Real-Time Digital Hardware Pitch Detector

JOHN J. DUBNOWSKI, RONALD W. SCHAFER, SENIOR MEMBER, IEEE, AND
LAWRENCE R. RABINER, SENIOR MEMBER, IEEE

Abstract—A high-quality pitch detector has been built in digital hardware and operates in real time at a 10 kHz sampling rate. The hardware is capable of providing energy as well as pitch-period estimates. The pitch and energy computations are performed 100 times/s (i.e., once per 10 ms interval). The algorithm to estimate the pitch period uses center clipping, infinite peak clipping, and a simplified autocorrelation analysis. The analysis is performed on a 300 sample section of speech which is both center clipped and infinite peak clipped, yielding a three-level speech signal where the levels are −1, 0, and +1 depending on the relation of the original speech sample to the clipping threshold. Thus computation of the autocorrelation function of the clipped speech is easily implemented in digital hardware using simple combinatorial logic, i.e., an up-down counter can be used to compute each correlation point. The pitch detector has been interfaced to the NOVA computer facility of the Acoustics Research Department at Bell Laboratories.

I. Introduction

Although a wide variety of pitch-detection algorithms have been proposed [1]-[6], as yet few of them have been built in special-purpose digital hardware capable of real-time operation. This is because most of the pitch-detection algorithms require either a great deal of logic, or an excessive amount of computation. Neither of these situations is conducive to inexpensive implementations in digital hardware. Although software versions of these pitch detectors are useful in some applications, there are many cases in which system requirements include real-time operation.[1] Examples of such systems include on-line systems for speaker verification and identification [7], and systems for helping to correct speech impediments of the handicapped [8].

For such applications where reliable, real-time pitch detection is a requirement, a digital hardware pitch detector has been built. The pitch detector operates in real time at a 10 kHz sampling rate. The hardware also computes energy as well as pitch-period estimates. The pitch and energy computations are performed 100 times/s, i.e., once per 10 ms interval.

The pitch-detection algorithm which was implemented in digital hardware is similar to the center-clipped autocorrelation method studied by Sondhi [1], but with one important modification. In the method proposed by Sondhi the speech is center clipped and then autocorrelated. In the hardware implementation the speech is both center clipped and infinitely peak clipped, thereby reducing the speech samples to two-bit data words. Thus computation of the autocorrelation function is simplified in complexity from a sum of products to a simple logical combination of two-bit data words. This modification to the Sondhi method serves to minimize both computation time as well as hardware complexity, thereby enabling the algorithm to be implemented in real time in special purpose digital hardware. A number of additional threshold parameters have been incorporated in the hardware for the following purposes: to help make a voiced-unvoiced decision, to adapt to the wide dynamic range of speech, and to distinguish speech from background silence. These parameters will all be described later in this paper.

In the next section a detailed discussion of the pitch-detection algorithm, and the various parameters which give the algorithm flexibility, is presented. In Section III the specific hardware structure is described. Finally, in Section IV a brief discussion of a performance evaluation of the algorithm is given.

II. The Pitch-Period Estimation Algorithm

One of the difficulties in making a reliable estimate of the pitch period across a wide range of speech utterances and speakers is the effect of the formant structure on measurements related to the periodicity of the waveform. Thus for reliable pitch detection, it is highly desirable that the effects of the formants be greatly reduced, or entirely eliminated, if possible. The technique of removing the spectral shaping in the waveform due to the formants has been called spectral flattening [1]. Sondhi has proposed two methods for performing this spectral flattening—a filter bank method and the technique of center clipping. For the filter bank method the signal is filtered by a bank of bandpass filters which span the bandwidth of the signal. The signal at the output of the filter is normalized to unit amplitude (spectrally flattened) by dividing it by its short-time energy. The total spectrally flattened signal is obtained by adding the individually flattened channels with the appropriate delays. Although this method works very well in many cases, there are several drawbacks to practical implementations of this method. First, the method requires a considerable amount of hardware for filtering and equalization. Second, there are cases where the flattening produces very bad results. These cases occur when no pitch harmonic is contained within an individual bandpass filter. In this case the filter output is low level; therefore the equalized output is essentially high-level noise which tends to obscure rather than aid the pitch detection process.

Manuscript received August 7, 1975; revised October 27, 1975.

J. J. Dubnowski and L. R. Rabiner are with Bell Laboratories, Murray Hill, NJ 07974.

R. W. Schafer is with the Department of Electrical Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

[1] It should be noted that some special purpose extremely fast processors such as the fast digital processor (FDP) and the digital voice terminal (DVT) at the M.I.T. Lincoln Laboratories, and the programmable signal processor (PSP) at Sylvania, have been built which are capable of running most pitch detection algorithms in real time. Generally these processors are either expensive, or are not commercially available.

An alternative way of spectrally flattening a signal is the process of center clipping in which signal values below the clipping level are set to zero, whereas those above the clipping level are offset by the clipping level. Fig. 1 shows the input-output characteristic of the center clipper used by Sondhi [1] and an illustration of how the center-clipping method effectively acts as a spectral flattener. It can be seen from Fig. 1 that if the clipping level is appropriately set, most of the waveform structure, due to the formants, can be entirely eliminated. Thus a center clipper effectively yields a spectrally flattened signal whose periodicity is much easier to measure than the comparable nonflattened signal.

