(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification:
G10H 1/00 (2006.01)   G10H 1/043 (2006.01)

(21) International Application Number:
PCT/US2009/050275

(22) International Filing Date:
10 July 2009 (10.07.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/079,691   10 July 2008 (10.07.2008)   US

(71) Applicant (for all designated States except US): STRINGPORT LLC [US/US]; 970 Miller Avenue, Berkeley, CA 94708 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): MCMILLEN, Keith [US/US]; 970 Miller Avenue, Berkeley, CA 94708 (US). SHAVER, Chris [US/US]; 1254 10th Avenue, Apt 1, San Francisco, CA 94122 (US).

(74) Agents: VILLENEUVE, Joseph, M. et al.; Weaver Austin Villeneuve & Sampson LLP, P.O. Box 70250, Oakland, CA 94612-0250 (US).

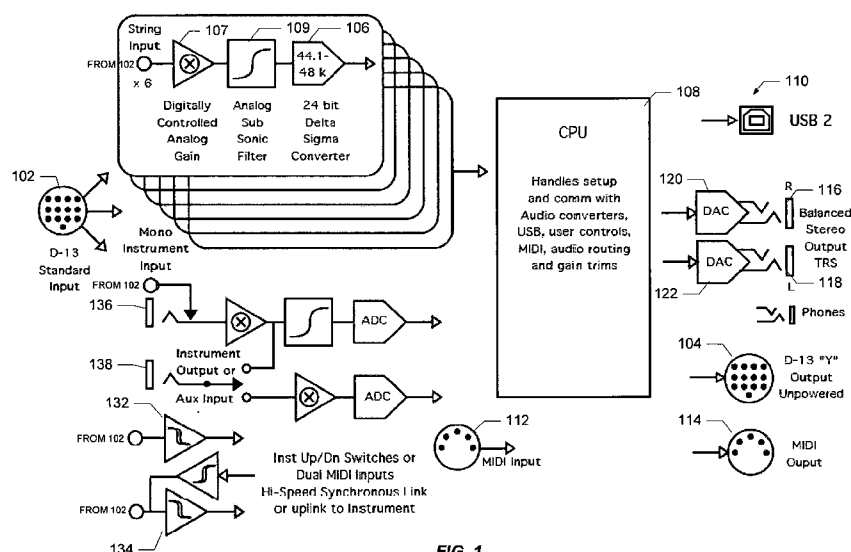(54) Title: COMPUTER INTERFACE FOR POLYPHONIC STRINGED INSTRUMENTS

FIG. 1

(57) Abstract: An interface device is described that allows the audio signals from a polyphonic stringed instrument to be introduced into a personal computer environment for feature extraction and signal processing.

COMPUTER INTERFACE FOR
POLYPHONIC STRINGED INSTRUMENTS

RELATED APPLICATION DATA

[0001]   The present application claims priority under 35 U.S.C. 119(e) to U.S. Provisional Patent Application No. 61/079,691 for COMPUTER INPUT DEVICE FOR POLYPHONIC STRINGED INSTRUMENTS filed on July 10, 2008 (Attorney Docket No. SPRTP001P), the entire disclosure of which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0002]   The present invention relates to interfaces between musical instruments and computing devices and, in particular, to interfaces for polyphonic stringed instruments.

[0003]   While the electronic keyboard has been married to synthesis control since its inception, stringed instruments have basically had no entry method for computer interface appropriate for this instrument family. The only available solutions have been bulky and expensive hardware devices that reduce the nuance of a stringed instrument to keyboard-like MIDI commands or rigid signal processing chains.

SUMMARY OF THE INVENTION

[0004]   According to a particular class of embodiments of the present invention, a computer interface for a polyphonic stringed instrument is provided. An analog interface is configured to receive a plurality of individual analog audio signals. Each analog audio signal corresponds to one of a plurality of strings of the stringed instrument. Analog-to-digital conversion (ADC) circuitry is configured to convert each of the analog audio signals to a corresponding digital audio signal. A processor is configured to combine the digital audio signals into a single serial data stream. A serial data interface is configured to transmit the serial data stream to a computer system.

[0005]   According to another class of embodiments, methods, apparatus, and computer program products are provided for processing audio signals for a stringed instrument. A serial data stream is received with a serial data interface of a computing device. The serial

data stream encodes a plurality of digital audio signals. Each digital audio signal represents one of a plurality of strings of the stringed instrument. The encoded digital audio signals are extracted from the serial data stream using the computing device. Each of the extracted digital audio signals is processed with the computing device, thereby generating a plurality of processed digital audio signals. Each of the processed digital audio signals corresponds to one of the plurality of strings of the stringed instrument.

[0006]   A further understanding of the nature and advantages of the present invention may be realized by reference to the remaining portions of the specification and the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007]   FIG. 1 is a simplified diagram of a specific embodiment of the invention.

[0008]   FIG. 2 shows the front and back panels of a specific embodiment of the invention.

[0009]   FIG. 3 is a simplified block diagram illustrating operation of a specific embodiment of the invention.

[0010]   FIGs. 4-7 are graphs illustrating various aspects of the operation of a specific embodiment of the invention.

[0011]   FIG. 8 is a table illustrating a message format for use with various embodiments of the invention.

[0012]   FIGs. 9-13 are representations of graphical user interfaces by which users may interact with various embodiments of the invention.

[0013]   FIG. 14 is an illustration of a computing platform that may be used in conjunction with various embodiments of the invention.

[0014]   FIG. 15 is an illustration of the parallel processing of string audio in accordance with specific embodiment of the invention.

[0015]   FIG. 16 is an illustration of a network of computing platforms that may be used in conjunction with specific embodiments of the invention.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

[0016]   Reference will now be made in detail to specific embodiments of the invention including the best modes contemplated by the inventors for carrying out the invention. Examples of these specific embodiments are illustrated in the accompanying drawings.

2

While the invention is described in conjunction with these specific embodiments, it will be understood that it is not intended to limit the invention to the described embodiments. On the contrary, it is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims. In the following description, specific details are set forth in order to provide a thorough understanding of the present invention. The present invention may be practiced without some or all of these specific details. In addition, well known features may not have been described in detail to avoid unnecessarily obscuring the invention.

[0017]   Various embodiments of the present invention relate to devices and related techniques that enable the conversion of polyphonic string audio to digital form for use by various types of applications including, for example, feature extraction and signal processing applications. Embodiments of the invention provide an interface that converts the polyphonic output of any stringed instrument for presentation to a general purpose computer where the converted data may be processed in more sophisticated and elaborate ways than previous integrated solutions in which only limited types of processing are enabled in bulky, stand-alone boxes. Specific implementations of an interface designed in accordance with a particular class of embodiments (referred to herein as the StringPort) are described below.

[0018]   According to a specific embodiment illustrated in FIGs. 1 and 2, the StringPort accepts polyphonic string audio through an industry standard Din 13 connector 102 on the front panel 103 of the StringPort. The depicted embodiment assumes a pickup system (not shown) on the stringed instrument that has one or more dedicated transducers for each string such as, for example, the Zeta Violin family or Roland/Yamaha guitar pickup systems. It will be understood that other suitable pickup systems may also be employed. In this example, Din 13 connector 102 transfers six strings of audio as well as a monophonic summed audio signal. It should be noted that embodiments are contemplated in which fewer or more string signals may be handled.

[0019]   A second Din 13 connector 104 is located on the rear panel 105 of the StringPort and acts as a "pass through" for some or all of the signals presented to input Din 13 connector 102. Back panel switches 128 and 130 on the StringPort allow the user to pass a volume control voltage (e.g., to affect a voltage controlled amplifier in an attached legacy device) and to switch options such as, for example, program selection, as signal data to Din

13 connector 104. Such pass through signals may, for example, be provided for use with legacy equipment.

[0020]   Internally, the StringPort conveys the seven audio signals (i.e., 6 polyphonic string signals and one "sum of" signal often from a different pickup system or microphone on the instrument) and one auxiliary signal to eight high quality Delta Sigma analog-to-digital converters (e.g., ADC 106). In this example, each audio signal passes through a digitally controlled gain stage 107, is filtered 109, and then converted to 24-bit data at either a 44.1KHz or 48 kHz sample rate. Subsonic analog filters 109 are intended to eliminate movement and/or displacement noise from string displacement caused, for example, by change of bow direction or body and bridge noise caused by vibrato style bridges). Such noise is typically relatively low frequency noise, and is undesirable in that it can move the analog input to the ADC out of the optimal range, as well as present artifacts after conversion which interfere with subsequent digital signal processing.

[0021]   A central processing unit (CPU) 108, or the equivalent (which may be implemented using any of a wide variety of devices including, for example, conventional processors and controllers as well as custom integrated circuits), extracts and formats the eight serial digital audio streams from the ADCs and conveys them efficiently to a universal serial bus (USB) transport PHY and connector 110. The output audio signals may then be transported within the connected computer(s) in any of a wide variety of formats, e.g., Audio Stream Input/Output (ASIO) or Core Audio signals, depending on the operating system of the computer (not shown) to which the StringPort connects. ASIO is a computer soundcard driver protocol for digital audio. Core Audio is a low-level API for dealing with sound in Apple's Mac OS X operating system. More generally, drivers may be provided for any of a variety of operating systems such as, for example, Mac OS X, Windows OS, and Linux.

[0022]   USB interface 110 may comprise any of the variety of serial bus interfaces associated with the USB family of standards including, for example, USB 1.0, 2.0, or 3.0, and may employ any of the communication protocols within that family of standards. More generally, embodiments of the present invention are contemplated which may employ a much wider array of serial bus interfaces such as, for example, Firewire. Therefore, references to USB technologies should not be considered to unduly limit the scope of the present invention.

[0023]   According to some embodiments, a variety of additional information regarding the string signals and the manner in which the instrument is being played may be generated from the StringPort output by a host application on one or more connected computers. According to some embodiments, this additional information is provided in such a format so as to make it accessible and useful to a wide variety of commercially available synthesis applications. A particular format is discussed below. Such information might include, for example, frequency- e.g., the continuous pitch of the fundamental harmonic of the string, amplitude – e.g., the continuous measurement of energy of the strings vibration, triggers – e.g., whether the string is active and when the string starts its activity based upon the user picking or bowing or otherwise energizing the string, centroid – e.g., a measure of the brightness or timbre of a string measured as the spectral balance point at the center of amplitude weighted partials expressed as a frequency, even/odd – e.g., the ratio between even and odd harmonics, noise – e.g., the level of energy that is not harmonic usually created by the bow and or picking style, spectrum – e.g., the continuous representation of the sound as a collection of partials or harmonic components , etc.

[0024]   According to a particular class of implementations, gestural information (e.g., fret scanning data, accelerometer outputs, surfaces, etc.) is provided to the host application via the StringPort via the Din 13 cable coming from the instrument. Polyphonic pickups often include "up" and "down" switches which typically convey relatively low frequency "on" or "off" states or select the next or last preset of variables controlling the synthesized or processed audio. According to a particular implementation, these switch signals are repurposed for transmission of digital signals (e.g., high speed data superimposed on the primary signals) that may be used for a variety of advanced signal processing purposes. For example, the superimposed data might represent gestural information from the instrument such as string length. Such information might be generated, for example, by fret scanning sensors which can identify the finger positions of the musician before string vibration even begins. Other information might include, but is not limited to fingerboard scanning data, accelerometer data, touch surface data, knob data, switch data, slider data, hall effect sensor data, optical sensor data, pressure sensor data, proximity detector data, gyroscope data, breath controller data, etc.

