EURASIP Journal on Audio, Speech, and Music Processing

REVIEW

Open Access

Review of methods for coding of speech signals

Douglas O'Shaughnessy*

Abstract

Speech is the most common form of human communication, and many conversations use digital communication links. For efficient transmission, acoustic speech waveforms are usually converted to digital form, with reduced bit rates, while maintaining decoded speech quality. This paper reviews the history of speech coding techniques, from early mu-law logarithmic compression to recent neural-network methods. The techniques are examined in terms of output quality, algorithmic complexity, delay, and cost. Focus is on which aspects of speech can be exploited for high-quality transmission. The choices made to code speech are motivated by efficiency, the needs of applications, and access to information in the speech signal that is useful for both intelligibility and naturalness in the reconstructed speech at the decoder.

1 Introduction

A fundamental aspect of human nature is the need to communicate with each other, and speaking and listening provide the most common ways to convey information. This paper examines the technical methods that have been used to facilitate this communication over common transmission links. Issues of interest include the following: (1) the precision needed to quantify relevant speech signal parameters in the time and frequency domains, (2) estimating which aspects of the input speech are intentionally controlled by speakers and utilized by listeners, (3) how to accommodate the huge variability in speech communication due to environmental conditions and different speakers, (4) how to harness the power of computers, and (5) how to design coders efficiently.

Speech coding has two parts: *coder* for analysis of the input and *decoder* to synthesize or reconstruct the output speech; overall systems are called *codecs*. A continuous input speech signal is analyzed and transformed into a bit sequence, which can be stored or transmitted over a communication channel. The decoder converts received bits into an analog reconstructed speech signal, suitable for listening. Common bit rates are 64 kilobits/s (kbps) for many landline networks and 10–13 kbps for cellular telephony. One usually distinguishes *narrowband* transmission (e.g., landline telephony at 8 kHz sampling rate) vs. *wideband* (16 kHz, for Internet applications). While speech has some content above 8 kHz, high-quality entertainment systems are the main applications for rates above 16 kHz; for example, compact disks use 44.1 kHz and some modern coders half that rate. For comparison, the information content in speech is much lower, at about 100 bps [131], which suggests that further improvements to coding are surely feasible. This paper examines how coders function, focusing on properties of speech signals that can be exploited for speech coding.

To minimize transmission rates, coders typically use statistical models to reduce redundancy in signals, often exploiting signal correlations in time and frequency. Some random signals such as white noise have no correlations to exploit. Speech, on the other hand, is clearly redundant, as human acoustic communication functions in diverse conditions, e.g., noisy environments, unfamiliar people, and foreign accents. As sources of redundancy, note that natural speech consists of periodic bursts or noise exciting a speaker's *vocal tract* (VT),

*Correspondence:
Douglas O'Shaughnessy
dougo@emt.inrs.ca
INRS-EMT, Montreal, Canada

which can be modelled as a set of short and narrow tubes [21]. These physical conditions and constraints lead to many signal redundancies. For speech, VT shape changes slowly relative to signal sampling rates, often yielding successive brief portions of speech, called *pitch periods*, which are caused by periodic closures of the speaker's vocal cords and have strong correlation with each other. The acoustics of narrow tubes causes resonances spaced approximately 1 kHz apart for a typical VT of length 17 cm. This is related to the speed of sound, approximately 340 m/s in typical conditions, and to energy reinforcement when the VT acts as a quarter- or half-wavelength resonator [15].

1.1 Types of speech coders: waveform and vocoder

Traditionally, there have been two major approaches to speech coding: waveform matching and parametric *vocoder* ("voice coder"). The first replicates the real-valued time waveform, or its complex spectrum, directly, sample by sample, often based on some form of a minimum average distance criterion, such as *signal-to-noise ratio* (SNR), between the coder input and decoder output. Such an approach can apply to many signals, not just speech, although the choice of the distance measure may be weighted for speech applications, to exploit the nonuniform time-frequency resolution of the human auditory system. Neural network speech coders are waveform coders.

Vocoders, on the other hand, exploit a model of human speech production and perception, to greatly reduce bit rate, by representing dynamic speech information at a much lower *frame* rate (e.g., 100 Hz) than the *sampling* rate (e.g., 8–16 kHz) of speech. Like most analog signals, speech is sampled uniformly, which preserves the low-pass spectrum up to half the sampling rate (Nyquist rate [120]). However, as speech results from VT motion that is relatively slow compared to the sampling rate, its information evolves slowly, allowing coding representation in vocoders in terms of frames, which are short-time sections of 10–30 ms. Vocoders usually sacrifice some output speech quality by representing VT spectra and excitation simply, leading to loss of natural spectral details, while retaining intelligibility.

A traditional speech codec often relies on signal processing pipelines and specific design choices that exploit in-domain knowledge of psychoacoustics, to reduce coding bit rate and/or computational complexity. Vocoders rely on the specific assumption that the source audio is speech and introduce strong priors in the form of a parametric model. Unlike waveform codecs, the vocoding goal is not to obtain a faithful reconstruction sample by sample but to generate audio perceptually similar to the original. Vocoders date to the 1950s [20], exploiting a *source-filter model* of an excitation of noise or periodic pulses, with a VT filter [21]. Vocoders were refined to model VT resonances (called *formants*) [105], as well as using a more advanced model of spectral phase [24]. The common *linear predictive coders* (LPC) are discussed in detail in Section 4.4. Examples of modern vocoders are STRAIGHT [49] and WORLD [86]; WORLD uses hundreds of spectral and aperiodic features.

For speech, both the sampling rate and the frame rate use a uniform temporal representation of the dynamic speech signal [29], but the sampling rate is proportional to the desired spectral bandwidth, whereas the frame rate selects useful data corresponding to VT motion. A typical sampling rate is 8 kHz, e.g., for narrowband telephony, as it has little energy above 3.2 kHz; other applications have higher rates, e.g., 16, 32, and 48 kHz (wideband, super wideband, full band, respectively; e.g., G.729 standard [101];). A frame rate of 100 Hz is a compromise that corresponds to the much slower VT motion, using about eight updates per phoneme, as phonemes in speech average approximately 80 ms. This 100-Hz rate is common in most speech applications, including *automatic speech recognition* (ASR) [135] and text-to-speech synthesis [126]. As for bit rates, most communication systems employ a fixed rate, but *hierarchical coders* allow the user to select a suitable decoding rate, depending on available downloading bandwidth. These coders have different layers of information, where the lowest rate allows basic intelligible reconstructed speech, whereas use of higher decoding rates employs more layers of data available for higher quality.

Coders may use statistics to accommodate the huge variability in speech, i.e., across speakers, acoustic conditions, and speaking contexts. A sampled speech signal s_i is a discrete-time stochastic process, often characterized by a conditional *probability density function* (pdf) $f(s_i \,|s_{i-1}\,,\, s_{i-2}\,,\, \cdot\,\cdot\,\cdot\, s_{i-N})$. Successive samples in speech can have significant correlation over very large history N, e.g., several seconds of speech, but many applications limit analysis to a range of 10 samples, for efficiency. For example, LPC typically examines a range of $N = 10$ samples in telephony [79], as spectral envelope detail for speech averages one resonance per kHz to model (two real parameters can represent the center frequency and bandwidth of each resonance); this allows detailed spectral amplitude (all-pole) models of order 10. Recent powerful *artificial neural networks* (ANNs) have found ways to exploit much larger ranges of N, as discussed in Section 12.

Vocoders use a model for human speech production; the most common is the source-filter model, where speech is interpreted as the output of a VT filter, and the input can be an excitation of two or three types: (1)

quasiperiodic puffs of air from the lungs at the glottis, (2) steady airflow noise at the outlet of a constriction in the VT, or (3) bursts of noise at a periodic rate. Basic LPC uses a binary choice for excitation, either pulses or noise, avoiding the need to distinguish the 2nd and 3rd types; in text-to-speech synthesis applications, such classification is simple from text analysis. The periodicity of the (most common) first type of speech is the *fundamental frequency* (F0), the rate at which vocal cords vibrate. F0 is a very important aspect of periodic (*voiced*) speech, as all harmonics are multiples of F0, which allows very efficient coding of excitation.

The choice of vocoder model depends upon those aspects of speech that are more relevant than others and that can be exploited for efficient coding. Waveform coders, on the other hand, transmit data for each sample, without needing explicit decisions as to which aspects are more pertinent. To identify details in speech to focus on for coding, one may fruitfully seek aspects that are both intentionally controlled by speakers for communicative purposes and readily utilized by listeners. Most speech models exploit slowly varying VT resonances and F0, as these are very efficient representations. By estimation of parameters of VT models (e.g., LPC all-pole models) and of F0 every speech frame, one can transmit speech data at low bit rates.

1.2 Which acoustic features are important for speech coding?

Speech has much acoustic detail that is not under direct speaker control, owing to complex interactions in airflow and energy losses (friction, thermal) [21]. Much minor acoustic detail is incidental, unintended by speakers, and ignored by listeners, without affecting distinctions of phonetic content or speaker aspects. Most speech applications, including coders, focus instead on spectral amplitude, which is directly controlled by VT positioning, and place a lesser role for spectral phase [93]. There is much variability in phase in speech; as phase is not intentionally controlled by speakers, listeners pay attention primarily to frequency locations of spectral peaks. As a result, many vocoders emulate this behavior and focus on spectral envelope detail, especially for low-rate coding. To increase quality, speech applications that can afford higher bit rates include more phase detail, which can have little effect on intelligibility but is important for naturalness.

Two waveforms with the same amplitude but different phase may sound the same or different to listeners. Preserving phase detail is a simple and safe strategy, followed in many waveform coders, but it is inherently inefficient. Low-rate coders aim for intelligible decoded speech, while high-rate coders have increased quality. To achieve lower bit rates in speech coding may require a better understanding of phase. Currently, coders often approximate aspects of voiced excitation with thousands of bits/s in algebraic vector quantization or via use of millions of neural network parameters. Lack of understanding of VT excitation and phase has limited more efficient coding of speech.

Besides phase modelling, low-rate vocoders often are weak in preserving speaker information, e.g., identity for speaker verification and health status. Speech coders tend to replicate some speaker-specific data, e.g., that related to timing and pitch, but there is much unknown about how speech encodes much of data about speaker identity. Waveform coders inherently often preserve this information, but without any explicit understanding.

If one looks to ways that are currently used to identify speakers as a guide for how speech coders might preserve relevant speaker data, one may be disappointed. Speaker verification has often used massive statistical approaches to distinguish speakers, via methods such as eigenvoices, *x*-vectors, principal components analysis, Gaussian mixture models, and joint factor analysis [50]. Such optimization methods have not found their way yet into mainstream speech coders; they appear to be useful for classification but less so for the objectives of speech coding. The well-known phonetic features of F0 and spectral envelopes have been greatly exploited in speech coding but seem to have less benefit in distinguishing similar speakers. For emotional and health indicators in speech, the literature suggests that intonation is a prime factor [64], and most coders preserve intonation well.

1.3 Evaluation criteria for speech coding

The success of a speech coder is often measured in terms of its bit rate and complexity but mostly by the *intelligibility* and *quality* of the decoded speech. Both these characteristics are most reliably evaluated by *perceptual* experiments, asking listeners for their judgments. As such is costly and time-consuming; *objective* methods have been sought to estimate speech quality, e.g., perceptual evaluation of speech quality [40, 104], short-time objective intelligibility [124], and ITU-T P.863 [2]. One *subjective* measures are the MUltiple Stimuli with Hidden Reference and Anchor test [43]; the common *mean opinion score* is typically above 4 for good waveform coders and above 3 for vocoders [121].

Intelligibility and quality are highly correlated but different. For example, one simplistic way to eliminate much noise in distorted speech is to filter out low frequencies (where most environmental noise occurs), which makes speech sound less noisy, but often loses phonetic information critical for intelligibility. While intelligibility is readily measured by *word error rate*, the percentage of

words misunderstood, speech quality is far more complex. Noise, distortion, naturalness, reverberation, and intelligibility all affect the perceived quality.

When estimating output speech quality, one traditional (but limited) objective measure for waveform coders has been SNR [58], which treats all time and frequency samples equally, with the average squared difference between the input and reproduced signals as criterion. This is simple and commonly used but is an imperfect measure for speech quality, as it ignores many relevant nonlinear perceptual phenomena, which include masking, the mel scale [13], and other logarithmic effects in perception. Some coders weight SNR so that the coding noise is shaped to exploit masking properties of human perception, e.g., "hide" quantization noise in frequency ranges of speech where speech is strong, such as harmonics in formants [9]. Nonetheless, many waveform coders explicitly use maximal SNR as their estimate of success, by replicating either the speech time waveform or its spectrum, sample by sample. SNR is not pertinent for vocoders, as their reproduction is not sample by sample; thus, other success measures, e.g., spectral envelope distortion, are needed [59].

Among modern waveform speech coders are *generative* decoders and *end-to-end* neural audio coders; these can handle many audio signals, not just speech, as they often do not employ explicit speech models. Such neural models include *WaveNet* [90], its variants *WaveGRU* in *Lyra* [56], *WaveRNN* in *LPCNet* [130], and *SampleRNN* [82]. These will be further discussed in Section 12. The *E-model* can determine the effect of packet loss on voice quality if the RTP/UTP voice-over-Internet protocol (VoIP) is used in digital networks [85] Table 1.

2 Sampling and frame rates

As almost all signal processors are digital, a first step for analysis is *analog-to-digital conversion* (ADC). To produce analog speech to listen to, the final step in a speech codec uses the inverse digital-to-analog conversion (DAC). Many physical phenomena, including speech, can be viewed as signals that vary continuously in time, but digital computers require binary bit sequences. Analog speech occurs as air pressure waves emitted from a speaker's *vocal tract* (VT). Such an analog signal is represented uniformly in time at sampling rate *Fs*, e.g., 8 kHz for telephone applications, which is chosen to preserve the signal spectrum up to *Fs/2* Hz [70]. Many telephony waveform coders, e.g., mu-law, ADPCM, *adaptive transform coding* (ATC), and *sub-band coding* (SBC), send digital samples at 8 kHz, leading to typical bit rates of 8, 16, 32, and 64 kbps, as discussed below. Early phone applications were limited by carbon microphones and high-frequency line losses, leading to only retaining the 300–3200 Hz range [25]; the resulting 8-kHz sample rate has persisted for decades, owing to the huge investment in transmission networks.

