Speech Synthesis: Toward a "Voice" for All

H. Timothy Bunnell

Text to speech (TTS) has become so much a part of our everyday lives thanks to Alexa, Google, Siri, and many others that we have come to know (if not always love) that it is difficult to recall a time when it was not so. Synthetic voices like those for Siri and others fill multiple roles today. They deliver announcements of important information over public address systems in noisy places like airports where high intelligibility of the speech in noise is crucial to ensure the information they carry is heard correctly. A synthetic voice may be the first entity a customer interacts with when contacting a company and it is important for the voice, as a representative of the company, to present a natural and pleasing voice quality that is representative of the company's image. Synthetic voices serve as the *only* voice for individuals whose own voice is lost due to injury or a progressive neurological disease like amyotrophic lateral sclerosis (ALS; also called Lou Gehrig's disease or motor neuron disease [MND]) or who have a congenital dysarthria due to a condition such as cerebral palsy. And TTS voices allow blind or nonliterate users to read content from news stories, books, and computer screens while giving busy people an opportunity to "read" email even when driving their car.

A Framework and Baseline for Text to Speech

These current use cases for TTS voices provide insight into the successes of the underlying technology and also highlight areas where work remains. The need for intelligibility, naturalness, and ability to convey an individual's vocal identity are obvious from these examples. Less obvious but no less important is the expressiveness of the synthetic speech: the ability to express through intonation or "tone of voice" (Pullin and Hennig, 2015) the intent underlying the words of an utterance.

In this article, I trace how we arrived at the current state of the science for TTS, showing how the technology improved with the adoption of newer approaches and improved numerical techniques. A natural start is with the work of Klatt (1980) who provided Fortran software for implementing a cascade/parallel formant synthesizer. Klatt (1987) provided a history of TTS conversion, which was remarkable for the inclusion of a collection of audio examples for many of the synthesizers he discussed (see Ramsay, 2019, for an interesting review of early mechanical synthesizers).

Crucially, the period around the publication of these two articles by Klatt (1980, 1987) marked an important era in the TTS field. From a purely commercial perspective, it was arguably during this time that TTS systems became commercially mainstream, largely through improvements in the intelligibility of the speech that they generated.

Second, during this period, TTS technology started to be adopted by nonvocal persons to enhance their ability to communicate with others. One of Klatt's visions for Digital Equipment Corporation's DECtalk system, which emerged directly from his work at MIT, Cambridge, Massachusetts, was its application in augmentative and alternative communication (AAC) devices for communication by individuals who are nonvocal. Until that time, augmented communicators depended mainly on mechanical communication boards that required communicants to point to words or letters to express themselves. Recently, the field has come to refer to these speech-enabled communication devices as speech-generating devices (SGDs), the term I use in this article.

In this article, I present a framework that captures the structure and function of the TTS advances. Throughout, a goal is to focus on the implications for SGD users' communication.

Figure 1 provides a unified framework for discussing modern TTS systems. Each block or component in the

https://doi.org/10.1121/AT.2022.18.1.14

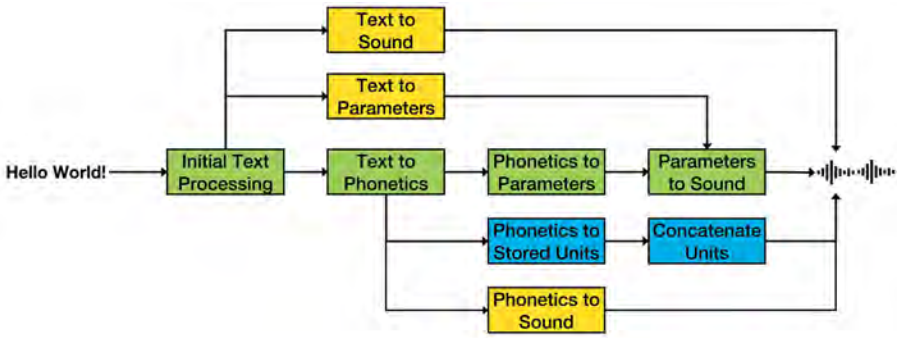


Figure 1. *Unified schematic covering current text to speech (TTS) system designs.* ***Colors*** *highlight components for different types of TTS systems.* ***Green components*** *are shared by many types of TTS systems. See* ***Figure 2***, ***green*** *and* ***blue****, and* ***5***, ***green*** *and* ***yellow****, for specific pathways.*

figure represents a *logical* element of the TTS process as it is usually conceived. I start with a description of a generic rule-based formant synthesizer like DECtalk (**Figure 1**, *green*). I focus on this pipeline to set the baseline to show the types of changes that have been made over time to improve the technology.

