

## Real-Time Speech Analysis-Synthesis System Using Digital Signal Processors

Yasuo ARIKI and Toshiyuki SAKAI

### SUMMARY

In this study, a system for real-time speech analysis-synthesis is designed and implemented using Digital Signal Processors (DSP). The hardware architecture is based on a hierarchy of A/D (D/A) converter, DSPs, micro-computer, super-minicomputer. The DSP-software is developed for FFT, IFFT, bandpass-filter and PARCOR analysis-synthesis. Le Roux algorithm and optimal scaling technique are employed in PARCOR analysis and proved to be more precise in computation. Moreover, several experiments are carried out to improve the quality of PARCOR synthesis speech. It is shown that a semi-square root of the power, proposed in this paper, is as effective as an accurate square root of the power for the amplitude information. Finally, a speech play-back method with variable speed control without changing tone is designed and implemented based on PARCOR synthesis.

### 1. INTRODUCTION

Digital signal processing of speech has been performed on a mini-computer or a super-minicomputer due to its requirement of high speed multiplication and accumulation. However, recently, specialized LSI chips with low cost and high performance for digital signal processing are produced.

Digital Signal Processor (DSP) is a LSI chip with generality such as program ROM and data RAM in itself. Using the DSPs, specialized vocoder has been implemented<sup>1)</sup>. Our purpose is to develop general purpose system for real-time speech analysis and synthesis using the DSPs instead of simple vocoder.

At present, Fast Fourier Transform (FFT), Inverse FFT (IFFT), bandpass filter bank, PARCOR analysis, pitch extraction and PARCOR synthesis software are developed for real-time processing on the DSPs.

The hardware architecture of the system is based on a hierarchy of A/D (D/A) converter, DSPs, micro-computer and super-minicomputer so that the processing is distributed. The real-time processing software is loaded on the DSP, micro-computer and super-minicomputer according to the application.

---

Yasuo ARIKI (有木康雄): Assistant Professor, Department of Information Science, Kyoto University.

Toshiyuki SAKAI (坂井利之): Professor, Department of Information Science, Kyoto University.

## 2. HARDWARE OF THE REAL-TIME ANALYSIS-SYNTHESIS SYSTEM

### 2.1 Digital Signal Processor $\mu$ PD7720

Digital signal processor (DSP)  $\mu$ PD7720<sup>2)</sup> produced by NEC has a lot of data ROM to store coefficients in computation so that the DSP is suitable for our system to store FFT twiddle factor and filter coefficients of bandpass filter. The DSP has the following features about its hardware structure.

- (1) Data value  $x$  is limited to  $-1 \leq x \leq 1$  due to fixed point representation so that virtual fixed point is required to represent the data greater than 1.
- (2) Data RAM to store input data consists of 128 word memory and its address is referred to by data pointer with 7 bits. By using this data pointer effectively to store variables, processing simplicity and speed up can be obtained.
- (3) Program is controlled by flags and their conditional branching instructions so that comparison or repetition is implemented by using them.
- (4) Pipeline processing is employed in the DSP. Efficient processing is achieved by programing suitable for its architecture.

### 2.2 Hardware Structure of the System

Fig. 1 shows a block diagram of the hardware structure of the system. The left part corresponds to analysis block and right part to synthesis block. Host processor is a personal computer FM-11 (Fujitsu) with 8088 CPU. It is connected to a super-minicomputer MS-190 (NEC) via local area network SIMPLE net developed in our laboratory. Hardware architecture, therefore, is a hierarchy of A/D converter, DSP, micro-computer and super-minicomputer. Accuracy of A/D (D/A) conversion is 16 bit. Sampling rate can be variable up to 44 KHz. In the figure, frame buffer memory is inserted between A/D(D/A) converter and DSP. This enables DSP to deal with the data framewise by DMA as well as pointwise. Frame size is variable up to 2048 bytes. The following three functions are achieved by this architecture.

- (1) Overlapping of the consecutive frames is achieved in real time by changing the start point of each frame in the frame buffer memory.
- (2) Frame data stored on frame buffer memory can be read twice or it is read in division when the frame size is greater than data RAM size.
- (3) Sampled data are obtained free from sample period due to buffering them so that analysis with the processing time longer than sample period can be achieved.