The method used to estimate the pitch period in the hardware implementation is based on a modification of this center-clipping method. Fig. 2 shows a block diagram of the pitch-detection algorithm. The analog input speech signal is first low-pass filtered to a bandwidth of about 900 Hz, and then converted to digital form by a 12-bit analog-to-digital (A/D) converter. The signal at the output of the converter $s(n)$ is then sectioned into overlapping 30 ms sections for processing. (Since the pitch period computation is performed 100 times per second, i.e., every 10 ms, adjacent sections overlap by 20 ms.)

The first stage of processing for each section is the computation of the clipping level for that section. Because of the wide dynamic range of speech, the clipping level must be carefully chosen so as to prevent loss of waveform information when the waveform is either rising in amplitude or falling in amplitude during the section. Such cases occur when voicing is just beginning or ending, as well as during voicing transitions, e.g., from a vowel to a voiced fricative, or a nasal. The way in which the clipping level C_L is chosen is as follows. The 30 ms section of speech is divided into three consecutive 10 ms sections. For the first and third 10 ms sections the algorithm finds the maximum absolute peak levels. The clipping level is then set as a fixed percentage of the *smaller* of these two maximum absolute peak levels. The percentage that is actually used is a parameter of the pitch-detector hardware; however, extensive computer simulations have shown that a value of around 80 percent is appropriate for most cases. It should be noted that in Sondhi's original work, the percentage chosen for setting the clipping level was about 30 percent [1]. This was due to Sondhi's method of setting the clipping level based on the peak absolute value over the whole 30 ms section. To avoid losing low-level voiced information a low-clipping level was required. This more sophisticated method of choosing the clipping level has eliminated this problem.

Following the determination of the clipping level, the speech section is then both center clipped, and infinite peak clipped, resulting in a signal which assumes one of three possible values—+1 if the sample exceeds the positive clipping level, -1 if the sample falls below the negative clipping level, and 0 otherwise. Fig. 3 shows a plot of the input-output characteristic for the combination center clipper, infinite peak clipper. The use of infinite peak clipping following the center clipper greatly reduces the computational complexity of the autocorrelation measurement which follows the clipping. This is because no multiplications or additions are required in the computation of the autocorrelation function of the clipped signal.

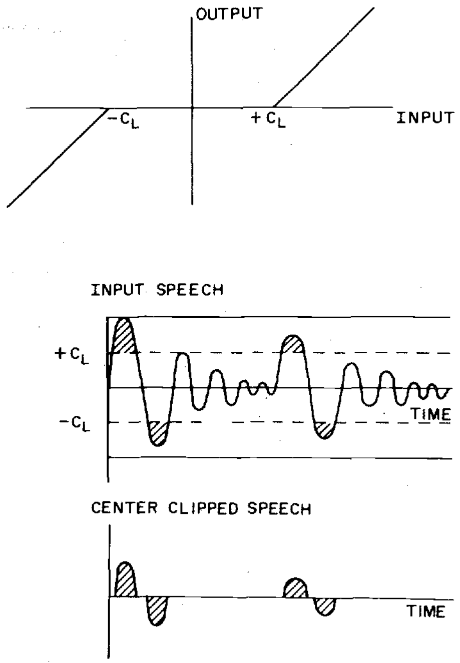

Fig. 1. Input-output characteristic and typical operation of a center clipper (after Sondhi).

The next stage in the processing is the autocorrelation computation. The autocorrelation for the clipped 30 ms section is defined as

$$R_x(m) = \sum_{n=0}^{299-m} x(n)x(n+m) \qquad m = M_i, M_i+1, \cdots, M_f \tag{1}$$

where M_i is the initial lag and M_f is the final lag for which the autocorrelation is computed. (These parameters are variables in the hardware and can be set by the user. Typical values of M_i and M_f are 25 and 200, respectively, corresponding to a pitch range of 400 Hz down to 50 Hz.) Additionally, $R_x(0)$ is computed for appropriate normalization of the autocorrelation function. Since the individual terms in (1) are of the form $x(n)x(n+m)$, and since $x(n)$ can only assume the values, +1, 0, or -1, then each combination of (1) can assume the values

$$\begin{aligned} x(n)x(n+m) &= 0 \text{ if } x(n) = 0, \text{or if } x(n+m) = 0 \\ &= 1 \text{ if } x(n) = x(n+m) = \pm 1 \\ &= -1 \text{ if } x(n) = -x(n+m) = \pm 1. \end{aligned} \tag{2}$$

Thus, a simple combinatorial logic circuit is all that is required to compute the individual terms in the autocorrelation function, and an up-down counter is all that is required to accumulate the actual autocorrelation value.

