

(19) **DANMARK**

(10) **DK/EP 3579228 T3**



(12)

Oversættelse af  
europæisk patentskrift

Patent- og  
Varemærkestyrelsen

- 
- (51) Int.Cl.: **G 10 L 19/005 (2013.01)** **G 10 L 19/09 (2013.01)** **H 03 M 7/30 (2006.01)**
- (45) Oversættelsen bekendtgjort den: **2025-04-22**
- (80) Dato for Den Europæiske Patentmyndigheds bekendtgørelse om meddelelse af patentet: **2025-04-02**
- (86) Europæisk ansøgning nr.: **19185490.0**
- (86) Europæisk indleveringsdag: **2013-11-12**
- (87) Den europæiske ansøgnings publiceringsdag: **2019-12-11**
- (30) Prioritet: **2012-11-15 JP 2012251646**
- (62) Stamansøgningsnr: **13854879.7**
- (84) Designerede stater: **AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR**
- (73) Patenthaver: **NTT DOCOMO, INC., 11-1, Nagatacho 2-chome, Chiyoda-ku, Tokyo, 100-6150, Japan**
- (72) Opfinder: **TSUTSUMI, Kimitaka, c/o Intellectual Property Department, NTT DOCOMO INC., Sanno Park Tower, 11-1 Nagatacho 2-chome, Chiyoda-ku, Tokyo 100-6150, Japan**  
**KIKUIRI, Kei, c/o Intellectual Property Department, NTT DOCOMO INC., Sanno Park Tower, 11-1 Nagatacho 2-chome, Chiyoda-ku, Tokyo 100-65150, Japan**  
**YAMAGUCHI, Atsushi, c/o Intellectual Property Department, NTT DOCOMO INC., Sanno Park Tower, 11-1 Nagatacho 2-chome, Chiyoda-ku, Tokyo 100-6150, Japan**
- (74) Fuldmægtig i Danmark: **Plougmann Vingtoft A/S, Strandvejen 70, 2900 Hellerup, Danmark**
- (54) Benævnelse: **AUDIOKODNINGSINDRETNING**
- (56) Fremdragne publikationer:  
**EP-A2- 1 288 915**  
**WO-A2-2012/110482**  
**JP-A- 2002 268 696**  
**JP-A- 2004 138 756**



# DESCRIPTION

Description

## Technical Field

**[0001]** The present invention relates to error concealment for transmission of audio packets through an IP network or a mobile communication network and, more specifically, relates to an audio encoding device for highly accurate packet loss concealment signal generation to implement error concealment.

## Background Art

**[0002]** In the transmission of audio and acoustic signals (which are collectively referred to hereinafter as "audio signal") through an IP network or a mobile communication network, the audio signal is encoded into audio packets at regular time intervals and transmitted through a communication network. At the receiving end, the audio packets are received through the communication network and decoded into a decoded audio signal by server, a MCU (Multipoint Control Unit), a terminal or the like.

**[0003]** The audio signal is generally collected in digital format. Specifically, it is measured and accumulated as a sequence of numerals whose number is the same as a sampling frequency per second. Each element of the sequence is called a "sample". In audio encoding, each time a predetermined number of samples of an audio signal is accumulated in a built-in buffer, the audio signal in the buffer is encoded. The above-described specified number of samples is called a "frame length", and a set of the same number of samples as the frame length is called "frame". For example, at the sampling frequency of 32 kHz, when the frame length is 20 ms, the frame length is 640 samples. Note that the length of the buffer may be more than one frame.

**[0004]** When transmitting audio packets through a communication network, a phenomenon (so-called "packet loss") can occur where some of the audio packets are lost, or an error can occur in part of information written in the audio packets due to congestion in the communication network or the like. In such a case, the audio packets cannot be correctly decoded at the receiving end, and therefore a desired decoded audio signal cannot be obtained. Further, the decoded audio signal corresponding to the audio packet where packet loss has occurred is detected as noise, which significantly degrades the subjective quality to a person who listens to the audio.

**[0005]** In order to overcome the above-described inconveniences, packet loss concealment technology is used as a way to interpolate a part of the audio/acoustic signal that is lost by packet loss. There are two types of packet loss concealment technology: "packet loss concealment technology without using side information" where packet loss concealment is performed only at the receiving end and "packet loss concealment technology using side information" where parameters that help packet loss concealment are obtained at the transmitting end and transmitted to the receiving end, where packet loss concealment is performed using the received parameters at the receiving end.

**[0006]** The "packet loss concealment technology without using side information" generates an audio signal corresponding to a part where packet loss has occurred by copying a decoded audio signal contained in a packet that has been correctly received in the past on a pitch-by-pitch basis and then multiplying it by a predetermined attenuation coefficient as described in Non Patent Literature 1, for example. Because the "packet loss concealment technology without using side information" is based on the assumption that the properties of the part of the audio where packet loss has occurred are similar to those of the audio immediately before the occurrence of loss, the concealment effect cannot be sufficiently obtained when the part of the audio where packet loss has occurred has different properties from the audio immediately before the occurrence of loss or when there is a sudden change in power.

**[0007]** On the other hand, the "packet loss concealment technology using side information" includes a technique that encodes parameters required for packet loss concealment at the transmitting end and transmits them for use in packet loss concealment at the receiving end as described in Patent Literature 1. In Patent Literature 1, the audio is encoded by two encoding methods: main encoding and redundant encoding. The redundant encoding encodes the frame immediately before the frame to be encoded by the main encoding at a lower bit rate than the main encoding (see Fig. 1 (a)). For example, the Nth packet contains an audio code obtained by encoding the Nth frame by major encoding and a side information code obtained by encoding the (N-1)th frame by redundant encoding.

**[0008]** The receiving end waits for the arrival of two or more temporally successive packets and then decodes the temporally earlier packet and obtains a decoded audio signal. For example, to obtain a signal corresponding to the Nth frame, the receiving end waits for the arrival of the (N+1)th packet and then performs decoding. In the case where the Nth packet and the (N+1)th packet are correctly received, the audio signal of the Nth frame is obtained by decoding the audio code contained in the Nth packet (see Fig. 1(b)). On the other hand, in the case where packet loss has occurred (when the (N+1)th packet is obtained in the condition where the Nth packet is lost), the audio signal of the Nth frame can be obtained by decoding the side information code contained in the (N+1)th packet (see Fig. 1(c)).

**[0009]** According to the method of Patent Literature 1, after a packet to be decoded arrives, it is necessary to wait to perform decoding until one or more packet arrives, and algorithmic delay increases by one packet or more. Accordingly, in the method of Patent Literature 1, although the audio quality can be improved by packet loss concealment, the algorithmic delay

increases to cause the degradation of the voice communication quality.

**[0010]** Further, in the case of applying the above-described packet loss concealment technology to CELP (Code Excited Linear Prediction) encoding, another problem caused by the characteristics of the operation of CELP arises. Because CELP is an audio model based on linear prediction and is able to encode an audio signal with high accuracy and with a high compression ratio, it is used in many international standards.

**[0011]** In CELP, an audio signal is synthesized by filtering an excitation signal  $e(n)$  using an all-pole synthesis filter. Specifically, an audio signal  $s(n)$  is synthesized according to the following equation:

$$s(n) = e(n) - \sum_{i=1}^P a(i) \cdot s(n-1) \quad \text{Equation 1}$$

where  $a(i)$  is a linear prediction coefficient (LP coefficient), and a value such as  $P=16$ , for example, is used as a degree.

**[0012]** The excitation signal is accumulated in a buffer called an adaptive codebook. When synthesizing the audio for a new frame, an excitation signal is newly generated by adding an adaptive codebook vector read from the adaptive codebook and a fixed codebook vector representing a change in excitation signal over time based on position information called a pitch lag. The newly generated excitation signal is accumulated in the adaptive codebook and is also filtered by the all-pole synthesis filter, and thereby a decoded signal is synthesized.

**[0013]** In CELP, an LP coefficient is calculated for all frames. In the calculation of the LP coefficient, a look-ahead signal of about 10 ms is required. Specifically, in addition to a frame to be encoded, a look-ahead signal is accumulated in the buffer, and then the LP coefficient calculation and the subsequent processing are performed (see Fig. 2). Each frame is divided into about four sub-frames, and processing such as the above-described pitch lag calculation, adaptive codebook vector calculation, fixed codebook vector calculation and adaptive codebook update are performed in each sub-frame. In the processing of each sub-frame, the LP coefficient is also interpolated so that the coefficient varies from sub-frame to sub-frame. Further, for quantization and interpolation, the LP coefficient is encoded after being converted into an ISP (Immittance Spectral Pair) parameter and an ISF (Immittance Spectral Frequency) parameter, which are equivalent representation(s) of the LP coefficient(s). A procedure for the interconversion of the LP coefficient(s) and the ISP parameter and the ISF parameter is described in Non Patent Literature 2.

**[0014]** In CELP encoding, encoding and decoding are performed based on the assumption that both of the encoding end and the decoding end have adaptive codebooks, and those adaptive codebooks are always synchronized with each other. Although the adaptive codebook at the encoding end and the adaptive codebook at the decoding end are synchronized under conditions where packets are correctly received and decoding correctly, once packet loss has occurred, the synchronization of the adaptive codebooks cannot be achieved.

**[0015]** For example, if a value that is used as a pitch lag is different between the encoding end and the decoding end, a time lag occurs between the adaptive codebook vectors. Because the adaptive codebook is updated with those adaptive codebook vectors, even if the next frame is correctly received, the adaptive codebook vector calculated at the encoding end and the adaptive codebook vector calculated at the decoding end do not coincide, and the synchronization of the adaptive codebooks is not recovered. Due to such inconsistency of the adaptive codebooks, the degradation of the audio quality occurs for several frames after the frame where packet loss has happened.

**[0016]** In the packet loss concealment in CELP encoding, a more advanced technique is described in Patent Literature 2. According to Patent Literature 2, an index of a transition mode codebook is transmitted instead of a pitch lag or an adaptive codebook gain in a specific frame that is largely affected by packet loss. The technique of Patent Literature 2 focuses attentions on a transition frame (transition from a silent audio segment to a sound audio segment, or transition between two vowels) as the frame that is largely affected by packet loss. By generating an excitation signal using the transition mode codebook in this transition frame, it is possible to generate an excitation signal that is not dependent on the past adaptive codebook and thereby recover from the inconsistency of the adaptive codebooks due to the past packet loss.

**[0017]** However, because the method of Patent Literature 2 does not use the transition frame codebook in a frame where a long vowel continues, for example, it is not possible to recover from the inconsistency of the adaptive codebooks in such a frame. Further, in the case where the packet containing the transition frame codebook is lost, packet loss affects the frames after the loss. This is the same when the next packet after the packet containing the transition frame codebook is lost.

**[0018]** Although it is feasible to apply a codebook to all frames that is not dependent on the past frames, such as the transition frame codebook, because the encoding efficiency is significantly degraded, it is not possible to achieve a low bit rate and high audio quality under these circumstances. JP 2004 138756 A describes that, when the vanished frame of a voice signal is caused in a voice decoding device, the compensation processing of the vanished frame is performed on the basis of pitch lag information for the compensation processing of vanished frame by computing a pitch lag for the compensation processing of frame vanishing in the voice coding device and by transmitting information expressing the pitch lag to the voice decoding device together with the coded voice signal. WO 2012/110482 A2 describes saving valuable bitrate with maintaining the noise generation quality within inactive phases, by continuously updating the parametric background noise estimate during an active phase so that the noise generation may immediately be started with upon the entrance of an inactive phase following the active phase.

#### **Citation List**

**Patent Literature**

**[0019]**

Patent Literature 1: Japanese Unexamined Patent Application Publication No. 2003-533916

Patent Literature 2: Japanese Unexamined Patent Application Publication No. 2010-507818

**Non Patent Literature**

**[0020]**

Non Patent Literature 1: ITU-T G711 Appendix I

Non Patent Literature 2: 3GPP TS26-191

Non Patent Literature 3: 3GPP TS26-190

Non Patent Literature 4: ITU-T G718

**Summary of Invention**

**Technical Problem**

**[0021]** With use of the method of Patent Literature 1, after the arrival of a packet to be decoded, decoding is not started before the arrival of the next packet. Therefore, although the audio quality is improved by packet loss concealment, the algorithmic delay increases, which causes the degradation of the voice communication quality.

**[0022]** In the event of packet loss in CELP encoding, the degradation of the audio quality occurs due to the inconsistency of the adaptive codebooks between the encoding unit and the decoding unit. Although the method as described in Patent Literature 2 can allow for recovery from the inconsistency of the adaptive codebooks, the method is not sufficient to allow recovery when a frame different from the frame immediately before the transition frame is lost.

**[0023]** The present invention has been accomplished to solve the above problems and an object of the present invention is thus to provide an audio encoding device, an audio encoding

method, an audio encoding program, an audio decoding device, an audio decoding method, and an audio decoding program that recover audio quality without increasing algorithmic delay in the event of packet loss in audio encoding.

### **Solution to Problem**

**[0024]** To achieve the above object, an audio coding device for encoding an audio signal according to one aspect of the present invention has the features of claim 1.

**[0025]** The side information is related to a pitch lag in a look-ahead signal and may be related to a pitch gain in a look-ahead signal, or related to a pitch lag and a pitch gain in a look-ahead signal. Further, the side information may contain information related to availability of the side information.

**[0026]** The side information encoding unit calculates side information for a look-ahead signal part and may encode the side information, and also generate a concealment signal, and the audio encoding device may further include an error signal encoding unit configured to encode an error signal between an input audio signal and a concealment signal output from the side information encoding unit, and a main encoding unit configured to encode an input audio signal.

### **Advantageous Effects of Invention**

**[0027]** It is possible to recover audio quality without increasing algorithmic delay in the event of packet loss in audio encoding. Particularly, in CELP encoding, it is possible to reduce degradation of an adaptive codebook that occurs when packet loss happens and thereby improve audio quality in the event of packet loss.

### **Brief Description of Drawings**

#### **[0028]**

Fig. 1 is a view showing a temporal relationship between packets and a decoded signal according to related art described in Patent Literature 1.

Fig. 2 is a view showing a temporal relationship between an LP analysis target signal and a look-ahead signal in CELP encoding.

Fig. 3 is a view showing a temporal relationship between packets and a decoded signal according to an embodiment of the present invention.

Fig. 4 is a view showing a functional configuration example of an audio signal transmitting device in an example 1 (first example) of the present invention.

Fig. 5 is a view showing a functional configuration example of an audio signal receiving device in the example 1 of the present invention.

Fig. 6 is a view showing a procedure of the audio signal transmitting device in the example 1 of the present invention.

Fig. 7 is a view showing a procedure of the audio signal receiving device in the example 1 of the present invention.

Fig. 8 is a view showing a functional configuration example of a side information encoding unit in the example 1 of the present invention.

Fig. 9 is a view showing a procedure of the side information encoding unit in the example 1 of the present invention.

Fig. 10 is a view showing a procedure of an LP coefficient calculation unit in the example 1 of the present invention.

Fig. 11 is a view showing a procedure of a target signal calculation unit in the example 1 of the present invention.

Fig. 12 is a view showing a functional configuration example of an audio parameter missing processing unit in the example 1 of the present invention.

Fig. 13 is a view showing a procedure of audio parameter prediction in the example 1 of the present invention.

Fig. 14 is a view showing a procedure of an excitation vector synthesis unit in an alternative example 1-1 of the example 1 of the present invention.

Fig. 15 is a view showing a functional configuration example of an audio synthesis unit in the example 1 of the present invention.

Fig. 16 is a view showing a procedure of the audio synthesis unit in the example 1 of the present invention.

Fig. 17 is a view showing a functional configuration example of a side information encoding unit (when a side information output determination unit is included) in an alternative example 1-2 of the example 1 of the present invention.

Fig. 18 is a view showing a procedure of the side information encoding unit (when the side information output determination unit is included) in the alternative example 1-2 of the example 1 of the present invention.

Fig. 19 is a view showing a procedure of audio parameter prediction in the alternative example 1-2 of the example 1 of the present invention.

Fig. 20 is a view showing a functional configuration example of an audio signal transmitting device in an example 2 of the present invention.

Fig. 21 is a view showing a functional configuration example of a main encoding unit in the example 2 of the present invention.

Fig. 22 is a view showing a procedure of the audio signal transmitting device in the example 2 of the present invention.

Fig. 23 is a view showing a functional configuration example of an audio signal receiving device in the example 2 of the present invention.

Fig. 24 is a view showing a procedure of the audio signal receiving device in the example 2 of the present invention.

Fig. 25 is a view showing a functional configuration example of an audio synthesis unit in the example 2 of the present invention.

Fig. 26 is a view showing a functional configuration example of an audio parameter decoding unit in the example 2 of the present invention.

Fig. 27 is a view showing a functional configuration example of a side information encoding unit in an example 3 of the present invention.

Fig. 28 is a view showing a procedure of the side information encoding unit in the example 3 of the present invention.

Fig. 29 is a view showing a procedure of a pitch lag selection unit in the example 3 of the present invention.

Fig. 30 is a view showing a procedure of a side information decoding unit in the example 3 of the present invention.

Fig. 31 is a view showing a configuration of an audio encoding program and a storage medium according to an embodiment of the present invention.

Fig. 32 is a view showing a configuration of an audio decoding program and a storage medium according to an embodiment of the present invention.

Fig. 33 is a view showing a functional configuration example of a side information encoding unit in an example 4 of the present invention.

Fig. 34 is a view showing a procedure of the side information encoding unit in the example 4 of the present invention.

Fig. 35 is a view showing a procedure of a pitch lag prediction unit in the example 4 of the present invention.

Fig. 36 is another view showing a procedure of the pitch lag prediction unit in the example 4 of

the present invention.

Fig. 37 is another view showing a procedure of the pitch lag prediction unit in the example 4 of the present invention.

Fig. 38 is a view showing a procedure of an adaptive codebook calculation unit in the example 4 of the present invention.

Fig. 39 is a view showing a functional configuration example of a side information encoding unit in an example 5 of the present invention.

Fig. 40 is a view showing a procedure of a pitch lag encoding unit in the example 5 of the present invention.

Fig. 41 is a view showing a procedure of a side information decoding unit in the example 5 of the present invention.

Fig. 42 is a view showing a procedure of a pitch lag prediction unit in the example 5 of the present invention.

Fig. 43 is a view showing a procedure of an adaptive codebook calculation unit in the example 5 of the present invention.

### **Description of Embodiments**

**[0029]** Embodiments of the present invention are described hereinafter with reference to the attached drawings. Note that, where possible, the same elements are denoted by the same reference numerals and redundant description thereof is omitted.

**[0030]** An embodiment of the present invention relates to an encoder and a decoder that implement "packet loss concealment technology using side information" that encodes and transmits side information calculated on the encoder side for use in packet loss concealment on the decoder side.

**[0031]** In the embodiments of the present invention, the side information that is used for packet loss concealment is contained in a previous packet. Fig. 3 shows a temporal relationship between an audio code and a side information code contained in a packet. As illustrated in Fig. 3, the side information in the embodiments of the present invention is parameters (pitch lag, adaptive codebook gain, etc.) that are calculated for a look-ahead signal in CELP encoding.

**[0032]** Because the side information is contained in a previous packet, it is possible to perform decoding without waiting for a packet that arrives after a packet to be decoded. Further, when

packet loss is detected, because the side information for a frame to be concealed is obtained from the previous packet, it is possible to implement highly accurate packet loss concealment without waiting for the next packet.

**[0033]** In addition, by transmitting parameters for CELP encoding in a look-ahead signal as the side information, it is possible to reduce the inconsistency of adaptive codebooks even in the event of packet loss.

**[0034]** The embodiments of the present invention can be composed of an audio signal transmitting device (audio encoding device) and an audio signal receiving device (audio decoding device). A functional configuration example of an audio signal transmitting device is shown in Fig. 4, and an example procedure of the same is shown in Fig. 6. Further, a functional configuration example of an audio signal receiving device is shown in Fig. 5, and an example procedure of the same is shown in Fig. 7.

**[0035]** As shown in Fig. 4, the audio signal transmitting device includes an audio encoding unit 111 and a side information encoding unit 112. As shown in Fig. 5, the audio signal receiving device includes an audio code buffer 121, an audio parameter decoding unit 122, an audio parameter missing processing unit 123, an audio synthesis unit 124, a side information decoding unit 125, and a side information accumulation unit 126.

**[0036]** The audio signal transmitting device encodes an audio signal for each frame and can transmit the audio signal by the example procedure shown in Fig. 6.

**[0037]** The audio encoding unit 111 can calculate audio parameters for a frame to be encoded and output an audio code (Step S131 in Fig. 6).

**[0038]** The side information encoding unit 112 can calculate audio parameters for a look-ahead signal and output a side information code (Step S132 in Fig. 6).

**[0039]** It is determined whether the audio signal ends, and the above steps can be repeated until the audio signal ends (Step S133 in Fig. 6).

**[0040]** The audio signal receiving device decodes a received audio packet and outputs an audio signal by the example procedure shown in Fig. 7.

**[0041]** The audio code buffer 121 waits for the arrival of an audio packet and accumulates an audio code. When the audio packet has correctly arrived, the processing is switched to the audio parameter decoding unit 122. On the other hand, when the audio packet has not correctly arrived, the processing is switched to the audio parameter missing processing unit 123 (Step S141 in Fig. 7).

**<When audio packet is correctly received>**

**[0042]** The audio parameter decoding unit 122 decodes the audio code and outputs audio parameters (Step S142 in Fig. 7).

**[0043]** The side information decoding unit 125 decodes the side information code and outputs side information. The outputted side information is sent to the side information accumulation unit 126 (Step S143 in Fig. 7).

**[0044]** The audio synthesis unit 124 synthesizes an audio signal from the audio parameters output from the audio parameter decoding unit 122 and outputs the synthesized audio signal (Step S144 in Fig. 7).

**[0045]** The audio parameter missing processing unit 123 accumulates the audio parameters output from the audio parameter decoding unit 122 in preparation for packet loss (Step S145 in Fig. 7).

**[0046]** The audio code buffer 121 determines whether the transmission of audio packets has ended, and when the transmission of audio packets has ended, stops the processing. While the transmission of audio packets continues, the above Steps S141 to S146 are repeated (Step S147 in Fig. 7).

**<When audio packet is lost>**

**[0047]** The audio parameter missing processing unit 123 reads the side information from the side information accumulation unit 126 and carries out prediction for the parameter(s) not contained in the side information and thereby outputs the audio parameters (Step S146 in Fig. 7).

**[0048]** The audio synthesis unit 124 synthesizes an audio signal from the audio parameters output from the audio parameter missing processing unit 123 and outputs the synthesized audio signal (Step S144 in Fig. 7).

**[0049]** The audio parameter missing processing unit 123 accumulates the audio parameters output from the audio parameter missing processing unit 123 in preparation for packet loss (Step S145 in Fig. 7).

**[0050]** The audio code buffer 121 determines whether the transmission of audio packets has ended, and when the transmission of audio packets has ended, stops the processing. While the transmission of audio packets continues, the above Steps S141 to S146 are repeated (Step S147 in Fig. 7).

**[Example 1]**

**[0051]** In this example of a case where a pitch lag is transmitted as the side information, the pitch lag can be used for generation of a packet loss concealment signal at the decoding end.

**[0052]** The functional configuration example of the audio signal transmitting device is shown in Fig. 4, and the functional configuration example of the audio signal receiving device is shown in Fig. 5. An example of the procedure of the audio signal transmitting device is shown in Fig. 6, and an example of the procedure of the audio signal receiving device is shown in Fig. 7.

**<Transmitting end>**

**[0053]** In the audio signal transmitting device, an input audio signal is sent to the audio encoding unit 111.

**[0054]** The audio encoding unit 111 encodes a frame to be encoded by CELP encoding (Step 131 in Fig. 6). For the details of CELP encoding, the method described in Non Patent Literature 3 is used, for example. The details of the procedure of CELP encoding are omitted. Note that, in the CELP encoding, local decoding is performed at the encoding end. The local decoding is to decode an audio code also at the encoding end and obtain parameters (ISP parameter and corresponding ISF parameter, pitch lag, long-term prediction parameter, adaptive codebook, adaptive codebook gain, fixed codebook gain, fixed codebook vector, etc.) required for audio synthesis. The parameters obtained by the local decoding include: at least one or both of the ISP parameter and the ISF parameter, the pitch lag, and the adaptive codebook, which are sent to the side information encoding unit 112. In the case where the audio encoding as described in Non Patent Literature 4 is used in the audio encoding unit 111, an index representing the characteristics of a frame to be encoded may also be sent to the side information encoding unit 112. In embodiments, encoding different from CELP encoding may be used in the audio encoding unit 111. In embodiments using different encoding, at least one or both of the ISP parameter and the ISF parameter, the pitch lag, and the adaptive codebook can be separately calculated from an input signal, or a decoded signal obtained by the local decoding, and sent to the side information encoding unit 112.

**[0055]** The side information encoding unit 112 calculates a side information code using the parameters calculated by the audio encoding unit 111 and the look-ahead signal (Step 132 in Fig. 6). As shown in the example of Fig. 8, the side information encoding unit 112 includes an LP coefficient calculation unit 151, a target signal calculation unit 152, a pitch lag calculation unit 153, an adaptive codebook calculation unit 154, an excitation vector synthesis unit 155, an adaptive codebook buffer 156, a synthesis filter 157, and a pitch lag encoding unit 158. An example procedure in the side information encoding unit is shown in Fig. 9.

**[0056]** The LP coefficient calculation unit 151 calculates an LP coefficient using the ISF parameter calculated by the audio encoding unit 111 and the ISF parameter calculated in the

past several frames (Step 161 in Fig. 9). The procedure of the LP coefficient calculation unit 151 is shown in Fig. 10.

**[0057]** First, the buffer is updated using the ISF parameter obtained from the audio encoding unit 111 (Step 171 in Fig. 10). Next, the ISF parameter  $\omega_j$  in the look-ahead signal is calculated. The ISF parameter  $\omega_j$  is calculated by the following equation (Step 172 in Fig. 10).

$$\dot{\omega}_i = \alpha \omega_i^{(-1)} + (1 - \alpha) \bar{\omega}_i \quad \text{Equation 2}$$

$$\bar{\omega}_i = \beta \omega_i^C + (1 - \beta) \frac{\omega_i^{(3)} + \omega_i^{(2)} + \omega_i^{(1)}}{3} \quad \text{Equation 3}$$

where

$$\omega_i^{(-j)}$$

is the ISF parameter, stored in the buffer, which is for the frame preceding by j-number of frames. Further,

$$\omega_i^C$$

is the ISF parameter during the speech period that is calculated in advance by learning or the like.  $\beta$  is a constant, and it may be a value such as 0.75, for example, though not limited thereto. Further,  $\alpha$  is also constant, and it may be a value such as 0.9, for example, though not limited thereto.

$$\omega_i^C$$

,  $\alpha$  and  $\beta$  may be varied by the index representing the characteristics of the frame to be encoded as in the ISF concealment described in Non Patent Literature 4, for example.

**[0058]** In addition, the values of i are arranged so that  $\omega_j$  satisfies  $0 < \omega_0 < \omega_1 < \dots < \omega_{14}$ , and the values of  $\omega_j$  can be adjusted so that the adjacent  $\omega_j$  is not too close. As a procedure to adjust the value of  $\omega_j$ , Non Patent Literature 4 (Equation 151) may be used, for example (Step 173 in Fig. 10).