Formant Synthesis from Rules

Formant synthesis systems (and virtually all other TTS systems I discuss) require some form of *initial text processing* (**Figure 1**, *green*). Typically, this involves tokenizing the input text stream into distinct words or tokens and text normalization to convert nonword tokens such as numbers and abbreviations into the words one would speak when reading the tokens aloud. Thus, consider the text input "Dr. Smith lives at 1702 S. Park Drive and can be reached by phone at 555-456-7890." The first instance of "Dr." must be converted to the word "doctor," while the second instance should be replaced with the word "drive." Given that 1702 S. Park Drive appears to be an address, a likely rendering would be "seventeen oh two south park drive." The final phone number would be replaced with the words "five five five, four five six, seven eight nine oh," with commas or other textual markers to indicate the appropriate phrasing for a phone number. Of course, the challenge for text normalization is to derive enough information of the textual input to make accurate guesses about things like phone numbers or addresses.

A related problem for text normalization is disambiguating the pronunciation of homophonous words. Often, context can provide helpful clues; if someone is "playing a bass," they are more likely to be a musician than an actor impersonating a fish. But sometimes disambiguation requires much deeper semantic/pragmatic knowledge that can easily be guessed from context alone. Is a shiny white bow a holiday decoration or the front of a boat?

The tokenized and normalized input text, along with any additional meta information related to prosodic properties (the intonation and timing properties) derived from the initial text processing, is next passed to the *text to phonetics* component (**Figure 1**, *green*), which produces a symbolic phonetic representation. In the original rule-based formant synthesis systems like DECtalk, this representation consisted of little more than a string of phoneme symbols along with some formal boundary and intonation symbols. Boundary symbols indicate the degree of acoustic/phonetic separation between two adjacent phonemes. For example, the boundaries between words are often marked by distinct acoustic features; consider the distinction between "gray day" and "grade A." Moreover, the boundaries between phrases of different types are also marked by phonetic duration differences, pauses, and intonational features such as the rising pitch at the end of many questions or the falling pitch at the end of a declarative sentence.

The intonation symbols express the relative locations and types of pitch accents or "tones" relative to the phonetic symbols. Over time, a standardized system has developed based on the concepts of "tones and break indices" (ToBI; e.g., Silverman et al., 1992) that describes the intonational structure of English and other languages in terms of a discrete set of tones corresponding to a relative maximum or minimum in fundamental frequency

(perceived as voice pitch) that aligns to a specific syllable within an utterance. Similarly, break indices are single-digit integers that indicate the relative separation between two elements in an utterance. ToBI-like symbol sets are often used for the boundary and intonation symbols in current TTS systems.

Next, the *phonetics to parameters* components (**Figure 1**, *green*) maps the symbolic phonetic description of the input text to a numerical representation suitable for input to a vocoder or parametric synthesizer to generate a speech waveform from the numerical parameter values. Whereas the phonetic symbols imply a sequence of related acoustic events, there are no time units at the symbolic level. In a rule-based formant synthesizer like DECtalk, the phonetics to parameters component is responsible for laying out the parameters as a dynamic time-varying sequence with defined temporal coordinates. Typically, parameters are updated every few milliseconds at a constant prespecified rate, for example, every five milliseconds.

Finally, the *parameters to sound* component (**Figure 1**, *green*), often referred to as a "vocoder," accepts the parametric representation of speech and generates audio output. In many parametric systems, a source/filter model of speech is adopted wherein a source signal consisting of either a periodic impulse train or white noise is passed through a digital filter representing the human vocal tract.