Information content in speech occurs at a much lower rate than the sampling rate. Taking advantage of this,

Table 1 Summary of speech coding methods

Method	Type	Characteristics	Advantages	Disadvantages
Basic PCM	Waveform	Uniform ADC	Very simple	High bit rate
Logarithmic PCM	Waveform	Log amplitude compression	No latency	Medium-high bit rate
Adaptive PCM	Waveform	Quantizer follows energy changes	Simple	Medium-high bit rate
Differential PCM	Waveform	Short-time spectral predictor	Exploits speech spectral envelope detail	Medium bit rate
Linear predictive coding	Vocoder	All-pole spectral model	Low bit rate; standard model for cellular telephony	Loss of phase in basic model
Adaptive transform coding	Waveform	Transmits much spectral detail	Good speech quality	High complexity
Sub-band coding	Waveform	Band-pass filters	Good speech quality	High complexity
Sinusoidal (harmonic) coding	Waveform	Codes individual harmonics	Good speech quality	Requires F0 estimator
Channel vocoder	Vocoder	Flat spectrum in each channel	Low rate	Reverberation; loss of phase
Formant vocoder	Vocoder	Direct formant model	Low rate	Requires estimates of formant frequencies
Variational autoencoder	Neural network	Encoder/decoder	Basic neural model	Costly
Flow neural model	Neural network	Transforms Gaussian noise sequences	Can use parallel processing	More difficult to train
Generative adversarial network	Neural network	Adversarial discriminator and generator	Fast processing	Lower quality than other neural methods
Autoregressive neural model	Neural network	Exploits long conditional pdfs	Very high quality	High latency; costly

vocoders transmit data at lower *frame rates*, portioning speech into successive (often overlapping) time frames of 10–30 ms, with many speech applications operating at a frame rate of 100 Hz [120]. The span of analysis for each frame is called a *window*; window size varies with the need of the application, being short for VT state estimation and longer for F0 estimation. Acoustic detail for phonemes lies in short-term amplitude variations within individual pitch periods, as the impulse response of the VT relates to its shape. Estimating the spacing between repeated periods of F0, on the other hand, needs a window of multiple periods. Speaking rates are rarely above 250 wpm, and phoneme durations average about 80 ms. One-hundred frames/s gives several frames per phoneme, which is thus suitable to track VT motion across phonemes, called *coarticulation* [26].

A coder usually removes redundancy from its input signal by representation of dynamic model parameters that are estimated each frame. Energy, VT resonances, F0, and related parameters vary with articulation; coding information at such a frame rate allows large data reductions in vocoders. There are many coding variations, e.g., standard G.711 encapsulates two sets of 10 ms into one 20-ms packet [42].

3 Acoustic aspects of speech that are relevant for coding

Before discussing details of different coders, it is useful to examine the nature of speech signals, to discover aspects, e.g., redundancy, which coding may exploit. Speech signals have a variety of information, e.g., the text of what one says, who is talking, the language in use, and emotional and health conditions of the speaker. However, all this is encoded in the speech signal in a highly complex fashion. Preserving pertinent information in coding aims to lower the number of bits in its digital representation while retaining both the intelligibility and naturalness of the reconstituted output signal.

The phonemic content of speech resides primarily in the local dynamic behavior of its spectral resonances, but intonation, which spans much longer time ranges, also has a significant impact. Aspects of speech that involve speaker identity and that may reflect speaker condition are far more complex but evolve very slowly in the speech signal. Thus, it is helpful to examine human speech production and perception in more detail.

Human listeners interpret the diverse information in speech according to their learned perceptual processes [122]. When machines process (e.g., code or recognize) speech, they use methods of *signal analysis*, i.e., transform input audio to forms more useful to allow efficient representation. This paper focusses on signal processing that has been used for speech coding, rather than on classification or general signal analysis. Thus, the emphasis here is not on general neural networks, nor on the many pattern recognition classifiers (e.g., k-means, distance measures, maximum likelihood [27]), as these are not specific to speech coding. This paper examines efficient speech signal transformations, analysis concepts, and their motivations and leaves most mathematical and algorithmic details to the cited references. Some coding methods emulate aspects of human audition, but others simply reduce redundancy.

3.1 Phonemes

A speech signal consists of variations in air pressure caused by waves from a speaker's VT (Fig. 1). The signal results from a complex sequence of human processes, including some that are still poorly understood [21]:

1) A speaker transforms ideas into conceptual word sequences.
2) Commands to muscles invoke VT motion, with the speaker aiming for a series of VT shapes corresponding to brief individual linguistic units called *phonemes*.
3) Air is pushed from the lungs.
4) After propagating through space, sound pressure variations received by ears cause a listener's eardrums to pass these vibrations to the basilar membrane in the cochlea.
5) These induce auditory neural firings to the brain, via thousands of cochlear hair cells.

Speech is a communicative process, intended to pass a message to listeners. The speaker controls the VT to facilitate this transfer of information. At one linguistic level, a speech signal consists of a sequence of phonemes whose durations and spectral characteristics vary in time. The speaker moves the VT to achieve a series of positions that produce a sequence of phoneme sounds suitable for interpretation by listeners [83].

Most languages have a set of about 30–40 phonemes from two main classes: vowels and consonants, each distinguished by acoustic properties of periodicity, timing, and spectral detail [67]. Coders that replicate these properties well have good intelligibility. Naturalness, on the other hand, is far more difficult to quantify, which has led to a diverse range of speech coders.

Most phonemes use vocal cords vibrating at F0 Hz; these are called *voiced* phonemes. Puffs of air from the lungs, spaced roughly uniformly in time, excite the VT, which acts as a filter, to produce this strong speech. If the VT has a major constriction, the periodic airflow creates a voiced *fricative* in the form of repeated bursts of spectrally shaped noise. Such *obstruent* phonemes consist of

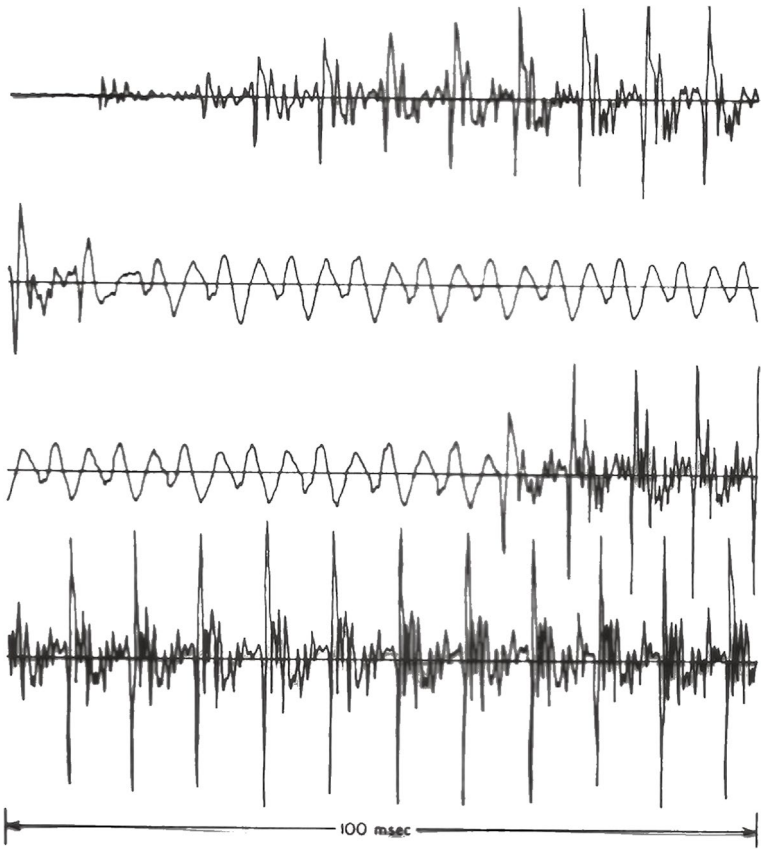

Fig. 1 Example of a speech waveform in time; four successive sections of 100 ms each. Two quasiperiodic strong vowels (10–115 and 260–400 ms) have a nasal (115–260 ms) between them (/ana/)

broadband noise, initiated at the outlet of a VT constriction and exciting the upper VT above the constriction. If the VT has no major constriction, on the other hand, speech has repeated segments called *pitch periods*; these *sonorant* phonemes include vowels, nasal consonants, and related consonants (liquids and glides). If the vocal cords are not vibrating, resulting sounds are *unvoiced obstruents*.

During speech, the VT is always changing shape, owing to a combination of lagging effects from earlier phonemes, positioning for each current phoneme, and anticipating future phonemes. These dynamics cause *coarticulation* effects in which pitch periods in sonorants are only quasi-periodic, not perfect copies. Articulatory and acoustic effects of coarticulation from neighboring phonemes can extend over several phonemes [26]; e.g., in the phoneme sequence /stru/, lips are rounded throughout, in anticipation of vowel /u/.

Vocoders estimate relevant and dynamic aspects of speech production, such as energy, spectral detail, and periodicity [72]. These three properties of speech are often used because they have phonetic relevance, are readily controlled by speakers and interpreted by listeners, and allow for low-rate representation.

3.1.1 Properties of phonemes: articulatory and acoustic

It is useful to examine both articulatory and acoustic aspects of speech, as low-dimensional parameterization is feasible for phonemes, which can allow great reduction in coding bit rates, e.g., 8 phonemes/s vs. 8000 samples/s. However, estimation of many traditional categorical *phonetic features* such as tongue height, voicing, nasality, resonances, and F0 is often unreliable [19, 100]. Thus, both ASR and speech coding generally avoid features, instead using *parameters*, which are directly calculated from time-frequency (T-F) analysis without classification estimations, such as average energy, LPC, and MFCC (see later). Nonetheless, phonetic features are worthwhile to examine, as they can be efficient representations and may provide ways toward future low-rate coders.

Tongue height in vowels correlates inversely with the center frequency of the lowest-frequency resonance of the VT, called the first *formant* (F1) (Fig. 2). (F_i is the center frequency of the i-th resonance; note that F0 is not a formant). These center frequencies have far more

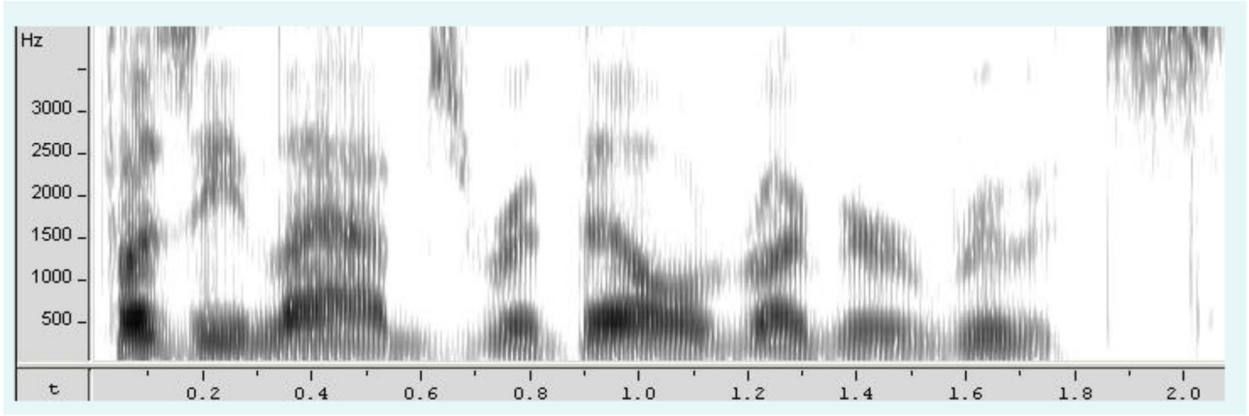

Fig. 2 Example of a wide-band speech spectrogram. Note the horizontal dark bands (formants), roughly spaced every 1 kHz; vertical axis is frequency in kHz; horizontal axis is time in seconds

relevance for communication than the bandwidths of resonances. For phoneme discrimination, listeners pay close attention to the frequencies of prominent spectral peaks. Other speech production effects include the following: lateral tongue position correlates with F2, and a retroflex tongue tip causes a lowering of F3. Other articulatory behavior of the VT (e.g., velum lowering, lip rounding) has various known effects on speech spectra. VT formant bandwidths and spectral zeros (antiresonances) have less value in communicating information [23]. Thus, low-rate coders focus on modelling spectra at peak frequencies.

For obstruent consonants (stops and fricatives), a closure or constriction point (e.g., labial, alveolar, velar) creates roughly tubular shapes in the VT that are relatively isolated acoustically from each other, with related spectral effects. As shorter tubes in obstruents cause resonances at higher frequencies, the need to model detail in spectra is less than for sonorants [52]. Modelling 2–4 strong low-frequency resonances well is essential for sonorants, but obstruents can be well represented by a simple bandpass model, specified primarily by its lower cutoff frequency. Much speech research has shown that timing and the spectral distribution of energy are critical to both speech production and speech perception [21, 122].

3.2 Intonation

While spectral detail, e.g., frequencies of major energy, which is narrowly localized in time is critical to phoneme representation, speech has relevant information on time scales much longer than brief phonemes. *Intonation*, which consists of signal amplitude, F0, and sound durations, is used greatly in human communication and must be replicated properly in speech coding. Intonation is nonetheless often ignored in most ASR classification, owing to difficulty integrating acoustic information on different time scales. Waveform speech coders include intonation directly, by replicating fine details of the speech time signal. On the other hand, most vocoders explicitly estimate F0, sending its values along with spectral detail for each frame; e.g., LPC sends the all-pole model parameters called LPC coefficients, while *channel vocoders* send amplitudes of band-pass filter outputs [31].

The temporal domain of intonation is words and phrases, e.g., several seconds, much longer than coarticulation, which is usually much less than 1 s. For purposes of intonation, one averages acoustic measures over time scales of frames of 10–30 ms. Such temporal ranges are appropriate to handle both coarticulation and intonation.

3.2.1 F0 estimation

Among the aspects of intonation, amplitude is easiest to calculate, via a simple average of waveform peaks or energy during each frame. Duration is rarely estimated in current coders, as there is little need to seek unit boundaries in coders that send data every frame, and neural coders generally avoid such prior processing. The 1980s saw several *segmental*, or *phonetic*, speech vocoders [106], which grouped successive speech frames that were similar spectrally into phonetic segments, e.g., parts of phonemes, thus allowing nonuniform, lower transmission rates once per segment. Modern coders tend to avoid such specific decisions as to segment boundaries, as their estimation is often unreliable [91], and there is little demand for very-low-rate coders, as most users prefer quality to incremental cost reductions.

Unlike amplitude and duration, F0 requires a complex algorithm to reliably estimate pitch-period duration, which varies widely. While spectral peak detail (resonance positions) is a prime determining factor in speech

quality, perhaps the single most important individual feature of voiced speech is F0. If a speech coder does not replicate periods correctly, listeners immediately detect flaws in the decoded speech. In theory, a periodic signal has a spectrum consisting of discrete lines at harmonics, which are multiples of F0. In practice, a sonorant phoneme (the most common speech sound) can be analyzed as a windowed periodic signal, whose resulting spectrum is smeared over a frequency range inversely proportional to the window duration (also called the "leakage effect"), which causes the spectral lines to spread out, e.g., 50 Hz width [16]. The amplitudes of the harmonics are weighted by the spectral envelope of the VT filter. Searching for equally spaced harmonic peaks in speech spectra is one way to estimate F0 [100]. Since speakers change F0 relatively slowly (usually much less than an octave during a phoneme), F0 estimators may assume small changes from period to period, except when switching between voiced and unvoiced speech.