**[0059]** After that, the ISF parameter  $\omega_j$  is converted into an ISP parameter and interpolation can be performed for each sub-frame. As a method of calculating the ISP parameter from the ISF parameter, the method described in the section 6.4.4 in Non Patent Literature 4 may be used, and as a method of interpolation, the procedure described in the section 6.8.3 in Non Patent Literature 4 may be used (Step 174 in Fig. 10).

**[0060]** Then, the ISP parameter for each sub-frame is converted into an LP coefficient

$$\dot{\alpha}_j^i (0 < i \leq P, 0 \leq j < M_{1a})$$

. The number of sub-frames contained in the look-ahead signal is  $M_{1a}$ . For the conversion from the ISP parameter to the LP coefficient, the procedure described in the section 6.4.5 in Non Patent Literature 4 may be used (Step 175 in Fig. 10).

**[0061]** The target signal calculation unit 152 calculates a target signal  $x(n)$  and an impulse response  $h(n)$  by using the LP coefficient

$$\dot{\alpha}_j^i$$

(Step 162 in Fig. 9). As described in the section 6.8.4.1.3 in Non Patent Literature 4, the target

signal is obtained by applying an perceptual weighting filter to a linear prediction residual signal (Fig. 11).

**[0062]** First, a residual signal  $r(n)$  of the look-ahead signal

$$s_{pre}^l(n) (0 \leq n < L')$$

is calculated using the LP coefficient according to the following equation (Step 181 in Fig. 11).

$$r(n) = s_{pre}^l(n) + \sum_{i=1}^{\mu} \hat{a}_i^l \cdot s_{pre}^l(n-i) \quad \text{Equation 4}$$

**[0063]** Note that  $L'$  indicates the number of samples of a sub-frame, and  $L$  indicates the number of samples of a frame to be encoded  $s_{pre}(n) (0 \leq n < L)$ . Then,

$$s_{pre}^l(n-p) = s_{pre}(n+L-p)$$

is satisfied.

**[0064]** In addition, the target signal  $x(n) (0 \leq n < L')$  is calculated by the following equations (Step 182 in Fig. 11).

$$e(n) = r(n) - \sum_{i=1}^{\mu} \hat{a}_i^l \cdot e(n-i) (0 \leq n < L') \quad \text{Equation 5}$$

$$e(n) = s(n+L-1) - \hat{s}(n+L-1) (-P \leq n < 0) \quad \text{Equation 6}$$

$$\hat{e}(n) = r(n) + \sum_{i=1}^P \hat{a}_i^l \cdot \hat{e}(n-i) \quad \text{Equation 7}$$

$$x(n) = e(n) + \gamma \cdot e(n-1) \quad \text{Equation 8}$$

where an perceptual weighting filter  $\gamma=0.68$ . The value of the perceptual weighting filter may be a different value according to the design policy of audio encoding.

**[0065]** Then, the impulse response  $h(n) (0 \leq n < L')$  is calculated by the following equations (Step 183 in Fig. 11).

$$\hat{h}(n) = \hat{a}_i^l + \sum_{i=1}^P \hat{a}_i^l \cdot \hat{h}(n-i) \quad \text{Equation 9}$$

$$h(n) = \hat{h}(n) + \gamma \cdot \hat{h}(n-1) \quad \text{Equation 10}$$

**[0066]** The pitch lag calculation unit 153 calculates a pitch lag for each sub-frame by calculating  $k$  that maximizes the following equation (Step 163 in Fig. 9). Note that, in order to reduce the amount of calculations, the above-described target signal calculation (Step 182 in Fig. 11) and the impulse response calculation (Step 183 in Fig. 11) may be omitted, and the residual signal may be used as the target signal.

$$T_p = \operatorname{argmax} T_k$$

$$T_k = \frac{\sum_{n=0}^{L'-1} x(n)y_k(n)}{\sqrt{\sum_{n=0}^{L'-1} y_k(n)y_k(n)}} \quad \text{Equation 11}$$

$$y_k(n) = \sum_{i=0}^n v'(i) \cdot h(n-i) \quad \text{Equation 12}$$

$$v'(n) = \sum_{i=1}^l \text{Int}(i) \cdot u(n + N_{\text{adapt}} - T_p + i) \quad \text{Equation 13}$$

**[0067]** Note that  $y_k(n)$  is obtained by convoluting the impulse response with the linear prediction residual.  $\text{Int}(i)$  indicates an interpolation filter. The details of the interpolation filter are as described in the section 6.8.4.1.4.1 in Non Patent Literature 4. As a matter of course,  $v'(n)=u(n+N_{\text{adapt}}-T_p+i)$  may be employed without using the interpolation filter.

**[0068]** Although the pitch lag can be calculated as an integer by the above-described calculation method, the accuracy of the pitch lag may be increased to after the decimal point accuracy by interpolating the above  $T_k$ . For the details of the procedure to calculate the pitch lag after the decimal point by interpolation, the processing method described in the section 6.8.4.1.4.1 in Non Patent Literature 4 may be used.

**[0069]** The adaptive codebook calculation unit 154 calculates an adaptive codebook vector  $v'(n)$  and a long-term prediction parameter from the pitch lag  $T_p$  and the adaptive codebook  $u(n)$  stored in the adaptive codebook buffer 156 according to the following equation (Step 164 in Fig. 9).

$$v'(n) = \sum_{i=1}^l \text{Int}(i) \cdot u(n + N_{\text{adapt}} - T_p + i) \quad \text{Equation 14}$$

**[0070]** For the details of the procedure to calculate the long-term parameter, the method described in the section 5.7 in Non Patent Literature 3 may be used.

**[0071]** The excitation vector synthesis unit 155 multiplies the adaptive codebook vector  $v'(n)$  by a predetermined adaptive codebook gain

$$g_p^c$$

and outputs an excitation signal vector according to the following equation (Step 165 in Fig. 9).

$$e(n) = g_p^c \cdot v'(n) \quad \text{Equation 15}$$

**[0072]** Although the value of the adaptive codebook gain

$$g_p^c$$

may be 1.0 or the like, for example, a value obtained in advance by learning may be used, or it may be varied by the index representing the characteristics of the frame to be encoded.

**[0073]** Then, the state of the adaptive codebook  $u(n)$  stored in the adaptive codebook buffer 156 is updated by the excitation signal vector according to the following equations (Step 166 in Fig. 9).

$$u(n)=u(n+L) \quad (0 \leq n < N-L) \quad \text{Equation 16}$$

$$u(n+N-L)=e(n) \quad (0 \leq n < L)$$

Equation 17

**[0074]** The synthesis filter 157 synthesizes a decoded signal according to the following equation by linear prediction inverse filtering using the excitation signal vector as an excitation source (Step 167 in Fig. 9).

$$\hat{s}(n) = e(n) - \sum_{i=1}^p \hat{a}_i \cdot \hat{s}(n-i)$$

Equation 18

**[0075]** The above-described Steps 162 to 167 in Fig. 9 are repeated for each sub-frame until the end of the look-ahead signal (Step 168 in Fig. 9).

**[0076]** The pitch lag encoding unit 158 encodes the pitch lag

$$T_p^{(j)} \quad (0 \leq j < M_{1a})$$

that is calculated in the look-ahead signal (Step 169 in Fig. 9). The number of sub-frames contained in the look-ahead signal is  $M_{1a}$ .

**[0077]** Encoding may be performed by a method such as one of the following methods, for example, although any method may be used for encoding.

1. 1. A method that performs binary encoding, scalar quantization, vector quantization or arithmetic encoding on a part or the whole of the pitch lag

$$T_p^{(j)} \quad (0 \leq j < M_{1a})$$

and transmits the result.

2. 2. A method that performs binary encoding, scalar quantization, vector quantization or arithmetic encoding on a part or the whole of a difference

$$T_p^{(j)} - T_p^{(j-1)} \quad (0 < j < M_{1a})$$

from the pitch lag of the previous sub-frame and transmits the result, where

$$T_p^{(-1)}$$

is the pitch lag of the last sub-frame in the frame to be encoded.

3. 3. A method that performs vector quantization or arithmetic encoding on either of a part, or the whole, of the pitch lag

$$T_p^{(j)} \quad (0 \leq j < M_{1a})$$

and a part or the whole of the pitch lag calculated for the frame to be encoded and transmits the result.

4. 4. A method that selects one of a number of predetermined interpolation methods based on a part or the whole of the pitch lag

$$T_p^{(j)} \quad (0 \leq j < M_{1a})$$

and transmits an index indicative of the selected interpolation method. At this time, the pitch lag of a plurality of sub-frames used for audio synthesis in the past also may be used for selection of the interpolation method.

**[0078]** For scalar quantization and vector quantization, a codebook determined empirically or a codebook calculated in advance by learning may be used. Further, a method that performs encoding after adding an offset value to the above pitch lag may also be included in the scope of the embodiment of the present invention as a matter of course.

**<Decoding end>**

**[0079]** As shown in Fig. 5, an example of the audio signal receiving device includes the audio code buffer 121, the audio parameter decoding unit 122, the audio parameter missing processing unit 123, the audio synthesis unit 124, the side information decoding unit 125, and the side information accumulation unit 126. The procedure of the audio signal receiving device is as shown in the example of Fig. 7.

**[0080]** The audio code buffer 121 determines whether a packet is correctly received or not. When the audio code buffer 121 determines that a packet is correctly received, the processing is switched to the audio parameter decoding unit 122 and the side information decoding unit 125. On the other hand, when the audio code buffer 121 determines that a packet is not correctly received, the processing is switched to the audio parameter missing processing unit 123 (Step 141 in Fig. 7).

**<When packet is correctly received>**

**[0081]** The audio parameter decoding unit 122 decodes the received audio code and calculates audio parameters required to synthesize the audio for the frame to be encoded (ISF parameter and corresponding ISF parameter, pitch lag, long-term prediction parameter, adaptive codebook, adaptive codebook gain, fixed codebook gain, fixed codebook vector etc.) (Step 142 in Fig. 7).

**[0082]** The side information decoding unit 125 decodes the side information code, calculates a pitch lag

$$\hat{T}_p^{(j)} (0 \leq j < M_{ls})$$

and stores it in the side information accumulation unit 126. The side information decoding unit 125 decodes the side information code by using the decoding method corresponding to the encoding method used at the encoding end (Step 143 in Fig. 7).

**[0083]** The audio synthesis unit 124 synthesizes the audio signal corresponding to the frame to be encoded based on the parameters output from the audio parameter decoding unit 122 (Step 144 in Fig. 7). The functional configuration example of the audio synthesis unit 124 is shown in Fig. 15, and an example procedure of the audio synthesis unit 124 is shown in Fig. 16. Note that, although the audio parameter missing processing unit 123 is illustrated to show the flow of the signal, the audio parameter missing processing unit 123 is not included in the

functional configuration of the audio synthesis unit 124.

**[0084]** An LP coefficient calculation unit 1121 converts an ISF parameter into an ISP parameter and then performs interpolation processing, and thereby obtains an ISP coefficient for each sub-frame. The LP coefficient calculation unit 1121 then converts the ISP coefficient into a linear prediction coefficient (LP coefficient) and thereby obtains an LP coefficient for each sub-frame (Step 11301 in Fig. 16). For the interpolation of the ISP coefficient and the ISP-LP coefficient, the method described in, for example, section 6.4.5 in Non Patent Literature 4 may be used. The procedure of parameter conversion is not the essential part of the embodiment of the present invention and thus not described in detail.

**[0085]** An adaptive codebook calculation unit 1123 calculates an adaptive codebook vector by using the pitch lag, a long-term prediction parameter and an adaptive codebook 1122 (Step 11302 in Fig. 16). An adaptive codebook vector  $v'(n)$  is calculated from the pitch lag

$$\hat{T}_p^{(j)}$$

and the adaptive codebook  $u(n)$  according to the following equation.

$$v'(n) = \sum_{i=-1}^L \text{Int}(i) \cdot u(n + N_{\text{adapt}} - \hat{T}_p^{(j)} + i) \quad (0 \leq n < L') \quad \text{Equation 19}$$

**[0086]** The adaptive codebook vector is calculated by interpolating the adaptive codebook  $u(n)$  using FIR filter  $\text{Int}(i)$ . The length of the adaptive codebook is  $N_{\text{adapt}}$ . The filter  $\text{Int}(i)$  that is used for the interpolation is the same as the interpolation filter of

$$v'(n) = \sum_{i=-1}^L \text{Int}(i) \cdot u(n + N_{\text{adapt}} - T_p + i). \quad \text{Equation 20}$$

**[0087]** This is the FIR filter with a predetermined length  $2L+1$ .  $L'$  is the number of samples of the sub-frame. It is not necessary to use a filter for the interpolation, whereas at the encoder end a filter is used for the interpolation.

**[0088]** The adaptive codebook calculation unit 1123 carries out filtering on the adaptive codebook vector according to the value of the long-term prediction parameter (Step 11303 in Fig. 16). When the long-term prediction parameter has a value indicating the activation of filtering, filtering is performed on the adaptive codebook vector by the following equation.

$$v''(n) = 0.18v'(n-1) + 0.64v'(n) + 0.18v'(n+1) \quad \text{Equation 21}$$

**[0089]** On the other hand, when the long-term prediction parameter has a value indicating no filtering is needed, filtering is not performed, and  $v(n) = v'(n)$  is established.

**[0090]** An excitation vector synthesis unit 1124 multiplies the adaptive codebook vector by an adaptive codebook gain  $g_p$  (Step 11304 in Fig. 16). Further, the excitation vector synthesis unit 1124 multiplies a fixed codebook vector  $c(n)$  by a fixed codebook gain  $g_c$  (Step 11305 in Fig.

16). Furthermore, the excitation vector synthesis unit 1124 adds the adaptive codebook vector and the fixed codebook vector together and outputs an excitation signal vector (Step 11306 in Fig. 16).

$$e(n) = g_p \cdot v^{\wedge}(n) + g_c \cdot c(n)$$

Equation 22

**[0091]** A post filter 1125 performs post processing such as pitch enhancement, noise enhancement and low-frequency enhancement, for example, on the excitation signal vector. The details of techniques such as pitch enhancement, noise enhancement and low-frequency enhancement are described in the section 6.1 in Non Patent Literature 3. The processing in the post filter is not significantly related to the essential part of the embodiment of the present invention and thus not described in detail (Step 11307 in Fig. 16).

**[0092]** The adaptive codebook 1122 updates the state by an excitation signal vector according to the following equations (Step 11308 in Fig. 16).

$$u(n) = u(n+L) \quad (0 \leq n < N-L)$$

Equation 23

$$u(n+N-L) = e(n) \quad (0 \leq n < L)$$

Equation 24

**[0093]** A synthesis filter 1126 synthesizes a decoded signal according to the following equation by linear prediction inverse filtering using the excitation signal vector as an excitation source (Step 11309 in Fig. 16).

$$\hat{s}(n) = e(n) - \sum_{i=1}^P \hat{a}(i) \cdot \hat{s}(n-i)$$

Equation 25

**[0094]** An perceptual weighting inverse filter 1127 applies an perceptual weighting inverse filter to the decoded signal according to the following equation (Step 11310 in Fig. 16).

$$\hat{s}(n) = \hat{s}(n) + \beta \cdot \hat{s}(n-1)$$

Equation 26

**[0095]** The value of  $\beta$  is typically 0.68 or the like, though not limited to this value.

**[0096]** The audio parameter missing processing unit 123 stores the audio parameters (ISF parameter, pitch lag, adaptive codebook gain, fixed codebook gain) used in the audio synthesis unit 124 into the buffer (Step 145 in Fig. 7).

**<When packet loss is detected>**

**[0097]** The audio parameter missing processing unit 123 reads a pitch lag

$$\hat{T}_p^{(j)} \quad (0 \leq j < M_a)$$

from the side information accumulation unit 126 and predicts audio parameters. The functional configuration example of the audio parameter missing processing unit 123 is shown in the

example of Fig. 12, and an example procedure of audio parameter prediction is shown in Fig. 13.

**[0098]** An ISF prediction unit 191 calculates an ISF parameter using the ISF parameter for the previous frame and the ISF parameter calculated for the past several frames (Step 1101 in Fig. 13). The procedure of the ISF prediction unit 191 is shown in Fig. 10.

**[0099]** First, the buffer is updated using the ISF parameter of the immediately previous frame (Step 171 in Fig. 10). Next, the ISF parameter  $\omega_j$  is calculated according to the following equation (Step 172 in Fig. 10).

$$\dot{\omega}_i = \alpha \omega_i^{(-1)} + (1 - \alpha) \bar{\omega}_i \quad \text{Equation 27}$$

$$\bar{\omega}_i = \beta \omega_i^C + (1 - \beta) \frac{\omega_i^{(-3)} + \omega_i^{(-2)} + \omega_i^{(-1)}}{3} \quad \text{Equation 28}$$

where

$$\omega_i^{(-j)}$$

is the ISF parameter, stored in the buffer, which is for the frame preceding by j-number of frames. Further,

$$\omega_i^C$$

,  $\alpha$  and  $\beta$  are the same values as those used at the encoding end.

**[0100]** In addition, the values of i are arranged so that  $\omega_j$  satisfies  $0 < \omega_0 < \omega_1 < \dots < \omega_{14}$ , and values of  $\omega_j$  are adjusted so that the adjacent  $\omega_j$  is not too close. As an example procedure to adjust the value of  $\omega_j$ , Non Patent Literature 4 (Equation 151) may be used (Step 173 in Fig. 10).

**[0101]** A pitch lag prediction unit 192 decodes the side information code from the side information accumulation unit 126 and thereby obtains a pitch lag

$$\hat{T}_p^{(i)} (0 \leq i < M_{la})$$

. Further, by using a pitch lag

$$\hat{T}_p^{(-j)} (0 \leq j < J)$$

used for the past decoding, the pitch lag prediction unit 192 outputs a pitch lag

$$\hat{T}_p^{(i)} (M_{la} \leq i < M)$$

. The number of sub-frames contained in one frame is M, and the number of pitch lags contained in the side information is  $M_{1a}$ . For the prediction of the pitch lag

$$\hat{T}_p^{(i)} (M_{la} \leq i < M)$$

, the procedure described in, for example, section 7.11.1.3 in Non Patent Literature 4 may be used (Step 1102 in Fig. 13).

**[0102]** An adaptive codebook gain prediction unit 193 outputs an adaptive codebook gain

$$g_p^{(i)} (M_{la} \leq i < M)$$

by using a predetermined adaptive codebook gain

$$g_p^C$$

and an adaptive codebook gain

$$g_p^{(j)} (0 \leq j < J)$$

used in the past decoding. The number of sub-frames contained in one frame is M, and the

number of pitch lags contained in the side information is  $M_{1a}$ . For the prediction of the adaptive codebook gain

$$g_p^{(i)} (M_{1a} \leq i < M)$$

, the procedure described in, for example, section 7.11.2.5.3 in Non Patent Literature 4 may be used (Step 1103 in Fig. 13).

**[0103]** A fixed codebook gain prediction unit 194 outputs a fixed codebook gain

$$g_c^{(i)} (0 \leq i < M)$$

by using a fixed codebook gain

$$g_c^{(j)} (0 \leq j < J)$$

used in the past decoding. The number of sub-frames contained in one frame is  $M$ . For the prediction of the fixed codebook gain

$$g_c^{(i)} (0 \leq i < M)$$

, the procedure described in the section 7.11.2.6 in Non Patent Literature 4 may be used, for example (Step 1104 in Fig. 13).

**[0104]** A noise signal generation unit 195 outputs a noise vector, such as a white noise, with a length of  $L$  (Step 1105 in Fig. 13). The length of one frame is  $L$ .

**[0105]** The audio synthesis unit 124 synthesizes a decoded signal based on the audio parameters output from the audio parameter missing processing unit 123 (Step 144 in Fig. 7). The operation of the audio synthesis unit 124 is the same as the operation of the audio synthesis unit <When audio packet is correctly received> and not redundantly described in detail (Step 144 in Fig. 7).

**[0106]** The audio parameter missing processing unit 123 stores the audio parameters (ISF parameter, pitch lag, adaptive codebook gain, fixed codebook gain) used in the audio synthesis unit 124 into the buffer (Step 145 in Fig. 7).

**[0107]** Although the case of encoding and transmitting the side information for all sub-frames contained in the look-ahead signal is described in the above example, the configuration that transmits only the side information for a specific sub-frame may be employed.

#### **[Alternative example 1-1]**

**[0108]** As an alternative example of the previously discussed example 1, an example that adds a pitch gain to the side information is described hereinafter. A difference between the alternative example 1-1 and the example 1 is only the operation of the excitation vector synthesis unit 155, and therefore description of the other parts is omitted.

**<Encoding end>**

**[0109]** The procedure of the excitation vector synthesis unit 155 is shown in the example of Fig. 14.

**[0110]** An adaptive codebook gain

$$g_p^c$$

is calculated from the adaptive codebook vector  $v'(n)$  and the target signal  $x(n)$  according to the following equation (Step 1111 in Fig. 14).

$$g_p = \frac{\sum_{n=0}^{L'-1} x(n)y(n)}{\sum_{n=0}^{L'-1} y(n)y(n)}, \text{ bounded by } 0 \leq g_p \leq 1.2, \quad \text{Equation 29}$$

where  $y(n)$  is a signal  $y(n)=v(n)*h(n)$  that is obtained by convoluting the impulse response with the adaptive codebook vector.

**[0111]** The calculated adaptive codebook gain is encoded and contained in the side information code (Step 1112 in Fig. 14). For the encoding, scalar quantization using a codebook obtained in advance by learning may be used, although any other technique may be used for the encoding.

**[0112]** By multiplying the adaptive codebook vector by an adaptive codebook gain  $\hat{g}_p$  obtained by decoding the code calculated in the encoding of the adaptive codebook gain, an excitation vector is calculated according to the following equation (Step 1113 in Fig. 14).

$$e(n) = \hat{g}_p \cdot v'(n) \quad \text{Equation 30}$$

**<Decoding end>**

**[0113]** The excitation vector synthesis unit 155 multiplies the adaptive codebook vector  $v'(n)$  by an adaptive codebook gain  $\hat{g}_p$  obtained by decoding the side information code and outputs an excitation signal vector according to the following equation (Step 165 in Fig. 9).

$$e(n) = \hat{g}_p \cdot v'(n) \quad \text{Equation 31}$$

**[Alternative example 1-2]**

**[0114]** As an alternative example of the example 1, an example that adds a flag for determination of use of the side information to the side information is described hereinafter.

**<Encoding end>**

**[0115]** The functional configuration example of the side information encoding unit is shown in Fig. 17, and the procedure of the side information encoding unit is shown in the example of

Fig. 18. A difference from the example 1 is only a side information output determination unit 1128 (Step 1131 in Fig. 18), and therefore description of the other parts is omitted.

**[0116]** The side information output determination unit 1128 calculates segmental SNR of the decoded signal and the look-ahead signal according to the following equation, and only when segmental SNR exceeds a threshold, sets the value of the flag to ON and adds it to the side information.

$$segSNR = \frac{\sum_{n=0}^{L'-1} \hat{s}^2(n)}{\sum_{n=0}^{L'-1} (s(n) - \hat{s}(n))^2} \quad \text{Equation 32}$$

**[0117]** On the other hand, when segmental SNR does not exceed a threshold, the side information output determination unit 1128 sets the value of the flag to OFF and adds it to the side information (Step 1131 in Fig. 18). Note that, the amount of bits of the side information may be reduced by adding the side information such as a pitch lag and a pitch gain to the flag and transmitting the added side information only when the value of the flag is ON, and transmitting only the value of the flag when the value of the flag is OFF.

<Decoding end>

**[0118]** The side information decoding unit decodes the flag contained in the side information code. When the value of the flag is ON, the audio parameter missing processing unit calculates a decoded signal by the same procedure as in the example 1. On the other hand, when the value of the flag is OFF, it calculates a decoded signal by the packet loss concealment technique without using side information (Step 1151 in Fig. 19).

**[Example 2]**

**[0119]** In this example, the decoded audio of the look-ahead signal part is also used when a packet is correctly received. For purposes of this discussion, the number of sub-frames contained in one frame is M sub-frames, and the length of the look-ahead signal is M' sub-frame(s).

<Encoding end>

**[0120]** As shown in the example of Fig. 20, the audio signal transmitting device includes a main encoding unit 211, a side information encoding unit 212, a concealment signal accumulation unit 213, and an error signal encoding unit 214. The procedure of the audio signal transmitting device is shown in Fig. 22.

**[0121]** The error signal encoding unit 214 reads a concealment signal for one sub-frame from the concealment signal accumulation unit 213, subtracts it from the audio signal and thereby calculates an error signal (Step 221 in Fig. 22).

**[0122]** The error signal encoding unit 214 encodes the error signal. As a specific example procedure, AVQ described in the section 6.8.4.1.5 in Non Patent Literature 4, can be used. In the encoding of the error signal, local decoding is performed, and a decoded error signal is output (Step 222 in Fig. 22).

**[0123]** By adding the decoded error signal to the concealment signal, a decoded signal for one sub-frame is output (Step 223 in Fig. 22).

**[0124]** The above Steps 221 to 223 are repeated for M' sub-frames until the end of the concealment signal.

**[0125]** An example functional configuration of the main encoding unit 211 is shown in Fig. 21. The main encoding unit 211 includes an ISF encoding unit 2011, a target signal calculation unit 2012, a pitch lag calculation unit 2013, an adaptive codebook calculation unit 2014, a fixed codebook calculation unit 2015, a gain calculation unit 2016, an excitation vector calculation unit 2017, a synthesis filter 2018, and an adaptive codebook buffer 2019.

**[0126]** The ISF encoding unit 2011 obtains an LP coefficient by applying the Levinson-Durbin method to the frame to be encoded and the look-ahead signal. The ISF encoding unit 2011 then converts the LP coefficient into an ISF parameter and encodes the ISF parameter. The ISF encoding unit 2011 then decodes the code and obtains a decoded ISF parameter. Finally, the ISF encoding unit 2011 interpolates the decoded ISF parameter and obtains a decoded LP coefficient for each sub-frame. The procedures of the Levinson-Durbin method and the conversion from the LP coefficient to the ISF parameter are the same as in the example 1. Further, for the encoding of the ISF parameter, the procedure described in, for example, section 6.8.2 in Non Patent Literature 4 can be used. An index obtained by encoding the ISF parameter, the decoded ISF parameter, and the decoded LP coefficient (which is obtained by converting the decoded ISF parameter into the LP coefficient) can be obtained by the ISF encoding unit 2011 (Step 224 in Fig. 22).

**[0127]** The detailed procedure of the target signal calculation unit 2012 is the same as in Step 162 in Fig. 9 in the example 1 (Step 225 in Fig. 22).

**[0128]** The pitch lag calculation unit 2013 refers to the adaptive codebook buffer and calculates a pitch lag and a long-term prediction parameter by using the target signal. The detailed procedure of the calculation of the pitch lag and the long-term prediction parameter is the same as in the example 1 (Step 226 in Fig. 22).