Application of Text to Speech to Speech-Generating Devices

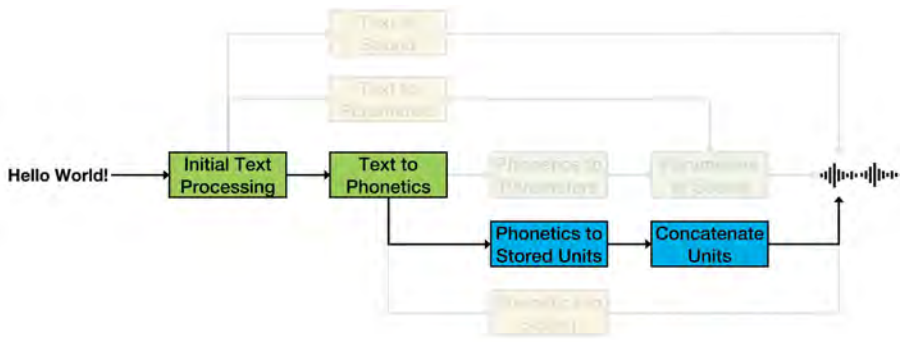
Formant-based TTS systems were intelligible enough to become widely adopted by assisted communicators in the late 1980s and 1990s, with DECtalk being the most commonly used system in the SGDs of the time (see https://bit.ly/31E9A54). Perfect Paul, which was demonstrably the most intelligible of the DECtalk voices (Green et al., 1986), was the voice of choice for many AAC users of the time. Even women would often choose to use the male Perfect Paul voice because it was more easily understood by others. Imagine attending a meeting in a conference room with multiple people using SGDs all tuned to Perfect Paul and not being entirely certain whose device had just emitted an important comment! So, although many nonvocal persons now had a voice, they did not have their *own* voice for communication.

In addition to not providing every AAC user with a unique voice, the formant synthesis systems of the time did not sound particularly human. As I discuss in **Diphone Synthesis**, a technique called diphone synthesis emerged as one possible way to generate more human-sounding and identity-bearing synthetic speech. But neither formant synthesis nor diphone synthesis addressed another shortcoming, a lack of expressiveness. Attempts were made to create a more expressive output for DECtalk by modifying the synthesis parameters to convey emotional states such as boredom or sadness (Murray and Arnott, 1993), but they were not widely implemented.

Diphone Synthesis

Diphone systems represented an important bifurcation in TTS technology: the distinction between knowledge-based systems and data-based systems. This distinction can also be described as between rule-based systems where a human expert must design the rules and corpus-based systems where a corpus of speech data provides the

Figure 2. *Component pipeline for diphone and other concatenative synthesis methods from* ***Figure 1****.*



information that would otherwise need to be expanded from rules. Or, as seen in **Statistical Parametric Speech Synthesis**, the corpus can be used to automatically discover the rules through machine-learning algorithms so that no expert is needed. Thus, the rules needed for the phonetics to parameters component of a formant synthesis system required expert knowledge of acoustic phonetics and a lot of hard work. However, corpus-based systems were able to replace much of that work by simply storing the data that would otherwise need to be developed from rules.

As illustrated in **Figure 2**, diphone synthesis (and related "concatenative" methods) follows a slightly different path within our overall TTS model.

A diphone is the region of speech spanning roughly the middle of one phoneme to the middle of the next phoneme. **Figure 3** illustrates this using the word "bob." The initial and final /b/ segments are relatively stable as is the /a/ vowel near its center. However, the acoustic structure changes rapidly around the borders between the consonants and the vowel. As long as the phoneme centers are reasonably similar across different phonetic contexts (they really are not, but we are assuming that they are close enough!), then cutting speech up into diphone-sized units ought to allow one to concatenate the diphones in novel ways to produce nearly any utterance. For example, take the [ba] from [bab] and the [at] from "cot" [kat] to create "bought" [bat]. This was the insight that led Dixon and Maxey (1968) to develop a formant diphone synthesizer (see #18 at https://bit.ly/3qxs3uL) that used stored formant synthesis parameters rather that a rule system to generate the parameters prior to synthesis.

Formant synthesis parameters are an interesting choice for the diphone storage because they have several useful properties. (1) They do not require a large amount of storage (a factor that was especially important in 1968!). (2) They are orthogonal, that is, it is possible to change any one parameter value without impacting the values of other parameters. (3) Interpolation between values for any parameter will yield another valid parameter value.

However, formant synthesis parameter values have not been the most common format for storing diphone units. More commonly, diphones have been stored as linear predictive coding (LPC) coefficients (e.g., see #34 at https://bit.ly/30n0V6V) or as waveform data stored in a format amenable to the fundamental frequency (F0) and duration modification using an algorithm like Pitch Synchronous OverLap Add (PSOLA; Moulines and Charpentier, 1990).