[0025]   According to a specific embodiment, this information is provided in MIDI format, but is not necessarily limited to MIDI data rates. An uplink path may also be

provided to control behavior of the instrument which would conventionally be under control of the user through switch or knob settings directly on the instrument. This allows a preset on the host application to control and change the instrument's sound or other advanced features such as string sustainers or mechanical/acoustic modifiers that directly affect the way the string vibrates.

[0026] In addition to the polyphonic audio stream, three digital data paths are conveyed to the computer. MIDI in and out connectors 112 and 114 allow users to add standard peripherals such as foot pedals and other controllers. Din 13 connector 102 may supply analog control signals such as a volume potentiometer and select switches from the instrument through the cable. These signals may then be converted and conveyed to the computer via Din 13 connector 104.

[0027] According to one embodiment, a beefed up power supply is provided to the instrument with a separate return path via the Din 13 connector in anticipation of power needs for embedded processing within the instrument. The supply is intended to be sufficient to handle any reasonable load in a guitar (e.g., up to a couple of watts), and the return path (via an unused wire in the cable) will help keep the audio clean. Finally, two separate serial data input streams are supplied. These are represented by comparators 132 and 134 which receive their inputs from Din 13 connector 102. According to a specific embodiment and as mentioned above, these datapaths may be used to transfer fingerboard tracking information (e.g., fret scanning information as was used in the Zeta Mirror Six guitar) allowing a more rapid and robust pitch extractor. Other uses include transferring gestural data such as pressure sensors, joy sticks, accelerometers which can be mounted in or on the instrument . All data paths are tagged then merged in the StringPort and appear to the OS of the connected computer as a single input of MIDI data or other data formats such as UDP.

[0028] The ¼" jacks (136 and 138) on the front panel allow the user to insert a summed mono signal of the instruments sound to replace the summed instrument sound that normally travels down the D13 cable. This allows the user to modify the sound of this signal before it enters the StringPort and then the Host PC. This same signal is available through a second ¼" jack so a user can process the analog signal in parallel with the digital signal path within the host PC. Under user selection from the StringPort host application on the Host PC, the user can reconfigure this 2ⁿᵈ ¼" jack (138) as an auxiliary input for additional analog inputs,

e.g., other instruments or microphones, to be encoded in the serial data output. This jack may also offer phantom power so a microphone can be directly used without any additional powered preamps making the entire performance system more compact and reliable.

[0029]   A pair of outputs 116 and 118 provide a high quality stereo representation of the string input signals via CPU 108 and digital-to-analog converters (DACs) 120 and 122. That is, the stereo signal may be a representation of processed versions of the plurality of digital audio signals received back from the computer system. Alternatively, the stereo signal may be synthesized audio rendered with reference to any information extracted from the plurality of digital audio signals or serial data stream by the computer system. According to a specific implementation, these outputs are differential ¼" jacks with switchable -10 and +4 dB levels (e.g., using switch 123). A 3.5mm headphone jack 124 and volume control 126 are present on the front panel.

[0030]   According to a particular implementation, the electronics of the StringPort are enclosed in an aluminum extruded chassis measuring roughly 4" x 1.5" x 5" (W x H x D). This implementation of the StringPort is designed such that up to four units can fit in a single 1U rack space (using a rack mount adapter accessory), anticipating compact portable stage-worthy support for string quartets. Since power requirements approach the limits of USB powered devices, a rear mounted power jack and universal power supply may also be provided to ensure reliable operations.

[0031]   According to a particular class of embodiments, the StringPort host application on the computer(s) to which the StringPort is connected includes real-time event detection and classification capabilities. Prior attempts at reliably detecting events such as, for example, the beginning of a note, have generated unsatisfactory results. The inadequacy of such previous techniques may be understood with reference to the example of a large string (e.g., a bass string) which is struck by the musician multiple times in succession. That is, when the musician strikes the string the first time, it begins at rest, and therefore the interaction may be detected fairly reliably. However, when the musician strikes the string a second time while the string is still vibrating, the energy of the string may not change sufficiently for conventional techniques to detect the event. Therefore, embodiments of the invention have been provided which address the limitations of previous techniques.

[0032]   According to one such embodiment illustrated in FIG. 3, the real-time detection and classification of events produced by string instrument performance may be represented

as a two stage system which receives a digital representation of each individual string's audio (302) from the StringPort, identifies event candidates (304), and then classifies the event candidates using various characteristics (306). Classification includes classifying some event candidates as to be ignored. According to a specific embodiment, the first stage

5   classifies peak segments separately for positive and negative parts of an audio signal into trajectories composed of peak segments similar in constituency. The second stage of the system classifies events with a set of neural networks, determining inclusiveness from a set of time-series, frequency, and statistical information about the signal.

[0033]   According to a particular class of implementations, event types or classifications

10   correspond to various types of performance techniques, e.g., picks, plucks, taps, etc. In addition, event data for each event identified may include information regarding various characteristics of the event such as, for example, its intensity. The identification and classification of events in real time can be advantageous, for example, in enabling a synthesizer or similar generative device to produce an output corresponding to the "attack"

15   of an instrument performance, i.e., how quickly a signal reaches full amplitude. This is particularly the case where, as with some embodiments of the invention, events are determinable prior to the availability of pitch information or other more conventionally derived envelope characteristics. Additional data from the system can be used to modify the processes initiated by the event to emulate the response of the stringed instrument to the

20   performance from which the event had been generated.

[0034]   According to the specific embodiment illustrated in FIG. 3, event extraction may be subdivided into three stages: preprocessing of the audio signal, extraction of peak segments as atomic events, and classification of the atomic events into trajectories while maintaining a set of active trajectories. The audio signal sent into the system is first divided

25   into two parallel segments, one corresponding to the positive part of the audio signal and one corresponding to the negative part (352). This may be understood with reference to the graph of FIG. 4 in which audio input signal 402 is shown in comparison with corresponding positive and negative signals 404 and 406. Although not depicted in FIG. 3, it should be noted that the topology of the system bifurcates upon this division into two parallel pathways

30   that each extract peak segments independently as discussed below. As will be described, a set of trajectories for each part of the signal is kept in parallel, and shared information between sets is used to determine the correspondence of positive and negative peaks.

[0035]   Each audio signal segment is squared (354) and passed through a smoothing filter (356) to mitigate roughness in the signal and to simplify the identification of the peak segments. According to a specific embodiment, the smoothing filter is a finite impulse response (FIR) filter morphologically similar to the shape of a typical peak segment and is constructed as a 64-sample Gaussian window having a precision of around 0.1. Typically, the smoothing filter has a low frequency response to remove sharp quick changes. As will be understood by those of skill in the art, depending on the relevant frequency range of the audio being processed, a suitable filter may be chosen from among a wide range of alternatives. For example, for higher frequency instrument strings, the peaks are usually sharper and more coherent making smoothing less important and allowing narrower, higher roll-off smoothing filters to be used.

[0036]   For each preprocessed segment of audio, peak segments are extracted and recorded as events (358) which are then classified in trajectories (360). According to one approach, the initial and final boundaries of a peak segment are determined by the processed signal rising above and falling below a threshold, respectively. This threshold may be a constant or, alternatively, be maintained adaptively with reference, for example, to the amplitude of the segment. An illustration of peak locations relative to a threshold 502 is shown in FIG. 5.

[0037]   Within a peak segment itself, metrics relating to the general characteristics of the peak are acquired. Such peak segment metrics may include, for example, the maxima, the width, the average amplitude, and higher moments of the peak segment. When the final boundary of a peak segment is reached, an event is generated which includes data about the peak. These data may include, for example, the amplitude of and position of the maxima within the segment, the metrics measured about the peak, and derivative metrics such as the variance, tilt, and kurtosis of the segment. A terminal heuristic check may also be performed on the generated events to throw out events that are exceptionally small, squat, or otherwise unfitting of further consideration and classification.

[0038]   According to a specific embodiment, given a stream of peak segment events, the trajectory classifier maintains a set of trajectories, each including correspondingly admissible events. The trajectories form relative bands of active event amplitudes that track independently the envelopes of the fundamental and overtones. This relationship may be understood with reference to FIG. 6 which illustrates trajectory extraction from signal 602.

According to one approach, the primary measure of admission into a trajectory is the absolute ratio between an event and a derivative of events within the trajectory. This derivative is typically weighted heavily against the most recent event; the extreme case being a simple comparison with the most recent event in the trajectory.

[0039]   The absolute ratio r is calculated from the event amplitude Ae and the effective trajectory amplitude At using the following relationship: $log\ r = |\ log\ Ae - log\ At\ |$. Tested in this manner against each active trajectory, if a trajectory is found to be close within a threshold for maximum absolute ratio the event is added to the trajectory. If the closest trajectory is beyond this threshold for inclusion, or no active trajectories exist, a new trajectory is created from the event. When multiple trajectories admit an event, further classification may be performed by threshold proximity, similarity with other events in the trajectory, and contextual similarity with the relationships between events with the trajectory. In addition, for trajectories that have enough events in them to make such calculations, the effective characteristic of the most recent event in a trajectory can be altered to reflect the projected amplitude and distance of the next event as opposed to the simple characteristics of the most recent event. This is useful because regularity becomes a more indicative means of classification for longer trajectories. A visual illustration of an example of peak comparison for inclusion is provided in FIG. 7.

[0040]   According to a specific embodiment, while including new events into trajectories, each trajectory is updated to reflect the time since the most recently included event. If the time since the inclusion of the most recent event exceeds a lifetime threshold the trajectory is closed out and removed from active trajectories. Among initiated trajectories, a new trajectory that exceeds the amplitude of all current trajectories is taken to potentially mark the initiation of a performance event. When this occurs a performance event is created and forwarded to the second stage for classification.

[0041]   The performance event data include the peak segment event from which they were derived along with data indicative of the context of the event relative to other trajectories. Performance events are discarded if they are in sufficiently close proximity to events generated from a nearby segment of opposite polarity. For example, the initial peak created by a performance event often is extracted from a segment of audio that includes a large peak of opposite polarity immediately following a zero crossing. Each trajectory classifier (i.e., for the positive and negative parts of the signal) picks up a corresponding

performance event, with the latter of the two being discarded because of its proximity to the former.

[0042]   According to one class of embodiments, the second stage of the system (e.g., Event Classification 306) receives performance events generated by the first stage and classifies them using a neural network. Included in the performance event data for each event is information pertaining to the characteristics of a peak segment and trajectory context out of which the event was generated. Along with these data, a window of the audio around the generated event is taken for classification analysis. From this window, a corresponding frequency response is computed so that a time and frequency series can be used. The general shape of each of these series along with other metrics are collected into an input vector which is passed to the neural network.

