



- (51) **International Patent Classification:**
G10L 15/28 (2013.01) *H04R 25/00* (2006.01)
- (21) **International Application Number:**
PCT/US2014/040359
- (22) **International Filing Date:**
30 May 2014 (30.05.2014)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
61/829,242 30 May 2013 (30.05.2013) US
- (63) **Related by continuation (CON) or continuation-in-part (CIP) to earlier application:**
US 13/546,465 (CIP)
Filed on 27 February 2012 (27.02.2012)
- (71) **Applicant: EISNER, Mark** [US/US]; 193 Hemenway Road, Framingham, MA 01701 (US).
- (72) **Inventors; and**
- (71) **Applicants : HUANG, Zezhen** [US/US]; 5 Beaver Brook Road, Canton, MA 02021 (US). **DUEHNREN, David** [US/US]; 171 Paul Revere Road, Needham, MA 02494 (US).
- (74) **Agents: BURNS, David, R.** et al.; McCarter & English, LLP, 265 Franklin Street, Boston, MA 02110 (US).

(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) **Title:** SYSTEMS AND METHODS FOR ENHANCING TARGETED AUDIBILITY

(57) **Abstract:** The invention includes method and apparatus to provide a feature-rich hearing assistance device, which utilizes software that runs in the standard operating environment of commercially available Mobile Platforms. Examples of these operating systems are: iOS for Apple's iPhone/iTouch/iPad product line; Android Google utilized by many Smartphones, tablets and other Mobile Platforms; and Windows Mobile by Microsoft. In addition, the operating environment would include low level routines called by the operating environment such as device drivers and the firmware that may be used in supporting chipsets, which are included in the commercially available Mobile Platform. In conjunction with the use of these Mobile Platforms, the illustrated embodiment utilizes wired earpieces and wireless earpieces. The Mobile Platform transmits to the wireless earpieces stereo or monaural audio using communication protocols such as the A2DP Bluetooth protocol for high quality audio stereo transmission, very low latency digital transmission, or low cost specialized RF or Bluetooth transmission. [See second 4 for a description of the required earpiece. The illustrated embodiment of the invention also has the use of a convenient-to-use microphone extenders that plug into the microphone jack of the Mobile Platform. Utilized microphones may have specific features to ensure effective processing of ambient sound, especially speech.



SYSTEMS AND METHODS FOR ENHANCING TARGETED AUDIBILITY

Related Application

This application claims priority to U.S. Provisional Patent Application Serial Number
5 61/829,242, filed May 30, 2013, and is a continuation-in-part of U.S. Patent Application Serial
Number 13/546,465, filed February 27, 2012, which claims priority to U.S. Provisional Patent
Application Serial Number 61/603,633, filed February 27, 2012, and U.S. Provisional Patent
Application Serial Number 61/522,919, filed August 12, 2011 and U.S. Provisional Patent
Application Serial Number 61/506,354, filed July 11, 2011, the contents of each application is
10 incorporated herein by reference in their entirety.

Background

One need for anyone with slight-to-moderate hearing loss is speech intelligibility. Even
listeners with normal hearing can have difficulty understanding speech in a very noisy
environment. Hearing assistance products such as hearing aids and Personal Amplification
15 traditionally have some deficiencies in terms of speech clarity because of their limited ability to
detect speech. A hearing device is turned on all the time and the microphone is very sensitive,
background noise such as street traffic, wind, car engine, music/TV shows is often actually
amplified and can be very annoying and distracting. One reason that most hearing assistance
devices do not do a good job detecting speech is the complexity of speech detection in noise
20 requires powerful hardware and software that cannot be accommodated on these devices.

One goal of the systems and methods disclosed herein is to make hearing assistance algorithms
easily accessible and available by implementing such algorithms using mobile devices such as
smartphones, PDA's and the like. In exemplary embodiments, the systems and methods of the
present disclosure integrate hearing assistance algorithms with multi-media algorithms in an
25 API stack (similar to the implementation of audio effects such as stereo widening and
psychoacoustic bass enhancement).

Current hearing assistance applications for consumer devices, such as a smartphone, provide basic functionality that falls short because of operating system limitations. These operating systems are not "real-time" environments so that the user level hearing assistance application cannot guarantee an acceptable processing time. If there is significant delay (for example, 5 greater than 40ms) between the arrival at the ear of the ambient sound and the processed sound, the user will hear a confusing echo. This constraint on processing causes even the basic signal processing algorithms to run with too much delay; certainly putting the use of more sophisticated noise suppression algorithms out of reach.

Hearing loss is a global problem, with nearly 700 million people suffering from hearing 10 problems, and the rate of hearing loss accelerating around the world. As many as 47 million Americans currently suffer hearing loss in one or both ears. As the U.S. population ages, hearing loss is growing at 160% the rate of population growth. Hearing loss is now considered an emerging public health issue. In the U.S., 75% to 80% of those suffering hearing-loss do not have a hearing aid.

15 The primary barriers to wider adoption of hearing aids for those with slight-to-moderate/severe hearing loss are high cost, stigma, the need for an audiologist's involvement to make complex adjustments, and processing limitations including a lack of speech clarity, whistling due to feedback, and noise. Among these barriers, cost is the single biggest problem across market segments; for those with slight-to-moderate hearing loss the thousands of dollars 20 needed for a hearing aid purchase does not present a viable value proposition. Hearing aids are generally not covered by insurance. Alternatives such as Personal Sound Amplification products (PSAPs) are often low-cost but fail to have enough processing power and features to provide practical solutions for everyday situations.

The physically tight space in which hearing aid digital signal processing is done makes hearing 25 aid components expensive. Also the design of these components is made more expensive by feedback issues because of the close proximity of the microphone and the speaker.

Summary of the Invention

Systems and methods disclosed herein provide for low cost hearing assistance to improve intelligible hearing for those with normal hearing and to greatly improve hearing intelligibility for those with hearing problems. More particularly, the systems and methods of the present disclosure utilize commercially available general purpose central processing units (CPUs) and standard operating systems such as those inherently available in many mobile devices to improve targeted audibility, for example of speech.

A large segment of the population already own and/or regularly utilize/carry with them mobile devices with powerful general purpose CPUs, such as smartphones, tablets, music/media players, and the like. Many modern mobile devices run a standard operating system, for example Android, iOS, Windows RT, Blackberry and the like, that can accommodate applications. Thus, the systems and methods of the present disclosure reduce cost by utilizing such general purpose CPUs and standard operating systems, thereby reducing the need for expensive proprietary hardware.

Also, since many mobile devices can be used as phones or recording devices, they often include analog-to-digital converters (ADCs) and digital-to-analog-converters (DACs) that can deal with the conversion of analog sound waves from a microphone into digital and digital sound waves into analog to drive a speaker.

One important feature of the systems and methods of the present disclosure is the configuration of the digital signal processing (DSP), ADC, DAC and communications (for example, between a microphone and a mobile device and/or between a mobile device and an earpiece) such as wireless (e.g., Bluetooth) communications, so as to minimize latency. In exemplary embodiments, the systems and methods of the present disclosure result in a total latency that reduces/eliminates the perceptibility of a time delay echo, for example, less than 40ms and in some embodiments less than 20ms.

In exemplary embodiments, the systems and methods of the present disclosure may use a novel Bluetooth communication protocol, e.g., with reduced error checking, to reduce communications latency. In some embodiments, the systems and methods of the present disclosure may utilize sampling rates and buffers in ADC and DAC which reduce latency.

- 5 Additional features of the systems and methods disclosed are described in the detailed description section which follows.

Having described, herein, various exemplary embodiments of the disclosure, it is to be appreciated various alterations, modifications, and improvements will readily occur to those skilled in the art. Accordingly, the present description and drawings are by way of example
10 only. In addition, it is appreciated that exemplary embodiments presented herein do not limit the scope of the subject application.

Detailed Description

The following sections provide a detailed description of the methods used in the invention.

15 Hearing Assistance on Mobile Device

Definitions

Some terms used in the following description of an exemplary embodiment:

Speech: This term as used herein may refer to speech that is spoken by a source to whom the User is listening and the representation of that speech by an audio signal. This is also referred
20 to as "targeted speech" when it is advantageous to differentiate the speech that is listened to from the other speech that is considered a component of noise.