**[0129]** The adaptive codebook calculation unit 2014 calculates an adaptive codebook vector

by using the pitch lag and the long-term prediction parameter calculated by the pitch lag calculation unit 2013. The detailed procedure of the adaptive codebook calculation unit 2014 is the same as in the example 1 (Step 227 in Fig. 22).

**[0130]** The fixed codebook calculation unit 2015 calculates a fixed codebook vector and an index obtained by encoding the fixed codebook vector by using the target signal and the adaptive codebook vector. The detailed procedure is the same as the procedure of AVQ used in the error signal encoding unit 214 (Step 228 in Fig. 22).

**[0131]** The gain calculation unit 2016 calculates an adaptive codebook gain, a fixed codebook gain and an index obtained by encoding these two gains using the target signal, the adaptive codebook vector and the fixed codebook vector. A detailed procedure which can be used is described in, for example, section 6.8.4.1.6 in Non Patent Literature 4 (Step 229 in Fig. 22).

**[0132]** The excitation vector calculation unit 2017 calculates an excitation vector by adding the adaptive codebook vector and the fixed codebook vector to which the gain is applied. The detailed procedure is the same as in example 1. Further, the excitation vector calculation unit 2017 updates the state of the adaptive codebook buffer 2019 by using the excitation vector. The detailed procedure is the same as in the example 1 (Step 2210 in Fig. 22).

**[0133]** The synthesis filter 2018 synthesizes a decoded signal by using the decoded LP coefficient and the excitation vector (Step 2211 in Fig. 22).

**[0134]** The above Steps 224 to 2211 are repeated for M-M' sub-frames until the end of the frame to be encoded.

**[0135]** The side information encoding unit 212 calculates the side information for the look-ahead signal M' sub-frame. A specific procedure is the same as in the example 1 (Step 2212 in Fig. 22).

**[0136]** In addition to the procedure of the example 1, the decoded signal output by the synthesis filter 157 of the side information encoding unit 212 is accumulated in the concealment signal accumulation unit 213 in the example 2 (Step 2213 in Fig. 22).

#### <Decoding unit>

**[0137]** As shown in Fig. 23, an example of the audio signal receiving device includes an audio code buffer 231, an audio parameter decoding unit 232, an audio parameter missing processing unit 233, an audio synthesis unit 234, a side information decoding unit 235, a side information accumulation unit 236, an error signal decoding unit 237, and a concealment signal accumulation unit 238. An example procedure of the audio signal receiving device is shown in Fig. 24. An example functional configuration of the audio synthesis unit 234 is shown in Fig. 25.

**[0138]** The audio code buffer 231 determines whether a packet is correctly received or not. When the audio code buffer 231 determines that a packet is correctly received, the processing is switched to the audio parameter decoding unit 232, the side information decoding unit 235 and the error signal decoding unit 237. On the other hand, when the audio code buffer 231 determines that a packet is not correctly received, the processing is switched to the audio parameter missing processing unit 233 (Step 241 in Fig. 24).

**<When packet is correctly received>**

**[0139]** The error signal decoding unit 237 decodes an error signal code and obtains a decoded error signal. As a specific example procedure, a decoding method corresponding to the method used at the encoding end, such as AVQ described in the section 7.1.2.1.2 in Non Patent Literature 4 can be used (Step 242 in Fig. 24).

**[0140]** A look-ahead excitation vector synthesis unit 2318 reads a concealment signal for one sub-frame from the concealment signal accumulation unit 238 and adds the concealment signal to the decoded error signal, and thereby outputs a decoded signal for one sub-frame (Step 243 in Fig. 24).

**[0141]** The above Steps 241 to 243 are repeated for M' sub-frames until the end of the concealment signal.

**[0142]** The audio parameter decoding unit 232 includes an ISF decoding unit 2211, a pitch lag decoding unit 2212, a gain decoding unit 2213, and a fixed codebook decoding unit 2214. The functional configuration example of the audio parameter decoding unit 232 is shown in Fig. 26.

**[0143]** The ISF decoding unit 2211 decodes the ISF code and converts it into an LP coefficient and thereby obtains a decoded LP coefficient. For example, the procedure described in the section 7.1.1 in Non Patent Literature 4 is used (Step 244 in Fig. 24).

**[0144]** The pitch lag decoding unit 2212 decodes a pitch lag code and obtains a pitch lag and a long-term prediction parameter (Step 245 in Fig. 24).

**[0145]** The gain decoding unit 2213 decodes a gain code and obtains an adaptive codebook gain and a fixed codebook gain. An example detailed procedure is described in the section 7.1.2.1.3 in Non Patent Literature 4 (Step 246 in Fig. 24).

**[0146]** An adaptive codebook calculation unit 2313 calculates an adaptive codebook vector by using the pitch lag and the long-term prediction parameter. The detailed procedure of the adaptive codebook calculation unit 2313 is as described in the example 1 (Step 247 in Fig. 24).

**[0147]** The fixed codebook decoding unit 2214 decodes a fixed codebook code and calculates a fixed codebook vector. The detailed procedure is as described in the section 7.1.2.1.2 in Non

Patent Literature 4 (Step 248 in Fig. 24).

**[0148]** An excitation vector synthesis unit 2314 calculates an excitation vector by adding the adaptive codebook vector and the fixed codebook vector to which the gain is applied. Further, an excitation vector calculation unit updates the adaptive codebook buffer by using the excitation vector (Step 249 in Fig. 24). The detailed procedure is the same as in the example 1.

**[0149]** A synthesis filter 2316 synthesizes a decoded signal by using the decoded LP coefficient and the excitation vector (Step 2410 in Fig. 24). The detailed procedure is the same as in the example 1.

**[0150]** The above Steps 244 to 2410 are repeated for M-M' sub-frames until the end of the frame to be encoded.

**[0151]** The functional configuration of the side information decoding unit 235 is the same as in the example 1. The side information decoding unit 235 decodes the side information code and calculates a pitch lag (Step 2411 in Fig. 24).

**[0152]** The functional configuration of the audio parameter missing processing unit 233 is the same as in the example 1.

**[0153]** The ISF prediction unit 191 predicts an ISF parameter using the ISF parameter for the previous frame and converts the predicted ISF parameter into an LP coefficient. The procedure is the same as in Steps 172, 173 and 174 of the example 1 shown in Fig. 10 (Step 2412 in Fig. 24).

**[0154]** The adaptive codebook calculation unit 2313 calculates an adaptive codebook vector by using the pitch lag output from the side information decoding unit 235 and an adaptive codebook 2312 (Step 2413 in Fig. 24). The procedure is the same as in Steps 11301 and 11302 in Fig. 16.

**[0155]** The adaptive codebook gain prediction unit 193 outputs an adaptive codebook gain. A specific procedure is the same as in Step 1103 in Fig. 13 (Step 2414 in Fig. 24).

**[0156]** The fixed codebook gain prediction unit 194 outputs a fixed codebook gain. A specific procedure is the same as in Step 1104 in Fig. 13 (Step 2415 in Fig. 24).

**[0157]** The noise signal generation unit 195 outputs a noise, such as a white noise as a fixed codebook vector. The procedure is the same as in Step 1105 in Fig. 13 (Step 2416 in Fig. 24).

**[0158]** The excitation vector synthesis unit 2314 applies gain to each of the adaptive codebook vector and the fixed codebook vector and adds them together and thereby calculates an excitation vector. Further, the excitation vector synthesis unit 2314 updates the adaptive

codebook buffer using the excitation vector (Step 2417 in Fig. 24).

**[0159]** The synthesis filter 2316 calculates a decoded signal using the above-described LP coefficient and the excitation vector. The synthesis filter 2316 then updates the concealment signal accumulation unit 238 using the calculated decoded signal (Step 2418 in Fig. 24).

**[0160]** The above steps are repeated for  $M'$  sub-frames, and the decoded signal is output as the audio signal.

**<When a packet is lost>**

**[0161]** A concealment signal for one sub-frame is read from the concealment signal accumulation unit and is used as the decoded signal (Step 2419 in Fig. 24).

**[0162]** The above is repeated for  $M'$  sub-frames.

**[0163]** The ISF prediction unit 191 predicts an ISF parameter (Step 2420 in Fig. 24). As the procedure, Step 1101 in Fig. 13 can be used.

**[0164]** The pitch lag prediction unit 192 outputs a predicted pitch lag by using the pitch lag used in the past decoding (Step 2421 in Fig. 24). The procedure used for the prediction is the same as in Step 1102 in Fig. 13.

**[0165]** The operations of the adaptive codebook gain prediction unit 193, the fixed codebook gain prediction unit 194, the noise signal generation unit 195 and the audio synthesis unit 234 are the same as in the example 1 (Step 2422 in Fig. 24).

**[0166]** The above steps are repeated for  $M$  sub-frames, and the decoded signal for  $M-M'$  sub-frames is output as the audio signal, and the concealment signal accumulation unit 238 is updated by the decoded signal for the remaining  $M'$  sub-frames.

**[Example 3]**

**[0167]** A case of using glottal pulse synchronization in the calculation of an adaptive codebook vector is described hereinafter.

**<Encoding end>**

**[0168]** The functional configuration of the audio signal transmitting device is the same as in example 1. The functional configuration and the procedure are different only in the side

information encoding unit, and therefore only the operation of the side information encoding unit is described below.

**[0169]** The side information encoding unit includes an LP coefficient calculation unit 311, a pitch lag prediction unit 312, a pitch lag selection unit 313, a pitch lag encoding unit 314, and an adaptive codebook buffer 315. The functional configuration of an example of the side information encoding unit is shown in Fig. 27, and an example procedure of the side information encoding unit is shown in the example of Fig. 28.

**[0170]** The LP coefficient calculation unit 311 is the same as the LP coefficient calculation unit in example 1 and thus will not be redundantly described (Step 321 in Fig. 28).

**[0171]** The pitch lag prediction unit 312 calculates a pitch lag predicted value  $\hat{T}_p$ , using the pitch lag obtained from the audio encoding unit (Step 322 in Fig. 28). The specific processing of the prediction is the same as the prediction of the pitch lag

$$\hat{T}_p^{(i)} (M_{la} \leq i < M)$$

in the pitch lag prediction unit 192 in the example 1 (which is the same as in Step 1102 in Fig. 13).

**[0172]** Then, the pitch lag selection unit 313 determines a pitch lag to be transmitted as the side information (Step 323 in Fig. 28). The detailed procedure of the pitch lag selection unit 313 is shown in the example of Fig. 29.

**[0173]** First, a pitch lag codebook is generated from the pitch lag predicted value  $\hat{T}_p$  and the value of the past pitch lag

$$\hat{T}_p^{(j)} (0 \leq j < J)$$

according to the following equations (Step 331 in Fig. 29).

<When

$$\hat{T}_p - \hat{T}_p^{(-1)} \geq 0$$

>

**[0174]**

$$\hat{T}_C^j = \begin{cases} \hat{T}_p & (j = 0) \\ \hat{T}_p^{(-1)} - j \cdot \delta_j + \rho & (0 < j < I) \end{cases}$$

Equation 33

<When

$$\hat{T}_p - \hat{T}_p^{(-1)} < 0$$

>

**[0175]**

$$\hat{T}_C^j = \begin{cases} \hat{T}_p & (j = 0) \\ \hat{T}_p^{(-1)} + j \cdot \delta_j + \rho & (0 < j < I) \end{cases}$$

Equation 34

**[0176]** The value of the pitch lag for one sub-frame before is

$$\hat{T}_p^{(-1)}$$

. Further, the number of indexes of the codebook is  $l$ .  $\delta_j$  is a predetermined step width, and  $\rho$  is a predetermined constant.

**[0177]** Then, by using the adaptive codebook and the pitch lag predicted value  $\hat{T}_p$ , an initial excitation vector  $u_0(n)$  is generated according to the following equation (Step 332 in Fig. 29).

$$u_0(n) = \begin{cases} 0.18u_0(n - \hat{T}_p - 1) + 0.64u_0(n - \hat{T}_p) + 0.18u_0(n - \hat{T}_p + 1) & (0 \leq n < \hat{T}_p) \\ u_0(n - \hat{T}_p) & (\hat{T}_p \leq n < L) \end{cases}$$

Equation 35

**[0178]** The procedure of calculating the initial excitation vector is the same as the equations (607) and (608) in Non Patent Literature 4.

**[0179]** Then, glottal pulse synchronization is applied to the initial excitation vector by using all candidate pitch lags

$$\hat{T}_c^j (0 \leq j < J)$$

in the pitch lag codebook to thereby generate a candidate adaptive codebook vector  $u^j(n)$  ( $0 \leq j < l$ ) (Step 333 in Fig. 29). For the glottal pulse synchronization, the same procedure can be used as in the case described in section 7.11.2.5 in Non Patent Literature 4 where a pulse position is not available. Note, however, that  $u(n)$  in Non Patent Literature 4 corresponds to  $u_0(n)$  in the embodiment of the present invention, and extrapolated pitch corresponds to

$$\hat{T}_c^j$$

in the embodiment of the present invention, and the last reliable pitch ( $T_c$ ) corresponds to

$$\hat{T}_p^{(-1)}$$

in the embodiment of the present invention.

**[0180]** For the candidate adaptive codebook vector  $u^j(n)$  ( $0 \leq j < l$ ), a rate scale is calculated (Step 334 in Fig. 29). In the case of using segmental SNR as the rate scale, a signal is synthesized by inverse filtering using the LP coefficient, and segmental SNR is calculated with the input signal according to the following equation.

$$\hat{s}_j(n) = u^j(n) - \sum_{i=1}^p \hat{a}(i) \cdot \hat{s}_j(n-i) \quad \text{Equation 35}$$

$$segSNR_j = \frac{\sum_{n=0}^{L-1} \hat{s}_j^2(n)}{\sum_{n=0}^{L-1} (s(n) - \hat{s}_j(n))^2} \quad \text{Equation 36}$$

[0181] Instead of performing inverse filtering, segmental SNR may be calculated in the region of the adaptive codebook vector by using a residual signal according to the following equation.

$$r(n) = s(n) + \sum_{i=1}^P \hat{a}(i) \cdot s(n-i) \quad \text{Equation 37}$$

$$segSNR_j = \frac{\sum_{n=0}^{L-1} u^j(n)}{\sum_{n=0}^{L-1} (r(n) - u^j(n))^2} \quad \text{Equation 38}$$

[0182] In this case, a residual signal  $r(n)$  of the look-ahead signal  $s(n)$  ( $0 \leq n < L'$ ) is calculated by using the LP coefficient (Step 181 in Fig. 11).

[0183] An index corresponding to the largest rate scale calculated in Step 334 is selected, and a pitch lag corresponding to the index is calculated (Step 335 in Fig. 29).

$$\arg \max_j [segSNR_j] \quad \text{Equation 39}$$

<Decoding end>

[0184] The functional configuration of the audio signal receiving device is the same as in the example 1. Differences from the example 1 are the functional configuration and the procedure of the audio parameter missing processing unit 123, the side information decoding unit 125 and the side information accumulation unit 126, and only those are described hereinbelow.

<When packet is correctly received>

[0185] The side information decoding unit 125 decodes the side information code and calculates a pitch lag

$\hat{T}_C^{idx}$   
and stores it into the side information accumulation unit 126. The example procedure of the side information decoding unit 125 is shown in Fig. 30.

[0186] In the calculation of the pitch lag, the pitch lag prediction unit 312 first calculates a pitch lag predicted value  $\hat{T}_p$ , by using the pitch lag obtained from the audio decoding unit (Step 341 in Fig. 30). The specific processing of the prediction is the same as in Step 322 of Fig. 28 in the example 3.

[0187] Then, a pitch lag codebook is generated from the pitch lag predicted value  $\hat{T}_p$ , and the value of the past pitch lag

$$\hat{T}_C^{(j)} (0 \leq j < J)$$

, according to the following equations (Step 342 in Fig. 30).

<When

$$\hat{T}_p - \hat{T}_p^{(-1)} \geq 0$$

>

[0188]

$$\hat{T}_C^j = \begin{cases} \hat{T}_p & (j = 0) \\ \hat{T}_p^{(-1)} - j \cdot \delta_j + \rho & (0 < j < I) \end{cases} \quad \text{Equation 40}$$

<When

$$\hat{T}_p - \hat{T}_p^{(-1)} < 0$$

>

[0189]

$$\hat{T}_C^j = \begin{cases} \hat{T}_p & (j = 0) \\ \hat{T}_p^{(-1)} + j \cdot \delta_j + \rho & (0 < j < I) \end{cases} \quad \text{Equation 41}$$

[0190] The procedure is the same as in Step 331 in Fig. 29. The value of the pitch lag for one sub-frame before is

$$\hat{T}_p^{(-1)}$$

. Further, the number of indexes of the codebook is I.  $\delta_j$  is a predetermined step width, and  $\rho$  is a predetermined constant.

[0191] Then, by referring to the pitch lag codebook, a pitch lag

$$\hat{T}_C^{idx}$$

corresponding to the index idx transmitted as part of the side information is calculated and stored in the side information accumulation unit 126 (Step 343 in Fig. 30).

<When packet loss is detected>

[0192] Although the functional configuration of the audio synthesis unit is also the same as in the example 1 (which is the same as in Fig. 15), only the adaptive codebook calculation unit 1123 that operates differently from that in the example 1 is described hereinbelow.

[0193] The audio parameter missing processing unit 123 reads the pitch lag from the side information accumulation unit 126 and calculates a pitch lag predicted value according to the following equation, and uses the calculated pitch lag predicted value instead of the output of the pitch lag prediction unit 192.

$$\hat{T}_p = \hat{T}_p^{(-1)} + \kappa \cdot (\hat{T}_C^{idx} - \hat{T}_p^{(-1)}) \quad \text{Equation 42}$$

where  $\kappa$  is a predetermined constant.

**[0194]** Then, by using the adaptive codebook and the pitch lag predicted value

$$\hat{T}_p$$

, an initial excitation vector  $u_0(n)$  is generated according to the following equation (Step 332 in Fig. 29).

$$u_0(n) = \begin{cases} 0.18u_0(n - \hat{T}_p^{(-1)} - 1) + 0.64u_0(n - \hat{T}_p^{(-1)}) + 0.18u_0(n - \hat{T}_p^{(-1)} + 1) & (0 \leq n < \hat{T}_p^{(-1)}) \\ u_0(n - \hat{T}_p^{(-1)}) & (\hat{T}_p^{(-1)} \leq n < L) \end{cases}$$

Equation 43

**[0195]** Then, glottal pulse synchronization is applied to the initial excitation vector by using the pitch lag

$$\hat{T}_C^{idx}$$

to thereby generate an adaptive codebook vector  $u(n)$ . For the glottal pulse synchronization, the same procedure as in Step 333 of Fig. 29 is used.

**[0196]** Hereinafter, an audio encoding program 70 that causes a computer to execute the above-described processing by the audio signal transmitting device is described. As shown in Fig. 31, the audio encoding program 70 is stored in a program storage area 61 formed in a recording medium 60 that is inserted into a computer and accessed, or included in a computer.

**[0197]** The audio encoding program 70 includes an audio encoding module 700 and a side information encoding module 701. The functions implemented by executing the audio encoding module 700 and the side information encoding module 701 are the same as the functions of the audio encoding unit 111 and the side information encoding unit 112 in the audio signal transmitting device described above, respectively.

**[0198]** Note that a part or the whole of the audio encoding program 70 may be transmitted through a transmission medium such as a communication line, received and stored (including being installed) by another device. Further, each module of the audio encoding program 70 may be installed not in one computer but in any of a plurality of computers. In this case, the above-described processing of the audio encoding program 70 is performed by a computer system composed of the plurality of computers.

**[0199]** Hereinafter, an audio decoding program 90 that causes a computer to execute the above-described processing by the audio signal receiving device is described. As shown in Fig. 32, the audio decoding program 90 is stored in a program storage area 81 formed in a recording medium 80 that is inserted into a computer and accessed, or included in a computer.

**[0200]** The audio decoding program 90 includes an audio code buffer module 900, an audio parameter decoding module 901, a side information decoding module 902, a side information accumulation module 903, an audio parameter missing processing module 904, and an audio

synthesis module 905. The functions implemented by executing the audio code buffer module 900, the audio parameter decoding module 901, the side information decoding module 902, the side information accumulation module 903, an audio parameter missing processing module 904 and the audio synthesis module 905 are the same as the function of the audio code buffer 231, the audio parameter decoding unit 232, the side information decoding unit 235, the side information accumulation unit 236, the audio parameter missing processing unit 233 and the audio synthesis unit 234 described above, respectively.

**[0201]** Note that a part or the whole of the audio decoding program 90 may be transmitted through a transmission medium such as a communication line, received and stored (including being installed) by another device. Further, each module of the audio decoding program 90 may be installed not in one computer but in any of a plurality of computers. In this case, the above-described processing of the audio decoding program 90 is performed by a computer system composed of the plurality of computers.

**[Example 4]**

**[0202]** An example that uses side information for pitch lag prediction at the decoding end is described hereinafter.

**<Encoding end>**

**[0203]** The functional configuration of the audio signal transmitting device is the same as in the example 1. The functional configuration and the procedure are different only in the side information encoding unit 112, and therefore the operation of the side information encoding unit 112 only is described hereinbelow.

**[0204]** The functional configuration of an example of the side information encoding unit 112 is shown in Fig. 33, and an example procedure of the side information encoding unit 112 is shown in Fig. 34. The side information encoding unit 112 includes an LP coefficient calculation unit 511, a residual signal calculation unit 512, a pitch lag calculation unit 513, an adaptive codebook calculation unit 514, an adaptive codebook buffer 515, and a pitch lag encoding unit 516.

**[0205]** The LP coefficient calculation unit 511 is the same as the LP coefficient calculation unit 151 in example 1 shown in Fig. 8 and thus is not redundantly described.

**[0206]** The residual signal calculation unit 512 calculates a residual signal by the same processing as in Step 181 in example 1 shown in Fig. 11.

**[0207]** The pitch lag calculation unit 513 calculates a pitch lag for each sub-frame by

calculating  $k$  that maximizes the following equation (Step 163 in Fig. 34). Note that  $u(n)$  indicates the adaptive codebook, and  $L'$  indicates the number of samples contained in one sub-frame.

$$T_p = \arg_k \max T_k$$

$$T_k = \frac{\sum_{n=0}^{L'-1} r(n)u(n-k)}{\sqrt{\sum_{n=0}^{L'-1} u(n-k)u(n-k)}} \quad \text{Equation 43}$$

**[0208]** The adaptive codebook calculation unit 514 calculates an adaptive codebook vector  $v'(n)$  from the pitch lag  $T_p$  and the adaptive codebook  $u(n)$ . The length of the adaptive codebook is  $N_{\text{adapt}}$  (Step 164 in Fig. 34).

$$v'(n) = u(n + N_{\text{adapt}} - T_p) \quad \text{Equation 44}$$

**[0209]** The adaptive codebook buffer 515 updates the state by the adaptive codebook vector  $v'(n)$  (Step 166 in Fig. 34).

$$u(n) = u(n + L') \quad (0 \leq n < N - L') \quad \text{Equation 45}$$

$$u(n + N - L') = v'(n) \quad (0 \leq n < L') \quad \text{Equation 46}$$

**[0210]** The pitch lag encoding unit 516 is the same as that in example 1 and thus not redundantly described (Step 169 in Fig. 34).

<Decoding end>

**[0211]** The audio signal receiving device includes the audio code buffer 121, the audio parameter decoding unit 122, the audio parameter missing processing unit 123, the audio synthesis unit 124, the side information decoding unit 125, and the side information accumulation unit 126, just like in example 1. The procedure of the audio signal receiving device is as shown in Fig. 7.

**[0212]** The operation of the audio code buffer 121 is the same as in example 1.

<When packet is correctly received>

**[0213]** The operation of the audio parameter decoding unit 122 is the same as in the example 1.

**[0214]** The side information decoding unit 125 decodes the side information code, calculates a

pitch lag

$$\hat{T}_p^{(j)} (0 \leq j < M_{la})$$

and stores it into the side information accumulation unit 126. The side information decoding unit 125 decodes the side information code by using the decoding method corresponding to the encoding method used at the encoding end.

**[0215]** The audio synthesis unit 124 is the same as that of example 1.

**<When packet loss is detected>**

**[0216]** The ISF prediction unit 191 of the audio parameter missing processing unit 123 (see Fig. 12) calculates an ISF parameter the same way as in the example 1.

**[0217]** An example procedure of the pitch lag prediction unit 192 is shown in Fig. 35. The pitch lag prediction unit 192 reads the side information code from the side information accumulation unit 126 and obtains a pitch lag

$$\hat{T}_p^{(i)} (0 \leq i < M_{la})$$

in the same manner as in example 1 (Step 4051 in Fig. 35). Further, the pitch lag prediction unit 192 outputs the pitch lag

$$\hat{T}_p^{(i)} (M_{la} \leq i < M)$$

by using the pitch lag

$$\hat{T}_p^{(j)} (0 \leq j < J)$$

used in the past decoding (Step 4052 in Fig. 35). The number of sub-frames contained in one frame is M, and the number of pitch lags contained in the side information is  $M_{1a}$ . In the prediction of the pitch lag

$$\hat{T}_p^{(i)} (M_{la} \leq i < M)$$

, the procedure as described in Non Patent Literature 4 can be used (Step 1102 in Fig. 13).

**[0218]** In the prediction of the pitch lag

$$\hat{T}_p^{(i)} (M_{la} \leq i < M)$$

, the pitch lag prediction unit 192 may predict the pitch lag

$$\hat{T}_p^{(i)} (M_{la} \leq i < M)$$

by using the pitch lag

$$\hat{T}_p^{(j)} (1 \leq j < J)$$

used in the past decoding and the pitch lag

$$\hat{T}_p^{(i)} (0 \leq i < M_{la})$$

. Further,

$$\hat{T}_p^{(i)} = \hat{T}_p^{(M_{la})}$$

may be established. The procedure of the pitch lag prediction unit in this case is as shown in Fig. 36.

**[0219]** Further, the pitch lag prediction unit 192 may establish

$$\hat{T}_n^{(i)} = \hat{T}_n^{(M_{la})}$$

only when the reliability of the pitch lag predicted value is low. The procedure of the pitch lag prediction unit in this case is shown in Fig. 37. Instruction information as to whether the predicated value is used, or the pitch lag

$$\hat{T}_p^{(M_0)}$$

obtained by the side information is used may be input to the adaptive codebook calculation unit 154.

**[0220]** The adaptive codebook gain prediction unit 193 and the fixed codebook gain prediction unit 194 are the same as those of the example 1.

**[0221]** The noise signal generation unit 195 is the same as that of the example 1.

**[0222]** The audio synthesis unit 124 synthesizes, from the parameters output from the audio parameter missing processing unit 123, an audio signal corresponding to the frame to be encoded.

**[0223]** The LP coefficient calculation unit 1121 of the audio synthesis unit 124 (see Fig. 15) obtains an LP coefficient in the same manner as in example 1 (Step S11301 in Fig. 16).

**[0224]** The adaptive codebook calculation unit 1123 calculates an adaptive codebook vector in the same manner as in example 1. The adaptive codebook calculation unit 1123 may perform filtering on the adaptive codebook vector or may not perform filtering. Specifically, the adaptive codebook vector is calculated using the following equation. The filtering coefficient is  $f_j$ .

$$v(n) = f_{-1}v'(n-1) + f_0v'(n) + f_1v'(n+1) \quad \text{Equation 47}$$

**[0225]** In the case of decoding a value that does not indicate filtering,  $v(n) = v'(n)$  is established (adaptive codebook calculation step A).

