A Digital Channel Vocoder

THEODORE BIALLY, MEMBER, IEEE, AND WALTER M. ANDERSON

Abstract—An all digital realization of a channel vocoder is described, including both computational algorithms and hardware structures. The machine, constructed almost entirely of digital elements, operates at a 2400 bit/s rate and produces synthetic speech, whose intelligibility and speaker recognition properties are comparable to those of an average telephone line. The extensive use of integrated digital circuitry allows size and cost reductions that are not currently possible in conventional high-quality analog vocoder systems.

OVER THE past several years vocoder systems have evolved to the point where many of their objectionable characteristics have for the most part been eliminated. 2400-bit/s vocoders have been built, which perform well for a large class of speakers and in a variety of environments, and which exhibit intelligibility and speaker recognition properties that are similar to those of an average telephone line [1]. In spite of these improvements the use of vocoders in speech communication systems has been rather limited. This condition stems from the fact that the more sophisticated vocoder designs tend to be expensive to implement and are generally bulky in size.

Early in 1968 a program was initiated at the M.I.T. Lincoln Laboratory to develop a 2400-bit/s vocoder system that would be small and relatively inexpensive, while at the same time exhibiting speaker recognition and intelligibility characteristics comparable to those of existing high-quality vocoders. What has resulted is a nearly all digital channel vocoder that can be packaged in a fraction of a cubic foot using commercially available integrated circuits, and in considerably less space by using large-scale integration techniques.

In addition to taking advantage of the fact that present-day digital integrated circuits are both compact and economical, the vocoder has been designed to exploit the highly repetitious nature of channel vocoder hardware. Analog vocoder realizations typically consist of a large number of nearly identical circuits, which function in parallel to derive the required speech parameters. The same result can be obtained digitally through the use of a single high-speed digital arithmetic unit, which is multiplexed among the various circuit functions. In addition, the analysis and synthesis algorithms have been chosen such that the hardware requirements for their implementations are very nearly identical. The vocoder has been designed for half-duplex operation so that the same physical hardware is used in both operating modes. These approaches result in considerable size and cost economies.

Paper 70TP30-COM, approved by the Wire Communication Committee of the IEEE Communication Technology Group for publication without oral presentation. This work was sponsored by the Department of the Air Force. Manuscript received September 8, 1969.

T. Bially is with the M.I.T. Lincoln Laboratory, Lexington, Mass.

W. M. Anderson is with Applicon, Inc., Burlington, Mass.

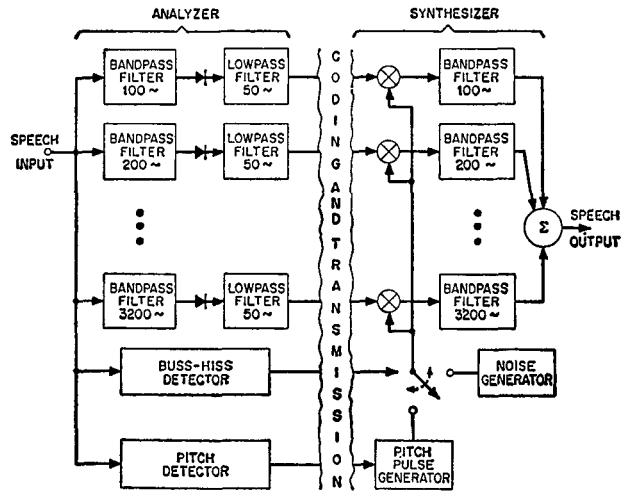

Fig. 1. Typical channel vocoder.

Vocoder Analyzer

Speech bandwidth compression is achieved in a vocoder system by extracting slowly varying parameters from the speech waveform. These parameters relate to the physical configuration of the vocal tract and to the character of its excitation. They convey sufficient information to enable one to recreate or synthesize an approximation to the original speech at the receiving end of the system. Essential speech parameters generally fall into two classes. The first, called excitation, includes a measurement of the fundamental frequency or pitch of voiced sounds, and an auxiliary (buzz–hiss) signal which indicates the presence or absence of voicing.

The second parameter class is concerned with the characterization of the vocal tract. In a channel vocoder the vocal cavity parameters are derived from the short-time Fourier-amplitude spectrum, as measured by a bank of contiguous bandpass filters, detectors, and low-pass filters, as shown in Fig. 1. The rectifiers and low-pass filters serve to envelope–detect the bandpass outputs and to smooth out variations in the spectral measurement, which are due to the periodic nature of the exciting waveform.

