



(11) **EP 3 588 485 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention  
of the grant of the patent:  
**24.03.2021 Bulletin 2021/12**

(51) Int Cl.:  
**G10H 1/00** (2006.01) **G10H 1/12** (2006.01)  
**G10H 1/36** (2006.01) **G10L 13/033** (2013.01)

(21) Application number: **19181435.9**

(22) Date of filing: **20.06.2019**

(54) **ELECTRONIC MUSICAL INSTRUMENT, ELECTRONIC MUSICAL INSTRUMENT CONTROL METHOD, AND STORAGE MEDIUM**

ELEKTRONISCHES MUSIKINSTRUMENT, VERFAHREN ZUR STEUERUNG EINES  
ELEKTRONISCHEN MUSIKINSTRUMENTS UND SPEICHERMEDIUM

INSTRUMENT DE MUSIQUE ÉLECTRONIQUE, PROCÉDÉ DE COMMANDE D'INSTRUMENT DE  
MUSIQUE ÉLECTRONIQUE ET SUPPORT D'INFORMATIONS

(84) Designated Contracting States:  
**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**

- **SETOGUCHI, Masaru**  
Tokyo, 205-8555 (JP)
- **NAKAMURA, Atsushi**  
Tokyo, 205-8555 (JP)

(30) Priority: **21.06.2018 JP 2018118057**

(74) Representative: **Plougmann Vingtoft a/s**  
**Strandvejen 70**  
**2900 Hellerup (DK)**

(43) Date of publication of application:  
**01.01.2020 Bulletin 2020/01**

(73) Proprietor: **Casio Computer Co., Ltd.**  
**Tokyo 151-8543 (JP)**

(56) References cited:  
**EP-A1- 2 270 773 JP-A- 2017 107 228**  
**US-A1- 2015 278 686**

(72) Inventors:

- **DANJYO, Makoto**  
Tokyo, 205-8555 (JP)
- **OTA, Fumiaki**  
Tokyo, 205-8555 (JP)

- **MERLIJN BLAAUW ET AL: "A Neural Parametric Singing Synthesizer Modeling Timbre and Expression from Natural Songs", APPLIED SCIENCES, vol. 7, no. 12, 18 December 2017 (2017-12-18), page 1313, XP055627719, DOI: 10.3390/app7121313**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

**EP 3 588 485 B1**

**Description**

## BACKGROUND OF THE INVENTION

## 5 Technical Field

**[0001]** The present invention relates to an electronic musical instrument that generates a singing voice in accordance with the operation of an operation element on a keyboard or the like, an electronic musical instrument control method, and a storage medium.

## 10 Background Art

**[0002]** Hitherto known electronic musical instruments output a singing voice that is synthesized using concatenative synthesis, in which fragments of recorded speech are connected together and processed (for example, see Patent Document 1) EP 2 270 773 A1 defines an apparatus for synthesizing singing voices (human voices) in accordance with score data representative of a musical score of a singing music piece.

**[0003]** "A Neural Parametric Singing Synthesizer Modeling Timbre and Expression from Natural Songs" by Merlijn Blaauw et al - 18 December 2017, applied sciences, vol. 7, no. 12, page 1313 discloses a vocoder based singing voice synthesis. The vocoder parameters are learned using neural network defining a timbre model. Feeding the vocoder with these parameters allows bypassing the direct sample generation by the deep neural network, as performed in the Wavenet model, and lowering the complexity of the synthesis.

## Related Art Documents

## 25 Patent Documents

**[0004]** Patent Document 1: Japanese Patent Application Laid-Open Publication No. H09-050287

## SUMMARY OF THE INVENTION

**[0005]** However, the method of Patent Document 1, which can be considered an extension of pulse code modulation (PCM), requires long hours of recording when being developed. Complex calculations for smoothly joining fragments of recorded speech together and adjustments so as to provide a natural-sounding singing voice are also required with this method.

**[0006]** An object of the present invention is to provide an electronic musical instrument that sings well in the singing voice of a given singer at pitches specified through the operation of operation elements by a user due to being equipped with a trained model that has learned the singing voice of the given singer.

**[0007]** Additional or separate features and advantages of the invention will be set forth in the descriptions that follow and in part will be apparent from the description, or may be learned by practice of the invention. The objectives and other advantages of the invention will be realized and attained by the structure particularly pointed out in the written description and claims thereof as well as the appended drawings.

**[0008]** To achieve these and other advantages and in accordance with the purpose of the present invention, as embodied and broadly described, in one aspect, the present disclosure provides An electronic musical instrument including: a plurality of operation elements respectively corresponding to mutually different pitch data; a memory that stores a trained acoustic model obtained by performing machine learning on training musical score data including training lyric data and training pitch data, and on training singing voice data of a singer corresponding to the training musical score data, the trained acoustic model being configured to receive lyric data and prescribed pitch data and output acoustic feature data of a singing voice of the singer in response to the received lyric data and pitch data; and at least one processor in which a first mode and a second mode are interchangeably selectable, wherein in the first mode, the at least one processor: in accordance with a user operation on an operation element in the plurality of operation elements, inputs prescribed lyric data and pitch data corresponding to the user operation of the operation element to the trained acoustic model so as to cause the trained acoustic model to output the acoustic feature data in response to the inputted prescribed lyric data and the inputted pitch data, and digitally synthesizes and outputs inferred singing voice data that infers a singing voice of the singer on the basis of at least a portion of the acoustic feature data output by the trained acoustic model in response to the inputted prescribed lyric data and the inputted pitch data, and on the basis of instrument sound waveform data that are synthesized in accordance with the pitch data corresponding to the user operation of the operation element, and wherein in the second mode, the at least one processor: in accordance with a user operation on an operation element in the plurality of operation elements, inputs prescribed lyric data and pitch data corresponding

to the user operation of the operation element to the trained acoustic model so as to cause the trained acoustic model to output the acoustic feature data in response to the inputted prescribed lyric data and the inputted pitch data, and digitally synthesizes and outputs inferred singing voice data that infers a singing voice of the singer on the basis of the acoustic feature data output by the trained acoustic model in response to the inputted prescribed lyric data and the inputted pitch data, without using instrument sound waveform data that are or could be synthesized in accordance with the pitch data corresponding to the user operation of the operation element.

**[0009]** In another aspect, the present disclosure provides a method performed by the at least one processor in the electronic musical instrument described above, the method including, via the at least one processor, each step performed by the at least one processor described above.

**[0010]** In another aspect, the present disclosure provides a non-transitory computer-readable storage medium having stored thereon a program executable by the at least one processor in the above-described electronic musical instrument, the program causing the at least one processor to perform each step performed by the at least one processor described above.

**[0011]** According to an aspect of the present invention, an electronic musical instrument can be provided that sings well in the singing voice of a given singer at pitches specified through the operation of operation elements by a user due to being equipped with a trained model that has learned the singing voice of the given singer.

**[0012]** It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory, and are intended to provide further explanation of the invention as claimed.

## BRIEF DESCRIPTION OF THE DRAWINGS

### **[0013]**

FIG. 1 is a diagram illustrating an example external view of an embodiment of an electronic keyboard instrument of the present invention.

FIG. 2 is a block diagram illustrating an example hardware configuration for an embodiment of a control system of the electronic keyboard instrument.

FIG. 3 is a block diagram illustrating an example configuration of a voice training section and a voice synthesis section.

FIG. 4 is a diagram for explaining a first embodiment of statistical voice synthesis processing.

FIG. 5 is a diagram for explaining a second embodiment of statistical voice synthesis processing.

FIG. 6 is a diagram illustrating an example data configuration in the embodiments.

FIG. 7 is a main flowchart illustrating an example of a control process for the electronic musical instrument of the embodiments.

FIGs. 8A, 8B, and 8C depict flowcharts illustrating detailed examples of initialization processing, tempo-changing processing, and song-starting processing, respectively.

FIG. 9 is a flowchart illustrating a detailed example of switch processing.

FIG. 10 is a flowchart illustrating a detailed example of automatic-performance interrupt processing.

FIG. 11 is a flowchart illustrating a detailed example of song playback processing.

## DETAILED DESCRIPTION OF EMBODIMENTS

**[0014]** Embodiments of the present invention will be described in detail below with reference to the drawings.

**[0015]** FIG. 1 is a diagram illustrating an example external view of an embodiment of an electronic keyboard instrument 100 of the present invention. The electronic keyboard instrument 100 is provided with, *inter alia*, a keyboard 101, a first switch panel 102, a second switch panel 103, and a liquid crystal display (LCD) 104. The keyboard 101 is made up of a plurality of keys serving as performance operation elements. The first switch panel 102 is used to specify various settings, such as specifying volume, setting a tempo for song playback, initiating song playback, playing back an accompaniment, and for a vocalization mode (a first mode indicating that a vocoder mode is ON, and a second mode indicating that the vocoder mode is OFF). The second switch panel 103 is used to make song and accompaniment selections, select tone color, and so on. The LCD 104 displays a musical score and lyrics during the playback of a song, and information relating to various settings. Although not illustrated in the drawings, the electronic keyboard instrument 100 is also provided with a speaker that emits musical sounds generated by playing of the electronic keyboard instrument 100. The speaker is provided at the underside, a side, the rear side, or other such location on the electronic keyboard instrument 100.

**[0016]** FIG. 2 is a diagram illustrating an example hardware configuration for an embodiment of a control system 200 in the electronic keyboard instrument 100 of FIG. 1. In the control system 200 in FIG. 2, a central processing unit (CPU) 201, a read-only memory (ROM) 202, a random-access memory (RAM) 203, a sound source large-scale integrated circuit (LSI) 204, a voice synthesis LSI 205, a key scanner 206, and an LCD controller 208 are each connected to a

system bus 209. The key scanner 206 is connected to the keyboard 101, to the first switch panel 102, and to the second switch panel 103 in FIG. 1. The LCD controller 208 is connected to the LCD 104 in FIG. 1. The CPU 201 is also connected to a timer 210 for controlling an automatic performance sequence. Musical sound output data 218 (instrument sound waveform data) output from the sound source LSI 204 is converted into an analog musical sound output signal by a D/A converter 211, and inferred singing voice data 217 output from the voice synthesis LSI 205 is converted into an analog singing voice sound output signal by a D/A converter 212. The analog musical sound output signal and the analog singing voice sound output signal are mixed by a mixer 213, and after being amplified by an amplifier 214, this mixed signal is output from an output terminal or the non-illustrated speaker. The sound source LSI 204 and the voice synthesis LSI 205 may of course be integrated into a single LSI. The musical sound output data 218 and the inferred singing voice data 217, which are digital signals, may also be converted into an analog signal by a D/A converter after being mixed together by a mixer.

**[0017]** While using the RAM 203 as working memory, the CPU 201 executes a control program stored in the ROM 202 and thereby controls the operation of the electronic keyboard instrument 100 in FIG. 1. In addition to the aforementioned control program and various kinds of permanent data, the ROM 202 stores musical piece data including lyric data and accompaniment data.

**[0018]** The ROM 202 (memory) is also pre-stored with melody pitch data (215d) indicating operation elements that a user is to operate, singing voice output timing data (215c) indicating output timings at which respective singing voices for pitches indicated by the melody pitch data (215d) are to be output, and lyric data (215a) corresponding to the melody pitch data (215d).

**[0019]** The CPU 201 is provided with the timer 210 used in the present embodiment. The timer 210, for example, counts the progression of automatic performance in the electronic keyboard instrument 100.

**[0020]** Following a sound generation control instruction from the CPU 201, the sound source LSI 204 reads musical sound waveform data from a non-illustrated waveform ROM, for example, and outputs the musical sound waveform data to the D/A converter 211. The sound source LSI 204 is capable of 256-voice polyphony.

**[0021]** When the voice synthesis LSI 205 is given, as singing voice data 215, lyric data 215a and either pitch data 215b or melody pitch data 215d by the CPU 201, the voice synthesis LSI 205 synthesizes voice data for a corresponding singing voice and outputs this voice data to the D/A converter 212.

**[0022]** The lyric data 215a and the melody pitch data 215d are pre-stored in the ROM 202. Either the melody pitch data 215d pre-stored in the ROM 202 or pitch data 215b for a note number obtained in real time due to a user key press operation is input to the voice synthesis LSI 205 as pitch data.

**[0023]** In other words, when there is a user key press operation at a prescribed timing, an inferred singing voice is produced at a pitch corresponding to the key on which there was a key press operation, and when there is no user key press operation at a prescribed timing, an inferred singing voice is produced at a pitch indicated by the melody pitch data 215d stored in the ROM 202.

**[0024]** When the vocoder mode has been turned ON (when the first mode has been specified) using the first switch panel 102, musical sound output data outputted from designated sound generation channels (single or plural channels) of the sound source LSI 204 are inputted to the voice synthesis LSI 205 as instrument sound waveform data 220.

**[0025]** The key scanner 206 regularly scans the pressed/released states of the keys on the keyboard 101 and the operation states of the switches on the first switch panel 102 and the second switch panel 103 in FIG. 1, and sends interrupts to the CPU 201 to communicate any state changes.

**[0026]** The LCD controller 609 is an integrated circuit (IC) that controls the display state of the LCD 505.

**[0027]** FIG. 3 is a block diagram illustrating an example configuration of a voice synthesis section, an acoustic effect application section, and a voice training section of the present embodiment. The voice synthesis section 302 and the acoustic effect application section 322 are built into the electronic keyboard instrument 100 as part of functionality performed by the voice synthesis LSI 205 in FIG. 2.

**[0028]** Along with lyric data 215a, the voice synthesis section 302 is input with pitch data 215b instructed by the CPU 201 on the basis of a key press on the keyboard 101 in FIG. 1 via the key scanner 206 in FIG. 2. With this, the voice synthesis section 302 synthesizes and outputs output data 321. If no key on the keyboard 101 is pressed and pitch data 215b is not instructed by the CPU 201, melody pitch data 215d stored in memory is input to the voice synthesis section 302 in place of the pitch data 215b. A trained acoustic model 306 takes this data and outputs spectral data 318 and sound source data 319.

**[0029]** In the first mode, the voice synthesis section 302 outputs inferred singing voice data 217 for which the singing voice of a given singer has been inferred on the basis of the spectral data 318 output from the trained acoustic model 306 and on the instrument sound waveform data 220 output by the sound source LSI 204, and not on the basis of the sound source data 319. Also, even when a user does not press a key at a prescribed timing, a corresponding singing voice is produced at an output timing indicated by singing voice output timing data 215c stored in the ROM 202.

**[0030]** In the second mode, the voice synthesis section 302 outputs inferred singing voice data 217 for which the singing voice of a given singer has been inferred on the basis of the spectral data 318 and the sound source data 319

output from the trained acoustic model 306. Also, even when a user does not press a key at a prescribed timing, a corresponding singing voice is produced at an output timing indicated by singing voice output timing data 215c stored in the ROM 202.

**[0031]** It is important to note that the electronic musical instrument constituting one embodiment of the present invention is equipped with a first mode and a second mode, and that the first mode and the second mode can be switched 320 between by user operation. It is thereby possible to switch between the first mode (a polyphonic mode) and the second mode (a monophonic mode), for example, in accordance with the song performed by a user, as appropriate.

**[0032]** Thus, in this aspect of the present invention, in the vocoder mode, the electronic musical instrument 100 uses the instrument sound waveform data 220 output by the sound source LSI 204 instead of (in other words, without using) sound source data 319 output by the trained acoustic model 306. The instrument sound waveform data 220 are instrument sound waveform data having one or more pitches specified by the user by operating the keyboard 101 (or specified by the melody pitch data 215d stored in the ROM 202 if there is no keyboard operation by the user). The instrument sounds for the waveform data that are synthesized here preferably include, but not limited to, sounds of brass instruments, strings instruments, organ, sound of animals, for example. The instrument sound may be the sound of just one of these instrumental sounds selected by a user operation of the first switch panel 102. Through diligent research, the present inventors have discovered that these listed instrument sounds are particularly effective when combined with the spectral data 318 that carry characteristics of a human singing voice.

**[0033]** In this embodiment of the present invention, in the vocoder mode, if the user presses multiple keys at the keyboard 101 at the same time (specifying a chord, for example), a synthesized singing voice having certain characteristics of a human singing voice having the corresponding multiple pitches is output (i.e., polyphonic output). That is, in the vocoder mode of this embodiment, for each of the pitches specified in the chord, the waveform data of the music instrument having the corresponding pitch is modified by the spectral data 318 (formant information) outputted from the acoustic model 306, thereby adding the vocal characteristics of the singer with respect to which the acoustic model 306 has been trained to the inferred singing voice data 217, which is polyphonically output. This aspect is advantageous because when the user presses multiple keys at the same time, the polyphonic singing voice corresponding to the specified multiple pitches are outputted.

**[0034]** In conventional vocoders, users needed to sing while operating the keyboard; a microphone to pick up the user's singing voice was necessary. In the vocoder mode of the present embodiment of the present invention, the user need not sing, and a microphone is not needed. Also, as noted above, in the vocoder mode of this embodiment, with respect to the acoustic feature data 317 (explained below) including spectral data 318 and sound source data 319, only the spectral data 318 is used in synthesizing the inferred singing voice data.

**[0035]** According to this embodiment of the present invention, the user only needs to select On and Off of the vocoder mode in order to switch voice sound generation modes. Therefore, the electronic musical instrument of the present embodiment is more advantageous than electronic musical instruments having only one of these modes.

**[0036]** The acoustic effect application section 322 is input with effect application instruction data 215e, as a result of which the acoustic effect application section 320 applies an acoustic effect such as a vibrato effect, a tremolo effect, or a wah effect to the output data 321 output by the voice synthesis section 302.

**[0037]** Effect application instruction data 215e is input to the acoustic effect application section 320 in accordance with the pressing of a second key (for example, a black key) within a prescribed range from a first key that has been pressed by a user (for example, within one octave). The greater the difference in pitch between the first key and the second key, the greater the acoustic effect that is applied by the acoustic effect application section 320.

**[0038]** As illustrated in FIG. 3, the voice training section 301 may, for example, be implemented as part of functionality performed by a separate server computer 300 provided outside the electronic keyboard instrument 100 in FIG. 1. Alternatively, although not illustrated in FIG. 3, if the voice synthesis LSI 205 in FIG. 2 has spare processing capacity, the voice training section 301 may be built into the electronic keyboard instrument 100 and implemented as part of functionality performed by the voice synthesis LSI 205.

**[0039]** The voice training section 301 and the voice synthesis section 302 in FIG. 2 are implemented on the basis of, for example, the "statistical parametric speech synthesis based on deep learning" techniques described in Non-Patent Document 1, cited below.

(Non-Patent Document 1)

**[0040]** Kei Hashimoto and Shinji Takaki, "Statistical parametric speech synthesis based on deep learning", Journal of the Acoustical Society of Japan, vol. 73, no. 1 (2017), pp. 55-62

**[0041]** The voice training section 301 in FIG. 2, which is functionality performed by the external server computer 300 illustrated in FIG. 3, for example, includes a training text analysis unit 303, a training acoustic feature extraction unit 304, and a model training unit 305.

**[0042]** The voice training section 301, for example, uses voice sounds that were recorded when a given singer sang

a plurality of songs in an appropriate genre as training singing voice data for a given singer 312. Lyric text (training lyric data 311a) for each song is also prepared as training musical score data 311.

[0043] The training text analysis unit 303 is input with training musical score data 311, including lyric text (training lyric data 311a) and musical note data (training pitch data 311b), and the training text analysis unit 303 analyzes this data. The training text analysis unit 303 accordingly estimates and outputs a training linguistic feature sequence 313, which is a discrete numerical sequence expressing, *inter alia*, phonemes and pitches corresponding to the training musical score data 311.

[0044] In addition to this input of training musical score data 311, the training acoustic feature extraction unit 304 receives and analyzes training singing voice data for a given singer 312 that has been recorded via a microphone or the like when a given singer sang (for approximately two to three hours, for example) lyric text corresponding to the training musical score data 311. The training acoustic feature extraction unit 304 accordingly extracts and outputs a training acoustic feature sequence 314 representing phonetic features corresponding to the training singing voice data for a given singer 312.

[0045] As described in Non-Patent Document 1, in accordance with Equation (1) below, the model training unit 305 uses machine learning to estimate an acoustic model  $\hat{\lambda}$  with which the probability ( $P(o|l, \lambda)$ ) that a training acoustic feature sequence 314 ( $o$ ) will be generated given a training linguistic feature sequence 313 ( $l$ ) and an acoustic model ( $\lambda$ ) is maximized. In other words, a relationship between a linguistic feature sequence (text) and an acoustic feature sequence (voice sounds) is expressed using a statistical model, which here is referred to as an acoustic model.

$$\hat{\lambda} = \arg \max_{\lambda} P(o|l, \lambda) \quad (1)$$

[0046] Here, *arg max* denotes a computation that calculates the value of the argument underneath *arg max* that yields the greatest value for the function to the right of *arg max*.

[0047] The model training unit 305 outputs, as training result 315, model parameters expressing the acoustic model  $\hat{\lambda}$  that have been calculated using Equation (1) through the employ of machine learning.

[0048] As illustrated in FIG. 3, the training result 315 (model parameters) may, for example, be stored in the ROM 202 of the control system in FIG. 2 for the electronic keyboard instrument 100 in FIG. 1 when the electronic keyboard instrument 100 is shipped from the factory, and may be loaded into the trained acoustic model 306, described later, in the voice synthesis LSI 205 from the ROM 202 in FIG. 2 when the electronic keyboard instrument 100 is powered on. Alternatively, as illustrated in FIG. 3, as a result of user operation of the second switch panel 103 on the electronic keyboard instrument 100, the training result 315 may, for example, be downloaded from the Internet, a universal serial bus (USB) cable, or other network via a non-illustrated network interface 219 and into the trained acoustic model 306, described later, in the voice synthesis LSI 205.

[0049] The voice synthesis section 302, which is functionality performed by the voice synthesis LSI 205, includes a text analysis unit 307, the trained acoustic model 306, and a vocalization model unit 308. The voice synthesis section 302 performs statistical voice synthesis processing in which output data 321, corresponding to singing voice data 215 including lyric text, is synthesized by making predictions using the statistical model, referred to herein as an acoustic model, set in the trained acoustic model 306.

[0050] As a result of a performance by a user made in concert with an automatic performance, the text analysis unit 307 is input with singing voice data 215, which includes information relating to phonemes, pitches, and the like for lyrics specified by the CPU 201 in FIG. 2, and the text analysis unit 307 analyzes this data. The text analysis unit 307 performs this analysis and outputs a linguistic feature sequence 316 expressing, *inter alia*, phonemes, parts of speech, and words corresponding to the singing voice data 215.