Ambient Sound: Sound as picked up by a microphone that is traveling through the air in the proximity of the Mobile Computing Device.

Audio: Either an analog or digital electronic transmission that represents sound.

25 **Noise:** The component of sound or an audio that is not the targeted audio or sound.

Mobile Computing Device: A computing device that has a general purpose CPU, runs standard operating systems, and is designed to be portable.

Speaker: A device that translates analog audio into sound.

User: An entity that is using an embodiment of the invention.

5 **Mobile Computing Device**

In exemplary embodiments, systems and methods may utilize a Mobile Computing Device to process ambient sound and enhance the audibility of a targeted portion of the ambient sound, for example, to enhance speech. As used herein, a Mobile Computing Device may refer to any portable or semi-portable device that includes a general purpose CPU and a standard operating
10 system capable of executing an application — for example, an application stored in memory.

Examples of Mobile Computing Devices may include smartphones, tablets, laptops, PDAs, media players, such as mp3 players, and the like.

Examples of general CPUs are ARM Cortex-A9, Samsung S5PC100, and TIOMAP4 Platform, Apple A4. The key definition of a general CPU is that it has a general instruction set that may be
15 augmented by specific instructions or co-processors to do DSP, but a general CPU is designed to execute a wide range of applications. General purpose CPUs are thus distinguished from proprietary processing chipsets found in dedicated hearing aid devices, in that they are designed to handle a wide variety of applications.

Examples of standard operating systems are any of the versions of the Microsoft Windows
20 mobile operating systems (Windows RT), different releases of the UNIX and Linux operating systems (e.g., Android), any version of the iOS, or any other non-proprietary operating system capable of running on the Mobile Computing Device and performing the operations described herein.

In exemplary embodiments, a Mobile Computing Device may include or otherwise be
25 operatively associated with one or more sensors for detecting ambient sound, such as a

microphone. A microphone may either be an integral component to the Mobile Computing Device or an external component operatively associated with the Mobile Computing Device via a wired or wireless connection.

In exemplary embodiments, the Mobile Computing Device may include or otherwise be operatively associated with one or more speakers for outputting sound processed by the mobile device. A speaker may be embedded in an integral component to the Mobile Computing Device or embedded in an external component operatively associated with the Mobile Computing Device via a wired or wireless connection, for example a headset, earphones and the like. In some embodiment, a same external component, for example a headset, may include both a speaker and a microphone.

Aggregate Latency

Fig. 1a and Fig.1b describe the components of latency from input Sound reaching the microphone to processed Sound output from the Speakers. The key feature of the systems and methods of the present disclosure is the construction and configuration of the digital signal processing (DSP), ADC, DAC and communications (for example, between a microphone and a mobile device and/or between a mobile device and an earpiece) such as wireless (e.g., Bluetooth) communications, so as to minimize aggregate latency, i.e., the latency is sufficiently low such that a user will not perceive an echo-like delay between the processed sound exiting from the speaker and the raw ambient sound entering the User's ear. In exemplary embodiments, the systems and methods of the present disclosure result in aggregate latency, for example, of less than 40ms and in wireless embodiments and less than 25ms in wired embodiments.

Audio Processing on Mobile Device

Fig. 2 depicts an exemplary embodiment of a general method 9000, for processing ambient sound, including improving the audibility of targeted speech. Method 9000, including steps of

receiving via an internal or external microphone ambient sound; converting the ambient sound to a digital audio signal using an ADC [9001]; performing DSP to process the digital audio signal and enhance the audibility the ambient sound [9002], where the SP outputs a digital audio signal that is either further processed as a digital audio signal; or converted, using a DAC [9003],
5 to an analog audio signal, which is then transmitted to a Speaker.

Fig. 3 depicts a block diagram of the processes utilized in an exemplary embodiment to increase the audibility of targeted speech, which utilizes the Apple's iOS family of operating systems.

Audio processing in the iOS operating system is based on event-oriented processing. The **RCEngineMgrDelegate** [8040] is the primary event handler, processing events and setting
10 state variables. It is the primary mediator between the general application processes, e.g., the User Interface, and the active audio processing modules.

General application processes

The **ViewControllers** [8010] manage the display and interaction of the User Interface (UI), e.g., signaling to the **RCEngineMgrDelegate** that audio processing should be initiated, passing a
15 parameter to the **RCEngineMgrDelegate** to change in the volume setting. The **ViewControllers** also communicate with **RCPreferences** to display and update User-entered profile information.

RCPreferences [8020] manages the User settable preferences and profiles such as instantiating stored Hearing Profile or Equalizer Profile or retrieving a saved Sample Noise Profile.

RCPreferences interfaces with the **RCPermStoreDelegate** to either retrieve or update storable
20 User preferences and profiles.

RCPermStoreDelegate [8030] mediates between **RCPreferences** and the various mechanisms for permanently storing data, e.g., Hearing Profiles, Sample Noise Profiles, etc., delegating to the appropriate process and indicating the CRUD operation that is required.

RCProfileFiles [8031] stores and retrieve User profiles, such as Hearing Profiles, Equalizer
25 Profiles and Sample Noise Profiles in the iOS file system.

OS X User Defaults [8032] retrieves and updates in permanent storage User preferences and other parameters used when the exemplary embodiment is initiated or are changed during the execution of the exemplary embodiment.

Audio processes

5 **RealClarityAudio** [8050] is the audio engine that manages the processing of digital audio. An instance of RealClarityAudio is instantiated when the exemplary embodiment is started and initiates the processing of a digital audio signal by iOS. RealClarityAudio then provides the overall management of the processing, specifically by instantiating the Audio Processing Graph unit.

10 **Audio Processing Graph** [8060] is an object that contains an event-oriented flow describing processes to be executed based on call-backs from iOS. These flows provide the key set of functions that need to be executed by the exemplary embodiment to increase the audibility of speech that is being delivered within the digital audio input. The major call-back executes the exemplary embodiment's DSP. Additional call-backs include a Speaker Response Estimator and
15 a Noise Estimator.

RealClarity DSP [8061] contains the exemplary embodiment's algorithm that performs the core DSP to increase audibility. That algorithm is described in "Digital Signal Processing", below.

Speaker Response Estimator [8062] is a unique process, triggered from a UI screen, that
20 generates white noise that is broadcast through the Speakers of an earpiece and input through the Mobile Computing Device's microphone. The Estimator creates an adjustment profile calibrated to correct gain anomalies in the Speaker of the wired earpiece, based on the difference between the expected noise profile of white noise and the actual noise profile output from the Speaker. The adjustment profile, which is stored, enables the RealClarity DSP to adjust
25 for the anomalies.

Noise Estimator [8063] is a process, triggered from a UI screen, that creates a Noise Profile based on, for example, a 5 sec audio stream of the ambient noise in an environment. This Noise Profile is stored and is then available to be utilized by the RealClarity DSP.

Digital Signal Processing – Overview

5 This section describes the DSP algorithms used in the exemplary embodiment to improve the audibility of speech. In the exemplary embodiment, DSP is performed within a time constraint acceptable for aggregate latency, i.e., a time period such that the brain can integrate the processed sound with the ambient sound directly coming to the ear. This aggregate latency, from microphone to Speaker, is best if it is around 20ms or 25ms. However, an aggregate
10 latency up to around 40ms may be tolerable by most people. The exemplary embodiment discloses an effective set of DSP processes and a system design that fit within the aggregate time constraint, i.e., a target delay at 10ms for all signal processing (ADC, DSP and DAC).

DSP is typically implemented by way of an application which is stored and executed, without requiring changing the underlying firmware of the device. Thus the application is upgradable
15 and may be utilized with any number of different mobile devices that support the same operating environment. In the exemplary embodiments, the core algorithms are coded in the “C” computer language. C-based source can be ported to all current mobile operating systems. Alternate embodiments of the core DSP processing can be implemented with low cost. For example, by modifying the core DSP to interface with standardized API stacks, such as the open
20 source OpenMAX AL (OpenMAX AL defines an interface between multimedia applications, such as a media player, and the platform media framework), the DSP can easily migrate to different platforms including Android.