Each DSP has three bus lines. They are A/D converter bus, frame buffer bus and host processor bus. These buses are selected by each DSP. Data transfer from host processor to DSP and its inverse are achieved either program control or DMA control. In DMA transfer between host processor and DSP as well as between frame buffer memory and DSP, interrupt signal is used to start DMA. DSP starts data transfer according to the interrupt factor. Start and stop of A/D

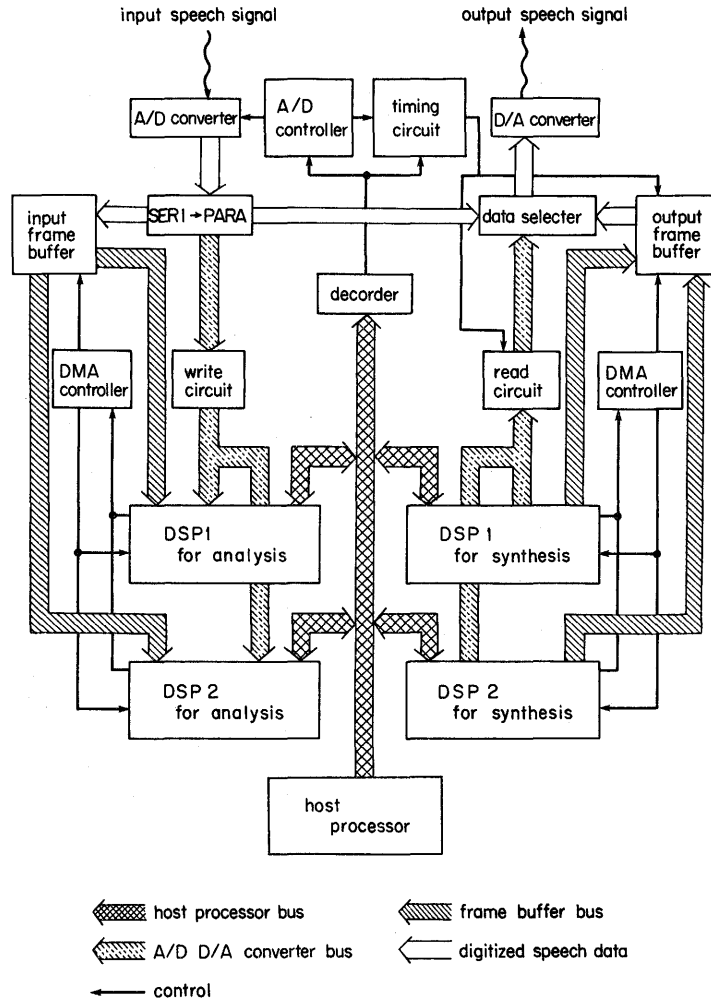


Fig. 1. Hardware block of the system.

(D/A) converter, sample period and the number of acquisition data can be specified interactively.

### 3. SOFTWARE OF THE REAL-TIME SPEECH ANALYSIS

#### 3.1 Fast Fourire Transform

Decimation-In Frequency algorithm (DIF) is employed. Real 128 points FFT is feasible by one DSP in our hardware architecture. This limitation of FFT points is derived from the limitation of data RAM (with 128 words) of the DSP.

Output from the first stage of butterfly computation is complex 128 points (256 words). These data exceed the data RAM capacity so that input speech data (real 128 points) are read in two times and complex 64 points are produced re-