[0043]   From the neural network, a set of indicators are derived which classify the performance events. Certain performance events are classified as false triggers not indicative of an intentional performance technique. The remaining admissible events are classified by the performance technique that apparently generated them. On a guitar, for example, such techniques include picking the string, tapping the string against the fretboard, "hammering on" the string with the fretting hand, etc.

[0044]   According to a particular implementation, the topology of the neural network is a set of visible inputs and outputs repeated for sets of classifications that can be trained separately. The training of the network is performed against sets of performance events generated from the first stage and classified manually. The parameters of the neural network can be loaded dynamically to reflect training specific to particular stringed instruments and performance characteristics. In addition, parameters governing operation of the first stage (e.g., Performance Event Extraction 304) can be loaded to optimize performance for similar situations. As shown in FIG. 3, the system can output performance events prior to second stage processing and/or admit externally generated or stored events as direct inputs to the second stage so that each stage can be treated independently.

[0045]   According to some embodiments, data for trajectories maintained in the first stage may also accessed for use in synthesis. For example, useful envelope information can be obtained from the trajectories as they are updated which can serve to emulate sound characteristics of the performance on stringed instruments. Various outputs of the system

may also be used in the recording of a performance, yielding a precise and robust transcription of performance events and auxiliary envelope characteristics.

[0046]   According to a particular implementation class, the StringPort host application is written in Max/MSP (an authoring system for interactive computer music developed by Miller Puckette). The host application (which is resident on the computer to which the StringPort is connected) enables the extraction of a wide variety of features from the various StringPort outputs such as, for example, a continuous pitch of a fundamental harmonic, amplitude, centroid, brightness, even/odd harmonic balance, noise, spectral shape, complete spectrum, etc., as well as trigger and articulation events. This information is then provided in a format (described below) anticipating the High Definition Protocol for MIDI devices from the MIDI Manufacturers Association (MMA), i.e., the publisher and source of MIDI specifications. Commercial synthesis and notation packages (e.g., Synful Orchestra from Eric Lindemann) may also be modified for control by stringed instruments using the StringPort and the StringPort host application

[0047]   A specific implementation of an interface message format generated by the StringPort host application, referred to herein as an Acoustic Instrument Message (AIM), will now be described. As mentioned above, this format anticipates the new HD MIDI format, and addresses the longstanding problems of guitar synthesis and controllers that are continuous and acoustic in nature (e.g., stringed instruments, brass, woodwinds, etc).

[0048]   AIM forms the basis of a messaging system that adequately and succinctly transmits descriptors that represent an acoustic instrument's output in such a way so as to readily control a synthesizer. Other applications, e.g., notation, effects control (e.g., visual, robotic, etc.), pedagogy, etc., may also take advantage of these formatted data as well. The formatted data are sent at the frame rate of the analysis (i.e., the process that finds pitch, amplitude, etc.) using a 16 byte (128 bit) message. A specific implementation of such a message is provided in FIG. 8. According to a specific embodiment, an AIM message is sent per instrument string after each analysis window (which is related to FFT frame rate). That is, FFTs run at a specific frame rate based on the number of samples they use for each FFT frame. A frame of 512 sample points at 44.1KHz corresponds to about 1.16 ms of audio. While this could be accomplished using MIDI 2 single packet messages (SPMs), the overhead may be unacceptable for some applications.

12

[0049]   According to a specific implementation of a host application that receives and processes the StringPort output(s), individual processes are instantiated for each string that enable a wide variety of feature extraction and signal processing capabilities. Various aspects of such functionality are illustrated in the interfaces shown in FIGs. 9-13. The setup screen shown in FIG. 9 allows the user to assign string signals to the different string interfaces 1-6. The gain control for each string controls the digitally controlled analog gain stage in each string signal path (e.g., gain stage 107 of FIG. 1). This accommodates differing input levels, and may be done for each string individually and for the instrument as a whole (i.e., all strings) to ensure, for example, that the analog input from the instrument is optimized for the ADC input range.

[0050]   According to a specific embodiment, an automatic gain adjustment is provided in which the user strums all of the instrument strings, and the signal level for each string is automatically measured and adjusted to some default level, e.g., -6 dB. The adjustment may then be validated with another strum of the strings. Different sets of gain presets may be stored for different instruments that might have different loudness levels. As shown, each string interface has an associated tuning meter which allows the user to tune each string separately, or even to determine with a single strum whether any of the individual strings are out of tune, e.g., if all of the meters register green the instrument is in tune.

[0051]   DSP based processes can separate multiple audio sources into their individual sources. Some polyphonic pickups have poor isolation between detecting a single string and often hear adjacent strings. This crosstalk may have multiple magnetic, electrical, or mechanical causes depending on the pickup method and mounting scheme. Therefore, according to some embodiments, the host application or other software running on the host PC can help separate mixed string signals before sending each cleaned up signal onto its chain of processing and analysis.

[0052]   In addition to accurately determining the pitch of a note, one of the most difficult processing challenges relates to determining when the musician began the note; particularly when the new note is begun on a string that is already in motion. As discussed above, embodiments of the invention include an event capture and classification functionality that employs a time-domain analysis that marks each inflection point (e.g., local maxima and minima) in a string signal waveform, stores these data in an array, and searches through the

data with a trained neural net that can accurately determine when an event begins and the event type.

[0053]   According to a particular implementation, the neural network is trained to determine whether a particular event corresponds to a right hand pick or a left hand trigger (for a right-handed guitarist). This information is extremely useful in that it can be used for processing the string signals in any of a wide variety of ways. For example, this information could be used to distinguish a legato phrasing from a staccato phrasing, and therefore to inform a synthesizer how to articulate the corresponding note(s).

[0054]   The PolyFuzz application interface shown in FIG. 10 provides a sophisticated array of controls for applying distortion effects to each string of the instrument individually, or collectively (i.e., by selection of the "all" button). Such effects include, for example, compression, dynamics processing, parametric equalization, pitch-shifting, resonant filtering, frequency modulation, amplitude modulation, delay, reverberation, distortion, wave shaping, driving wave tables, stimulating resonances, gating, or limiting, amplifier simulation, equalization, etc.

[0055]   The SMACK application (the interface for which is shown in FIG. 11) is a phase-driven synthesizer and waveform modifier that enables the musician to store a table of sounds, and select from among the sounds in the table based on the phase of the corresponding string.

[0056]   According to a specific embodiment, and as illustrated in the interface of FIG. 12, a VST Wall interface allows the musician to map any of the thousands of existing virtual studio technology modules (VSTs are common industry standard audio processing units) to each string individually, or collectively (up to four VSTs on each string in the example shown).

[0057]   A set of six Phase Vocoders (see the interface of FIG. 13) allows different audio files to be controlled by characteristics extracted from each string. For example, mapping loudness onto location or pitch onto speed is easily accomplished.

[0058]   While the invention has been particularly shown and described with reference to specific embodiments thereof, it will be understood by those skilled in the art that changes in the form and details of the disclosed embodiments may be made without departing from the spirit or scope of the invention. For example, particular implementations have been described herein which employ CPU-based data and signal processing techniques. The

operation of particular implementations of the code which governs the operation of such CPUs may be understood with reference to the discussion above. Such code may be stored in physical memory or any suitable storage medium associated with the CPUs, as software or firmware, as understood by those of skill in the art. However, it should be noted that the use of a CPU or similar device is not necessary to implement all aspects of the invention. That is, at least some of the functionality described herein may be implemented using alternative technologies without departing from the scope of the invention. For example, embodiments are contemplated which implement such functionalities using programmable or application specific logic devices, e.g., PLDs, FPGAs, ASICs, etc. Alternatively, analog circuits and components may be employed. These and other variations, as well as various combinations thereof, are within the knowledge of those of skill in the art, and are therefore within the scope of the present invention.

[0059]   In another example, a host application is described above as being implemented using a particular programming language and using a particular messaging format. However, those of skill in the art will understand that the described functionality may be implemented using any of a wide variety of software and programming tools as well as any of a wide variety of messaging formats. In addition to the diversity of tools and formats that may be employed, such host application functionality may be implemented on a wide variety of computing platforms, an example of which is provided in FIG. 14.

[0060]   Computing system 1400 is an example of a system suitable for implementing particular embodiments of the present invention, and includes a processor 1401, a memory 1403, and an interface 1405. It should be noted that a variety of components such as caches, buses, controllers, persistent storage, and human interface devices may also be included in system 1400. In particular embodiments, memory 1403 holds instructions for processor 1401 to perform tasks such as, for example, those discussed above with reference to FIGs. 9-13. Various specially configured devices can also be used in place of, or in addition to processor 1401. In some examples, specially configured devices or hardware accelerators may supplement or replace processor tasks. The interface 1405 is typically configured to send and receive data over a network. Particular examples of interfaces include serial, network, frame relay, wireless, satellite, cable, and token ring interfaces.

[0061]   According to particular example embodiments, the system 1400 uses memory 1403 to store data, algorithms, and program instructions configured to enable various of the

functionalities related to the present invention. Such data, algorithms, and program instructions can be obtained from computer-readable media including computer-readable storage, examples of which include magnetic and optical media as well as solid state memory and flash memory devices.

5   [0062]   FIG. 15 is an illustration of the parallel processing of string audio in accordance with specific embodiment of the invention. That is, the figure illustrates how, in accordance with some embodiments, a string's audio may be processed in parallel to simultaneously generate any of the variety of feature extraction data (e.g., pitch, amplitude, etc.), as well as to apply any of the wide variety of audio processing (e.g., equalization, filtering, etc.).

10   [0063]   FIG. 16 is an illustration of a network of computing platforms that may be used in conjunction with specific embodiments of the invention. The path of the polyphonic string audio from string to a computer 1602 is shown. Computer 1602, in turn, is shown connected via an Ethernet infrastructure 1603 to computers 1604, 1606, and 1606 to illustrate the notion that embodiments of the invention support multiprocessor and/or multi-

15   core computing. That is the audio and the extracted feature data may be sent to one or more additional computers on a network where additional CPU power can be used to render synthesized audio as well as additional signal processing. Such information may be moved among applications on various machines using, for example, UDP over Ethernet. This allows for an expansion of processing power not possible in fixed hardware configurations.

20   It will be understood that a wide variety of network configurations and communication protocols may be employed to achieve this expansion without departing from the scope of the invention.

       [0064]   In addition, although various advantages, aspects, and objects of the present invention have been discussed herein with reference to various embodiments, it will be

25   understood that the scope of the invention should not be limited by reference to such advantages, aspects, and objects. Rather, the scope of the invention should be determined with reference to the appended claims.

What is claimed is:

1.   A computer interface for a polyphonic stringed instrument, comprising:

an analog interface configured to receive a plurality of individual analog audio signals, each analog audio signal corresponding to one of a plurality of strings of the stringed instrument;

analog-to-digital conversion (ADC) circuitry configured to convert each of the analog audio signals to a corresponding digital audio signal;

a processor configured to combine the digital audio signals into a single serial data stream; and

a serial data interface configured to transmit the serial data stream to a computer system.

2.   The interface of claim 1 further comprising one or more digitally-controlled gain stages configured to receive the analog audio signals and adjust one or more gains for the analog audio signals relative to one or more ranges associated with the ADC circuitry.