The spectral envelope is a function of VT shape, which is independent of F0. F0 estimation often simplifies the speech spectrum by "flattening" its envelope and/or eliminating phase effects. For the latter, *autocorrelation* of the speech signal obtains a zero-phase and squared-amplitude spectrum, as it is the convolution of a signal with its time-reversed version [69]. In autocorrelation, speech is multiplied by a delayed version of itself, and then averaged, yielding maxima at multiples of the pitch period. LPC analysis often uses autocorrelation as an efficient way to estimate a spectral envelope. A less costly version of autocorrelation is the *average magnitude difference function*, which subtracts the speech waveform from itself, delayed by possible values of pitch periods.

Flattening the spectrum may use an LPC inverse filter, whose output preserves harmonic structure [81]. Cheaper ways use a simple nonlinear, time-domain distortion that retains pitch-period peaks, while suppressing other detail, e.g., a (full- or half-wave) rectifier, or using a threshold that eliminates all waveform details below a certain level [100]. Here, one wishes to render speech into a version resembling a flat, line spectrum, which corresponds to a uniform impulse train in time, which is the easiest to measure for F0.

3.3 Summary of useful speech features

Many years of research have shown that certain phonetic features present in speech signals are clearly useful for representations in speech applications. One cannot formally prove that any of these are explicitly controlled by speakers or directly exploited by listeners, but very strong correlations with human communication are linked to the following: formants, F0, overall energy, and periodicity [22]. As reliable estimation of some of these categorical features, e.g., formants and F0, has been difficult in common practical acoustic environments, we see below that parametric measures, calculated directly by formula, have become common in speech coding. It is nonetheless clear that exploiting versions of the spectral distribution of speech energy helps coding performance; e.g., LPC replicates major VT resonances with less than a dozen parameters, allowing listeners to perceive decoded speech well with low bit rates [79]. Thus, preserving these features, under reduced bit rates, is a main task for speech coders.

4 Time-domain speech coding

The simplest speech coders operate on the time waveform directly. Several of the waveform techniques in this section apply to a wide range of signals other than speech, e.g., video, rainfall, and X-rays, while other techniques exploit specific aspects of speech.

4.1 Analog-to-digital conversion (ADC)

Section 2 briefly discussed digitization as the first step in speech coding. The parametric choices for ADC/DAC are as follows: sampling rate, type of quantization (e.g., uniform vs. logarithmic), number of bits, and assumed input amplitude range. A basic quantizer samples a signal uniformly in time at a chosen rate (*Fs* Hz), i.e., selecting a sequence of real numbers that correspond to the values of the input signal at times spaced every $Ts = 1/Fs$; sampling signals non-uniformly in time adds complexity that is rarely useful [114]. *Fs* is chosen to exceed twice the highest frequency (the *Nyquist rate*) in the input signal, as uniform sampling in time produces copies of the signal spectrum spaced at intervals of *Fs* [16]. When energy is present above *Fs/2*, these copies overlap, causing corruption called *aliasing*. Coders cannot recover from such distortion, so they usually try to minimize energy above *Fs/2* via use of an analog low-pass filter prior to ADC [120].

Most physical signals of interest, including speech, have energy primarily at low frequencies. As the standard telephone system heavily attenuates energy above 3.2 kHz [25], an 8-kHz sampling rate is very common. Audio signals, in general, lack energy below 200 Hz, but coders rarely exploit that small gap. Sub-band speech coders (Section 6) divide the spectrum into distinct smaller frequency ranges, via digital band-pass filters, which can then each use lower sampling rates than *Fs*. The Nyquist principle of using *Fs* sample/s for a bandwidth of *Fs/2* can apply not only to low-pass signals but also to band-limited signals, but the frequency bands must be chosen to minimize aliasing; e.g., decimation to reduce sampling rates in the narrow bands puts equally spaced copies of

each spectral band, and thus, choices for filters must minimize alias overlap [12].

A major ADC parameter of interest is the number N bits/sample for quantization. Time sampling prevents retaining spectra outside the selected frequency range (0–*Fs/2*), while quantization causes amplitude distortion for each sample, as one represents each real-valued signal sample with a number selected from a finite set of possible values. These values can be uniformly spaced by a step size, i.e., equal to the range divided by the number of levels, *2^N*, which is constant in time, or a step size that varies with amplitude and/or with time. The sampling precision is a compromise between cost (more bits to send) and quality. For audio signals, one usually selects $N = 8$ for *mu-law* telephone applications (see below), because listeners cannot detect the presence of such low quantization noise in normal speech at that precision [29].

4.2 Logarithmic compression

The simplest waveform coder (uniform ADC) is called *pulse-code modulation* (PCM). The basic digital telephone network instead uses *mu-law* or *A-law* logarithmic ADC, which compresses each sample's amplitude on a scale approximating a logarithm (64 kbps: 8 bits/sample [120]). The actual compression is linear at low amplitudes, as log (0) is infinite. This process has no memory, and thus no coding delay (*latency*). The amplitude warping exploits the average signal amplitude pdf, which is Laplacian (exponential) [92], as coders are most efficient when using all quantizer levels equally on average. Thus, the encoder does an approximate logarithmic compression, and the decoder does an exponential expansion. Coders often have this inverse relationship to decoders, i.e., the decoder "undoes" what the coder has done. The analysis process is thus to compress the input to the quantizer, so that the ADC quantization noise is reduced.

4.3 Time-adaptive coding

Memoryless mu-law compression is simple and efficient, but does not exploit dynamic amplitude variations in speech. For example, vowels are much more intense than consonants, and when the vocal cords close, the VT has a large excitation, i.e., each pitch period starts strongly and fades in time [21]. Thus, using a quantizer that changes dynamically to match step-size to the varying energy in speech would maintain a good *signal-to-quantization-noise ratio* (SNR) throughout the frequent wide energy swings in speech. A simple version is *adaptive pulse-code modulation* (APCM) [16]. The only analysis needed is a short-term estimate of dynamic speech energy, at a suitable frame rate, averaging periodically within a limited time window.

Speech is a nonstationary random process, and analysis is usually repeated over brief windows that are shifted periodically in time at a suitable frame rate. APCM simply uses average energy per frame to adjust quantizer step size, while other adaptive techniques use more detail. Unlike most speech applications, frame updates for APCM can vary well beyond the traditional 100 Hz, for the so-called *syllabic* and *instantaneous* processing; the former averages long spans of speech samples; the latter uses short analysis windows and thus needs more frequent updates.

APCM exploits temporal correlations in a coarse fashion, e.g., total energy. At some additional cost and much more useful for efficiency are *predictive coders* [120]. Most audio, including speech, is dominated by energy found mostly at low frequencies. Coders must still retain high frequencies for full representation, but the predominance of low frequencies allows use of a *differential* coder, coding the (smaller) difference between successive samples, rather than samples themselves. These coders exploit waveform detail within individual pitch periods of voiced speech, which is the result of VT filtering. Such predictive coders model the spectral envelope of windowed speech. A predictive coder encodes the difference between each successive speech sample and an estimate of that sample based on the recent history of the signal. For most of speech (sonorants), this difference is much smaller than waveform samples themselves. With smaller step sizes, there is less quantization noise.

In minimizing noise, and thus maximizing SNR, all frequencies are often treated equally. However, human perception has nonuniform time-frequency resolution, and many coders "hide" portions of quantization noise in T-F sections where speech energy is high, to exploit masking effects, e.g., noise feedback coding [80].

4.4 Linear predictive coding (LPC) and adaptive differential PCM (ADPCM)

The most common form of speech coding that combines adaptation and prediction is *linear predictive coding*, where the coder forms a predicted estimate of each waveform sample as a linear combination of N immediately prior samples; $N = 10$ in most telephony applications, as this accommodates 4–5 VT resonances, using two parameters per formant (Fig. 3). In sonorants, the primary excitation in each period is at vocal cord closure [21], so that ensuing samples are mostly based on the impulse response of the VT, as modelled by N coefficients/frame. As each resonance corresponds to two complex-conjugate poles in the z-plane, a 10th-order LPC all-pole (*autoregression* (AR)) model is standard for telephony. Modelling the spectral envelope with only ten parameters is a significant reduction of data

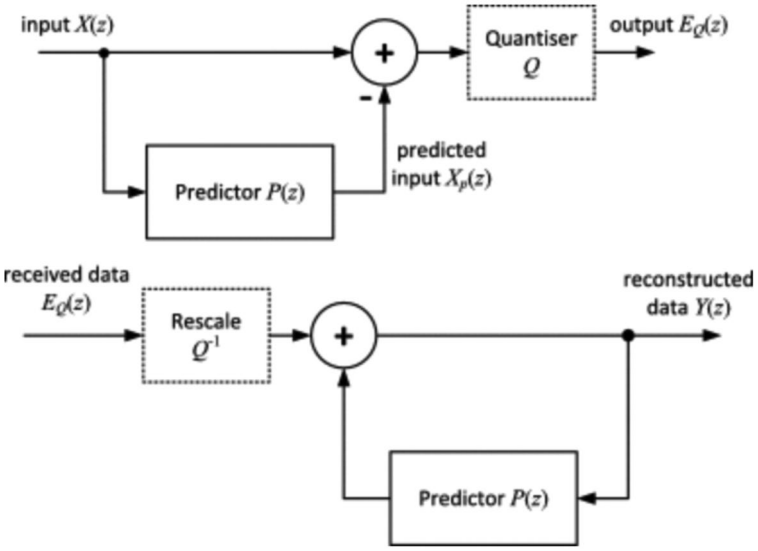

Fig. 3 Basic differential coding. An all-zero predictor P(z) produces a reduced (and more random) quantizer input; all-pole decoder reconstructs the VT spectral shape

for transmission, compared with a Fourier representation. The VT excitation is handled with amplitude, F0, and a voiced/unvoiced bit. The analysis prediction filter for LPC is all-zero (moving average), and the decoder synthesizer is all-pole (Fig. 4). This model is less accurate for non-sonorants, as their noisy excitation is continuous, rather than concentrated at vocal cord closure; however, this is of minor concern, as listeners pay far less attention to phase in noisy obstruents [94].

Speech is a stochastic process. Let $S = \{s1,s2,...,sN\}$ be a vector of N successive speech waveform samples. The speech pdf $p(S)$ can be factored into a product of conditional probabilities $p(S) = \text{Prod}\ p\ (st|st-1,\ st-2,\ \ldots,\ s2,\ s1)$. Each speech sample st is conditioned on previous samples. Dependence exists over a wide time range, but, for efficiency, one usually limits N to 10. A common stochastic modelling approach has an objective to minimize the *Kullback-Leibler divergence* between the "ground truth" (actual) speech joint pdf $p\ (s_i\ ,\ \cdots,\ s_{i-N})$ and its model distribution $q\ (s_i,\ \cdots,\ s_{i-N})$ [56]; this is equivalent to the *cross-entropy* (CE) between these distributions [116]. CE is a common loss function for ANN training, e.g., maximum likelihood-based teacher forcing.

The difference between each speech sample and its predicted estimate is the LPC *residual* error, and average *minimum mean-square error* (MMSE) between input and output signals is used to determine the model parameters, i.e., the multiplier weights of the linear prediction combination [79]. This quadratic error is minimized by a solution of linear equations; optimal weights are found using a partial derivative of the error equalling zero. MMSE focuses on matching spectral peaks, which is desirable perceptually. The all-pole speech synthesizer (decoder) corresponds to the inverse of its (feedforward, all-zero) differential analysis model.

ADPCM (e.g., G.726 standard) uses this coding analyzer to transmit the residual error at *Fs* Hz [109], whereas basic LPC uses a very coarse excitation model for its all-pole synthesizer. A simple version of ADPCM is *continuously variable slope delta modulation*, which handles very noisy audio conditions better than LPC [61]. Basic LPC sends only data at the 100-Hz frame rate; one bit notes voiced vs. unvoiced (i.e., periodic impulse or noise excitation). Some ADPCM uses an *autoregressive moving-average* model, as speech has spectral zeros [102], but the LPC model is far simpler if all-pole, and listeners focus far more on spectral peaks than valleys, as perception is dominated by the presence of energy, rather than its absence [122]. "Moving average" means a finite-duration impulse response, whereas autoregressive models have output depending on prior samples.

Calculation of the LPC model usually involves inversion of a matrix of an autocorrelation of the windowed speech time waveform. Full matrix inversion is not needed as the symmetry of this matrix can be exploited to calculate the inverse efficiently, e.g., by the Levinson-Durbin algorithm. Averaging the product of signal samples spaced by very short-time delays yields an efficient representation for signals that have simple spectral structure, e.g., an amplitude envelope that has a few resonances, which is appropriate for speech. Owing to its emphasis on spectral peaks, the resulting LPC spectrum models resonance

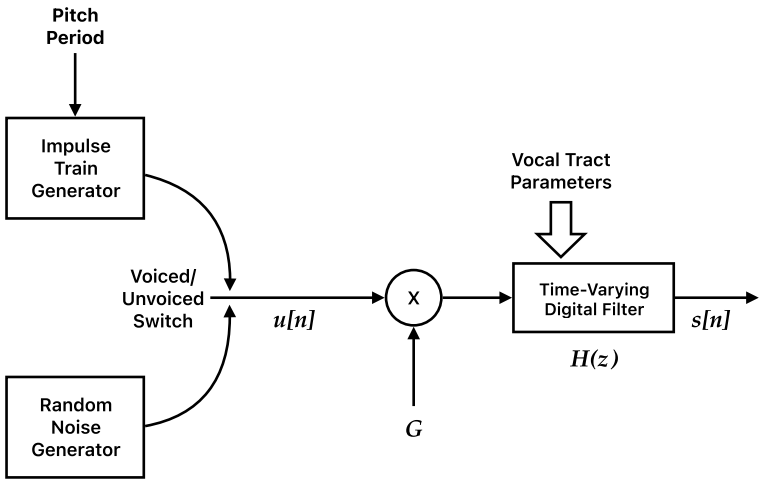

Fig. 4 Basic LPC decoder. All-pole synthesizer is excited by a binary choice between random noise (for unvoiced speech) and impulses (for voiced speech)

center frequencies very well but tends to underestimate resonance bandwidths (frequency spans between −3-dB formant amplitude values [127]). For most applications, this is not a significant disadvantage. The all-pole LPC model continues to be used in modern cellular telephony, but higher quality output speech is obtained via use of more advanced excitation models (see next subsection) than the basic binary voiced/unvoiced model. It is widely used in the Global System for Mobile Communications (GSM) [87] and in VoIP [35].