[0051] As described in Non-Patent Document 1, the trained acoustic model 306 is input with the linguistic feature sequence 316, and using this, the trained acoustic model 306 estimates and outputs an acoustic feature sequence 317 (acoustic feature data 317) corresponding thereto. In other words, in accordance with Equation (2) below, the trained acoustic model 306 estimates a value ( $\hat{o}$ ) for an acoustic feature sequence 317 at which the probability ( $P(o|l, \hat{\lambda})$ ) that an acoustic feature sequence 317 ( $o$ ) will be generated based on a linguistic feature sequence 316 ( $l$ ) input from the text analysis unit 307 and an acoustic model  $\hat{\lambda}$  set using the training result 315 of machine learning performed in the model training unit 305 is maximized.

$$\hat{o} = \arg \max_o P(o|l, \hat{\lambda}) \quad (2)$$

[0052] The vocalization model unit 308 is input with the acoustic feature sequence 317. With this, the vocalization model unit 308 generates output data 321 corresponding to the singing voice data 215 including lyric text specified by the CPU 201. An acoustic effect is applied to the output data 321 in the acoustic effect application section 322, described

later, and the output data 321 is converted into the final inferred singing voice data 217. This inferred singing voice data 217 is output from the D/A converter 212, goes through the mixer 213 and the amplifier 214 in FIG. 2, and is emitted from the non-illustrated speaker.

**[0053]** The acoustic features expressed by the training acoustic feature sequence 314 and the acoustic feature sequence 317 include spectral information that models the vocal tract of a person, and sound source information that models the vocal chords of a person. A mel-cepstrum, line spectral pairs (LSP), or the like may be employed for the spectral information. A power value and a fundamental frequency (F0) indicating the pitch frequency of the voice of a person may be employed for the sound source information. The vocalization model unit 308 includes a sound source generator 309 and a synthesis filter 310. The sound source generator 309 models the vocal cords of a person. When a user turns the vocoder mode OFF using the first switch panel 102 in FIG. 1 (when the second mode is specified), a vocoder mode switch 320 connects the sound source generator 309 to the synthesis filter 310. As a result, the sound source generator 309 is sequentially input with a sound source data 319 sequence from the trained acoustic model 306. Thereby, the sound source generator 309, for example, generates a sound source signal that is made up of a pulse train (for voiced phonemes) that periodically repeats with a fundamental frequency (F0) and power value contained in the sound source data 319, that is made up of white noise (for unvoiced phonemes) with a power value contained in the sound source data 319, or that is made up of a signal in which a pulse train and white noise are mixed together. This sound source signal is input to the synthesis filter 310 via the vocoder mode switch 320. However, when the user turns the vocoder mode ON using the first switch panel 102 in FIG. 1 (when the first mode is specified by operation of a switching operation element), the vocoder mode switch 320 causes instrument sound waveform data 220 in the designated sound generation channels (single or plural channels) of the sound source LSI 204 in FIG. 2 to be input to the synthesis filter 310. The synthesis filter 310 models the vocal tract of a person. The synthesis filter 310 forms a digital filter that models the vocal tract on the basis of a spectral data 318 sequence sequentially input thereto from the trained acoustic model 306, and using either the sound source signal input from the sound source generator 309 or the instrument sound waveform data 220 from the designated sound generation channels (single or plural channels) of the sound source LSI 204 as an excitation signal, generates and outputs inferred singing voice data 217 in the form of a digital signal. When the vocoder mode is ON, the instrument sound waveform data 220 input from the sound source LSI 204 is polyphonic data corresponding to the designated sound generation channel(s).

**[0054]** As described above, when the vocoder mode is turned OFF by a user using the first switch panel 102 in FIG. 1 (the second mode is specified by operation of the switching operation element), a sound source signal generated by the sound source generator 309 on the basis of sound source data 319 input from the trained acoustic model 306 is input to the synthesis filter 310 operating on the basis of spectral data 318 input from the trained acoustic model 306, and output data 321 is output from the synthesis filter 310. Output data 321 generated and output in this manner has been entirely modeled by the trained acoustic model 306, and thus results in a singing voice that is both natural-sounding and very faithful to the singing voice of the singer.

**[0055]** However, when the vocoder mode is turned ON (the first mode) by a user using the first switch panel 102 in FIG. 1, instrument sound waveform data 220 generated and output by the sound source LSI 204 based on the playing of the user on the keyboard 101 (FIG. 1) is input to the synthesis filter 310 operating on the basis of spectral data 318 input from the trained acoustic model 306, and output data 321 is output from the synthesis filter 310. Output data 321 generated and output in this manner uses instrument sounds generated by the sound source LSI 204 as a sound source signal. For this reason, although some faithfulness is lost when compared to the singing voice of the singer, the essence of instrument sounds set in the sound source LSI 204 as well as the vocal characteristics of the singing voice of the singer come through clearly, thus allowing effective output data 321 to be output. An effect in which a plurality of singing voices seem to be in harmony can also be achieved owing to polyphonic operation being possible in the vocoder mode.

**[0056]** The sound source LSI 204 may be operated such that, for example, at the same time that the output from a plurality of designated sound generation channels is supplied to the voice synthesis LSI 205 as instrument sound waveform data 220, the output of another channel(s) is output as normal musical sound output data 218. Operation is thus possible in which singing voices for a melody are vocalized by the voice synthesis LSI 205 at the same time that accompaniment sounds are produced as normal instrument sounds or instrument sounds for a melody line are produced.

**[0057]** The instrument sound waveform data 220 input to the synthesis filter 310 in the vocoder mode may be any kind of signal, but in terms of qualities as a sound source signal, instrument sounds that have many harmonic components and can be sustained for long durations, such as, for example, brass sounds, string sounds, and organ sounds, are preferable. Of course, a very amusing effect may be obtained even when, to achieve a greater effect, an instrument sound that does not remotely adhere to this standard, for example an instrument sound that sounds like an animal cry, is used. As one specific example, data obtained by sampling the cry of a pet dog, for example, is input to the synthesis filter 310 as an instrument sound. Sound is then produced from the speaker on the basis of inferred singing voice data 217 output from the synthesis filter 310 and through the acoustic effect application section 322. This results in a very amusing effect in which it sounds as if the pet dog were singing the lyrics.

**[0058]** A user can select an instrument sound to be used from among a plurality of instrument sounds by operating

an input operation element (selection operation element) on the switch panel 102 or the like.

**[0059]** With the electronic musical instrument constituting one embodiment of the present invention, a user can easily switch between the first mode and the second mode merely by switching the vocoder mode ON (the first mode)/OFF (the second mode) in an operation on the first switch panel 102 in FIG. 1. In the first mode, singing voice data for which the way a singer sings has been inferred is output. In the second mode, a plurality of pieces of singing voice data reflecting characteristics of the way a singer sings are output. Further, a singing voice can be easily generated and output in either mode of the electronic musical instrument constituting one embodiment of the present invention. In other words, because it is possible to easily generate and output a variety of singing voices with the present invention, users are able to enjoy performances more.

**[0060]** The sampling frequency of the training singing voice data for a given singer 312 is, for example, 16 kHz (kilohertz). When a mel-cepstrum parameter obtained through mel-cepstrum analysis, for example, is employed for a spectral parameter contained in the training acoustic feature sequence 314 and the acoustic feature sequence 317, the frame update period is, for example, 5 msec (milliseconds). In addition, when mel-cepstrum analysis is performed, the length of the analysis window is 25 msec, and the window function is a twenty-fourth-order Blackman window function.

**[0061]** An acoustic effect such as a vibrato effect, a tremolo effect, or a wah effect is applied to the output data 321 output from the voice synthesis section 302 by the acoustic effect application section 322 in the voice synthesis LSI 205.

**[0062]** A "vibrato effect" refers to an effect whereby, when a note in a song is drawn out, the pitch level is periodically varied by a prescribed amount (depth).

**[0063]** A "tremolo effect" refers to an effect whereby one or more notes are rapidly repeated.

**[0064]** A "wah effect" is an effect whereby the peak-gain frequency of a bandpass filter is moved so as to yield a sound resembling a voice saying "wah-wah".

**[0065]** When a user performs an operation whereby a second key (second operation element) on the keyboard 101 (FIG. 1) is repeatedly struck while a first key (first operation element) on the keyboard 101 for instructing a singing voice sound is causing output data 321 to be continuously output (while the first key is being pressed), an acoustic effect that has been preselected from among a vibrato effect, a tremolo effect, or a wah effect using the first switch panel 102 (FIG. 1) can be applied by the acoustic effect application section 322.

**[0066]** In this case, the user is able to vary the degree of the pitch effect in the acoustic effect application section 322 by, with respect to the pitch of the first key specifying a singing voice, specifying the second key that is repeatedly struck such that the difference in pitch between the second key and the first key is a desired difference. For example, the degree of the pitch effect can be made to vary such that the depth of the acoustic effect is set to a maximum value when the difference in pitch between the second key and the first key is one octave and such that the degree of the acoustic effect is weaker the lesser the difference in pitch.

**[0067]** The second key on the keyboard 101 that is repeatedly struck may be a white key. However, if the second key is a black key, for example, the second key is less liable to interfere with a performance operation on the first key for specifying the pitch of a singing voice sound.

**[0068]** In the present embodiment, it is thus possible to apply various additional acoustic effects in the acoustic effect application section 322 to output data 321 that is output from the voice synthesis section 302 to generate the final inferred singing voice data 217.

**[0069]** It should be noted that the application of an acoustic effect ends when no key presses on the second key have been detected for a set time (for example, several hundred milliseconds).

**[0070]** As another example, such an acoustic effect may be applied by just one press of the second key while the first key is being pressed, in other words, without repeatedly striking the second key as above. In this case too, the depth of the acoustic effect may change in accordance with the difference in pitch between the first key and the second key. The acoustic effect may be also applied while the second key is being pressed, and application of the acoustic effect ended in accordance with the detection of release of the second key.

**[0071]** As yet another example, such an acoustic effect may be applied even when the first key is released after the pressing the second key while the first key was being pressed. This kind of pitch effect may also be applied upon the detection of a "trill", whereby the first key and the second key are repeatedly struck in an alternating manner.

**[0072]** In the present specification, as a matter of convenience, the musical technique whereby such acoustic effects are applied is sometimes called "what is referred to as a legato playing style".

**[0073]** Next, a first embodiment of statistical voice synthesis processing performed by the voice training section 301 and the voice synthesis section 302 in FIG. 3 will be described. In the first embodiment of statistical voice synthesis processing, hidden Markov models (HMMs), described in Non-Patent Document 1 above and Non-Patent Document 2 below, are used for acoustic models expressed by the training result 315 (model parameters) set in the trained acoustic model 306.



(Non-Patent Document 2)

**[0074]** Shinji Sako, Keijiro Saino, Yoshihiko Nankaku, Keiichi Tokuda, and Tadashi Kitamura, "A trainable singing voice synthesis system capable of representing personal characteristics and singing styles", Information Processing Society of Japan (IPSJ) Technical Report, Music and Computer (MUS) 2008 (12 (2008-MUS-074)), pp. 39-44, 2008-02-08

**[0075]** In the first embodiment of statistical voice synthesis processing, when a user vocalizes lyrics in accordance with a given melody, HMM acoustic models are trained on how singing voice feature parameters, such as vibration of the vocal cords and vocal tract characteristics, change over time during vocalization. More specifically, the HMM acoustic models model, on a phoneme basis, spectrum and fundamental frequency (and the temporal structures thereof) obtained from the training singing voice data.

**[0076]** First, processing by the voice training section 301 in FIG. 3 in which HMM acoustic models are employed will be described. As described in Non-Patent Document 2, the model training unit 305 in the voice training section 301 is input with a training linguistic feature sequence 313 output by the training text analysis unit 303 and a training acoustic feature sequence 314 output by the training acoustic feature extraction unit 304, and therewith trains maximum likelihood HMM acoustic models on the basis of Equation (1) above. The likelihood function for the HMM acoustic models is expressed by Equation (3) below.

$$\begin{aligned}
 P(o|l, \lambda) \\
 &= \sum_q P(o|q, \lambda) P(q|l, \lambda) \\
 &= \sum_q \prod_{t=1}^T P(o_t|q_t, \lambda) P(q_t|q_{t-1}, l, \lambda) \\
 &= \sum_q \prod_{t=1}^T \mathcal{N}(o_t|\mu_{q_t}, \Sigma_{q_t}) a_{q_{t-1}q_t} \quad (3)
 \end{aligned}$$

**[0077]** Here,  $o_t$  represents an acoustic feature in frame  $t$ ,  $T$  represents the number of frames,  $q = (q_1, \dots, q_T)$  represents the state sequence of a HMM acoustic model, and  $q_t$  represents the state number of the HMM acoustic model in frame

$t$ . Further,  $a_{q_{t-1}q_t}$  represents the state transition probability from state  $q_{t-1}$  to state  $q_t$  and  $\mathcal{N}(o_t|\mu_{q_t}, \Sigma_{q_t})$  is the normal distribution of a mean vector  $\mu_{q_t}$  and a covariance matrix  $\Sigma_{q_t}$  and represents an output probability distribution for state  $q_t$ . An expectation-maximization (EM) algorithm is used to efficiently train HMM acoustic models based on maximum likelihood criterion.

**[0078]** The spectral parameters of singing voice sounds can be modeled using continuous HMMs. However, because logarithmic fundamental frequency (F0) is a variable dimension time series signal that takes on a continuous value in voiced segments and is not defined in unvoiced segments, fundamental frequency (F0) cannot be directly modeled by regular continuous HMMs or discrete HMMs. Multi-space probability distribution HMMs (MSD-HMMs), which are HMMs based on a multi-space probability distribution compatible with variable dimensionality, are thus used to simultaneously model mel-cepstrums (spectral parameters), voiced sounds having a logarithmic fundamental frequency (F0), and unvoiced sounds as multidimensional Gaussian distributions, Gaussian distributions in one-dimensional space, and Gaussian distributions in zero-dimensional space, respectively.

**[0079]** As for the features of phonemes making up a singing voice, it is known that even for identical phonemes, acoustic features may vary due to being influenced by various factors. For example, the spectrum and logarithmic fundamental frequency (F0) of a phoneme, which is a basic phonological unit, may change depending on, for example, singing style, tempo, or on preceding/subsequent lyrics and pitches. Factors such as these that exert influence on acoustic features are called "context". In the first embodiment of statistical voice synthesis processing, HMM acoustic models that take context into account (context-dependent models) can be employed in order to accurately model acoustic features in voice sounds. Specifically, the training text analysis unit 303 may output a training linguistic feature sequence 313 that takes into account not only phonemes and pitch on a frame-by-frame basis, but also factors such as preceding and subsequent phonemes, accent and vibrato immediately prior to, at, and immediately after each position, and so on. In order to make dealing with combinations of context more efficient, decision tree based context clustering may be employed. Context clustering is a technique in which a binary tree is used to divide a set of HMM acoustic models into a tree structure, whereby HMM acoustic models are grouped into clusters having similar combinations of context. Each node within a tree is associated with a bifurcating question such as "Is the preceding phoneme /a/?" that distinguishes context, and each leaf node is associated with a training result 315 (model parameters) corresponding to a particular HMM acoustic model. For any combination of contexts, by traversing the tree in accordance with the questions at the

nodes, one of the leaf nodes can be reached and the training result 315 (model parameters) corresponding to that leaf node selected. By selecting an appropriate decision tree structure, highly accurate and highly generalized HMM acoustic models (context-dependent models) can be estimated.

**[0080]** FIG. 4 is a diagram for explaining HMM decision trees in the first embodiment of statistical voice synthesis processing. States for each context-dependent phoneme are, for example, associated with a HMM made up of three states 401 (#1, #2, and #3) illustrated at (a) in FIG. 4. The arrows coming in and out of each state illustrate state transitions. For example, state 401 (#1) models the beginning of a phoneme. Further, state 401 (#2), for example, models the middle of the phoneme. Finally, state 401 (#3), for example, models the end of the phoneme.

**[0081]** The duration of states 401 #1 to #3 indicated by the HMM at (a) in FIG. 4, which depends on phoneme length, is determined using the state duration model at (b) in FIG. 4. As a result of training, the model training unit 305 in FIG. 3 generates a state duration decision tree 402 for determining state duration from a training linguistic feature sequence 313 corresponding to context for a large number of phonemes relating to state duration extracted from training musical score data 311 in FIG. 3 by the training text analysis unit 303 in FIG. 3, and this state duration decision tree 402 is set as a training result 315 in the trained acoustic model 306 in the voice synthesis section 302.

**[0082]** As a result of training, the model training unit 305 in FIG. 3 also, for example, generates a mel-cepstrum parameter decision tree 403 for determining mel-cepstrum parameters from a training acoustic feature sequence 314 corresponding to a large number of phonemes relating to mel-cepstrum parameters extracted from training singing voice data for a given singer 312 in FIG. 3 by the training acoustic feature extraction unit 304 in FIG. 3, and this mel-cepstrum parameter decision tree 403 is set as the training result 315 in the trained acoustic model 306 in the voice synthesis section 302.

**[0083]** As a result of training, the model training unit 305 in FIG. 3 also, for example, generates a logarithmic fundamental frequency decision tree 404 for determining logarithmic fundamental frequency (F0) from a training acoustic feature sequence 314 corresponding to a large number of phonemes relating to logarithmic fundamental frequency (F0) extracted from training singing voice data for a given singer 312 in FIG. 3 by the training acoustic feature extraction unit 304 in FIG. 3, and sets this logarithmic fundamental frequency decision tree 404 is set as the training result 315 in the trained acoustic model 306 in the voice synthesis section 302. It should be noted that as described above, voiced segments having a logarithmic fundamental frequency (F0) and unvoiced segments are respectively modeled as one-dimensional and zero-dimensional Gaussian distributions using MSD-HMMs compatible with variable dimensionality to generate the logarithmic fundamental frequency decision tree 404.

**[0084]** Moreover, as a result of training, the model training unit 305 in FIG. 3 may also generate a decision tree for determining context such as accent and vibrato on pitches from a training linguistic feature sequence 313 corresponding to context for a large number of phonemes relating to state duration extracted from training musical score data 311 in FIG. 3 by the training text analysis unit 303 in FIG. 3, and set this decision tree as the training result 315 in the trained acoustic model 306 in the voice synthesis section 302.

**[0085]** Next, processing by the voice synthesis section 302 in FIG. 3 in which HMM acoustic models are employed will be described. The trained acoustic model 306 is input with a linguistic feature sequence 316 output by the text analysis unit 307 relating to phonemes in lyrics, pitch, and other context. For each context, the trained acoustic model 306 references the decision trees 402, 403, 404, etc., illustrated in FIG. 4, concatenates the HMMs, and then predicts the acoustic feature sequence 317 (spectral data 318 and sound source data 319) with the greatest probability of being output from the concatenated HMMs.

**[0086]** As described in the above-referenced Non-Patent Documents, in accordance with Equation (2), the trained acoustic model 306 estimates a value ( $\hat{o}$ ) for an acoustic feature sequence 317 at which the probability ( $P(o|l, \hat{\lambda})$ ) that an acoustic feature sequence 317 ( $o$ ) will be generated based on a linguistic feature sequence 316 ( $l$ ) input from the text analysis unit 307 and an acoustic model  $\hat{\lambda}$  set using the training result 315 of machine learning performed in the

$$\hat{q} = \arg \max_q P(q|l, \hat{\lambda})$$

model training unit 305 is maximized. Using the state sequence  $\hat{q}$  estimated by the state duration model at (b) in FIG. 4, Equation (2) is approximated as in Equation (4) below.

$$\begin{aligned}
\hat{o} &= \arg \max_o \sum_q P(o|q, \hat{l}) P(q|\hat{l}, \hat{l}) \\
&\approx \arg \max_o P(o|\hat{q}, \hat{l}) \\
&= \arg \max_o \mathcal{N}(o|\mu_{\hat{q}}, \Sigma_{\hat{q}}) \\
&= \mu_{\hat{q}}
\end{aligned} \tag{4}$$

[0087] Here,

$$\mu_{\hat{q}} = [\mu_{\hat{q}_1}^T, \dots, \mu_{\hat{q}_T}^T]^T$$

$$\Sigma_{\hat{q}} = \text{diag}[\Sigma_{\hat{q}_1}, \dots, \Sigma_{\hat{q}_T}],$$

and  $\mu_{\hat{q}_t}$  and  $\Sigma_{\hat{q}_t}$  are the mean vector and the covariance matrix, respectively, in state  $\hat{q}_t$ . Using linguistic feature sequence  $l$ , the mean vectors and the covariance matrices are calculated by traversing each decision tree that has been set in the trained acoustic model 306. According to Equation (4), the estimated value ( $\hat{o}$ ) for an acoustic feature sequence 317 is obtained using the mean vector  $\mu_{\hat{q}}$ . However,  $\mu_{\hat{q}}$  is a discontinuous sequence that changes in a step-like manner where there is a state transition. In terms of naturalness, low quality voice synthesis results when the synthesis filter 310 synthesizes output data 321 from a discontinuous acoustic feature sequence 317 such as this. In the first embodiment of statistical voice synthesis processing, a training result 315 (model parameter) generation algorithm that takes dynamic features into account may accordingly be employed in the model training unit 305. In cases where an acoustic feature sequence  $(o_t = [c_t^T, \Delta c_t^T]^T)$  in frame  $t$  is composed of a static feature  $c_t$  and a dynamic feature  $\Delta c_t$ , the acoustic feature sequence  $(o = [o_1^T, \dots, o_T^T]^T)$  is expressed over all times with Equation (5) below.

$$o = Wc \tag{5}$$

[0088] Here,  $W$  is a matrix whereby an acoustic feature sequence  $o$  containing a dynamic feature is obtained from static feature sequence  $c = [c_1^T, \dots, c_T^T]^T$ . With Equation (5) as a constraint, the model training unit 305 solves Equation (4) as expressed by Equation (6) below.

$$\hat{c} = \arg \max_c \mathcal{N}(Wc|\mu_{\hat{q}}, \Sigma_{\hat{q}}) \tag{6}$$

[0089] Here,  $\hat{c}$  is the static feature sequence with the greatest probability of output under dynamic feature constraint. By taking dynamic features into account, discontinuities at state boundaries can be resolved, enabling a smoothly changing acoustic feature sequence 317 to be obtained. This also makes it possible for high quality singing voice sound output data 321 to be generated in the synthesis filter 310.

[0090] It should be noted that phoneme boundaries in the singing voice data often are not aligned with the boundaries of musical notes established by the musical score. Such timewise fluctuations are considered to be essential in terms of musical expression. Accordingly, in the first embodiment of statistical voice synthesis processing employing HMM acoustic models described above, in the vocalization of singing voices, a technique may be employed that assumes that there will be time disparities due to various influences, such as phonological differences during vocalization, pitch, or rhythm, and that models lag between vocalization timings in the training data and the musical score. Specifically, as a model for lag on a musical note basis, lag between a singing voice, as viewed in units of musical notes, and a musical score may be represented using a one-dimensional Gaussian distribution and handled as a context-dependent HMM acoustic model similarly to other spectral parameters, logarithmic fundamental frequencies (F0), and the like. In singing voice synthesis such as this, in which HMM acoustic models that include context for "lag" are employed, after the

boundaries in time represented by a musical score have been established, maximizing the joint probability of both the phoneme state duration model and the lag model on a musical note basis makes it possible to determine a temporal structure that takes fluctuations of musical note in the training data into account.