3.   The interface of claim 1 further comprising one or more subsonic analog filters configured to receive the analog audio signal.

4.   The interface of claim 1 wherein the analog interface is configured to receive one or more of additional analog signals having digital information superimposed thereon prior to reception of the additional analog signals by the analog interface, the processor being further configured to extract the digital information from one or more additional digital signals corresponding to the additional analog signals and encode the digital information in the serial data stream.

5.   The interface of claim 4 wherein the additional analog signals are bi-directional, and wherein the processor is further configured to generate uplink data for transmission to the stringed instrument via the additional analog signals.

6.   The interface of claim 4 wherein the digital information represents one or more of fret scanning data, fingerboard scanning data, accelerometer data, touch surface data, knob data, switch data, slider data, hall effect sensor data, optical sensor data, pressure sensor data, proximity detector data, gyroscope data, or breath controller data.

7.   The interface of claim 1 further comprising one or more digital-to-analog converters (DACs) and a stereo output interface, the processor being further configured in conjunction with the DACs to provide a stereo output signal via the stereo output interface, the stereo output signal comprising a stereo representation of processed versions of the plurality of digital audio signals received from the computer system.

8.   The interface of claim 1 further comprising one or more digital-to-analog converters (DACs) and a stereo output interface, the processor being further configured in conjunction with the DACs to provide a stereo output signal via the stereo output interface, the stereo output signal comprising synthesized audio rendered with reference to information extracted from the serial data stream by the computer system.

9.   The interface of claim 1 wherein the analog interface is further configured to receive a mono audio signal corresponding to a combination of the analog audio signals from the plurality of strings of the stringed instrument, the processor further being configured to encode a digital version of the mono audio signal in the serial data stream.

10.   The interface of claim 9 further comprising an auxiliary input configured to receive an auxiliary analog signal, the processor further being configured to encode a digital version of the auxiliary analog signal in the serial data stream.

11.   The interface of claim 1 wherein the serial data interface comprises a universal serial bus (USB interface).

12.   A computer-implemented method for processing audio signals for a stringed instrument, comprising:

receiving a serial data stream with a serial data interface of a computing device, the serial data stream encoding a plurality of digital audio signals, each digital audio signal representing one of a plurality of strings of the stringed instrument;

extracting the encoded digital audio signals from the serial data stream using the computing device; and

processing each of the extracted digital audio signals with the computing device, thereby generating a plurality of processed digital audio signals, each of the processed digital audio signals corresponding to one of the plurality of strings of the stringed instrument.

13.   The method of claim 12 wherein processing the extracted digital audio signals comprises one or more of dynamics processing, equalization, pitch-shifting, filtering, frequency modulation, amplitude modulation, delay, reverberation, distortion, wave shaping, driving wave tables, stimulating resonances, gating, or limiting.

14.   The method of claim 12 wherein processing each of the extracted digital audio signals comprises processing each of the extracted digital audio signals using a plurality of processing modules that simultaneously process each extracted digital audio signal in both a time domain and a frequency domain.

15.   The method of claim 12 further comprising recording each of the digital audio signals for subsequent processing.

16.   The method of claim 12 further comprising generating graphical representations corresponding to each of the digital audio signals, the graphical representations representing one or more of pitch, dynamics, or timbre for the corresponding string of the string instrument.

17.   The method of claim 12 further comprising controlling a special effect using information extracted from the serial digital stream.

18.   The method of claim 12 further comprising generating performance event data representing performance events for each of the plurality of strings with reference to the corresponding extracted digital audio signals, the performance events relating to specific types of interaction with the strings by a musician, wherein processing of the extracted digital audio signals is done with reference to the performance event data.

19.   The method of claim 18 wherein generating the performance event data comprises detecting event candidates with reference to peaks associated with the extracted digital audio signals, and classifying the event candidates into one or more of a plurality of event classifications.

20.   The method of claim 19 further comprising identifying a beginning of a note with reference to classification of the event candidates.

21.   The method of claim 19 wherein classifying the event candidates into the one or more event classifications is done using a neural network.

22.   The method of claim 12 further comprising extracting a plurality of audio signal characteristics from the extracted digital audio signals, wherein processing of the extracted digital audio signals is done with reference to the audio signal characteristics.

23.   The method of claim 22 wherein the audio signal characteristics include any of a continuous pitch of a fundamental harmonic of a string, an amplitude, a centroid, brightness, even/odd harmonic balance, noise, spectral shape, or complete spectrum.

24.   The method of claim 12 further comprising generating a plurality of acoustic instrument messages with reference to the extracted digital audio signals, each acoustic instrument message summarizing spectral information corresponding to a particular one of the plurality of strings of the stringed instrument.

25.   A computer program product for processing audio signals for a stringed instrument, the computer program product comprising at least one computer-readable storage medium having computer program instructions stored therein configured to enable at least one computing device to:

receive a serial data stream with a serial data interface of a computing device, the serial data stream encoding a plurality of digital audio signals, each digital audio signal representing one of a plurality of strings of the stringed instrument;

extract the encoded digital audio signals from the serial data stream using the computing device; and

process each of the extracted digital audio signals with the computing device, thereby generating a plurality of processed digital audio signals, each of the processed digital audio signals corresponding to one of the plurality of strings of the stringed instrument.