**[0226]** The adaptive codebook calculation unit 1123 may calculate an adaptive codebook vector in the following procedure (adaptive codebook calculation step B).

**[0227]** An initial adaptive codebook vector is calculated using the pitch lag and the adaptive codebook 1122.

$$v(n) = f_{-1}v'(n-1) + f_0v'(n) + f_1v'(n+1) \quad \text{Equation 48}$$

$v(n) = v'(n)$  may be established according to a design policy.

**[0228]** Then, glottal pulse synchronization is applied to the initial adaptive codebook vector. For the glottal pulse synchronization, the same procedure as in the case where a pulse position is not available in the section 7.11.2.5 in Non Patent Literature 4 is used. Note that, however,  $u(n)$  in Non Patent Literature 4 corresponds to  $v(n)$  in the embodiment of the present invention, and extrapolated pitch corresponds to

$$\hat{T}_p^{(M-1)}$$

in the embodiment of the present invention, and the last reliable pitch ( $T_c$ ) corresponds to

$$\hat{T}_p^{(M_b-1)}$$

in the embodiment of the present invention.

**[0229]** Further, in the case where the pitch lag prediction unit 192 outputs the above-described instruction information for the predicated value, when the instruction information indicates that the pitch lag transmitted as the side information should not be used as the predicated value (NO in Step 4082 in Fig. 38), the adaptive codebook calculation unit 1123 may use the above-described adaptive codebook calculation step A, and if it is indicated that the pitch value should be used (YES in Step 4082 in Fig. 38), the adaptive codebook calculation unit 1123 may use the above-described adaptive codebook calculation step B. The procedure of the adaptive codebook calculation unit 1123 in this case is shown in the example of Fig. 38.

**[0230]** The excitation vector synthesis unit 1124 outputs an excitation vector in the same manner as in example 1 (Step 11306 in Fig. 16).

**[0231]** The post filter 1125 performs post processing on the synthesis signal in the same manner as in the example 1.

**[0232]** The adaptive codebook 1122 updates the state by using the excitation signal vector in the same manner as in the example 1 (Step 11308 in Fig. 16).

**[0233]** The synthesis filter 1126 synthesizes a decoded signal in the same manner as in the example 1 (Step 11309 in Fig. 16).

**[0234]** The perceptual weighting inverse filter 1127 applies a perceptual weighting inverse filter in the same manner as in the example 1.

**[0235]** The audio parameter missing processing unit 123 stores the audio parameters (ISF parameter, pitch lag, adaptive codebook gain, fixed codebook gain) used in the audio synthesis unit 124 into the buffer in the same manner as in the example 1 (Step 145 in Fig. 7).

#### **[Example 5]**

**[0236]** In this embodiment, a configuration is described in which a pitch lag is transmitted as side information only in a specific frame class, and otherwise a pitch lag is not transmitted.

**<Transmitting end>**

**[0237]** In the audio signal transmitting device, an input audio signal is sent to the audio encoding unit 111.

**[0238]** The audio encoding unit 111 in this example calculates an index representing the characteristics of a frame to be encoded and transmits the index to the side information encoding unit 112. The other operations are the same as in example 1.

**[0239]** In the side information encoding unit 112, a difference from the examples 1 to 4 is only with regard to the pitch lag encoding unit 158, and therefore the operation of the pitch lag encoding unit 158 is described hereinbelow. The configuration of the side information encoding unit 112 in the example 5 is shown in Fig. 39.

**[0240]** The procedure of the pitch lag encoding unit 158 is shown in the example of Fig. 40. The pitch lag encoding unit 158 reads the index representing the characteristics of the frame to be encoded (Step 5021 in Fig. 40) and, when the index representing the characteristics of the frame to be encoded is equal to a predetermined value, the pitch lag encoding unit 158 determines the number of bits to be assigned to the side information as B bits ( $B > 1$ ). On the other hand, when the index representing the characteristics of the frame to be encoded is different from a predetermined value, the pitch lag encoding unit 158 determines the number of bits to be assigned to the side information as 1 bit (Step 5022 in Fig. 40).

**[0241]** When the number of bits to be assigned to the side information is 1 bit (No in Step 5022 in Fig. 40), a value indicating non-transmission of the side information, is used as the side information code, and is set to the side information index (Step 5023 in Fig. 40).

**[0242]** On the other hand, when the number of bits to be assigned to the side information is B bits (Yes in Step 5022 in Fig. 40), a value indicating transmission of the side information is set to the side information index (Step 5024 in Fig. 40), and further, a code of B-1 bits obtained by encoding the pitch lag by the method described in example 1 is added, for use as the side information code (Step 5025 in Fig. 40).

**<Decoding end>**

**[0243]** The audio signal receiving device includes the audio code buffer 121, the audio parameter decoding unit 122, the audio parameter missing processing unit 123, the audio synthesis unit 124, the side information decoding unit 125, and the side information accumulation unit 126, just like in example 1. The procedure of the audio signal receiving device is as shown in Fig. 7.

**[0244]** The operation of the audio code buffer 121 is the same as in example 1.

**<When packet is correctly received>**

**[0245]** The operation of the audio parameter decoding unit 122 is the same as in example 1.

**[0246]** The procedure of the side information decoding unit 125 is shown in the example of Fig. 41. The side information decoding unit 125 decodes the side information index contained in the side information code first (Step 5031 in Fig. 41). When the side information index indicates non-transmission of the side information, the side information decoding unit 125 does not perform any further decoding operations. Also, the side information decoding unit 125 stores the value of the side information index in the side information accumulation unit 126 (Step 5032 in Fig. 41).

**[0247]** On the other hand, when the side information index indicates transmission of the side information, the side information decoding unit 125 further performs decoding of B-1 bits and calculates a pitch lag

$$\hat{T}_p^{(j)} (0 \leq j < M_{ls})$$

and stores the calculated pitch lag in the side information accumulation unit 126 (Step 5033 in Fig. 41). Further, the side information decoding unit 125 stores the value of the side information index into the side information accumulation unit 126. Note that the decoding of the side information of B-1 bits is the same operation as the side information decoding unit 125 in example 1.

**[0248]** The audio synthesis unit 124 is the same as that of example 1.

**<When packet loss is detected>**

**[0249]** The ISF prediction unit 191 of the audio parameter missing processing unit 123 (see Fig. 12) calculates an ISF parameter the same way as in example 1.

**[0250]** The procedure of the pitch lag prediction unit 192 is shown in the example of Fig. 42. The pitch lag prediction unit 192 reads the side information index from the side information accumulation unit 126 (Step 5041 in Fig. 42) and checks whether it is the value indicating transmission of the side information (Step 5042 in Fig. 42).

**<When the side information index is a value indicating transmission of side information >**

**[0251]** In the same manner as in example 1, the side information code is read from the side information accumulation unit 126 to obtain a pitch lag

$$\hat{T}_p^{(i)} (0 \leq i < M_{ls})$$

(Step 5043 in Fig. 42). Further, the pitch lag

$$\hat{T}_p^{(i)} (M_{ls} \leq i < M)$$

is output by using the pitch lag

$$\hat{T}_p^{(-j)} (0 \leq j < J)$$

used in the past decoding and

$$\hat{T}_p^{(i)} (0 \leq i < M_{la})$$

obtained as the side information (Step 5044 in Fig. 42). The number of sub-frames contained in one frame is  $M$ , and the number of pitch lags contained in the side information is  $M_{la}$ . In the prediction of the pitch lag

$$\hat{T}_p^{(i)} (M_{la} \leq i < M)$$

, the procedure as described in Non Patent Literature 4 can be used (Step 1102 in Fig. 13). Further,

$$\hat{T}_p^{(i)} = \hat{T}_p^{(M_{la})}$$

may be established.

**[0252]** Further, the pitch lag prediction unit 192 may establish

$$\hat{T}_p^{(i)} = \hat{T}_p^{(M_{la})}$$

only when the reliability of the pitch lag predicted value is low, and otherwise set the predicted value to

$$\hat{T}_p^{(i)}$$

(Step 5046 in Fig. 42). Further, pitch lag instruction information indicating whether the predicated value is used, or the pitch lag

$$\hat{T}_p^{(M_{la})}$$

obtained by the side information is used, may be input into the adaptive codebook calculation unit 1123.

**<When the side information index is a value indicating non-transmission of side information >**

**[0253]** In the prediction of the pitch lag

$$\hat{T}_p^{(i)} (M_{la} \leq i < M)$$

, the pitch lag prediction unit 192 predicts the pitch lag

$$\hat{T}_p^{(i)} (0 \leq i < M)$$

by using the pitch lag

$$\hat{T}_p^{(-j)} (1 \leq j < J)$$

used in the past decoding (Step 5048 in Fig. 42).

**[0254]** Further, the pitch lag prediction unit 192 may establish

$$\hat{T}_p^{(i)} = \hat{T}_p^{(-1)}$$

only when the reliability of the pitch lag predicted value is low (Step 5049 in Fig. 42), and the pitch lag prediction unit 192 can otherwise set the predicted value to

$$\hat{T}_p^{(i)}$$

. Further, pitch lag instruction information indicating whether the predicated value is used, or the pitch lag

$$\hat{T}_p^{(-1)}$$

used in past decoding is used, is input to the adaptive codebook calculation unit 1123 (Step 5050 in Fig. 42).

**[0255]** The adaptive codebook gain prediction unit 193 and the fixed codebook gain prediction unit 194 are the same as those of example 1.

**[0256]** The noise signal generation unit 195 is the same as that of the example 1.

**[0257]** The audio synthesis unit 124 synthesizes, from the parameters output from the audio parameter missing processing unit 123, an audio signal which corresponds to the frame to be encoded.

**[0258]** The LP coefficient calculation unit 1121 of the audio synthesis unit 124 (see Fig. 15) obtains an LP coefficient in the same manner as in example 1 (Step S11301 in Fig. 16).

**[0259]** The procedure of the adaptive codebook calculation unit 1123 is shown in the example of Fig. 43. The adaptive codebook calculation unit 1123 calculates an adaptive codebook vector in the same manner as in example 1. First, by referring to the pitch lag instruction information (Step 5051 in Fig. 43), when the reliability of the predicted value is low (YES in Step 5052 in Fig. 43), the adaptive codebook vector is calculated using the following equation (Step 5055 in Fig. 43). The filtering coefficient is  $f_i$ .

$$v(n)=f_{-1}v'(n-1)+f_0v'(n)+f_1v'(n+1) \quad \text{Equation 49}$$

**[0260]** Note that  $v(n)=v'(n)$  may be established according to the design policy.

**[0261]** By referring to the pitch lag instruction information, when the reliability of the predicted value is high (NO in Step 5052 in Fig. 43), the adaptive codebook calculation unit 1123 calculates the adaptive codebook vector by the following procedure.

**[0262]** First, the initial adaptive codebook vector is calculated using the pitch lag and the adaptive codebook 1122 (Step 5053 in Fig. 43).

$$v(n)=f_{-1}v'(n-1)+f_0v'(n)+f_1v'(n+1) \quad \text{Equation 50}$$

$v(n)=v'(n)$  may be established according to the design policy.

**[0263]** Then, glottal pulse synchronization is applied to the initial adaptive codebook vector. For the glottal pulse synchronization, the same procedure can be used as in the case where a pulse position is not available in section 7.11.2.5 in Non Patent Literature 4 (Step 5054 in Fig. 43). Note that, however,  $u(n)$  in Non Patent Literature 4 corresponds to  $v(n)$  in the embodiment of the present invention, extrapolated pitch corresponds to

$$\hat{T}_p^{(M-1)}$$

in the embodiment of the present invention, and the last reliable pitch( $T_c$ ) corresponds to

$$\hat{T}_p^{(-1)}$$

in the embodiment of the present invention.

**[0264]** The excitation vector synthesis unit 1124 outputs an excitation signal vector in the same manner as in the example 1 (Step 11306 in Fig. 16).

**[0265]** The post filter 1125 performs post processing on the synthesis signal in the same manner as in example 1.

**[0266]** The adaptive codebook 1122 updates the state using the excitation signal vector in the same manner as in the example 1 (Step 11308 in Fig. 16).

**[0267]** The synthesis filter 1126 synthesizes a decoded signal in the same manner as in example 1 (Step 11309 in Fig. 16).

**[0268]** The perceptual weighting inverse filter 1127 applies an perceptual weighting inverse filter in the same manner as in example 1.

**[0269]** The audio parameter missing processing unit 123 stores the audio parameters (ISF parameter, pitch lag, adaptive codebook gain, fixed codebook gain) used in the audio synthesis unit 124 into the buffer in the same manner as in example 1 (Step 145 in Fig. 7).

### Reference Signs List

**[0270]** 60,80...storage medium, 61, 81...program storage area, 70...audio encoding program, 90...audio decoding program, 111...audio encoding unit, 112...side information encoding unit, 121, 231...audio code buffer, 122, 232...audio parameter decoding unit, 123, 233...audio parameter missing processing unit, 124, 234...audio synthesis unit, 125, 235...side information decoding unit, 126, 236...side information accumulation unit, 151, 511, 1121...LP coefficient calculation unit, 152, 2012...target signal calculation unit, 153, 513, 2013...pitch lag calculation unit, 154, 1123, 514, 2014, 2313...adaptive codebook calculation unit, 155, 1124, 2314...excitation vector synthesis unit, 156, 315, 515, 2019...adaptive codebook buffer, 157, 1126, 2018, 2316...synthesis filter, 158, 516...pitch lag encoding unit, 191...ISF prediction unit, 192...pitch lag prediction unit, 193...adaptive codebook gain prediction unit, 194...fixed codebook gain prediction unit, 195...noise signal generation unit, 211...main encoding unit, 212...side information encoding unit, 213, 238...concealment signal accumulation unit, 214...error signal encoding unit, 237...error signal decoding unit, 311...LP coefficient calculation unit, 312...pitch lag prediction unit, 313...pitch lag selection unit, 314...pitch lag encoding unit, 512...residual signal calculation unit, 700...audio encoding module, 701...side information encoding module, 900...audio parameter decoding module, 901...audio parameter missing processing module, 902...audio synthesis module, 903...side information decoding module, 1128...side information output determination unit, 1122, 2312...adaptive codebook, 1125...post filter, 1127...perceptual weighting inverse filter, 2011...ISF encoding unit, 2015...fixed codebook calculation unit, 2016...gain calculation unit, 2017...excitation vector calculation unit, 2211...ISF

decoding unit, 2212...pitch lag decoding unit, 2213...gain decoding unit, 2214...fixed codebook decoding unit, 2318...look-ahead excitation vector synthesis unit.

## REFERENCES CITED IN THE DESCRIPTION

### Cited references

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

### Patent documents cited in the description

- [JP2004138756A \[0018\]](#)
- [WO2012110482A2 \[0018\]](#)
- [JP2003533916A \[0019\]](#)
- [JP2010507818A \[0019\]](#)

**Patentkrav**

**1.** Audiokodningsindretning til kodning af et audiosignal, hvilken audiokodningsindretning omfatter:

- 5 en audiokoder til kodning af audiosignalet; og  
en sideinformationskoder til beregning af et parameter for et fremadrettet signal i CELP-kodning som sideinformation, der skal anvendes i skjulning af pakkeab i CELP-kodningen,  
hvor audiokoderen beregner et indeks, der repræsenterer karakteristika af en
- 10 ramme, der skal kodes, og sender indekset til sideinformationskoderen,  
når indekset indikerer, at rammen, der skal kodes, er i en specifik rammeklasse,  
inkluderes en forskel mellem en tonehøjdeforsinkelse af rammen, der skal kodes,  
og en tonehøjdeforsinkelse af en ramme umiddelbart efter rammen, der skal  
kodes, som sideinformationen, i en pakke af rammen, der skal kodes,
- 15 idet en audiokode er et kodet audiosignal af rammen, der skal kodes, og  
sideinformation kan sendes, før audiokoderen koder et audiosignal af rammen  
umiddelbart efter rammen, der skal kodes, og  
når indekset indikerer, at rammen, der skal kodes, ikke er i en specifik  
rammeklasse, inkluderes tonehøjdeforsinkelsen af rammen umiddelbart efter
- 20 rammen, der skal kodes, ikke i pakken af rammen, der skal kodes.

# DRAWINGS

Drawing

**Fig.1**

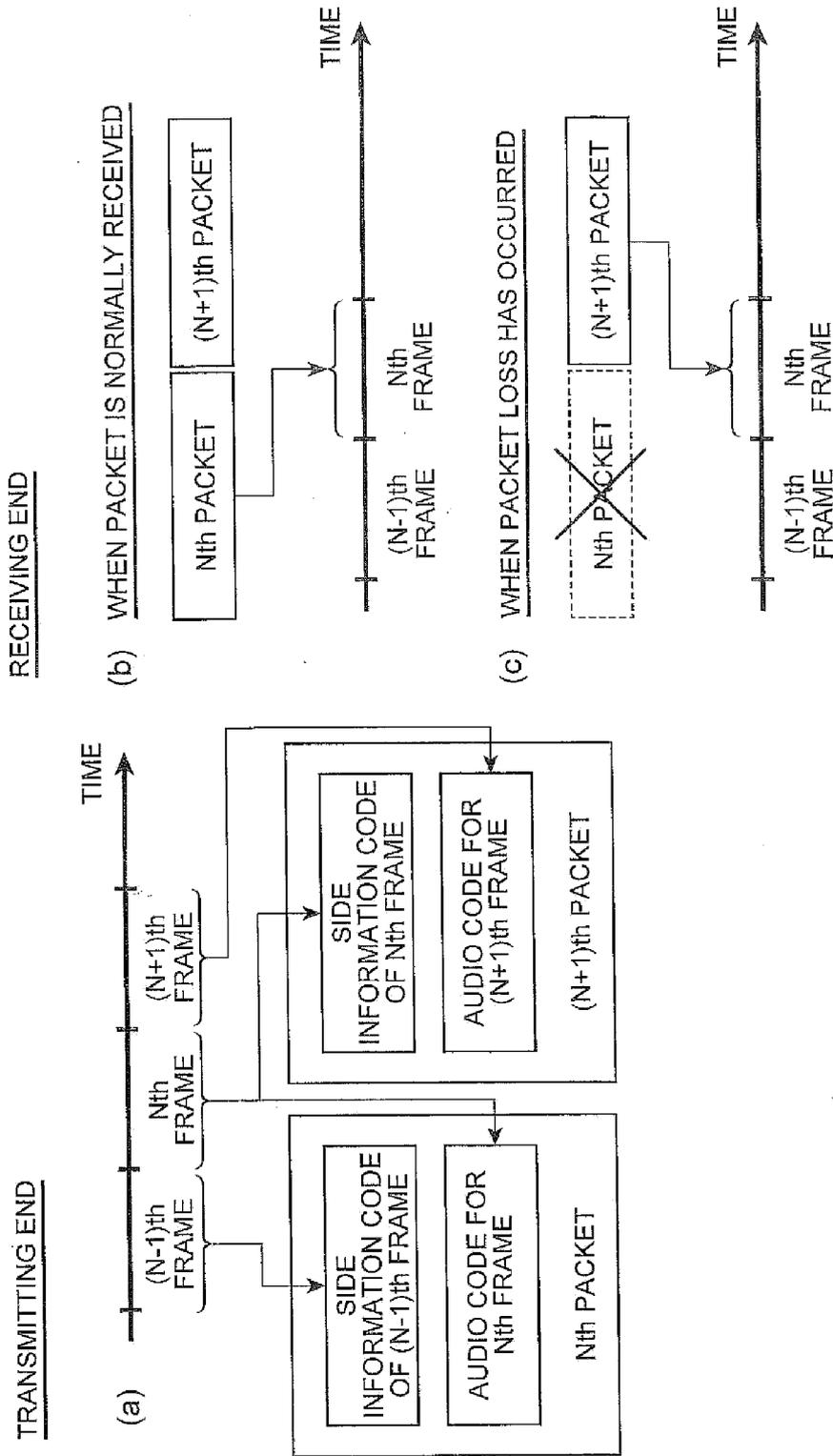
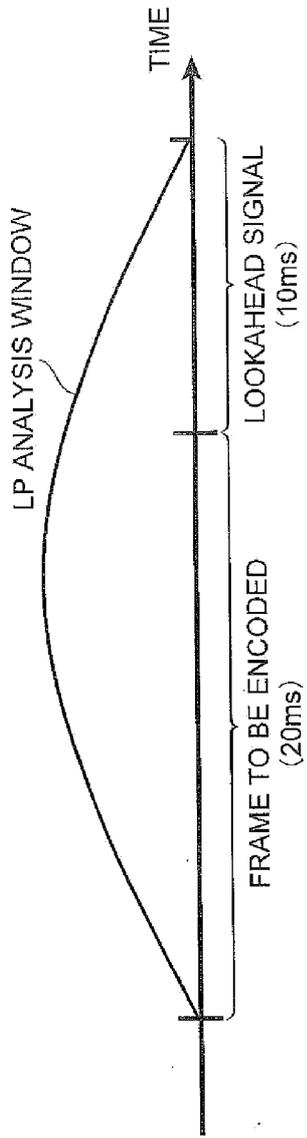
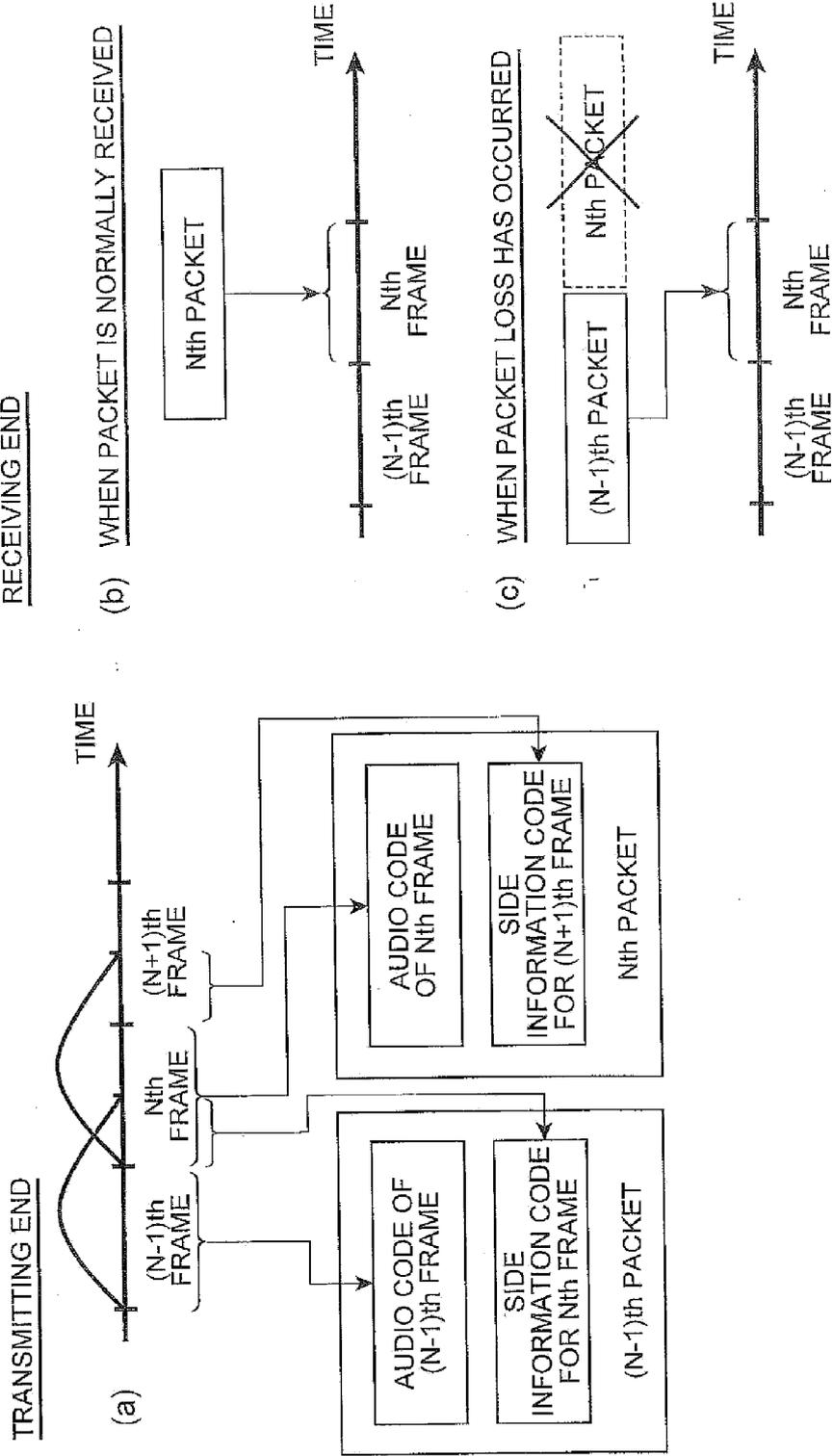