4.4.1 Advanced excitation models for LPC

Modern telephony (e.g., GSM) uses bit rates around 10 kbps and employs the original LPC VT model but uses ACELP (algebraic code) excitation to send more detail about the residual [108], at bit rates that are lower than 16–32 kbps ADPCM (Fig. 5). Early LPC [79] used a binary impulse-or-noise selection as input to the all-pole VT filter synthesis model; such output speech is intelligible but has a mechanical ("buzzy") quality. Human speech is never truly periodic; the so-called pitch periods always vary in time, sometimes slightly and often more during coarticulation. These variations are not random but are difficult to model; so when researchers applied

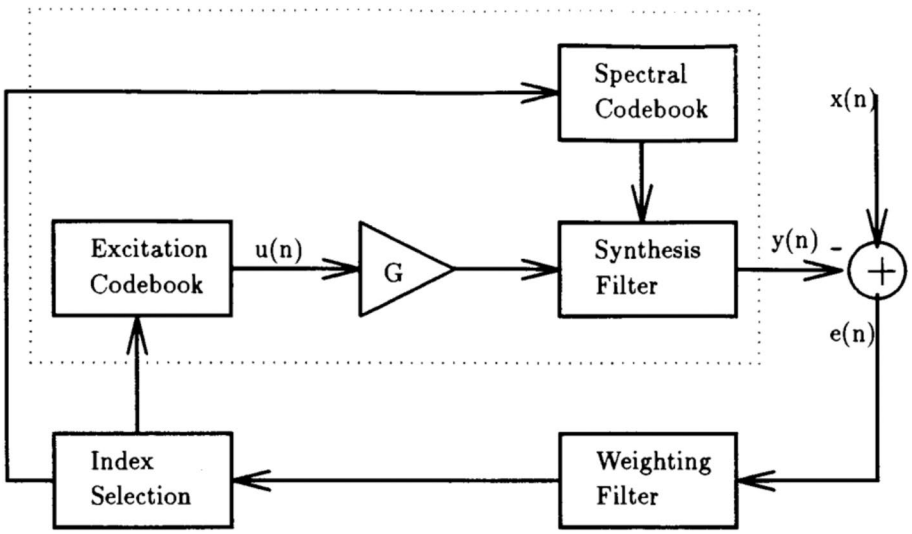

Fig. 5 CELP generates reconstructed speech y(n) via a basic LPC all-pole filter of order 10, excited by brief excitation patterns in a 10-bit vector quantization (VQ) codebook. The LPC filter itself is also selected from a trained codebook

jitter (amplitude randomness) or *shimmer* (duration randomness) [39] to pitch-period impulses, quality showed little improvement.

Since harmonics are sharper at low than high frequencies, another candidate excitation consisted of low-pass-filtered impulses and high-pass-filtered noise, which only slightly improved quality [68]. This spectral difference in natural harmonics is related to plane-wave propagation in the VT, which is most valid at low frequencies, where long wavelengths block oblique wave propagation. After these simple (inexpensive) modifications to the LPC excitation model, a much more computation-intensive *multipulse* (MP) approach [65] became the state of the art in the late 1980s. MP-LPC approximates the actual LPC residual error signal, transmitted in high-rate ADPCM, with a reduced skeleton-type excitation, sending amplitudes and times of individual impulses to be used as excitation. The temporal locations and amplitudes of excitation impulses were selected to minimize a weighted squared error between the input and output speech. This MMSE was weighted to emphasize precision at strong spectral regions, where listeners pay most attention.

For CELP, an excitation sequence selected from a trained codebook is input to a cascade of linear prediction and pitch filters to reconstruct speech. The LP filter restores the spectral (short-term correlation) information, while the pitch filter creates periodicity. Modern ACELP coders use an algebraic codebook with a predefined regular structure, in which excitation pulses, all of the same amplitude, are organized by tracks. (Further discussion is in Section 5.)

4.4.2 *Line spectral pairs*

For efficient vocoder transmission, the ten traditional LP coefficients are usually transformed into a set of ten *reflection coefficients* or *line spectral pairs* (LSPs) [119]. The LP polynomial *A(z) = 1* — sum *{k = 1* to *p} a_k z^{-k}* describes a direct-form digital filter, where the ten *a_k* LPC coefficients do not have good quantization properties, e.g., require too many bits to avoid perceived spectral distortion in the reconstructed speech. A lattice filter with reflection coefficients [79] instead models the same VT filter, but with values bounded by +/−1, guaranteeing stable synthesis, as in models of actual traveling waves in the VT.

Even more efficient are LSPs, which use two equations to replace *A(z)*: *P* a palindromic polynomial and *Q* an anti-palindromic polynomial:

where *A(z) = 0.5 [P(z) + Q(z)]*

P(z) = A(z)+z^{-(p+1)} A(z^{-1})}

Q(z) = A(z)-z^{-(p+1)} A(z^{-1})}

P corresponds to a model of the VT with the glottis closed and *Q* with the glottis open.

The roots of *P* and *Q* lie on the unit circle in the complex *z*-plane, and their roots alternate around the circle. The coefficients are simply angles in *z* and thus real and in conjugate pairs.

5 Vector quantization (VQ) methods

Block encoding is more efficient than memoryless (instantaneous or scalar) coding [134], as sending individual data samples independently ignores correlations among multiple parameters of a data representation. Successive speech samples in either time or frequency, as well as other representations, e.g., LPC parameters, are often highly correlated. Optimal scalar quantization can be extended to a high-dimensional space via the generalized Lloyd algorithm [75], which is similar to *k*-means clustering, where data points are assigned to clusters so that the sum of the squared distances between the data points and their centroid model is minimized [74]. In VQ, a point in a high-dimensional space is mapped onto a discrete set of *L* code vectors. Structured vector quantizers, e.g., residual, product, or lattice, seek a trade-off between computational complexity and computational efficiency.

Consider any set (e.g., block or vector) of *N* numbers representing a portion of a signal from a random process such as speech, i.e., *x (1), x (2), … x(N)*. These could be *N* successive time samples or *N* spectral values. If the signal is not white noise, samples have correlation to exploit in efficiently coding the block as a unit (LBG algorithm [6, 33]).

If the data have a sequential relationship, one may use Bayes theorem: *P(A, B, C, D, … Z) = P(A) P(B|A) P(C|A,B) … P(Z|A,B, … Y)*; any joint pdf is the product of such a series of conditional densities. This is especially useful for Markov models, which greatly simplify this product, under some strong assumptions, and has been widely used for ASR [135].

Training starts with a large set of *M* blocks of examples, using many speakers and acoustic conditions. From these, *L* vectors (a *codebook*) are selected or generated to represent the range of all possibilities. For LPC-VQ, *N*-th-order LPC vectors from many frames of speech provide the training data. As in ADC quantization, where each speech sample is reduced to a discrete value from a finite set of scalar output levels, a VQ coder selects, from the *L*-element codebook, the closest vector for each speech frame. A suitable distance measure determines vector proximity. *L* is often 1024, allowing transmission of a single 10-bit code rather than *N* LPC coefficients, each needing more than 5 bits.

The VQ training minimizes the distance measure that is an average of the distances between each of the *M* training vectors and the codebook vector that is closest

to each vector in the L subset. At each training iteration, the current codebook yields this evaluation measure, and then, the codebook is revised with updated centroids, perturbed in directions to reduce the measure.

If the representation space has evident structure, e.g., spectral patterns for different phonemes, more efficient searches are possible. For example, the codebook may be searched as a binary tree, log_2 L comparisons, if organized suitably, but L comparisons may be needed in the coding stage to find the optimal vector. To reconstruct from a coded signal, the decoder, using the same codebook, simply finds the corresponding vector for each frame from its transmitted log_2 L-bit code. Such VQ can apply to both the spectral envelope and the residual of LPC (Fig. 5), although the envelope codebook is easier to organize for faster search.

6 Sub-band analysis

Modern telephony speech coding uses a version of LPC with advanced excitation such as ACELP, with bit rates around 10 kbps. Alternative waveform coders operating at slightly higher rates are SBC and ATC. Both of these exploit the same typical redundant aspects of speech spectra, i.e., a few prominent resonances with quasiperiodic or noise excitation, but not with a differential predictor. Aspects of both are commonly found in modern coders, such as *Opus* [129], as well as older systems, such as MP-3 [5]. As SBC and ATC do not assume a speech source-filter model, they are suitable for general audio coding.

Sub-band coding (SBC) ([12]; e.g., G. 722 standard) exploits human audition's better resolution at lower frequencies, as well as the predominance of lower frequency energy in most audio. Input speech passes through a set of M bandpass filters, covering the full range of spectra but with usually narrower bands at lower frequencies, and each filter output (*channel*) is coded with APCM, with quantizer step sizes and bit assignment adjusted for lower energy at higher frequencies (Fig. 6). Each channel is decimated, for its much narrower bandwidth, and then interpolated back at the decoder, which sums all channel signals to form the reconstructed waveform. This usually requires more bits than ACELP but is found in wideband applications [46]. A low-rate version is called channel vocoder, which has similar quality to basic LPC speech, but is obsolete, as it cannot upgrade its excitation as LPC can.

SBC has M output channels, all much narrower than the original speech bandwidth B Hz, i.e., an average of B/M Hz, although typical use of the mel scale means a nonuniform distribution. This requires *decimation* to reduce the sampling rate in each transmitted channel, so that the overall combined rate does not increase [16]. The cutoff frequencies for each channel using specific filter design can minimize aliasing that can occur as a result of the downsampling [128].

A related downsampling occurs in neural vocoders (Section 12), as each network layer may typically output half as many samples as its input layer; successive layers then can lower the number of samples to transmit significantly. However, this analogy to decimation does

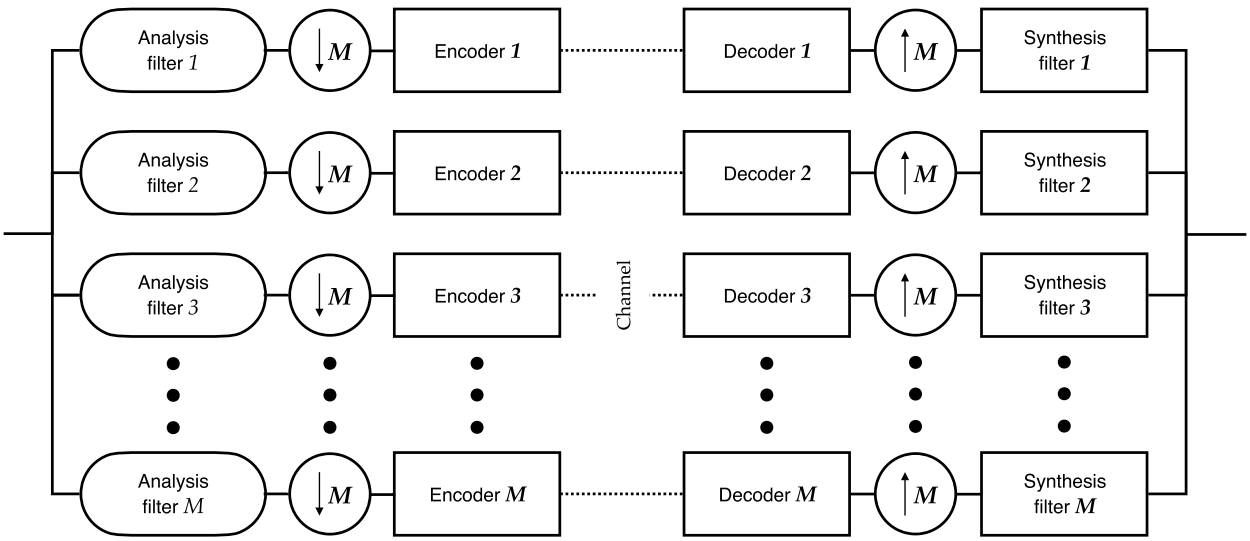

Fig. 6 Sub-band coder with M equal-bandwidth filters; the output of each filter in the encoder is downsampled *M:1* and then upsampled *1:M* in the decoder. In each channel, the analysis and synthesis filters are the same

not apply for ANNs to the mechanism of bandwidth and aliasing, as the operation of ANNs is nonlinear. Similarly, while SBC decoder interpolation simply inserts zero values between the decimated samples, and then bandpass filters (before summing all channels), the upsampling found in ANN decoders is far more complex. The idea of treating frequency bands separately, according to their utility in speech communication, persists in some neural approaches [88].

7 Adaptive transform coding (ATC)

Also used in medium-rate applications is a direct encoding of the speech spectrum in *adaptive transform coding* [136]. Rather than having a small set of filter channels as in SBC, one codes samples directly from a spectral transform (Fourier transform or modified discrete cosine transform — MDCT), assigning individual APCM encoding to each sample. MDCT is an extension of basic spectral representation to overlapping blocks [133]; MDCT is common in VoIP, e.g., the G.729.1 coder. ATC uses a block approach, requiring a delay of one frame in streaming (real-time) applications. Basic LPC also has this same frame delay, but some high-order LPC uses instantaneous updating that does not invert a covariance or autocorrelation matrix each frame; in that case, the delay is only the brief predictor history of 10 samples [8]. The ATC decoder synthesizer simply inverse-transforms back into a time waveform.

A challenge for ATC is to efficiently estimate each sample's coder parameters (step sizes and numbers of bits), so that the decoder uses specific APCM, without sending much side information. The numbers of bits for samples are assigned in approximate proportion to sample energy. One usually estimates both F0 and a rough spectral model (e.g., LPC envelope) for this, which allows reducing the average number of bits per sample to as low as 1–2 vs. 3–4 for ADPCM.

In ATC, choosing a short block size allows for low latency but can yield poor frequency resolution. Coefficients can be grouped to resemble the critical bands of human audition. Opus uses pyramid vector quant ization — a spherical VQ. This encoding leads to code words of fixed (predictable) length, which enables robustness against bit errors and does not require entro py encoding. An open-source, low-delay audio coder called CELT (Constrained Energy Lapped Transform [129]) uses band folding, which delivers a similar effect to spectral band replication by reusing coefficients of lower bands for higher ones. Band folding was used as well in older codecs like the GSM codec [45]. The MPEG Unified Speech and Audio Coding standard uses piecewise constant envelope models known as scale factor bands [41].

8 Harmonic sinusoidal coding

Voiced speech requires far more coding precision than unvoiced speech, as listeners pay little attention to phase in unvoiced speech. Thus, one direct vocoder method called *sinusoidal coding* encodes parameters for harmonics in each frame of voiced speech. Spoken audio is recreated as a sum of harmonically related sine waves with coded amplitudes and phases. The open-source *Codec2* [107, 118] uses this for bit rates of 450 bps–3.2 kbps.