The exemplary embodiment utilizes the standard chipsets in a Mobile Computing Device, requiring no proprietary hardware. Since the power of these chipsets (given Moore's law that is
25 certainly being followed in the mobile-device space) is dramatically increasing, progressively

more sophisticated algorithms, particularly for noise reduction and speech clarity, will be enabled.

DSP Processing – The Algorithm

This section describes DSP for an exemplary embodiment with the primary goal of reducing the speech-to-noise ratio. These signal processing algorithms can be applied to electronic audio as well as ambient sound. However, the time constraint on DSP relates to ambient sound where it is important to avoid an echo effect, i.e., to deliver sound to the speaker with an aggregate latency of less than 40ms.

In general, DSP is performed on a real-time or high priority thread utilizing Call-backs from the operating environment.

In the exemplary embodiment, the DSP component of aggregate latency is reduced by executing an effective set of algorithms for the DSP, where these algorithms are driven by parametric input. The values of the parameters are derived by background processing or from User input such that computation of these parameters does not add processing latency to the real-time DSP. The parametric input is either supplied as arguments to the DSP process or indirectly via profiles, sound samples, and state variables stored in a shared common memory space.

An effective set of DSP could include, but are not limited to, gain control and gain shaping, frequency gain adjustment, frequency mapping, dynamic range compression, noise suppression, noise removal, speech detection, speech enhancement, detection and suppression of non-speech impulse sound.

The exemplary embodiment (as shown in Fig. 4) contains an effective set of DSP algorithms that have been implemented for mobile devices running Apple's iOS operating environment. The DSP processing takes a frame of digital audio input in the time domain, transforms it into a Frequency Spectrum using a Fast Fourier Transform, processes that Frequency Spectrum and reconstructs a frame of digital audio output. There may be averaging or smoothing done

between sequential Frequency Spectrum and between time domain audio frames. The DSP process, including buffering, is designed to take less than 10 ms.

In the exemplary embodiment, the DSP is based on a filter bank architecture with the following components:

5 **Audio Input [1]**

The audio input [1] is a digital stream that can come from a number of sources, such as the electronic sound from applications running on the mobile device or telephone conversations. In the exemplary embodiment, the primary audio input comes from an analog-to-digital converter which receives its analog signal from an internal microphone or from an external microphone.

10 **Fast Fourier Transform (FFT) [2]**

The time domain signal is then converted to the frequency domain using a Fast Fourier Transform [2] by transforming a time frame with 256 samples to a Frequency Spectrum of 256 bins where the frequency is represented by a complex number indicating the amplitude and phase of the bin.

15 All FFT-based measurements assume that the signal is periodic in the time frame. When the measured signal is not periodic then leakage occurs. Leakage results in misleading information about the spectral amplitude and frequency. The exemplary embodiment applies a Hann Window transformation to reduce the effect of leakage.

One of the disadvantages of windowing functions like Hann is that the beginning and end of the
20 signal is attenuated in the calculation of the spectrum. This means that more averages may be taken to get a good statistical representation of the spectrum. This would increase the latency of FFT algorithm. A 75% overlap process is implemented in the exemplary embodiment where only 64 samples are added and the remaining 196K come from the previous window. This moving average approach minimizes latency while compensating for the attenuated signal. The

expected latency including the buffering of the time frame and the delay because of the averaging is estimated to be 5.8ms.

Manual Noise Estimator [3]

The speech enhancement process [4] has a capability to perform continuous noise estimation, i.e., estimate what is noise. However, if a User is in a stable noise environment the speech enhancement algorithms work better with a fixed measurement of the noise profile. The Manual Noise Estimator process gets input from the FFT process and creates a stable Noise Profile. The Noise Profile is output as a Frequency Spectrum, which then can be input to the Speech Enhancement process.

In the exemplary embodiment, the creation of a noise sample by the Manual Noise Estimator process is initiated by the User pressing the “sample noise” control in the Filter screen [Fig. 6.16]. Sound is then gathered for a period of five seconds, the sound is transformed by the FFT process and input to the Manual Noise Estimator process that will create a Noise Profile. The created Noise Profile is then stored in the Noise Profile buffer where it will then be accessed by the Speech Enrichment process.

Given the creation of the Noise Profile [Fig. 6.17], the User has the option of naming and saving the created Noise Profile for later use [Fig. 6.18]. Rather than creating a current noise sample the User can select a stored Noise Profile [Fig. 19]. The selected Noise Profile will be stored in the Noise Profile buffer where it can be accessed by the Speech enhancement module.

There are three parametric arguments to the Manual Noise Sample process that are used to inform the process controller of the status of the Noise Profile creation:

#	Argument Name	Argument Description
1	manEstTime	How much time has elapsed in gathering the noise sample
2	manEstRunning	(Boolean) is the noised sampling running
3	manEstRunning	(Boolean) is noise sampling ready

Speech Enhancement [4]

In the exemplary embodiment, the Speech Enhancement process is the core process for improving the speech-to-noise ratio by removing noise from the audio input. The process implements an algorithm described by Diethorn (SUBBAND NOISE REDUCTION METHODS FOR SPEECH

5 ENHANCEMENT, Eric J. Diethorn, *Microelectronics and Communications Technologies, Lucent Technologies*) that is “less complex” so that it does not significantly add to the aggregate latency. The algorithm consists of four key processes: sub-band analysis, envelope estimation, gain computation, and sub-band synthesis [Fig. 5].

The Speech Enhancement algorithm is designed to continually estimate the noise component of
10 the audio 10 input (in the diagram).

However, if the User has indicated that a Manual Noise estimate should be used, then the Frequency Spectrum stored in the Noise Buffer will be used.

The Speech Enhancement process also estimates when speech is present through a soft Voice Activity Detection algorithm (VAD). The VAD limits the possible gain reduction for noise. In
15 some situations, it may be possible to substitute the background-computed time-domain estimate of when speech is present. The time domain estimate may be more accurate and would allow more flexibility in terms of gain reduction.

There are seven arguments that the Speech Enhancement process. Two of the parameters control a tradeoff between increasing noise identification and delivering clearer speech. A User
20 can set the balance of this trade-off by modifying a slider on the Advanced Filter screen [Fig. 6.19].

There are four parameters, used for a smoothing function, that indicate how to transition to Speech and to noise. These parameters are preset, but can be changed through a “back-door” UI available to a developer. There may be situations where different parameter sets will be used
25 depending on the environment.

There is also a Boolean argument that indicates whether a manual noise estimate is to be used.

#	Argument Name	Argument Description
1	pEnhAm	Amplitude
2	pEnhTh	Threshold
3	pEnhSA	The attack smoothing parameter for start of speech
	pEnhSD	The decay smoothing code for end of speech
	pEnhNA	The attack smoothing code for start of noise
	pEnhND	The decay smoothing code for end of noise

The output of the Speech Enhancement process is a Frequency Spectrum with an increase in Speech-to-Noise ratio.

Broadband Squelch [5]

- 5 While the Speech Enhancement process increases the Speech-to-Noise ratio, it can be advantageous to remove low frequency sounds that are not part of speech, such as the rumble of an air conditioner or other machinery. In the exemplary embodiment, the Broadband Squelch process removes these frequencies from the Frequency Spectrum. While that low frequency noise will still be heard by Users as ambient sound reaching their ears, it will not be presented
- 10 in the audio output for the Speaker.

The level of low frequency sound to be removed is chosen by the User by setting the lower slider control on the slider bar on the Filter screen [Fig. 6.16].

The Broadband Squelch has three controlling arguments:

#	Argument Name	Argument Description
1	pSquelchKnee	Controls how the Squelch is averaged in
2	pSquelchTH	The squelch threshold as indicated by the User
3	pSquelchDecay	The length of smoothing to remove the Squelch when the low frequency noise is gone.

- 15 The output of the Broadband Squelch process is a Frequency Spectrum with the low frequencies appropriately removed.

User Profile [6]

The most important feature of hearing assistance is to be able to adjust the gain of different frequencies to match the User's hearing ability and hearing preference.

