TECHNICAL REPORT

A real-time formant tracker based on the inverse filter control method

Yuichi Ueda[1,*], Tomoya Hamakawa[1], Tadashi Sakata[1],
Syota Hario[1] and Akira Watanabe[2]

[1]*Graduate School of Science and Technology, Kumamoto University,*
2–39–1, Kurokami, Kumamoto, 860–8555 Japan
[2]*Kumamoto Prefectural College of Technology,*
4455–1, Haramizu, Kikuyou-town, Kikuchi-gun, Kumamoto, 869–1102 Japan

(*Received 23 October 2006, Accepted for publication 24 November 2006*)

Abstract: Formant frequencies are very important speech features in the area of speech perception and practical application. Although some formant estimation methods have been proposed previously, those have been inferior in real-time operation. Previously, we have proposed a new type of formant estimation based on IFC (Inverse Filter Control) method and applied into speech analysis and application systems in our study. Since the IFC method is based on speech waveform processing, the realization of a real-time system has been expected. On the other hand, recently the cheap electric boards which mounted a very high-speed DSP (Digital Signal Processor) have been marketed. We have applied such a DSP board into developing a real-time formant tracker based on the IFC method. By correcting the original C-language program for DSP operation, we have confirmed the real-time operation, which have estimated four formant frequencies from F_1 to F_4. Moreover, those estimates were almost the same as those by the original PC simulation. In this paper, we describe system modification for DSP realization and discuss effects of software modifications from C-language into DSP program.

Keywords: Formant estimation, DSP system, IFC method, Real-time system

PACS number: 43.72.Ar [doi:10.1250/ast.28.271]

1. INTRODUCTION

We proposed a formant estimation method based on the inverse filter control (IFC) [1]. As the formant estimation methods by software, the linear prediction method [2], the analysis-by-synthesis method [3], and so on are known well. However, since the complicated calculations and the repetitions of processing are necessary, those methods are not suitable for the real-time processing. On the other hand, since the formant tracker by IFC method is based on decomposing speech signal into several formant components and measuring those weighted means of zero-crossing frequencies in a time domain, it is able to estimate formant frequencies with high-accuracy and at high-speed [1]. Therefore, the IFC-based formant estimation at the PC simulation level has been used in our application systems: for example, the speech visualization system [4], the single resonant analysis type of hearing aid [5], a speech processor for cochlear implant system [6], and so on. In practical use of those systems, it is necessary to develop a real-time formant tracker. Previously, we have designed a real-time system consisting of analog circuits [7] or a multi-DSP-based real-time system [8]. However, those systems were large-scale and could estimate two or three formant frequencies only. On the other hand, the technology in DSP (Digital Signal Processor) has progressed rapidly and then some higher-speed DSP chips have been used in various application fields. By making such a situation into a background, we have developed the IFC-based real-time formant tracker on an electric board with single chip of DSP. In this paper, we describe the way to realize the C-language-based software in PC simulation on a stand-alone type of DSP system, and discuss the processing speed in real-time formant estimation and the accuracy in comparison with PC-simulation.

2. TARGET IFC-SYSTEM FOR REAL TIME FORMANT ESTIMATION

The IFC-based formant estimation system has been proposed based on the following original ideas and investigations [1]:

(1) A total of 32 inverse filters are repeatedly controlled

[*] e-mail: ueda@cs.kumamoto-u.ac.jp

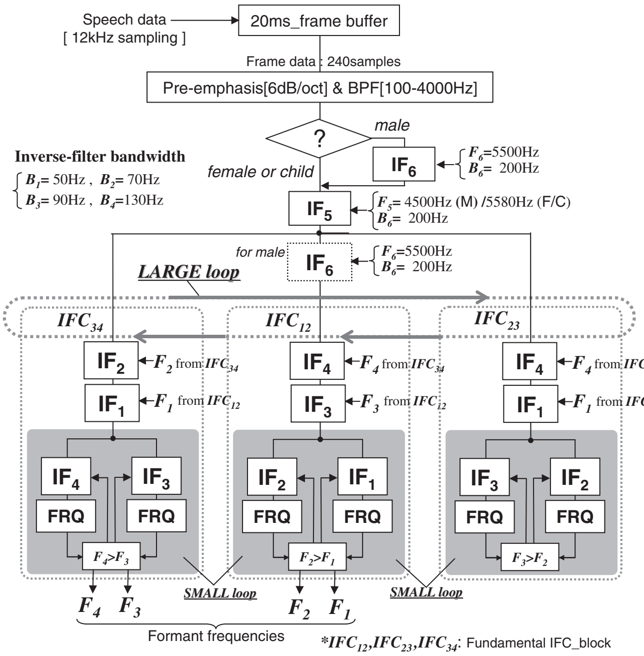


Fig. 1 Target IFC-system for realization of real-time formant tracker.