Fig. 4 shows an example of the processing for a typical 30 ms section of speech. At the top of this figure is shown the low-pass filtered waveform, and the clipping thresholds $\pm C_L$, for this example. At the middle of this figure is shown the clipped speech. Finally, at the bottom of Fig. 4 is shown the autocorrelation function of the clipped speech. The range in which the pitch period generally lies is shown by the dotted lines at $m = 20$ and $m = 200$. For this example the pitch

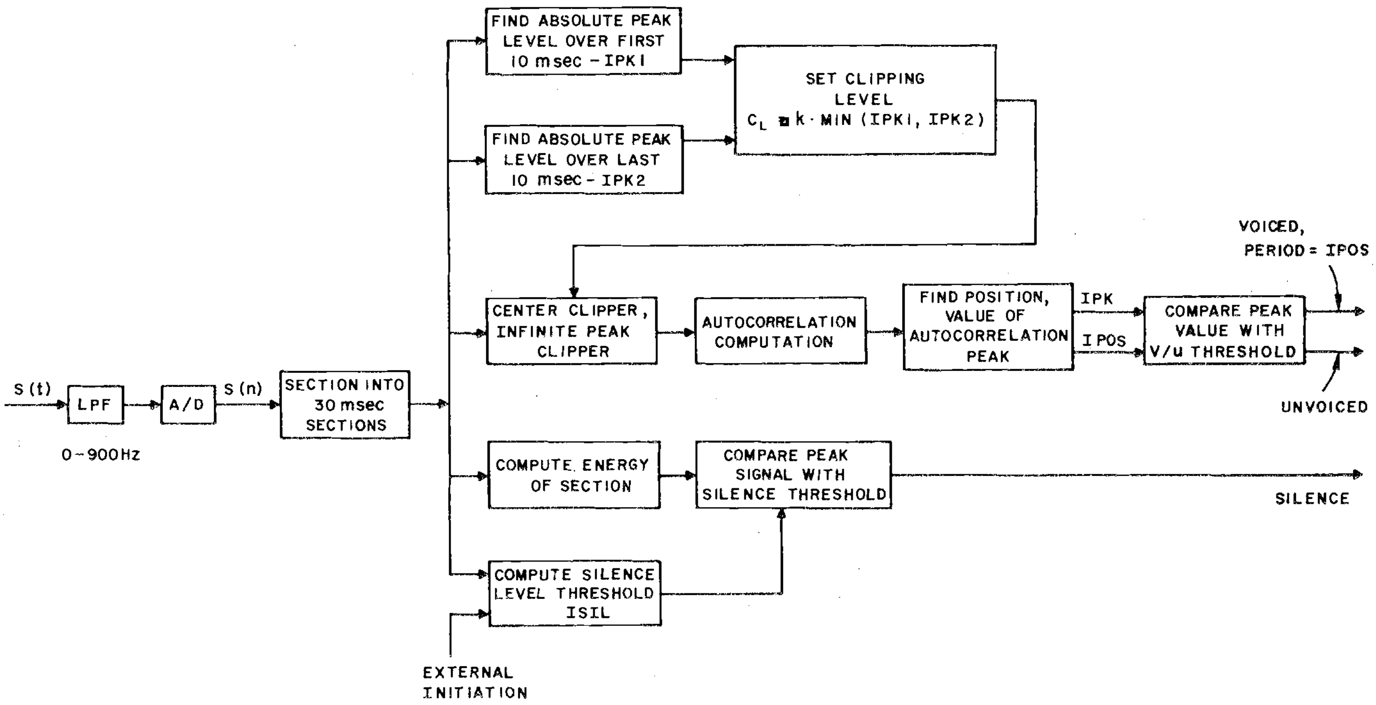
FIND ABSOLUTE PEAK LEVEL OVER FIRST 10 msec - IPK1
FIND ABSOLUTE PEAK LEVEL OVER LAST 10 msec - IPK2
SET CLIPPING LEVEL C_L = k·MIN (IPK1, IPK2)
VOICED, PERIOD = IPOS
CENTER CLIPPER, INFINITE PEAK CLIPPER
AUTOCORRELATION COMPUTATION
FIND POSITION, VALUE OF AUTOCORRELATION PEAK
IPK
IPOS
COMPARE PEAK VALUE WITH V/u THRESHOLD
S(t)
LPF
A/D
S(n)
SECTION INTO 30 msec SECTIONS
0-900Hz
UNVOICED
COMPUTE ENERGY OF SECTION
COMPARE PEAK SIGNAL WITH SILENCE THRESHOLD
SILENCE
COMPUTE SILENCE LEVEL THRESHOLD ISIL
EXTERNAL INITIATION

Fig. 2. Block diagram of the overall pitch detector built in digital hardware.

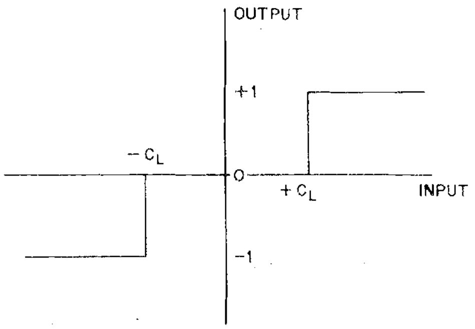

Fig. 3. Input-output characteristic of the combination center clipper, infinite peak clipper.