FIG. 1

FIG. 2

FIG. 3

FIG. 4

FIG. 5

502

FIG. 6

New Trajectory
initiated

Comparison for
inclusion into
trajectory
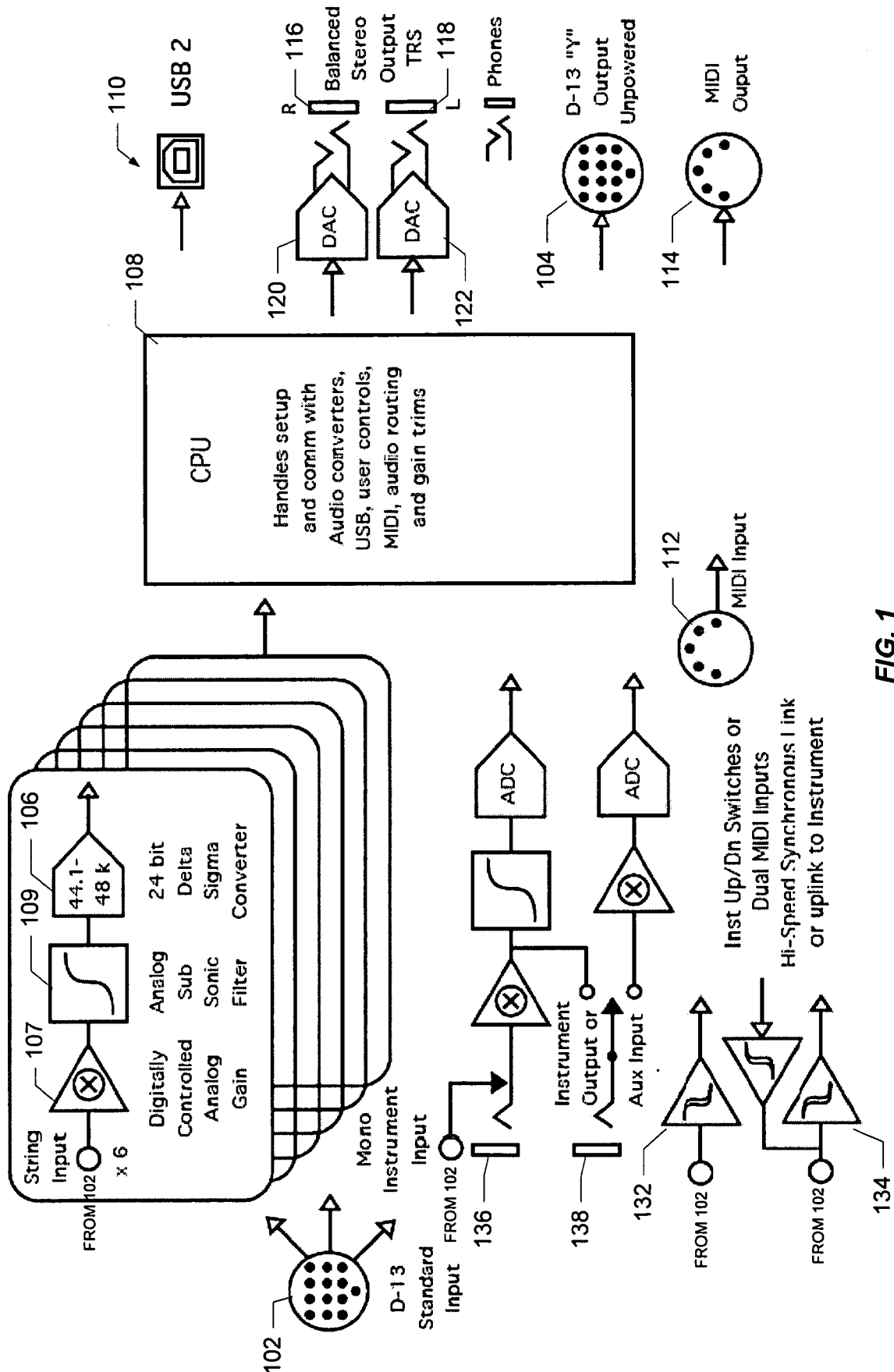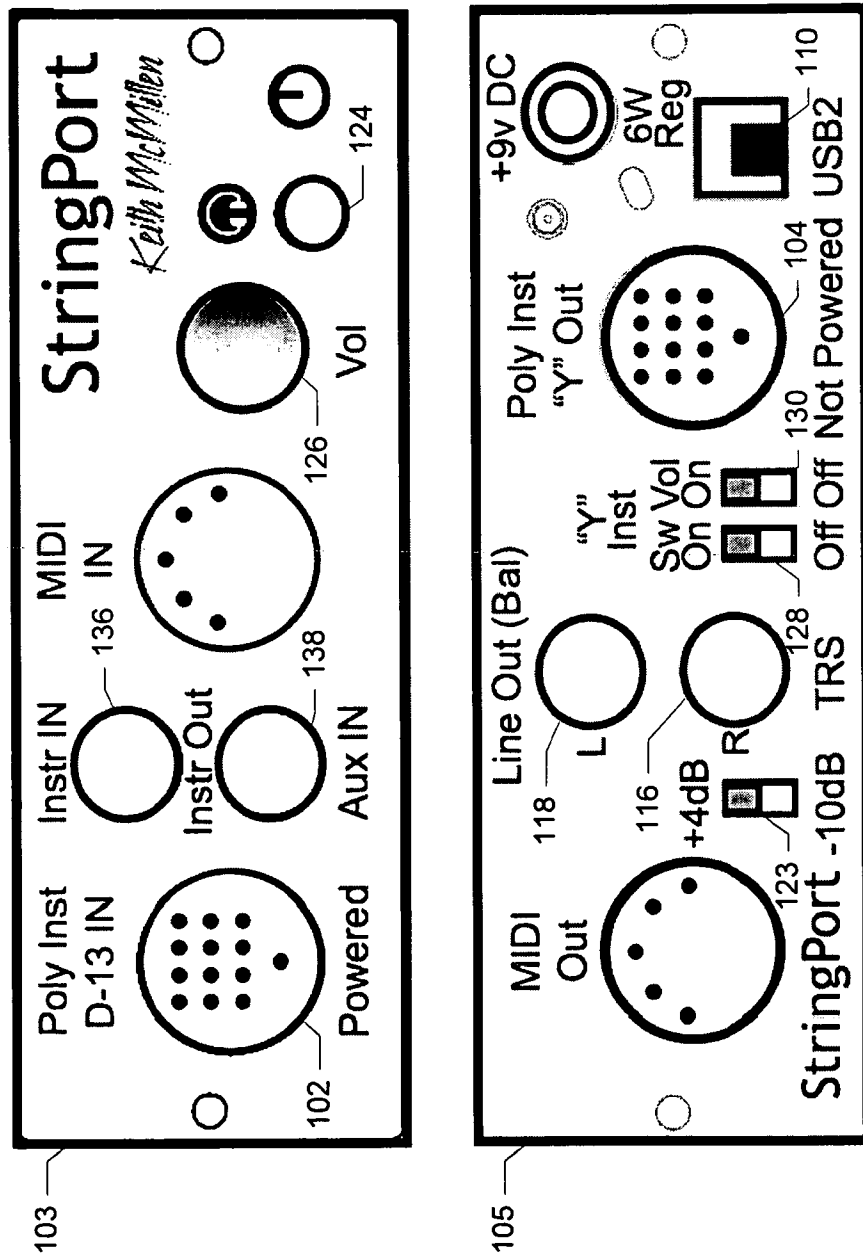