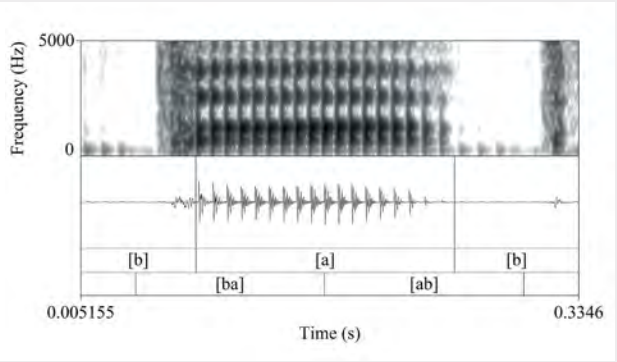


Figure 3. *Illustration of phonemes versus diphones.* ***Top****, spectrogram of the word bob.* ***Dark bands****, regions of high energy, corresponding to formants.* ***Middle****, acoustic waveform.* ***Bar*** *below waveform, phoneme locations ([b], [a], and [b]).* ***Bottom bar****, locations of the two diphone regions ([ba] and [ab]).*

As is often true with speech processing, the most natural sounding of these formats in terms of voice quality would be waveform data because that is the least processed. LPC coding preserves much of the speaker identity information, but some voice quality may be lost in processing. Formant synthesis generally produces the least natural-sounding audio. Unfortunately, waveform data are the least compact storage format and also the most difficult to work with in that they afford little opportunity to adjust for discontinuities at diphone boundaries.

The *phonetics to stored units* (**Figure 2**, *blue*) is the path taken from the text to phonetics component for diphone synthesis. There are a relatively small number of diphones for any language. For example, Dixon and Maxey (1968) based their inventory on a total of 41 phonemes, so a theoretical maximum of $41^2 = 1{,}681$ possible diphones. Consequently, the conversion from phonetics to stored units amounts to simply looking up the needed sequence of diphone units.

The selected diphone units can then be passed to the *concatenate units* (**Figure 2**, *blue*) component that concatenates

the selected units to form the desired output utterance. If the storage format permits, there may be additional adjustments to the units during the concatenating process. This could include adjustments such as smoothing potential discontinuities across diphone boundaries, adjusting diphone duration per a timing model, or even adjusting the F0 per an intonation model. Once the diphones have been assembled and concatenated to form an utterance, additional processing, if any, is applied to map from the diphone storage format to a digital audio waveform.

Diphone synthesis held one particularly intriguing possibility for SGD users, the ability to capture an individual's vocal identity. Because only a small amount of recorded speech is needed to create a diphone inventory, it would be possible to inexpensively mass produce unique diphone voices as long as the process of selecting diphones from recordings could be automated. People using SGDs could have a unique personal voice by selecting a suitable voice donor to do the recording. Moreover, people diagnosed with a condition such as ALS that threatens the loss of their voice could do the recording themselves and thus "bank" their voice for later use as a synthetic voice in an AAC device. In the mid-1990s, my laboratory at the Nemours Children's Hospital, Delaware, began experimenting with an extension of diphone synthesis (e.g., Bunnell et al., 1998) that would allow ALS patients to bank their voice in this way, a process referred to as "voice banking."

Diphone TTS voices, although a promising technology, did not generally gain much traction among AAC device manufacturers or SGD users. The small memory footprint for rule-based formant synthesis was certainly an important factor in favor of the formant-based TTS voices for AAC manufacturers. Furthermore, diphone TTS voices did capture the vocal identity of the person who recorded the diphone inventory but did not permit expressiveness, particularly for systems that used waveform concatenation, and despite capturing voice quality well, diphone synthesis tended not to flow in a natural manner. Moreover, many of the inexpensive diphone TTS systems available in the 1980s and later were less pleasing to listen to than the DECtalk voices that were provided with most AAC devices (e.g., see #29 at https://bit.ly/30n0V6V). That changed, however, with the emergence of unit selection TTS systems in the 1990s.

Unit Selection Text-to-Speech Voices

One of the greatest difficulties with diphone synthesis was the impossibility of selecting a collection of diphones that did not suffer from sometimes jarring discontinuities at concatenation boundaries. This was less of an issue for diphones stored, as per Dixon and Maxey (1968), in a format that was amenable to substantial adjustments to smooth over or entirely eliminate disjunction by interpolating smoother parameter trajectories at segment boundaries. However, the highest voice quality obtainable from diphone synthesis was for diphones stored as waveform data or equivalently prewindowed PSOLA epochs. Unfortunately, with waveform concatenation and other issues, notably jarring differences in spectral features, F0, and amplitude at diphone boundaries were common.

These issues with waveform concatenation were largely addressed by an extended approach called "unit selection" (e.g., Zen et al., 2009) wherein a large amount of speech from a single individual is recorded and segmented into units that could be diphone size or smaller. This approach is illustrated in **Figure 4** using the word *two* as the target utterance and assuming each unit is roughly half of a phoneme. The units are stored along with additional features describing the linguistic details of the phoneme or waveform region from which they were drawn, such as the type of word (function vs. content word), syllable stress, syllable location, phrase location, presence and type of pitch accent on the associated syllable, and boundary level for the associated syllable. Because a unit selection database may contain a large number of candidates for each possible unit, there is a much greater chance of finding one or more units that exactly or nearly match the intended output context along all of the coded linguistic dimensions. Moreover, in the process of selecting units for concatenation, it is possible to select the specific candidates that will also minimize spectral discontinuities or sudden jumps in F0 or other factors that cannot be indexed as specific linguistic features.