**[0091]** Next, a second embodiment of the statistical voice synthesis processing performed by the voice training section 301 and the voice synthesis section 302 in FIG. 3 will be described. In the first embodiment of statistical voice synthesis processing, in order to predict an acoustic feature sequence 317 from a linguistic feature sequence 316, the trained acoustic model 306 is implemented using a deep neural network (DNN). Correspondingly, the model training unit 305 in the voice training section 301 learns model parameters representing non-linear transformation functions for neurons in the DNN that transform linguistic features into acoustic features, and the model training unit 305 outputs, as the training result 315, these model parameters to the DNN of the trained acoustic model 306 in the voice synthesis section 302.

**[0092]** As described in the above-referenced Non-Patent Documents, normally, acoustic features are calculated in units of frames that, for example, have a width of 5.1 msec (milliseconds), and linguistic features are calculated in phoneme units. Accordingly, the unit of time for linguistic features differs from that for acoustic features. In the first embodiment of statistical voice synthesis processing in which HMM acoustic models are employed, correspondence between acoustic features and linguistic features is expressed using a HMM state sequence, and the model training unit 305 automatically learns the correspondence between acoustic features and linguistic features based on the training musical score data 311 and training singing voice data for a given singer 312 in FIG. 3. In contrast, in the second embodiment of statistical voice synthesis processing in which a DNN is employed, the DNN set in the trained acoustic model 306 is a model that represents a one-to-one correspondence between an input linguistic feature sequence 316 and an output acoustic feature sequence 317, and so the DNN cannot be trained using an input-output data pair having differing units of time. For this reason, in the second embodiment of statistical voice synthesis processing, the correspondence between acoustic feature sequences given in frames and linguistic feature sequences given in phonemes is established in advance, whereby pairs of acoustic features and linguistic features given in frames are generated.

**[0093]** FIG. 5 is a diagram for explaining the operation of the voice synthesis LSI 205, and illustrates the aforementioned correspondence. For example, when the singing voice phoneme sequence (linguistic feature sequence) /k/ /i/ /r/ /a/ /k/ /i/ ((b) in FIG. 5) corresponding to the lyric string "Ki Ra Ki" ((a) in FIG. 5) at the beginning of a song has been acquired, this linguistic feature sequence is mapped to an acoustic feature sequence given in frames ((c) in FIG. 5) in a one-to-many relationship (the relationship between (b) and (c) in FIG. 5). It should be noted that because linguistic features are used as inputs to the DNN of the trained acoustic model 306, it is necessary to express the linguistic features as numerical data. Numerical data obtained by concatenating binary data (0 or 1) or continuous values responsive to contextual questions such as "Is the preceding phoneme /a/?" and "How many phonemes does the current word contain?" is prepared for the linguistic feature sequence for this reason.

**[0094]** In the second embodiment of statistical voice synthesis processing, the model training unit 305 in the voice training section 301 in FIG. 3, as depicted using the group of dashed arrows 501 in FIG. 5, trains the DNN of the trained acoustic model 306 by sequentially passing, in frames, pairs of individual phonemes in a training linguistic feature sequence 313 phoneme sequence (corresponding to (b) in FIG. 5) and individual frames in a training acoustic feature sequence 314 (corresponding to (c) in FIG. 5) to the DNN. The DNN of the trained acoustic model 306, as depicted using the groups of gray circles in FIG. 5, contains neuron groups each made up of an input layer, one or more middle layer, and an output layer.

**[0095]** During voice synthesis, a linguistic feature sequence 316 phoneme sequence (corresponding to (b) in FIG. 5) is input to the DNN of the trained acoustic model 306 in frames. The DNN of the trained acoustic model 306, as depicted using the group of heavy solid arrows 502 in FIG. 5, consequently outputs an acoustic feature sequence 317 in frames. For this reason, in the vocalization model unit 308, the sound source data 319 and the spectral data 318 contained in the acoustic feature sequence 317 are respectively passed to the sound source generator 309 and the synthesis filter 310, and voice synthesis is performed in frames.

**[0096]** The vocalization model unit 308, as depicted using the group of heavy solid arrows 503 in FIG. 5, consequently outputs 225 samples, for example, of output data 321 per frame. Because each frame has a width of 5.1 msec, one sample corresponds to  $5.1 \text{ msec} \div 225 \approx 0.0227 \text{ msec}$ . The sampling frequency of the output data 321 is therefore  $1/0.0227 \approx 44 \text{ kHz}$  (kilohertz).

**[0097]** As described in the above-referenced Non-Patent Documents, the DNN is trained so as to minimize squared error. This is computed according to Equation (7) below using pairs of acoustic features and linguistic features denoted in frames.

$$\hat{\lambda} = \arg \min_{\lambda} \frac{1}{2} \sum_{t=1}^T \|o_t - g_{\lambda}(l_t)\|^2 \quad (7)$$

**[0098]** In this equation,  $o_t$  and  $l_t$  respectively represent an acoustic feature and a linguistic feature in the  $t^{\text{th}}$  frame  $t$ ,  $\hat{\lambda}$  represents model parameters for the DNN of the trained acoustic model 306, and  $g_{\lambda}(\cdot)$  is the non-linear transformation

function represented by the DNN. The model parameters for the DNN are able to be efficiently estimated through backpropagation. When correspondence with processing within the model training unit 305 in the statistical voice synthesis represented by Equation (1) is taken into account, DNN training can be represented as in Equation (8) below.

$$\begin{aligned}\hat{\lambda} &= \arg \max_{\lambda} P(o|l, \lambda) \\ &= \arg \max_{\lambda} \prod_{t=1}^T \mathcal{N}(o_t | \tilde{\mu}_t, \tilde{\Sigma}_t)\end{aligned}\quad (8)$$

[0099] Here,  $\tilde{\mu}_t$  is given as in Equation (9) below.

$$\tilde{\mu}_t = g_{\lambda}(l_t) \quad (9)$$

[0100] As in Equation (8) and Equation (9), relationships between acoustic features and linguistic features are able

to be expressed using the normal distribution  $\mathcal{N}(o_t | \tilde{\mu}_t, \tilde{\Sigma}_t)$ , which uses output from the DNN for the mean vector. In the second embodiment of statistical voice synthesis processing in which a DNN is employed, normally, independent covariance matrices are used for linguistic feature sequences  $l_t$ . In other words, in all frames, the same covariance matrix  $\tilde{\Sigma}_g$  is used for the linguistic feature sequences  $l_t$ . When the covariance matrix  $\tilde{\Sigma}_g$  is an identity matrix, Equation (8) expresses a training process equivalent to that in Equation (7).

[0101] As described in FIG. 5, the DNN of the trained acoustic model 306 estimates an acoustic feature sequence 317 for each frame independently. For this reason, the obtained acoustic feature sequences 317 contain discontinuities that lower the quality of voice synthesis. Accordingly, a parameter generation algorithm employing dynamic features similar to that used in the first embodiment of statistical voice synthesis processing is, for example, used in the present embodiment. This allows the quality of voice synthesis to be improved.

[0102] Detailed description follows regarding the operation of the embodiment of the electronic keyboard instrument 100 of FIGs. 1 and 2 in which the statistical voice synthesis processing described in FIGs. 3 to 5 is employed. FIG. 6 is a diagram illustrating, for the present embodiment, an example data configuration for musical piece data loaded into the RAM 203 from the ROM 202 in FIG. 2. This example data configuration conforms to the Standard MIDI (Musical Instrument Digital Interface) File format, which is one file format used for MIDI files. The musical piece data is configured by data blocks called "chunks". Specifically, the musical piece data is configured by a header chunk at the beginning of the file, a first track chunk that comes after the header chunk and stores lyric data for a lyric part, and a second track chunk that stores performance data for an accompaniment part.

[0103] The header chunk is made up of five values: ChunkID, ChunkSize, FormatType, NumberOfTrack, and TimeDivision. ChunkID is a four byte ASCII code "4D 54 68 64" (in base 16) corresponding to the four half-width characters "MThd", which indicates that the chunk is a header chunk. ChunkSize is four bytes of data that indicate the length of the FormatType, NumberOfTrack, and TimeDivision part of the header chunk (excluding ChunkID and ChunkSize). This length is always "00 00 00 06" (in base 16), for six bytes. FormatType is two bytes of data "00 01" (in base 16). This means that the format type is format 1, in which multiple tracks are used. NumberOfTrack is two bytes of data "00 02" (in base 16). This indicates that in the case of the present embodiment, two tracks, corresponding to the lyric part and the accompaniment part, are used. TimeDivision is data indicating a timebase value, which itself indicates resolution per quarter note. TimeDivision is two bytes of data "01 E0" (in base 16). In the case of the present embodiment, this indicates 480 in decimal notation.

[0104] The first and second track chunks are each made up of a ChunkID, ChunkSize, and performance data pairs. The performance data pairs are made up of DeltaTime\_1[i] and Event\_1[i] (for the first track chunk/lyric part), or DeltaTime\_2[i] and Event\_2[i] (for the second track chunk/accompaniment part). Note that  $0 \leq i \leq L$  for the first track chunk/lyric part, and  $0 \leq i \leq M$  for the second track chunk/accompaniment part. ChunkID is a four byte ASCII code "4D 54 72 6B" (in base 16) corresponding to the four half-width characters "MTrk", which indicates that the chunk is a track chunk. ChunkSize is four bytes of data that indicate the length of the respective track chunk (excluding ChunkID and ChunkSize).

[0105] DeltaTime\_1[i] is variable-length data of one to four bytes indicating a wait time (relative time) from the execution time of Event\_1[i-1] immediately prior thereto. Similarly, DeltaTime\_2[i] is variable-length data of one to four bytes indicating a wait time (relative time) from the execution time of Event\_2[i-1] immediately prior thereto. Event\_1[i] is a meta event (timing information) designating the vocalization timing and pitch of a lyric in the first track chunk/lyric part. Event\_2[i] is a MIDI event (timing information) designating "note on" or "note off" or is a meta event designating time

signature in the second track chunk/accompaniment part. In each DeltaTime\_1[i] and Event\_1[i] performance data pair of the first track chunk/lyric part, Event\_1[i] is executed after a wait of DeltaTime\_1[i] from the execution time of the Event\_1[i-1] immediately prior thereto. The vocalization and progression of lyrics is realized thereby. In each DeltaTime\_2[i] and Event\_2[i] performance data pair of the second track chunk/accompaniment part, Event\_2[i] is executed after a wait of DeltaTime\_2[i] from the execution time of the Event\_2[i-1] immediately prior thereto. The progression of automatic accompaniment is realized thereby.

**[0106]** FIG. 7 is a main flowchart illustrating an example of a control process for the electronic musical instrument of the present embodiment. For this control process, for example, the CPU 201 in FIG. 2 executes a control processing program loaded into the RAM 203 from the ROM 202.

**[0107]** After first performing initialization processing (step S701), the CPU 201 repeatedly executes the series of processes from step S702 to step S708.

**[0108]** In this repeat processing, the CPU 201 first performs switch processing (step S702). Here, based on an interrupt from the key scanner 206 in FIG. 2, the CPU 201 performs processing corresponding to the operation of a switch on the first switch panel 102 or the second switch panel 103 in FIG. 1.

**[0109]** Next, based on an interrupt from the key scanner 206 in FIG. 2, the CPU 201 performs keyboard processing (step S703) that determines whether or not any of the keys on the keyboard 101 in FIG. 1 have been operated, and proceeds accordingly. Here, in response to an operation by a user pressing or releasing any of the keys, the CPU 201 outputs musical sound control data 216 instructing the sound source LSI 204 in FIG. 2 to start generating sound or to stop generating sound.

**[0110]** Next, the CPU 201 processes data that should be displayed on the LCD 104 in FIG. 1, and performs display processing (step S704) that displays this data on the LCD 104 via the LCD controller 208 in FIG. 2. Examples of the data that is displayed on the LCD 104 include lyrics corresponding to the inferred singing voice data 217 being performed, the musical score for the melody corresponding to the lyrics, and information relating to various settings.

**[0111]** Next, the CPU 201 performs song playback processing (step S705). In this processing, the CPU 201 performs a control process described in FIG. 5 on the basis of a performance by a user, generates singing voice data 215, and outputs this data to the voice synthesis LSI 205.

**[0112]** Then, the CPU 201 performs sound source processing (step S706). In the sound source processing, the CPU 201 performs control processing such as that for controlling the envelope of musical sounds being generated in the sound source LSI 204.

**[0113]** Then, the CPU 201 performs voice synthesis processing (step S707). In the voice synthesis processing, the CPU 201 controls voice synthesis by the voice synthesis LSI 205.

**[0114]** Finally, the CPU 201 determines whether or not a user has pressed a non-illustrated power-off switch to turn off the power (step S708). If the determination of step S708 is NO, the CPU 201 returns to the processing of step S702. If the determination of step S708 is YES, the CPU 201 ends the control process illustrated in the flowchart of FIG. 7 and powers off the electronic keyboard instrument 100.

**[0115]** FIGs. 8A to 8C are flowcharts respectively illustrating detailed examples of the initialization processing at step S701 in FIG. 7; tempo-changing processing at step S902 in FIG. 9, described later, during the switch processing of step S702 in FIG. 7; and similarly, song-starting processing at step S906 in FIG. 9 during the switch processing of step S702 in FIG. 7, described later.

**[0116]** First, in FIG. 8A, which illustrates a detailed example of the initialization processing at step S701 in FIG. 7, the CPU 201 performs TickTime initialization processing. In the present embodiment, the progression of lyrics and automatic accompaniment progress in a unit of time called TickTime. The timebase value, specified as the TimeDivision value in the header chunk of the musical piece data in FIG. 6, indicates resolution per quarter note. If this value is, for example, 480, each quarter note has a duration of 480 TickTime. The DeltaTime\_1[i] values and the DeltaTime\_2[i] values, indicating wait times in the track chunks of the musical piece data in FIG. 6, are also counted in units of TickTime. The actual number of seconds corresponding to 1 TickTime differs depending on the tempo specified for the musical piece data. Taking a tempo value as Tempo (beats per minute) and the timebase value as TimeDivision, the number of seconds per unit of TickTime is calculated using the following equation.

$$\text{TickTime (sec)} = 60 / \text{Tempo} / \text{TimeDivision} \quad (10)$$

**[0117]** Accordingly, in the initialization processing illustrated in the flowchart of FIG. 8A, the CPU 201 first calculates TickTime (sec) by an arithmetic process corresponding to Equation (10) (step S801). A prescribed initial value for the tempo value Tempo, e.g., 60 (beats per second), is stored in the ROM 202 in FIG. 2. Alternatively, the tempo value from when processing last ended may be stored in non-volatile memory.

**[0118]** Next, the CPU 201 sets a timer interrupt for the timer 210 in FIG. 2 using the TickTime (sec) calculated at step S801 (step S802). A CPU 201 interrupt for lyric progression and automatic accompaniment (referred to below as an

"automatic-performance interrupt ") is thus generated by the timer 210 every time the TickTime (sec) has elapsed. Accordingly, in automatic-performance interrupt processing (FIG. 10, described later) performed by the CPU 201 based on an automatic-performance interrupt, processing to control lyric progression and the progression of automatic accompaniment is performed every 1 TickTime.

**[0119]** Then, the CPU 201 performs additional initialization processing, such as that to initialize the RAM 203 in FIG. 2 (step S803). The CPU 201 subsequently ends the initialization processing at step S701 in FIG. 7 illustrated in the flowchart of FIG. 8A.

**[0120]** The flowcharts in FIGs. 8B and 8C will be described later. FIG. 9 is a flowchart illustrating a detailed example of the switch processing at step S702 in FIG. 7.

**[0121]** First, the CPU 201 determines whether or not the tempo of lyric progression and automatic accompaniment has been changed using a switch for changing tempo on the first switch panel 102 in FIG. 1 (step S901). If this determination is YES, the CPU 201 performs tempo-changing processing (step S902). The details of this processing will be described later using FIG. 8B. If the determination of step S901 is NO, the CPU 201 skips the processing of step S902.

**[0122]** Next, the CPU 201 determines whether or not a song has been selected with the second switch panel 103 in FIG. 1 (step S903). If this determination is YES, the CPU 201 performs song-loading processing (step S904). In this processing, musical piece data having the data structure described in FIG. 6 is loaded into the RAM 203 from the ROM 202 in FIG. 2. The song-loading processing does not have to come during a performance, and may come before the start of a performance. Subsequent data access of the first track chunk or the second track chunk in the data structure illustrated in FIG. 6 is performed with respect to the musical piece data that has been loaded into the RAM 203. If the determination of step S903 is NO, the CPU 201 skips the processing of step S904.

**[0123]** Then, the CPU 201 determines whether or not a switch for starting a song on the first switch panel 102 in FIG. 1 has been operated (step S905). If this determination is YES, the CPU 201 performs song-starting processing (step S906). The details of this processing will be described later using FIG. 8C. If the determination of step S905 is NO, the CPU 201 skips the processing of step S906.

**[0124]** Next, the CPU 201 determines whether or not the vocoder mode has been changed with the first switch panel 102 in FIG. 1 (step S907). If this determination is YES, the CPU 201 performs vocoder-mode-changing processing (step S908). In other words, the CPU 201 sets the vocoder mode to ON if up to this point the vocoder mode had been set to OFF. Conversely, the CPU 201 sets the vocoder mode to OFF if up to this point the vocoder mode had been set to ON. If the determination of step S907 is NO, the CPU 201 skips the processing of step S908. The CPU 201 sets the vocoder mode to ON or OFF by, for example, changing the value of a prescribed variable in the RAM 203 to 1 or 0. When the CPU 201 sets the vocoder mode to ON, the vocoder mode switch 320 in FIG. 3 is controlled such that instrument sound waveform data 220 from designated sound generation channels (single or plural channels) of the sound source LSI 204 in FIG. 2 are inputted to the synthesis filter 310. However, when the CPU 201 sets the vocoder mode to OFF, the vocoder mode switch 320 in FIG. 3 is controlled such that a sound source signal from the sound source generator 309 in FIG. 3 is input to the synthesis filter 310.

**[0125]** Then, the CPU 201 determines whether or not a switch for selecting an effect on the first switch panel 102 in FIG. 1 has been operated (step S909). If this determination is YES, the CPU 201 performs effect-selection processing (step S910). Here, as described above, a user selects which acoustic effect to apply from among a vibrato effect, a tremolo effect, or a wah effect using the first switch panel 102 when an acoustic effect is to be applied to the vocalized voice sound of the output data 321 output by the acoustic effect application section 322 in FIG. 3. As a result of this selection, the CPU 201 sets the acoustic effect application section 322 in the voice synthesis LSI 205 with whichever acoustic effect was selected. If the determination of step S909 is NO, the CPU 201 skips the processing of step S910.

**[0126]** Depending on the setting, a plurality of effects may be applied at the same time.

**[0127]** Finally, the CPU 201 determines whether or not any other switches on the first switch panel 102 or the second switch panel 103 in FIG. 1 have been operated, and performs processing corresponding to each switch operation (step S911). This processing includes processing for a switch for selecting tone color on the second switch panel 103 allowing, from a plurality of instrument sounds including at least one of a brass sound, a string sound, an organ sound, or an animal cry, the selection of any instrument sound from among the brass sound, the string sound, the organ sound, and the animal cry as the instrument sound for instrument sound waveform data 220 supplied to the vocalization model unit 308 in the voice synthesis LSI 205 from the sound source LSI 204 in FIGs. 2 and 3 when the vocoder mode described above has been selected by a user.

**[0128]** The CPU 201 subsequently ends the switch processing at step S702 in FIG. 7 illustrated in the flowchart of FIG. 9. This processing includes, for example, switch operations such as that for selecting the tone color of musical sounds for the vocoder mode and selecting the designated sound generation channel(s) for the vocoder mode.

**[0129]** FIG. 8B is a flowchart illustrating a detailed example of the tempo-changing processing at step S902 in FIG. 9. As mentioned previously, a change in the tempo value also results in a change in the TickTime (sec). In the flowchart of FIG. 8B, the CPU 201 performs a control process related to changing the TickTime (sec).

**[0130]** Similarly to at step S801 in FIG. 8A, which is performed in the initialization processing at step S701 in FIG. 7,

the CPU 201 first calculates the TickTime (sec) by an arithmetic process corresponding to Equation (10) (step S811). It should be noted that the tempo value Tempo that has been changed using the switch for changing tempo on the first switch panel 102 in FIG. 1 is stored in the RAM 203 or the like.

**[0131]** Next, similarly to at step S802 in FIG. 8A, which is performed in the initialization processing at step S701 in FIG. 7, the CPU 201 sets a timer interrupt for the timer 210 in FIG. 2 using the TickTime (sec) calculated at step S811 (step S812). The CPU 201 subsequently ends the tempo-changing processing at step S902 in FIG. 9 illustrated in the flowchart of FIG. 8B.

**[0132]** FIG. 8C is a flowchart illustrating a detailed example of the song-starting processing at step S906 in FIG. 9.

**[0133]** First, with regards to the progression of automatic performance, the CPU 201 initializes the values of both a DeltaT\_1 (first track chunk) variable and a DeltaT\_2 (second track chunk) variable in the RAM 203 for counting, in units of TickTime, relative time since the last event to 0. Next, the CPU 201 initializes the respective values of an AutoIndex\_1 variable in the RAM 203 for specifying an i value ( $1 \leq i \leq L-1$ ) for DeltaTime\_1[i] and Event\_1[i] performance data pairs in the first track chunk of the musical piece data illustrated in FIG. 6, and an AutoIndex\_2 variable in the RAM 203 for specifying an i ( $1 \leq i \leq M-1$ ) for DeltaTime\_2[i] and Event\_2[i] performance data pairs in the second track chunk of the musical piece data illustrated in FIG. 6, to 0 (the above is step S821). Thus, in the example of FIG. 6, the DeltaTime\_1[0] and Event\_1[0] performance data pair at the beginning of first track chunk and the DeltaTime\_2[0] and Event\_2[0] performance data pair at the beginning of second track chunk are both referenced to set an initial state.

**[0134]** Next, the CPU 201 initializes the value of a SongIndex variable in the RAM 203, which designates the current song position, to 0 (step S822).

**[0135]** The CPU 201 also initializes the value of a SongStart variable in the RAM 203, which indicates whether to advance (=1) or not advance (=0) the lyrics and accompaniment, to 1 (progress) (step S823).