An equivalent method of realizing a vocoder channel that is particularly well suited to digital implementation is shown in Fig. 2(a). Speech is simultaneously modulated by two quadrature sine waves of the same frequency, and

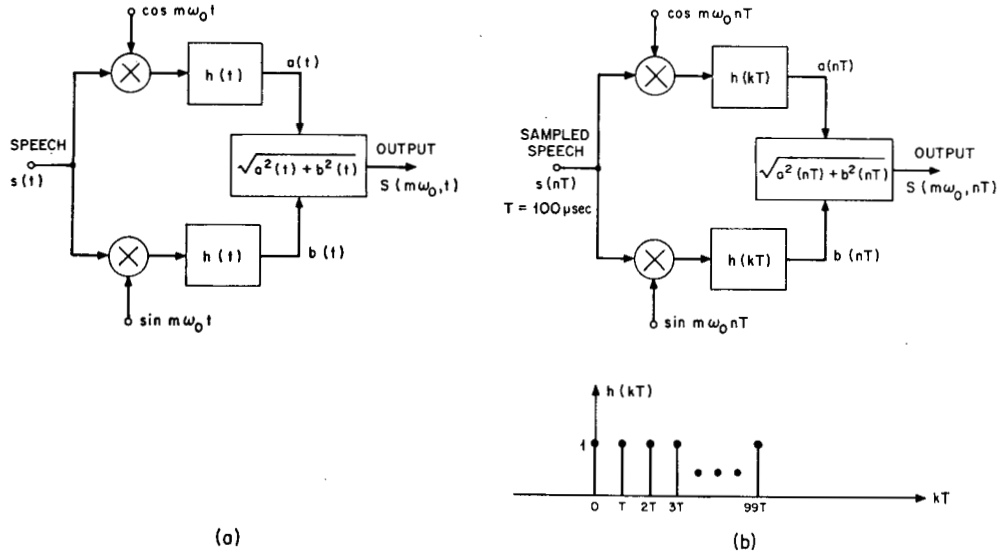

Fig. 2. Alternate channel realization.

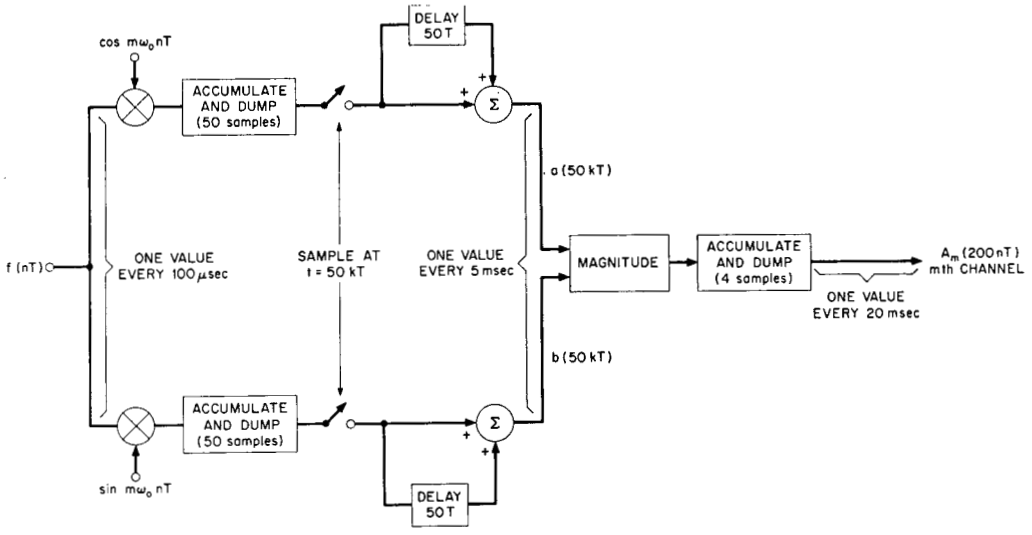

Fig. 3. Complete analyzer channel.

the resultant waveforms are separately applied to identical low-pass filters. The square root of the sum of the squares of the two filter outputs is a positive waveform, whose value at any time is the magnitude of the Fourier transform of the product of the past of the input waveform with the time-reversed impulse response of the low-pass filter, evaluated at the frequency of the modulating sinusoids. If, as indicated in Fig. 2(b), the structure is implemented digitally, and if the unit sample response of the low-pass filters is a square pulse of a 100-sample duration, then the output of the filter will be the magnitude of the discrete Fourier transform of the most recent 100 input speech samples, evaluated at the frequency $m\omega_0$. The configuration shown in Fig. 2(b) should be followed by a digital low-bass filter to complete the vocoder analyzer channel. The low-pass filter, as in the analog vocoder, serves to smooth out excitation-related spectral variations.

