Pearson Prentice Hall.

Karjalainen, M. and Laine, U.K. (1977) Speech synthesis project in Tampere: Results and applications. *Proc. of IV Nordic Meeting on Med. and Biol. Eng.,* pp. 30.1–3.

Karjalainen, M., Laine, U., and Toivonen, R. (1980) Aids for the handicapped based on “synte 2" speech synthesizer. *IEEE Int. Conf. Acoustics, Speech, and Signal Proc.*, volume 5, pp. 851–854 IEEE.

Kawahara, H. (2006) Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds. *Acoust. Sci. Technol.* **27**(6), 349–353.

Kishore, S. and Black, A.W. (2003) Unit size in unit selection speech synthesis. *INTERSPEECH*.

Klatt, D.H. (1987) Review of text-to-speech synthesis of English. *J. Acoust. Soc. Am.*, **82**(3), 737–793.

Kleijn, W.B. and Paliwal, K.K. (1995) *Speech Coding and Synthesis*. Elsevier Science.

Kohonen, T. (1988) The'neural'phonetic typewriter. *Computer*, **21**(3), 11–22.

Martin, W. (1930) Transmitted frequency range for telephone message circuits. *Bell Sys. Tech. J.* **9**(3), 483–486.

Masuko, T., Tokuda, K., Kobayashi, T., and Imai, S. (1996) Speech synthesis using HMMs with dynamic features. *IEEE Int. Conf. Acoustics, Speech, and Signal Proc.*, volume 1, pp. 389–392 IEEE.

Moulines, E. and Charpentier, F. (1990) Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Commun.* **9**(5), 453–467.

Neuendorf, M., Multrus, M., Rettelbach, N., Fuchs, G., Robilliard, J., Lecomte, J., Wilde, S., Bayer, S., Disch, S., Helmrich, C., Lefebvre, R., Gournay, P., Bessette, B., Lapierre, J., Kjörling, K., Purnhagen, H., Villemoes, L., Oomen, W., Schuijers, E., Kikuiri, K., Chinen, T., Norimatsu, T., Chong, K.S., Oh, E., Kim, M., Quackenbush, S., and Grill, B. (2013) The ISO/MPEG unified speech and audio coding standard – consistent high quality for all content types and at all bit rates. *J. Audio Eng. Soc.*, **61**(12), 956–977.

Paez, M. and Glisson, T. (1972) Minimum mean-squared-error quantization in speech pcm and dpcm systems. *IEEE Trans. Commun.*, **20**(2), 225–230.

Rabiner, L. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**(2), 257–286.

Rabiner, L.R. and Juang, B.H. (1993) *Fundamentals of Speech Recognition*, volume 14. Prentice Hall.

Raitio, T., Suni, A., Yamagishi, J., Pulakka, H., Nurminen, J., Vainio, M., and Alku, P. (2011) HMM-based speech synthesis utilizing glottal inverse filtering. *IEEE Trans. Audio, Speech, and Language Proc.*, **19**(1), 153–165.

Salami, R., Laflamme, C., Adoul, J.P., Kataoka, A., Hayashi, S., Moriya, T., Lamblin, C., Massaloux, D., Proust, S., Kroon, P., and Shoham, Y. (1998) Design and description of CS-ACELP: A toll quality 8 kb/s speech coder. *IEEE Trans. Speech and Audio Proc.*, **6**(2), 116–130.

Saon, G. and Chien, J.T. (2012) Large-vocabulary continuous speech recognition systems: A look at some recent advances. *IEEE Signal Proc. Mag.*, **29**(6), 18–33.

Schroeder, M.R. (1993) A brief history of synthetic speech. *Speech Commun.* **13**(1), 231–237.

Schwartz, R., Chow, Y., Kimball, O., Roucos, S., Krasner, M., and Makhoul, J. (1985) Context-dependent modeling for acoustic-phonetic recognition of continuous speech. *IEEE Int. Conf. Acoustics, Speech, and Signal Proc.*, volume 10, pp. 1205–1208 IEEE.

Segi, H., Takagi, T., and Ito, T. (2004) A concatenative speech synthesis method using context dependent phoneme sequences with variable length as search units. *Fifth ISCA Workshop on Speech Synthesis*.

Soong, F.K. and Juang, B.H. (1984) Line spectrum pair (lsp) and speech data compression. *IEEE Int. Conf. Acoustics, Speech, and Signal Proc.*, volume 9, pp. 37–40 IEEE.

Stylianou, Y., Cappé, O., and Moulines, E. (1998) Continuous probabilistic transform for voice conversion. *IEEE Trans. Speech and Audio Proc.*, **6**(2), 131–142.

Vary, P. and Martin, R. (2006) *Digital Speech Transmission: Enhancement, Coding and Error Concealment*. John Wiley & Sons.

Vintsyuk, T. (1968) Speech discrimination by dynamic programming. *Cybernet. Sys. Anal.*, **4**(1), 52–57.

Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., and Isogai, J. (2009) Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. *IEEE Trans. Audio, Speech, and Language Proc.*, **17**(1), 66–83.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X.A., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2006) Htk book (for htk version 3.4). Technical report. `http://htk.eng.cam.ac.uk/docs/docs.shtml`.

Zen, H., Tokuda, K., and Black, A.W. (2009) Statistical parametric speech synthesis. *Speech Commun.*, **51**(11), 1039–1064.