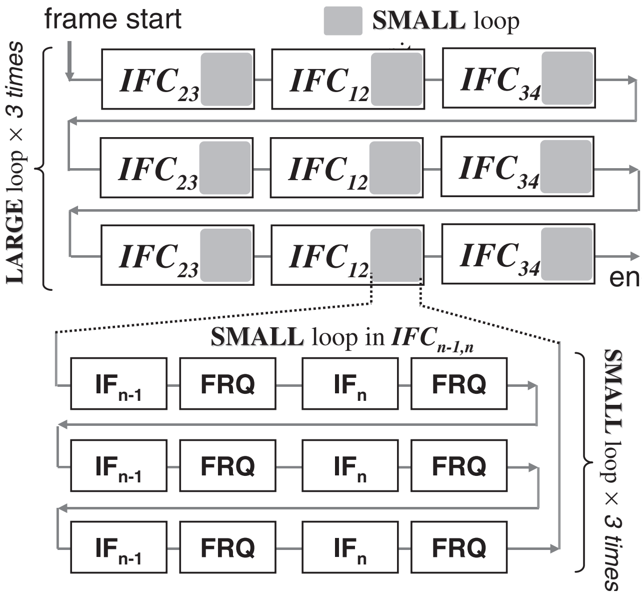


Fig. 2 Sequential operation of LARGE/SMALL loops in IFC-system.

to separate each of single resonant components from speech signal and then each formant frequency is estimated as a weighted mean of zero-crossing frequencies from signal of the respective component.

(2) At least three times of repeated control is necessary for the estimates to converge.

(3) All processing are executed in a time domain for a high-speed estimation.

(4) To obtain accurate formant frequencies, zero-crossing points should be determined by a linear interpolation between adjacent sample's amplitudes with different signs.

(5) Speaker's vocal tract length affects the number of formants in a signal bandwidth for analysis. As a result, it is desirable to adjust suitably the number of formants to be estimated and a characteristic of pre-filter according as the speaker is a male, a female, or a child. (The actual system estimates six formants for a male, five for a female, and four or five for a child respectively, within a signal bandwidth of 6 kHz.)

In this study, we aimed at realizing a real-time formant tracker with a single-chip-DSP-board that adopted the above features as faithfully as possible. Figure 1 shows a framework of the formant tracker designed in this study. As a compromise for a real time system, only the number of formants to be estimated has first been restricted to four within 4 kHz of speech signals because the number was enough for the application systems [4–6]. In the system of Fig. 1, speech signals as a frame data are led to the control routine consisting of three fundamental blocks, via pre-emphasis, anti-aliasing filter, and pre-filter. In Fig. 1, IF$_n$ shows an inverse filter to eliminate the nth resonant component, and FRQ is a routine for computing a weighted mean of zero-crossing frequency. This system first extracts temporary formants by sending signals from the top to the bottom in the figure. Next, the obtained formants control zeros of the corresponding inverse filters from the right to the left. Since the temporary formants (i.e. the control signals) are always updated by repetitions, we get finally the lowest four formant-frequencies from the outputs of fundamental blocks as a result of convergence. The control process is repeated three times in a frame. In Fig. 1, there are two kinds of control loops, that is, one is LARGE loop which controls fundamental blocks (IFC$_{12}$, IFC$_{23}$, IFC$_{34}$) sequentially, and the other is SMALL loop that controls mutually two inverse filters in the individual fundamental blocks. Figure 2 shows sequential operations of those loops to be used in a program.

After designing the system in Fig. 1, we simulated the whole system using C-language on a PC and confirmed that the four formant trajectories consisting of the estimated frequencies were very close to those by the original system [1], which is capable of estimating a maximum of six formant frequencies and has reliable accuracies of them (see Fig. 4). Thus, the program by C-language is useful for an accurate formant-estimation apart from whether it makes real-time operation possible on a DSP system or not. Next, we investigated a system-specification for a real time operation. The sampling rate, frame length and frame shift are 12 kHz, 20 ms, and 10 ms respectively. So, all the processing of a frame must end within 10 ms for the real-time operation.

3. PRACTICAL DESIGN OF DSP SYSTEM FOR REAL-TIME OPERATION

3.1. Specification of DSP Board and Direct Implementation by C-Program

We have adopted a commercial board, TMS320C6713-DSK (Texas Instruments Inc.) [9,10], which was equipped with a 32 bit floating-point DSP chip (TMS320C6713; 230 MHz). The software tool, Code Composer Studio (CCS) can compile C-language into codes of DSP and directly download those to DSP board. We first tried to implement, without any change, the compiled codes of C-program whose operation was confirmed on PC. As a result, the execution time per a frame reached about 94 ms which was much longer than 10 ms as the time limit. Thus, we have used modifications in the program described below to realize the computation at higher speed.