In the exemplary embodiment, the User Profile process accesses a Profile buffer, which is constructed by combining a Hearing Profile and an Equalizer Profile, to adjust the gain for frequencies in the Frequency Spectrum that is output from the Broadband Squelch process.

Since the User's hearing may vary between left and right ear, separate Hearing Profiles can be constructed for each ear, which are then combined with Equalizer Profiles, so that separate left ear and right ear profile buffers are provided to the DSP where separate left and right gain adjustments can be made (Fig. 4).

In the exemplary embodiment, Users have two ways to set up an appropriate Hearing Profile:

1. On the Clarify screen [Fig. 6.3], the right-side wheel allows a User to select one of a number of pre-stored Hearing Profiles. The pre-stored Hearing Profiles, for example, can represent average hearing loss profiles by age. The stored Hearing Profiles cover the normal frequency range and decibel deficit that are used in standard hearing tests. While Users may initially select a Hearing Profile that reflects their age, they may experiment through use and find other profiles that better match their hearing needs. The innovative use of these pre-stored Hearing Profiles allows many Users to adjust the sound output, such that they do not have to use the results of a hearing test to adequately meet their hearing needs.
2. Users can enter an audiogram that represents their personal hearing needs. This is done on the Hearing Profile screen. [Fig. 6.9] The User is given the option of entering one profile for both ears or entering separate profiles for each ear. Users with moderate to severe hearing loss and those with distinctive hearing needs are best suited to utilize the custom entry of an audiogram. Entered audiograms can also be named and saved so that

a User can define different Hearing Profiles for different situations and environments. A

User can select a named Hearing Profile on the myProfile screen. To use an entered

Hearing Profile, the right-side wheel on the Clarify screen is set to the array icon.

In the exemplary embodiment, the User can make additional adjustments to fit particular sound situations or their own hearing preferences by adjusting the Equalizer Profile, e.g., emphasizing the frequencies most used for speech, increasing the higher frequencies to get a better experience listening to music. The Equalizer Profile defines a set of additive gain amounts that modify the Hearing Profile.

In the exemplary embodiment, Users have two ways to setup an appropriate Equalizer Profile:

1. On the Clarify screen [Fig. 6.3] the left-side wheel allows a User to select one of a number of pre-stored Equalizer Profiles. The pre-set Equalizer Profiles have frequency gain settings for common sound situations.
2. Users can enter a customized profile on the Equalizer screen [Fig. 6.5], the amount of gain adjustment is indicated in the central vertical scale. Entered Equalizer Profiles can also be named and saved so that Users can define their own set of profiles for different sound situations and environments. A User can select a named Equalizer Profile on the Equalizer Select screen [Fig. 6.7]. To use an entered Equalizer Profile, the left-side wheel on the Clarify screen is set to the array icon.

Broadband AGC [7]

The Broadband AGC (automatic gain control) process adjusts the overall gain of the Frequency Spectrum to compensate for volume changes in the sound environment, e.g., going from a quiet environment to a loud environment. This is to make sure that a User does not hear any abrupt changes in the sound from the Speaker. The Broadband AGC process is important as it removes the threat that delivered sound from the Speaker would be loud enough to damage a User's hearing ability. The Broadband AGC process measures a moving average of the audio energy represented in the Frequency Spectrum to ascertain significant changes and will limit the absolute gain and will smooth the gain during an environmental transition. This Broadband AGC process cannot operate at low levels of sound energy, as the results would be too volatile, so an energy threshold may be set that indicates the energy level when the automatic gain control is activated.

The Broadband ADC process has two controlling parameters:

#	Argument Name	Argument Description
1	pComp1A	Controls the introduction of the Squelch
2	pComp1Th	Threshold needed to activate the automatic gain control

The output of the Broadband AGC is an adjusted Frequency Spectrum.

Volume Control process [8]

In the exemplary embodiment, in addition to setting the hardware volume, Users have the option of setting a software volume level. The software volume is set on the main RealClarity screen using the “boost” control. The Volume Control process adjusts the gain in the Frequency Spectrum for each time frame to reflect the volume control setting specified by a User. Offering the software volume control is important as it means that knowledge of the volume level and specifically changes in the set volume level are known to the DSP. The performance of the DSP is affected by the volume setting, in particular if the volume is too high feedback can be introduced. The best practice for a User would be to set the hardware volume at one level near its maximum and only modify the software volume control.

The Volume Control process has one control parameter:

#	Argument Name	Argument Description
1	pLG	Gain level that corresponds to the USER set volume control

The output of the Volume Control is an adjusted frequency Spectrum.

Broadband Limiter process [9]

When a sudden loud noise occurs, which has a broad frequency spectrum, it can distract from the normal speech processing, e.g., a plate drops making a loud noise next to a User talking in a restaurant. The Broadband Limiter process recognizes a potential loud noise interruption through a sudden increase to a high level in the energy of the audio signal. On recognizing the appearance of a sudden noise, the Broadband Limiter will reduce the overall gain in the Frequency Spectrum.

The level of volume that is to be considered a sudden loud noise is chosen by the User by setting the upper slider control on the slider bar on the Filter screen [Fig. 6.16].

The Broadband Limiter has two controlling parameters:

#	Argument Name	Argument Description
1	pBblimitTH	The energy threshold that constitutes a loud noise
2	pBblimitdecay	The length of time where the gain is returned to normal when the noise has abated

The output of the Broadband Limiter process is a modified Frequency Spectrum.

Multiband Limiter process [10]

It is possible that the Frequency Spectrum contains non-zero amplitude for frequencies outside of the range for which the Speaker can produce sound. In that case the Speaker will produce sound at its maximum for all frequencies above that physical limit. This will produce distortion.

5 The Multiband Limiter cuts off these high energy peaks preventing that distortion.

The Multiband Limiter process has one controlling parameter:

#	Argument Name	Argument Description
1	pCompTh	Threshold level to cut off the high energy peaks

The output of the Broadband Limiter process is a modified Frequency Spectrum.

Inverse Fast Fourier Transform [11]

The Inverse Fast Fourier Transform converts the Frequency Spectrum produced by the DSP

10 back to a time domain audio signal.

The process is based on the Diethorn algorithm that accurately reconstructs the audio stream.

The Diethorn algorithm is designed so that if the audio input signal [1] is transformed by the Fast Fourier Transform [2] and the resulting Frequency Spectrum is then inverted by the

Inverse Fast Fourier Transform [11], with no intervening processing, the original audio signal

15 will be near perfectly reproduced.

Audio output [12]

The reconstructed Audio output of the DSP is feed to a digital-to-analog converter. The analog signal is sent to the Speaker, which produces the processed sound for the User. Transmission to the Speaker can be through wired connections, for example, utilizing a standard audio jack or

20 USB connector that is part of the Mobile Computing Device. Transmission can also be through a radio component utilizing standard transmission protocols such as analog FM, digital FM, as long as the latency of that transmission maintains an aggregate latency of under 40ms. The exemplary embodiment includes an invention of a proprietary Bluetooth protocol. Use of the proprietary Bluetooth protocol requires a modification of the DSP algorithm.

Microphone Input

The exemplary embodiment takes advantage of multiple microphones to receive input if available on the Mobile Computing Device. For example, the iPhone 5 has three built-in microphones and can also receive input from an external microphone, such as a microphone
5 that is associated with a wired or wireless earpiece.

The exemplary embodiment can:

- Select the mic that is closest to the sound as the primary and the others as secondary, and thus improve the Signal-to-noise profile that is provided to the DSP.
- Use the mics in pairs for directionality, both in determining where the sound is coming
10 from, especially towards and away from the targeted Speech, and adjusting overall gain that is delivered to the DSP.

Utilize the external microphone associated with the earpiece to recognize speech from the User to adjust (lower) the speaker's volume in comparison to the speech of a targeted speaker.

The User Interface

15 Mobile Computing Devices display digital content and controls to Users through an intuitive User Interface (UI) displayed as discrete screens. The UI may include, for example, various windows, tabs, icons, menus, sub-menus, and touch screen controls such as radio buttons, check boxes, slider bars, etc.