Fig.2



**Fig. 3**



**Fig.4**

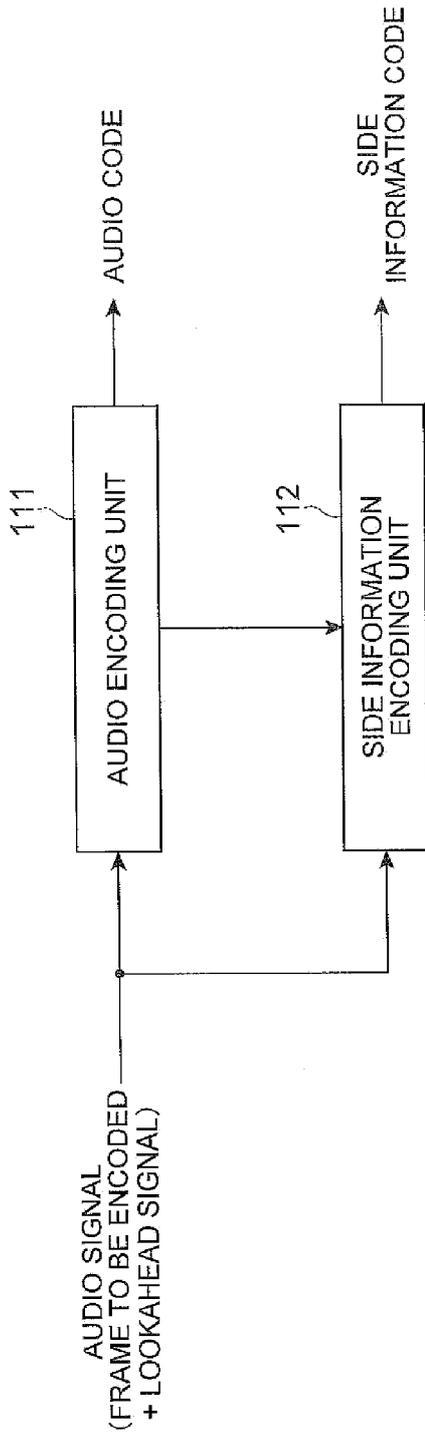


Fig.5

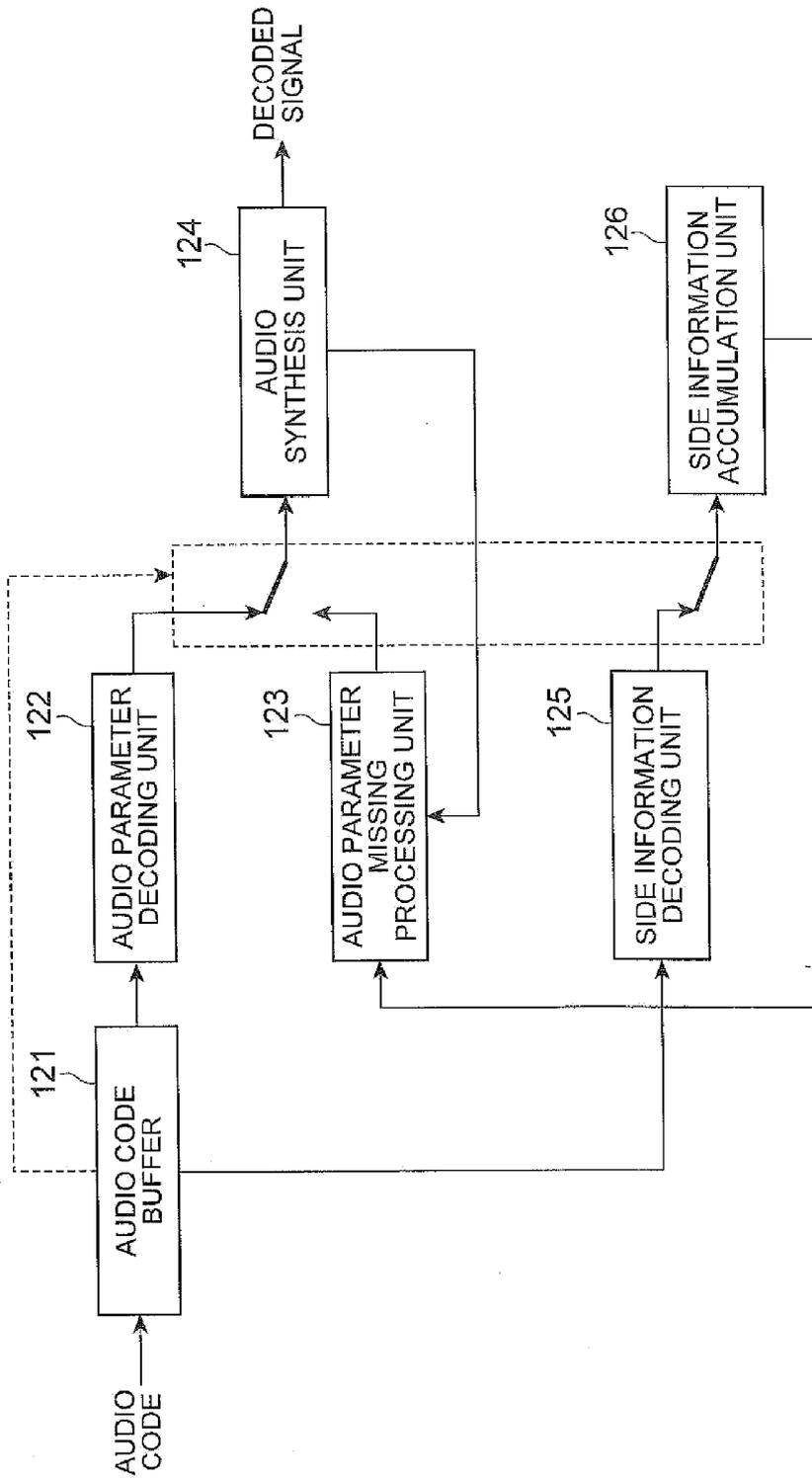


Fig.6

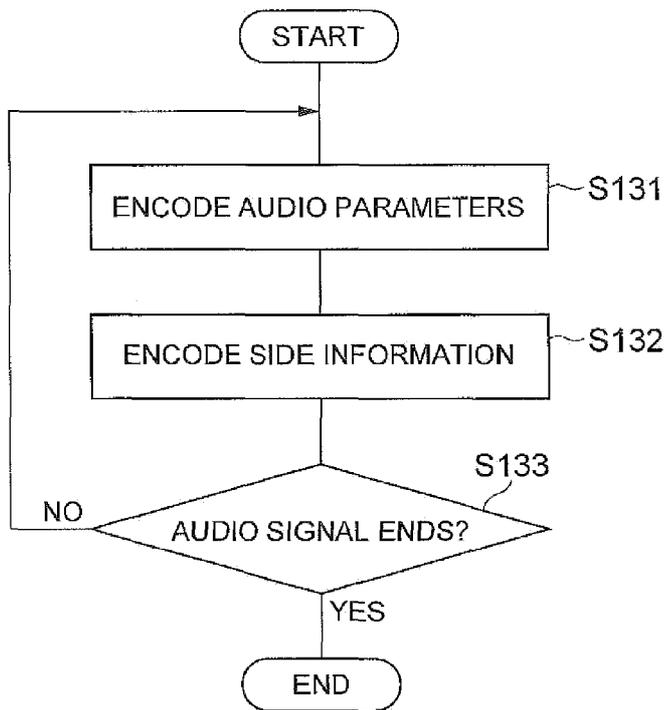
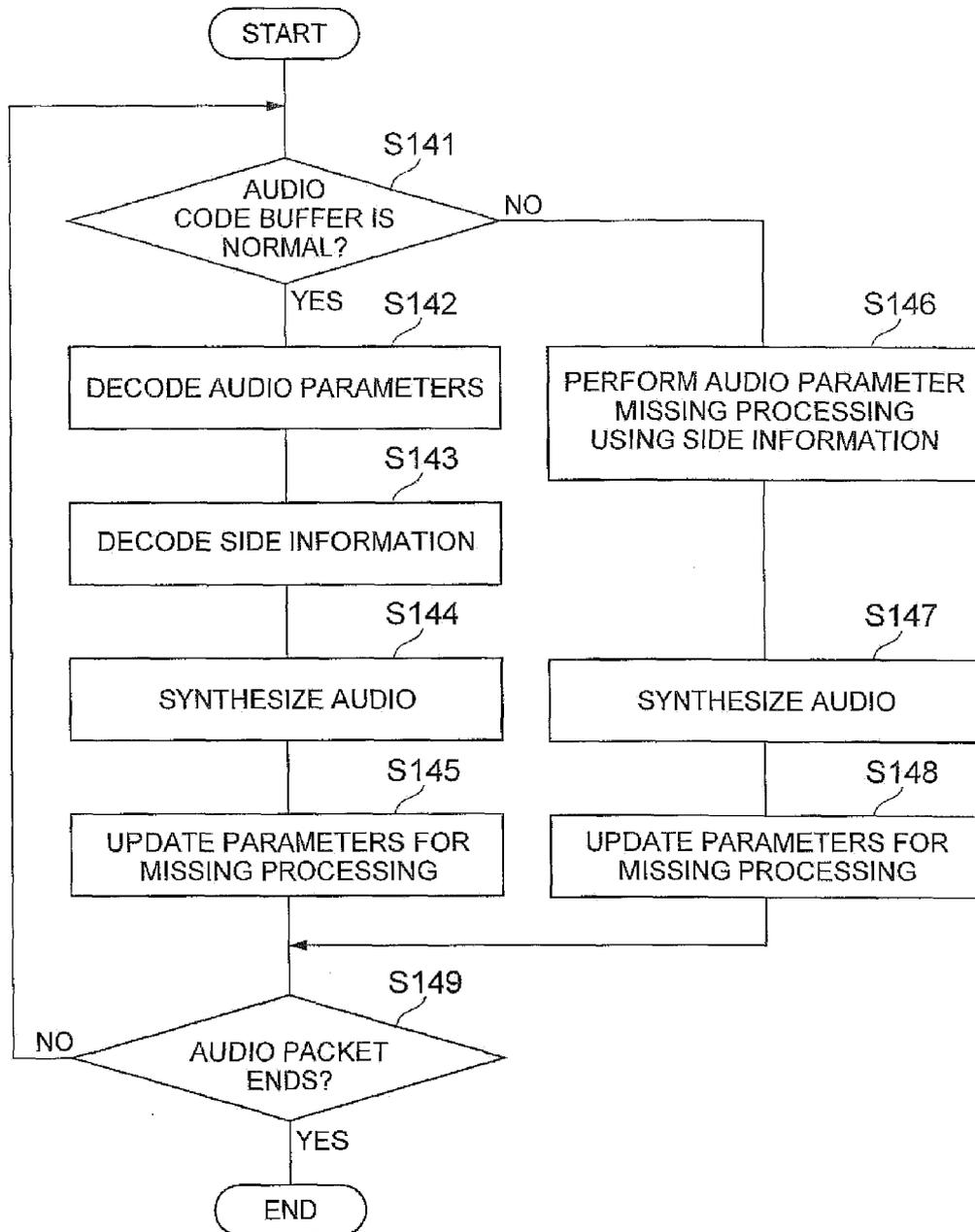


Fig.7



**Fig.8**

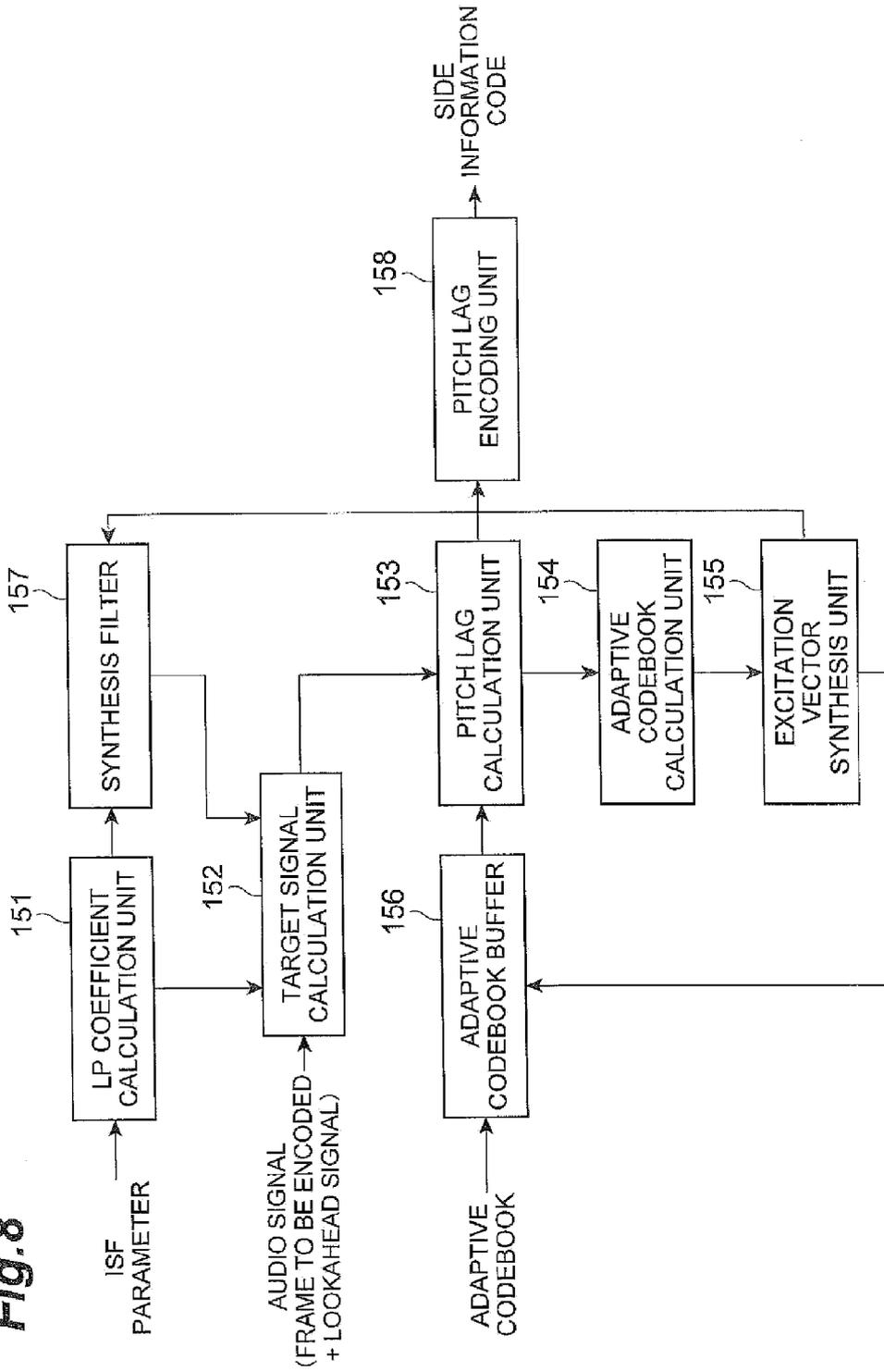


Fig.9

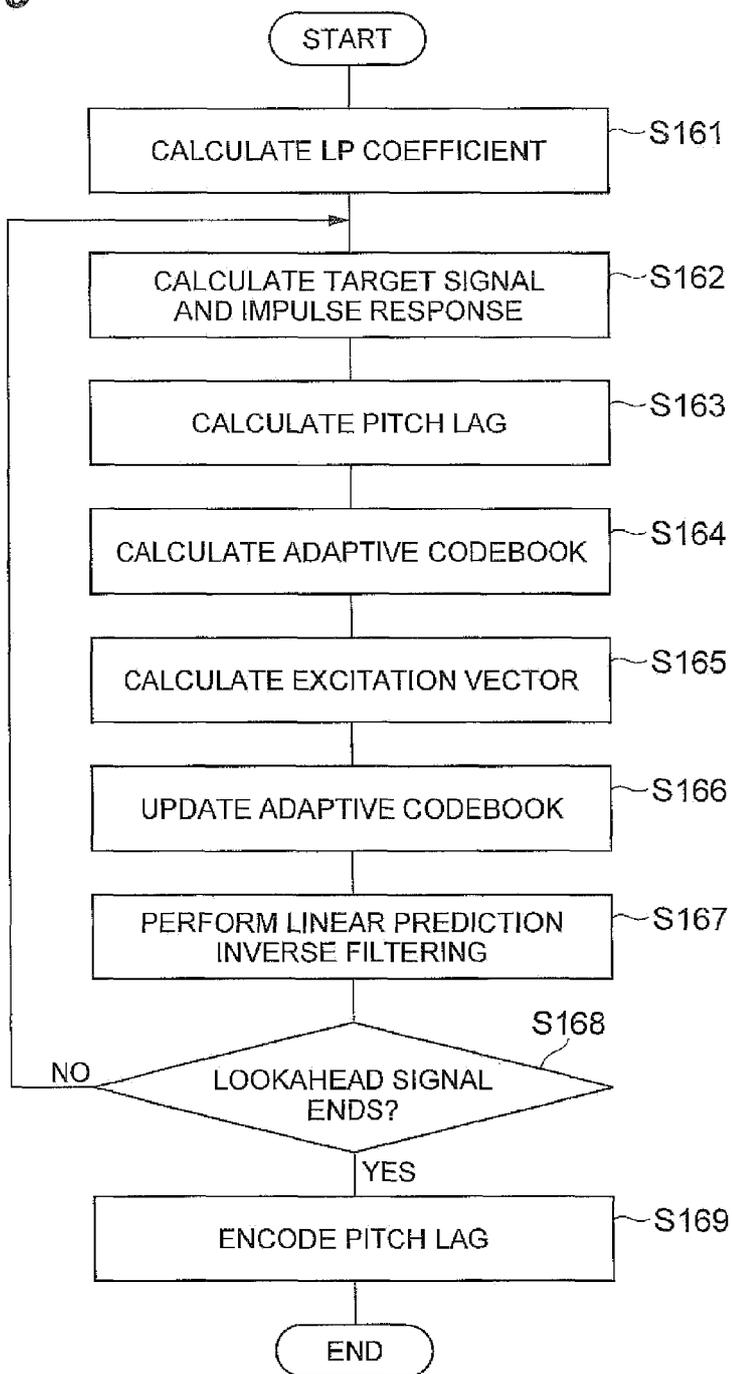
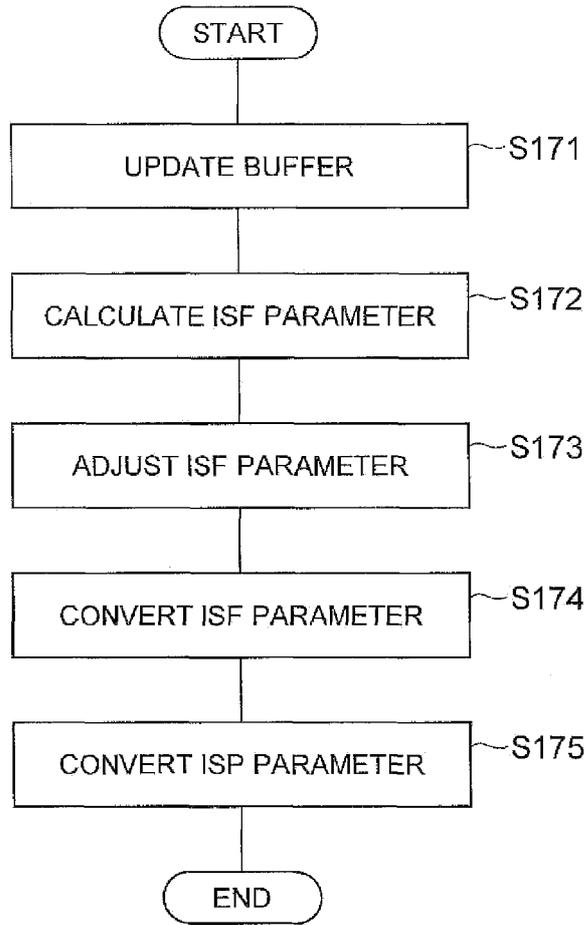


Fig.10



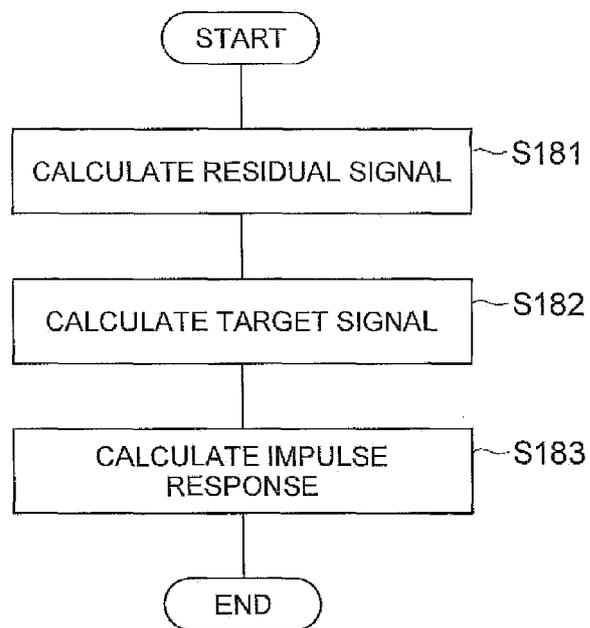
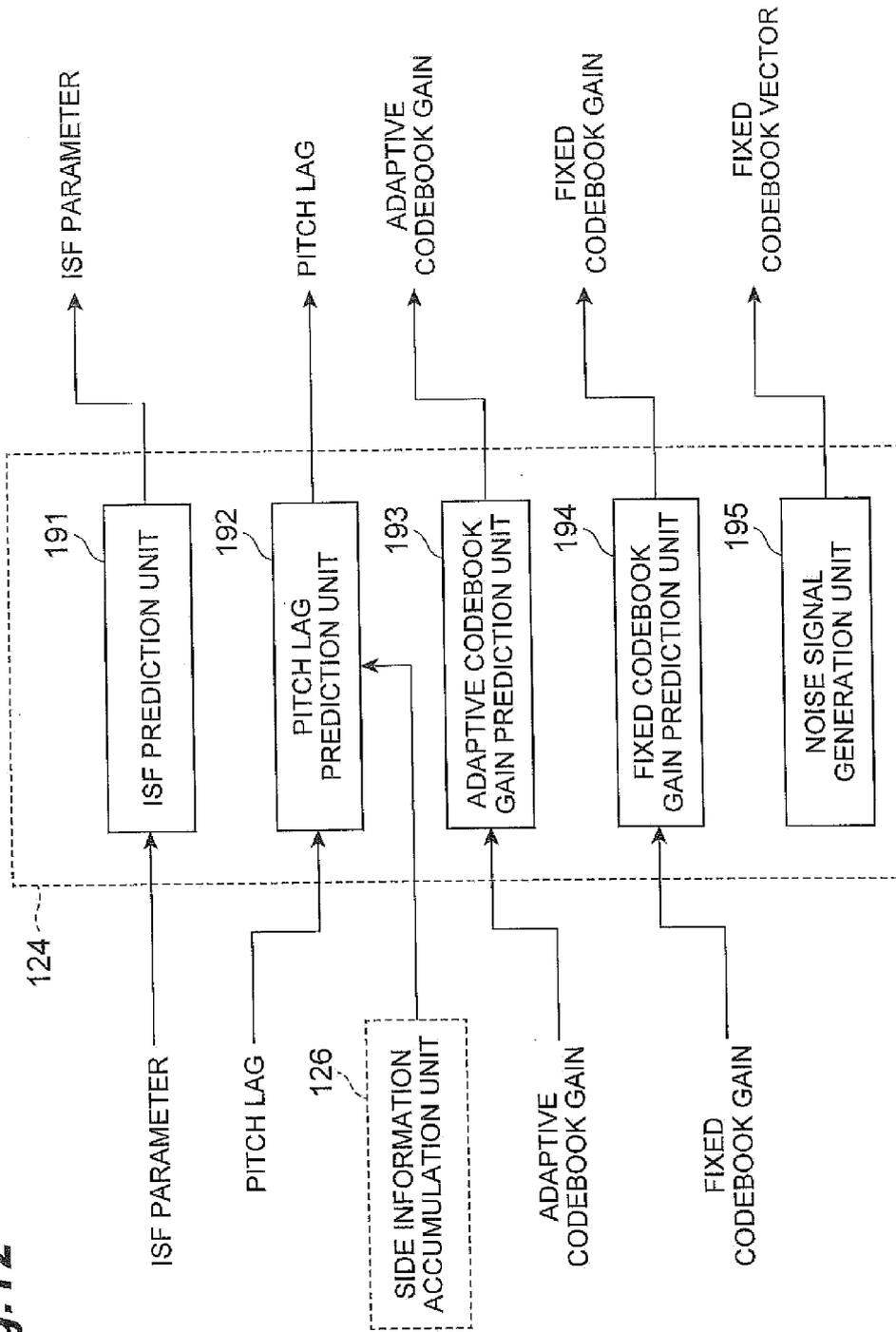
**Fig.11**

Fig.12



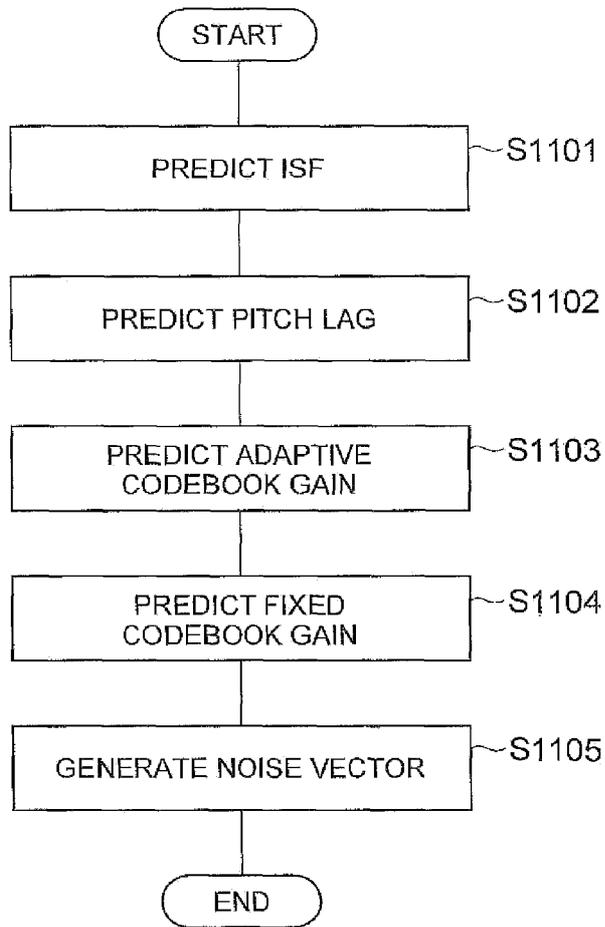
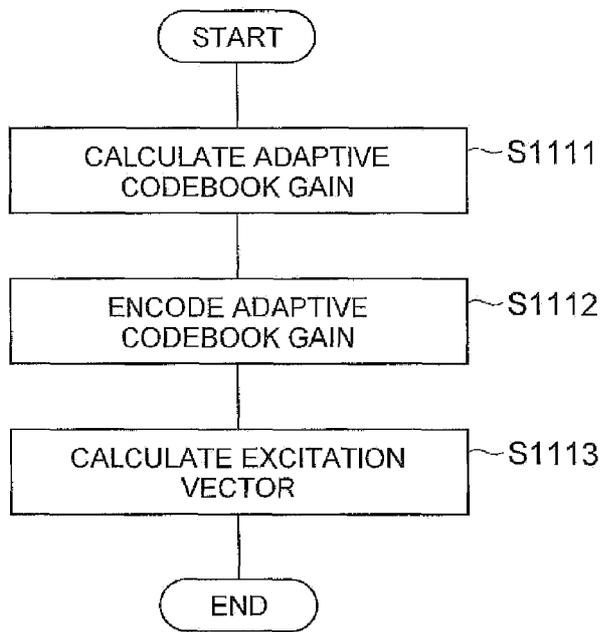
**Fig.13**