**[0136]** Then, the CPU 201 determines whether or not a user has configured the electronic keyboard instrument 100 to playback an accompaniment together with lyric playback using the first switch panel 102 in FIG. 1 (step S824).

**[0137]** If the determination of step S824 is YES, the CPU 201 sets the value of a Bansou variable in the RAM 203 to 1 (has accompaniment) (step S825). Conversely, if the determination of step S824 is NO, the CPU 201 sets the value of the Bansou variable to 0 (no accompaniment) (step S826). After the processing at step S825 or step S826, the CPU 201 ends the song-starting processing at step S906 in FIG. 9 illustrated in the flowchart of FIG. 8C.

**[0138]** FIG. 10 is a flowchart illustrating a detailed example of the automatic-performance interrupt processing performed based on the interrupts generated by the timer 210 in FIG. 2 every TickTime (sec) (see step S802 in FIG. 8A, or step S812 in FIG. 8B). The following processing is performed on the performance data pairs in the first and second track chunks in the musical piece data illustrated in FIG. 6.

**[0139]** First, the CPU 201 performs a series of processes corresponding to the first track chunk (steps S1001 to S1006). The CPU 201 starts by determining whether or not the value of SongStart is equal to 1, in other words, whether or not advancement of the lyrics and accompaniment has been instructed (step S1001).

**[0140]** When the CPU 201 has determined there to be no instruction to advance the lyrics and accompaniment (the determination of step S1001 is NO), the CPU 201 ends the automatic-performance interrupt processing illustrated in the flowchart of FIG. 10 without advancing the lyrics and accompaniment.

**[0141]** When the CPU 201 has determined there to be an instruction to advance the lyrics and accompaniment (the determination of step S1001 is YES), the CPU 201 then determines whether or not the value of DeltaT\_1, which indicates the relative time since the last event in the first track chunk, matches the wait time DeltaTime\_1 [AutoIndex\_1] of the performance data pair indicated by the value of AutoIndex\_1 that is about to be executed (step S1002).

**[0142]** If the determination of step S1002 is NO, the CPU 201 increments the value of DeltaT\_1, which indicates the relative time since the last event in the first track chunk, by 1, and the CPU 201 allows the time to advance by 1 TickTime corresponding to the current interrupt (step S1003). Following this, the CPU 201 proceeds to step S1007, which will be described later.

**[0143]** If the determination of step S1002 is YES, the CPU 201 executes the first track chunk event Event\_1[AutoIndex\_1] of the performance data pair indicated by the value of AutoIndex\_1 (step S1004). This event is a song event that includes lyric data.

**[0144]** Then, the CPU 201 stores the value of AutoIndex\_1, which indicates the position of the song event that should be performed next in the first track chunk, in the SongIndex variable in the RAM 203 (step S1004).

**[0145]** The CPU 201 then increments the value of AutoIndex\_1 for referencing the performance data pairs in the first track chunk by 1 (step S1005).

**[0146]** Next, the CPU 201 resets the value of DeltaT\_1, which indicates the relative time since the song event most recently referenced in the first track chunk, to 0 (step S1006). Following this, the CPU 201 proceeds to the processing at step S1007.

**[0147]** Then, the CPU 201 performs a series of processes corresponding to the second track chunk (steps S1007 to S1013). The CPU 201 starts by determining whether or not the value of DeltaT\_2, which indicates the relative time since the last event in the second track chunk, matches the wait time DeltaTime\_2[AutoIndex\_2] of the performance data pair



indicated by the value of AutoIndex\_2 that is about to be executed (step S1007).

**[0148]** If the determination of step S1007 is NO, the CPU 201 increments the value of DeltaT\_2, which indicates the relative time since the last event in the second track chunk, by 1, and the CPU 201 allows the time to advance by 1 TickTime corresponding to the current interrupt (step S1008). The CPU 201 subsequently ends the automatic-performance interrupt processing illustrated in the flowchart of FIG. 10.

**[0149]** If the determination of step S1007 is YES, the CPU 201 then determines whether or not the value of the Bansou variable in the RAM 203 that denotes accompaniment playback is equal to 1 (has accompaniment) (step S1009) (see steps S824 to S826 in FIG. 8C).

**[0150]** If the determination of step S1009 is YES, the CPU 201 executes the second track chunk accompaniment event Event\_2[AutoIndex\_2] indicated by the value of AutoIndex\_2 (step S1010). If the event Event\_2[AutoIndex\_2] executed here is, for example, a "note on" event, the key number and velocity specified by this "note on" event are used to issue a command to the sound source LSI 204 in FIG. 2 to generate sound for a musical tone in the accompaniment. However, if the event Event\_2[AutoIndex\_2] is, for example, a "note off" event, the key number and velocity specified by this "note off" event are used to issue a command to the sound source LSI 204 in FIG. 2 to silence a musical tone being generated for the accompaniment.

**[0151]** However, if the determination of step S1009 is NO, the CPU 201 skips step S1010 and proceeds to the processing at the next step S1011 without executing the current accompaniment event Event\_2[AutoIndex\_2]. Here, in order to progress in sync with the lyrics, the CPU 201 performs only control processing that advances events.

**[0152]** After step S1010, or when the determination of step S1009 is NO, the CPU 201 increments the value of AutoIndex\_2 for referencing the performance data pairs for accompaniment data in the second track chunk by 1 (step S1011).

**[0153]** Next, the CPU 201 resets the value of DeltaT\_2, which indicates the relative time since the event most recently executed in the second track chunk, to 0 (step S1012).

**[0154]** Then, the CPU 201 determines whether or not the wait time DeltaTime\_2[AutoIndex\_2] of the performance data pair indicated by the value of AutoIndex\_2 to be executed next in the second track chunk is equal to 0, or in other words, whether or not this event is to be executed at the same time as the current event (step S1013).

**[0155]** If the determination of step S1013 is NO, the CPU 201 ends the current automatic-performance interrupt processing illustrated in the flowchart of FIG. 10.

**[0156]** If the determination of step S1013 is YES, the CPU 201 returns to step S1009, and repeats the control processing relating to the event Event\_2[AutoIndex\_2] of the performance data pair indicated by the value of AutoIndex\_2 to be executed next in the second track chunk. The CPU 201 repeatedly performs the processing of steps S1009 to S1013 same number of times as there are events to be simultaneously executed. The above processing sequence is performed when a plurality of "note on" events are to generate sound at simultaneous timings, as for example happens in chords and the like.

**[0157]** FIG. 11 is a flowchart illustrating a detailed example of the song playback processing at step S705 in FIG. 7.

**[0158]** First, at step S1004 in the automatic-performance interrupt processing of FIG. 10, the CPU 201 determines whether or not a value has been set for the SongIndex variable in the RAM 203, and that this value is not a null value (step S1101). The SongIndex value indicates whether or not the current timing is a singing voice playback timing.

**[0159]** If the determination of step S1101 is YES, that is, if the present time is a song playback timing, the CPU 201 then determines whether or not a new user key press on the keyboard 101 in FIG. 1 has been detected by the keyboard processing at step S703 in FIG. 7 (step S1102).

**[0160]** If the determination of step S1102 is YES, the CPU 201 sets the pitch specified by a user key press to a non-illustrated register, or to a variable in the RAM 203, as a vocalization pitch (step S1103).

**[0161]** Next, the CPU 201 determines whether the vocoder mode is currently ON or OFF by, for example, checking the value of the prescribed variable in the RAM 203 (step S1105).

**[0162]** If the determination at step S1105 is that the vocoder mode is ON, the CPU 201 generates "note on" data for producing musical sound in the designated sound generation channel(s) having the tone color set previously at step S909 in FIG. 9 and at a vocalization pitch set to the pitch based on a key press set at step S1103, and instructs the sound source LSI 204 to perform processing to produce musical sound (step S1106). The sound source LSI 204 generates a musical sound signal for the designated sound generation channel(s) with the designated tone color specified by the CPU 201, and this signal is input to the synthesis filter 310 as instrument sound waveform data 220 via the vocoder mode switch 320 in the voice synthesis LSI 205.

**[0163]** If the determination of step S1105 is that the vocoder mode is OFF, the CPU 201 skips the processing of step S1106. As a result, a sound source signal from the sound source generator 309 in the voice synthesis LSI 205 is input to the synthesis filter 310 via the vocoder mode switch 320.

**[0164]** Then, the CPU 201 reads the lyric string from the song event Event\_1[SongIndex] in the first track chunk of the musical piece data in the RAM 203 indicated by the SongIndex variable in the RAM 203. The CPU 201 generates singing voice data 215 for vocalizing, at the vocalization pitch set to the pitch based on a key press that was set at step

S1103, output data 321 corresponding to the lyric string that was read, and instructs the voice synthesis LSI 205 to perform vocalization processing (step S1107). The voice synthesis LSI 205 implements the first embodiment or the second embodiment of statistical voice synthesis processing described with reference to FIGs. 3 to 5, whereby lyrics from the RAM 203 specified as musical piece data are, in real time, synthesized into and output as output data 321 to be sung at the pitch(es) of keys on the keyboard 101 pressed by a user.

**[0165]** As a result, if the determination at step S1105 is that the vocoder mode is ON, musical sound output data 220 generated and output by the sound source LSI 204 based on the playing of a user on the keyboard 101 (FIG. 1) is input to the synthesis filter 310 operating on the basis of spectral data 318 input from the trained acoustic model 306, and output data 321 is output from the synthesis filter 310 in a polyphonic manner.

**[0166]** However, if the determination at step S1105 is that the vocoder mode is OFF, a sound source signal generated and output by the sound source generator 309 based on the playing of a user on the keyboard 101 (FIG. 1) is input to the synthesis filter 310 operating on the basis of spectral data 318 input from the acoustic model unit 306, and, operating monophonically, output data 321 is output from the synthesis filter 310.

**[0167]** If at step S1101 it is determined that the present time is a song playback timing and the determination of step S1102 is NO, that is, if it is determined that no new key press is detected at the present time, the CPU 201 reads the data for a pitch from the song event Event\_1[SongIndex] in the first track chunk of the musical piece data in the RAM 203 indicated by the SongIndex variable in the RAM 203, and sets this pitch to a non-illustrated register, or to a variable in the RAM 203, as a vocalization pitch (step S1104).

**[0168]** Then, by performing the processing at step S1105 and subsequent steps, described above, the CPU 201 instructs the voice synthesis LSI 205 to perform vocalization processing of the output data 321, 217 (step S1105 to S1107). In implementing the first embodiment or the second embodiment of statistical voice synthesis processing described with reference to FIGs. 3 to 5, even if a user has not pressed a key on the keyboard 101, the voice synthesis LSI 205, as inferred singing voice data 217 to be sung in accordance with a default pitch specified in the musical piece data, synthesizes and outputs lyrics from the RAM 203 specified as musical piece data in a similar manner.

**[0169]** After the processing of step S1107, the CPU 201 stores the song position at which playback was performed indicated by the SongIndex variable in the RAM 203 in a SongIndex\_pre variable in the RAM 203 (step S1108).

**[0170]** Then, the CPU 201 clears the value of the SongIndex variable so as to become a null value and makes subsequent timings non-song playback timings (step S1109). The CPU 201 subsequently ends the song playback processing at step S705 in FIG. 7 illustrated in the flowchart of FIG. 11.

**[0171]** If the determination of step S1101 is NO, that is, if the present time is not a song playback timing, the CPU 201 then determines whether or not "what is referred to as a legato playing style" for applying an effect has been detected on the keyboard 101 in FIG. 1 by the keyboard processing at step S703 in FIG. 7 (step S1110). As described above, this legato style of playing is a playing style in which, for example, while a first key is being pressed in order to playback a song at step S1102, another second key is repeatedly struck. In such case, at step S1110, if the speed of repetition of the presses is greater than or equal to a prescribed speed when the pressing of a second key has been detected, the CPU 201 determines that a legato playing style is being performed.

**[0172]** If the determination of step S1108 is NO, the CPU 201 ends the song playback processing at step S705 in FIG. 7 illustrated in the flowchart of FIG. 11.

**[0173]** If the determination of step S1110 is YES, the CPU 201 calculates the difference in pitch between the vocalization pitch set at step S1103 and the pitch of the key on the keyboard 101 in FIG. 1 being repeatedly struck in "what is referred to as a legato playing style" (step S1111).

**[0174]** Then, the CPU 201 sets the effect size in the acoustic effect application section 322 (FIG. 3) in the voice synthesis LSI 205 in FIG. 2 in correspondence with the difference in pitch calculated at step S1111 (step S1112). Consequently, the acoustic effect application section 322 subjects the output data 321 output from the synthesis filter 310 in the voice synthesis section 302 to processing to apply the acoustic effect selected at step S908 in FIG. 9 with the aforementioned size, and the acoustic effect application section 320 outputs the final inferred singing voice data 217 (FIG. 2, FIG. 3).

**[0175]** The processing of step S1111 and step S1112 enables an acoustic effect such as a vibrato effect, a tremolo effect, or a wah effect to be applied to output data 321 output from the voice synthesis section 302, and a variety of singing voice expressions are implemented thereby.

**[0176]** After the processing at step S1112, the CPU 201 ends the song playback processing at step S705 in FIG. 7 illustrated in the flowchart of FIG. 11.

**[0177]** In the first embodiment of statistical voice synthesis processing employing HMM acoustic models described with reference to FIGs. 3 and 4, it is possible to reproduce subtle musical expressions, such as for particular singers or singing styles, and it is possible to achieve a singing voice quality that is smooth and free of connective distortion. The training result 315 can be adapted to other singers, and various types of voices and emotions can be expressed, by performing a transformation on the training results 315 (model parameters). All model parameters for HMM acoustic models are able to be machine-learned from training musical score data 311 and training singing voice data for a given

singer 312. This makes it possible to automatically create a voice synthesis system in which the features of a particular singer are acquired as HMM acoustic models and these features are reproduced during synthesis. The fundamental frequency and duration of a singing voice follows the melody and tempo in a musical score, and changes in pitch overtime and the temporal structure of rhythm can be uniquely established from the musical score. However, a singing voice synthesized therefrom is dull and mechanical, and lacks appeal as a singing voice. Actual singing voices are not standardized as in a musical score, but rather have a style that is specific to each singer due to voice quality, pitch of voice, and changes in the structures thereof over time. In the first embodiment of statistical voice synthesis processing in which HMM acoustic models are employed, time series variations in spectral information and pitch information in a singing voice is able to be modeled on the basis of context, and by additionally taking musical score information into account, it is possible to reproduce a singing voice that is even closer to an actual singing voice. The HMM acoustic models employed in the first embodiment of statistical voice synthesis processing correspond to generative models that consider how, with regards to vibration of the vocal cords and vocal tract characteristics of a singer, an acoustic feature sequence of a singing voice changes over time during vocalization when lyrics are vocalized in accordance with a given melody. In the first embodiment of statistical voice synthesis processing, HMM acoustic models that include context for "lag" are used. The synthesis of singing voice sounds that able to accurately reproduce singing techniques having a tendency to change in a complex manner depending on the singing voice characteristics of the singer is implemented thereby. By fusing such techniques in the first embodiment of statistical voice synthesis processing, in which HMM acoustic models are employed, with real-time performance technology using the electronic keyboard instrument 100, for example, singing techniques and vocal qualities of a model singer that were not possible with a conventional electronic musical instrument employing concatenative synthesis or the like are able to be reflected accurately, and performances in which a singing voice sounds as if that singer were actually singing are able to be realized in concert with, for example, a keyboard performance on the electronic keyboard instrument 100.

**[0178]** In the second embodiment of statistical voice synthesis processing employing a DNN acoustic model described with reference to FIGs. 3 and 5, the decision tree based context-dependent HMM acoustic models in the first embodiment of statistical voice synthesis processing are replaced with a DNN. It is thereby possible to express relationships between linguistic feature sequences and acoustic feature sequences using complex non-linear transformation functions that are difficult to express in a decision tree. In decision tree based context-dependent HMM acoustic models, because corresponding training data is also classified based on decision trees, the training data allocated to each context-dependent HMM acoustic model is reduced. In contrast, training data is able to be efficiently utilized in a DNN acoustic model because all of the training data used to train a single DNN. Thus, with a DNN acoustic model it is possible to predict acoustic features with greater accuracy than with HMM acoustic models, and the naturalness of voice synthesis is able to be greatly improved. In a DNN acoustic model, it is possible to use linguistic feature sequences relating to frames. In other words, in a DNN acoustic model, because correspondence between acoustic feature sequences and linguistic feature sequences is determined in advance, it is possible to utilize linguistic features relating to frames, such as "the number of consecutive frames for the current phoneme" and "the position of the current frame inside the phoneme". Such linguistic features are not easily taken into account in HMM acoustic models. Thus using linguistic feature relating to frames allows features to be modeled in more detail and makes it possible to improve the naturalness of voice synthesis. By fusing such techniques in the second embodiment of statistical voice synthesis processing, in which a DNN acoustic model is employed, with real-time performance technology using the electronic keyboard instrument 100, for example, singing voice performances based on a keyboard performance, for example, can be made to more naturally approximate the singing techniques and vocal qualities of a model singer.

**[0179]** In the embodiments described above, statistical voice synthesis processing techniques are employed as voice synthesis methods, can be implemented with markedly less memory capacity compared to conventional concatenative synthesis. For example, in an electronic musical instrument that uses concatenative synthesis, memory having several hundred megabytes of storage capacity is needed for voice sound fragment data. However, the present embodiments get by with memory having just a few megabytes of storage capacity in order to store training result 315 model parameters in FIG. 3. This makes it possible to provide a lower cost electronic musical instrument, and allows singing voice performance systems with high quality sound to be used by a wider range of users.

**[0180]** Moreover, in a conventional fragmentary data method, it takes a great deal of time (years) and effort to produce data for singing voice performances since fragmentary data needs to be adjusted by hand. However, because almost no data adjustment is necessary to produce training result 315 model parameters for the HMM acoustic models or the DNN acoustic model of the present embodiments, performance data can be produced with only a fraction of the time and effort. This also makes it possible to provide a lower cost electronic musical instrument. Further, using a server computer 300 available for use as a cloud service, or training functionality built into the voice synthesis LSI 205, general users can train the electronic musical instrument using their own voice, the voice of a family member, the voice of a famous person, or another voice, and have the electronic musical instrument give a singing voice performance using this voice for a model voice. In this case too, singing voice performances that are markedly more natural and have higher quality sound than hitherto are able to be realized with a lower cost electronic musical instrument.

**[0181]** In particular, users are able to switch the vocoder mode ON/OFF using the first switch panel 102 in the present embodiment, and when the vocoder mode is OFF, output data 321 generated and output by the voice synthesis section 302 in FIG. 3 is entirely modeled by the trained acoustic model 306, and as described above, this enables a singing voice that is both natural-sounding and very faithful the singing voice of the singer to be produced. However, when the vocoder mode is ON, because instrument sound waveform data 220 for instrument sounds generated by the sound source LSI 204 is used as a sound source signal, the essence of instrument sounds set in the sound source LSI 204 as well as the vocal characteristics of the singing voice of the singer come through clearly, allowing effective output data 321 to be output. An effect in which a plurality of singing voices seem to be in harmony can also be achieved owing to polyphonic operation being possible in the vocoder mode. It is thus possible to provide an electronic musical instrument that sings well in a singing voice corresponding to the singing voice of a singer that has been learned on the basis of pitches specified by a user.

**[0182]** If the voice synthesis LSI 205 has spare processing capacity, when the vocoder mode is OFF, the sound source signal generated by the sound source generator 309 may be made polyphonic such that polyphonic output data 321 is output from the synthesis filter 310.

**[0183]** It should be noted that the vocoder mode may be switched between ON/OFF in the middle of performing a single song.

**[0184]** In the embodiments described above, the present invention is embodied as an electronic keyboard instrument. However, the present invention can also be applied to electronic string instruments and other electronic musical instruments.

**[0185]** Voice synthesis methods able to be employed for the vocalization model unit 308 in FIG. 3 are not limited to cepstrum voice synthesis, and various voice synthesis methods, such as LSP voice synthesis, may be employed therefor.

**[0186]** In the embodiments described above, a first embodiment of statistical voice synthesis processing in which HMM acoustic models are employed and a second embodiment of a voice synthesis method in which a DNN acoustic model is employed were described. However, the present invention is not limited thereto. Any voice synthesis method using statistical voice synthesis processing may be employed by the present invention, such as, for example, an acoustic model that combines HMMs and a DNN.

**[0187]** In the embodiments described above, lyric information is given as musical piece data. However, text data obtained by voice recognition performed on content being sung in real time by a user may be given as lyric information in real time. The present invention is not limited to the embodiments described above, and various changes in implementation are possible without departing from the spirit of the present invention. Insofar as possible, the functionalities performed in the embodiments described above may be implemented in any suitable combination. Moreover, there are many aspects to the embodiments described above, and the invention may take on a variety of forms through the appropriate combination of the disclosed plurality of constituent elements. For example, if after omitting several constituent elements from out of all constituent elements disclosed in the embodiments the advantageous effect is still obtained, the configuration from which these constituent elements have been omitted may be considered to be one form of the invention.

**[0188]** It will be apparent to those skilled in the art that various modifications and variations can be made in the present invention without departing from the scope of the appended claims. Thus, it is intended that the present invention cover modifications and variations that come within the scope of the appended claims. In particular, it is explicitly contemplated that any part or whole of any two or more of the embodiments and their modifications described above can be combined and regarded within the scope of the present invention.

## Claims

1. An electronic musical instrument comprising:

a plurality of operation elements (101) respectively corresponding to mutually different pitch data;  
 a memory (202) configured to store a trained acoustic model (306) obtained by performing machine learning (305) on training musical score data (311) including training lyric data (311a) and training pitch data (311b), and on training singing voice data (312) of a singer corresponding to the training musical score data (311), the trained acoustic model (306) being configured to receive lyric data (215a) and prescribed pitch data (215b) and output acoustic feature data (317) of a singing voice of the singer in response to the received lyric data and pitch data; and  
 at least one processor (205) in which a first mode and a second mode are interchangeably selectable, wherein in the first mode, the at least one processor (205) is configured to:

in accordance with a user operation on an operation element in the plurality of operation elements (101),

input prescribed lyric data (215a) and pitch data (215b) corresponding to the user operation of the operation element to the trained acoustic model (306) so as to cause the trained acoustic model (306) to output the acoustic feature data (317) in response to the inputted prescribed lyric data (215a) and the inputted pitch data (215b), and

digitally synthesize and output inferred singing voice data (217) that infers a singing voice of the singer on the basis of at least a portion of the acoustic feature data (317) output by the trained acoustic model (306) in response to the inputted prescribed lyric data (215a) and the inputted pitch data (215b), and on the basis of instrument sound waveform data (220) that are synthesized in accordance with the pitch data (215b) corresponding to the user operation of the operation element, and

wherein in the second mode, the at least one processor (205) is configured to:

in accordance with a user operation on an operation element in the plurality of operation elements (101), input prescribed lyric data (215a) and pitch data (215b) corresponding to the user operation of the operation element to the trained acoustic model (306) so as to cause the trained acoustic model (306) to output the acoustic feature data (317) in response to the inputted prescribed lyric data (215a) and the inputted pitch data (215b), and

digitally synthesize and output inferred singing voice data (217) that infers a singing voice of the singer on the basis of the acoustic feature data (317) output by the trained acoustic model (306) in response to the inputted prescribed lyric data (215a) and the inputted pitch data (215b), without using instrument sound waveform data (202) that are synthesized in accordance with the pitch data (215b) corresponding to the user operation of the operation element.