3.2. Three Modifications in DSP Programming

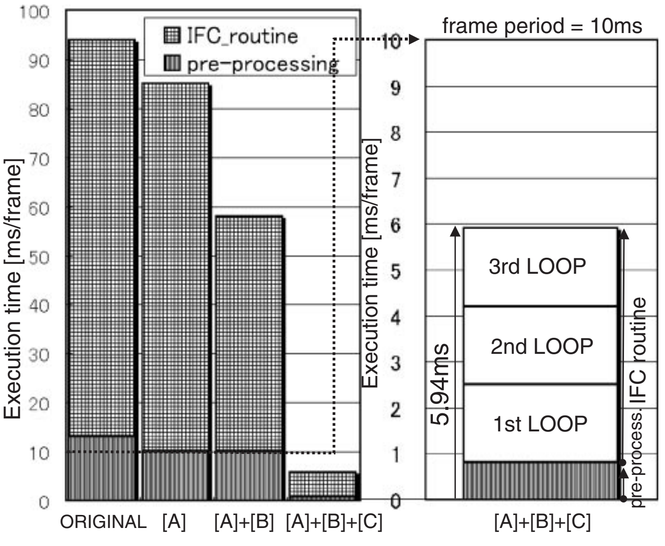
Modifications in DSP programming for real-time operation are as follows:

[A] Replacing a data type of variables:
Normally, "float" type of data is prepared as 32bit-operation in the DSP. In dealing with "double" type of data, modifications of the register structures are needed and, as a result, the execution time increases. Therefore, we replaced all "double" type variables in C-program with "float" type.

[B] Using "Intrinsic function" for fundamental numerical functions:
Generally, it takes more time to execute some numerical functions, such as a division, a square root, and so on. In DSP codes, some fundamental operations named intrinsic functions are prepared so as to implement those calculations with a few steps. Therefore, we have replaced all the corresponding operations with the intrinsic functions respectively.

[C] Optimizing repeated computation by software pipeline:
Since a software pipeline for parallel processing, which was known as a high-speed processing, is prepared in the target DSP, we have used the software pipeline for the repetition of loop operations of C-program.

Figure 3(a) shows changes in processing time by the above modifications [A], [B], and [C], which are indicated separately for the pre-processing and the IFC routine. The processing time can be obtained by measuring pulse-widths with an oscilloscope. The pulse-levels are switched to record the beginning and end in each of the pre-processing and IFC routines in program. We have extracted the pulse-signals from the board together with a marker signal of frame processing and measured the widths of pulses repeated synchronizing with the frame signal.



(a) Improvements in processing speed **(b) Final real time operation**

Fig. 3 Execution time in a frame-processing on DSP system. [A]: using Float type of data, [B]: using Intrinsic functions, [C]: using software pipeline.

As seen in Fig. 3(a), it is possible to reduce the processing time to 6 ms by using the three modifications. Especially, the software pipeline of [C] is remarkably effective. Fig. 3(b) shows the processing time-lengths occupied by each of the preprocessing and three LARGE loops in the fastest operation. Based on the results, we can confirm the real-time operation in the realized formant tracker.

3.3. Evaluation of the Estimated Formant Frequencies

The real-time operation based on DSP programming includes some approximations like the intrinsic functions (see 3.2[B]), where a square root, a reciprocal and so on are obtained as the approximated solution of an equation by two repeated calculations in the Newton-Raphson method. Therefore, the formant frequencies estimated by the real-time DSP system were compared with those of a PC simulation by C-program. In the DSP operation, speech signal in PC was moved into an external memory on the DSP board, and then processed for every frame using an internal timer's interrupt. Speech materials for the comparison have been 40 words uttered by two males and two females. In this case, errors were defined as the averages of absolute differences in two kinds of formant trajectories which consisted of 3,904 voiced frames in total. The errors in F_1–F_4 were 0.01–0.09 Hz for 1,866 frames of male voice and likewise below 0.01 Hz for 2,038 frames of female voice. The difference significant in both groups was not able to be seen. From the results, we regarded the errors in DSP operation as at most 0.1 Hz in comparison with the estimates by PC simulation.