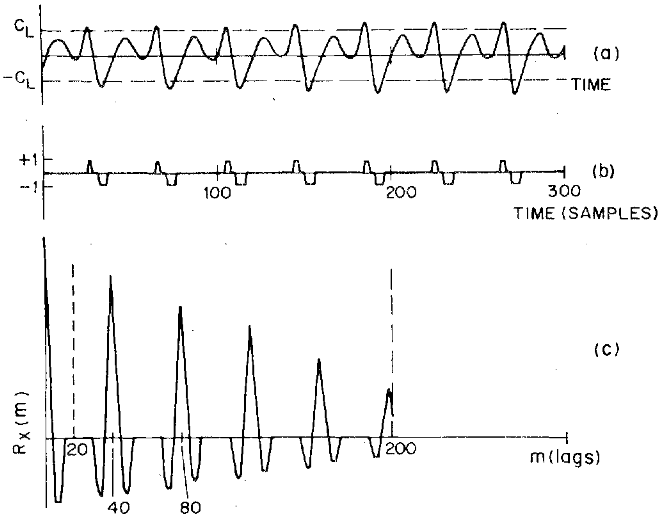

Fig. 4. Example of clipped speech and its autocorrelation function.

period of the section is about 40 samples, or 4 ms, or a pitch frequency of 250 Hz.

It should be noted that in the computation of the autocorrelation function (1) it is assumed that samples outside the current 30 ms section are zero. This effectively weights the autocorrelation function by a linear taper which starts at 1 for $m = 0$ and goes to 0 at $m = 300$. This effect is clearly seen in Fig. 4 where the peaks of the autocorrelation function linearly taper to zero. The use of a linear taper on the autocorrelation function effectively enhances the peak at the pitch period with respect to peaks at multiples of the pitch period, thereby reducing the possibility of doubling or tripling the pitch-period estimate because of higher correlations at these lags than at the lag of the actual pitch period.

In addition to pitch, the hardware also makes a computation that represents the energy for each section. The actual computation used (which will be denoted as the energy of the section) is

$$E = \sum_{n=0}^{99} |s(n)|, \tag{3}$$

i.e., the energy is computed as the sum of the absolute values of the speech samples over a 10 ms interval.

Additionally, based on peak signal levels, a silence level threshold can be chosen. This threshold serves to distinguish low-level background noise from speech. The silence level threshold is obtained by measuring the peak signal level for 50 ms of background silence. This silence level threshold is stored in a register and is then compared against the peak signal level in a given 30 ms section. If the peak level falls below the silence level threshold, the 30 ms section is classified as background silence, and no pitch-period computation is performed. This silence level threshold can be reset either manually or under program control whenever desired. Thus the pitch detector has the capability of adapting to a variety of background environments.

If the section of speech is not classified as background silence (i.e., its peak level exceeds the threshold level) the autocorrelation computation is performed and the autocorrelation function is searched for its maximum value in the interval $m = M_i$ to $m = M_f$. Both the location and the value of this maximum are saved. If the value of this maximum (relative to the autocorrelation at $m = 0$) exceeds a voiced-unvoiced threshold (on the order of 0.30) the section is classified as voiced and the pitch period is the position of the maximum peak. If the peak value falls below the threshold, the interval is classified as unvoiced.

III. Digital Hardware Implementation of the Pitch Detector

As discussed earlier, the pitch detector of Section II has been implemented in digital hardware. The hardware is divided into two distinct processors. The first processor performs the adaptive clipping operation, and makes the energy computation, whereas the second processor computes the autocorrelation function and makes the final pitch period estimate and voiced-unvoiced decision. These two processors operate in parallel—thus while the current pitch-period estimate is being computed, the next segment of speech is being loaded and processed in the first processor. This pipelined structure allows real-time operation at a 10 kHz sampling rate with pitch and energy computations made every 10 ms (100 times a second).

Fig. 5 shows the hardware organization of the pitch detector. The analog speech waveform is low-pass filtered with a 900 Hz cutoff filter and converted to digital form by a 12 bit A/D converter. The clipping level and energy computation are made simultaneously as each new 100-sample speech segment is clocked into a 300 word buffer every 10 ms. The 300 word data buffer is processed by the clipper and the output is shifted into the autocorrelation processor.

The autocorrelation computation is performed over the preset lag interval. The block labeled pitch logic finds the largest autocorrelation peak, and stores both the amplitude and location of the peak. The amplitude of the maximum peak is compared with an autocorrelation threshold to make the final voiced-unvoiced decision.

Fig. 6 shows a detailed block diagram of the first processor. Speech data are loaded into three 100 word × 8-bit MOS shift registers. The use of three shift registers enables the processor to make the peak signal computation as the data are received. The two comparator-latch combinations monitor the signal level from the first and last shift registers for the clipping-level computation.

The minimum level control is an externally initiated function which scans a 512-sample sequence to determine the maximum signal level. Since this function can be initiated at any time, the hardware is esentially capable of training itself or adapting to any background environment.