2. The electronic musical instrument according to claim 1, wherein the at least one processor is configured to switch (320) between the first mode and the second mode based on a user operation of a mode selection operation element provided in the electronic musical instrument.

3. The electronic musical instrument according to claim 1, wherein the memory (202) is configured to contain melody pitch data (215d) indicating operation elements that a user is to operate, singing voice output timing data (215c) indicating output timings at which respective singing voices for pitches indicated by the melody pitch data (215d) are to be output, and lyric data (215a) respectively corresponding to the melody pitch data (215d), and wherein in the first mode, the at least one processor (205) is configured to:

when a user operation for producing a singing voice is performed at an output timing indicated by the singing voice output timing data (215c), input pitch data (215b) corresponding to the user-operated operation element and lyric data (215a) corresponding to said output timing to the trained acoustic model (306), and output, at said output timing, inferred singing voice data (217) that infers the singing voice of the singer on the basis of the at least a portion of the acoustic feature data (317) output by the trained acoustic model (306) in response to the input, and

when a user operation for producing a singing voice is not performed at the output timing indicated by the singing voice output timing data (215c), input melody pitch data (215d) corresponding to said output timing and lyric data (215a) corresponding to said output timing to the trained acoustic model (306), and output, at said output timing, inferred singing voice data (217) that infers the singing voice of the singer on the basis of the at least a portion of the acoustic feature data (317) output by the trained acoustic model (306) in response to the input.

4. The electronic musical instrument according to claim 1, wherein the acoustic feature data (317) of the singing voice of the singer includes spectral data (318) that models a vocal tract of the singer and sound source data (319) that models vocal cords of the singer, and wherein in the second mode, the at least one processor (205) is configured to synthesize the inferred singing voice data (217) that infers the singing voice of the singer on the basis of the spectral data (318) and the sound source data (319).

5. The electronic musical instrument according to claim 1, further comprising a selection operation element (102) that, from a plurality of instrument sounds including at least one of a brass sound, a string sound, an organ sound, or an animal cry, is configured to specify one of the instrument sounds in response to a user operation, and wherein in the first mode, the instrument sound waveform data (220) corresponds to the instrument sound specified by the selection operation element.

6. The electronic musical instrument according to claim 1,  
wherein the acoustic feature data (317) of the singing voice of the singer includes spectral data (318) that models  
a vocal tract of the singer and sound source data (319) that models vocal cords of the singer, and  
wherein in the first mode, the at least one processor (205) is configured to synthesize the inferred singing voice data  
(217) that infers the singing voice of the singer by applying an acoustic feature of the spectral data (318) to the  
instrument sound waveform data (220) without using the sound source data (319) of the acoustic feature data (317).
7. The electronic musical instrument according to claim 1, wherein the trained acoustic model (306) has been trained  
via machine learning (305) using at least one of a deep neural network or a hidden Markov model.
8. The electronic musical instrument according to claim 1,  
wherein the plurality of operation elements (101) include a first operation element as the operation element that was  
operated by the user and a second operation element that meets a prescribed condition with respect to the first  
operation element, and  
wherein in both of the first and second modes, the at least one processor (205) is configured to apply an acoustic  
effect (322) to the inferred singing voice data (217) when the second operation element is operated while the first  
operation element is being operated.
9. The electronic musical instrument according to claim 8, wherein the at least one processor (205) is configured to  
change a depth of the acoustic effect (322) in accordance with a difference in pitch (S1111) between a pitch corre-  
sponding to the first operation element and a pitch corresponding to the second operation element.
10. The electronic musical instrument according to claim 8, wherein the second operation element is a black key.
11. The electronic musical instrument according to claim 8, wherein the acoustic effect (322) includes at least one of a  
vibrato effect, a tremolo effect, or a wah-wah effect.
12. A method performed by at least one processor (205) in an electronic musical instrument that includes, in addition  
to the at least one processor (205): a plurality of operation elements (101) respectively corresponding to mutually  
different pitch data; and a memory (202) that stores a trained acoustic model (306) obtained by performing machine  
learning (305) on training musical score data (311) including training lyric data (311a) and training pitch data (311b),  
and on training singing voice data (312) of a singer corresponding to the training musical score data (311), the  
trained acoustic model (306) being configured to receive lyric data (215a) and prescribed pitch data (215b) and  
output acoustic feature data (317) of a singing voice of the singer in response to the received lyric data and pitch  
data, a first mode and a second mode being interchangeably selectable in the at least one processor (205), the  
method comprising, via the at least one processor (205):

selecting one of the first mode and the second mode in response to user operation;  
in the first mode:

in accordance with a user operation on an operation element in the plurality of operation elements (101),  
inputting prescribed lyric data (215a) and pitch data (215b) corresponding to the user operation of the  
operation element to the trained acoustic model (306) so as to cause the trained acoustic model (306) to  
output the acoustic feature data (317) in response to the inputted prescribed lyric data (215a) and the  
inputted pitch data (215b), and  
digitally synthesizing and outputting inferred singing voice data (217) that infers a singing voice of the singer  
on the basis of at least a portion of the acoustic feature data (317) output by the trained acoustic model  
(306) in response to the inputted prescribed lyric data (215a) and the inputted pitch data (215b), and on  
the basis of instrument sound waveform data (220) that are synthesized in accordance with the pitch data  
(215b) corresponding to the user operation of the operation element, and

in the second mode:

in accordance with a user operation on an operation element in the plurality of operation elements (101),  
inputting prescribed lyric data (215a) and pitch data (215b) corresponding to the user operation of the  
operation element to the trained acoustic model (306) so as to cause the trained acoustic model (306) to  
output the acoustic feature data (317) in response to the inputted prescribed lyric data (215a) and the  
inputted pitch data (215b), and

digitally synthesizing and outputting inferred singing voice data (217) that infers a singing voice of the singer on the basis of the acoustic feature data (317) output by the trained acoustic model (306) in response to the inputted prescribed lyric data (215a) and the inputted pitch data (215b), without using instrument sound waveform data (220) that are synthesized in accordance with the pitch data (215b) corresponding to the user operation of the operation element.

13. The method according to claim 12, wherein the method includes, via the at least one processor (205), switching between the first mode and the second mode based on a user operation of a mode selection operation element provided in the electronic musical instrument.

14. The method according to claim 12, wherein the memory (202) contains melody pitch data (215d) indicating operation elements that a user is to operate, singing voice output timing data (215c) indicating output timings at which respective singing voices for pitches indicated by the melody pitch data (215d) are to be output, and lyric data (215a) respectively corresponding to the melody pitch data (215d), and wherein in the first mode, the method include, via the at least one processor (205):

when a user operation for producing a singing voice is performed at an output timing indicated by the singing voice output timing data (215c), inputting pitch data (215b) corresponding to the user-operated operation element and lyric data (215a) corresponding to said output timing to the trained acoustic model (306), and outputting, at said output timing, inferred singing voice data (217) that infers the singing voice of the singer on the basis of the at least a portion of the acoustic feature data (317) output by the trained acoustic model (306) in response to the input, and

when a user operation for producing a singing voice is not performed at the output timing indicated by the singing voice output timing data (215c), inputting melody pitch data (215d) corresponding to said output timing and lyric data (215a) corresponding to said output timing to the trained acoustic model (306), and outputting, at said output timing, inferred singing voice data (217) that infers the singing voice of the singer on the basis of the at least a portion of the acoustic feature data (317) output by the trained acoustic model (306) in response to the input.

15. The method according to claim 12, wherein the acoustic feature data (317) of the singing voice of the singer includes spectral data (318) that models a vocal tract of the singer and sound source data (319) that models vocal cords of the singer, and wherein the method includes, in the second mode, causing the at least one processor (205) to synthesize the inferred singing voice data (217) that infers the singing voice of the singer on the basis of the spectral data (318) and the sound source data (319).

16. The method according to claim 12, wherein the electronic musical instrument further includes a selection operation element (102) that, from a plurality of instrument sounds including at least one of a brass sound, a string sound, an organ sound, or an animal cry, specifies one of the instrument sounds in response to a user operation, and wherein in the first mode, the instrument sound waveform data (220) corresponds to the instrument sound specified by the selection operation element (102).

17. The method according to claim 12, wherein the acoustic feature data (317) of the singing voice of the singer includes spectral data (318) that models a vocal tract of the singer and sound source data (319) that models vocal cords of the singer, and wherein in the first mode, the inferred singing voice data (217) that infers the singing voice of the singer is synthesized by applying an acoustic feature of the spectral data (318) to the instrument sound waveform data (220) without using the sound source data (319) of the acoustic feature data (317).

18. A non-transitory computer-readable storage medium having stored thereon a program executable by at least one processor (205) in an electronic musical instrument that includes, in addition to the at least one processor (205): a plurality of operation elements (101) respectively corresponding to mutually different pitch data; and a memory (202) that stores a trained acoustic model (306) obtained by performing machine learning (305) on training musical score data (311) including training lyric data (311a) and training pitch data (311b), and on training singing voice data (312) of a singer corresponding to the training musical score data (311), the trained acoustic model (306) being configured to receive lyric data (215a) and prescribed pitch data (215b) and output acoustic feature data (317) of a singing

voice of the singer in response to the received lyric data and pitch data, a first mode and a second mode being interchangeably selectable in the at least one processor (205), the program causing the at least one processor (205) to perform the following:

5 selecting one of the first mode and the second mode in response to user operation;  
in the first mode:

10 in accordance with a user operation on an operation element in the plurality of operation elements (101),  
inputting prescribed lyric data (215a) and pitch data (215b) corresponding to the user operation of the  
operation element to the trained acoustic model (306) so as to cause the trained acoustic model (306) to  
output the acoustic feature data (317) in response to the inputted prescribed lyric data (215a) and the  
inputted pitch data (215b), and  
15 digitally synthesizing and outputting inferred singing voice data (217) that infers a singing voice of the singer  
on the basis of at least a portion of the acoustic feature data (317) output by the trained acoustic model  
(306) in response to the inputted prescribed lyric data (215a) and the inputted pitch data (215b), and on  
the basis of instrument sound waveform data (220) that are synthesized in accordance with the pitch data  
(215b) corresponding to the user operation of the operation element, and

20 in the second mode:

25 in accordance with a user operation on an operation element in the plurality of operation elements (101),  
inputting prescribed lyric data (215a) and pitch data (215b) corresponding to the user operation of the  
operation element to the trained acoustic model (306) so as to cause the trained acoustic model (306) to  
output the acoustic feature data (317) in response to the inputted prescribed lyric data (215a) and the  
inputted pitch data (215b), and  
30 digitally synthesizing and outputting inferred singing voice data (217) that infers a singing voice of the singer  
on the basis of the acoustic feature data (317) output by the trained acoustic model (306) in response to  
the inputted prescribed lyric data (215a) and the inputted pitch data (215b), without using instrument sound  
waveform data (220) that are synthesized in accordance with the pitch data (215b) corresponding to the  
user operation of the operation element.

## Patentansprüche

### 35 1. Elektronisches Musikinstrument, umfassend:

eine Vielzahl von Betriebselementen (101), die entsprechend mit gegenseitig verschiedenen Tonhöhendaten korrespondiert;  
einen Speicher (202), der konfiguriert ist, um ein trainiertes akustisches Modell (306) zu speichern, das mittels  
40 Durchführung von maschinellem Lernen (305) auf Trainingsmusikpartiturdaten (311), einschließlich Trainings-  
textdaten (311a) und Trainingstonhöhendaten (311b), und auf Trainingsgesangsstimmendaten (312) eines  
Sängers, die mit den Trainingsmusikpartiturdaten (311) korrespondieren, erhalten wurde, wobei das trainierte  
akustische Modell (306) konfiguriert ist, um Textdaten (215a) zu empfangen und vorgeschriebene Tonhöhen-  
daten (215b) und akustische Merkmalsdaten (317) einer Gesangsstimme des Sängers als Antwort auf die  
45 empfangenen Textdaten und Tonhöhendaten auszugeben; und  
mindestens einen Prozessor (205), in dem ein erster Modus und ein zweiter Modus auswechselbar wählbar sind;  
wobei in dem ersten Modus der mindestens eine Prozessor (205) konfiguriert ist, um:

50 in Übereinstimmung mit einem Benutzerbetrieb auf einem Betriebselement in der Vielzahl von Betriebse-  
lementen (101) vorgeschriebene Textdaten (215a) und Tonhöhendaten (215b), die mit dem Benutzerbetrieb  
des Betriebselements korrespondieren, in das trainierte akustische Modell (306) einzugeben, um das trai-  
nierte akustische Modell (306) zu veranlassen, die akustischen Merkmalsdaten (317) als Antwort auf die  
eingegebenen vorgeschriebenen Textdaten (215a) und die eingegebenen Tonhöhendaten (215b) auszu-  
geben, und  
55 abgeleitete Gesangsstimmendaten (217), die eine Gesangsstimme des Sängers auf der Basis von min-  
destens einem Abschnitt der akustischen Merkmalsdaten (317) ableiten, die mittels des trainierten akusti-  
schen Modells (306) als Antwort auf die eingegebenen vorgeschriebenen Textdaten (215a) und die einge-  
geben Tonhöhendaten (215b) ausgegeben werden, und auf der Basis von Instrumentenklangwellenform-



daten (220), die in Übereinstimmung mit den Tonhöhendaten (215b), die mit dem Benutzerbetrieb des Betriebselements korrespondieren, synthetisiert werden, digital zu synthetisieren und auszugeben und

wobei in dem zweiten Modus der mindestens eine Prozessor (205) konfiguriert ist, um:

in Übereinstimmung mit einem Benutzerbetrieb auf einem Betriebselement in der Vielzahl von Betriebselementen (101) vorgeschriebene Textdaten (215a) und Tonhöhendaten (215b), die mit dem Benutzerbetrieb des Betriebselements korrespondieren, in das trainierte akustische Modell (306) einzugeben, um das trainierte akustische Modell (306) zu veranlassen, die akustischen Merkmalsdaten (317) als Antwort auf die eingegebenen vorgeschriebenen Textdaten (215a) und eingegebenen Tonhöhendaten (215b) auszugeben, und  
abgeleitete Gesangsstimmendaten (217), die eine Gesangsstimme des Sängers auf der Basis der akustischen Merkmalsdaten (317) ableiten, die mittels des trainierten akustischen Modells (306) als Antwort auf die eingegeben vorgeschriebenen Textdaten (215a) und die eingegebenen Tonhöhendaten (215b) ausgegeben werden, ohne Verwendung von Instrumentenklangwellenformdaten (202), die in Übereinstimmung mit den Tonhöhendaten (215b), die mit dem Benutzerbetrieb des Benutzerelements korrespondieren, synthetisiert werden, digital zu synthetisieren und auszugeben.

2. Elektronisches Musikinstrument nach Anspruch 1, wobei mindestens ein Prozessor konfiguriert ist, um zwischen dem ersten Modus und dem zweiten Modus umzuschalten (320), basierend auf einem Benutzerbetrieb eines Modusauswahlbetriebselements, das in dem elektronischen Musikinstrument bereitgestellt wird.

3. Elektronisches Musikinstrument nach Anspruch 1,

wobei der Speicher (202) konfiguriert ist, um Melodietonhöhendaten (215d), die Betriebselemente anzeigen, die ein Benutzer betreiben soll, Gesangsstimmenausgabetimeingdaten (215c), die Ausgabetimeings anzeigen, bei denen entsprechende Gesangsstimmen für die mittels der Melodietonhöhendaten (215d) angezeigten Tonhöhen ausgegeben werden sollen, und Textdaten (215a), die entsprechend mit den Melodietonhöhendaten (215d) korrespondieren, zu enthalten, und

wobei in dem ersten Modus der mindestens eine Prozessor (205) konfiguriert ist, um:

wenn ein Benutzerbetrieb zum Produzieren einer Gesangsstimme bei einem Ausgabetimeing, das mittels der Gesangsstimmenausgabetimeingdaten (215c) angezeigt wird, durchgeführt wird, Tonhöhendaten (215b), die mit dem benutzerbetriebenen Betriebselement korrespondieren, und Textdaten (215a), die mit dem Ausgabetimeing korrespondieren, an das trainierte akustische Modell (306) einzugeben, und, bei dem Ausgabetimeing, abgeleitete Gesangsstimmendaten (217) auszugeben, die die Gesangsstimme des Sängers ableiten, auf der Basis von dem mindestens einen Abschnitt der akustischen Merkmalsdaten (317), die mittels des trainierten akustischen Modells (306) als Antwort auf die Eingabe ausgegeben werden, und  
wenn ein Benutzerbetrieb zum Produzieren einer Gesangsstimme bei dem Ausgabetimeing, das mittels der Gesangsstimmenausgabetimeingdaten (215c) angezeigt wird, nicht durchgeführt wird, Melodietonhöhendaten (215d), die mit dem Ausgabetimeing korrespondieren, und Textdaten (215a), die mit dem Ausgabetimeing korrespondieren, an das trainierte akustische Modell (306) einzugeben, und, bei dem Ausgabetimeing, abgeleitete Gesangsstimmendaten (217) auszugeben, die die Gesangsstimme des Sängers ableiten, auf der Basis von dem mindestens einen Abschnitt der akustischen Merkmalsdaten (317), die mittels des trainierten akustischen Modells (306) als Antwort auf die Eingabe ausgegeben werden.

4. Elektronisches Musikinstrument nach Anspruch 1,

wobei die akustischen Merkmalsdaten (317) der Gesangsstimme des Sängers Spektraldaten (318), die einen Vokaltrakt des Sängers modellieren, und Klangquellendaten (319) einschließen, die Stimmbänder des Sängers modellieren, und

wobei in dem zweiten Modus der mindestens eine Prozessor (205) konfiguriert ist, um die abgeleiteten Gesangsstimmendaten (217) zu synthetisieren, die die Gesangsstimme des Sängers ableiten, auf der Basis von den Spektraldaten (318) und den Klangquellendaten (319).

5. Elektronisches Musikinstrument nach Anspruch 1, weiter umfassend ein Auswahlbetriebselement (102), das, aus einer Vielzahl von Instrumentenklängen, einschließend mindestens einen aus einem Blechbläserklang, einem Streichinstrumentenklang, einem Orgelklang oder einem Tierschrei, konfiguriert ist, um einen aus den Instrumen-

tenklängen als Antwort auf einen Benutzerbetrieb zu spezifizieren, und wobei in dem ersten Modus die Instrumentenklangwellenformdaten (220) mit dem Instrumentenklang korrespondieren, der mittels des Auswahlbetriebselements spezifiziert wird.

- 5     **6.** Elektronisches Musikinstrument nach Anspruch 1,  
  
wobei die akustischen Merkmalsdaten (317) der Gesangsstimme des Sängers Spektraldaten (318), die einen Vokaltrakt des Sängers modellieren, und Klangquellendaten (319), die die Stimmbänder des Sängers model-  
10     lieren, einschließen, und  
wobei in dem ersten Modus der mindestens eine Prozessor (205) konfiguriert ist, um die abgeleiteten Gesangs-  
stimmendaten (217) zu synthetisieren, die die Gesangsstimme des Sängers ableiten, mittels Anwendung eines  
akustischen Merkmals der Spektraldaten (318) auf die Instrumentenklangwellenformdaten (220) ohne Verwen-  
dung der Klangquellendaten (319) der akustischen Merkmalsdaten (317).
- 15     **7.** Elektronisches Musikinstrument nach Anspruch 1, wobei das trainierte akustische Modell (306) trainiert worden ist  
über maschinelles Lernen (305) unter Verwendung mindestens eines Deep Neural Network oder eines Hidden  
Markov Models.
- 20     **8.** Elektronisches Musikinstrument nach Anspruch 1,  
  
wobei die Vielzahl der Betriebselemente (101) ein erstes Betriebselement als das Betriebselement, das mittels  
des Benutzers betrieben wurde, und ein zweites Betriebselement einschließt, das eine vorgeschriebene Be-  
dingung in Bezug auf das erste Betriebselement erfüllt, und  
wobei in sowohl den ersten als auch den zweiten Modi der mindestens eine Prozessor (205) konfiguriert ist,  
25     um einen akustischen Effekt (322) auf die abgeleiteten Gesangsstimmendaten (217) anzuwenden, wenn das  
zweite Betriebselement betrieben wird, während das erste Betriebselement betrieben wird.
- 30     **9.** Elektronisches Musikinstrument nach Anspruch 8, wobei der mindestens eine Prozessor (205) konfiguriert ist, um  
eine Tiefe des akustischen Effekts (322) in Übereinstimmung mit einem Unterschied in der Tonhöhe (S1111) zwi-  
schen einer Tonhöhe, die mit dem ersten Betriebselement korrespondiert, und einer Tonhöhe, die mit dem zweiten  
Betriebselement korrespondiert, zu ändern.
- 35     **10.** Elektronisches Musikinstrument nach Anspruch 8, wobei das zweite Betriebselement eine schwarze Taste ist.
- 40     **11.** Elektronisches Musikinstrument nach Anspruch 8, wobei der akustische Effekt (322) mindestens einen aus einem  
Vibratoeffekt, einem Tremoloeffekt, oder einem Wah-Wah-Effekt einschließt.
- 45     **12.** Verfahren, das mittels mindestens einem Prozessor (205) in einem elektronischen Musikinstrument durchgeführt  
wird, das zusätzlich zu dem mindestens einen Prozessor (205) einschließt: eine Vielzahl von Betriebselementen  
(101), die entsprechend mit gegenseitig verschiedenen Tonhöhendaten korrespondiert; und einen Speicher (202),  
der ein trainiertes akustisches Modell (306) speichert, das mittels Durchführung von maschinellem Lernen (305)  
auf Trainingsmusikpartiturdaten (311), einschließend Trainingstextdaten (311a) und Trainingstonhöhendaten  
(311b), und auf Trainingsgesangsstimmendaten (312) eines Sängers, die mit den Trainingsmusikpartiturdaten (311)  
korrespondieren, erhalten wurde, wobei das trainierte akustische Modell (306) konfiguriert ist, um Textdaten (215a)  
und vorgeschriebene Tonhöhendaten (215b) zu empfangen und akustische Merkmalsdaten (317) einer Gesangs-  
stimme des Sängers als Antwort auf die empfangenen Textdaten und Tonhöhendaten auszugeben, wobei ein erster  
Modus und ein zweiter Modus in dem mindestens einen Prozessor (205) auswechselbar auswählbar sind, wobei  
das Verfahren über den mindestens einen Prozessor (205) umfasst:  
  
50     Auswählen eines aus dem ersten Modus und dem zweiten Modus als Antwort auf Benutzerbetrieb;  
in dem ersten Modus:  
  
in Übereinstimmung mit einem Benutzerbetrieb auf einem Betriebselement in der Vielzahl von Betriebse-  
lementen (101), Eingeben vorgeschriebener Textdaten (215a) und Tonhöhendaten (215b), die mit dem  
Benutzerbetrieb des Betriebselements korrespondieren, an das trainierte akustische Modell (306), um das  
55     trainierte akustische Modell (306) zu veranlassen, die akustischen Merkmalsdaten (317) als Antwort auf  
die eingegebenen vorgeschriebenen Textdaten (215a) und die eingegebenen Tonhöhendaten (215b) aus-  
zugeben, und

digitales Synthetisieren und Ausgeben abgeleiteter Gesangsstimmendaten (217), die eine Gesangsstimme des Sängers auf der Basis von mindestens einem Abschnitt der akustischen Merkmalsdaten (317) ableiten, die mittels des trainierten akustischen Modells (306) als Antwort auf die eingegebenen vorgeschriebenen Textdaten (215a) und die eingegebenen Tönhöhendaten (215b) ausgegeben werden, und auf der Basis von Instrumentenklangwellenformdaten (220), die in Übereinstimmung mit den Tönhöhendaten (215b), die mit dem Benutzerbetrieb des Betriebselements korrespondieren, synthetisiert werden und

in dem zweiten Modus:

in Übereinstimmung mit einem Benutzerbetrieb auf einem Betriebselement in der Vielzahl von Betriebselementen (101), Eingeben vorgeschriebener Textdaten (215a) und Tönhöhendaten (215b), die mit dem Benutzerbetrieb des Betriebselements korrespondieren, in das trainierte akustische Modell (306), um das trainierte akustische Modell (306) zu veranlassen, die akustischen Merkmalsdaten (317) als Antwort auf die eingegebenen vorgeschriebenen Textdaten (215a) und eingegebenen Tönhöhendaten (215b) auszugeben, und  
 digitales Synthetisieren und Ausgeben abgeleiteter Gesangsstimmendaten (217), die eine Gesangsstimme des Sängers auf der Basis der akustischen Merkmalsdaten (317) ableiten, die mittels des trainierten akustischen Modells (306) als Antwort auf die eingegebenen vorgeschriebenen Textdaten (215a) und die eingegebenen Tönhöhendaten (215b) ausgegeben werden, ohne Verwendung von Instrumentenklangwellenformdaten (220), die in Übereinstimmung mit den Tönhöhendaten (215b), die mit dem Benutzerbetrieb des Benutzerelements korrespondieren, synthetisiert werden.