9 Mel-spectral analysis

Audition is highly nonlinear; in particular, spacing along the basilar membrane of the inner ear corresponds, above 1 kHz, to frequency on a logarithmic scale. This nonlinear mapping is called the *mel scale*, which is also viewed as 24 *critical bands* covering the full spectral auditory range [139]. Since 1990, the most common method of speech analysis for ASR has been the *mel-frequency cepstral coefficients* (MFCC [13];). Cepstral analysis was first developed for deconvolution, i.e., separating the two components of a convolution, such as the output *s(n)* of a VT filter with impulse response *h(n)* driven by a VT excitation *e(n)*. As speech is often viewed as periodicity or noise exciting a VT model, such a cepstral process can estimate both the excitation input and the VT filter. The high computational complexity of such a cepstral coder has prevented its practical use in speech coding, however. We discuss these ideas here, as recent neural speech coders often employ mel-spectrogram features.

MFCC combines spectral analysis with aspects of audition. It transforms each set of *N* speech samples *s(n)* into spectral amplitude (discarding phase), weights (multiplies) all spectral values using about 26 triangular-shaped "filters" spaced by the mel-scale, takes an amplitude logarithm, and then, inverse transforms back to the time domain, as a set of *c(n)* coefficients in time. Simpler logarithmic *band-pass filter energies* — a version of MFCC, but without the final inverse frequency transform step — have been increasingly used in ANN ASR [36].

The mel scale is useful in speech coding, as it focuses speech data along an axis more suitable for perceptual evaluation, as the coding output is destined for human ears. However, a mel spectrogram shows only amplitude, and phase must be handled separately. ANN methods for speech coding often take inputs in the time-frequency domain from a short time Fourier transform, MDCT, or a mel spectrogram [98].

10 Hybrid speech coders

Vocoders provide intelligible speech at low bit rates, e.g., 2.4 kbps, but their elimination of phase limits quality. The common 2.4-kbps rate is a holdover from the days of the original 300-bps modems, using a power-of-two

multiple; similarly, the common use of 8 and 16 kbps coders derives from submultiples of the 64-kbps telephony standard. Waveform coders yield excellent reconstructed speech but typically require more than 6 kbps. Hybrid systems combine elements of both classes of coder to allow many practical applications, over a range of bit rates. ACELP is a common hybrid coder, found in telephone networks, using the basic spectral LP model of vocoders but having excellent phase via use of an advanced excitation model. Many other systems combine elements of VQ, SBC, ATC, ADPCM, and LPC. Hybrid coders are used in most mobile telephony and VoIP standards, e.g., the AMR (adaptive multi-rate) coder [3]. Various governmental organizations have established standards for speech coding, including the International Telecommunications Union (ITU), European Telecommunications Standards Institute (ETSI), Moving Picture Experts Group (MPEG), and Telecommunications Industry Association (TIA) [62]. ITU has focused on landline telephony, MPEG on multimedia, and ETSI and TIA on digital cellular.

Opus [129] is a hybrid coder that provides good quality above 6 kbps. Opus is one of the main audio coders on YouTube for streaming and by Zoom. Opus and EVS (Enhanced Voice Services — successor to AMR-WB [18]) are state-of-the-art audio codecs, with various bit rates from 5.9 to 128 kbps, with four sampling rates (8, 16, 32, and 48 kHz). Opus uses a combination of an LPC component (called SILK, as used by Skype) and an ATC part (CELT). A speech residual is coded as a sum of pulses, plus a pulse-dependent dither signal. Both SILK and CELP coders are hybrid coders, with weighted waveform matching loss functions. *Lyra* is a generative model that encodes quantized mel-spectrogram features of speech, which are decoded with an autoregressive WaveGRU model to achieve excellent output at 3 kbps [56].

When the receiver in a speech coder is resource-constrained, it can use a parametric vocoder for output; otherwise, higher quality is possible with neural coders (see below). Parametric vocoders often use a modified version of the Griffin-Lim algorithm [34] for synthesis; it provides a higher quality than use of an LPC synthesis model, which is more limited in both excitation and its all-pole spectrum. It does phase reconstruction based on redundancy in the short-time amplitude Fourier transform but often requires many iterations, repeatedly converting between frequency and time domains, until it converges.

11 Formant vocoders

Another possibility for speech coding would be to estimate traditional features such as formants and F0, which could allow large reductions in bit rate, e.g., from waveform coders of 8-16 bits/sample at a 8-kHz Nyquist rate (for telephone speech), to as low as a dozen parameters (using a few bits each) at a 100-Hz frame rate. There have been numerous efforts to estimate frequencies for the lowest 3–4 formants in sonorants [19]. Early ASR efforts were expert systems using formant trackers [103] but had limited success, partly due to the fact that formants vary greatly in energy, e.g., /u/ has very little energy above F2. So, this option is rarely used.

12 Neural speech coding

In the last two decades, much of speech processing has turned to machine learning (ML) models, such as artificial neural networks (ANNs), especially ones with more than three layers, called deep neural networks (DNNs). The main motivation has been the ability of ANNs to learn relevant patterns through simple automatic training methods using stochastic gradients and large amounts of training data. Early ML applications were in pattern recognition, first for images and then for audio, as the original work in *multilayer perceptrons* (MLPs) for classification [84].

Unlike biological neurons, which typically output a binary signal — a brief firing (shorter than 1 ms) to a sufficient linear combination of weighted inputs, with a nil baseline, artificial neurons (often called *nodes*) may be designed to output a wide range of transformed data. In particular, reconstructed speech can be obtained from ANNs. As the focus of this paper is speech coding, we will present a brief introduction to ANNs here, to help understand how they are used in speech coding.

12.1 Description of artificial neural networks

An ANN is a nonlinear algorithm that maps an input sequence of data to an output sequence [14]. For speech coding, the input is either time samples of a speech waveform or a set of frame-based spectral representations, e.g., log-spectral magnitudes or MFCCs, and the output is a sequence of reconstructed speech samples. An ANN has stacked and connected layers, each with a set of nodes. Each node typically receives input values from nodes in a previous (lower) layer, weights and sums them, and passes this scalar combination to a nonlinear function. The output of each node is usually a monotonic function of its weighted inputs, in a rough model of biological neurons. In a natural neuron, many dendrites feed input values to an axon, whose output is binary [44]. Each neuron "fires" (output of 1) when the weighted sum of its inputs exceeds a specific *bias* or threshold (otherwise 0).

In an ANN, the nonlinear threshold operator, called an *activation function*, is a smooth, monotonic mapping such as a *logistic sigmoid* or a hyperbolic tangent function [115]. It often has a bounded output [0, 1]; an

exception is the ReLU (*rectified linear unit*). Such functions allow use of derivatives for gradient descent search in iterative training of the ANN parameters (weights and biases) [1]. The basic ANN is an MLP, with a few layers of nodes (Fig. 7). Practical ANNs often have many millions of nodes, including operations other than thresholded linear weighting, e.g., *pooling*, which selects a maximal value from among inputs [110].

Each perceptron node with N inputs specifies a hyperplane in N-dimensional space, by the linear combination of its weighted inputs: a 0/1 output specifies either side of the hyperplane. Varying the bias level allows operations that are more complex than a binary choice. Such complexity is needed to handle the huge variability seen in many applications, including speech coding. However, the resulting complexity hinders heuristic interpretation of ANN actions. An ANN is not a "black box," as its parameters are accessible to designers, but its typical huge size and complex operation greatly hinder any attempt to debug.

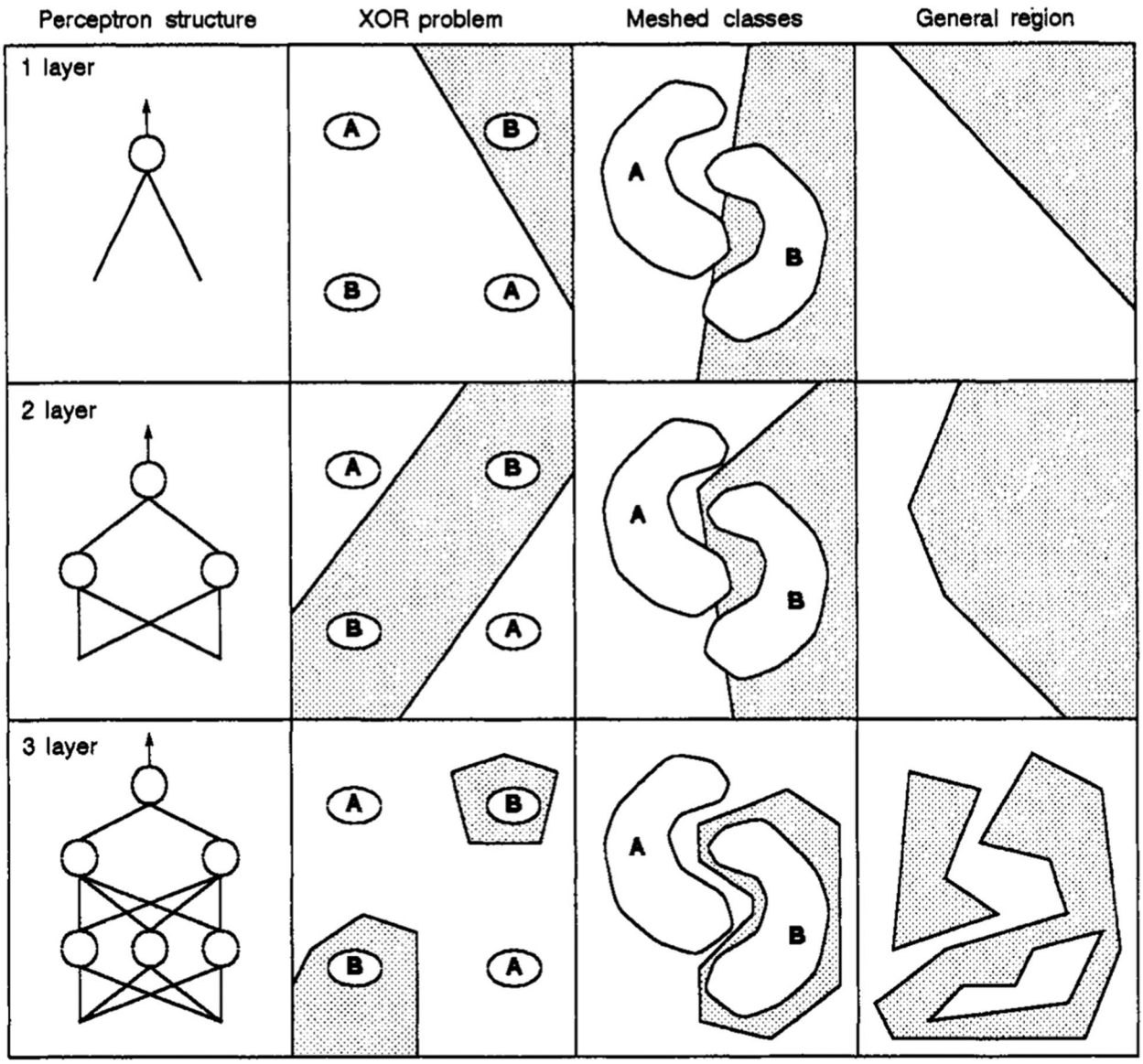

Fig. 7 Possible regions for MLPs (from [76]); given the extreme complexity of regions in most ANN applications, many layers are needed (three are shown here). The challenge of partitioning for complex spaces is illustrated for simple cases of handling targets that distribute in non-convex ways. The middle column poses a case of two classes (A, B) that are not contiguous, and the next column theorizes a case where classes A and B may have more arbitrary shapes, blocking simple classification. Possible class regions are shown and shaded or not

12.2 Training criteria: loss functions

ANN parameters are trained to minimize a differentiable *loss* function, i.e., a cost aimed to facilitate successful network outputs. The ideal alternative procedure, i.e., direct minimization of errors or distortion, is often difficult as the relationship between network parameters and success criteria is complex, unlike minimizing the simple quadratic error in basic LPC. Costs such as *cross-entropy* are common [137] but are usually modified to avoid *overfitting*, where models become too close to specific limited training data and insufficiently general to handle the huge variability in signals such as speech.

Entropy is a physical property associated with randomness, and the concept is used in coding to relate to pdfs and model training criteria. Shannon's source coding theorem [113] states that the optimal code length for a symbol to transmit is $-\log P$, where P is its probability. Common entropy coding methods are Huffman coding and arithmetic coding [60]. Entropy implies the minimum bitrate achievable with lossless coding. Coding often transforms input data into latent features with the smallest possible entropy under a certain level of distortion, yielding a nontrivial rate-distortion optimization problem [123]. During training, atypical outliers can cause a predictive model pdf with heavy tails and thus signal reconstruction with high entropy, which can yield noisy output.

One issue in ANN training is the sensitivity associated with attributes of a typical *log-likelihood* objective function [97]. This objective function incurs a significant penalty if the model assigns a low probability to observed data. In autoregressive structures, this encourages an overly broad predictive distribution when some training data are difficult to predict accurately.

Available training data are rarely adequate to anticipate most possible future inputs. To generalize, one often augments the primary loss function with a *regularization* term [30]. Regularization perturbs or diversifies training data, to generalize models and limit excess model flexibility. *Data augmentation* is also common, in which actual training data are modified by artificial distortion, e.g., additive noise, and/or deletion of random portions in time and in frequency, to increase the training set [117]. ANN training is subject to many problems, including mode collapse (Section 12.6), posterior collapse (where a generative model learns to ignore a subset of the latent variables), and vanishing gradients. For example, in each iteration of training an ANN, network weights are updated in proportion to the partial derivative of the error function with respect to the current weight; with many layers, the gradient may be vanishingly small.

12.3 Types of ANNs

Basic ANNs are *fully connected feedforward* (FFNN), i.e., all nodes in each layer are input to all nodes in the next layer [125]. This, however, is too general for most applications, as data to model, including speech, tend to have a diversity of local and global aspects, which do not require large general structures. For example, voiced speech samples are highly correlated: (a) over 10–20 samples (owing to VT shape), (b) over many dozens of samples (pitch periods), and (c) among phonemes (longer-range phonological phenomena). However, having network parameters for all samples that are individually trained is likely overly complex, leading to both extra cost and lower output quality.