At this point two factors can be exploited to effect computational and hardware savings. First, the filtered spectral data will eventually be sampled for transmission only once every 20 ms, so that a complete spectral computation at the system sampling rate of 100 μs is rather wasteful. Second, and more important, is the fact that the magnitude of the spectrum varies relatively slowly and is, in fact, a low-pass quantity which is band limited to about 50 Hz. Instead of processing one sample of the spectral magnitude in each channel every 100 μs one can, in view of the previous discussion, sample the spectrum only once every 5 ms without losing any significant data.

Fig. 3 depicts a complete analyzer channel. The contents of the accumulators immediately to the right of the multipliers are transferred to delay registers and then cleared after every 50 samples (5 ms). The resultant signal at point z is identical to that which would have

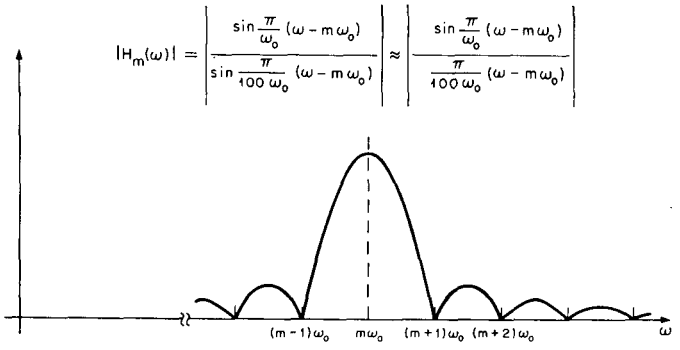

Fig. 4. Channel function transfer.

been obtained had 5 ms samples been taken of the filter of Fig. 2(b). The square root of the sum of the squares is computed once each 5 ms and four successive spectral magnitude samples are then summed to yield a smoothed estimate of the spectrum every 20 ms. The overall channel transfer function is shown in Fig. 4. All vocoder channels are identical in all parameters, except for the frequencies of the modulating sine waves. This leads, as will be shown, to an efficient arithmetic structure that can be multiplexed with a minimum of control.

32 identical channels spaced 100 Hz apart cover the range from 100 to 3200 Hz. For the particular filter design in question the operation of the analyzer can be viewed as a discrete Fourier transform of the input speech. The transform of the most recent 100 input samples is measured, frequency components at zero and above 3200 Hz are ignored, and phase information is discarded. The 20-ms spectral measure is then obtained by averaging four successive 100-sample discrete Fourier transforms, where each transform is made on data which overlaps that of its successor by 50 samples (5 ms). It should be noted that the discarded frequency components, as well as the phase information, contribute little to speech quality and intelligibility and may therefore be ignored.

Analyzer Hardware Configuration

As mentioned earlier, one of the more attractive features of a digital vocoder realization is the possibility of time-sharing a single arithmetic unit among all the filter elements in the analyzer bank. The 32 bandpass filters of Fig. 3 can be implemented in this manner with the structure of Fig. 5. In addition to a read-only memory, an address-control generator, an adder, and a multiplier, the figure indicates two circulating memory banks, each of 64 words. The memories are organized in the form of a drum; a number of 64-stage serial shift registers operating in parallel, so that entire words are simultaneously shifted through the structure.

Two data paths are shown in the diagram, only one of which is active at any time. Consider first the dotted route. A sample of a sine or cosine wave is taken from the read-only memory and is multiplied by the current sample of the input speech signal. The product is added to the contents of one of the upper 64 circulating registers. Exactly

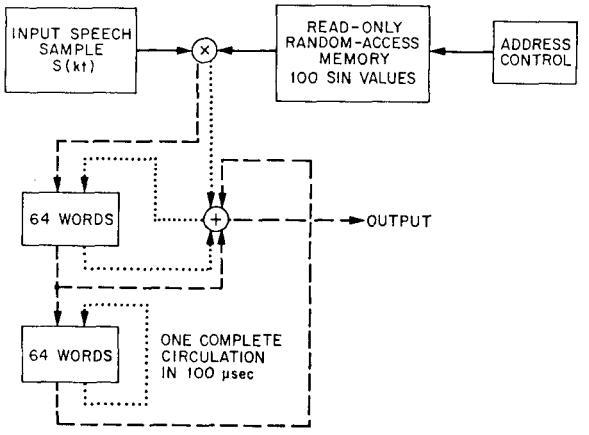

Fig. 5. Implementation of 64 bandpass integrate and dump filters.

100 μs later, when the next speech sample is available, this same register is again incremented. By addressing the read-only memory in the proper sequence, it is possible to maintain 64 independent accumulators in the upper loop. These are the accumulators for the 32 analyzer bandpass filters. Note that the lower loop merely circulates without change during the previously described operations. After 50 circulations have elapsed the configuration of the hardware is changed to that indicated by the dashed path. The contents of each register in the upper memory is added to the contents of one of the lower registers to form the output. The upper memory is then transferred to the lower loop, and a new accumulation is begun in the upper section. The synchronization of the machine is such that the lower 64 registers act as the delay elements of Fig. 2(b).