Describing a set of primary screens of an exemplary embodiment can instruct and describe the
20 underlying processes that they control. The described screens are for an exemplary embodiment implemented on an Apple device supporting iOS 7, which utilizes that devices touch screen interface.

Main Screen (“RealClarity”)

This is the main screen. This screen, as well as most other screens, has a ‘share’ button and an ‘info’ button in the upper right corner.

- The ‘share’ button activates a screen that allows a User to communicate with others about the app. The ‘share’ button can be used to send and share audio profiles, noise profiles, customized equalizer settings, etc.
- The ‘info’ button produces a text screen that provides information about source screen.

This screen, as well as numerous other screens, has a vertical display slider which visually shows the presence of audio input through a colored column. If there is no column displayed then no source input is being received, most often because the ‘On/Off’ button is set to Off.

In the upper left corner has an ‘option’ icon that when pushed the main screen to the left and reveals the ‘Options’ panel.

“RealClarity” screen, Fig. 1

When the exemplary embodiment application is loaded, the main screen “RealClarity” is displayed. The application is activated by pushing the ‘On/Off’ button control.

“RealClarity” screen, Fig. 2

The lighted ‘On/Off’ button indicates the application is active.

Two adjustments for volume are available.

The slider bar labeled “volume” corresponds to overall device volume, which may also be adjusted using hardware buttons on the device, in some embodiments. It is best that this control be close to the maximum as the gain reflected in the setting is out of the purview of the exemplary embodiment processing.

The 'boost' stepper allows you to change the internal volume (or gain) of the audio as [processed by the embodiment. The best sound quality is achieved by first maximizing the hardware volume, and then increasing the internal volume.

As the volume increases, the likelihood of unwanted audio feedback may increase. The audio feedback may be decreased by reducing either the 'boost' stepper or 'volume' slider.

This main "RealClarity" screen includes two large buttons, the 'filter' button which deals with noise control and the 'clarify' button which deals with gain adjustments.

Selecting the "Clarify sound" button will display the "Clarify" screen.

Selecting the "Filter noise" button will display the "Filter" screen.

10 Clarify Screens

These screens create and modify the frequency gain profiles used in the DSP processing.

"Clarify" Screen, Fig. 3

The 'Clarify' screen has two wheel controls and a 'Customize' button. The two wheels allow a User to adjust the clarity of the processed sound by modifying a Hearing Profile or an Equalizer Profile. The 'Customize' button allows the User to create or modify a Profile or activate a stored Profile. Custom settings may be set by spinning the wheel to the setting where you will find the slider icon.

The symbols on the left wheel allow the User to select a pre-set Equalizer Profile setting, for example, Profiles may be selected for Speech, TV, Outdoors, Music, Movie and Live Event. There is an option to select "Off", which means to not use an Equalizer Profile and a setting, with a tuner icon, which means to use the selected customized Equalizer Profile.

The symbols on the right wheel allow the User to select a pre-set Hearing Profile. The preset Hearing Profiles reflect average hearing loss by age from 40 to 85 in increments of 2 or 3 years. The age chosen is shown in the small wheel. There is a flat-line setting for ages below 40. In

general, the higher the age, the more amplification there is for medium and high frequency sounds. Users can start with a setting close to their age, and then experiment up or down to find the setting that works best for them in different environments.

Selecting the “Customize” button will bring up the pop-up “ClarifyCustom” screen.

- 5 The “Clarify” screen has a ‘return’ control (left carrot) in the upper left corner, as do many other screens, that, if selected, returns to the calling screen.

“ClarifyCustom” pop-up Screen, Fig. 4

The “ClarifyCustom” screen displays three buttons that allow the User to customize input and has a “Cancel” button that returns to the “Clarify” screen. The User can enable and/or customize
10 a number of features with respect to the clarity of the desired sound.

- The “Customize the equalizer” allows the User to enter to enter or modify an Equalizer profile by bringing up the “Equalizer” screen.
- The “Enter your audiogram” allows the User to enter of modify a Hearing Profile by bringing up the “Hearing Profile” screen.
- 15 • The “Optimize headphone sound” allows the User to create a base profile that corrects for frequency anomalies in a wired earpiece by bringing up the “Headphone” screen.

Equalizer Profile Screens

These screens handle the details of creating, modifying and saving the Equalizer Profiles.

“Equalizer” Screen, Fig. 5

- 20 The “Equalizer screen allows the User to modify the current active Equalizer Profile, which is displayed. The Equalizer Profile shapes sound, much like the treble and bass controls on a stereo, but with more fine-grained frequency tuning. The horizontal axis displays the frequencies that can be set. The key voice frequencies are 500Hz to 4 KHz. The vertical axis

displays the decibels that will be added to the gain of a frequency. The display bar at the bottom of the screen identifies the current active Equalizer profile.

To modify the displayed Equalizer Profile, the User moves the frequency sliders to the shape desired.

5 The User can then select the 'return' control return to the calling Screen:

- If the User has modified the frequency settings and selects the 'return' control a "Save EQ file now" pop-up screen is displayed.
- If the User does not modify the displayed Equalizer Profile and selects the 'return' control then the calling screen is displayed and the displayed Equalizer profile remains the active
10 profile.

The User can save or activate a stored Equalizer Profile by selecting the 'next' control (right carrot) at the bottom right corner of the screen.

- If the User has not modified the displayed Equalizer Profile, the "Equalizer Select" screen is displayed, which allows the user to activate a saved Equalizer profile.
- 15 • If the User has modified the displayed Equalizer profile, the "Equalizer Name" screen is displayed, which requires the User to name the modified Profile and stores it. On having stored the named Equalizer Profile, the "Equalizer Select" screen is displayed with the newly saved Equalizer activated.

20 The User can save or activate a stored Equalizer Profile by selecting the "Save" button at the top right corner of the screen.

- If the User has modified the displayed Equalizer profile, the "Equalizer Name" screen is displayed, which requires the User to name the modified Profile and stores it. On having stored the named Equalizer Profile, the "Equalizer Select" screen is displayed with the newly saved Equalizer activated.
- 25 • If the User has not modified the displayed Equalizer Profile, no action occurs.

“Equalizer Name” screen, Fig. 6

The Equalizer name screen allows the user to name and save the modified displayed Equalizer Profile. When the name is entered and “return” on the keyboard is selected, the Equalizer Select screen is displayed. The newly stored and named Equalizer Profile will be listed and checked as
5 active.

“Equalizer Select” screen, Fig. 7

The “Equalizer Select” screen displays the set of saved Equalizer Profiles. The currently active Equalizer Profile is indicated by a check on the list of saved profiles. The User can activate another Equalizer Profile by selecting a name on the list. The check mark will move to that
10 entry indicating that that profile is now the active Equalizer Profile.

“Save EQ file now” pop-up screen, Fig. 8

If the User selects the “No, perhaps later” button, then the currently displayed Equalizer Profile becomes the active Equalizer Profile and the calling screen is displayed.

If the User selects the “Save” button, then the “Equalizer Name” screen is displayed. The User
15 can then name the modified displayed Equalizer Profile and save it. When saved it will be the active Equalizer Profile.

Hearing Profile Screens

These screens handle the details of creating, modifying and saving the Hearing Profiles.

“Hearing Profile” screen, Fig. 9

20 This screen allows the User to modify the current active Hearing Profile (aka audiogram), which is displayed.

The Hearing Profile provides input to the DSP to add frequency-based gain to improve the audibility of Sound. The Hearing Profile contains separate profile components for the right and

left ear. The vertical axis displays the decibels that will be added to the gain of a frequency. The vertical bar is inverted so that the frequency display mimics a typical audiogram that shows hearing loss in decibels, which increase at the lower settings. The display bar at the bottom of the screen identifies the displayed active Hearing Profile.

- 5 To modify the displayed Hearing Profile component, the User moves the frequency sliders to the shape desired.

The horizontal button bar selects the Hearing Profile component to display. The “Left” button displays the left-ear Hearing Profile component and the “Right” button displays the right-ear Hearing Profile component. If the Hearing Profile left and right components profiles are the
10 same then the User can select the “Both” button. The supplied pre-set Hearing Profile have the same profile for both the left and right ears. Modifications on the “Both” displayed screen will be recorded in both the right-ear and left-ear Hearing Profile components. If there is a difference between the right and left component then modifying the displayed Hearing Profile will only modify the right-ear Hearing Profile component.