13. Verfahren nach Anspruch 12, wobei das Verfahren einschließt, über den mindestens einen Prozessor (205), zwischen dem ersten Modus und dem zweiten Modus umzuschalten, basierend auf einem Benutzerbetrieb eines Modusauswahlbetriebselements, das in dem elektronischen Musikinstrument bereitgestellt wird.

14. Verfahren nach Anspruch 12,

wobei der Speicher (202) Melodietönhöhendaten (215d), die Betriebselemente anzeigen, die ein Benutzer betreiben soll, Gesangsstimmenausgabetimingdaten (215c), die Ausgabetimings anzeigen, bei denen entsprechende Gesangsstimmen für die mittels der Melodietönhöhendaten (215d) angezeigten Tönhöhen ausgegeben werden sollen, und Textdaten (215a), die entsprechend mit den Melodietönhöhendaten (215d) korrespondieren, enthält, und

wobei in dem ersten Modus das Verfahren über den mindestens einen Prozessor (205) einschließt:

wenn ein Benutzerbetrieb zum Produzieren einer Gesangsstimme bei einem Ausgabetiming, das mittels der Gesangsstimmenausgabetimingdaten (215c) angezeigt wird, durchgeführt wird, Eingeben von Tönhöhendaten (15b), die mit dem benutzerbetriebenen Betriebselement korrespondieren, und Textdaten (215a), die mit dem Ausgabetiming korrespondieren, an das trainierte akustische Modell (306), und, bei dem Ausgabetiming, Ausgeben von abgeleiteten Gesangsstimmendaten (217), die die Gesangsstimme des Sängers ableiten, auf der Basis von dem mindestens einen Abschnitt der akustischen Merkmalsdaten (317), die mittels des trainierten akustischen Modells (306) als Antwort auf die Eingabe ausgegeben werden, und  
 wenn ein Benutzerbetrieb zum Produzieren einer Gesangsstimme bei dem Ausgabetiming, das mittels der Gesangsstimmenausgabetimingdaten (215c) angezeigt wird, nicht durchgeführt wird, Eingeben von Melodietönhöhendaten (215d), die mit dem Ausgabetiming korrespondieren, und Textdaten (215a), die mit dem Ausgabetiming korrespondieren, an das trainierte akustische Modell (306), und, bei dem Ausgabetiming, Ausgeben von abgeleiteten Gesangsstimmendaten (217), die die Gesangsstimme des Sängers ableiten, auf der Basis von dem mindestens einen Abschnitt der akustischen Merkmalsdaten (317), die mittels des trainierten akustischen Modells (306) als Antwort auf die Eingabe ausgegeben werden.

15. Verfahren nach Anspruch 12,

wobei die akustischen Merkmalsdaten (317) der Gesangsstimme des Sängers Spektraldaten (318), die einen Vokaltrakt des Sängers modellieren, und Klangquellendaten (319), die Stimmbänder des Sängers modellieren, einschließen, und

wobei das Verfahren in dem zweiten Modus einschließt, den mindestens einen Prozessor (205) zu veranlassen, die abgeleiteten Gesangsstimmendaten (217) zu synthetisieren, die die Gesangsstimme des Sängers ableiten, auf der Basis von den Spektraldaten (318) und den Klangquellendaten (319).

## 16. Verfahren nach Anspruch 12,

wobei das elektronische Musikinstrument weiter ein Auswahlbetriebselement (102) einschließt, das, aus einer Vielzahl von Instrumentenklängen, einschließlich mindestens einen aus einem Blechbläserklang, einem Streichinstrumentenklang, einem Orgelklang oder einem Tierschrei, konfiguriert ist, um einen aus den Instrumentenklängen als Antwort auf einen Benutzerbetrieb zu spezifizieren, und wobei in dem ersten Modus die Instrumentenklangwellenformdaten (220) mit dem Instrumentenklang korrespondieren, der mittels des Auswahlbetriebselements (102) spezifiziert wird.

## 17. Verfahren nach Anspruch 12,

wobei die akustischen Merkmalsdaten (317) der Gesangsstimme des Sängers Spektraldaten (318), die einen Vokaltrakt des Sängers modellieren, und Klangquellendaten (319), die die Stimmbänder des Sängers modellieren, einschließen, und

wobei in dem ersten Modus die abgeleiteten Gesangsstimmendaten (217), die die Gesangsstimme des Sängers ableiten, mittels Anwendung eines akustischen Merkmals der Spektraldaten (318) auf die Instrumentenklangwellenformdaten (220) ohne Verwendung der Klangquellendaten (319) der akustischen Merkmalsdaten (317) synthetisiert werden.

18. Nicht transitorisches computerlesbares Speichermedium, das ein darauf gespeichertes Programm aufweist, das mittels mindestens eines Prozessors (205) in einem elektronischen Musikinstrument ausführbar ist, das zusätzlich zu dem mindestens einen Prozessor (205) einschließt: eine Vielzahl von Betriebselementen (101), die entsprechend mit gegenseitig verschiedenen Tonhöhendaten korrespondieren; und einen Speicher (202), der ein trainiertes akustisches Modell (306) speichert, das mittels Durchführen von maschinellem Lernen (305) auf Trainingsmusikpartiturdaten (311), einschließlich Trainingstextdaten (311a) und Trainingstonhöhendaten (311b), und auf Trainingsgesangsstimmendaten (312) eines Sängers, die mit den Trainingsmusikpartiturdaten (311) korrespondieren, erhalten wird, wobei das trainierte akustische Modell (306) konfiguriert ist, um Textdaten (215a) und vorgeschriebene Tonhöhendaten (215b) zu empfangen und akustische Merkmalsdaten (317) einer Gesangsstimme des Sängers als Antwort auf die empfangenen Textdaten und Tonhöhendaten auszugeben, wobei ein erster Modus und ein zweiter Modus in dem mindestens einen Prozessor (205) auswechselbar auswählbar sind, wobei das Programm den mindestens einen Prozessor (205) veranlasst, folgendes durchzuführen:

Auswählen eines aus dem ersten Modus und dem zweiten Modus als Antwort auf Benutzerbetrieb; in dem ersten Modus:

in Übereinstimmung mit einem Benutzerbetrieb auf einem Betriebselement in der Vielzahl von Betriebselementen (101), Eingeben vorgeschriebener Textdaten (215a) und Tonhöhendaten (215b), die mit dem Benutzerbetrieb des Betriebselements korrespondieren, in das trainierte akustische Modell (306), um das trainierte akustische Modell (306) zu veranlassen, die akustischen Merkmalsdaten (317) als Antwort auf die eingegebenen vorgeschriebenen Textdaten (215a) und die eingegebenen Tonhöhendaten (215b) auszugeben, und digitales Synthetisieren und Ausgeben abgeleiteter Gesangsstimmendaten (217), die eine Gesangsstimme des Sängers auf der Basis von mindestens einem Abschnitt der akustischen Merkmalsdaten (317) ableiten, die mittels des trainierten akustischen Modells (306) als Antwort auf die eingegebenen vorgeschriebenen Textdaten (215a) und die eingegebenen Tonhöhendaten (215b) ausgegeben werden, und auf der Basis von Instrumentenklangwellenformdaten (220), die in Übereinstimmung mit den Tonhöhendaten (215b), die mit dem Benutzerbetrieb des Betriebselements korrespondieren, synthetisiert werden, und

in dem zweiten Modus:

in Übereinstimmung mit einem Benutzerbetrieb auf einem Betriebselement in der Vielzahl von Betriebselementen (101), Eingeben vorgeschriebener Textdaten (215a) und Tonhöhendaten (215b), die mit dem Benutzerbetrieb auf dem Betriebselement korrespondieren, in das trainierte akustische Modell (306), um das trainierte akustische Modell (306) zu veranlassen, die akustischen Merkmalsdaten (317) als Antwort auf die eingegebenen vorgeschriebenen Textdaten (215a) und eingegebenen Tonhöhendaten (215b) auszugeben, und digitales Synthetisieren und Ausgeben abgeleiteter Gesangsstimmendaten (217), die eine Gesangsstimme des Sängers auf der Basis der akustischen Merkmalsdaten (317) ableiten, die mittels des trainierten akus-

tischen Modells (306) als Antwort auf die eingegeben vorgeschriebenen Textdaten (215a) und die eingegebenen Tonhöhendaten (215b) ausgegeben werden, ohne Verwendung von Instrumentenklangwellenformdaten (220), die in Übereinstimmung mit den Tonhöhendaten (215b), die mit dem Benutzerbetrieb des Benutzerelements korrespondieren, synthetisiert werden.

## Revendications

### 1. Instrument musical électronique comprenant :

une pluralité d'éléments opérationnels (101) correspondant respectivement à des données de hauteur de son différentes les unes des autres ;  
 une mémoire (202) configurée pour stocker un modèle acoustique entraîné (306), obtenu par exécution d'apprentissage automatique (305) sur des données de partition musicale d'apprentissage (311) incluant des données lyriques d'apprentissage (311a) et des données de hauteur de son d'apprentissage (311b), et sur des données de voix chantée d'apprentissage (312) d'un(e) chanteur(se) correspondant aux données de partition musicale d'apprentissage (311), le modèle acoustique entraîné (306) étant configuré pour recevoir des données lyriques (215a) et des données de hauteur de son (215b) prescrites et fournir des données de caractéristiques acoustiques (317) d'une voix chantée du chanteur/de la chanteuse en réponse aux données lyriques et données de hauteur de son reçues ; et  
 au moins un processeur (205) dans lequel un premier mode et un second mode sont sélectionnables de façon interchangeable,  
 dans lequel, dans le premier mode, ledit au moins un processeur (205) est configuré pour :

conformément à un actionnement par l'utilisateur d'un élément opérationnel dans la pluralité d'éléments opérationnels (101), entrer dans le modèle acoustique entraîné (306) des données lyriques (215a) prescrites et des données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel de manière à amener le modèle acoustique entraîné (306) à fournir les données de caractéristiques acoustiques (317) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, et  
 synthétiser numériquement et fournir des données de voix chantée déduites (217) qui déduisent une voix chantée du chanteur/de la chanteuse sur la base d'au moins une partie des données de caractéristiques acoustiques (317) fournies par le modèle acoustique entraîné (306) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, et sur la base de données de formes d'ondes (220) de son d'instrument qui sont synthétisées conformément aux données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel, et

dans lequel, dans le second mode, ledit au moins un processeur (205) est configuré pour :

conformément à un actionnement par l'utilisateur d'un élément opérationnel dans la pluralité d'éléments opérationnels (101), entrer dans le modèle acoustique entraîné (306) des données lyriques (215a) prescrites et des données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel de manière à amener le modèle acoustique entraîné (306) à fournir les données de caractéristiques acoustiques (317) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, et  
 synthétiser numériquement et fournir des données de voix chantée déduites (217) qui déduisent une voix chantée du chanteur/de la chanteuse sur la base des données de caractéristiques acoustiques (317) fournies par le modèle acoustique entraîné (306) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, sans utiliser de données de formes d'ondes (202) de son d'instrument qui sont synthétisées conformément aux données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel.

2. Instrument musical électronique selon la revendication 1, dans lequel ledit au moins un processeur est configuré pour basculer (320) entre le premier mode et le second mode sur la base d'un actionnement par l'utilisateur d'un élément opérationnel de sélection de mode fourni dans l'instrument musical électronique.

3. Instrument musical électronique selon la revendication 1,

dans lequel la mémoire (202) est configurée pour contenir des données de hauteur de son mélodique (215d) indiquant les éléments opérationnels qu'un utilisateur doit actionner, des données de positionnement temporel (215c) de sortie de voix chantée indiquant les positionnements temporels de sortie en lesquels les voix chantées respectives pour les hauteurs de son indiquées par les données de hauteur de son mélodique (215d) doivent être émises, et des données lyriques (215a) correspondant respectivement aux données de hauteur de son mélodique (215d), et

dans lequel, dans le premier mode, ledit au moins un processeur (205) est configuré pour :

lorsqu'un actionnement par l'utilisateur pour produire une voix chantée est exécuté en un positionnement temporel de sortie indiqué par les données de positionnement temporel (215c) de sortie de voix chantée, entrer dans le modèle acoustique entraîné (306) des données de hauteur de son (215b) correspondant à l'élément opérationnel actionné par l'utilisateur et des données lyriques (215a) correspondant audit positionnement temporel de sortie, et fournir, en ledit positionnement temporel de sortie, des données de voix chantée déduites (217) qui déduisent la voix chantée du chanteur/de la chanteuse sur la base de ladite au moins une partie des données de caractéristiques acoustiques (317) fournies par le modèle acoustique entraîné (306) en réponse à l'entrée, et

lorsqu'un actionnement par l'utilisateur pour produire une voix chantée n'est pas exécuté en le positionnement temporel de sortie indiqué par les données de positionnement temporel (215c) de sortie de voix chantée, entrer dans le modèle acoustique entraîné (306) des données de hauteur de son mélodique (215d) correspondant audit positionnement temporel de sortie et des données lyriques (215a) correspondant audit positionnement temporel de sortie, et fournir, en ledit positionnement temporel de sortie, des données de voix chantée déduites (217) qui déduisent la voix chantée du chanteur/de la chanteuse sur la base de ladite au moins une partie des données de caractéristiques acoustiques (317) fournies par le modèle acoustique entraîné (306) en réponse à l'entrée.

#### 4. Instrument musical électronique selon la revendication 1,

dans lequel les données de caractéristiques acoustiques (317) de la voix chantée du chanteur/de la chanteuse incluent des données spectrales (318) qui modélisent un conduit vocal du chanteur/de la chanteuse et des données de source sonore (319) qui modélisent les cordes vocales du chanteur/de la chanteuse, et dans lequel, dans le second mode, ledit au moins un processeur (205) est configuré pour synthétiser les données de voix chantée déduites (217) qui déduisent la voix chantée du chanteur/de la chanteuse sur la base des données spectrales (318) et des données de source sonore (319).

#### 5. Instrument musical électronique selon la revendication 1, comprenant en outre un élément opérationnel de sélection (102) qui, parmi une pluralité de sons d'instruments incluant au moins un parmi un son d'instrument de la famille des cuivres, un son d'instrument à cordes, un son d'orgue, ou un cri d'animal, est configuré pour spécifier l'un des sons d'instruments en réponse à un actionnement par l'utilisateur, et dans lequel, dans le premier mode, les données de formes d'ondes (220) de son d'instrument correspondent au son d'instrument spécifié par l'élément opérationnel de sélection.

#### 6. Instrument musical électronique selon la revendication 1,

dans lequel les données de caractéristiques acoustiques (317) de la voix chantée du chanteur/de la chanteuse incluent des données spectrales (318) qui modélisent un conduit vocal du chanteur/de la chanteuse et des données de source sonore (319) qui modélisent les cordes vocales du chanteur/de la chanteuse, et dans lequel, dans le premier mode, ledit au moins un processeur (205) est configuré pour synthétiser les données de voix chantée déduites (217) qui déduisent la voix chantée du chanteur/de la chanteuse par application d'une caractéristique acoustique des données spectrales (318) aux données de formes d'ondes (220) de son d'instrument sans utiliser les données de source sonore (319) des données de caractéristiques acoustiques (317).

#### 7. Instrument musical électronique selon la revendication 1, dans lequel le modèle acoustique entraîné (306) a été entraîné par apprentissage automatique (305) utilisant au moins un parmi un réseau de neurones profonds ou un modèle de Markov caché.

#### 8. Instrument musical électronique selon la revendication 1,

dans lequel la pluralité d'éléments opérationnels (101) incluent un premier élément opérationnel en tant que l'élément opérationnel qui a été actionné par l'utilisateur et un second élément opérationnel qui remplit une condition prescrite en ce qui concerne le premier élément opérationnel, et  
 dans lequel dans l'un et l'autre des premier et second modes, ledit au moins un processeur (205) est configuré pour appliquer un effet acoustique (322) aux données de voix chantée déduites (217) lorsque le second élément opérationnel est actionné tandis que le premier élément opérationnel est en train d'être actionné.

9. Instrument musical électronique selon la revendication 8, dans lequel ledit au moins un processeur (205) est configuré pour modifier une profondeur de l'effet acoustique (322) conformément à une différence de hauteur de son (S1111) entre une hauteur de son correspondant au premier élément opérationnel et une hauteur de son correspondant au second élément opérationnel.

10. Instrument musical électronique selon la revendication 8, dans lequel le second élément opérationnel est une touche noire.

11. Instrument musical électronique selon la revendication 8, dans lequel l'effet acoustique (322) inclut au moins un effet de vibrato, un effet de trémolo, ou un effet wah-wah.

12. Procédé exécuté par au moins un processeur (205) dans un instrument musical électronique qui inclut, en plus dudit au moins un processeur (205) : une pluralité d'éléments opérationnels (101) correspondant respectivement à des données de hauteur de son différentes les unes des autres ; et une mémoire (202) qui stocke un modèle acoustique entraîné (306), obtenu par exécution d'apprentissage automatique (305) sur des données de partition musicale d'apprentissage (311) incluant des données lyriques d'apprentissage (311a) et des données de hauteur de son d'apprentissage (311b), et sur des données de voix chantée d'apprentissage (312) d'un(e) chanteur(se) correspondant aux données de partition musicale d'apprentissage (311), le modèle acoustique entraîné (306) étant configuré pour recevoir des données lyriques (215a) et des données de hauteur de son (215b) prescrites et fournir des données de caractéristiques acoustiques (317) d'une voix chantée du chanteur/de la chanteuse en réponse aux données lyriques et données de hauteur de son reçues, un premier mode et un second mode étant sélectionnables de façon interchangeable dans ledit au moins un processeur (205), le procédé comprenant, via ledit au moins un processeur (205) :

sélectionner l'un parmi le premier mode et le second mode en réponse à un actionnement par l'utilisateur ;  
 dans le premier mode :

conformément à un actionnement par l'utilisateur d'un élément opérationnel dans la pluralité d'éléments opérationnels (101), entrer dans le modèle acoustique entraîné (306) des données lyriques (215a) prescrites et des données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel de manière à amener le modèle acoustique entraîné (306) à fournir les données de caractéristiques acoustiques (317) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, et  
 synthétiser numériquement et fournir des données de voix chantée déduites (217) qui déduisent une voix chantée du chanteur/de la chanteuse sur la base d'au moins une partie des données de caractéristiques acoustiques (317) fournies par le modèle acoustique entraîné (306) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, et sur la base de données de formes d'ondes (220) de son d'instrument qui sont synthétisées conformément aux données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel, et

dans le second mode :

conformément à un actionnement par l'utilisateur d'un élément opérationnel dans la pluralité d'éléments opérationnels (101), entrer dans le modèle acoustique entraîné (306) des données lyriques (215a) prescrites et des données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel de manière à amener le modèle acoustique entraîné (306) à fournir les données de caractéristiques acoustiques (317) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, et  
 synthétiser numériquement et fournir des données de voix chantée déduites (217) qui déduisent une voix chantée du chanteur/de la chanteuse sur la base des données de caractéristiques acoustiques (317) fournies par le modèle acoustique entraîné (306) en réponse aux données lyriques (215a) prescrites entrées

et aux données de hauteur de son (215b) entrées, sans utiliser de données de formes d'ondes (220) de son d'instrument qui sont synthétisées conformément aux données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel.

5 13. Procédé selon la revendication 12, dans lequel le procédé inclut, via ledit au moins un processeur (205), basculer entre le premier mode et le second mode sur la base d'un actionnement par l'utilisateur d'un élément opérationnel de sélection de mode fourni dans l'instrument musical électronique.