Once the clipping level is determined, the center clipper performs the entire clipping function of the pitch detector directly on the 300 samples stored in the shift registers. The output of the clipper is stored directly in a 512 word × 2 bit bipolar memory of the second processor as shown in Fig. 7. Two counters and a memory address selector are used to access the data for the autocorrelation computation. Counter B and counter A provide the memory addressing associated with $x(n)$ and $x(m + n)$ in computing the autocorrelation. Counter B is initially set to zero prior to each computation and is incremented after each data access. Counter A is loaded from the starting address counter prior to each computation and is also incremented following each data access. The completion of the computation for an autocorrelation element is indicated by the comparison of counter A's output with the data range stored in the range latch. This generates a "correlation element complete" signal which increments the starting address counter. While the memory is being accessed, data output pass through the combinatorial logic and the result appropriately clocks the up-down counter. This accumulated count is compared against the max count from any past autocorrelation element computation when the correlation element complete signal occurs. If the new computation is higher, it is stored in the max peak latch and the address from the starting address counter is stored in the pitch latch. In this way when the entire autocorrelation has been computed, as indicated by comparing the incremented starting-address counter with the end of correlation lag, the value remaining in the max peak latch represents the largest autocorrelation peak and the respective value stored in the pitch latch corresponds to the pitch period. The additional comparison made with the voiced-unvoiced threshold clears the pitch latch if the maximum autocorrelation peak does not exceed the threshold, thereby indicating unvoiced speech.

The hardware described above used about 150 IC chips. Aside from the MOS shift registers, and the fast bipolar memory, all other circuits are standard speed T^2L logic.

IV. Discussion and Summary

The hardware pitch detector described in Section III has been built and interfaced to the NOVA computer facility of the Acoustics Research Department. An extensive performance evaluation was made of the capabilities of this and several other pitch-detection algorithms using software simulations [9]. To test the performance of these algorithms, a speech data base, consisting of eight utterances spoken by 3 males, 3 females, and 1 child was constructed. Simultaneous telephone and close talking microphone recordings were made of each of the utterances. For each of the utterances in the data base a "standard" pitch contour was semiautomatically measured using a highly sophisticated interactive pitch detection program [10]. The "standard" pitch contour was then compared with the pitch contour that was obtained from each of the programmed pitch detectors. A set of measurements

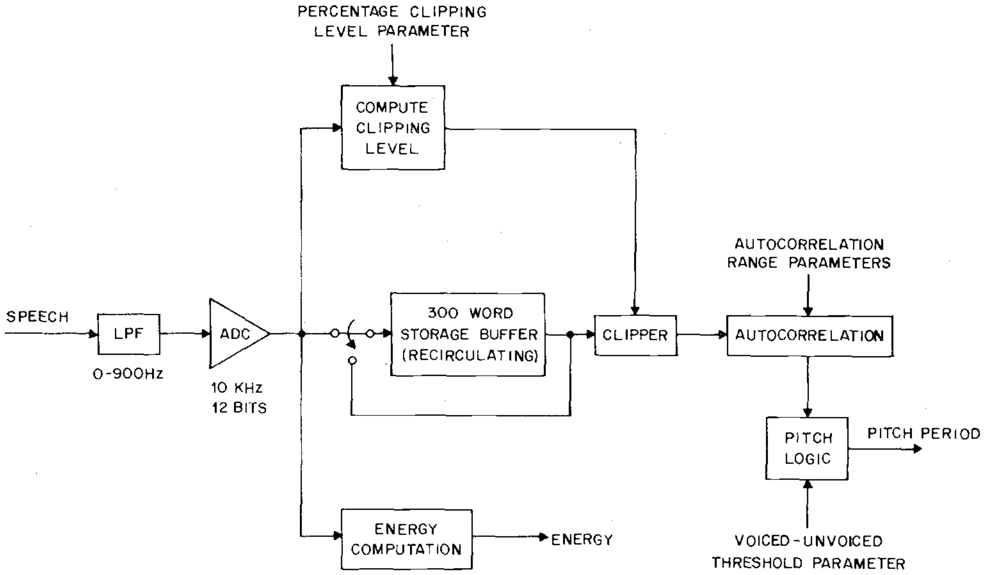
PERCENTAGE CLIPPING LEVEL PARAMETER
COMPUTE CLIPPING LEVEL
AUTOCORRELATION RANGE PARAMETERS
SPEECH
LPF
0-900Hz
ADC
10 KHz 12 BITS
300 WORD STORAGE BUFFER (RECIRCULATING)
CLIPPER
AUTOCORRELATION
PITCH LOGIC
PITCH PERIOD
ENERGY COMPUTATION
ENERGY
VOICED-UNVOICED THRESHOLD PARAMETER

Fig. 5. Hardware structure of the pitch detector.