Unit selection voices came to dominate the commercial TTS voice market in the late 1990s and 2000s because they are much more natural-sounding and intelligible than other commercially available TTS voices. Sometime in the 2000s, most SGD manufacturers included at least a few unit selection voices in their products. Moreover,

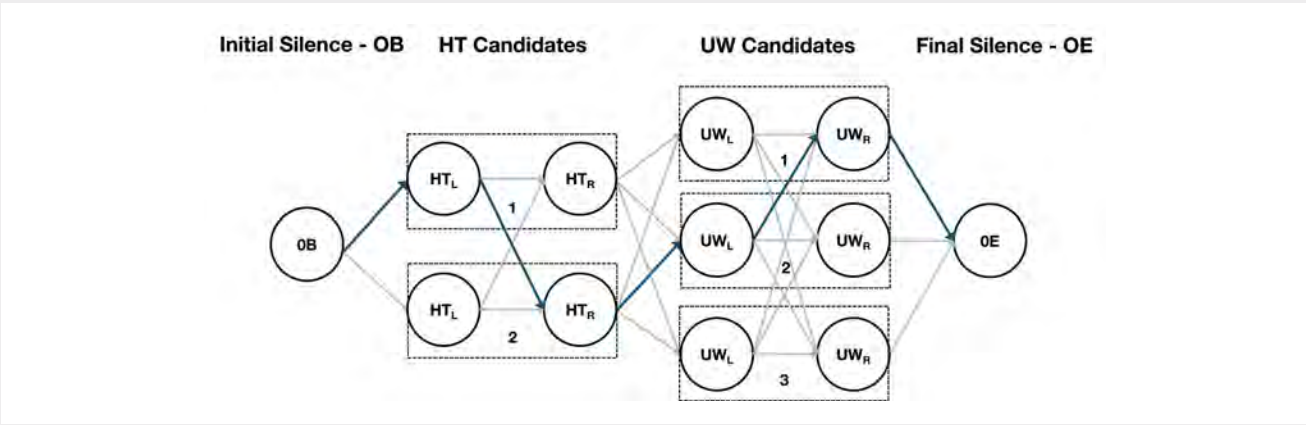


Figure 4. *Unit selection search process for the word "two." Two phonemes are required: /t/ (HT) and /u/ (UW) along with initial and final silence pseudo phonemes (0B and 0E). Multiple instances of each phoneme (**numbers** in boxes) are selected, each of which has two subphonemic "units" (e.g., HT_L and HT_R). Each unit receives a target cost based on linguistic appropriateness and joined costs are assigned between units based on the acoustic continuity (**gray arrows**). The search locates the specific candidate units that minimize the combined target and joined costs over the utterance (paths shown with **blue arrows**).*

most SGDs transitioned from proprietary hardware to being software running on embedded Microsoft Windows systems. Because of this, most SGDs were also able to include voices provided by Microsoft or third-party voices written to published Microsoft standards.

My laboratory moved to a full unit selection system for voice bankers based on 1,600 utterances of various lengths and composition, comprising roughly one hour of running speech at normal speaking rates. With funding from the National Institute for Disability and Rehabilitation Research and later from the National Institutes of Health (NIH), I was able to offer a free experimental voice-banking service and provided a small number of voices to participants throughout most of the 2000s. Voices built in the laboratory could be incorporated with any Windows-based SGD. I formally began referring to the service as the ModelTalker project (Bunnell et al., 2005). Although the ModelTalker service was the first such service regularly used by ALS patients for voice banking, there are now excellent voice-banking services offered by a variety of commercial TTS companies, notably Acapela.com and Cereproc.com, who also offer voices for languages other than English. I have live example voices on the ModelTalker.org website (see https://bit.ly/3C57WpT; it might be slow when the website is busy).