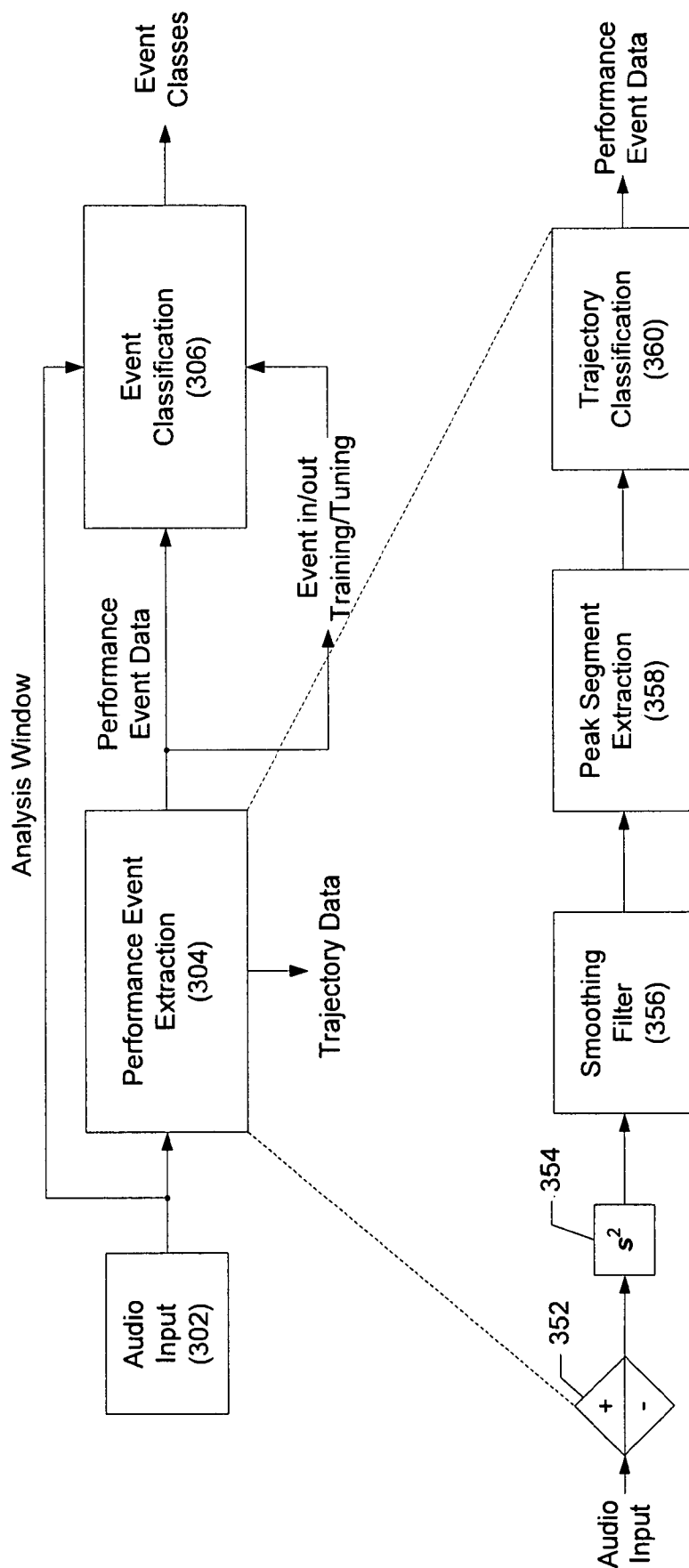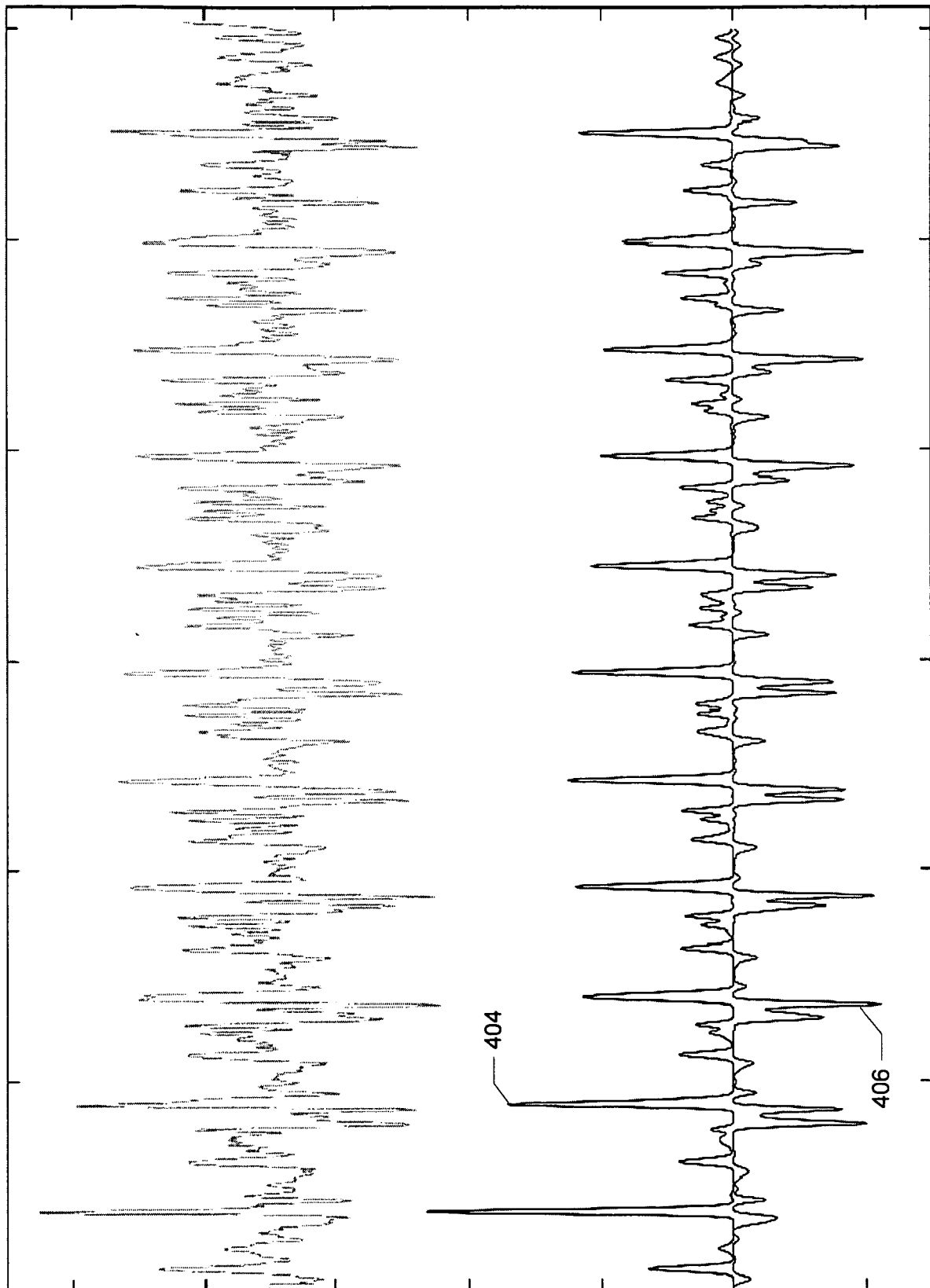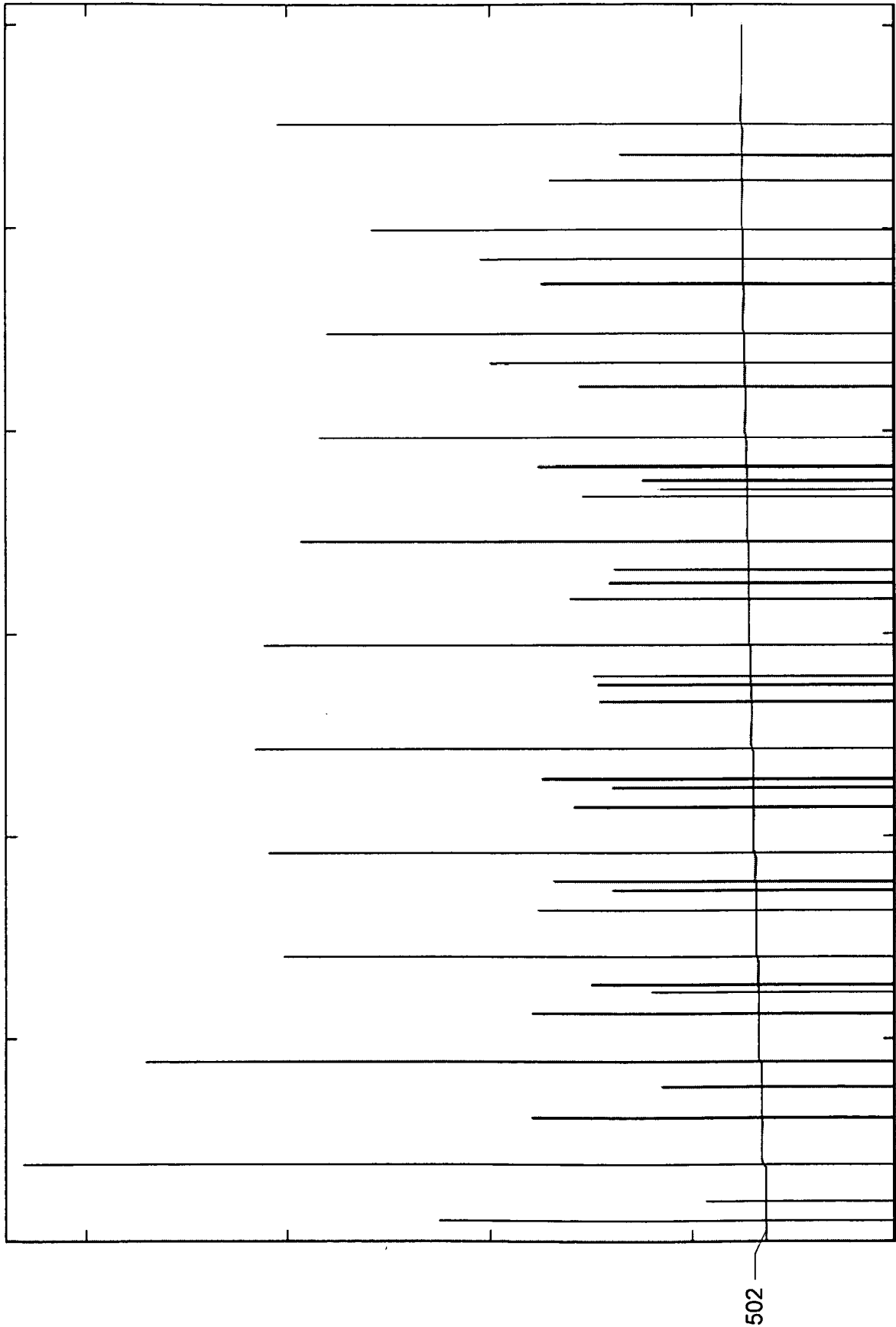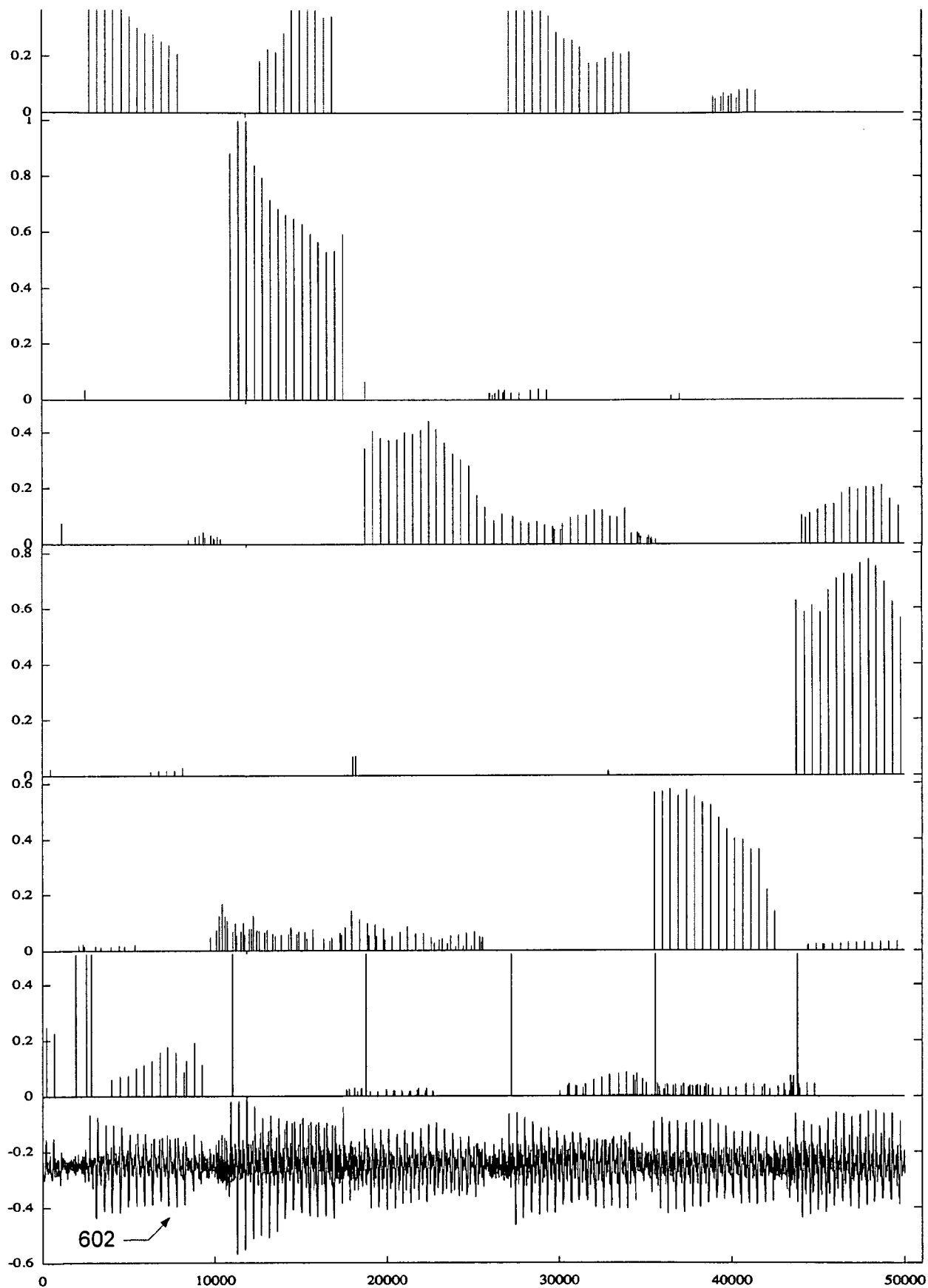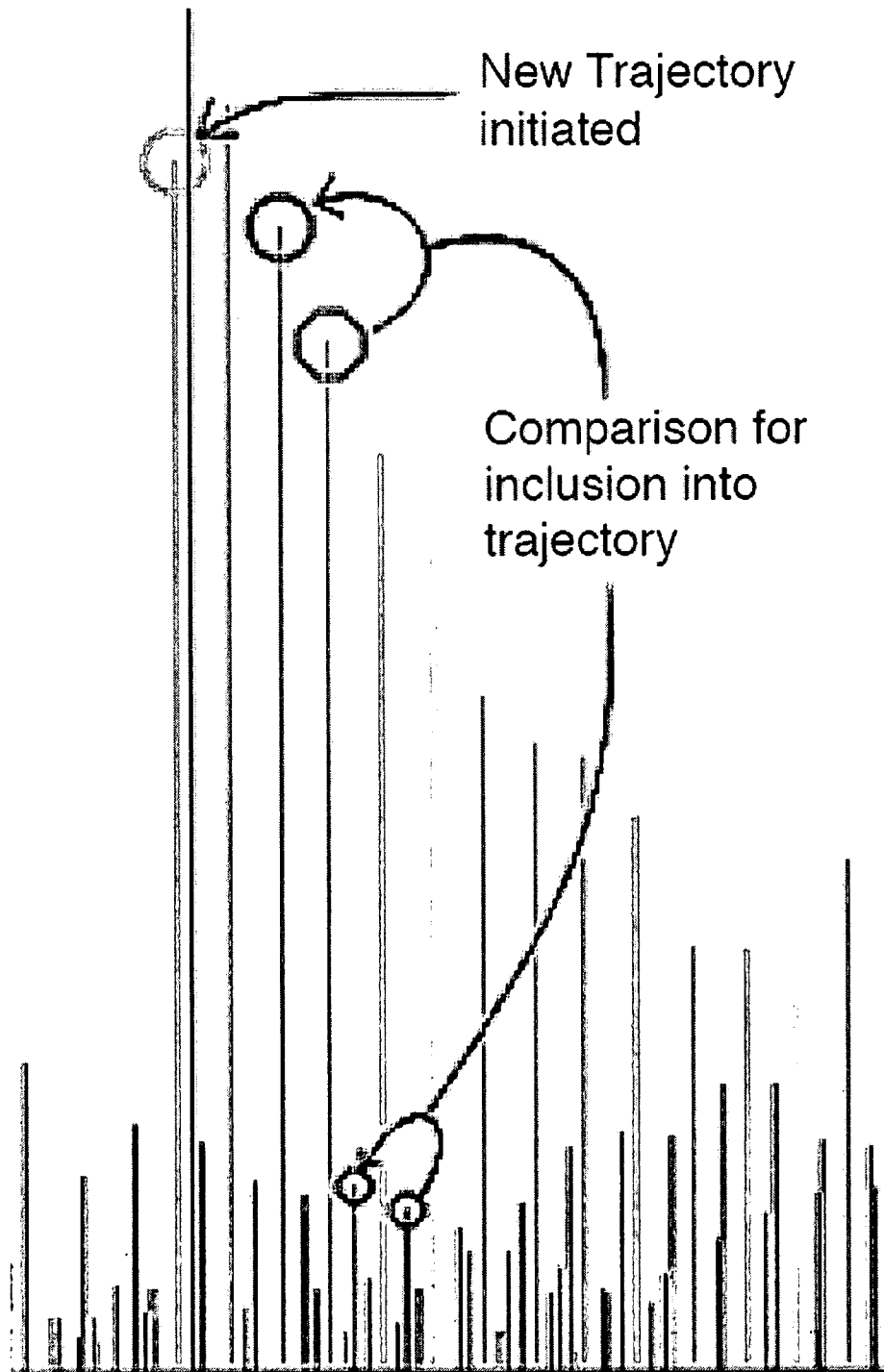
FIG. 7