Fig.14



**Fig. 15**

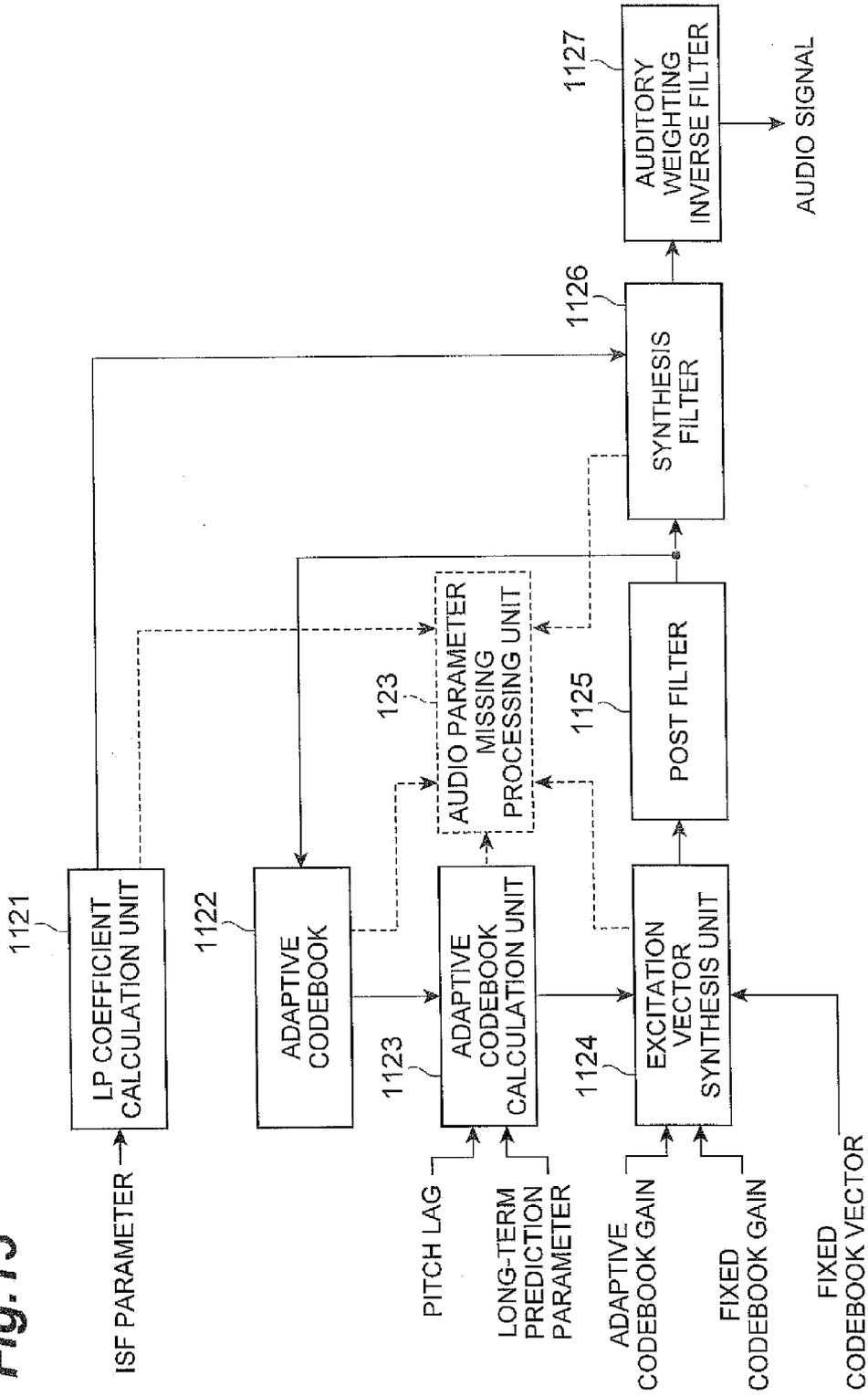


Fig. 16

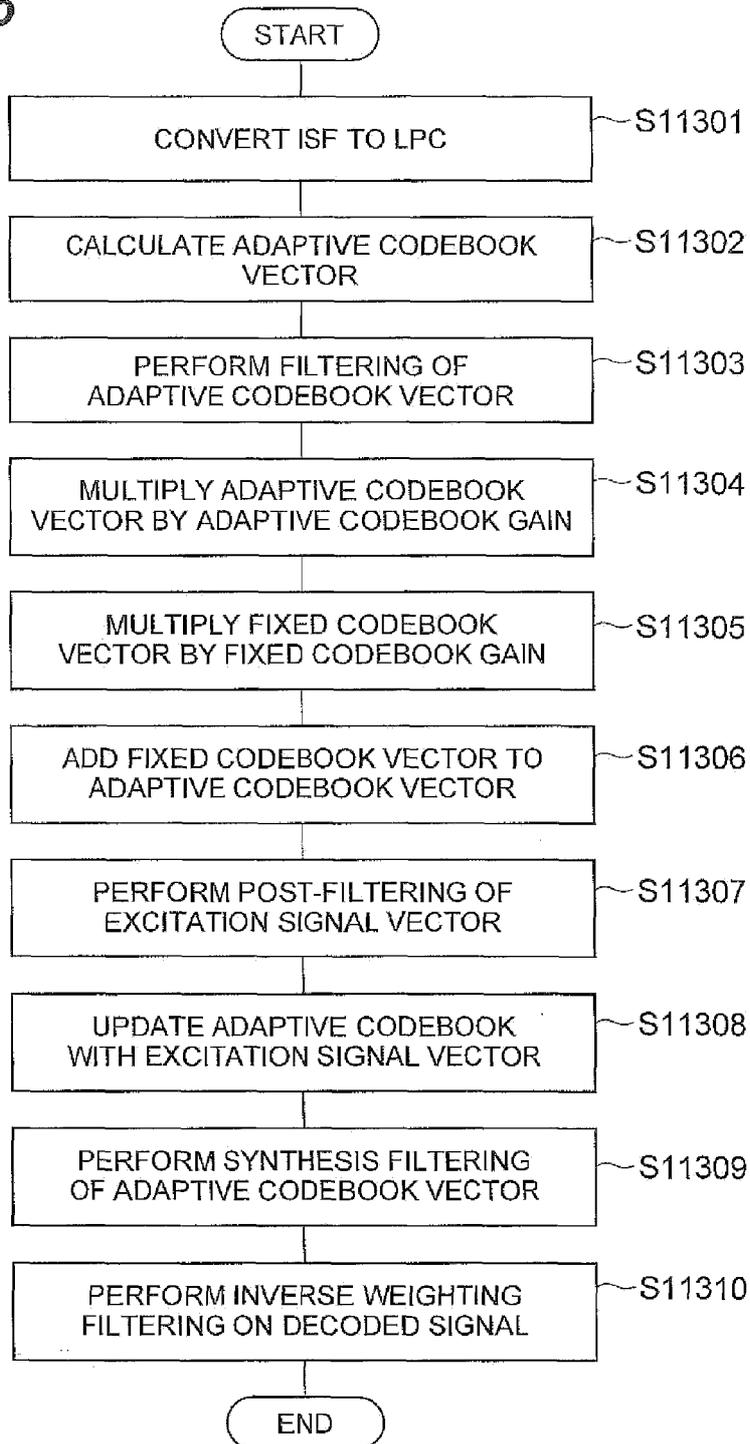


Fig.17

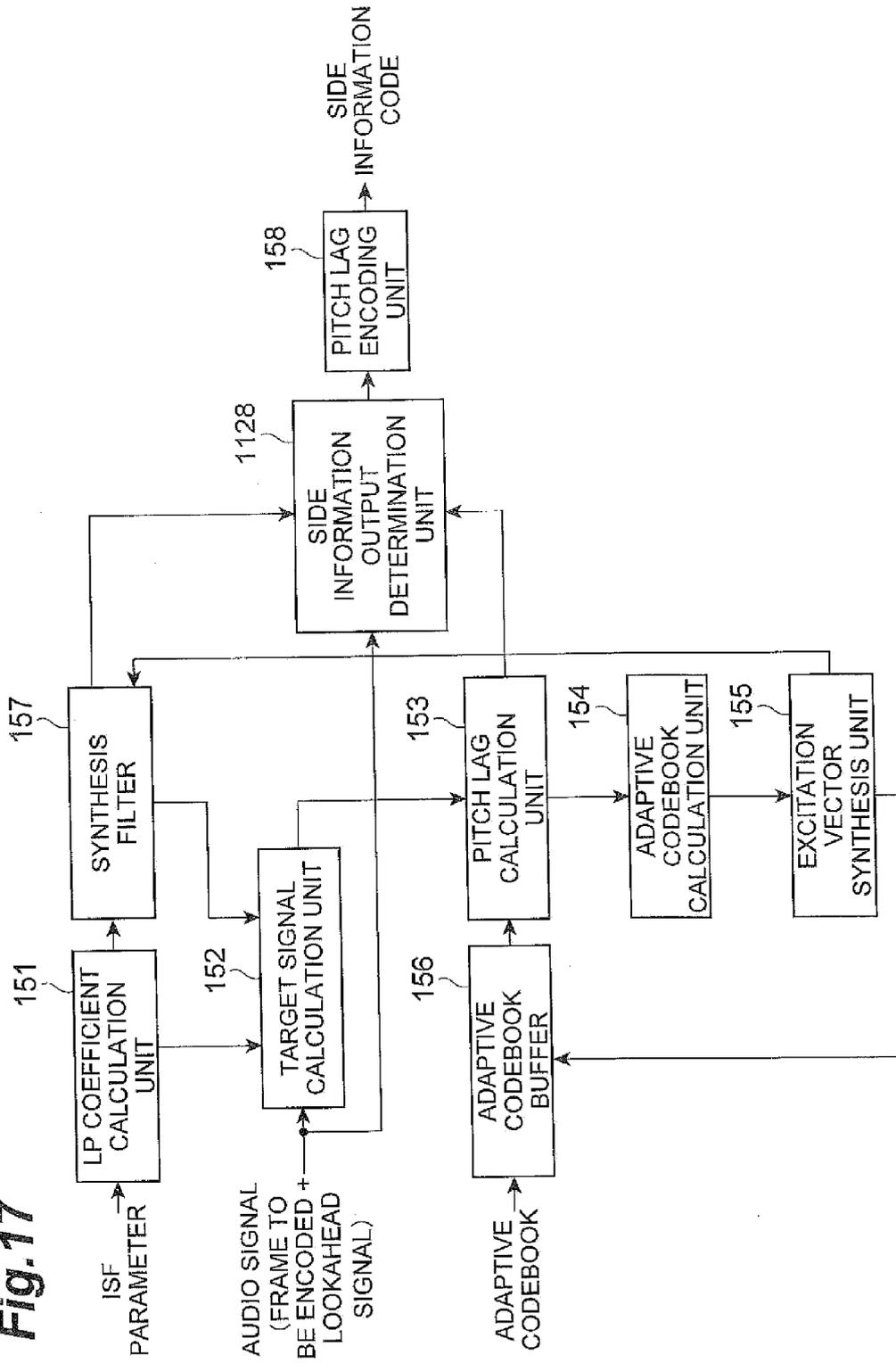


Fig.18

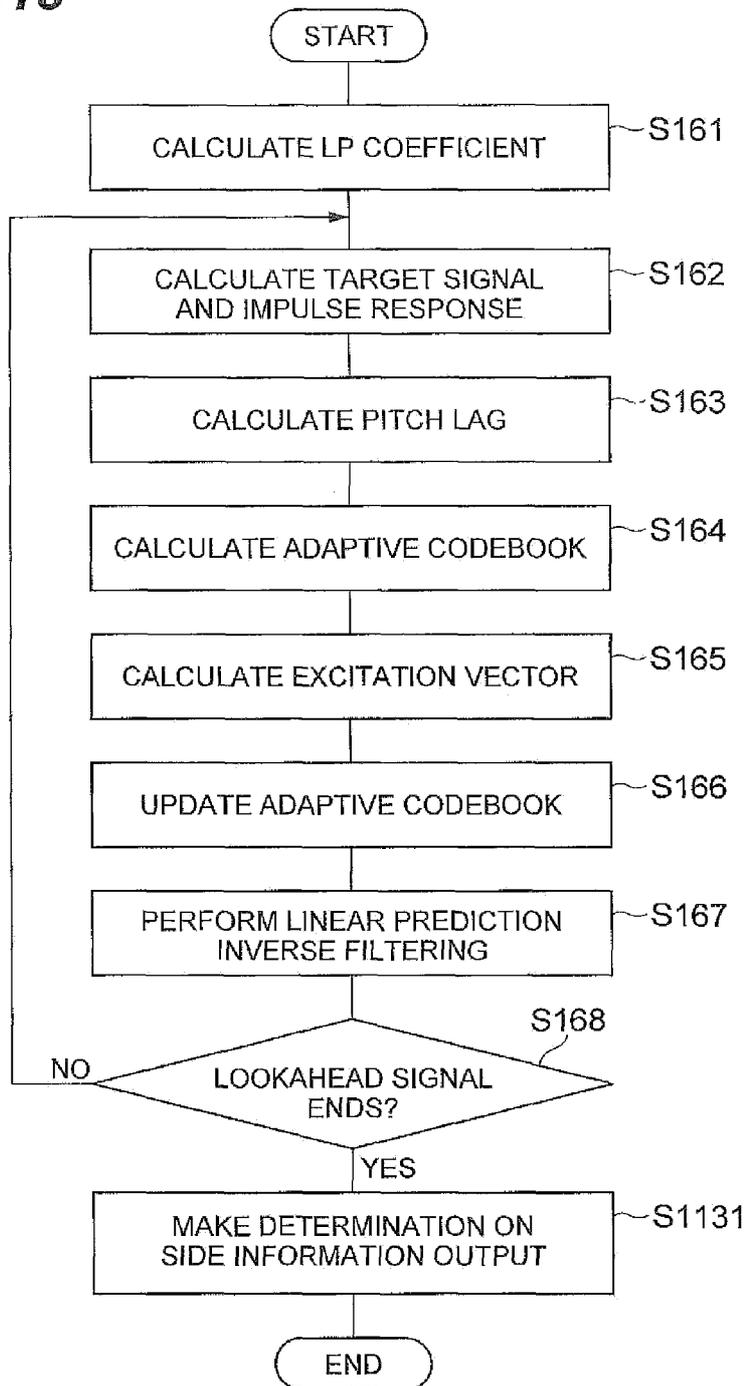


Fig. 19

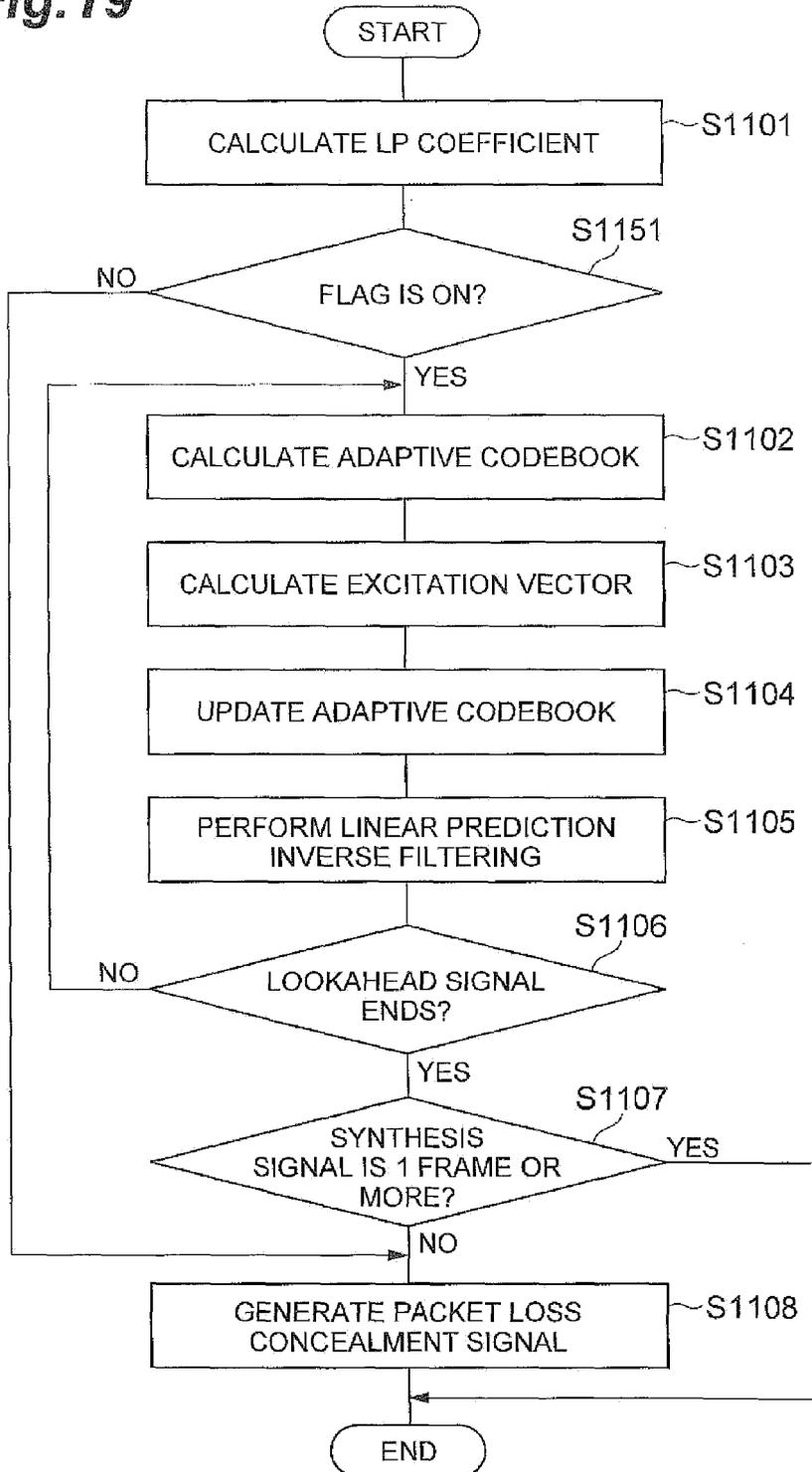
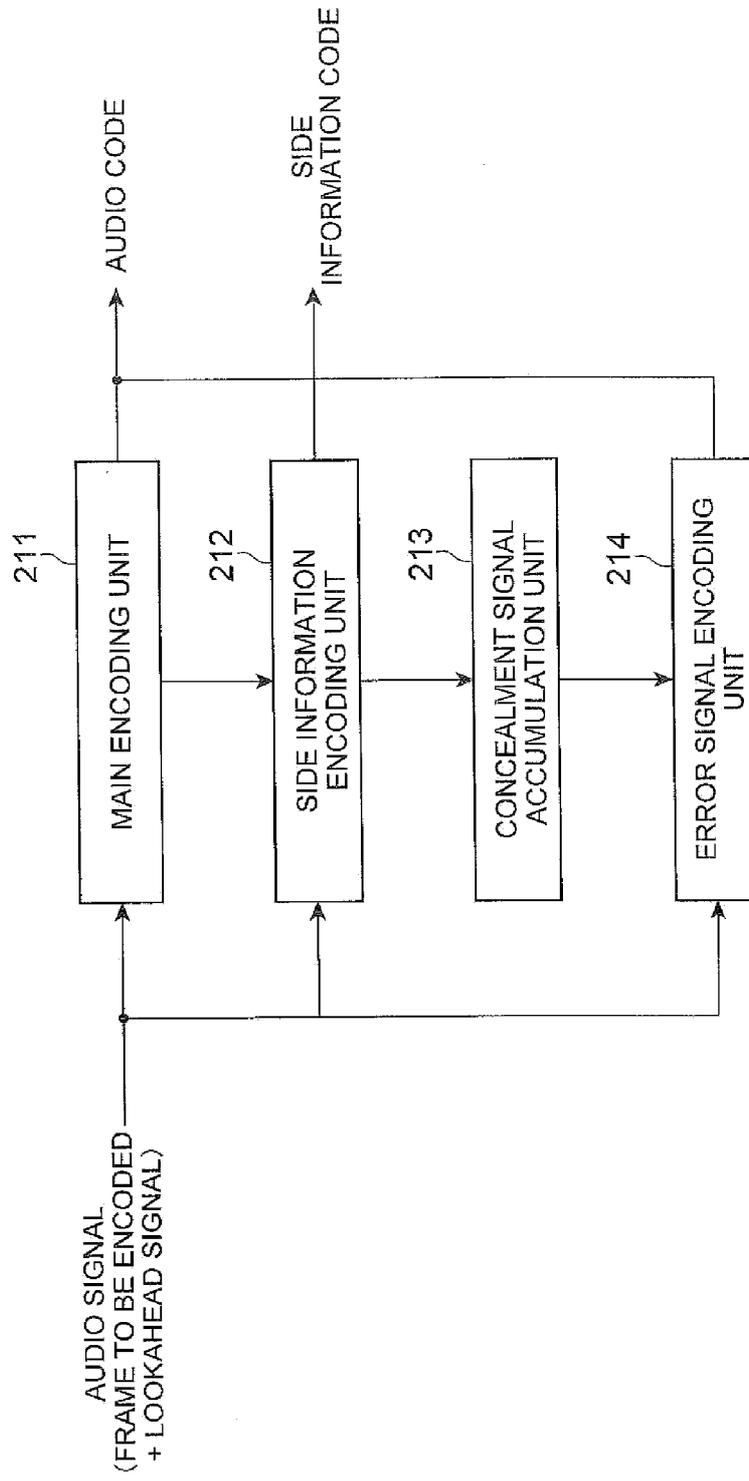