- 15 The User can then select the ‘return’ control to return to the calling Screen

- If the User has modified the frequency settings and selects the ‘return’ control a “Save your profile now?” pop-up screen is displayed.
- If the User does not modify the displayed Hearing Profile and selects the ‘return’ control then the calling screen is displayed and the displayed Hearing Profile remains the active
20 profile.

The User can save or activate a stored Hearing Profile by selecting the ‘next’ control (right carrot) at the bottom right corner of the screen.

- If the User has not modified the displayed Hearing Profile, the “Profile Select” screen is displayed, which allows the user to activate a saved Hearing Profile.

- If the User has modified the displayed Equalizer profile, the “myProfile Name” screen is displayed, which requires the User to name the modified Hearing Profile and then stores it. On having stored the named Equalizer Profile, the “Profile Select” screen is displayed with the newly saved Equalizer activated.

5 The User can save or activate a stored Equalizer Profile by selecting the “Save” button at the top right corner of the screen.

- If the User has modified the displayed Hearing Profile, the “myProfile Name” screen is displayed, which requires the User to name the modified Hearing Profile and stores it. On having stored the named Hearing Profile, the “Profile Select” screen is displayed with the
10 newly saved Hearing Profile activated.

- If the User has not modified the displayed Hearing Profile, no action occurs.

“Profile Select” screen, Fig. 10

The “Profile Select” screen displays the set of saved Hearing Profiles. The currently active Hearing Profile is indicated by a check on the list of saved profiles. The User can activate
15 another Hearing Profile by selecting a name on the list. The check mark will move to that entry indicating that that profile is now the active Hearing Profile.

“Save your profile now?” pop-up screen, Fig. 11

If the User selects the “No, perhaps later” button, then the currently displayed Hearing Profile becomes the active Hearing Profile and the calling screen is displayed.

20 If the User selects the “Save” button, then the “myProfile Name” screen is displayed. The User can then name the modified displayed Equalizer Profile and save it. When saved it will be the active Hearing Profile.

Headphone (wired earpiece) Screens

These screens initiate a test of Speakers in a wired earpiece to identify any anomalies in the frequency gain. This is done by executing the Speaker Response Estimator process that results in an active Headphone Profile. The resulting Headphone Profile can be stored for later sessions
5 that use the same earpiece.

“Headphone” screen, Fig. 12

Each model of earpiece has its own frequency characteristic or profile. This screen allows the exemplary embodiment to measure that characteristic. Once measured the sample is used to create a profile that is used by the DSP to produce the best sound possible and to minimize the
10 likelihood of audio feedback. The smaller earbuds often have a frequency bump that can cause feedback.

To perform an optimization, the User sets or holds the earpiece as pictured. The best results are obtained 1) by doing it in a relatively quiet place, and 2) by setting the hardware volume control about two-thirds of the way to the right. Then the User selects the “Start” button. The

15 “Optimizing headphone screen” pop-up screen is displayed.

The bar at the bottom of the screen, displays the name of the active Headphone Profile.

“Optimizing Headphone” screen, Fig. 13

The optimization process takes about 15 seconds. This screen displays the duration of that optimization process. When the process is complete the “Save optimization?” pop-up screen is
20 displayed.

“Save optimization?” screen, Fig. 14

If the User selects the “Just use” button, then the computed optimization is the active optimization profile for the current session.

If the User selects the “Save and use” button, then a “Headphone Name” pop-up screen will display. Once the optimization profile is named it will be stored and displayed as the active optimization profile in the “Headphone Select” screen.

Headphone Select screen, Fig. 15

- 5 The “Headphone Select” screen displays the set of saved Headphone Profiles. The currently active Headphone Profile is indicated by a check on the list of saved profiles. The User can activate another Headphone Select by selecting a name on the list. The check mark will move to that entry indicating that that profile is now the active Headphone Select.

Filter Screens

- 10 These screens provide parameters and Noise Profiles that are utilized by the DSP for noise control.

“Filter” screen, Fig. 16

- The vertical display bar on the “Filter” screen has a vertical display slider which visually shows the presence of audio input through a colored column. Users can use the sliders on the vertical
- 15 bar to reduce noise. The upper slider indicates a gain level that is used by the DSP to recognized sharp, sudden sound that should not be amplified. The lower slide represents the gain level for low frequency audio, i.e., out of the speech range, that should not be damped. The reason these controls are on the slider is to give a User a visual clue on the appropriate settings by seeing visualization of the audio being processed.
- 20 The DSP processor has a capability to continuously estimate what is noise in the audio input. However the algorithm works better with a static Noise Profile as long as that profile reflects noise in a stable environment, e.g., the air conditioner noise in an otherwise quiet room in which the User is participating in a meeting, the fairly constant noise produced in a traveling car, and, somewhat ironically, in a very quiet environment so the DSP algorithm does not guess wrong
- 25 about what is noise and what is speech.

If the user selects the “Sample Noise” button the “Sampling pop-up screen is displayed and the Noise estimating process is initiated.

If the User selects the “Customize” button the “Advanced Filter” screen is displayed where some advanced noise control features are available, and where a saved Noise Profile can be activated.

5 **“Sampling” screen, Fig. 17**

The sampling process takes about 5 seconds. It's best to sample when people are not speaking (since you will probably NOT want to filter or eliminate speech), so the User may often ask for a moment of silence. This screen displays the duration of the sampling process. When the sampling process is complete the “Save noise sample?” pop-up screen is displayed.

10 **“Save noise sample?” screen, Fig. 18**

If the User selects the “Just use” button then the computed Sample Noise Profile is the active profile for the current session.

If the User selects the “Save and use” button then a pop-up screen will require the User to name the Sample Noise Profile. Once the Sample Noise Profile is named it will be stored and displayed
15 as the active Noise Profile in the “Advanced Filter” screen.

“Advanced Filter” screen, Fig. 19

The slider on the horizontal bar of the “Advanced Filter” screen allows the user to fine tune the DSP process. This is primarily by affecting the timing of the transition once the DSP process decides that targeted speech has begun or that it has ended. In noisy environments the slider
20 should be moved to the right towards the label; “Reduce Noise”. With this setting the DSP will quickly reduce noise but in the process may clip the beginning of speech. In a quiet environment the slider should be moved to the right towards the label “Optimize speech”. With this setting the DSP will more slowly reduce noise but will avoid clipping any Speech sounds.

The “Select a Noise Filter to Use” section of the screen lists the stored Sample Noise Profiles, with the active Sample Noise Profile indicated by a check-mark. The User can select a different Sample Noise Profile from the list, which is then activated. The “Continuous adaption” profile is always available and is the default, if the user has not created or activated a stored Sample Noise Profile.

If the “Continuous adaption” is active then a parameter is sent to the DSP to do continuous noise estimation.

Alternate Embodiments

Optional Speech Detection

In an alternate embodiment, the aggregate latency may be reduced by forgoing the need to analyze the audio input in the frequency domain (e.g., by performing a Fast Fourier Transform). Thus, in alternate embodiments, DSP

In an alternate embodiment, both time and frequency domain Voice Activity Detection may be utilized by processing the audio input in a separate thread that identifies Speech in the time-domain. Processing in this thread may include dividing the audio signal, at regular intervals reflecting the acceptable latency, into two frames — a small frame and large frame. The energy parameter (E) is calculated by frequency from the small frame and the calculated energy (E) is used to detect a start-point and endpoint of audio that is identified as speech. It is initially identified in the speech mode where a pitch period (P) is detected and measured from the large frame, and the pitch detection is used to determine whether there is voiced speech to validate that the audio is speech and may be identified as speech mode. The start and end of speech, as detected in this thread, would be sent as an argument to the DSP process.