By the late 2000s, unit selection was considered the best available TTS technology. The major voices for services like Siri and Alexa were built on unit selection technology as were enterprise-grade voices for large business call centers. However, the amount of recorded speech from voice talent needed to create the highest quality general-use voices exceeded tens of hours of running speech and many more hours of studio time. Even then, it is fairly easy to find examples of words that did not sound entirely natural within some specific context. There is no way to anticipate and record all of the possible acoustic phonetic variation within any language, even if factors like vocal effort, voice quality (breathy, hoarse, modal, fry, pressed), speaking rate, articulatory precision, and so forth are held constant. Moreover, for a truly natural-sounding and expressive TTS voice, one would not want to hold those factors constant!

The massive increase in memory density and decrease in memory cost over several decades made it feasible to work with unit selection voices despite their rapidly growing data footprint. But no amount of memory is really able to overcome the combinatorial ceiling that unit selection voices ultimately must hit. This prompted much interest in the possibility of returning to parametric synthesis, but rather than parametric synthesis with expertly crafted rules to describe dynamic parameter variation, statistical machine-learning techniques could be used to automatically capture the temporal patterning in synthesis parameters. The improvements brought by this effort to synthesize speech are now discussed.

Statistical Parametric Speech Synthesis

As with unit selection synthesis, statistical parametric speech synthesis (SPSS)(Zen et al., 2009) requires a substantial corpus of speech data to be used in training its parametric phonetic models. Unlike unit selection synthesis, once the training process is completed, however, the original speech waveform data are no longer needed. Instead, the SPSS machine-learning process develops models for the acoustic structure of each phoneme. These models are then able to generate the time-varying parameters values for the parameters to sound component of the TTS system. Thus, fully trained SPSS models replace hand-coded rule systems in the phonetics to parameters component in **Figure 1**. In practice, the SPSS models are commonly sets of hidden Markov models (HMMs), one model for each phoneme, that describe the acoustic structure of the phoneme as a sequence of acoustic states, allowing the time-varying trajectories of parameters to be regenerated from the properties of the state sequence. The parameters the SPSS models learn are typically those describing the time-varying speech source function (voicing or friction) and moment-to-moment spectral features. The parameters to sound or vocoder component then uses the source and spectral parameters to regenerate audio data via digital filtering.

SPSS synthesis has several advantages over both rule-based formant synthesis and unit selection. First, because the SPSS models for parameter generation can be trained on a corpus of speech from a single talker, the output of the SPSS voice sounds recognizably like the talker who recorded the corpus. Moreover, because the training process is largely automatic, building multiple personal voices is not especially difficult or labor intensive. Compared with unit selection based on a similar-size speech corpus, particularly for smaller corpora (those having less than four hours of running speech), SPSS voices are not prone to discontinuities at segment boundaries and tend to have more natural-sounding prosodic structure. And because SPSS voices use parametric synthesis, it has the potential for changing characteristics of the voice quality or introducing expressiveness, but this potential is not yet realized.

There are, however, two main drawbacks to SPSS voices. First, the naturalness of the resulting synthetic voice is limited by the ability of the vocoder to reproduce natural-sounding voice quality. Some vocoder output sounds "buzzy" or "mechanical" when compared with unit selection voice quality. Second, in SPSS, each phonetic model represents an average of the acoustic patterns seen for all instances of the same contextually similar phonetic segment. This averaging tends to obscure some of the natural variability in human speech, leading to more monotonous sounding speech. Often, SPSS systems attempt to compensate for this averaging effect by exaggerating or boosting the variability of parameters over time. However, once the natural variability is lost due to averaging, it is not really possible to restore it.

Despite these two drawbacks, ACC users of ModelTalker voices have generally had favorable reactions to SPSS voices and the best of the SPSS laboratory TTS systems have been able to produce speech with audio quality closely approaching that of unit selection systems. Any long-term debate about the relative merits of unit selection versus SPSS voices, however, appears to rapidly becoming moot, particularly as it applies to large commercial grade TTS voices. This is due to the emergence of new deep-learning models.

Deep Neural Network Speech Synthesis

In the past decade, deep neural networks (DNNs) and deep learning have revolutionized machine learning and led to large-scale improvements in several application areas. Large improvements have been observed in areas as diverse as speech recognition, machine translation between languages, natural language processing, text summarization, and speech synthesis. Explaining, even grossly, how DNNs function is beyond the scope of this article, but a few examples and consideration of how some models are changing the flow within the TTS system framework shown in **Figure 5** may give a reasonable sense of the emerging changes.

In **Figure 5**, the path from text to phonetics through phonetics to sound is a good place to start because this is the path used by WaveNet (van den Oord et al., 2016), which was one of the first "end-to-end" neural TTS systems. The authors have created an excellent website that describes their work and provides audio examples (see https://bit.ly/3qtNrkm). Training for WaveNet required about 25 hours of speech from a single female speaker and required days of CPU and GPU processing on Google's servers.