Fig. 20



**Fig. 21**

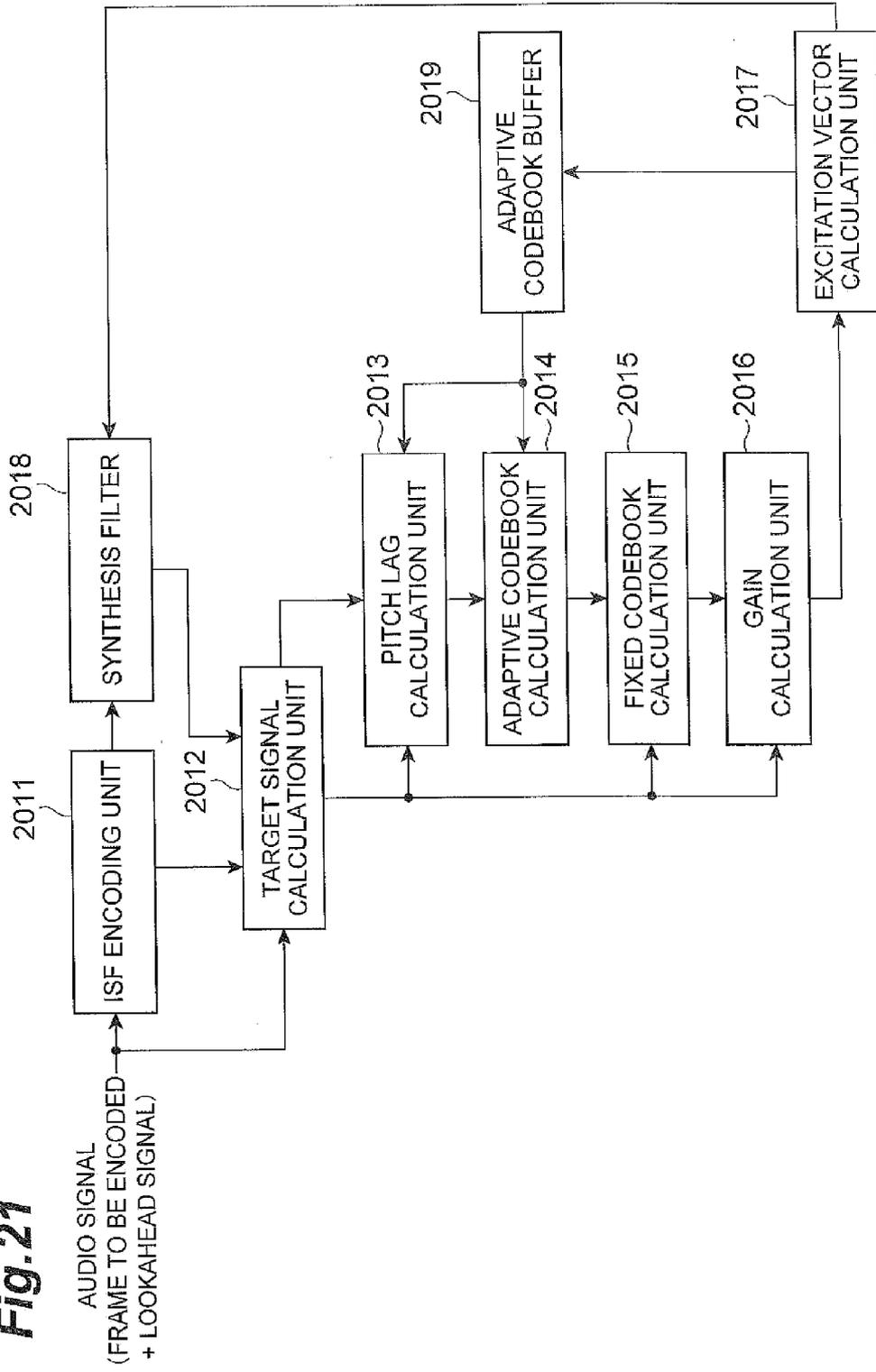


Fig.22

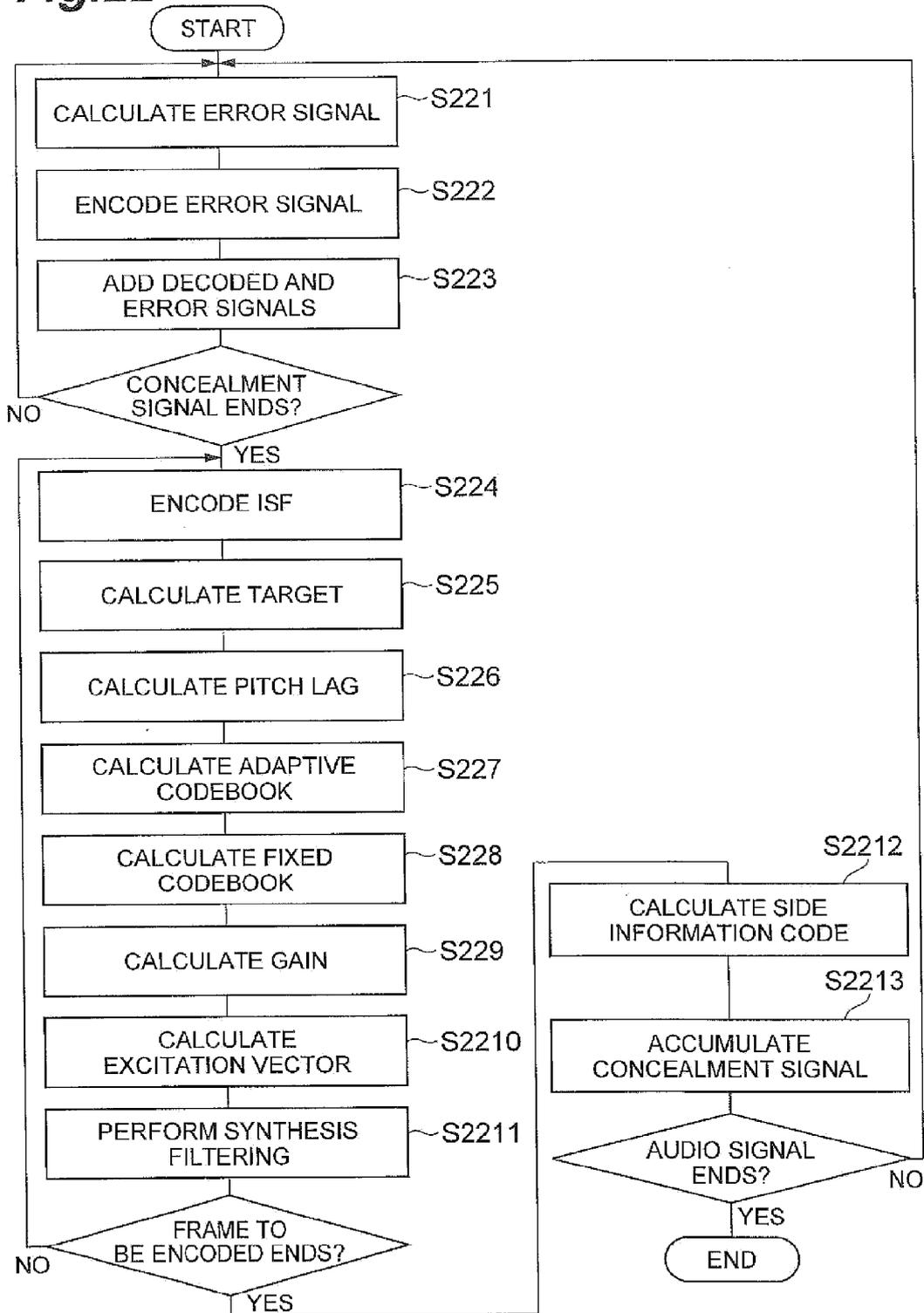


Fig. 23

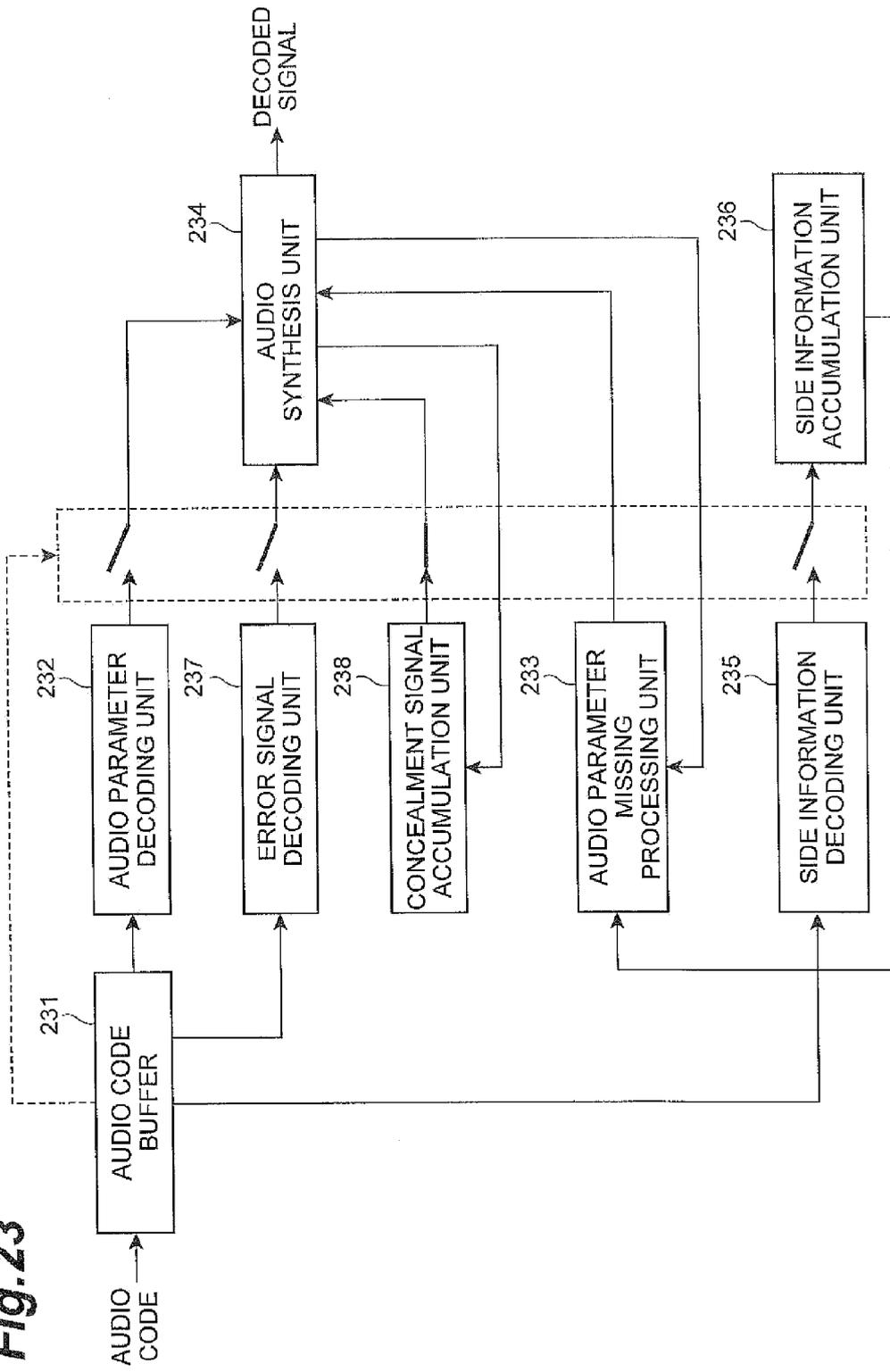


Fig. 24

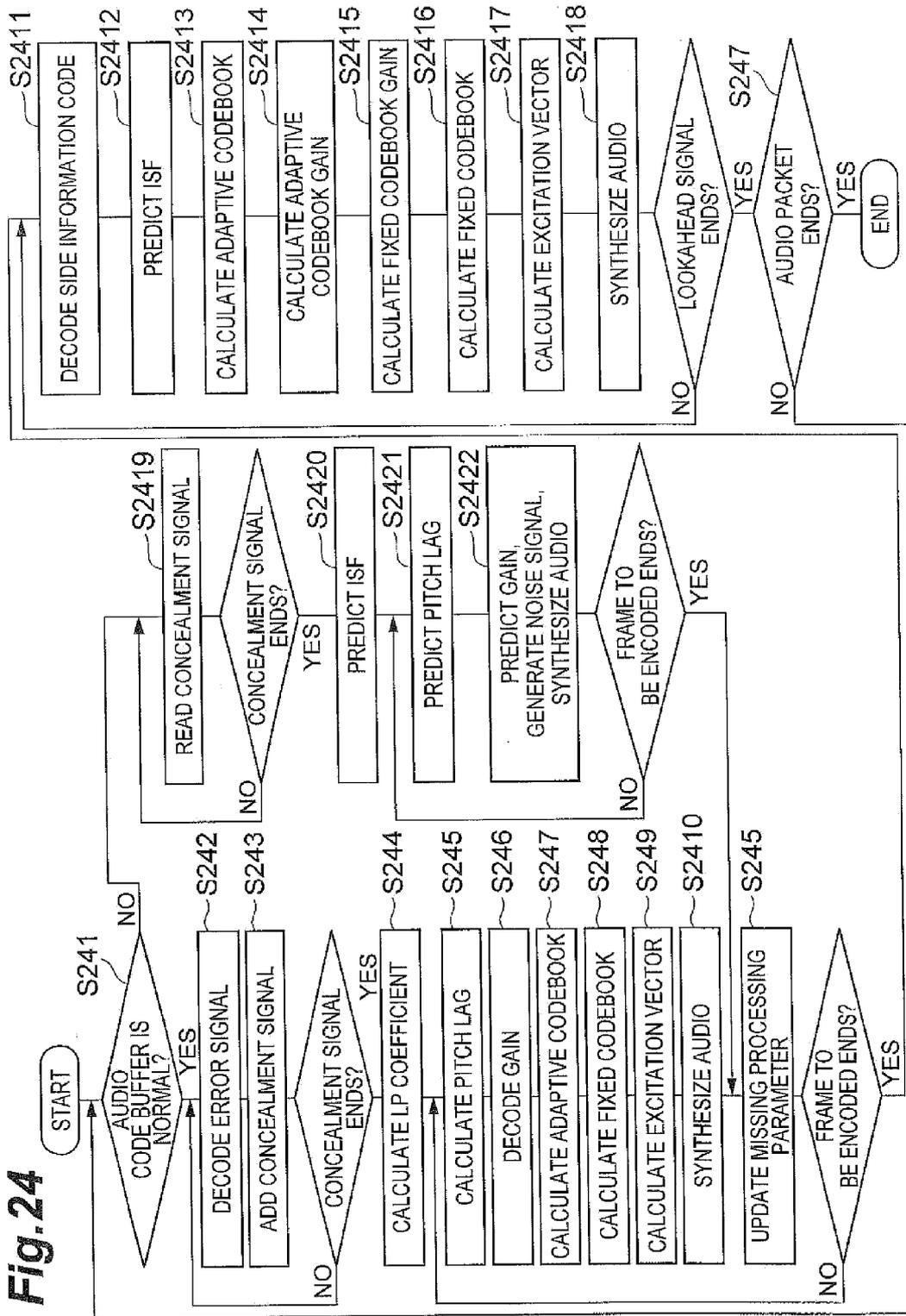


Fig. 25

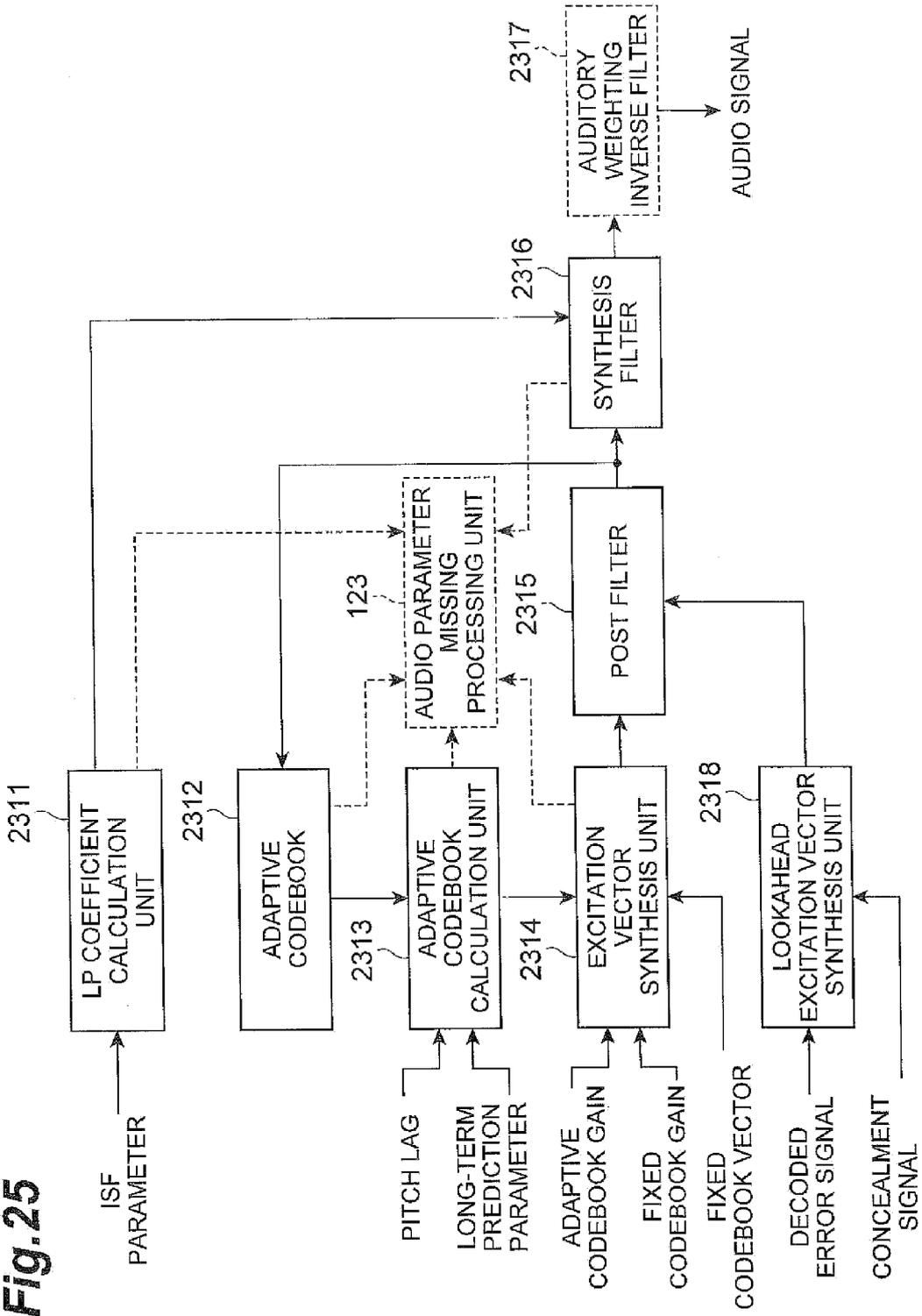


Fig. 26

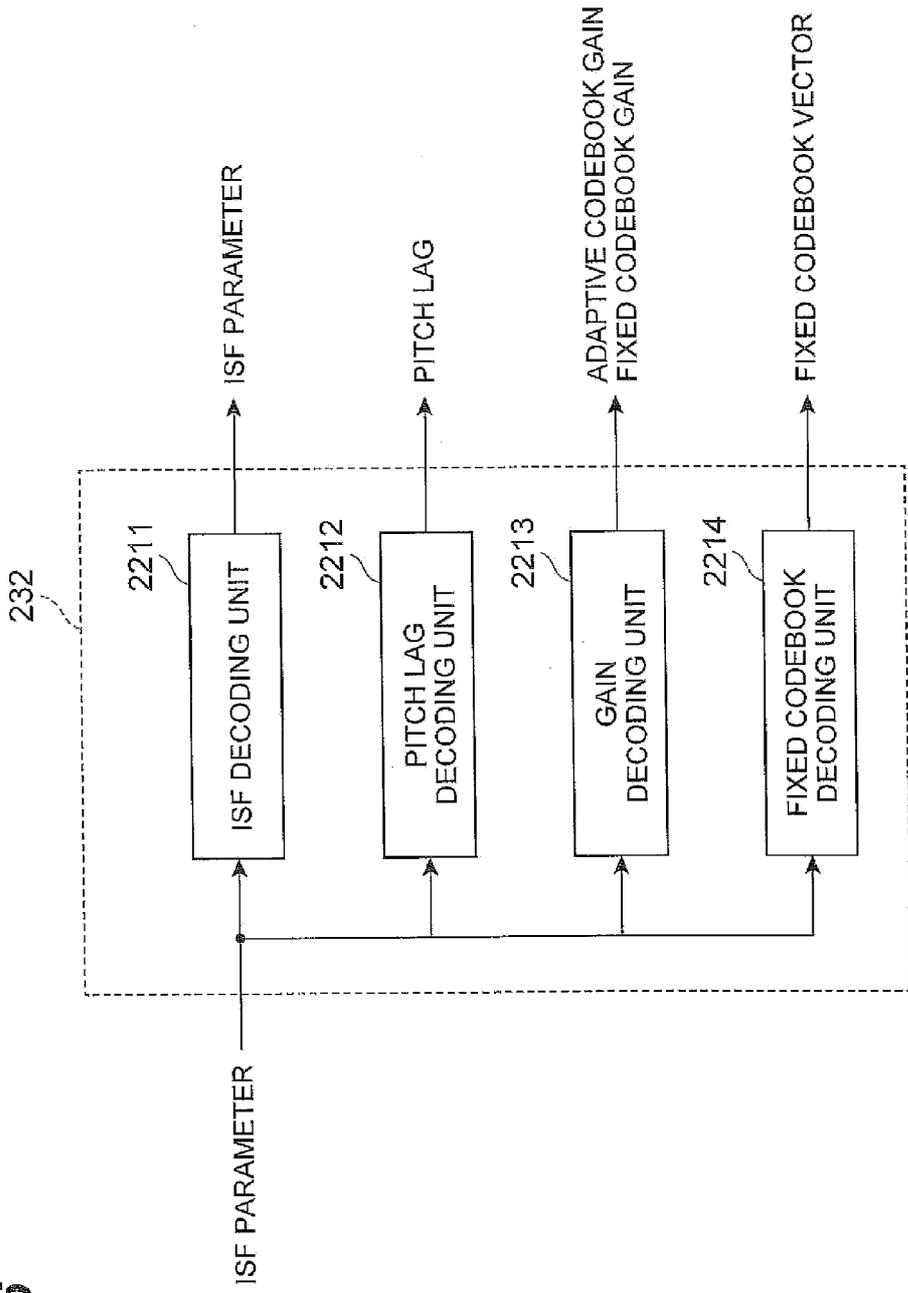


Fig.27

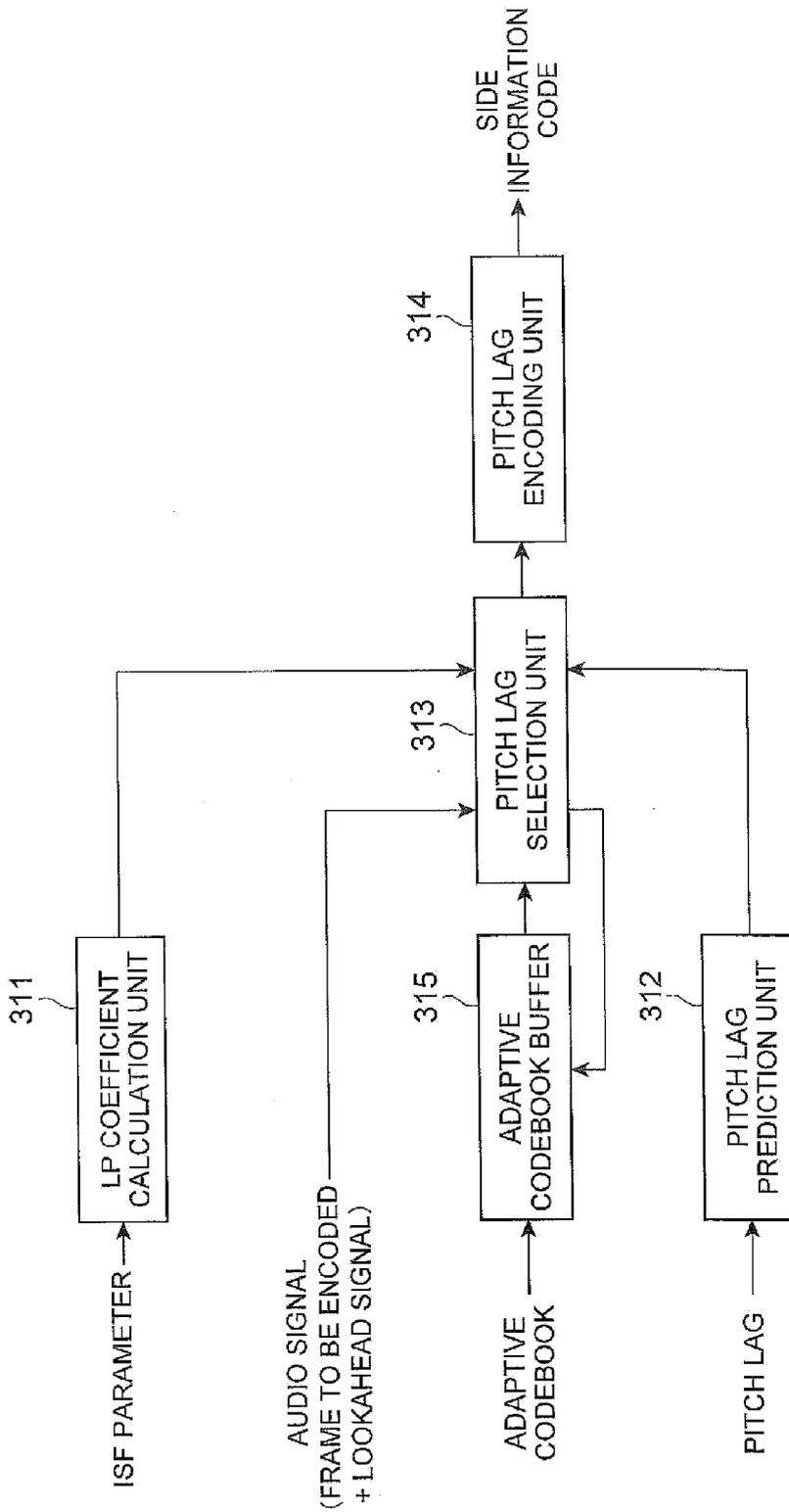


Fig.28

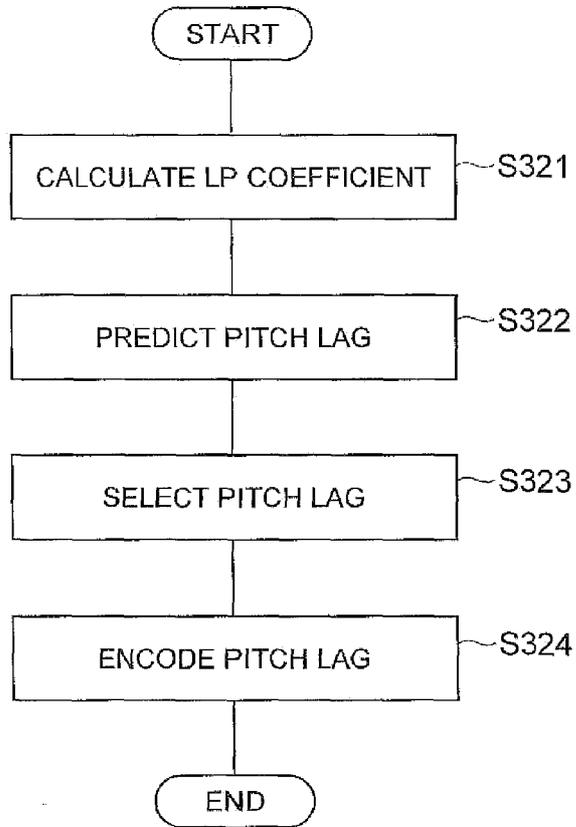
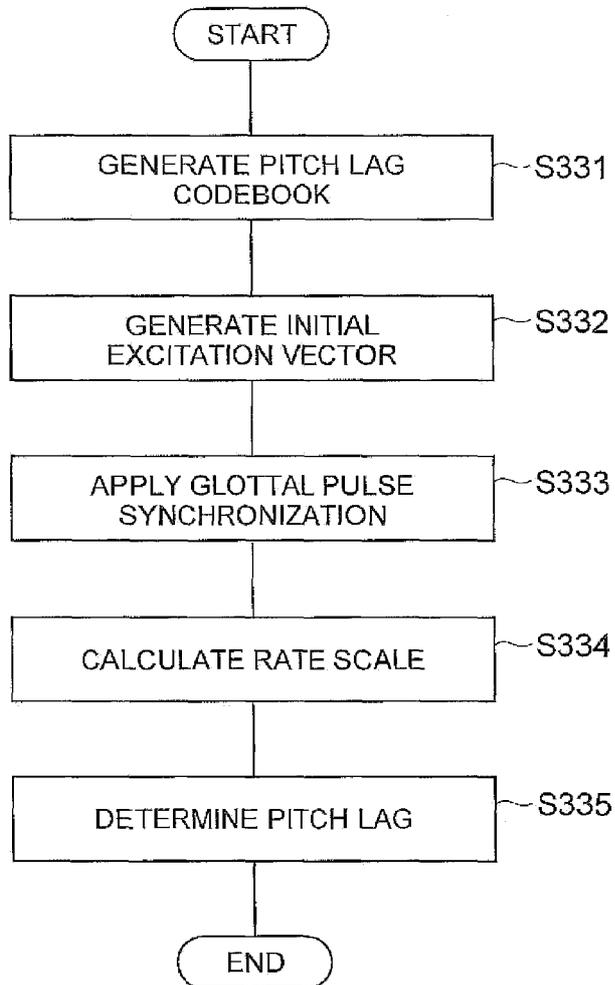
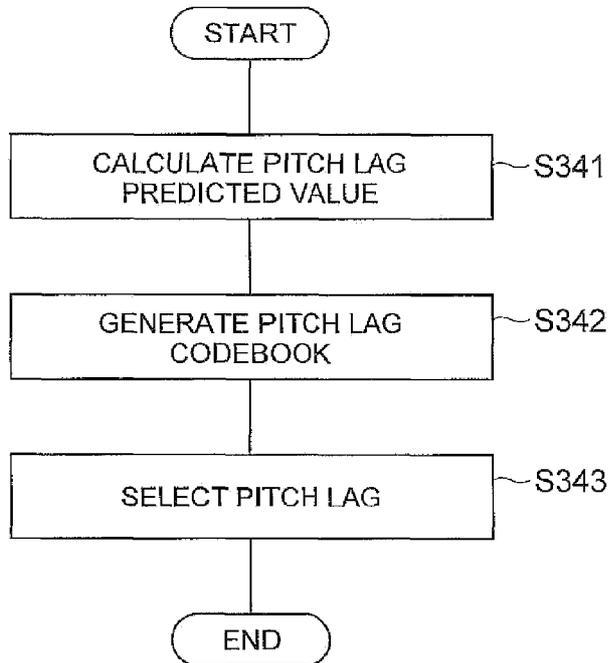