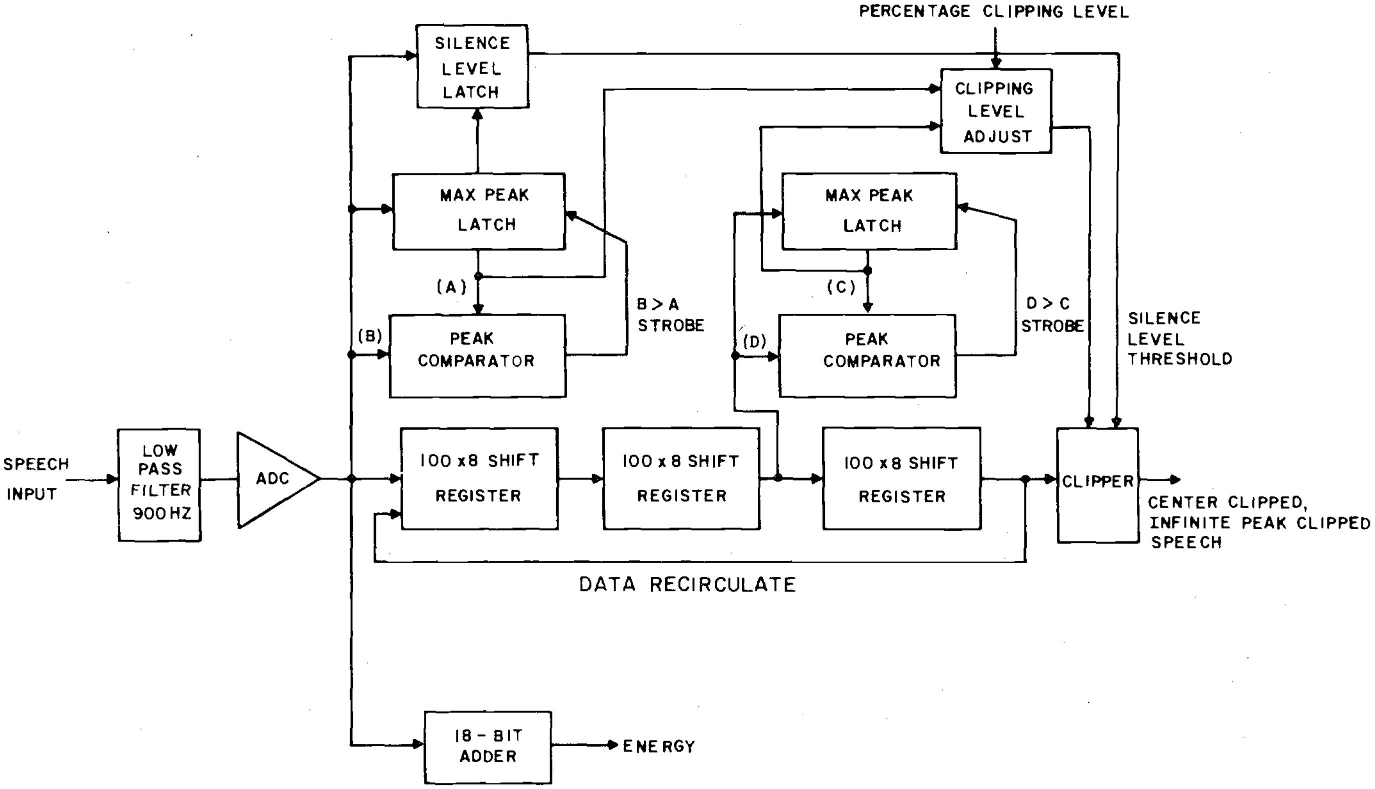

Fig. 6. Detailed hardware description of the clipper and gain computation.

were made on the pitch contours to quantify the various types of errors which occur in the pitch detection process. Included among the error measurements were the average and standard deviation of the error in pitch period during voiced regions, the number of gross errors in the pitch period, and the average and standard deviation of the error in locating the onset and offset of voicing. By pooling the various error measurements, the individual pitch detectors could be rank-ordered as a measure of this relative performance.

Since the details of the performance evaluation are available in [9], we will only summarize the results obtained for the pitch-detection algorithm described in this paper.

The errors made in measuring the pitch period during voiced regions were divided into two categories. The first category included all cases where the magnitude of the difference between the standard value of the pitch period, and that measured by the pitch detector, was less than 10 samples. The second category included all cases where the magnitude of the difference in pitch periods was 10 samples or larger. These errors were referred to as gross pitch-period errors. For the first category of errors, namely the fine errors in pitch period, the average pitch-period error was in the range of -0.3 to 0.1 samples across the different speakers, and across the different recording conditions. The standard deviation of the