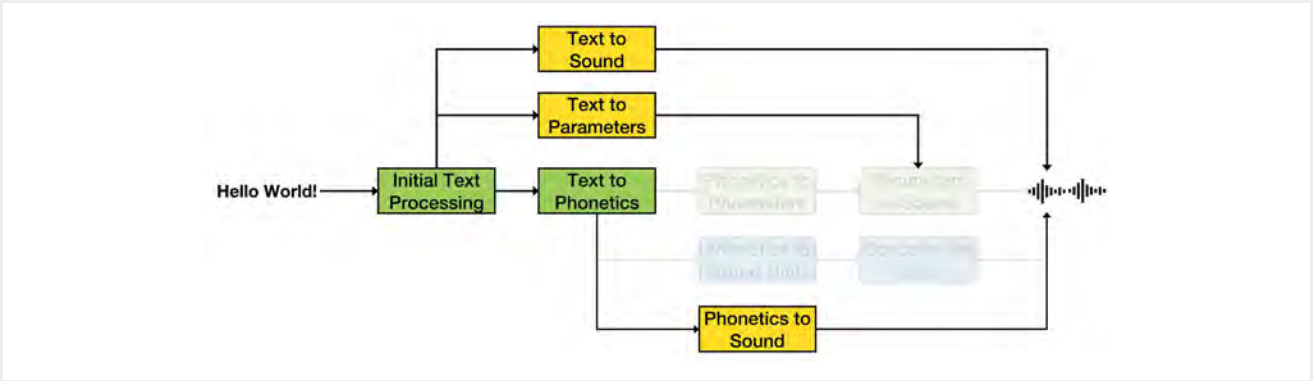


Figure 5. *Deep neural network (DNN) TTS pipelines emerging in current research efforts from* ***Figure 1****.*

A large number of current end-to-end neural TTS systems follow the path from initial text processing through text to parameters and thereafter to a parameters to sound component. In some cases, "text" is taken somewhat broadly to refer to both literal words or characters, or to a form in which standard word spellings are replaced with something like International Phonetic Alphabet (IPA) characters to resolve letter to sound ambiguity. This is particularly helpful for languages like English that have borrowed words from many other languages and also helps when building multitalker and multilanguage systems. Most systems on this path generate Mel-scaled spectrograms as the output of the text to parameters component, relying on one of several vocoder methods (e.g., Griffin and Lim, 1984) or DNN-based vocoders, for generating audio output from the Mel-scaled spectrograms without explicitly applying a source/filter model. (Note: the Mel scale is a perceptually motivated transformation of linear frequency to a scale with approximately equal pitch steps; see Stevens et al., 1937.) However, a few systems may also generate parameters for alternative vocoders such as the *WORLD* vocoder (Morise et al., 2016). Although no systems are presently doing this, output in terms of formant synthesis parameters is also conceivable, with the final parameters to sound component being a formant synthesis vocoder.

Finally, as the ultimate end-to-end DNN TTS approach there is the path from initial text processing through TTS directly to audio output. This is a system referred to as end-to-end adversarial TTS (EATS) by Donahue et al. (2020; see https://bit.ly/3wpQBGR for audio examples). There is nothing before the audio generation but a light text-processing stage to handle tokenization and text normalization, perhaps with an additional substitution of IPA word spellings instead of standard word spellings. The system is complex and requires a very large data corpus and much computer time to train, but their examples illustrate output that is virtually indistinguishable from human speech. Unfortunately, expressiveness remains a challenge for this technology. Neural TTS systems can learn to express anything that is present in their training data but generalizing beyond seen expressive modes is an area of active ongoing research (e.g., Skerry-Ryan et al., 2018; see examples at https://bit.ly/30epgeW).

Neural TTS systems come at substantial expense both in terms of the amount of data that is needed and in the computational resources to train the models. Many are currently so resource heavy that they are only usable by well-equipped industry or university laboratories. However, there are elements of this work that are already having an impact, notably the neural vocoder programs, which produce highly natural-sounding speech output given the correct input. It may take a very large amount of data and heavy server load to train these vocoders, but once trained, they can be used with Mel spectrograms generated by many other applications and are able to run in real time on desktop-grade computers.

Conclusions

The path from rule-based formant synthesis in the 1980s to the DNN voices being studied in research laboratories today represents significant growth in TTS technology. This growth has been followed through the lens of how the improvements impact one of the potentially most exciting applications of TTS technology: its potential to provide unique personal voices for people who are unable to communicate vocally without assistance. A notable subset of the potential users of TTS technology are those whose

speech is at risk of being lost due to disease or injury. For those users, the ability to bank their existing speech for its use later in as a personal TTS voice of the quality now emerging form the laboratory is a highly promising prospect.