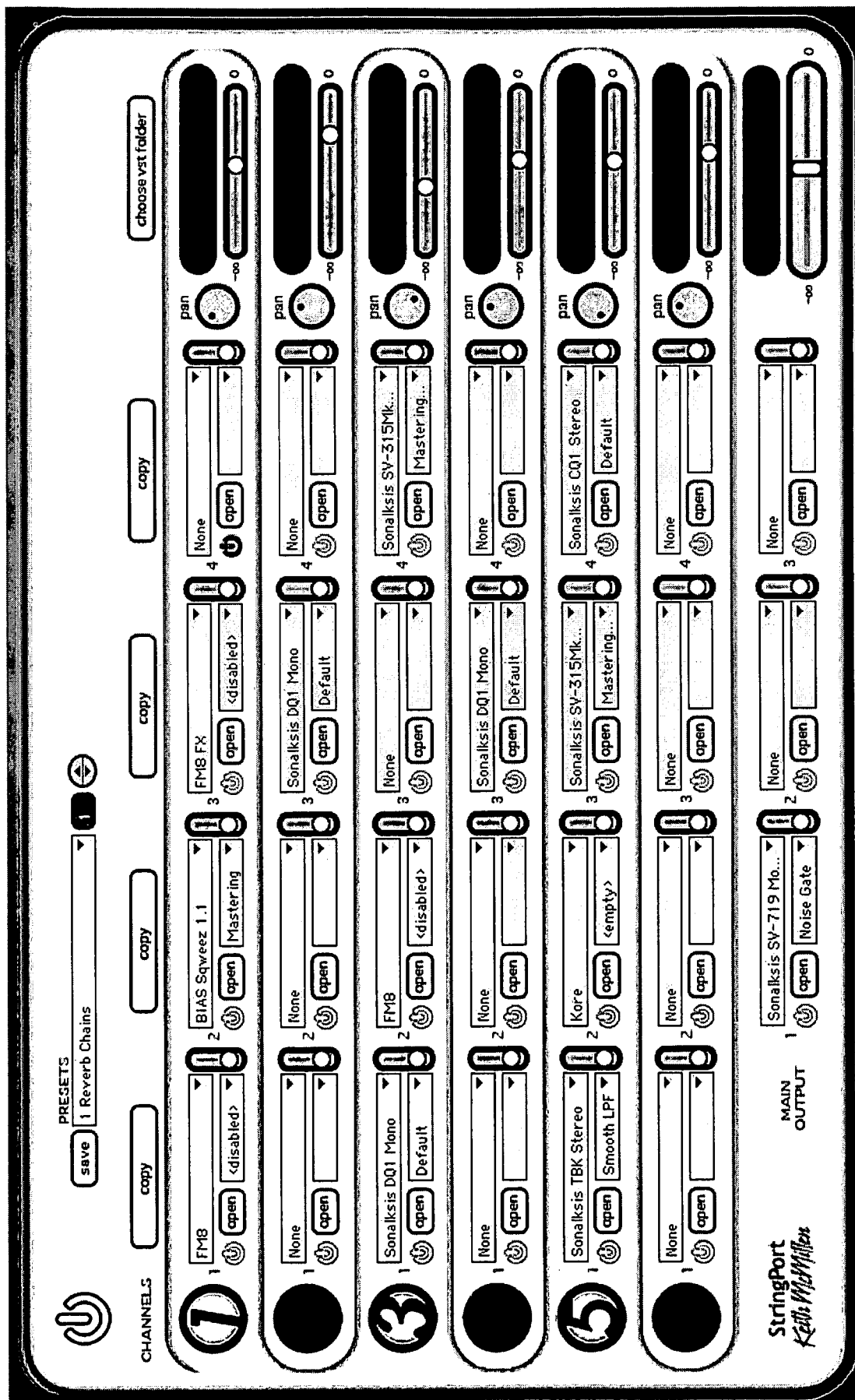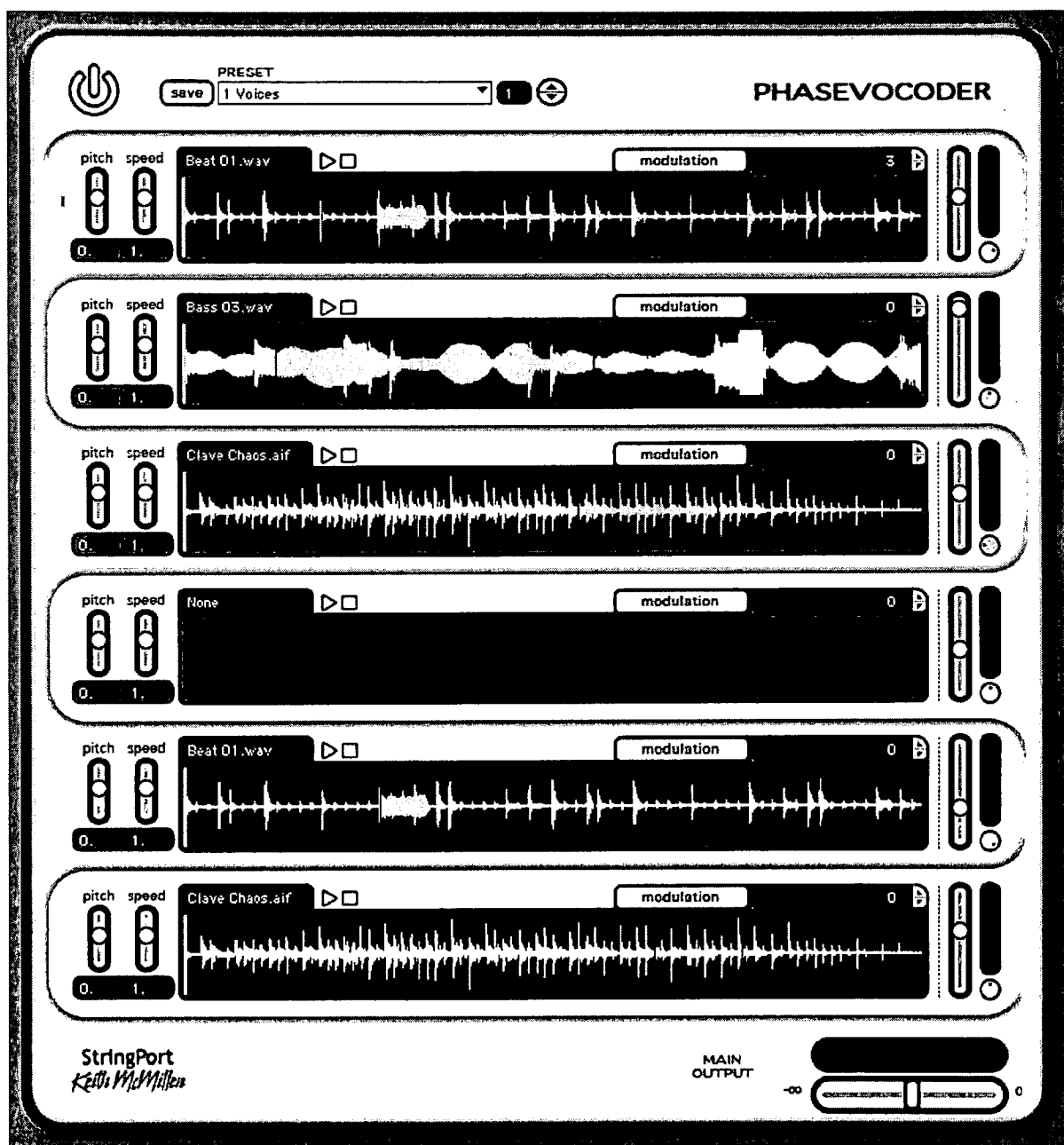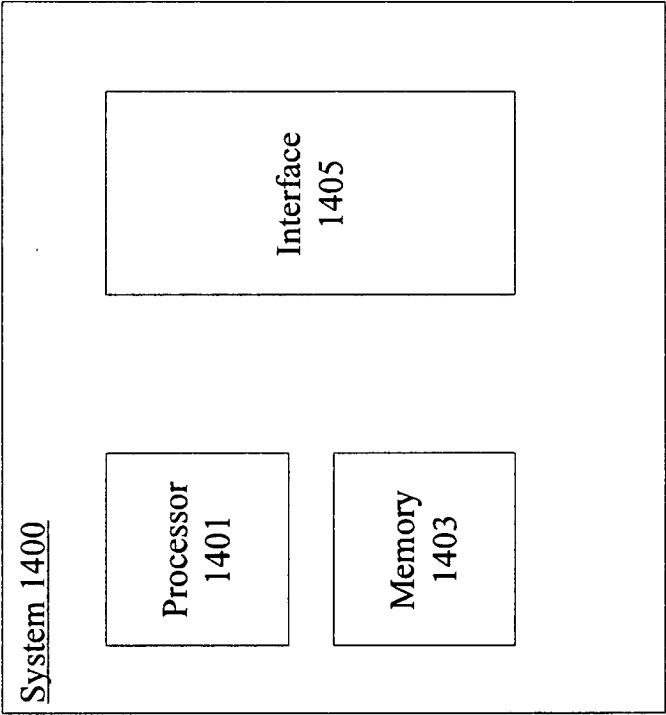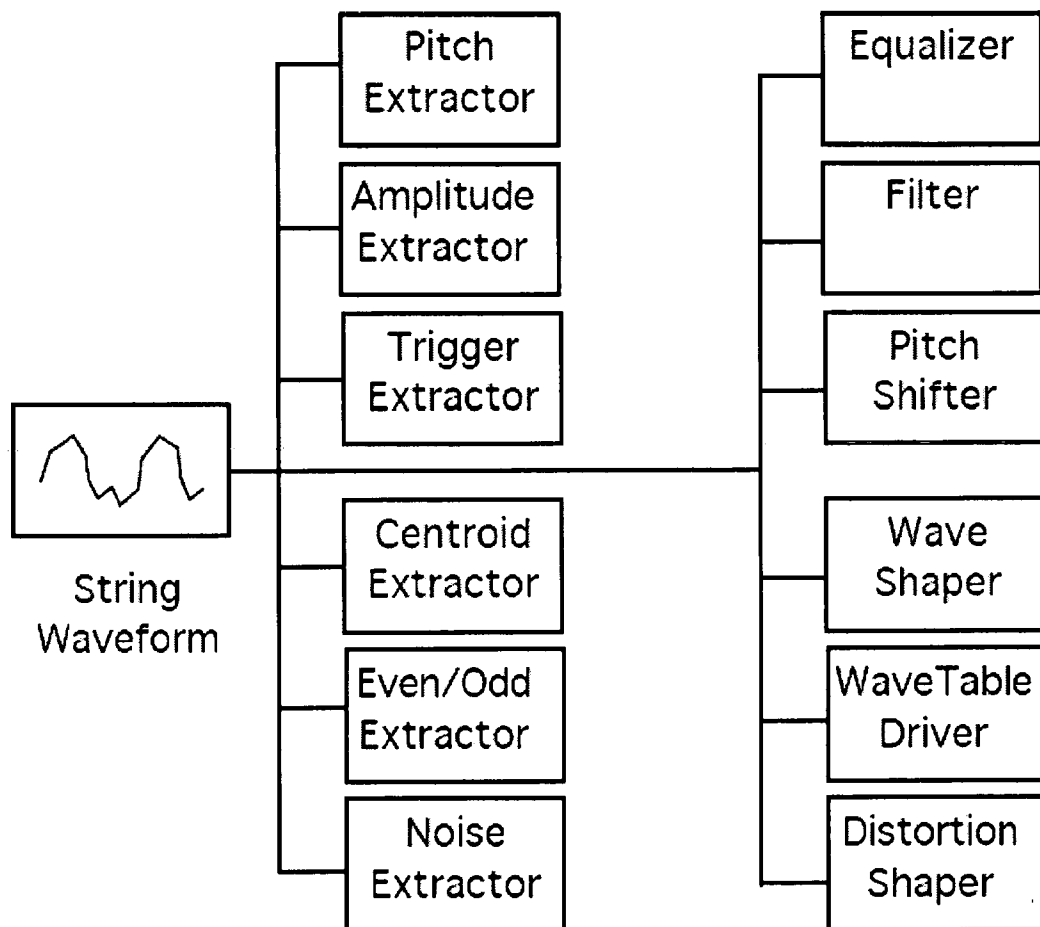| Param	Bytes	Units	Behaviour
On/Off Articulation	1		
Continuous Pitch	2	Semitone.Fractional (256.256 (60.00 = mid C))	
Triggered Pitch (note #)	1	0-127 MIDI 1 format	
Triggered Amp (velocity)	1	0-127 MIDI 1 format	0 does not turn off
Deviation from Trig'd Pitch (bend)	2	signed semitone.fractional (+/- 127.256)	
Continuous Amp	1	dB in .5dB increments	multiplicative
Centroid	2	Freq - Semitone.Fractional	multiplicative
Even/Odd	1	1=odd 127=50% 255 = even	multiplicative
Noise Amt	1	dB in .5dB increments	multiplicative
Noise Centroid	2	Freq - Semitone.Fractional	multiplicative
Inharmonicity	1	1=126%squeezed 127=none 255 = 126% stretched	multiplicative
reserved	1	reserved	
Packet size	16	is 128 bits	