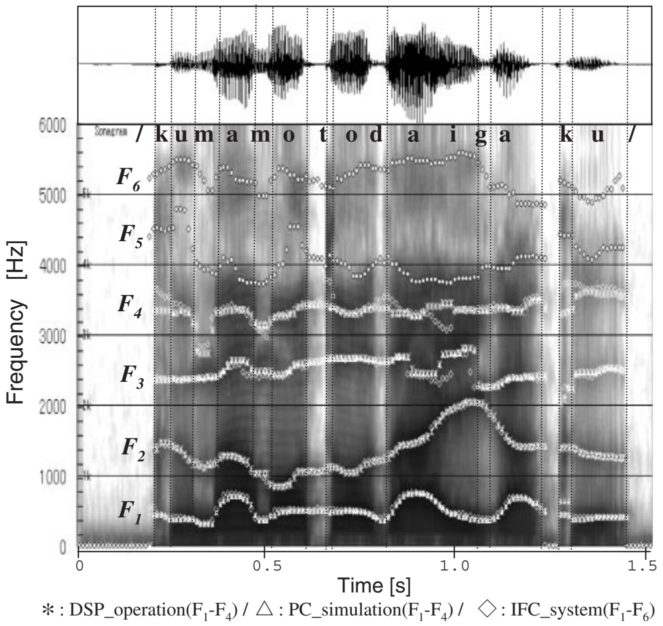


Fig. 4 An example of the estimated formant trajectories of an utterance/kumamoto daigaku/by a male speaker. The results of both the PC simulation and the DSP operation are plotted with the estimates in the original IFC system.

Figure 4 shows an example of the estimated formant trajectories of F_1–F_4. Although the results of both PC_simulation (△) and DSP_operation (∗) are plotted in the figure, those differences are almost indistinguishable because of overlapping. For reference, six formants (◇) estimated by the original IFC_system(F_1–F_6) are plotted in Fig. 4 too. (The original IFC_system is able to estimate six formant frequencies from F_1 to F_6, whose estimating errors have already been investigated using synthetic and real speech [1]) In comparison between estimates by IFC_system and those by the DSP_operation, some small differences in F_3 and F_4 can be seen. That is caused by a constant restriction (100–4,000 Hz) of signal bandwidth in the DSP system and in the PC_simulation. However, there is no distinguishable error in F_1 and F_2. Consequently, it has been confirmed that three modifications (see 3.2 [A], [B], and [C]) do not affect the precision of formant estimates. Thus, the realized DSP system operates in high accuracy and in real time.

4. CONCLUSIONS

We have developed the real-time formant tracker by using a marketed DSP board which has a single DSP chip. The system has been realized by adopting the inverse filter control method as faithfully as possible. To make real-time operation possible, three modifications in software have been considered for the DSP system. Among them, high-speed operation by software pipeline was remarkably effective. The final system has operated within about 6.0 ms per a frame to estimate the lowest four formants. Using 4.0 ms of the time margin, incorporation of the DSP-based real-time system into the practical systems would be possible. Finally, by using many utterances, we confirmed that the formant trajectories estimated by the DSP real-time system were almost the same as those by the accurate software system on PC.

REFERENCES

[1] A. Watanabe, “Formant estimation method using inverse-filter control,” *IEEE Trans. Speech Audio Process.*, **9**, 317–326 (2001).

[2] J. D. Markel and A. H. Gray, *Linear Prediction of Speech* (Springer Verlag, Amsterdam, 1976).

[3] C. G. Bell, H. Fujisaki, J. M. Heinz, K. N. Stevens and A. S. House, “Reduction of speech spectra by analysis-by-synthesis techniques,” *J. Acoust. Soc. Am.*, **33**, 1725–1736 (1961).

[4] A. Watanabe, S. Tomishige and M. Nakatake, “Speech visualization by integrating features for the hearing impaired,” *IEEE Trans. Speech Audio Process.*, **8**, 454–466 (2000).

[5] T. Ikeda, Y. Ueda and A. Watanabe, “A new hearing aid based on the single resonant analysis for delivering high quality speech sounds,” *J. Acoust. Soc. Jpn. (J)*, **57**, 326–336 (2001).

[6] M. Satou, T. Sakata, A. Watanabe and Y. Ueda, “Formant peak stimulating method based on Phantom sensation for Cochlear implant system,” *Proc. WESPACI IX*, CD-ROM, hu-2-5-370 (2006).

[7] A. Watanabe, Y. Ueda and A. Shigenaga, “Color display system for connected speech to be used for the hearing impaired,” *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-33**, 164–173 (1985).

[8] A. Watanabe, T. Ikeda and Y. Ueda, “Real-time visualization of telephonic speech,” *Proc. 14th Int. Congr. Acoust.*, H3-1 (1992).

[9] “TMS320C62x/C67x Programmer’s Guide,” Texas Instruments (1998).

[10] “TMS320C6713DSK Technical Reference,” Texas Instruments (2003).