This disclosed embodiment may utilize a unique two-step method to detect speech sounds. The embodiment works in the constrained environment of the operating system of a commercial mobile device. Given a sound input source, the embodiment detects speech in that sound in real time. The embodiment, used in conjunction with a mobile device, can assist improved speech

intelligibility for a listener. Once the speech is detected the speech can be amplified and non speech sound can be reduced or suppressed. An aspect of the embodiment is designed to detect speech in a short time. This is required if the latency between the processed speech and speech sound arriving directly to the listener's ear is too long, the listener's brain will not integrate the two sounds and speech clarity may be lost in the confusion of sound echoing.

In alternate embodiments, speech endpoints are detected in real time utilizing the computing power of an appropriate mobile device. The technique addresses a major constraint for detecting speech on such devices. One of the important requirements for hearing enhancement is that the time delay caused by processing the speech may be very short. The short period is defined as the time the majority of listeners would not hear the delay between the speech being processed and the unprocessed ambient speech directly reaching a listener's ear. Listeners would not hear the delayed sound because as long as the latency is very short, the listener's brain will integrate the two sounds. If the latency caused by the processing is longer, the latency would be noticeable and listening to the processed speech sound would be annoying or confusing.

The embodiment describes a method that detects speech so that the latency of processing speech on a mobile device, including the built-in latency for the required processing of the device's operating system to input and output the sound, is very short. In digital signal processing, speech detection (or voice activity detection (VAD)) has been widely used in applications of speech recognition and wireless phone communication. Speech detection identifies the starting and ending points of speech versus the ambient noise. Speech detection is typically based on changes in short time sound energy, some algorithms use additional parameter such crossing rate (number of times the signal has crossed the zero value) for assistance. This mechanism works because when someone talks, they may talk louder than the background noise to be heard. This increase in sound energy can then be interpreted as speech. The embodiment first assumes a certain ambient noise level derived either from the beginning of the input signal or from manual training, and establishes a speech threshold a few dB above

the noise level. It then continuously measures the input short time (10 20ms frames of data) energy. When the input short time energy exceeds the speech threshold for a period of time (N), it decides that the speech has started. When the input signal is in speech and the short time energy drops below a threshold set close to the background noise level for a period of time (M),
5 it decides that the speech has ended. To avoid false trigger of speech by short duration loud noise, the time period (N) for speech trigger may range from 50ms to 200ms. Once the speech start is detected, the system back tracks the input signal by the time period (N) to mark it as the real starting point of speech. The time period (N), therefore, is the delay of speech detection.

Speech recognition systems utilize methods with delays of up to about 200ms, as these systems
10 are not providing real-time hearing assistance. For wireless communication, (N) can be as short as 50ms. However, in these systems if (N) is too small, many short duration loud noises such as a tap on the table would trigger speech detection. Another problem with current speech detection systems when related to hearing assistance is the recognition that the ambient noise level has increased, such as when a person has just walked into a noisy restaurant. The
15 increased sound energy would cause the higher level of ambient noise to be detected as speech. Current speech detection systems utilize some mechanism, such as automatic reset after a long period of continuous speech (i.e., tens of seconds) or by a manual user reset, to readjust the ambient noise level. Thus, current speech detection methods, with a speech detection latency of 50ms to 200ms), and slow adaptation to ambient noise level, cannot be effectively utilized in
20 hearing enhancement applications. This embodiment proposes a two-step method of speech detection to overcome the weakness as mentioned above. Once speech is detected, that speech can be amplified and background noise suppressed or reduced.

This two-step method is used for detecting speech with very little delay and adapting to ambient noise quickly. In the following, this method is described with specific parameters and
25 means. However the same idea can be applied with different parameters and means.

Input signal is divided into two sequences of frames with frame size of 20ms and 40ms, respectively. Both sequences have the same frame interval of 10ms, that is, for every 10ms of input signal, a pair of frames, one with frame size 20ms and one with frame size 40ms are obtained. Therefore, the decision made based on a pair of frames (small and large) has an
5 inherent delay of 10ms.

Two parameters are calculated from the pair of frames, a total energy (E) is calculated from the small frame, and a pitch period (P) is detected and measured from the large frame. Energy calculation and pitch measurement are well known prior art that can be found in many digital signal processing textbooks and publications. When the input signal volume increases, either
10 from noise or from speech, the energy (E) value would increase. For human speech, vowels or voiced speech contain pitches that are caused by vibration of the vocal cord and display a periodic pattern. Human voice pitch frequencies range from 100Hz to 400Hz, which translate to pitch period of 10ms to 2.5ms. Since background noise rarely presents such periodic pitch pattern, detection of voiced speech or pitches is a reliable indication of speech, even in a noisy
15 environment. However, not all speech is voiced. Most consonants such as "f's" are unvoiced that don't have pitches and are difficult to distinguish from noise. Fortunately, almost every word contains voiced speech, and the beginning consonant is short, typically 20 100ms long, and the transitional period from consonant to vowel typically shows some pitch pattern as well. A large frame of 40ms contains multiple pitch cycles and can result in more reliable pitch detection and
20 measurement.

The two-step method uses the energy to detect endpoints of speech, and the pitch detection to determine whether there is voiced speech. The energy-based speech detection responds quickly to speech, in 10ms as determined by the frame interval. Such short delay is critical for hearing enhancement applications. However, it can be easily triggered by increased noise as
25 well. The pitch-based voiced speech detection distinguishes real speech from increased noise, but it takes longer duration (a few dozen milliseconds to a few seconds) to make a decision. If no voiced speech is detected after speech trigger, the detected speech is cut short and the

speech detection threshold is updated to the increased noise level. The effect of such a two-step approach is that when non speech background noise increases, such as the approaching of a car, wind, start of car engine or music, one may hear the noise for a short duration (i.e., 12 seconds) and then it may be suppressed and the energy-based speech detection threshold may be adapted promptly.

The speech detection algorithm has two modes, noise mode where input signal is assumed as noise, and speech mode where input signal is assumed as speech. An input frame is labeled as "noise" in noise mode, and "speech" in speech mode, until the detection mode switches from one to another. When speech is detected, it switches from noise mode to speech mode# when speech ends or cut short, it switches from speech mode to noise mode. The algorithm starts with noise mode. The following outlines the speech detection algorithm:

1. For every 10ms of input signal, an energy (E) is calculated from the small frame.
2. In noise mode, if (E) is above a speech detection threshold (T), detection enters speech mode and the current frame is labeled as speech# otherwise, update the overall noise level in sequence of previous "noise" frames, adapt the speech detection threshold (T) to the new noise level.
3. In speech mode, for every 10ms of input signal, a pitch measurement is calculated from the large frame. If pitch is detected and the pitch period is between 2.5ms and 10ms, the frame is labeled as 'voiced'. For a predetermined duration (M), typically between 100ms-5 seconds, if the number of "voiced" frames exceeds a threshold (L), it is determined that there is real voiced speech in the current speech mode# otherwise, there is no voiced speech and the speed mode is invalid, and:
 - a. the current frame is labeled as noise and detection mode switches to noise,
 - b. if no voiced speech has ever been detected in the current speech mode, update the overall noise level in sequence of previous frames including those labeled as "speech"

in the same speech mode, and adapt the speech detection threshold (T) to the new noise level.

4. In speech mode, if energy (E) is below a “none speech” threshold (T_n) continuously for a certain time period (Q) (typically 200ms to 4 seconds), it is determined that speech has ended and the detection mode switches to noise mode.

In another configuration, the voiced speech detection based on pitch measurement can be running continuously also in noise mode to reliably obtain a noise reference model. When voiced speech is detected, the frame is labeled as speech and the detection enters speech *mode*, and the speech detection threshold (T) is further lowered to reflect low signal to noise ratio.

This configuration would work better in a very low signal to noise ratio environment where energy level alone has difficulty distinguishing between noise and speech.

As described above, the energy-based speech detection depends on a threshold (T), which is set based on the noise energy. Therefore, the robustness of the detection depends on the reliability of obtaining a noise reference. Pitch detection can be used to reliably obtain a noise reference in the noise mode by detecting a period of sound at least one or a few seconds long where no pitch is detected, denoting this period of sound as unvoiced sound. By discarding the beginning and ending parts (e.g., a few hundredths milliseconds each) of this unvoiced sound, the center part of the unvoiced sound can reliably serve as noise reference. Since every word contains a voiced vowel, while an unvoiced consonant usually does not last longer than a few hundredths milliseconds and can only occur at the beginning or ending part of the unvoiced sound — possibly passing over from a previous word or the beginning of a following word — the center part of the unvoiced sound contains neither voiced vowel or unvoiced consonant. Such noise reference can be periodically updated to reflect the changing environmental noise.