Some description of the read-only memory is in order at this point. In order to develop the reference sinusoids required for the filter bank, a full cycle of a sine wave is stored in 100 words of the memory. Since a new sample is processed by each filter once every 100 μs, these 100 words represent one cycle of a 100-Hz reference signal, the lowest frequency in the system. In order to obtain a 200-Hz reference signal, one reads every second value in the table every 100 μs. Similarly, to obtain a $100n$-Hz reference signal, every nth table value is read. Of course, the table accessing is done modulo 100. Thus at time kT and for the nth channel, the reference sample is that which is found at address

$$[n \times k]_{\text{modulo 100}}$$

in the read-only memory. If the table represents a sine-wave cycle, then the cosine reference value for the nth channel is found at address

$$[n \times k + 25]_{\text{modulo 100}}.$$

With the addition of a separate device which computes $(x^2 + y^2)^{1/2}$, and with an increase in the length of the shift registers, the same structure can also be used to do the required low-pass smoothing of the spectral amplitude signals. As shown in Fig. 6, the shift registers are increased

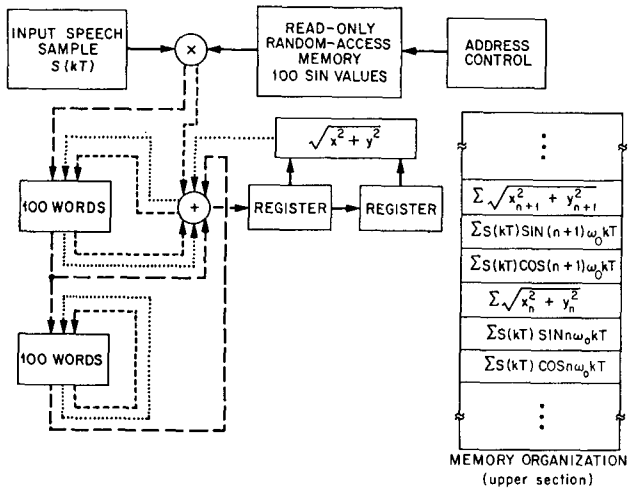

Fig. 6. Complete analyzer structure.

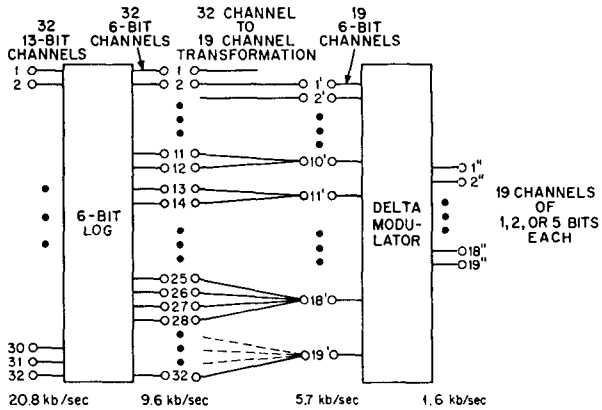

Fig. 7. Spectrum coding for 2400 bit/s transmission.

in length to 100 bits. Actually, only 96 are required; the 100-bit length is needed during synthesis. The final structure can assume any of three configurations, as shown. Data in the memory is organized into three-word groupings, in which every third register performs a spectral smoothing operation. The configuration of the machine is sequentially switched between its three states such that after every 20 ms every third register in the circulating memory contains the final spectral measurement for a particular channel. These parameters are then encoded for transmission at 2400 bit/s.

It might be mentioned that the square root of the sum of the squares is approximated by

$$(x^2 + y^2)^{1/2} \simeq \begin{cases} |x| + |y|/2, & \text{if } |x| > |y| \\ |y| + |x|/2, & \text{if } |y| > |x|. \end{cases}$$

This approximation, while not overly accurate, is within 8.6 percent of $1.06(x^2 + y^2)^{1/2}$. The factor of 1.06 is unimportant in this application since it merely affects the filter gains. In addition, the 8.6-percent maximum computation error is considerably smaller than the quantization error that is introduced in the encoding algorithm.

Coding for 2400 bit/s Operation

Fig. 7 shows the coding operations necessary for efficient transmission of the vocoder spectrum. Since there are 32 channels to be sampled 50 times per second, and excitation data (pitch and buss–hiss) to be sampled 100 times per second, rather extensive compression is needed to fit the data into a 2400 bit/s format. The coding is based on three facts: the spectral measures are highly correlated, the frequency discrimination of the ear of the listener is less at higher frequencies, and the ear responds logarithmically to sound intensity.