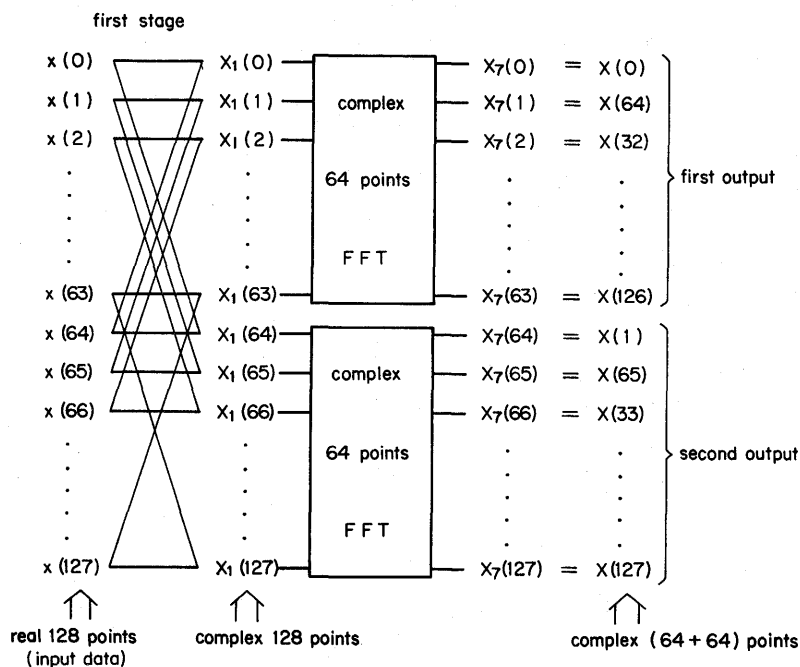


Fig. 2. 128 points FFT by DSP.

spectively at the first stage of butterfly computation as shown in Fig. 2. In this FFT, the first output of complex 64 points includes the even frequency and the second includes the odd frequency.

### 3.2 Bandpass Filter Bank

Specification of bandpass filter bank is as follows:

- (1) Magnitude response at central frequency of each filter is 0dB.
- (2) Neighboring filters cross at -3dB of magnitude response.
- (3) The fourth order IIR digital filter of Butterworth type is employed due to -10 dB of magnitude at the central frequency of neighboring filters.

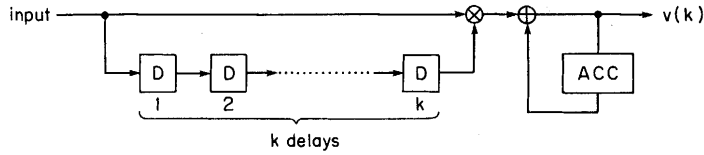
Our approach to this specification is to convert the second-order analog low-pass filter of Butterworth type to the fourth order analog bandpass filter. Then impulse invariant method is applied to the filter to obtain the digital bandpass filter.

### 3.3 PARCOR Analysis

Typical algorithms of PRACOR analysis are followings:

- (1) Durbin-Levinson-Itakura method
- (2) Le Roux method<sup>3)</sup>

Absolute value of PARCOR coefficients is less than 1. Method (1) sometimes has the absolute value greater than 1 during the computation. On the other hand, method (2) makes it be less than 1 so that the algorithm is suitable for the DSP with fixed point arithmetic. In PARCOR analysis, autocorrelations from the zero-th to the 12-th are computed at first. To compute them in high

Fig. 3. Filter to compute the  $k$ -th autocorrelation.

speed on small capacity of data RAM, filter shown in Fig. 3 is incorporated. This filter enables to compute all the autocorrelations just after the last sample is acquired. Le Roux method (2) is applied to these autocorrelations to produce the PARCOR analysis up to the 12-th. The hardware of input frame buffer is used to enable this algorithm by the DSP.

The DSP has no division arithmetic so that division software should be developed on DSP. PARCOR coefficients computed by the DSP with fixed point are very accurate compared with floating points arithmetic. The difference between them is less than 0.005. The first and second coefficients  $k_1$ ,  $k_2$  which are important for recognition and synthesis are accurate within the error of 0.02% compared with floating point arithmetic.

#### 3.4 Pitch Period Extraction

Autocorrelation or cepstrum method is impossible to extract pitch period in real time. We extract pitch period in real time by employing and modifying pitch period estimation algorithm based on parallel processing on time domain.