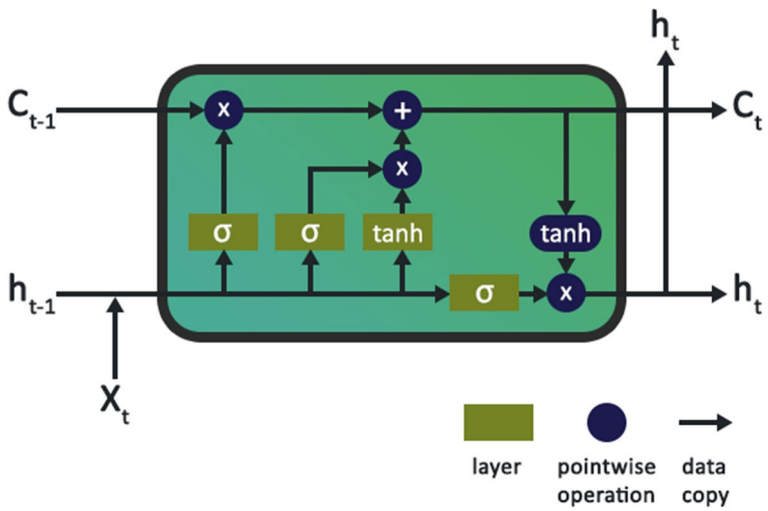

Fig. 8 LSTM cell (from C. Olah, http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

As data in many applications have strong local correlation, a common ANN variant is called *convolutional neural networks* (CNNs) [71]. A CNN processes input data over very small ranges (*kernels*), called *receptive fields*. CNNs are extremely common in ANNs and useful to smooth data (and downsample, accomplishing data reduction) that are often sampled at high rates.

Another major ANN variant is *recurrent* networks (RNNs), which better exploit longer-range patterns in data [112]. Pertinent information in speech occurs very unevenly in both time and frequency: weak portions of speech are far less perceptually important than strong portions, and coarticulation and intonation affect speech over tens and hundreds of frames, respectively. RNNs can deal with this nonuniform distribution of information via use of more complex architectures, while basic MLPs do well with inputs from stationary random processes [77].

RNNs have network architectures with feedback, using distributed hidden states to store information from prior inputs. A common RNN is *long short-term memory* (LSTM) [38] (Fig. 8). For short-term memory in human perception, listeners retain some forms of representation for portions of speech, up to a few seconds of speech, in their brain. Utilizing such a wide range of data in FFNNs and CNNs is very difficult. The range of analysis of an RNN can extend well beyond the very limited scope of CNN kernels. LSTMs have sigmoid gates called input, output, and forget, to utilize temporal data non-uniformly. The forget gates allow variations in phoneme durations, as they vary greatly in speech. A *gated recurrent unit* (GRU) is a simplified version of LSTM that combines forget and input gates into a single update gate and merges cell and hidden states [10]; it handles long-term dependencies, has a simple architecture, and is easy to implement.

A recent modification to ANNs is called *attention* [132], in which network focus can be placed on related portions of data. Attention is viewed as a correlation of relevant information and is calculated via matrix operations (e.g., dot products) that combine several terms: *queries* (inputs), *keys* (features), and *values* (desired outputs, weighted by the attention), with a *softmax* function to obtain normalized values for attention weights [48]. Softmax is a normalized exponential function: network values are raised to an exponential and then normalized by dividing by the sum of all these exponentials; this ensures that the sum of the components of the output layer is 1. This allows the output to be viewed as a likelihood, not a class choice. To date, attention has been very popular in ASR (for classification) but little used in speech coding, as the latter application must reconstruct all speech samples, not just focus on a limited set.

A *temporal convolution network* (TCN) [95] computes low-level features combining CNN (to encode spatial-temporal information) and RNN (using low-level features to capture high-level temporal information), exploiting both levels of information hierarchically. An interleaved structure with causal TCN and groupwise GRU can do temporal filtering for joint long-term and short-term correlation exploitation.

12.4 Applying ANNs to speech coding

Most ANN nodes output a value between 0 and 1, but final output layers often employ the softmax function that can yield a set of probabilities or categorical softmax to output audio samples. In many ANN applications, input is high-dimensional, e.g., huge numbers of samples in audio or video, and both their latent representations and output are low-dimensional, e.g., recognition of classes of

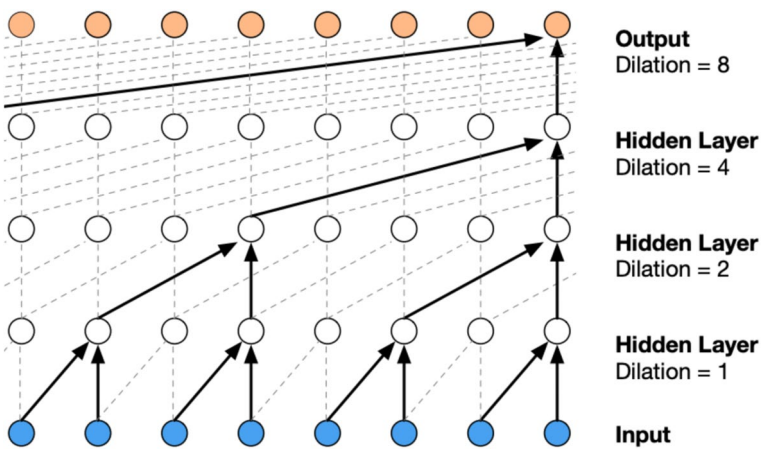

Fig. 9 Dilated neural network

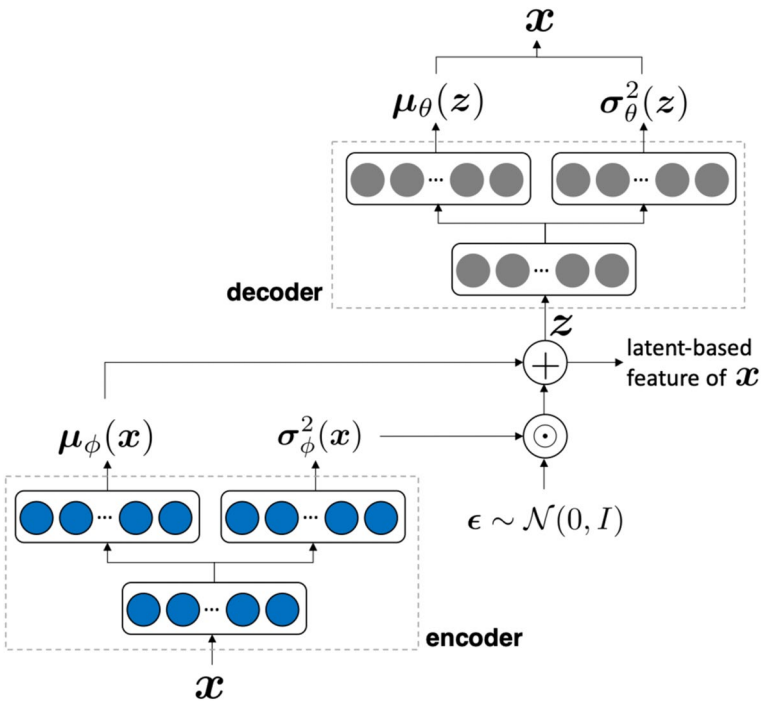

Fig. 10 Variational autoencoder (from Nishizaki [89])

objects present in the input signal (e.g., ASR). Achieving useful data reduction in ANNs requires downsampling to increasingly smaller arrays in network layers. This is often done with *dilated* convolution, also called *a trous*, or convolution with holes [73], by applying convolutional filters over an area larger than input length by skipping input values regularly with a certain step (Figs. 9 and 10). Pertinent information in data is often spread widely among successive data samples, and systematic skipping is an efficient way to "sparsify" a network. This data reduction process is useful for many ANN applications.

Speech coding, however, requires a high-dimensional output, i.e., a reconstructed speech signal, not just a classification (as in ASR). Thus, one needs to up-sample from reduced, latent information in the transmitted data, back to the many output samples [56]. This requires expanding dilation in successive layers in the decoder network. The dilated causal convolutions allow the network's receptive field to grow exponentially with depth. Neural speech coders use conditioning features to guide waveform reconstruction [57]. Examples of such features are spectral envelope information, F0, and gain.

Neural speech coders use many different combinations of the basic units described above, i.e., CNNs, activation functions, dilation, pooling, and RNNs. To help focus on relevant spectral resonance detail, a mel spectrogram of the speech is often used along with the speech samples as input to the ANN. These generative neural networks model the pdf of samples observed in natural speech signals.

12.5 Variational autoencoders

The most appropriate ANN for speech coding is likely an *encoder/decoder* structure, where the initial network layers act as an encoder to automatically learn hidden latent features in a compressed representation, and then, ensuing layers act as decoder to form the reconstructed speech signal [17]. As in many coding schemes, the decoder steps often proceed in inverse fashion to the encoder steps. When this encoder-decoder is trained on unlabelled data, unlike ASR training supervised on texts, as in speech coding, it is an *autoencoder*. The encoding finds "hidden" vector representations called *encoder embeddings*, mappings from high to low dimensions, in a latent space, while the decoding is trained to match input and output data; the difference, or loss, may be mean square, as in SNR. The encoder often consists of bi- or unidirectional LSTM layers and embeds the transmission data. Bidirectional LSTM is not real time, as it processes data backwards in time. The *decoder* step generates output as close as possible to the original input.

A VAE is an autoencoder with encoding distribution regularized during training to ensure its latent space has useful properties to generate reasonable speech samples [4]. The VAE model uses a simple joint likelihood, $P(X, Z) = P(X|Z)\, P(Z)$, where X is the input vector

and *P (Z)* is the prior of Gaussian latent variable *Z*, with dimension much less than that of *X*. A decoder ANN designs *P (X|Z)*, which is not analytically tractable and approximated with a parametric variational distribution (inference model), with parameters from another ANN (encoder). To minimize computation, the encoder pdf is usually Gaussian with a diagonal covariance matrix, which implies zero correlation among parameters; such is a very common simplifying assumption to reduce computation, even if a poor model for real data.

Autoencoder-based waveform codecs may encode a speech waveform directly into a discrete latent space using a VQ-VAE [28]. They may rely on objective loss functions, e.g., mean-squared error, which cause perceptual distortion in decoded signals, as such loss functions rarely incorporate all relevant perceptual aspects. Perceptually, more meaningful loss functions calibrate the loss with psychoacoustic weighting [7, 37, 138]. This approach exploits the irrelevance of masked signal components. The loss function may be augmented with the difference between the log mel spectra of the input and output samples, which then judges similarity both in time and frequency domains.

12.6 Generative adversarial network (GAN) speech coding

Contrasting true and false samples of data is common in the development of deep *generative* ML models, which synthesize data such as images and audio, e.g., variational autoencoders (VAEs) and also *generative adversarial networks* (GANs). GAN architecture [32] uses two networks: generator and discriminator. The generator produces data from a low-dimensional latent space, with some starting from a Gaussian noise vector; the discriminator learns to distinguish between "real" training data and "false" generator outputs. Generators train to maximize the discriminator's error rate, while discriminators minimize their error rate. GANs can represent phonetically or phonologically meaningful information. To generate, if one starts from a random initialization of the model weights, adversarial loss often leads to severe audio artifacts [88]. With more realistic initial models, adversarial training can direct the generated signal toward more naturalness.

The GAN-based models tend to reconstruct speech conditioned on mel spectrograms and can offer fast generation on graphic processing units — two orders of magnitude faster than ordinary processors. To handle different ranges of correlation in speech, GAN may use multi-scale and multi-period discriminators, trained adversarially [78].

An aim of GAN is to produce a wide variety of outputs. If, however, a generator finds a very good output, it may overly focus on that output; the discriminator may then learn to always reject that output, thus falling into a trap called *mode collapse* [99].

12.7 Autoregressive speech coding

Audio coders may offer several output quality options, depending on available bit rate and decoder power. Some recent high-quality coders, such as *WaveNet* [90], are too complex for many output devices, as they use tens of millions of network parameters, models too big and slow for real-time processing in a resource-constrained device. Such powerful generative models use pdfs conditioned on many, or indeed all, previous input signal samples. Utilizing such long signal histories blocks efficient use of parallel processing. WaveNet is a DNN with more than 25 layers and uses 100+ GFLOPS with a high latency of more than 400 ms.

To reduce complexity, WaveNet may use an autoregressive generative model [56] to represent complex speech conditional pdfs using a stack of dilated causal convolutions, with very large receptive fields and no pooling layers. Training maximizes the log-likelihood of data over the model parameters. With no recurrent connections, it is faster to train than RNNs. When using 8-bit mu-law data, training is much more efficient than use of linear 16-bit quantized samples. It also uses a custom activation function combining tanh and sigmoid nonlinearities, which produces better audio than when using the ReLU function. While much more costly, WaveNet has outperformed more traditional vocoders in speech quality, either using an existing vocoder bit stream or with a quantized learned representation set [11].

Replacing dilated CNNs with RNNs improved memory efficiency in SampleRNN [82], which relies on previous samples at different scales. However, it is difficult for these methods to handle conditional features not found in prior training, e.g., generate speech with F0 outside ranges observed in training data. One can distill WaveNet into a FFNN that can synthesize high-quality speech more efficiently, e.g., by using a WaveNet model as a "teacher" for a feedforward IAF model.

These waveform generation methods form an extreme form of autoregression, with thousands of samples predicted per second. Using such long-term conditioning is feasible during training, as the complete sequence of input samples is available and can be processed in parallel. At the generating stage, however, each input sample comes from the output distribution sequentially, thus allowing no parallel processing.

12.8 Flow-based speech coders

Unlike the coders in the last section, non-autoregressive speech decoders are either *flow-based* generative models, e.g., Parallel WaveNet and WaveGlow (90 layers) [99] or

GAN-based models, e.g., MelGAN [66], HiFiGAN [63], and StyleGAN [88]. Flow-based models can use parallel processing but are costly and model the joint speech sequence pdf directly. They can model raw waveforms by transforming Gaussian noise sequences of the same size in parallel, upsampling from low-dimension latent features through transposed convolutions. A normalizing flow is a transformation of a simple pdf, e.g., Gaussian noise, into a more complex distribution by invertible and differentiable mappings. Resulting sample densities can be evaluated by transforming back to the original distribution and using the product of the density of both samples.

Inverse autoregressive flows (IAF) [51] are hybrid models where elements of a high-dimensional observable sample can be generated in parallel (Parallel WaveNet [90]). IAF has similarities to GANs, with a "student" playing the role of generator and a "teacher" playing the role of discriminator, to train the student network on an approximation to the true likelihood. Unlike in GANs, the student here is not attempting to fool the teacher in an adversarial manner; rather, it cooperates by attempting to match the teacher's probabilities. While IAF networks can operate in parallel at the inference stage, the autoregression is costly.