The logarithmic sensitivity of the ear allows a logarithmic compression of the channel signals, reducing the channel data from 13-bit words to 6-bit words. Next, since the frequency discrimination of the ear is greater at low frequencies, the higher frequency channels can be averaged together by 2's and 4's to reduce the number of channels to 19. Finally, the remaining 19 channels are highly correlated; a channel-to-channel delta-modulation scheme exploits this correlation and permits more efficient quantization and encoding. The net result of the three stages of coding is a 32-bit representation of the entire spectrum, to which is appended two 8-bit excitation parameters to form a 48-bit frame each 20 ms.

It might be pointed out that the combining of adjacent equal-bandwidth high-frequency channel signals leads to a structure that is somewhat akin to a 19-channel vocoder with nonequal-bandwidth filters. It is computationally more efficient to first compute 32 equal-bandwidth parameters and then selectively combine them, rather than to realize nonuniform filter characteristics directly.

Pitch Detector

The pitch-detection algorithm is a simplified version of one developed by Gold [2]. Briefly, four independent pitch measurements are made simultaneously, and a scoring strategy is employed to decide which of the four represents the actual pitch period.

As shown in Fig. 8, speech is applied first to two bandpass filters, and then to four identical elementary pitch detectors. Since the elementary pitch detectors act only on the positive peak values of their inputs, each filter provides two possible waveforms, the actual output and the negative of the output. Both filters are of the linear-phase Bessel type and are built using analog hardware, as are the four elementary pitch detectors.

An elementary pitch detector is shown in Fig. 9. Its operation is best described by means of the waveforms of Fig. 10. The positive peaks of the incoming signal are sampled and applied to a holding circuit. 3 ms after a peak is sensed, the stored value begins to decay exponentially. A new peak is applied only if its value exceeds that of the exponentially decaying waveform, and only if it occurs at least 3 ms after the previously accepted peak.

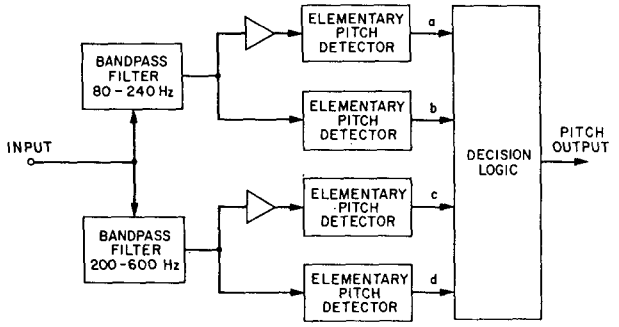

Fig. 8. Pitch detector organization.

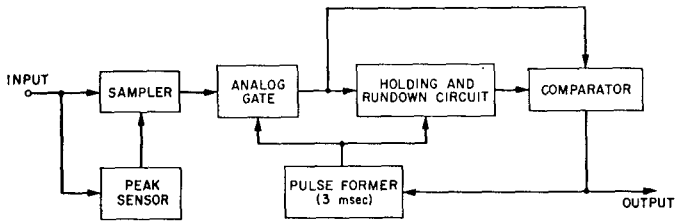

Fig. 9. Elementary pitch detector.

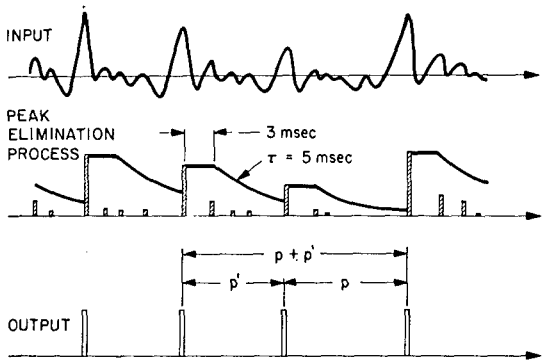

Fig. 10. Pitch detector waveforms.

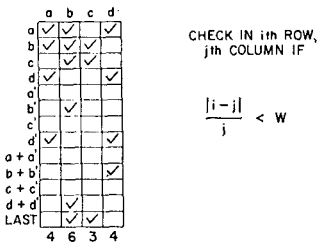

Fig. 11. Pitch scoring algorithm.

The output of the detector is a train of pulses which are coincident with those peaks of the incoming waveform, which were accepted by the holding circuit.