10 14. Procédé selon la revendication 12,

dans lequel la mémoire (202) contient des données de hauteur de son mélodique (215d) indiquant les éléments opérationnels qu'un utilisateur doit actionner, des données de positionnement temporel (215c) de sortie de voix chantée indiquant les positionnements temporels de sortie en lesquels les voix chantées respectives pour les hauteurs de son indiquées par les données de hauteur de son mélodique (215d) doivent être émises, et des données lyriques (215a) correspondant respectivement aux données de hauteur de son mélodique (215d), et dans lequel, dans le premier mode, le procédé inclut, via ledit au moins un processeur (205) :

lorsqu'un actionnement par l'utilisateur pour produire une voix chantée est exécuté en un positionnement temporel de sortie indiqué par les données de positionnement temporel (215c) de sortie de voix chantée, entrer dans le modèle acoustique entraîné (306) des données de hauteur de son (215b) correspondant à l'élément opérationnel actionné par l'utilisateur et des données lyriques (215a) correspondant audit positionnement temporel de sortie, et fournir, en ledit positionnement temporel de sortie, des données de voix chantée déduites (217) qui déduisent la voix chantée du chanteur/de la chanteuse sur la base de ladite au moins une partie des données de caractéristiques acoustiques (317) fournies par le modèle acoustique entraîné (306) en réponse à l'entrée, et

lorsqu'un actionnement par l'utilisateur pour produire une voix chantée n'est pas exécuté en le positionnement temporel de sortie indiqué par les données de positionnement temporel (215c) de sortie de voix chantée, entrer dans le modèle acoustique entraîné (306) des données de hauteur de son mélodique (215d) correspondant audit positionnement temporel de sortie et des données lyriques (215a) correspondant audit positionnement temporel de sortie, et fournir, en ledit positionnement temporel de sortie, des données de voix chantée déduites (217) qui déduisent la voix chantée du chanteur/de la chanteuse sur la base de ladite au moins une partie des données de caractéristiques acoustiques (317) fournies par le modèle acoustique entraîné (306) en réponse à l'entrée.

35 15. Procédé selon la revendication 12,

dans lequel les données de caractéristiques acoustiques (317) de la voix chantée du chanteur/de la chanteuse incluent des données spectrales (318) qui modélisent un conduit vocal du chanteur/de la chanteuse et des données de source sonore (319) qui modélisent les cordes vocales du chanteur/de la chanteuse, et dans lequel le procédé inclut, dans le second mode, amener ledit au moins un processeur (205) à synthétiser les données de voix chantée déduites (217) qui déduisent la voix chantée du chanteur/de la chanteuse sur la base des données spectrales (318) et des données de source sonore (319).

45 16. Procédé selon la revendication 12,

dans lequel l'instrument musical électronique comprend en outre un élément opérationnel de sélection (102) qui, parmi une pluralité de sons d'instruments incluant au moins un parmi un son d'instrument de la famille des cuivres, un son d'instrument à cordes, un son d'orgue, ou un cri d'animal, spécifie l'un des sons d'instruments en réponse à un actionnement par l'utilisateur, et

dans lequel, dans le premier mode, les données de formes d'ondes (220) de son d'instrument correspondent au son d'instrument spécifié par l'élément opérationnel de sélection (102).

17. Procédé selon la revendication 12,

dans lequel les données de caractéristiques acoustiques (317) de la voix chantée du chanteur/de la chanteuse incluent des données spectrales (318) qui modélisent un conduit vocal du chanteur/de la chanteuse et des données de source sonore (319) qui modélisent les cordes vocales du chanteur/de la chanteuse, et dans lequel, dans le premier mode, les données de voix chantée déduites (217) qui déduisent la voix chantée



du chanteur/de la chanteuse sont synthétisées par application d'une caractéristique acoustique des données spectrales (318) aux données de formes d'ondes (220) de son d'instrument sans utiliser les données de source sonore (319) des données de caractéristiques acoustiques (317).

5 **18.** Support d'informations lisible par ordinateur, non temporaire, sur lequel est enregistré un programme exécutable par au moins un processeur (205) dans un instrument musical électronique qui inclut, en plus dudit au moins un processeur (205) : une pluralité d'éléments opérationnels (101) correspondant respectivement à des données de hauteur de son différentes les unes des autres ; et une mémoire (202) qui stocke un modèle acoustique entraîné (306), obtenu par exécution d'apprentissage automatique (305) sur des données de partition musicale d'apprentissage (311) incluant des données lyriques d'apprentissage (311a) et des données de hauteur de son d'apprentissage (311b), et sur des données de voix chantée d'apprentissage (312) d'un(e) chanteur(se) correspondant aux données de partition musicale d'apprentissage (311), le modèle acoustique entraîné (306) étant configuré pour recevoir des données lyriques (215a) et des données de hauteur de son (215b) prescrites et fournir des données de caractéristiques acoustiques (317) d'une voix chantée du chanteur/de la chanteuse en réponse aux données lyriques et données de hauteur de son reçues, un premier mode et un second mode étant sélectionnables de façon interchangeable dans ledit au moins un processeur (205), le programme amenant ledit au moins un processeur (205) à exécuter ce qui suit :

sélectionner l'un parmi le premier mode et le second mode en réponse à un actionnement par l'utilisateur ;  
dans le premier mode :

conformément à un actionnement par l'utilisateur d'un élément opérationnel dans la pluralité d'éléments opérationnels (101), entrer dans le modèle acoustique entraîné (306) des données lyriques (215a) prescrites et des données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel de manière à amener le modèle acoustique entraîné (306) à fournir les données de caractéristiques acoustiques (317) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, et

synthétiser numériquement et fournir des données de voix chantée déduites (217) qui déduisent une voix chantée du chanteur/de la chanteuse sur la base d'au moins une partie des données de caractéristiques acoustiques (317) fournies par le modèle acoustique entraîné (306) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, et sur la base de données de formes d'ondes (220) de son d'instrument qui sont synthétisées conformément aux données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel, et

dans le second mode :

conformément à un actionnement par l'utilisateur d'un élément opérationnel dans la pluralité d'éléments opérationnels (101), entrer dans le modèle acoustique entraîné (306) des données lyriques (215a) prescrites et des données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel de manière à amener le modèle acoustique entraîné (306) à fournir les données de caractéristiques acoustiques (317) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, et

synthétiser numériquement et fournir des données de voix chantée déduites (217) qui déduisent une voix chantée du chanteur/de la chanteuse sur la base des données de caractéristiques acoustiques (317) fournies par le modèle acoustique entraîné (306) en réponse aux données lyriques (215a) prescrites entrées et aux données de hauteur de son (215b) entrées, sans utiliser de données de formes d'ondes (220) de son d'instrument qui sont synthétisées conformément aux données de hauteur de son (215b) correspondant à l'actionnement par l'utilisateur de l'élément opérationnel.