12.9 Residual network coding

Other coding architectures use quantized features from different layers of an autoencoder network to code speech at different bitrates [96]. For example, *residual networks* (ResNet) use "short-cuts" to pass information from one layer directly to a successor layer, as a bypass, while another approach [138] cascades residuals across a series of DNN modules. *WaveRNN* [47, 57] uses a feature ResNet that has 10 residual blocks, each of which uses two 1×1 1-D convolutions, batch normalization, and ReLU activation functions. The feature ResNet is followed by a few stretch layers that upsample the features to 16-kHz output. By adding linear prediction to WaveRNN, LPCNet [130] can go as low as 1.6-kbps coding rate, based on a sparse GRU layer. WaveRNN also demonstrated possibilities for synthesizing at lower complexities compared to WaveNet. Lower complexity and real-time operation are possible with LPCNet [130], by including LPC's limited-range autoregression. Thus, hybrid neural speech coders are approaching competitiveness with traditional vocoders, showing examples of higher quality at low rates. As research in this field is very dynamic, there are no neural speech coding standards yet.

13 Discussion

The important features to replicate in speech signals have been known for decades: spectral envelope (especially resonances), F0 (and all its harmonics, which are multiples of F0), and phase. These ideas were addressed early in various adaptive and predictive algorithms that adjusted quantizer parameters and input to exploit both short- and long-term redundancies. The importance of harmonics in voiced phonemes led some coders (e.g., ATC, sinusoidal coders) to concentrate almost entirely on these spectral aspects. One major roadblock has often been phase, which derives in complex unintentional fashion by airflow in the VT. Reconstructed speech with minimum phase can be highly intelligible but unnatural. As early as 1990 saw ACELP, which was readily adopted by telephone networks, to maintain very good quality for narrowband transmission at 10 kbps. Various combinations of LPC, SBC, and ATC have made up the bulk of speech coding applications since then, even as demand has increased for wider bandwidth speech than the standard phone network.

The advent of neural speech coders has shown a viable way to lower bit rates further, by mimicking phase well inside complex nonlinear networks, trained with suitable loss functions. There is no further understanding of how phase behaves, or which aspects need to be preserved, but output speech quality can be improved at lower bit rates. The price to pay here is use of very large opaque networks.

Traditional speech coders either directly modelled time waveforms, using simple sampling that can be adapted dynamically at a frame rate or using a source-filter VT model that allows separation of excitation and filter effects. These are very intuitive, allowing explicit expert design, based on knowledge of human speech production and perception. Neural speech coders do not have any explicit speech model and replicate the waveform. Traditional waveform coders adapt coder aspects (e.g., quantizers and predictor filters) to features (e.g., amplitude and spectral envelope detail) of the input speech that are directly estimated by speech analysis. Neural coders instead use standard ANN components in various architectures, with a range of loss functions that attempt to control the training in ways that retain speech quality.

Neural approaches are opaque, as the models automatically train many millions of parameters, with various combinations of architectures, activations, and loss functions. The components, so far, are mostly CNNs and RNNs, with time ranges that can vary. Neural methods have great capacity to find latent patterns that expert designers may overlook, but the complexity of speech may exceed that found in other neural successes, e.g., image recognition. Relevant details in most images lie in shapes, contours, colors, and shadows, which are functions of physical objects and optical viewing that may be simpler than the indirect generation of speech. Speech instead derives from concepts in one's brain, transformed

into muscle commands, and then VT motion, creating a signal that involves phonetics and intonation in complex fashion. Unlike images, one does not have access to the original ideas in the brain.

It is difficult to posit an ultimate low coding rate for speech coders that preserves all aspects of human speech, including speaker identity and naturalness. Speech contains a wide variety of information, on a broad range of time and frequency scales and in nonuniform fashion. One can quantify the textual content of speech readily, in terms of phonemes per second. Such content is below 60 bps, simply calculated from an average of 12 phonemes/s and 32 phones/language, on average. However, natural speech is far more than just a phoneme sequence. Intonation, phase effects, and speaker status (identity, health, emotion) are all complex factors not easily handled with any modern coding method.

The techniques discussed here are all attempts to remove redundancies in speech efficiently, which can be modelled either parametrically or via waveform approximation methods. Such redundancies occur often owing to physical constraints of the vocal tract and its control, as well as auditory mechanisms in the listener.

14 Conclusion

This paper has examined the diverse ways that have been used to code speech for efficient digital transmission. Methods have evolved greatly over the last few decades, exploiting advances in knowledge, data, and the power of computers. Historically, one must acknowledge the great advance of the Fourier transform in the development of speech coders. The foundations of speech analysis lie in fundamental ideas of spectrum, based on the Fourier transform, used in analysis of many signals well beyond speech. This led to use of the spectrogram, which was the basis of all speech analysis until the late 1960s. Understanding of the spectral behavior of both the vocal tract and the inner ear was essential to major methods of representing speech efficiently for transmission. One can also thank speech science for emphasizing the importance of spectral resonances and harmonics for speech reconstruction that is intelligible and natural.

In the late 1960s, LPC was clearly a breakthrough for speech coding and is still used in modern telephony. Other spectral methods (SBC, ATC) remain popular, exploiting redundancies in time-frequency representations. The use of vector quantization has also helped greatly in reduction of speech bit rates.

The difficulty of understanding phase in speech waveforms has caused bit rates to remain relatively high for high-quality speech coding, until recent neural network approaches that were able to focus accurate representations using a combination of automatic error minimization with auditory perceptual models. Given the cost of current neural methods, it appears that CELP may remain a mainstay of speech coding for the near future, even as there is increasing use of wider bandwidth speech coders. Nonetheless, the recent rapid increase in quality of hybrid neural coders, combined with more computational efficiency, suggests a potential major change in commercial speech coders in the near future.

Abbreviations

ACELP	Algebraic code-excited linear prediction
ADC	Analog-to-digital conversion
AMR	Adaptive multi-rate
ANN	Artificial neural network
APCM	Adaptive pulse-code modulation
ADPCM	Adaptive differential pulse-code modulation
AR	Autoregressive
ASR	Automatic speech recognition
ATC	Adaptive transform coding
CE	Cross-entropy
CELT	Constrained Energy Lapped Transform
CNN	Convolutional neural network
DAC	Digital-to-analog conversion
DNN	Deep neural network
ETSI	European Telecommunications Standards Institute
F0	Fundamental frequency
FFNN	Feed-forward neural network
GAN	Generative adversarial network
GRU	Gated recurrent unit
GSM	Global system for mobiles
IAF	Inverse autoregressive flow
ITU	International Telecommunications Union
LPC	Linear predictive coding
LSP	Line spectral pairs
LSTM	Long- and short-term memory
MDCT	Modified discrete cosine transform
MFCC	Mel-frequency cepstral coefficients
MLP	Multilayer perceptron
MMSE	Minimum mean-square error
MP	Multi-pulse
MPEG	Moving Picture Experts Group
RNN	Recurrent neural network
SBC	Sub-band coding
SNR	Signal-to-noise ratio
T-F	Time-frequency
TIA	Telecommunications Industry Association
VAE	Variational autoencoder
VoIP	Voice-over-Internet protocol
VQ	Vector quantization
VT	Vocal tract

Acknowledgements
Not applicable

Author's contributions
All work is from the sole author. The author read and approved the final manuscript.

Funding
This work was supported by NSERC (Canada) (Grant No. 142610).

Availability of data and materials
Not applicable, i.e., no specific extra materials used

Declarations

Ethics approval and consent to participate
Yes, I approve. No subjects were used in this work.

Consent for publication
Yes, I approve.

Competing interests
The author declares no competing interests.

Received: 29 July 2022 Accepted: 25 January 2023
Published online: 07 February 2023

References

1. M. Anthony, P.L. Bartlett, P.L. Bartlett, *Neural network learning: theoretical foundations*, vol 9 (Cambridge University Press, 1999)
2. J.G. Beerends, C. Schmidmer, J. Berger, M. Obermann, R. Ullmann, J. Pomy, M. Keyhl, Perceptual objective listening quality assessment (POLQA), the third generation ITU-T standard for end-to-end speech quality measurement part i—temporal alignment. J. Audio Engineer. Soc. **61**(6), 366–384 (2013)
3. B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, K. Jarvinen, The adaptive multi- rate wideband speech codec (AMR-WB). IEEE .Transact Speech Audio Process **10**(8), 620–636 (2002)
4. X. Bie, L. Girin, S. Leglaive, T. Hueber, X. Alameda-Pineda, A benchmark of dynamical variational autoencoders applied to speech spectrogram modeling. Interspeech, 46–50 (2021)
5. K. Brandenburg, in *Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*. MP3 and AAC explained (Audio Engineering Society, 1999)
6. A. Buzo, A. Gray, R.M. Gray, J. Markel, Speech coding based upon vector quantization. IEEE Transact. Acoustics Speech Signal Process. **28**(5), 562–574 (1980)
7. J. Byun, S. Shin, Y. Park, J. Sung, S. Beack, Development of a psychoacoustic loss function for the deep neural network (DNN)-based speech coder. Interspeech, 1694–1698 (2021)
8. J.H. Chen, R.V. Cox, Y.C. Lin, N. Jayant, M.J. Melchner, A low-delay CELP coder for the CCITT 16 kb/s speech coding standard. IEEE J. Select. Areas Commu. **10**(5), 830–849 (1992)
9. J.H. Chen, A. Gersho, Real-time vector APC speech coding at 4800 bps with adaptive postfiltering. ICASSP, 2185–2188 (1987)
10. K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation. Empirical Methods Natural Language Process. (2014)
11. J. Chorowski, R.J. Weiss, S. Bengio, A. Van Den Oord, Unsupervised speech representation learning using WaveNet autoencoders. IEEE/ACM Transact. Audio Speech Language Process. **27**(12), 2041–2053 (2019)
12. R.E. Crochiere, S.A. Webber, J.L. Flanagan, Digital coding of speech in sub-bands. Bell. Syst. Tech. J **55**(8), 1069–1085 (1976)
13. S.B. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. ASSP **28**, 357–366 (1980)
14. J.E. Dayhoff, *Neural network architectures: an introduction* (Van Nostrand Reinhold Co., 1990)
15. B. De Boer, W. Tecumseh Fitch, Computer models of vocal tract evolution: an overview and critique. Adapt. Behav. **18**(1), 36–47 (2010)
16. J.R. Deller, J.H.L. Hansen, J.G. Proakis, *Discrete-time processing of speech signals* (Prentice Hall, 1993)
17. L. Deng, M.L. Seltzer, D. Yu, A. Acero, A.R. Mohamed, G. Hinton, Binary coding of speech spectrograms using a deep auto-encoder. Interspeech (2010)
18. M. Dietz et al., Overview of the EVS codec architecture. ICASSP, 5698–5702 (2015)
19. Y. Dissen, J. Goldberger, J. Keshet, Formant estimation and tracking: a deep learning approach. J. Acoust. Soc. Am. **145**(2), 642–653 (2019)
20. H. Dudley, Phonetic pattern recognition vocoder for narrow-band speech transmission. J. Acoust. Soc. Am. **30**(8), 733–739 (1958)
21. G. Fant, *Acoustic theory of speech production* (Walter de Gruyter, 1970)
22. W.T. Fitch, The evolution of speech: a comparative review. Trends cognitive sci **4**, 258–267 (2000)
23. J.L. Flanagan, *Speech analysis synthesis and perception*, vol 3 (Springer Science & Business Media, 2013)
24. J.L. Flanagan, R.M. Golden, Phase vocoder. Bell. Syst. Tech. J. **45**(9), 1493–1509 (1966)
25. H. Fletcher, R.H. Galt, The perception of speech and its relation to telephony. J. Acoust. Soc. Am. **22**(2), 89–151 (1950)
26. C.A. Fowler, Coarticulation and theories of extrinsic timing. J. Phonetics **8**(1), 113–133 (1980)
27. K. Fukunaga, *Introduction to statistical pattern recognition* (Elsevier, 2013)
28. C. Garbacea, A. van den Oord, Y. Li, F.S.C. Lim, A. Luebs, O. Vinyals, T.C. Walters, Low bit-rate speech coding with VQ-VAE and a WaveNet decoder. ICASSP, 735–739 (2019)
29. J.D. Gibson, Speech coding methods, standards, and applications. IEEE Circuits. Syst. Magazine **5**(4), 30–49 (2005)
30. F. Girosi, M. Jones, T. Poggio, Regularization theory and neural networks architectures. Neural computation **7**(2), 219–269 (1995)
31. B. Gold, C. Rader, The channel vocoder. IEEE Transact. Audio Electroacoustics **15**(4), 148–161 (1967)
32. I. Goodfellow, Y. Bengio, A. Courville, *Deep learning* (MIT Press, Cambridge, 2016)
33. R. Gray, Vector quantization. IEEE ASSP Magazine **1**(2), 4–29 (1984)
34. D. Griffin, J. Lim, Signal estimation from modified short-time Fourier transform. IEEE Transact Acoustics Speech Signal Process **32**(2), 236–243 (1984)
35. M. Hasegawa-Johnson, A. Alwan, Speech coding: fundamentals and applications. Encyclopedia of Telecommunications (2003)
36. H. Hermansky, N. Morgan, RASTA processing of speech. IEEE Transact. Speech Audio Process. **2**(4), 578–589 (1994)
37. J. Herre, S. Dick, Psychoacoustic models for perceptual audio coding—a tutorial review. Applied Sci. **9**(14), 1–22 (2019)
38. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
39. Y. Horii, Fundamental frequency perturbation observed in sustained phonation. J. Speech Language Hearing Res. **22**(1), 5–19 (1979)
40. Y. Hu, P.C. Loizou, Evaluation of objective quality measures for speech enhancement. IEEE Transact Audio Speech Language Process. **16**(1), 229–238 (2007)
41. ISO/IEC23003–3 (2012). "MPEG-D (MPEGaudiotechnologies), Part 3: Unified Speech And Audio Coding."
42. ITU, *Pulse code modulation (PCM) of voice frequencies* (Recommendation, C.C.I.T.T, 1988)
43. ITU, *Method for the subjective assessment of intermediate quality level of audio systems* (International Telecommunication Union Radiocommunication Assembly, 2014, 2014)
44. A.K. Jain, J. Mao, K.M. Mohiuddin, Artificial neural networks: a tutorial. Computer **29**(3), 31–44 (1996)
45. K. Jarvinen, J. Vainio, P. Kapanen, T. Honkanen, P. Haavisto, R. Salami, C. Laflamme, J.P. Adoul, GSM enhanced full rate speech codec. ICASSP, 771–774 (1997)
46. N.S. Jayant, J.D. Johnston, Y. Shoham, Coding of wideband speech. Speech Comm. **11**(2-3), 127–138 (1992)
47. N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, K. Kavukcuoglu, Efficient neural audio synthesis. Int Conference Machine Learning. PMLR, 2410–2419 (2018)
48. S. Kankanahalli, End-to-end optimized speech coding with deep neural networks. ICASSP, 2521–2525 (2018)
49. H. Kawahara, I. Masuda-Katsuse, A. de Cheveigné, Restructuring speech representations using a pitch-adaptive time frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds. Speech Commun. **27**(3-4), 187–207 (1999)
50. P. Kenny, G. Boulianne, P. Ouellet, P. Dumouchel, Joint factor analysis versus eigenchannels in speaker recognition. IEEE Transact. Audio Speech Language Process. **15**(4), 1435–1447 (2007)