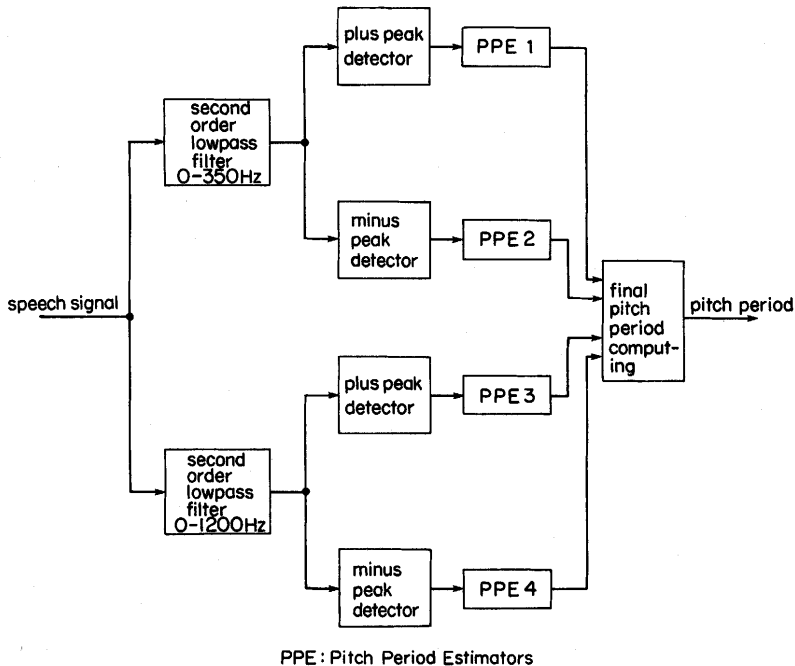


Fig. 4. Block diagram for extracting pitch period.

The algorithm was proposed by B. Gold<sup>4)</sup> and improved by L.R. Rabiner<sup>5)</sup> and is shown in Fig. 4. We describe the detail of each block in the figure.

(a) Lowpass filter

Two lowpass filters are used. One is for pitch period by setting the cut off frequency low. The other is for harmonics of pitch period by setting the cut off frequency high.

(b) Peak detector

Two kinds of peak detectors are applied to each signal from two filters. One is to detect the plus peaks. The other is to detect the minus peaks. When the peaks are detected, one impulse with the same magnitude as the peak is placed on that time. When the peaks are not detected, no impulses are placed. As the result of the peak detectors, four kinds of impulse trains are obtained.

(c) Pitch period estimator

Each impulse train is fed into corresponding pitch period estimator. Fig. 5 shows the process of pitch period estimator. When the impulse with high magnitude is detected, the impulse is passed to output. Blanking interval is set to inhibit any impulse to be detected. At the end of blanking interval, exponential window to detect impulse is set. When impulses are detected which exceed the exponential window level, the impulses are passed to output. Then the same process is repeated to detect the next impulse. The output is a train of semi-pitch period and is obtained by input impulse train smoothing. These four outputs from pitch period estimators are fed into the final pitch period estimator and pitch period is finally obtained on the basis of a majority decision.

(d) Unvoiced speech processing

The pitch period estimation described above is not applied well at the unvoiced speech or silence part. If the pitch period is not detected within 20 ms after last impulse, the time part is judged as unvoiced or silence part.

(e) Implementation

On the data ROM of the DSP, exponential data for the window are stored as the table. The size of the exponential table is 180 words in a case of 10 KHz sampling because the exponential window lasts 18 ms (20 ms minus 2 ms). Furthermore low pass filter coefficients are stored on the data ROM.

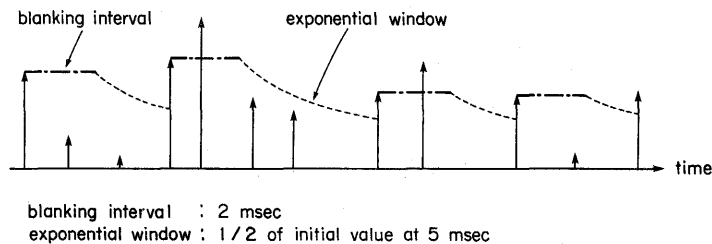


Fig. 5. Exponential window for peak detector.