Three period measurements are associated with each detector output, as indicated in Fig. 10. Of the three, only the most recent period (P) is a candidate for the final pitch estimate; the other quantities are used for scoring purposes. Fig. 11 summarizes the scoring strategy. Each of the four current pitch candidates is compared against 13 quantities, itself included. 12 of the 13 are the three period measurements from each of the four elemental pitch detectors. The 13th is the pitch period which was selected in the previous measurement (10 ms before). As indicated in the figure, a pitch candidate receives a vote if the test period against which it is being compared is within a given percentage of the candidate value. The value of W (Fig. 11) that is actually used is $\frac{1}{8}$. The candidate that receives the highest score is taken as the current estimate of the pitch period. In the figure, candidate b, with a score of 6, will be the final choice.

If the winning score is less than 4, then noise excitation (hiss) is assumed. Thus the scoring mechanism also serves as a buzz–hiss indicator. The scoring and selection portion of the machine is realized digitally, using circulating shift-register memory elements and a serial word organization. A new pitch measurement is made every 10 ms.

Synthesizer

As indicated in Fig. 1, both the analyzer and synthesizer of a conventional channel vocoder employ identical banks of bandpass filters. In an analog vocoder realization, one physical filter bank can serve both functions if the machine is of a half-duplex nature. In this digital vocoder, although the hardware behaves as a bank of filters during analysis, it cannot be used for synthesis without some modification. First, the analysis filter bank delivers low-pass outputs directly; i.e., there is no point in the structure of Fig. 2 at which the signal is truly bandpass. Second, the filter outputs are sampled rather infrequently in analysis, and the hardware design reflects this requirement. The synthesizer requires that the filter outputs be valid at each basic sampling instant, every 100 μs. To build such a bank of digital filters would require an inordinately large number of components and considerable expense. What is done instead, and which results in exactly the same output signal that would be obtained from a bank of filters, is to effect a direct convolution between the train of pitch pulses and the impulse response of the equivalent filter bank. This operation is accomplished with essentially the same hardware that the analyzer uses. The basic synthesizer is outlined in Fig. 12.

The incoming 2400 bit/s data stream is decoded to yield an excitation parameter once every 10 ms and a set of 32-channel amplitude values every 20 ms. The excitation data is used to generate a train of pitch pulses at the required rate. In the event that hiss excitation is called for, the pitch rate is held fixed at 1 kHz, and attenuated pitch

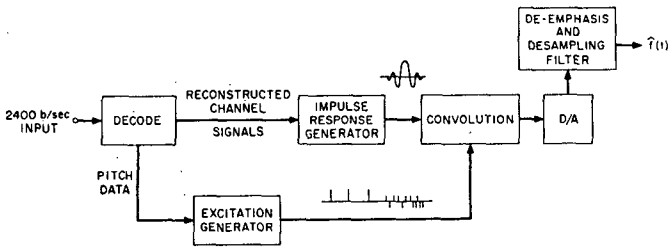

Fig. 12. Synthesizer structure.

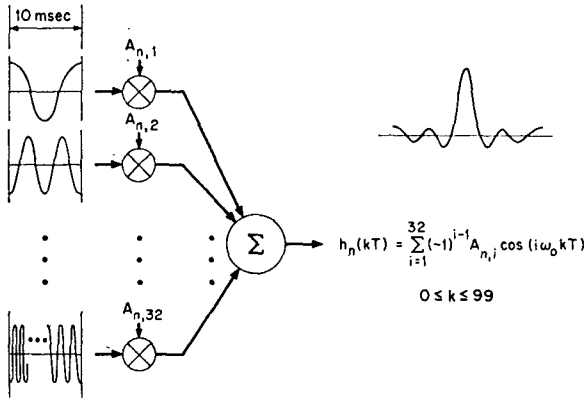

Fig. 13. Formation of impulse response.

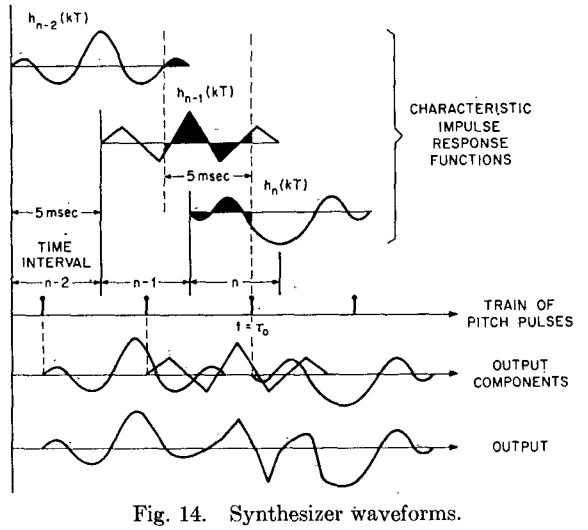

Fig. 14. Synthesizer waveforms.

pulses of random sign are produced. The 32 reconstructed channel signals are smoothed by linear interpolation to provide a new set once each 5 ms. Thus as far as the remainder of the synthesizer is concerned, new spectral data are available every 5 ms.