Fig.29



**Fig.30**

**Fig.31**

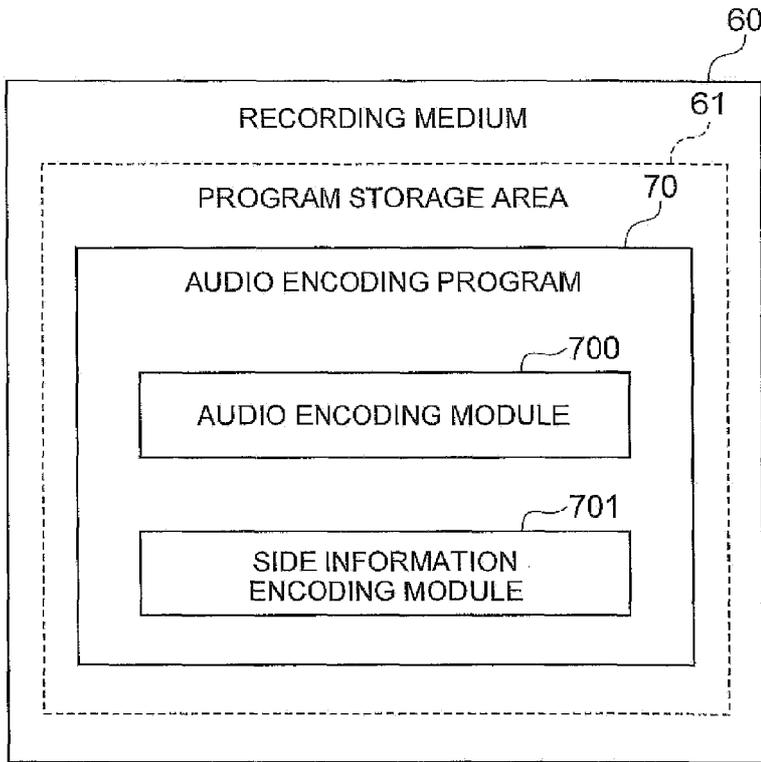


Fig.32

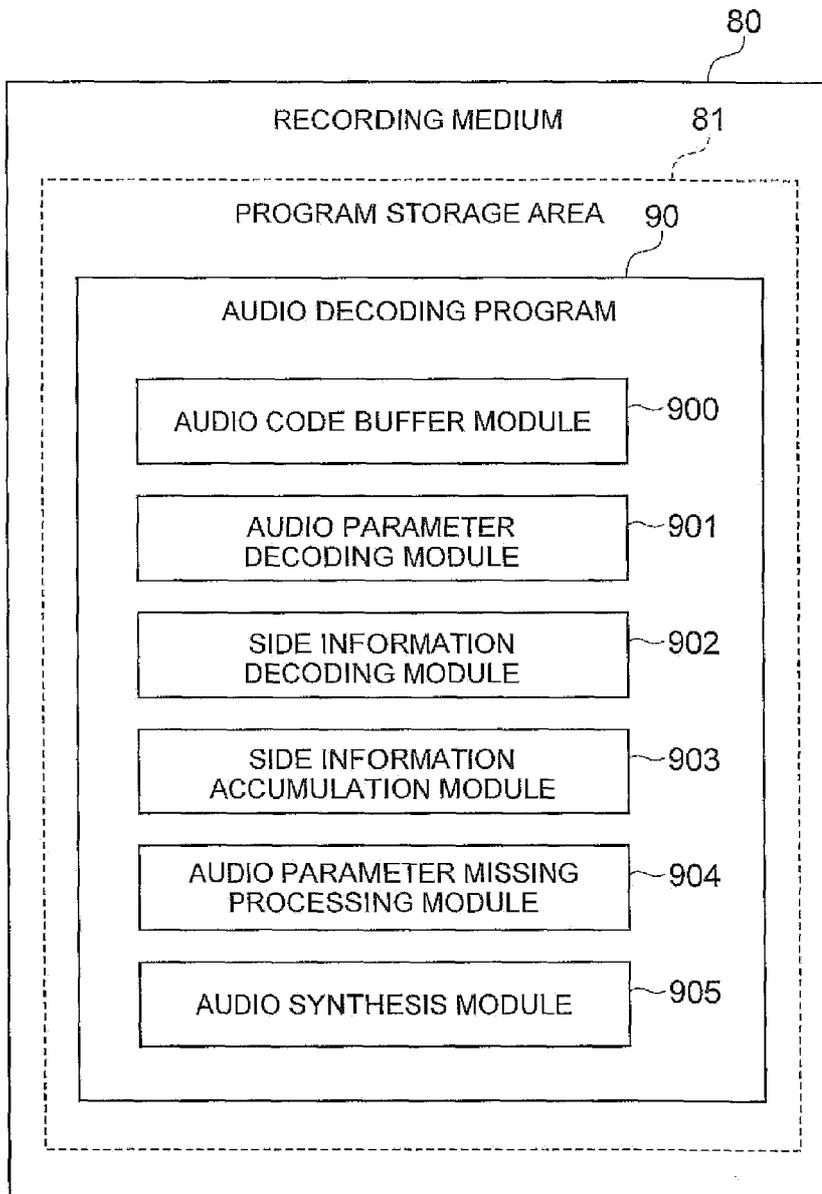


Fig.33

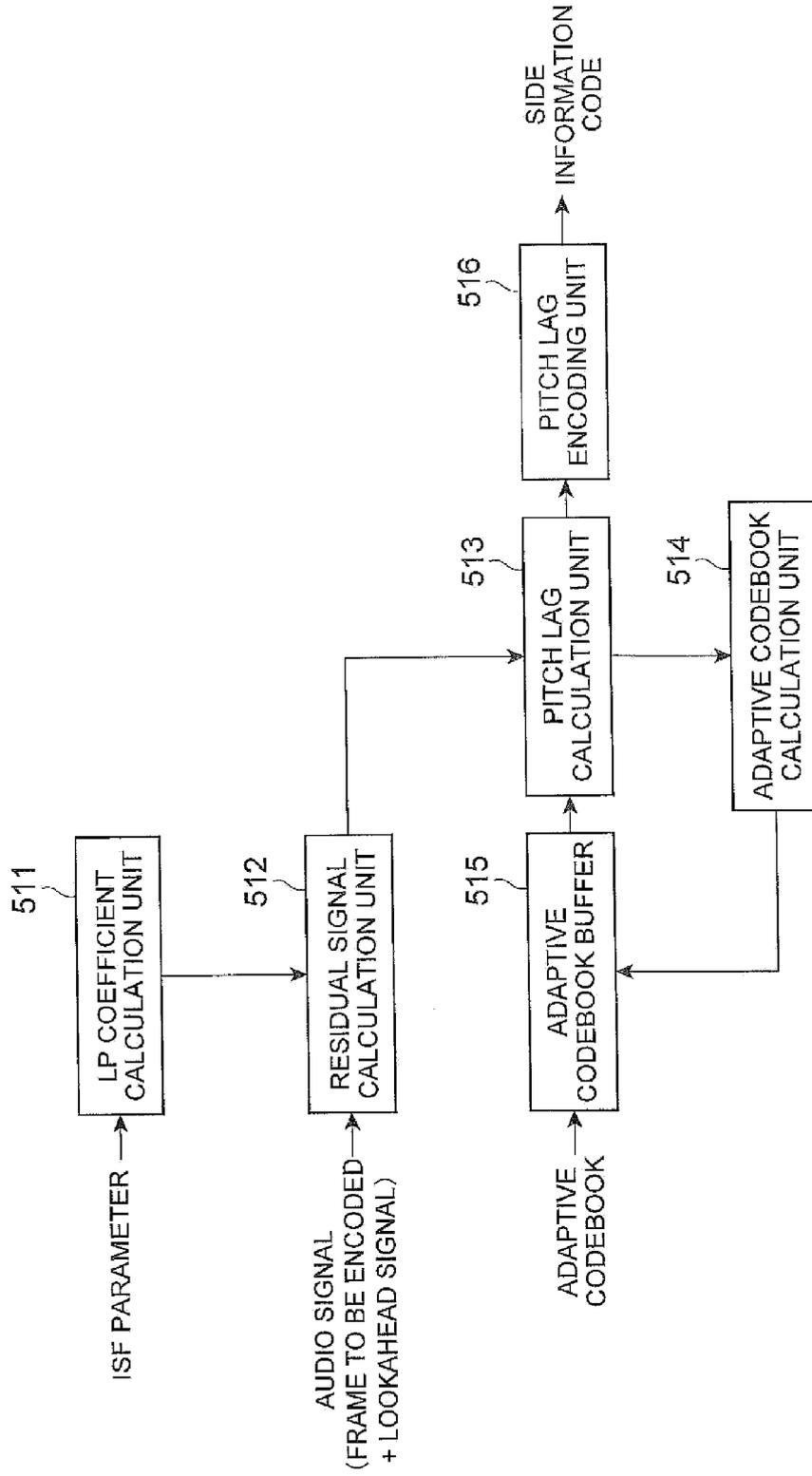
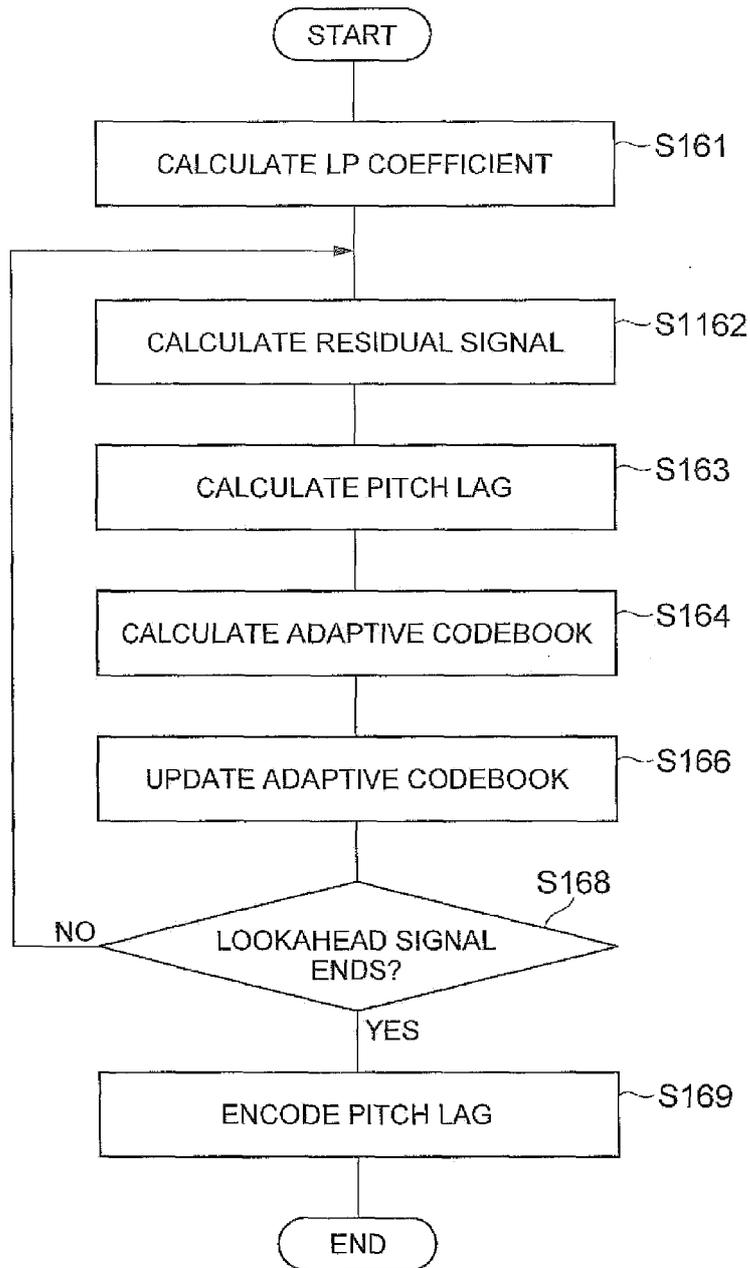
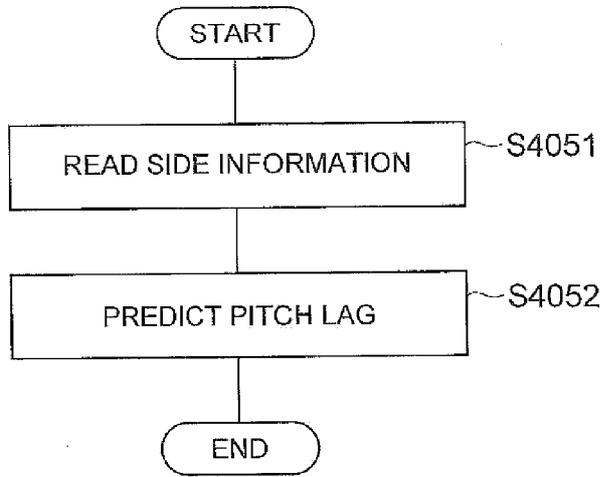


Fig.34



**Fig.35**



**Fig.36**

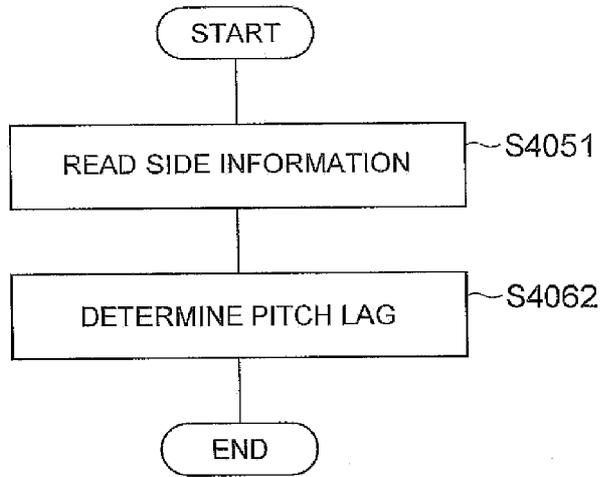


Fig.37

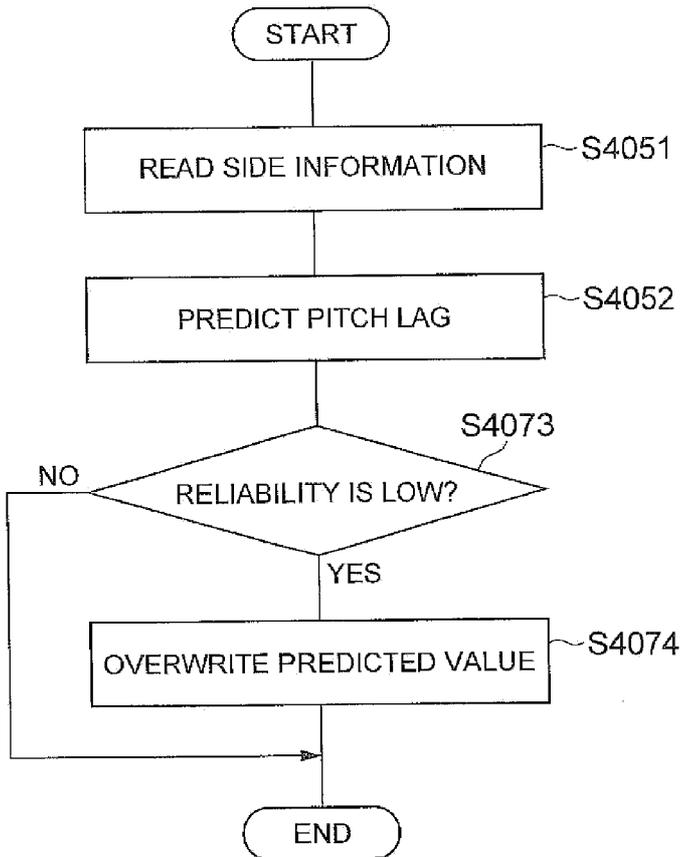


Fig.38

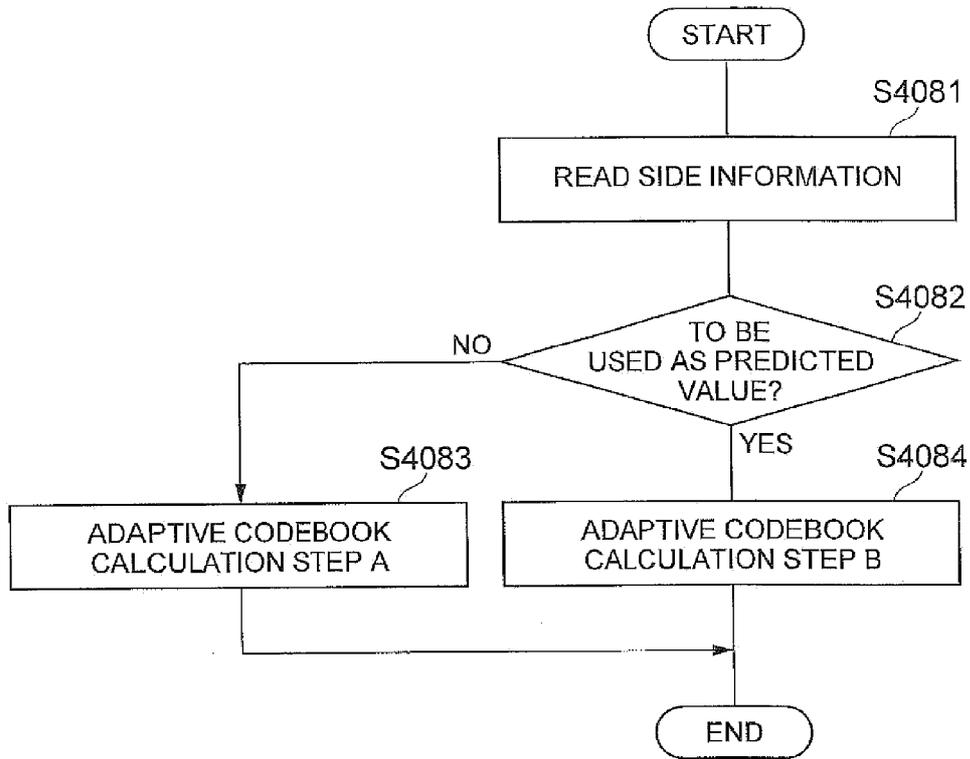


Fig.39

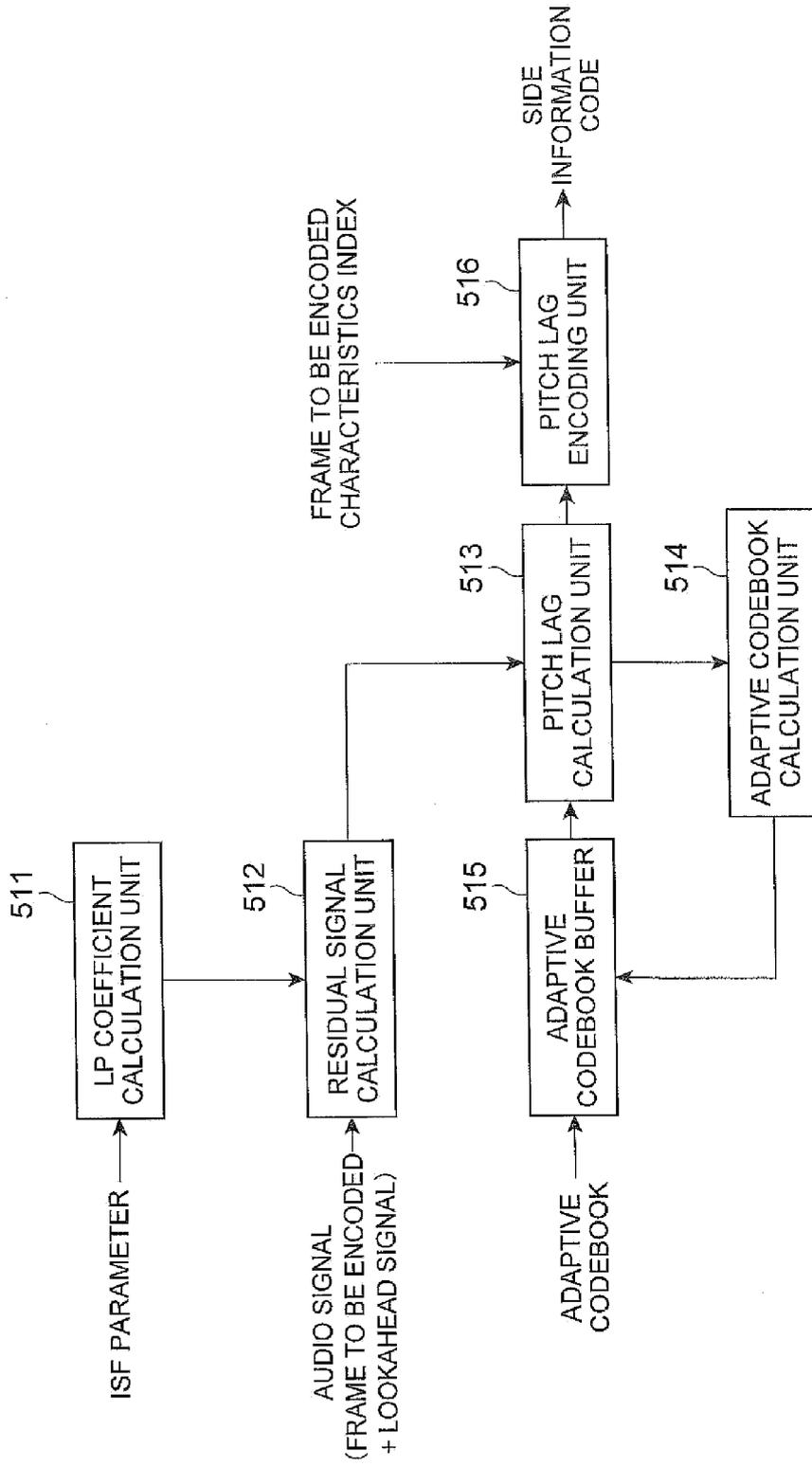


Fig.40

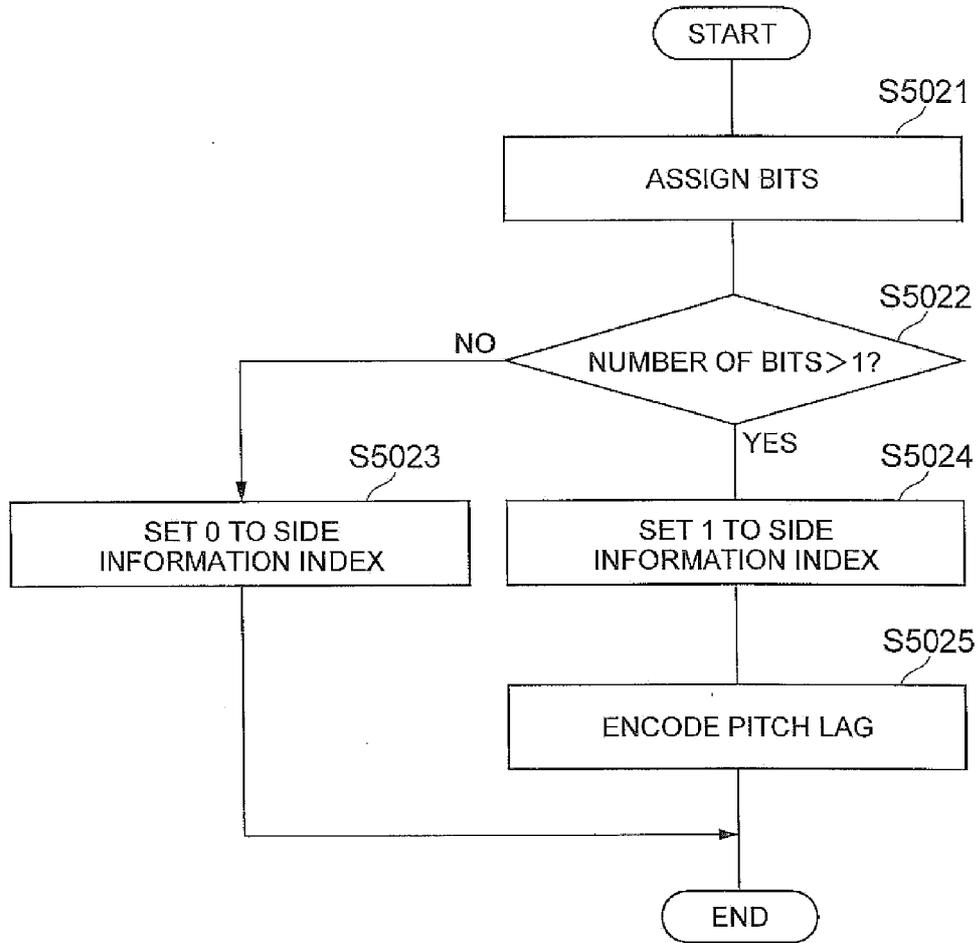


Fig.41

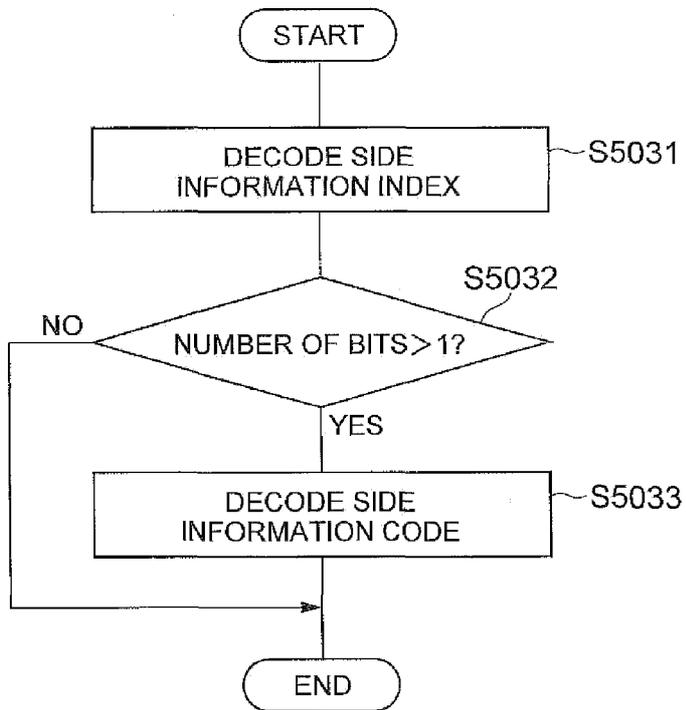


Fig.42

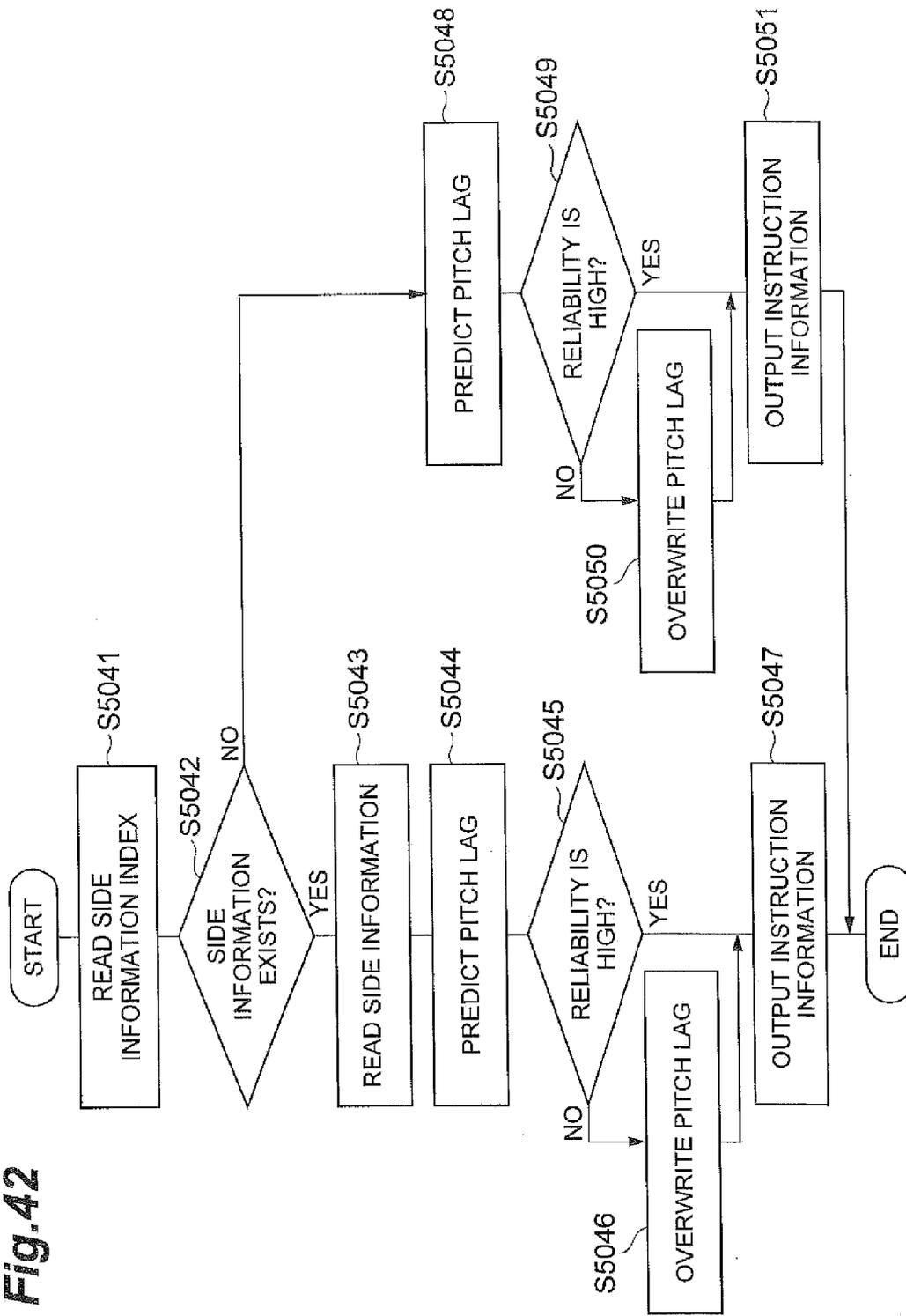


Fig.43