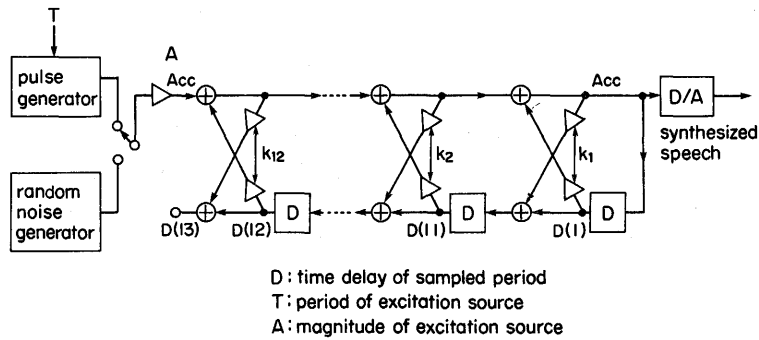


Fig. 6. Configuration of PARCOR synthesis.

One DSP is used to extract pitch period by using input frame buffer. The input frame buffer is effective because computation of filtering, peak detector and pitch period estimation sometimes exceed the sampling interval.

#### 4. REAL TIME SPEECH SYNTHESIS SOFTWARE

##### 4.1 Inverse FFT

Complex 64 points IFFT can be implemented on one DSP. This limitation is derived from capacity of data RAM. DIF method described in section 3.1 is employed.

##### 4.2 PARCOR synthesis

Fig. 6 shows the block diagram of PARCOR synthesis with 12 coefficients. Required parameters are PARCOR coefficients, pitch period and its magnitude. In the figure, pulse generator corresponds to voiced excitation and random noise generator corresponds to unvoiced excitation. Selection of excitations depends on the information whether pitch period is extracted or not. In a case of voiced excitation, pulse train with same period as extracted ones is generated. Random numbers are stored on the data ROM for unvoiced excitation.

Discrimination between speech and silence is sensitive to noise level and speech level. To discriminate them definitely introduces the error so that magnitude of excitation source is set to the same magnitude as the input speech.

#### 5. REAL TIME SPEECH INPUT AND DISPLAY SOFTWARE

Following processes are required to store the speech data from A/D converter on the system memory of the micro-computer and display the speech waveform on the graphic display at the same time.

- (1) Sampling and digitization by A/D converter.
- (2) Speech data transfer from A/D converter to the system memory.
- (3) Data scaling and co-ordinate computation to display the data.
- (4) Displaying the waveform on the graphic display.

If each sampled data is displayed as a component of the waveform, above four steps must be finished within every sampling period. One instruction, however, consumes several micro-second so that it is difficult to finish the four steps within sampling period (for example  $100 \mu\text{s}$ ). In order to store and display the speech data in real time, consecutive sampled points are blocked as frame and one vertical line to one frame is drawn on the graphic display as follows:

- (1) Maximum value  $M(n)$  and minimum value  $m(n)$  of the sampled point are searched within  $n$ -th frame. At present, frame length is 128 sampled points.
- (2) The vertical line with maximum and minimum value as the end point is drawn for each frame.
- (3) End points of vertical lines are modified to guarantee the connectivity between consecutive lines as follows:

$$YH(n) = \max(M(n), m(n-1))$$

$$YL(n) = \min(m(n), M(n-1))$$

$YH(n)$ : maximum value of the end point of the line.

$YL(n)$ : minimum value of the end point of the line.

In drawing the waveform, display window should be scrolled as a function of time to display the waveform endlessly. This display method is effective to locate the features of speech.

## 6. EXPERIMENTS IN PARCOR SYNTHESIS

We describe the experimental result to improve the speech quality in PARCOR synthesis.

### 6.1 Excitation source

- (1) Voiced excitation source

A sequence of triangle waveform<sup>6)</sup> as shown in Fig. 7 produced the best speech quality. Here,  $\tan \alpha : \tan \beta = 4 : 9$ .

- (2) Unvoiced excitation source

Random noise with zero mean and frame power being set same as that in voiced sound produces best quality.

### 6.2 Magnitude of Voiced Sound

Zero-th autocorrelation computed in PARCOR analysis represents the aver-

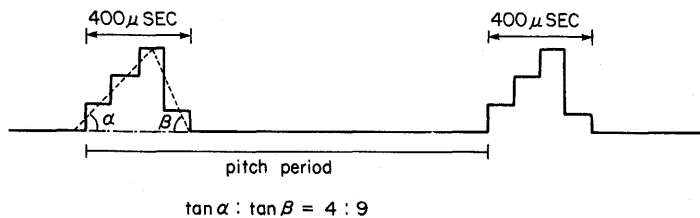


Fig. 7. Triangle waveform for excitation source.