Suppose first that the synthesizer were indeed composed of 32 bandpass filters of frequency response identical to that of those in the analyzer. The response of one of these filters to a unit impulse would be a sinusoidal waveform of frequency equal to the center frequency of its passband and of exactly 10-ms duration. It follows then that the impulse response of a bank of these filters connected as shown in Fig. 1 is simply the weighted sum of 32 sinusoidal components, lasting for 10 ms. The weighting coefficients are the channel amplitude values. This is illustrated in Fig. 13, in which the nth impulse response is formed using the nth set of 32 channel coefficients. Note the net result is a sequence of 100 sample values which, since the sampling rate is 100 μs, yields a function of a 10-ms duration.

As an alternative to constructing a bank of filters, one can compute an impulse response function and convolve this directly with the train of pitch pulses. With reference to Fig. 14, a new impulse response of a 10-ms duration is computed every 5 ms. Each impulse response function is "characteristic" of a 5-ms time interval, as indicated. The generation of the impulse response function can be equated to a discrete Fourier transform, similar to the equivalence previously mentioned for the analyzer. It is equivalent to an inverse discrete Fourier transform of 100 frequency points, where unimportant frequency points have been omitted and phase information has been set to zero. The resulting 100-sample time function is circularly shifted by 50 samples (or equivalently a linear phase added) to allow the major peak to appear at the center of the waveform and minimize discontinuities at the edges. This is accomplished by including the phase-reversal term shown in Fig. 13. For each pitch pulse that occurs in a particular 5-ms interval, the characteristic impulse response for that interval, starting at the time that the pitch pulse occurs, is added to the output waveform. The final output at any particular time is the sum of the output component values at that time.

In order to establish the memory requirements of the convolution synthesizer, one can easily verify that the output at any time τ_0 is the sum of a number of points, all of which are contained in those portions of the characteristic impulse responses which, in Fig. 14, occur within the 5 ms immediately preceding τ_0. There are 100 such points; thus to form the output at any time one requires, at most, a knowledge of 100 values. Furthermore, each value is retained for 5 ms, after which it is replaced by a new one. It follows that the memory requirement is 100 registers that get continually updated at the average rate of one value every 50 μs. The more often the impulse response of the filter bank is updated, the better the quality of the synthesized speech and the more closely the synthesizer structure of Fig. 1 is approximated. The impulse response can be updated as often as desired with no increase in memory requirements, but the rate of computation increases with the rate of update. The 5-ms rate was chosen because it produces acceptable quality while matching the computation rate required in the analyzer. A hardware structure that performs the synthesis function is described in the following. Physically, it consists of the same components that constitute the analyzer filter bank.

Referring to Fig. 15, let the output accumulator (in the lower right-hand corner of the diagram) be cleared at time τ_0. Furthermore, let the 100-word memory contain those portions of the characteristic impulse-response waveforms which are required to form the next output sample.

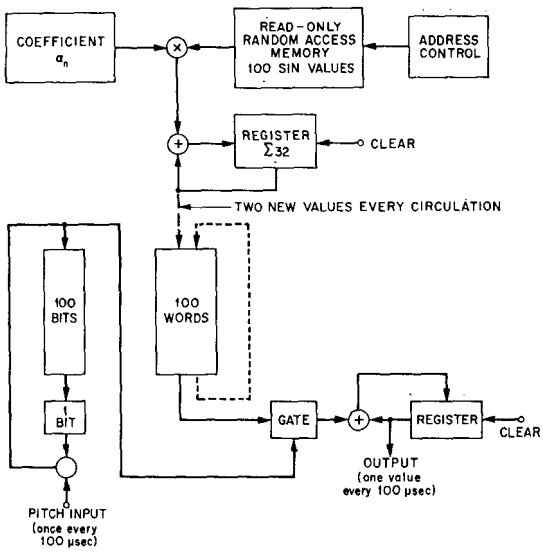

Fig. 15. Synthesizer hardware configuration.

If the memory makes one complete circulation in 100 μs, then each of its 100 words has the opportunity to be added to the output register before the next sample is generated. Which of these 100 points will be added is controlled by the input to the gate, which is shown to the left of the accumulator.

After each complete circulation of the memory a new output sample is produced, the accumulator is cleared, and the process begins again. Two words in the memory are updated in each circulation, so that the 100 values it contains always represent the 5 ms immediately preceding the current point in time. These are symbolized by the shaded portions of the characteristic impulse-response functions in Fig. 4. The values of the new points are computed in a parallel operation shown at the top of the diagram. 32 sine-wave values from the read-only memory are multiplied by the appropriate channel coefficients and added together to form a point on a characteristic impulse-response function.