We initially identified four features that seem to be of greatest importance to users for assistive voice technology: intelligibility, naturalness, identity, and expressivity. Of these four, the first three are essentially solved problems, at least for laboratory-grade neural TTS systems. Given the rate of progress with the technology, it seems likely that for these three features, medical and consumer applications will not be long in coming. Expressivity, however, remains the largest unsolved issue for TTS systems. Parametric synthesis affords the ability to control features known to relate to expressive modes of speaking, and it will be fascinating to see how natural language processing (NLP) may end up helping users quickly find the right emotion to convey along with their text when it is spoken aloud.

References

Bunnell, H. T., Hoskins, S., and Yarrington, D. (1998). A biphone constrained concatenation method for diphone synthesis. *Proceedings of the Third International Workshop on Speech Synthesis*, Jenolan Caves, Blue Mountains, NSW, Australia, November 26-29, 1998, pp. 171-176.

Bunnell, H. T., Pennington, C., Yarrington, D., and Gray, J. (2005). Automatic personal synthetic voice construction. *Proceedings of the Eurospeech 2005*, Lisbon, Portugal, September 4-8, pp. 89-92.

Donahue, J., Dieleman, S., Bińkowski, M., Elsen, E., and Simonyan, K. (2020). End-to-end adversarial text-to-speech. Available at https://bit.ly/3C7jVmI. Accessed November 12, 2021.

Greene, B. G., Logan, J. S., and Pisoni, D. B. (1986). Perception of synthetic speech produced automatically by rule: Intelligibility of eight text-to-speech systems. *Behavior Research Methods, Instruments, & Computers* 18(2), 100-107.

Griffin, D., and Lim J. (1984). Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech and Signal Processing* 32(2), 236-243. https://doi.org/10.1109/TASSP.1984.1164317.

Klatt, D. H. (1980). Software for a cascade/parallel synthesizer. *The Journal of the Acoustical Society of America* 67, 971. https://doi.org/10.1121/1.38940.

Klatt, D. H. (1987). Review of text-to-speech conversion for English. *The Journal of the Acoustical Society of America* 82, 737-793.

Morise, M., Yokomori, F., and Ozawa, K. (2016). WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems* E99-D(7), 1877-1884.

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

Murray, I. R., and Arnott, J. L. (1993). Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion. *The Journal of the Acoustical Society of America* 93(2), 1097-1108.

Pullin, G., and Hennig, S. (2015). 17 ways to say yes: Toward nuanced tone of voice in AAC and speech technology. *Augmentative and Alternative Communication* 31(2), 170-180.

Ramsay, G. (2019). Mechanical speech synthesis in early talking automata. *Acoustics Today* 15(2), 11-19.

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., and Hirschberg, J. (1992). ToBI: A standard for labeling English prosody. *Proceedings of the 2nd International Conference Spoken Language Processing*, Banff, AB, Canada, October 13-16, 1992, pp. 867-870.

Skerry-Ryan, R. J., Battenberg, E., Xiao, Y., Wang, Y., Stanton, D., Shor, J., Weiss, R., Clark, R., and Saurous, R. A. (2018). Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. *Proceedings of the International Conference on Machine Learning*, Stockholm, Sweden, July 10-15, 2018. Available at https://bit.ly/3CgXvPU. Accessed November 12, 2021, pp. 7471-7480.

Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America* 8(3), 185-190.

van den Oord, A. Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*. Available at https://bit.ly/3qtNrkm. Accessed November 12, 2021.

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

About the Author

H. Timothy Bunnell
tim.bunnell@nemours.org

Nemours Children's Hospital, Delaware
Center for Pediatric Auditory and Speech Sciences
1701 Rockland Road
Wilmington, Delaware 19803, USA

H. Timothy Bunnell is the director of the Center for Pediatric Auditory and Speech Sciences (CPASS) at the Nemours Children's Hospital, Delaware, Wilmington; head of the Speech Research Lab in the CPASS; and an adjunct professor of Computer and Information Sciences at the University of Delaware, Newark. He received his PhD in experimental psychology in 1983 from The Pennsylvania State University, University Park; served as research scientist at Gallaudet University, Washington, DC, from 1983 to 1989; and joined Nemours Children's Health to found the Speech Research Laboratory in 1989. His research has focused on the applications of speech technology for children with hearing and speech disorders.