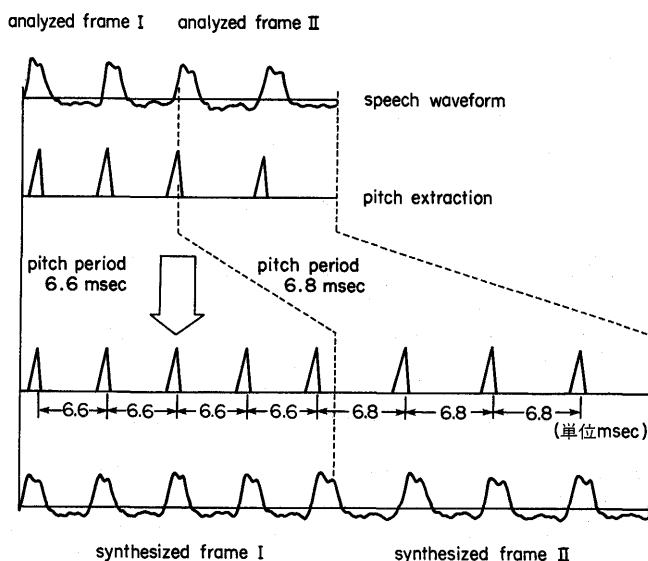


Fig. 8. Frame expansion in voiced sound.

aged power for the frame. Therefore, square root of the zero-th autocorrelation is best for the magnitude of excitation source of voiced sound. To implement the square root on DSP, left shift number  $k$  of zero-th autocorrelation before overflow is counted. The value  $(1/\sqrt{2})^k$  is used as semi-square root of zero-th autocorrelation.

### 6.3 Variable Speed Play-back without Changing Tone

Speech play-back method with variable speed control without changing tone is designed on the basis of PARCOR synthesis. We assume that human speaks slow and fast by varying the length of voiced sound. Based on this assumption, frames in voiced sound are synthesized by increasing the number of frames without changing PARCOR coefficients and pitch period. As for unvoiced sound, it is synthesized without processing.

Fig. 8 shows the slow speed play-back by expanding the voiced frame. Pitch period is repeated till desired duration is obtained without changing PARCOR coefficients and pitch period. Pitch continuity is guaranteed as shown in Fig. 8 between two kind of frames to be repeated.

## 7. CONCLUSION

Real-time speech analysis-synthesis system using digital signal processors is described. Software is swapped on the DSP, micro-computer and super-mini-computer. As the analysis software, 128 points FFT, bandpass filter bank, PARCOR coefficients and pitch period are developed on one DSP. The 64 points IFFT and PARCOR synthesis is also developed on one DSP as the synthesis software. In PARCOR synthesis, variable speed play-back without changing speech

tone is implemented. This system will be effective not only as a real time speech analysis and synthesis system but also as a vocoder system in which speech is coded in several coding methods suitable for application on a micro-computer or a super-minicomputer.

#### REFERENOES

- 1) J. A. Feldman, E. M. Hoftsetter, and M. L. Malpass: "A compact, flexible LPC vocoder based on a commercial signal processing microcomputer", IEEE Journal of Solid-state Circuits, Vol. SC-18, No. 1, Feb. 1983, 4-9.
- 2) N.E.C. Microcomputers,  $\mu$ PD7720 Signal Processing Interface User's Manual.
- 3) J. Le Roux, and C. Gueguen: "A fixed point computation of partial correlation coefficients", IEEE Trans., ASSP-25 (1977), 257-259.
- 4) B. Gold: "Computer program for pitch extraction", J. Acoust. Soc. Am., Vol. 34, 1962, 916-921.
- 5) B. Gold and L. Rabiner: "Parallel processing techniques for estimating pitch periods of speech in the time domain", J. Acoust. Soc. Am., Vol. 46, Aug. 1969, 442-448.
- 6) K. N. Stevens: "Physics of laryngeal behavior and larynx models", *Phonetica*, 34, 1977, 264.  
(Aug. 31, 1985, received)