Returning to the gate preceding the output accumulator, note that it is controlled by the output of a 101-stage circulating shift register. The operation is as follows. At the beginning of each circulation of the main memory, and in the absence of a pitch pulse at that instant, a 0 is introduced into the control shift register at the pitch input terminal. If a pitch pulse occurs at this time, a 1 is introduced. This control bit is also applied to the output gate. Thus if a pitch pulse occurs, then the first point of the current characteristic impulse response is added into the output register. One circulation plus one clock pulse later, this bit appears at the output of the control loop and is again applied to the gate. This time, however, it causes the second point of the characteristic impulse response to be added to the input. The control bit slips back one position with each circulation, and after it has gated 100 points into the accumulator, it appears at the output of the control loop simultaneously with the occurrence of a pitch input and is annihilated. The lifetime of a control bit is thus 100 samples or one impulse response long, as is required by the system. Unvoiced speech is synthesized in the same manner as the voiced portions, with the exception that the pitch period is fixed at 1 ms, pitch pulses are of reduced amplitude and occur with random sign.

The 100-bit section of the control loop is physically part of the 100-word memory on the right, and this in turn is the same physical piece of hardware that performs the analyzer function. The multiplier (of the array type) and read-only memory are of course common to both analyzer and synthesizer. It might be added that in reality the control bit shown in the left circulation loop of Fig. 15 is a 2-bit word. This conveys data relating to the sign and magnitude of the pitch pulses, as is required during the synthesis of unvoiced sounds.

Summary

An all digital vocoder analyzer and synthesizer and some of the hardware designs that are used in its implementation have been described. The hardware simplicity and economic advantages result primarily from the very simple nearly identical digital filters that are used, from the use of inexpensive digital memory elements, the time-sharing of arithmetic hardware, and the half-duplex nature of the machine. The net result is a vocoder that, while it is relatively inexpensive to implement and small in size, produces good-quality natural-sounding speech at a 2400 bit/s rate.

Two models of the machine are currently being constructed. One of these is a laboratory breadboard. The other occupies about half a cubic foot including its power supply and uses TTL dual in-line logic elements. These are interconnected by means of welded point-to-point wire connections on $3\frac{1}{4}$ by $3\frac{3}{4}$ inch cards. The circulating memory is composed of commercially available dual 100-stage dynamic metal-oxide-semiconductor (MOS) shift registers.

The design of the vocoder was greatly aided by a computer simulation of the entire system. Various analysis and synthesis configurations were tried in converging to a system which could take economic advantage of the digital hardware available and produce good quality speech. When an acceptable analysis–synthesis system was designed, the computer simulation was used to determine an efficient coding algorithm, and to resolve questions of required word lengths. During construction and debugging of the first system, the UNIVAC 1219 facility was used to provide synthesis of analyzer data, and to act as a source of synthesizer input, since the single vocoder could only be operated in the half-duplex mode.

References

[1] M. R. Schroeder, "Vocoders: analysis and synthesis of speech (A review of 30 years of applied speech research)," *Proc. IEEE*, vol. 54, pp. 720–734, May 1966.

[2] B. Gold "Description of a computer program for pitch detection," *Proc. 1962 International Congress on Acoustics*, Copenhagen, Denmark.

Theodore Bially (S'59–M'63) was born in Brooklyn, N. Y., on July 7, 1939. He received the B.E.E. degree from City College of New York, New York, in 1961 and the M.S.E.E. and Ph.D.(E.E.) degrees from the Polytechnic Institute of Brooklyn, Brooklyn, in 1962 and 1967, respectively.

From 1962 to 1963, he was a Lecturer in the Department of Electrical Engineering, City College of New York. From 1963 to 1967 he was employed by ITT Federal Laboratories and Electronic Instrument Company. Since 1967 he has been with the M.I.T. Lincoln Laboratory, Lexington, Mass.

Dr. Bially is a member of Eta Kappa Nu and Tau Beta Pi.

Walter M. Anderson was born in Providence, R. I., on June 29, 1942. He received the B.S.E.E., the M.S.E.E., and the E.E. degrees in 1964, 1966, and 1967, respectively, from the Massachusetts Institute of Technology, Cambridge.

From 1963 to 1967 he worked for the Precision Controls Department of Texas Instruments, Inc., on new product design of circuits and electromechanical devices. He joined the Surface Techniques and Equipment Group at M.I.T. Lincoln Laboratory, Lexington, Mass., in 1967 where he was involved in the design and simulation of speech communication systems. At present, he is with Applicon, Inc., Burlington, Mass., working on the development of interactive computer graphic design systems.

Mr. Anderson is a member of Eta Kappa Nu, Sigma Xi, and Tau Beta Pi.