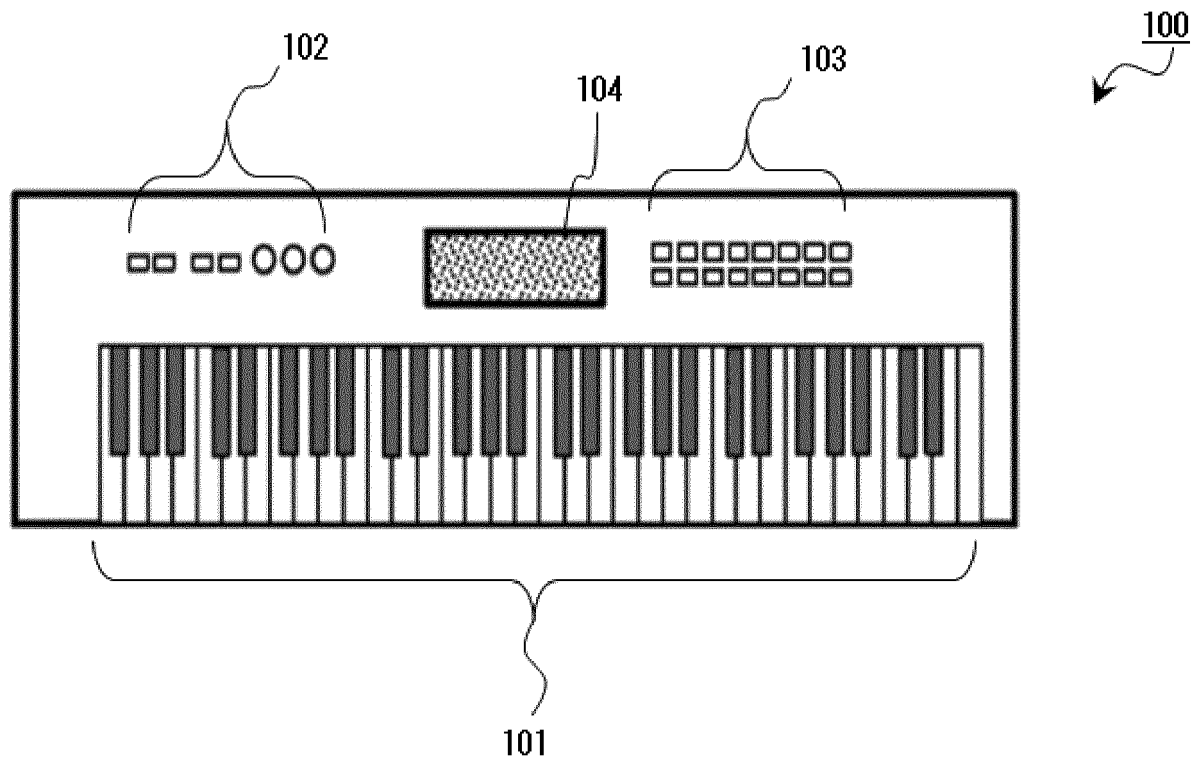


FIG. 1

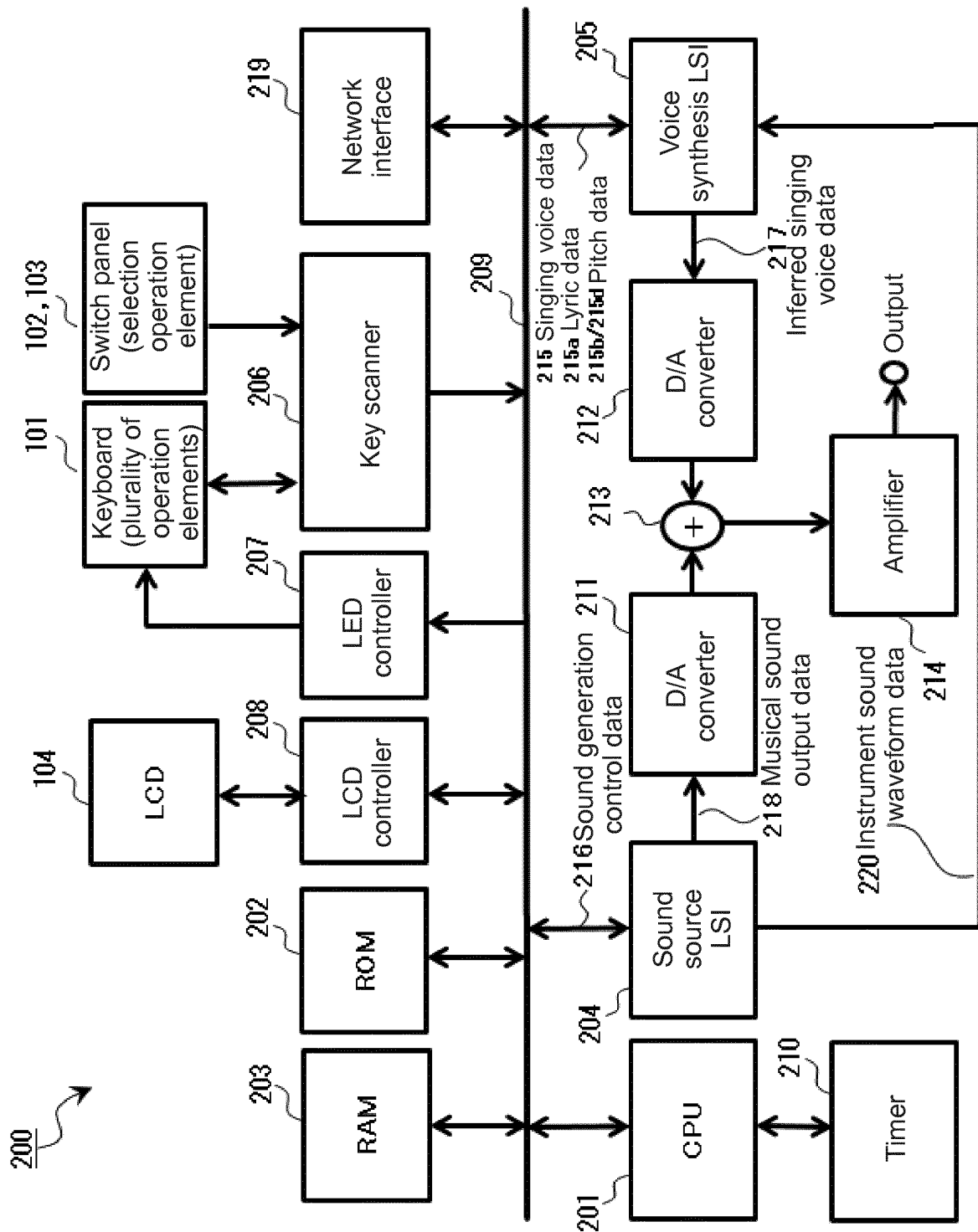


FIG. 2

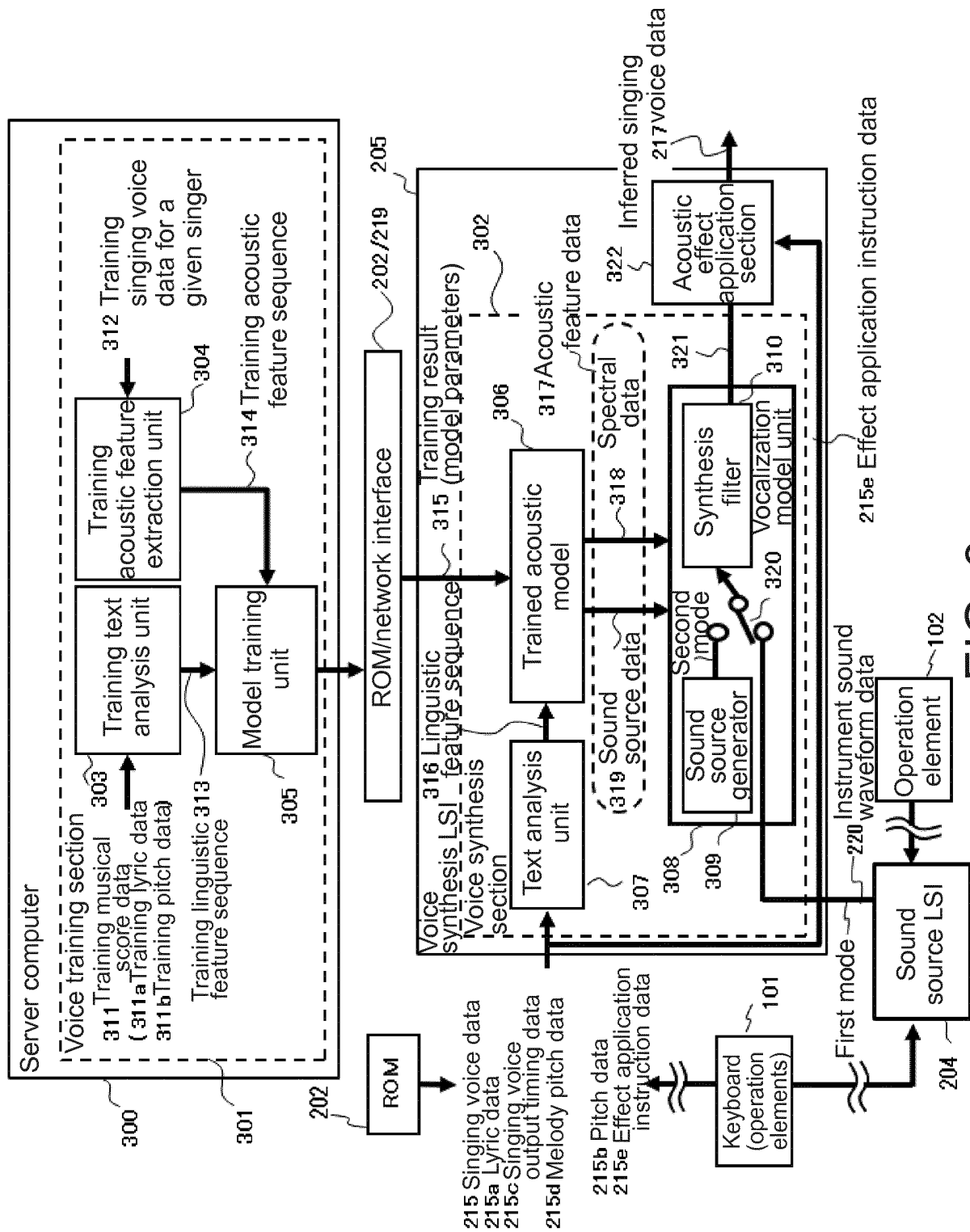


FIG. 3

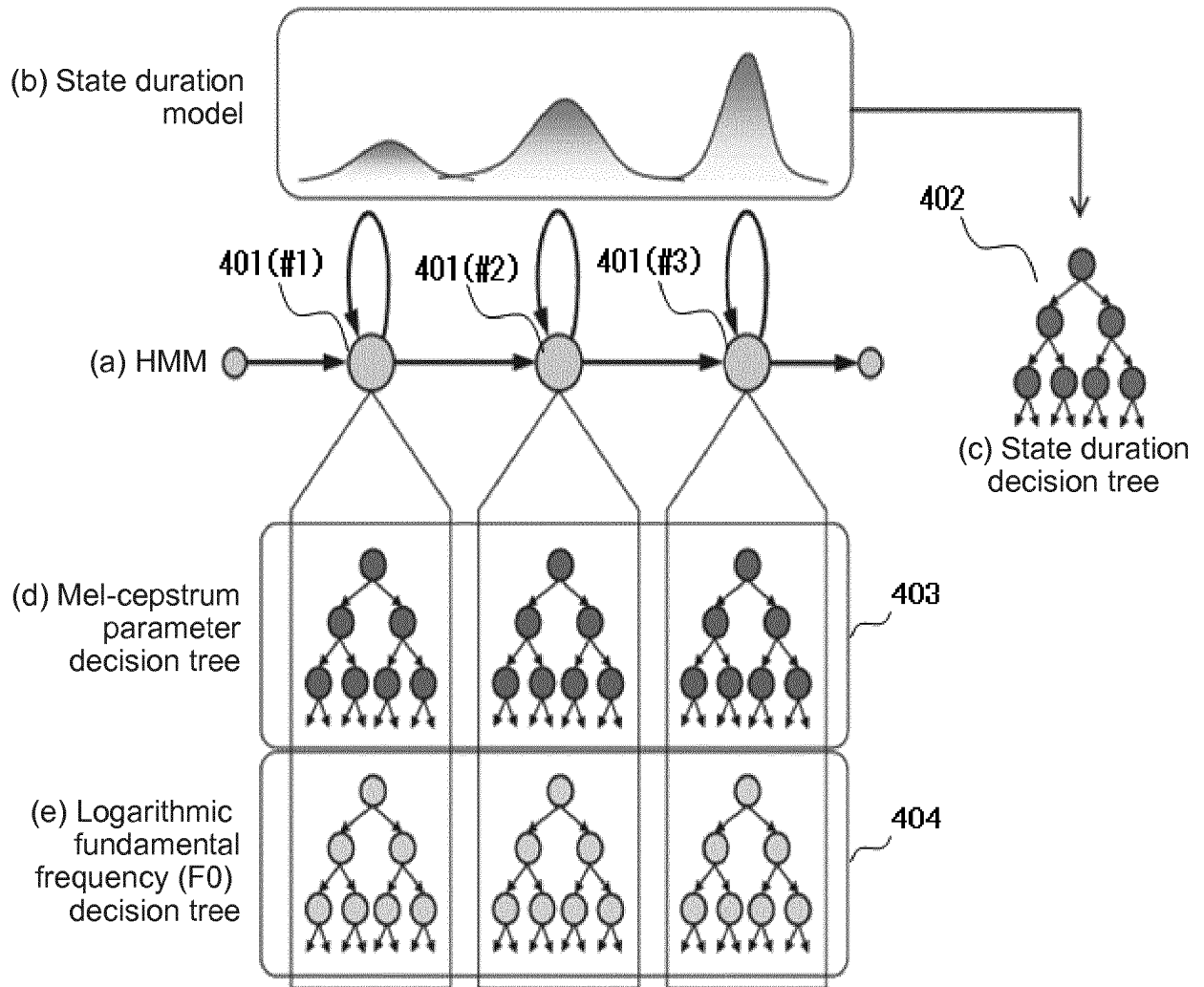


FIG. 4

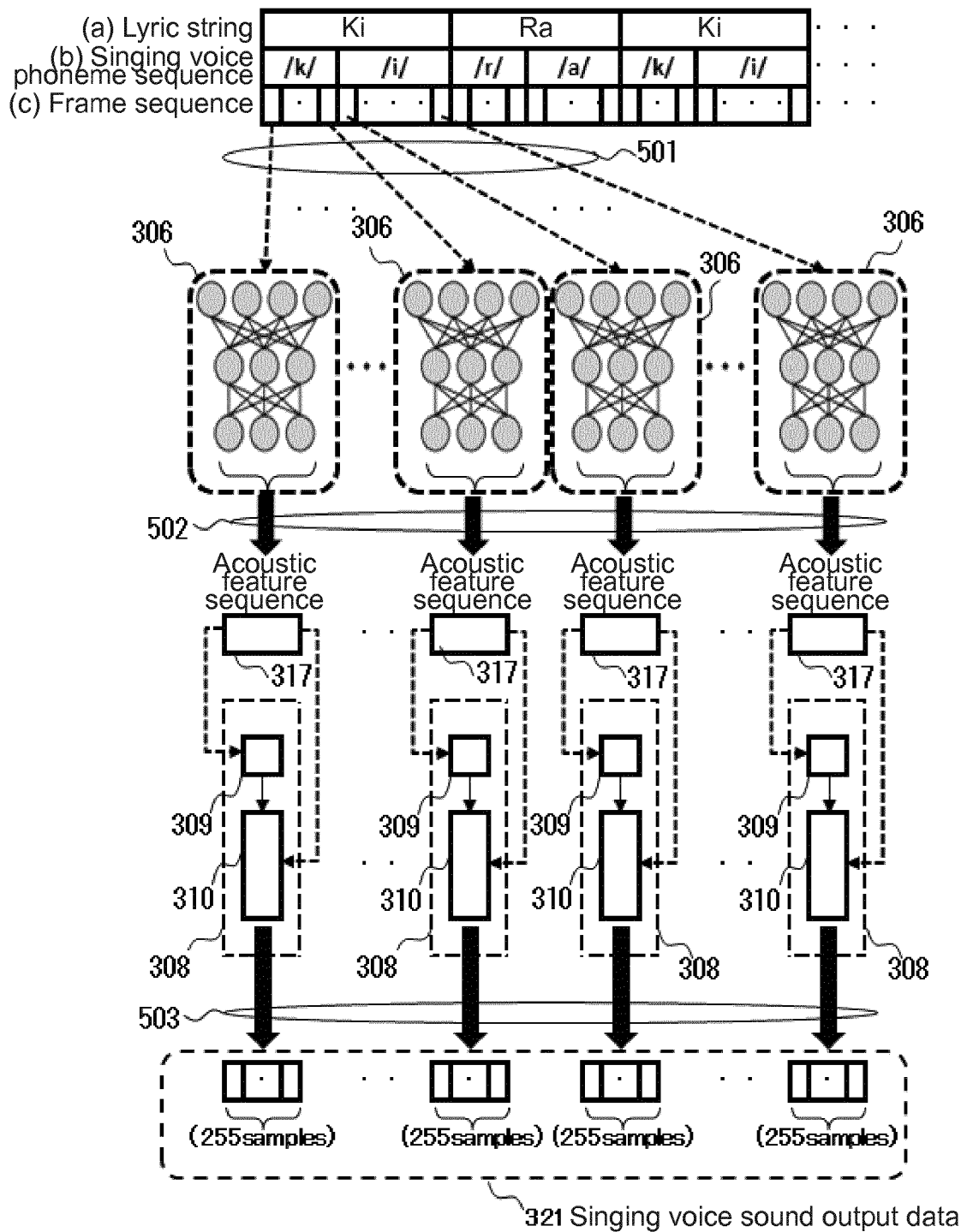


FIG. 5

Header chunk		<b>ChunkID</b>	Fixed string "MThd"=0x4d546864
		<b>ChunkSize</b>	Length of header chunk =0x00000006
		<b>FormatType</b>	Format: e.g., 0x0001
		<b>NumberOfTrack</b>	Number of tracks: e.g., 0x0002
		<b>TimeDivision</b>	Timebase: e.g., 0x1e0
First track chunk (lyric part)	<b>DeltaTime_1[0]</b> <b>Event_1[0]</b>	<b>ChunkID</b>	Fixed string "MTrk"=0x4d54726b
		<b>ChunkSize_1</b>	Length of first track chunk
	<b>DeltaTime_1[1]</b> <b>Event_1[1]</b>	<b>DeltaTime</b>	Wait time since last event
		<b>Event</b>	Event
	<b>DeltaTime_1[1]</b> <b>Event_1[1]</b>	<b>DeltaTime</b>	Wait time since last event
		<b>Event</b>	Event
	. . .		
	<b>DeltaTime_1[L-1]</b> <b>Event_1[L-1]</b>		
	<b>DeltaTime_1[L-1]</b> <b>Event_1[L-1]</b>	<b>DeltaTime</b>	Wait time since last event
		<b>Event</b>	Must have "End of Track" at end of track
Second track chunk (accompaniment part)	<b>DeltaTime_2[0]</b> <b>Event_2[0]</b>	<b>ChunkID</b>	Fixed string "MTrk"=0x4d54726b
		<b>ChunkSize_2</b>	Length of second track chunk
	<b>DeltaTime_2[1]</b> <b>Event_2[1]</b>	<b>DeltaTime</b>	Wait time since last event
		<b>Event</b>	Event
	<b>DeltaTime_2[1]</b> <b>Event_2[1]</b>	<b>DeltaTime</b>	Wait time since last event
		<b>Event</b>	Event
	. . .		
	<b>DeltaTime_2[M-1]</b> <b>Event_2[M-1]</b>		
	<b>DeltaTime_2[M-1]</b> <b>Event_2[M-1]</b>	<b>DeltaTime</b>	Wait time since last event
		<b>Event</b>	Must have "End of Track" at end of track

FIG. 6

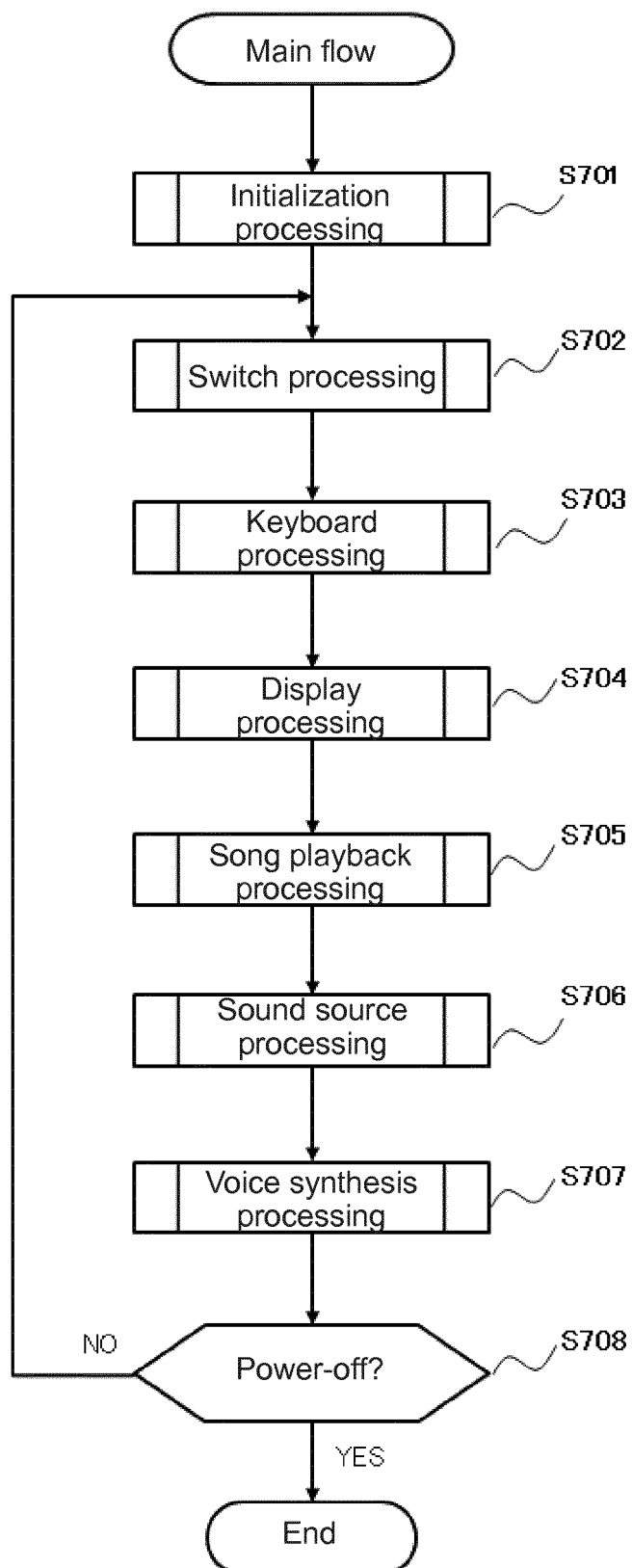


FIG. 7



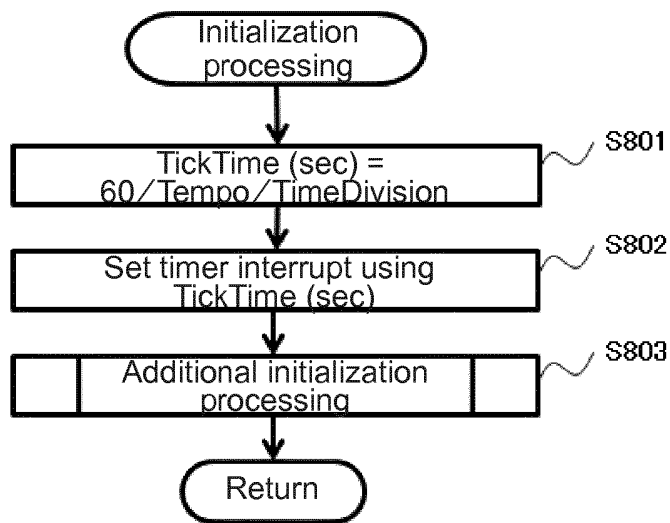


FIG. 8A

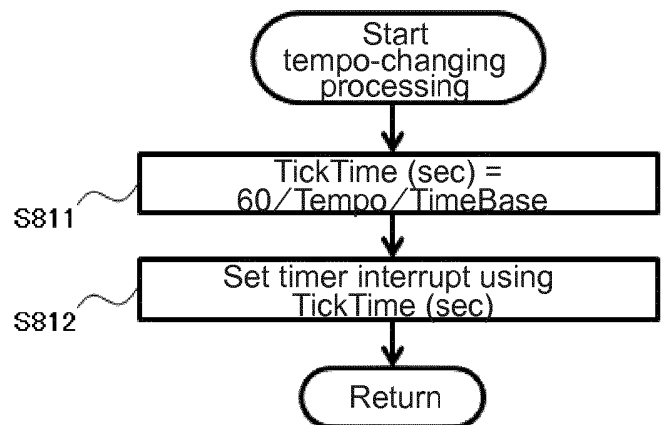


FIG. 8B

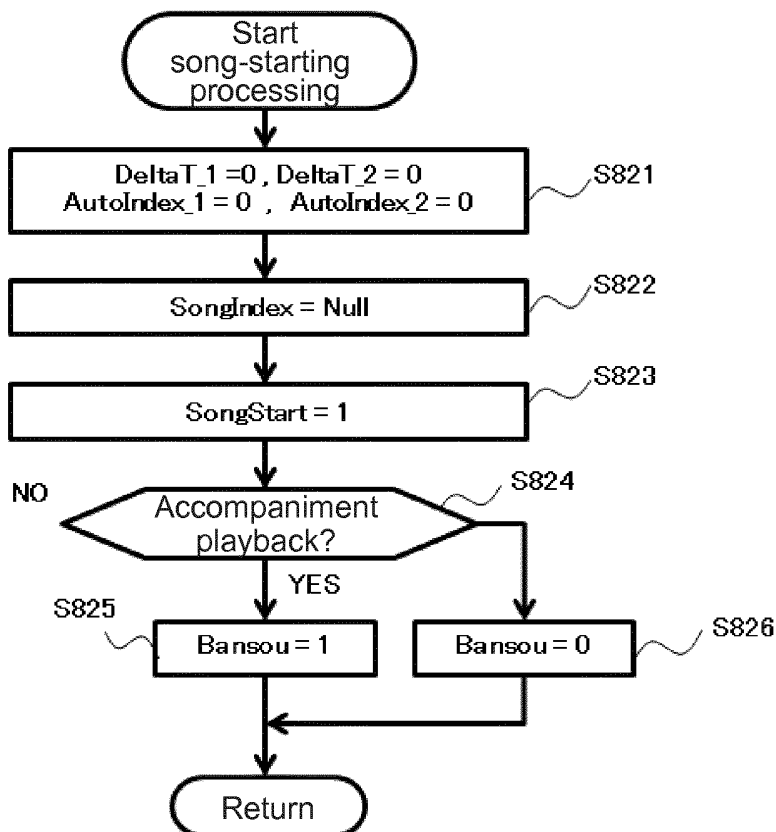


FIG. 8C

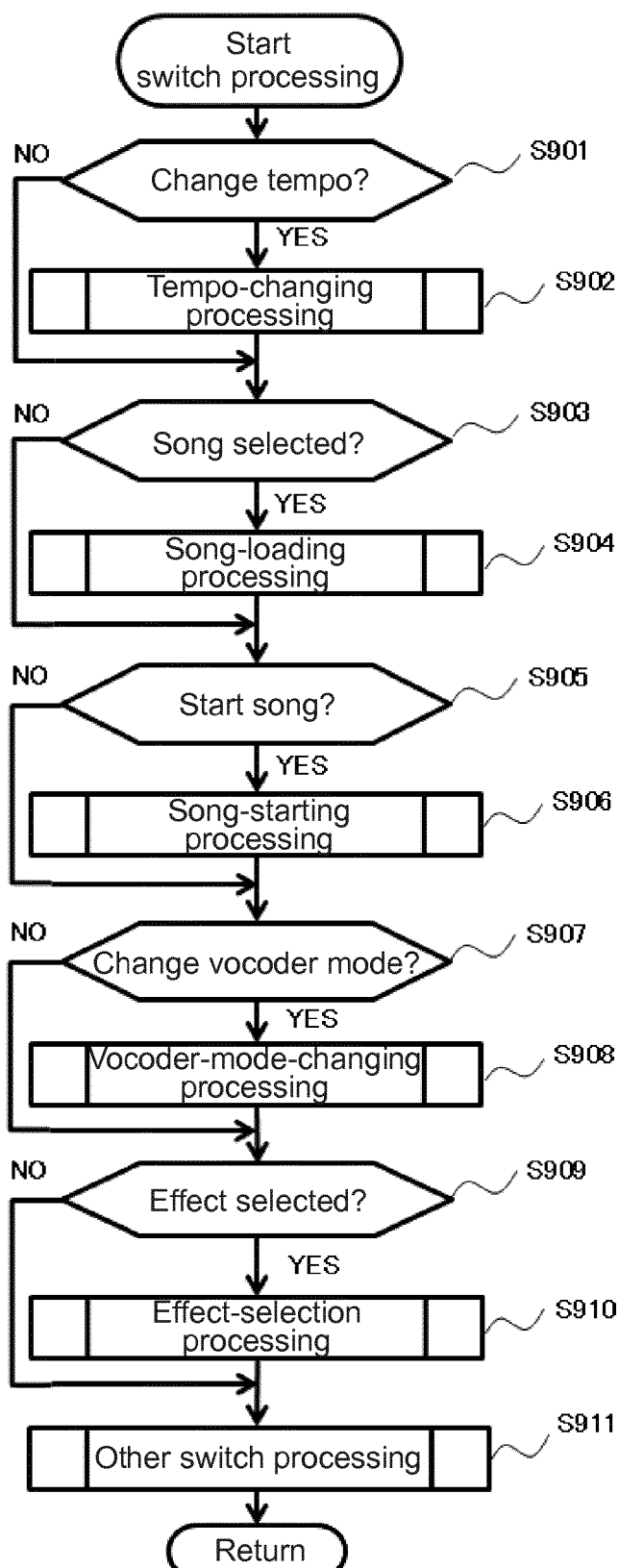


FIG. 9

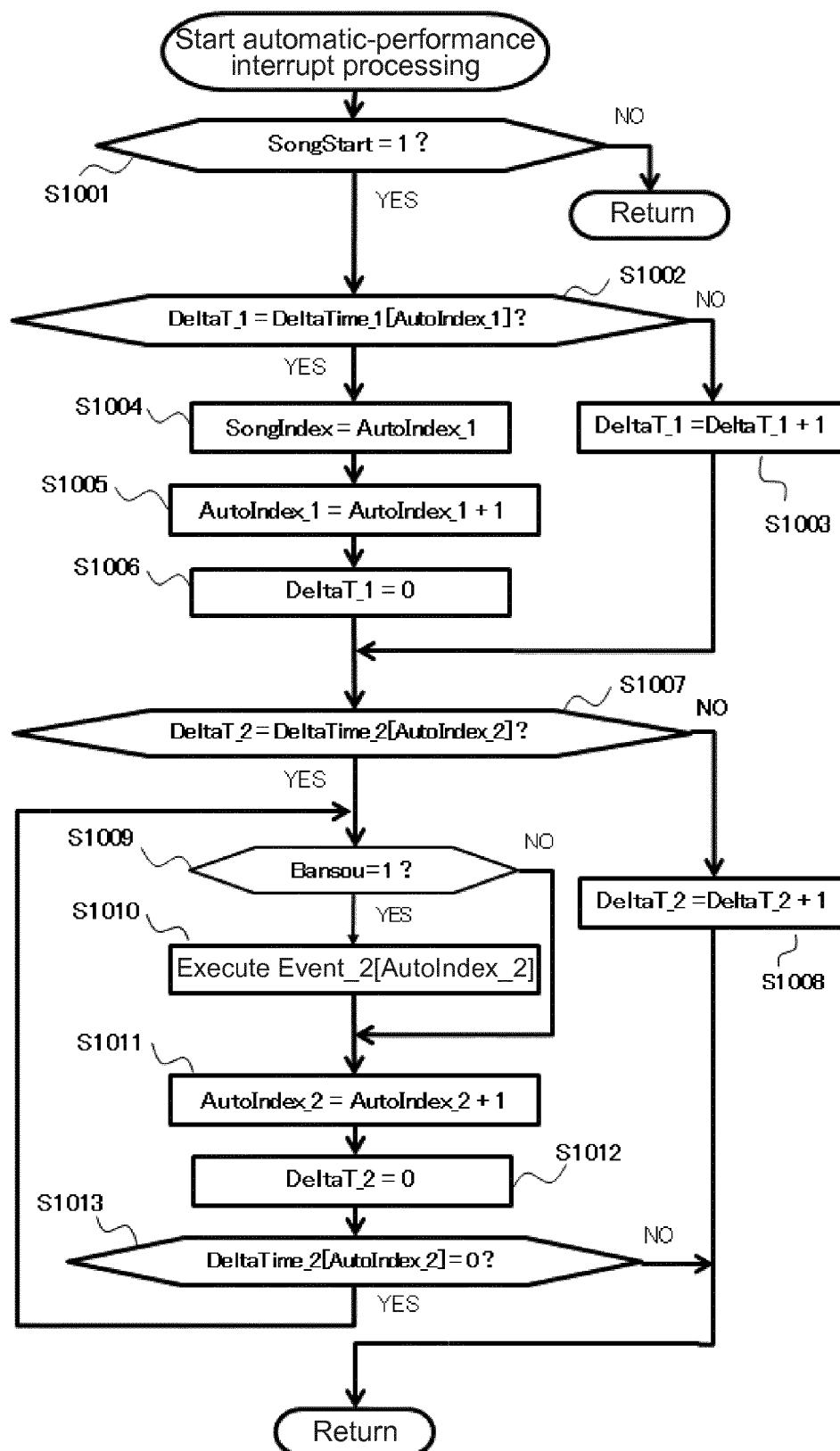


FIG. 10

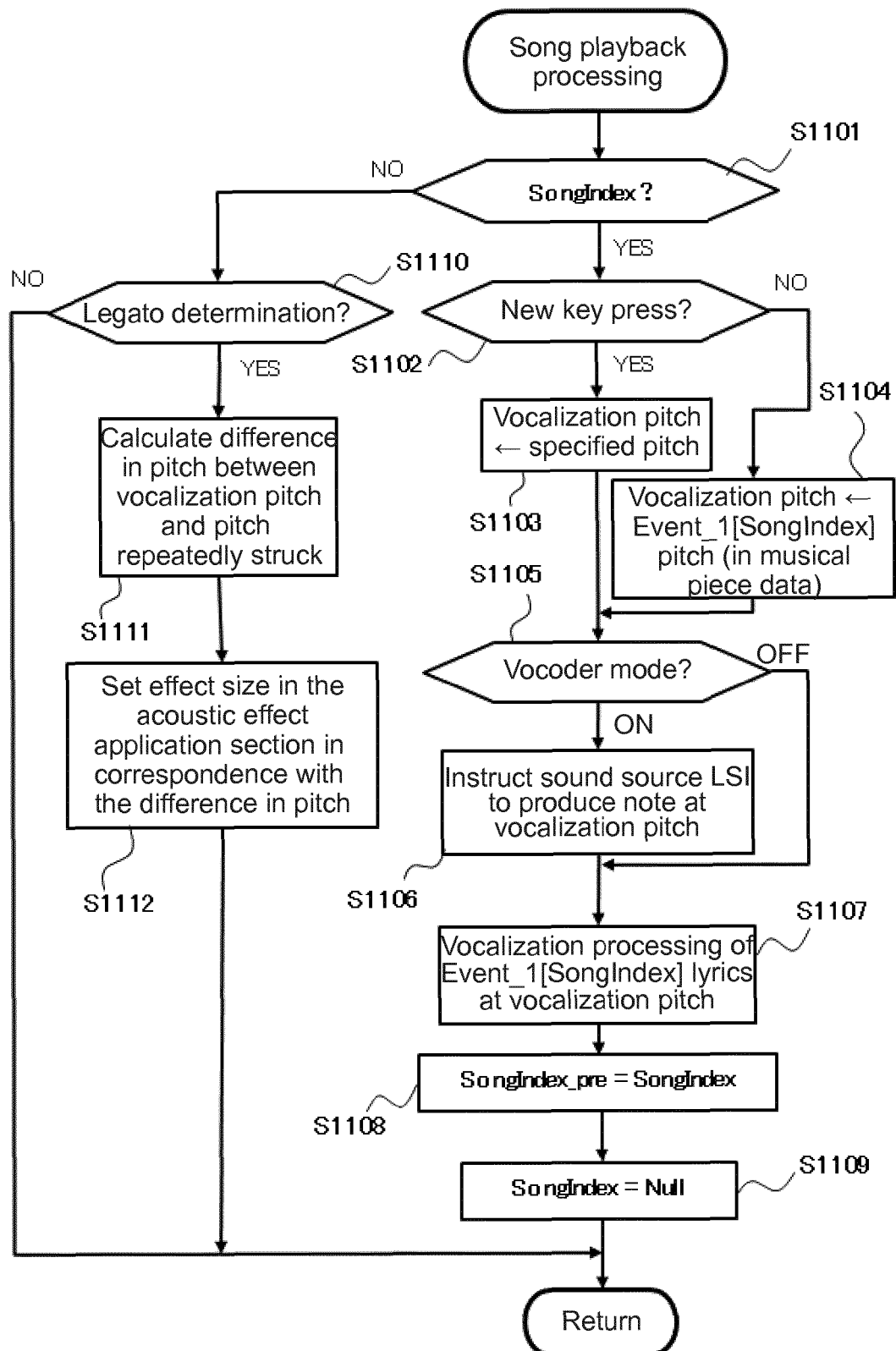


FIG. 11

## REFERENCES CITED IN THE DESCRIPTION

*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

- EP 2270773 A1 [0002]
- JP H09050287 B [0004]

### Non-patent literature cited in the description

- **MERLIJN BLAAUW et al.** A Neural Parametric Singing Synthesizer Modeling Timbre and Expression from Natural Songs. *applied sciences*, 18 December 2017, vol. 7 (12), 1313 [0003]
- **KEI HASHIMOTO ; SHINJI TAKAKI.** Statistical parametric speech synthesis based on deep learning. *Journal of the Acoustical Society of Japan*, 2017, vol. 73 (1), 55-62 [0040]
- **SHINJI SAKO ; KEIJIRO SAINO ; YOSHIHIKO NANKAKU ; KEIICHI TOKUDA ; TADASHI KITAMURA.** A trainable singing voice synthesis system capable of representing personal characteristics and singing styles. *Information Processing Society of Japan (IPSJ) Technical Report, Music and Computer (MUS) 2008 (12 (2008-MUS-074)*, 08 February 2008, 39-44 [0074]