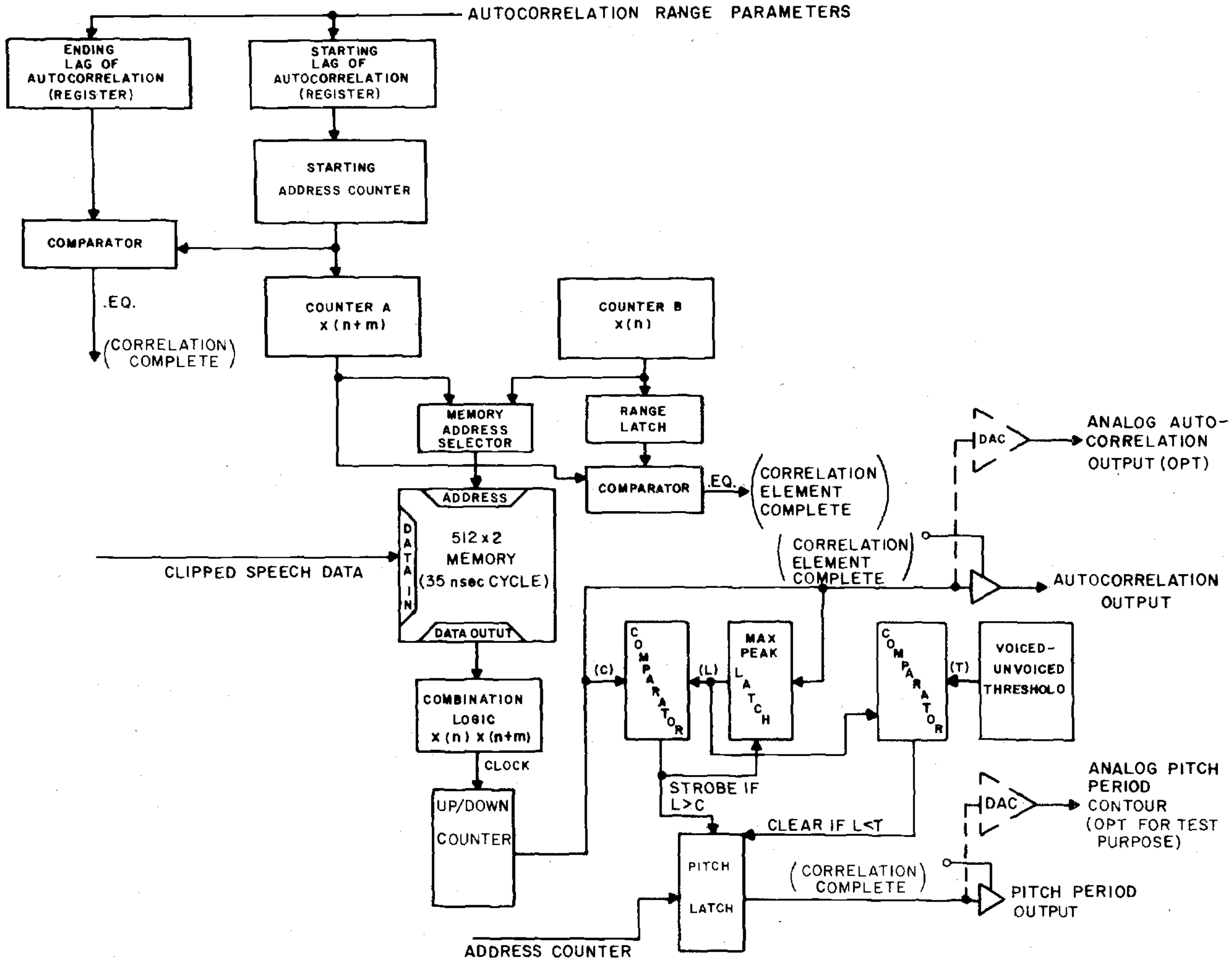

Fig. 7. Detailed hardware description of the autocorrelation and pitch-period logic.

pitch-period error varied from 0.4 to 1.0 samples across speakers and conditions. Both these error scores are essentially within the measurement accuracy of the pitch period. Thus the conclusion can be drawn that, for the cases where gross errors are excluded from the measurement, the autocorrelation pitch detector can determine the correct pitch period quite accurately.

In the case of gross errors, the autocorrelation pitch detector runs into difficulty primarily for low-pitch speakers where the pitch period is quite long. The errors that occur here are due to the fixed frame size of 300 samples used in the analysis. When the pitch period exceeds 150 samples, the analysis frame size is not large enough to hold two full periods of speech; thereby increasing the chance of a gross error in locating the correct pitch period. In the study of [9], the two low-pitch speakers used in the study both showed a large number of gross errors in the pitch period (for both the telephone and microphone recordings). All other speakers had only occassional gross errors in the pitch period (i.e., one gross error/s on average). A nonlinear smoothing algorithm [11] was used in the study of [9] to isolate and correct these gross errors, as well as isolated errors in the voiced-unvoiced decision. After processing by the nonlinear smoothing algorithm, essentially all gross pitch-period errors were corrected except for the case of the low-pitch male speaker where some of the errors occurred in clusters, and therefore were essentially not correctable by a median-type smoother.

The second category in the performance evaluation of [9] was the accuracy in voiced-unvoiced boundary location. For the autocorrelation pitch detector, the average error in locating the voiced-unvoiced boundary was on the order of 5 ms (half the average frame rate of 100 frames/s or 10 ms/frame), and the standard deviation of the error was on the order of 10 ms across all speakers and recording conditions. Thus the error in locating the voiced-unvoiced boundaries was on the order of the precision of the measurements.

The way in which these results are interpreted depends very strongly on the intended application for the pitch detector. The hardware pitch detector described in this paper will be used in a speaker verification system [7], and will be tested in a linear predictive coefficient (LPC) vocoder simulation.