FIG. 8
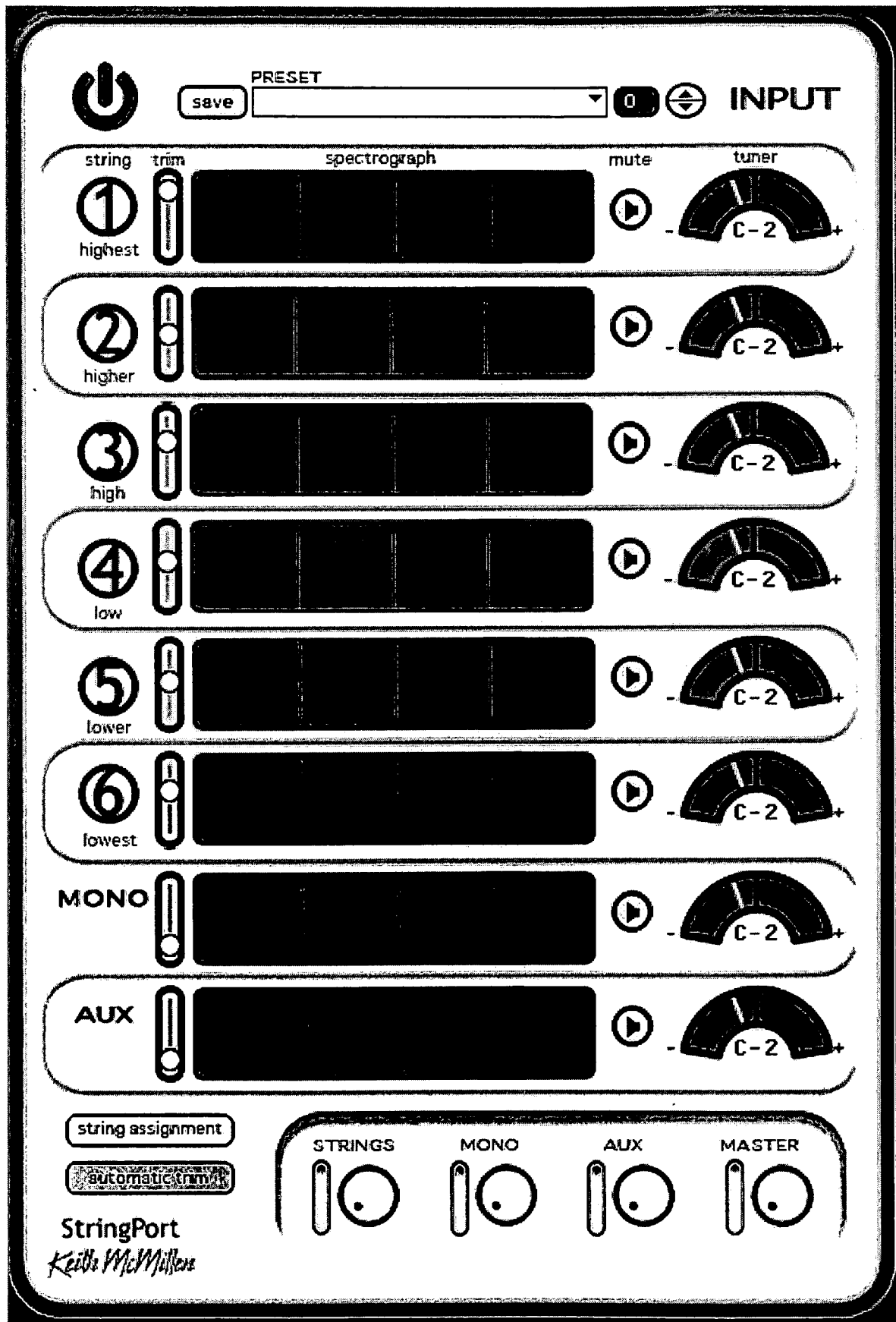
FIG. 9
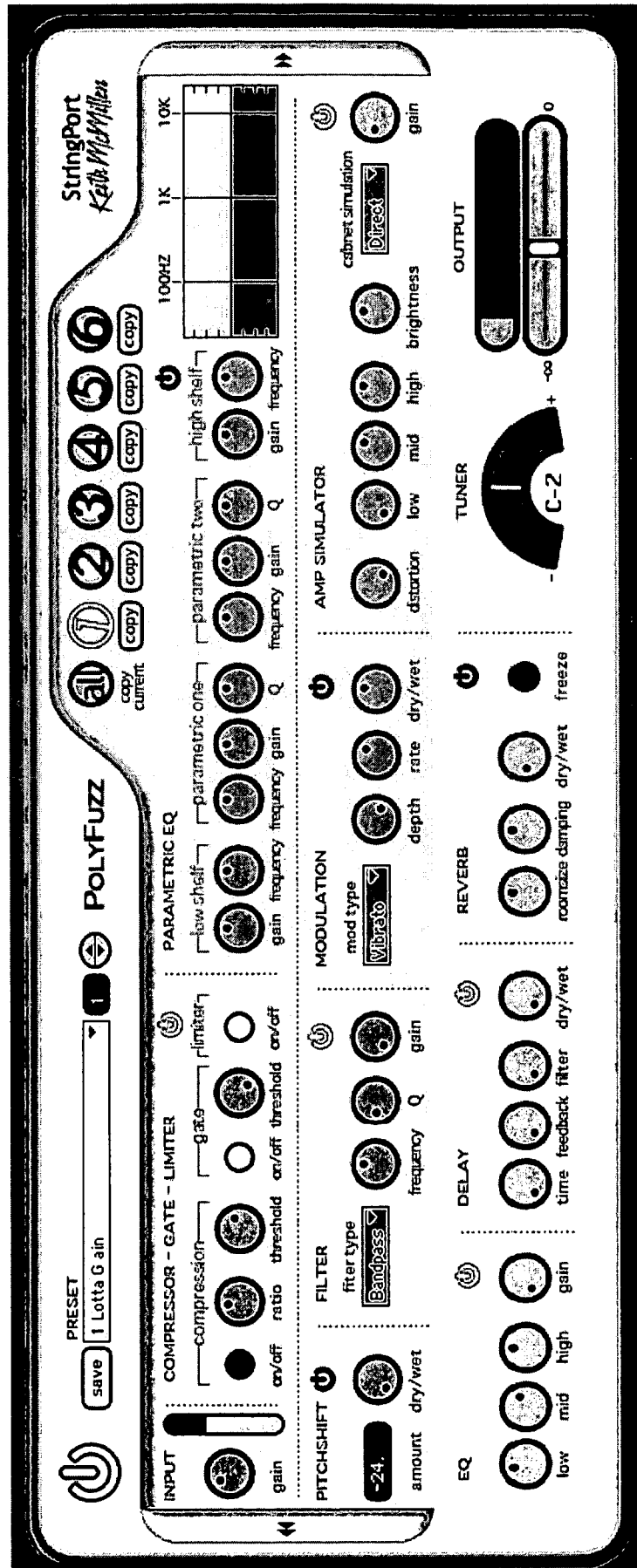
FIG. 10
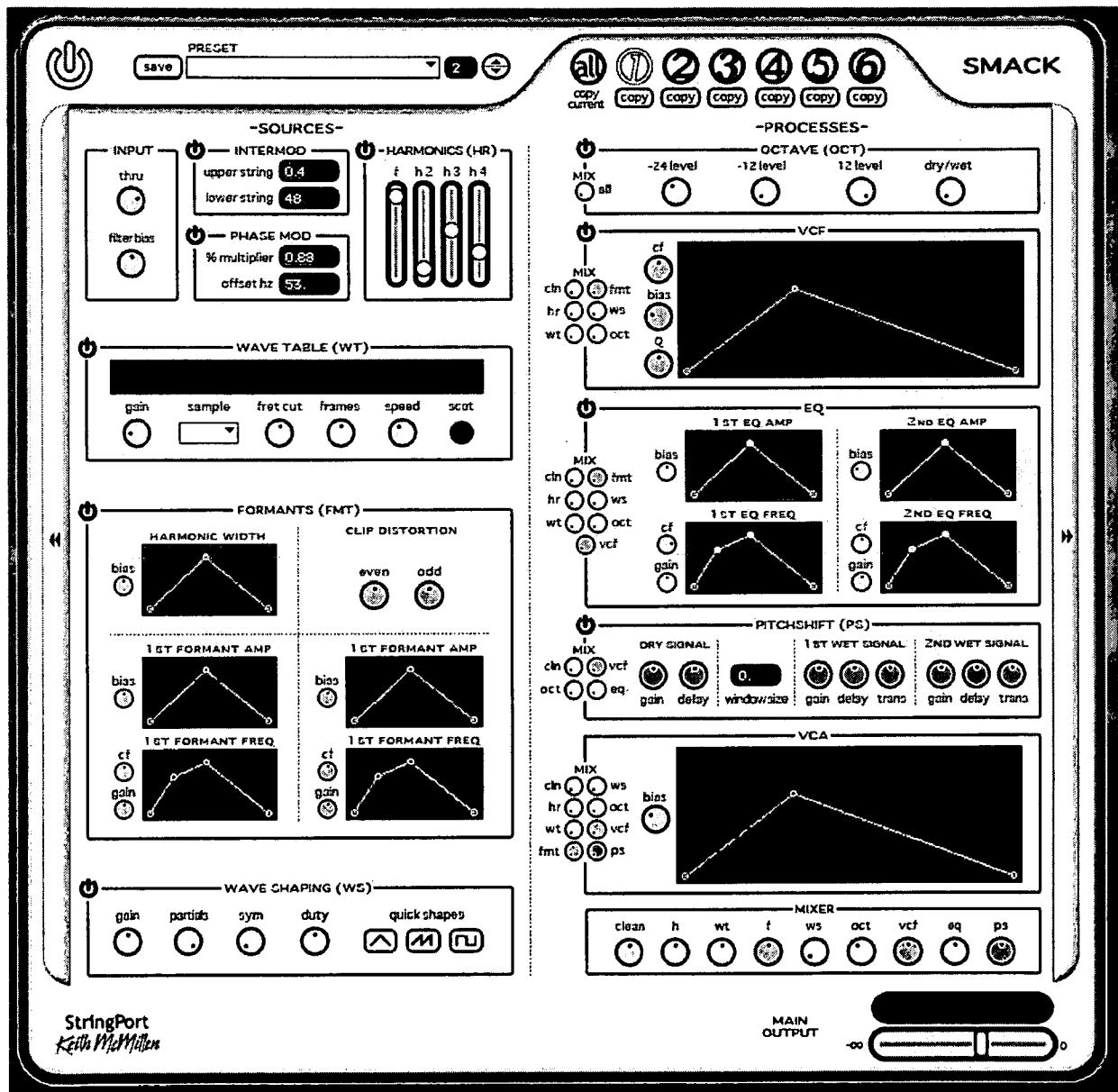
FIG. 11

FIG. 12

FIG. 13

FIG. 14

FIG. 15

FIG. 16