51. D.P. Kingma, T. Salimans, M. Welling, Improving variational inference with inverse autoregressive flow. arXiv preprint arXiv, 1606.04934 (2016)
52. D.H. Klatt, Software for a cascade/parallel formant synthesizer. J. Acoust. Soc. Am. **67**(3), 971–995 (1980)
53. W.B. Kleijn, F.S. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, T.C. Walters, WaveNet based low rate speech coding. ICASSP, 676–680 (2018)
54. W.B. Kleijn, A. Storus, M. Chinen, T. Denton, F.S. Lim, A. Luebs, J. Skoglund, H. Yeh, Generative speech coding with predictive variance regularization. ICASSP, 6478–6482 (2021b)
55. W.B. Kleijn, A. Storus, M. Chinen, T. Denton, F.S. Lim, A. Luebs, J. Skoglund, H. Yeh, Generative speech coding with predictive variance regularization. ICASSP, 6478–6482 (2021c)
56. W.B. Kleijn, A. Storus, M. Chinen, T. Denton, F.S.C. Lim, A. Luebs, J. Skoglund, H. Yeh, Generative speech coding with predictive variance regularization. ICASSP, 6478–6482 (2021a)
57. J. Klejsa, P. Hedelin, R.C. Zhou, R. Fejgin, L. Villemoes, High- quality speech coding with sample RNN. ICASSP, 7155–7159 (2019)
58. F. Klingholz, The measurement of the signal-to-noise ratio (SNR) in continuous speech. Speech Comm. **6**(1), 15–26 (1987)
59. H.P. Knagenhjelm, W.B. Kleijn, Spectral dynamics is more important than spectral distortion. ICASSP, 732–735 (1995)
60. D.E. Knuth, Dynamic Huffman coding. J. Algorithms **6**(2), 163–180 (1985)
61. M.A. Kohler, A comparison of the new 2400 bps MELP federal standard with other standard coders. ICASSP, 1587–1590 (1997)
62. A.M. Kondoz, *Digital speech: coding for low bit rate communication systems* (John Wiley & Sons, 2005)
63. J. Kong, J. Kim, J. Bae, Hifi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. Adv. Neural Info Process. Syst. **33**, 17022–17033 (2020)
64. S.G. Koolagudi, K.S. Rao, Emotion recognition from speech: a review. Int. J. Speech Technol. **15**(2), 99–117 (2012)
65. P. Kroon, E. Deprettere, R. Sluyter, Regular-pulse excitation--a novel approach to effective and efficient multipulse coding of speech. IEEE Transact Acoust. Speech Signal Process. **34**(5), 1054–1063 (1986)
66. K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W.Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, A.C. Courville, MelGAN: generative adversarial networks for conditional waveform synthesis. Adv. Neural Inform. Process. Syst. **32** (2019)
67. P. Ladefoged, S.F. Disner, *Vowels and consonants* (John Wiley & Sons, 2012)
68. C. Laflamme, R. Salami, R. Matmti, J.P. Adoul, *Harmonic-stochastic excitation (HSX) speech coding below 4 kbit/s* (ICASSP, 1996), pp. 204–207
69. M. Lahat, R. Niederjohn, D. Krubsack, A spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech. IEEE Transact. Acoust. Speech Signal Process. **35**(6), 741–750 (1987)
70. H.J. Landau, Sampling, data transmission, and the Nyquist rate. Proceedings of the IEEE **55**(10), 1701–1706 (1967)
71. Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series. Handbook Brain Theory Neural Networks **3361**(10), 1995 (1995)
72. W.J. Levelt, A. Roelofs, A.S. Meyer, A theory of lexical access in speech production. Behav. Brain Sci. **22**(1), 1–38 (1999)
73. Y. Li, X. Zhang, D. Chen, Csrnet: dilated convolutional neural networks for understanding the highly congested scenes. IEEE conference on computer vision and pattern recognition, 1091–1100 (2018)
74. A. Likas, N. Vlassis, J.J. Verbeek, The global k-means clustering algorithm. Pattern Recognition **36**(2), 451–461 (2003)
75. Y. Linde, A. Buzo, R. Gray, An algorithm for vector quantizer design. IEEE Transact. Comm. **28**(1), 84–95 (1980)
76. R. Lippmann, An introduction to computing with neural nets. IEEE ASSP Mag. **4 (2)** (1987)
77. R.P. Lippmann, Review of neural networks for speech recognition. Neural computation **1**(1), 1–38 (1989)
78. R. Lotfidereshgi, P. Gournay, *Practical cognitive speech compression* (IEEE Data Science and Learning Workshop, 2022)
79. J. Makhoul, Spectral analysis of speech by linear prediction. IEEE Transact. Audio Electroacoustics, 140–148 (1973)
80. J. Makhoul, M. Berouti, Adaptive noise spectral shaping and entropy coding in predictive coding of speech. IEEE Transact. Acoust. Speech Signal Process. **27**(1), 63–73 (1979)
81. J. Markel, The SIFT algorithm for fundamental frequency estimation. IEEE Transact. Audio Electroacoustics **20**(5), 367–377 (1972)
82. S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, Y. Bengio, SampleRNN: an unconditional end-to-end neural audio generation model. arXiv, 1612.07837 (2017)
83. P. Mermelstein, Articulatory model for the study of speech production. J. Acoust. Soc. Am. **53**(4), 1070–1082 (1973)
84. M. Minsky, S. Papert, *Perceptrons* (Press, M.I.T, 1969)
85. S. Möller, N. Côté, V. Gautier-Turbin, N. Kitawaki, A. Takahashi, Instrumental estimation of E-model parameters for wideband speech codecs. EURASIP J. Audio Speech Music Process., 1–16 (2010)
86. M. Morise, F. Yokomori, K. Ozawa, WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. Proc. IEICE Trans. Inf. Syst. **E99-D, no. 7**, 1877–1884 (2016)
87. M. Mouly, M.B. Pautet, *The GSM system for mobile communications* (Telecom publishing, 1992)
88. A. Mustafa, N. Pia, G. Fuchs, StyleMelGAN: an efficient high-fidelity adversarial vocoder with temporal adaptive normalization. ICASSP, 6034–6038 (2021)
89. H. Nishizaki, Data augmentation and feature extraction using variational autoencoder for acoustic modeling. Asia-Pacific Signal Inform Process Assoc Annual Summit Conference, 1222–1227 (2017)
90. A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg, N. Casagrande, Parallel WaveNet: fast high-fidelity speech synthesis. Int. Conference Machine Learn, 3918–3926 (2018)
91. M. Ostendorf, B. Favre, R. Grishman, D. Hakkani-Tur, M. Harper, D. Hillard, J. Hirschberg, H. Ji, J.G. Kahn, Y. Liu, S. Maskey, Speech segmentation and spoken document processing. IEEE Signal Processing Magazine **25**(3), 59–69 (2008)
92. M. Paez, T. Glisson, Minimum mean-squared-error quantization in speech PCM and DPCM systems. IEEE Transact Comm **20**(2), 225–230 (1972)
93. K.K. Paliwal, L. Alsteris, *Usefulness of phase spectrum in human speech perception* (European Conference on Speech Communication and Technology, 2003)
94. K.K. Paliwal, W.B. Kleijn, Quantization of LPC parameters. Speech Coding Synthesis **1**(1), 433–466 (1995)
95. V. Peddinti, Y. Wang, D. Povey, S. Khudanpur, Low latency acoustic modeling using temporal convolution and LSTMs. IEEE Signal Processing Letters **25**(3), 373–377 (2017)
96. D. Petermann, S. Beack, M. Kim, Harp-Net: hyper-autoencoded reconstruction propagation for scalable neural audio coding. IEEE Workshop App. Signal Process Audio Acoust. (WASPAA), 316–320 (2021)
97. J.C. Pinheiro, D.M. Bates, Approximations to the log-likelihood function in the nonlinear mixed-effects model. J. Comput. Graph Stat. **4**(1), 12–35 (1995)
98. A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.N. Hsu, A. Mohamed, E. Dupoux, Speech resynthesis from discrete disentangled self-supervised representations. arXiv preprint arXiv, 2104.00355 (2021)
99. R. Prenger, R. Valle, B. Catanzaro, WaveGlow: a flow-based generative network for speech synthesis. ICASSP, 3617–3621 (2019)
100. L. Rabiner, On the use of autocorrelation analysis for pitch detection. IEEE Transact. Acoust. Speech Signal Process. **25**(1), 24–33 (1977)
101. S. Ragot, B. Kovesi, R. Trilling, D. Virette, N. Duc, D. Massaloux, S. Proust, B. Geiser, M. Gartner, S. Schandl, H. Taddei, ITU-T G.729.1: an 8-32 kbit/s scalable coder interoperable with g. 729 for wideband telephony and voice over IP. ICASSP, IV-529 (2007)
102. V. Ramamoorthy, N.S. Jayant, R.V. Cox, M.M. Sondhi, Enhancement of ADPCM speech coding with backward-adaptive algorithms for post-filtering and noise feedback. IEEE J. Select Areas Comm. **6**(2), 364–382 (1988)
103. D. Reddy, L. Erman, R. Neely, A model and a system for machine recognition of speech. IEEE Transact. Audio Electroacoustics **21**(3), 229–238 (1973)
104. A.W. Rix, J.G. Beerends, M.P. Hollier, A.P. Hekstra, Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. ICASSP, 749–752 (2001)
105. A.E. Rosenberg, R.W. Schafer, L.R. Rabiner, Effects of smoothing and quantizing the parameters of formant-coded voiced speech. J. Acoust. Soc. Am **50**, 1532–1538 (1971)

106. S. Roucos, R. Schwartz, J. Makhoul, A segment vocoder at 150 b/s. ICASSP, 61–64 (1983)
107. D. Rowe, *Codec 2 - open source speech coding at 2400 bits/s and below* (TAPR and ARRL 30th Digital Communications Conference, 2011), pp. 80–84
108. R. Salami, C. Laflamme, J.P. Adoul, A. Kataoka, S. Hayashi, T. Moriya, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, Y. Shoham, Design and description of CS-ACELP: a toll quality 8 kb/s speech coder. IEEE Transact. Speech Audio Process. **6**(2), 116–130 (1998)
109. R. Salami, C. Laflamme, J.P. Adoul, D. Massaloux, A toll quality 8 kb/s speech codec for the personal communications system (PCS). IEEE Transact. Vehicular Tech. **43**(3), 808–816 (1994)
110. D. Scherer, A. Müller, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition. Int Conference Artificial Neural Net., 92–101 (2010)
111. M.R. Schroeder, Vocoders: analysis and synthesis of speech. Proceedings of the IEEE **54**(5), 720–734 (1966)
112. M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks. IEEE Transact. Signal Process. **45**(11), 2673–2681 (1997)
113. C.E. Shannon, A mathematical theory of communication. Bell. Syst. Tech. J **27**(3), 379–423 (1948)
114. N. Sharma, T.V. Sreenivas, Sparse signal reconstruction based on signal dependent non-uniform samples. ICASSP, 3453–3456 (2012)
115. S. Sharma, S. Sharma, A. Athaiya, Activation functions in neural networks. Int. J. Engineer App. Sci. Tech. **4**, 310–316 (2020)
116. J. Shore, R. Johnson, Properties of cross-entropy minimization. IEEE Transact. Inform. Theory **27**(4), 472–482 (1981)
117. C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning. J. Big. Data **6**(1), 1–48 (2019)
118. J. Skoglund, J.-M. Valin, Improving Opus low bit rate quality with neural speech synthesis. arXiv preprint arXiv, 1905.04628 (2019)
119. F. Soong, B. Juang, Line spectrum pair (LSP) and speech data compression. ICASSP, 37–40 (1984)
120. A.S. Spanias, Speech coding: a tutorial review. Proceedings of the IEEE **82**(10), 1541–1582 (1994)
121. R.C. Streijl, S. Winkler, D.S. Hands, Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives. Multimedia Systems **22**(2), 213–227 (2016)
122. M. Studdert-Kennedy, Speech perception. Contemporary Issues Exp. Phonetics, 243–293 (1976)
123. G.J. Sullivan, T. Wiegand, Rate-distortion optimization for video compression. IEEE Signal Processing Magazine **15**(6), 74–90 (1998)
124. C.H. Taal, R.C. Hendriks, R. Heusdens, J. Jensen, J., *A short-time objective intelligibility measure for time-frequency weighted noisy speech* (ICASSP, 2010)
125. S.I. Tamura, M. Tateishi, Capabilities of a four-layered feedforward neural network: four layers versus three. IEEE Transact. Neural Networks **8**(2), 251–255 (1997)
126. P. Taylor, *Text-to-speech synthesis* (Cambridge University Press, 2009)
127. Y. Tohkura, F. Itakura, S. Hashimoto, Spectral smoothing technique in PARCOR speech analysis-synthesis. IEEE Transact. Acoustics Speech Signal Process. **26**(6), 587–596 (1978)
128. P.P. Vaidyanathan, Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial. Proceedings of the IEEE **78**(1), 56–93 (1990)
129. J.-M. Valin, G. Maxwell, T. Terriberry, K. Vos, *High-quality, low-delay music coding in the Opus codec* (135th AES Convention, 2016)
130. J.-M. Valin, J. Skoglund, LPCNet: improving neural speech synthesis through linear prediction. ICASSP, 5891–5895 (2019)
131. S. Van Kuyk, W.B. Kleijn, R.C. Hendriks, On the information rate of speech communication. ICASSP, 5625–5629 (2017)
132. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need. Adv. Neural Inform. Process. Syst. **30** (2017)
133. Y. Wang, M. Vilermo, Modified discrete cosine transform: its implications for audio coding and error concealment. J. Audio Engineer. Soc. **51**(1/2), 52–61 (2003)
134. W. Weaver, C.E. Shannon, *The mathematical theory of communication*, vol 517 (University of Illinois Press, Champaign, IL, 1963)
135. D. Yu, L. Deng, *Automatic speech recognition*, vol 1 (Springer, Berlin, 2016)
136. R. Zelinski, P. Noll, Adaptive transform coding of speech signals. IEEE Transact. Acoust. Speech Signal Process. **25**(4), 299–309 (1977)
137. Z. Zhang, M. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels. Adv. Neural Inform. Process. Syst. **31** (2018)
138. K. Zhen, M.S. Lee, J. Sung, S. Beack, M. Kim, Psychoacoustic calibration of loss functions for efficient end-to-end neural audio coding. IEEE Signal Process. Letters **27**, 2159–2163 (2020)
139. E. Zwicker, G. Flottorp, S.S. Stevens, Critical band width in loudness summation. J. Acoust. Soc. Am. **29**(5), 548–557 (1957)

Publisher's Note