In summary, a fairly versatile real-time pitch detector has been built in digital hardware. The pitch-detection algorithm is based on a combination of center clipping and infinite peak clipping, and uses a simplified autocorrelation analysis to estimate the pitch period. Additional features incorporated in the hardware include an energy computation, a simple threshold comparison to eliminate low-level signals, and a final voiced-unvoiced decision based on the peak value of the correlation function.

References

[1] M. M. Sondhi, "New methods of pitch extraction," *IEEE Trans. Audio Electroacoust. (Special Issue on Speech Communication and Processing–Part II)*, vol. AU-16, pp. 262–266, June 1968.

[2] A. M. Noll, "Cepstrum pitch determination," *J. Acoust. Soc. Amer.*, vol. 41, pp. 293–309, Feb. 1967.

[3] J. D. Markel, "The SIFT algorithm for fundamental frequency estimation," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 367–377, Dec. 1972.

[4] B. Gold and L. R. Rabiner, "Parallel processing techniques for estimating pitch periods of speech in the time domain," *J. Acoust. Soc. Amer.*, vol. 46, pp. 442–448, Aug. 1969.

[5] N. J. Miller, "Pitch detection by data reduction," *IEEE Trans. Acoust., Speech, Signal Processing (Special Issue on IEEE Symposium on Speech Recognition)*, vol. ASSP-23, pp. 72–79, Feb. 1975.

[6] M. J. Ross *et al.*, "Average magnitude difference function pitch extractor," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 353–362, Oct. 1974.

[7] A. E. Rosenberg and M. R. Sambur, "New techniques for automatic speaker verification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 169–176, Apr. 1975.

[8] H. Levitt, "Speech processing aids for the deaf: An overview," *IEEE Trans. Audio Electroacoust. (Special Issue on 1972 Conference on Speech Communication and Processing)*, vol. AU-21, pp. 269–273, June 1973.

[9] M J. Cheng, "A comparative performance study of several pitch detection algorithms," M.S. thesis, Dep. Elec. Eng., Massachusetts Inst. Technol., Cambridge, June 1975.

[10] C. A. McGonegal, L. R. Rabiner, and A. E. Rosenberg, "A semiautomatic pitch detector (SAPD)," *IEEE Trans. Accoust., Speech, Signal Processing*, vol. ASSP-23, pp. 570–574, Dec. 1975.

[11] L. R. Rabiner, M. R. Sambur, and C. E. Schmidt, "Applications of a nonlinear smoothing algorithm to speech processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 552–557, Dec. 1975.

A Comparison of Three Methods of Extracting Resonance Information from Predictor-Coefficient Coded Speech

RANDALL L. CHRISTENSEN, WILLIAM J. STRONG, MEMBER, IEEE, AND E. PAUL PALMER

Abstract—**Three methods of extracting resonance information from predictor-coefficient coded speech are compared. The methods are finding roots of the polynomial in the denominator of the transfer function using Newton iteration, picking peaks in the spectrum of the transfer function, and picking peaks in the negative of the second derivative of the spectrum. A relationship was found between the bandwidth of a resonance and the magnitude of the second derivative peak. Data, accumulated from a total of about two minutes of running speech from both female and male talkers, are presented illustrating the relative effectiveness of each method in locating resonances. The second-derivative method was shown to locate about 98 percent of the significant resonances while the simple peak-picking method located about 85 percent.**

Introduction

MANY speech processing applications in use today require a knowledge of speech formant information. Formants are significant parameters for characterizing various speech sounds and as such are used in programs for machine recognition of speech, in machine voice-response systems, and in controlling terminal-analog synthesizers used in speech synthesis by rule. Formant frequency information is needed to realize a formant vocoder, although other more easily obtained parameters may be preferable if one is interested only in the vocoding problem. Formant frequencies are "natural" parameters due to their relationship to the underlying vocal tract configuration, and for this reason, they have an intuitive appeal for researchers in speech synthesis and recognition. There is also evidence that formant information is an efficient way to code speech sounds [11].

Both iterative and noniterative approaches have been used for estimating formant frequencies. An iterative approach is analysis by synthesis in which adjustments are made on the parameters of a speech synthesis model until some desired degree of matching is obtained between the actual speech spectrum and the spectrum resulting from the model. Analysis by synthesis permits great flexibility in making spectral matches but requires extensive processing in its iterations. Noniterative approaches are appealing because of their comparative computational efficiencies. These approaches often depend on detecting spectral peaks and identifying them as possible formants. Cepstral methods have been used to obtain smoothed spectra which are peak picked via human intervention [13] or by computer [12].

The recent application of linear-prediction methods to speech analysis has made formant estimation more tractable. The predictor-coefficient method matches the spectrum of a variable, multiresonance digital filter, and the spectral envelope of a speech segment so that the mean-squared error is

Manuscript received November 26, 1974; revised May 1, 1975 and September 9, 1975.

R. L. Christensen was with the Department of Physics and Astronomy, Brigham Young University, Provo, UT 84602. He is now with the Naval Weapons Center, China Lake, CA 93555.

W. J. Strong and E. P. Palmer are with the Department of Physics and Astronomy, Brigham Young University, Provo, UT 84602.