In order to further improve detection in soft speech, which energy may be very close to the background noise, a filter bank can be used to obtain a set of energy values across a frequency spectrum for speech detection instead of the total energy. A filter bank is an array of band pass

filters covering the voice spectrum, such as from 100Hz to 5000Hz, with each band pass filter covering a different frequency sub band. A soft speech, typically an unvoiced consonant, has higher energy in one or more sub bands even when its total energy may be very close to the background noise. For example, the consonant “f” or “s” has higher energy in frequency sub band of 2000Hz and above. A filter bank output therefore can be used to detect speech in each frequency sub band, which is more sensitive than the total energy. In steps 2 and 3 of the above algorithm outline wherein the speech detection threshold (T) (or an array of energy from a filter bank) is adapted to the new noise level, the adaptation may use different speeds depending on whether the noise level is increasing or decreasing and on the distance of energy level of previous detected voiced speech from the noise level. Faster adaptation to lower noise level makes it more likely to detect soft speech in rapidly changing ambient noise. And if the distance of energy level of detected, voiced speech from the noise level is small (an indication of low signal to noise level), speech detection threshold (T) may be set lower to more easily detect soft speech.

Additional Hearing Profile and Equalizer Settings

In an alternate embodiment, a number of additional techniques would be implemented to create or modify a Hearing Profile:

- A facility would be offered that allows a User to take a “standard” hearing test and create and store a resulting audiogram. The hearing test would be implemented by having the User recognizing whether they can hear a sound of a certain frequency and depreciating gain on frequency until it cannot be heard. Given that the hearing test would utilize the same earpiece and speaker system that the User will use for Hearing assistance, the resulting audiogram can be more useable than an audiogram resulting from an externally administered hearing test. Also the hearing test could be performed in controlled but different auditory settings, potentially providing more accurate audiogram variants.

- A speech intelligibility test would be offered to more precisely deal with a particular User's audibility. The intelligibility test would be accomplished by playing words at various levels of sound and noise. The result of the intelligibility test, for example, an inability to distinguish certain consonants, would be provided to an enhanced DSP that would be able to process the information and moderate that User's intelligibility issues.

In an alternate embodiment, a number of additional techniques would be implemented to create or modify an Equalizer Profile:

- A facility would be offered for a User to create paired equalizer setting for left and right ears. This would be especially useful for those where there is a marked difference in audibility between the left and right ear.
- An advanced facility would be offered which would automatically utilize different Equalizer Profiles based on an analysis of the audio input being processed. For example, different Equalizer Profiles would be selected as a User went from a quiet to a noisy environment or switched from listening to music to listening to targeted Speech. A UI would be provided to the User to associate Equalizer Profiles with an audio environment.
- Presenting a Profile-builder module that allows the User to create or edit various frequency-based profiles, to test the profiles based on stored exemplary speech and noise samples, to name and store the profiles.
- The speech intelligibility aspect of a hearing test could be accomplished by playing words at various levels of sound and noise. The processor could take information from the speech test to enhance and/or modify the basic hearing profile.

Features for Advanced Controls and User Interface

An alternative embodiment would have the following additional User Interface features and controls:

- The alternate embodiment would have controls to record and store any input audio or processed audio on the Mobile Computing Device's local storage or in the Cloud. The alternate embodiment would have controls to access the stored audio, so a User could re-hear the stored audio; controls to reprocess the stored audio, for example, to create or refine Profiles and re-sample noise; and controls to utilize restored audio in a hearing test.
- The alternate embodiment would have controls to set a preferred volume level. This would be implemented by allowing Users to select a volume level utilizing prerecorded sound. The embodiment would use the selected sound level to adjust for gain changes in the real-time audio input.
- The alternate embodiment would have a facility to be trained to recognize a keyword such that when a User utters that keyword the embodiment expects a following command phrase. The embodiment would provide a set of audio command phrases as an alternate User Interface.

Interaction with Other Applications

This section describes how an alternate embodiment would integrate with the other applications available on the Mobile Computing Device.

- In particular, the alternative embodiment would monitor ambient sound and when Speech is recognized, would reduce or mute the gain from the electronic audio that is produced by another application, allowing processed ambient sound to be heard by a User. The alternative embodiment would also have a UI control that explicitly switches between electronic audio and ambient sound processing.

- Given appropriate access to the electronic audio streams produced by other applications, including telephone conversations, the alternative embodiment would process the electronic audio in the same manner that ambient sound is processed, so that Users would get the benefits of hearing assistance for electronic audio.

- 5
- The alternate embodiment would have explicit mechanisms for other applications to provide audio input, allowing the other applications to take advantage of the “always-on” audio connection with a User. For example, Users could get appointment reminders whispered in their ear, and be connected to body-area health monitors where they could, for example receive and audio warning of unusually high blood pressure.

10 Utilizing the Internet and its Cloud

In an alternative embodiment, a history of a designated set of parameter settings, state variables, and Profiles are saved to storage on the Cloud, such that the history of use can be examined and that previous settings can be restored.

- Through the Cloud connectivity offered in the general purpose operating system of the Mobile Computing Device, control settings, Profiles, and at times audio input and output, are stored in the Cloud. Beyond the standard use of such data for backup and restore, this data would be used for accurate usage data. By saving all changes to settings, Profiles, speech-to-noise levels, duration of use, and even sample recorded sound to the Cloud, a continuous record of a User's hearing activity would be recorded. Applications could then be applied to that record to analyze a User's hearing situation and provide a warning if hearing is deteriorating inappropriately or if a medical condition is recognized. This record could be especially valuable to hearing professionals to help in understanding a deteriorating hearing situation.
 - In addition to utilizing the Internet to access Cloud storage, the embodiment could interface with other internet services. In particular, since the User would be using the
- 15
- 20
- 25

embodiment continuously for hearing assistance, there would be an “always on” audio connection to the Internet. This would be useful to push audio information to a User, specifically advertisements and marketing messages.

Wireless Transport for Low Latency Speech

5 Background

As previously discussed, minimizing latency in a hearing assist device is essential for user acceptance. This section describes how the signal processing chain can be extended to include a wireless link. The particular challenge is to convey digital signals over a Bluetooth link although many of the concepts would apply as well to other wireless links such as WiFi.

10 In addition to low latency, low power, and resilience to RF data loss are also required.

The Bluetooth radio link is comprised of various packets sent in “time-slots” where a time slot is 625 micro-seconds. There are two main types of packets – synchronous (SCO and eSCO) and asynchronous (ACL). The synchronous packets were designed to carry voice signals, whereas the asynchronous packets are designed to carry data. The SCO packets are real-time and
15 provide no recovery for lost packets. eSCO has a modest retransmit capability and ACL has a full ack type protocol to insure data reliability at the expense of uncertain delivery time.

Bluetooth profiles determine the type of packets that are used for each case. For wireless headsets two profiles are almost universally supported. One is called HFS (Hands Free) and the other is called A2DP. HFS uses SCO packets and sends data via the RFCOMM API, a serial port
20 emulation port that uses +AT commands to control call setup, select modes, etc. HFS supports bi-directional calls but only 64kbps data rates – mono and low fidelity. A2DP uses ACL packets and sends data via the GAVDP Interface. It is uni-directional and can support data rates up to 721 kbps.

Neither of these profiles are suitable for bi-directional transport of audio with a bandwidth of
25 up to 8KHz. Other implementations have bypassed the RFCOMM portion of the stack to get around the delay that it causes.

In order to minimize the delay of sending audio over the Bluetooth link, optimization is needed in several areas including:

1. The protocol layers. This document does not address this in detail, only to suggest that profile customization is needed to support the new mode and minimize delay.

5 Note: The innovation presented here may mimic the input for an A2DP profile, in which case a receiver that handles the A2DP profile may be useable. However, it may be that a proprietary profile needs to be defined, this would be third profile beyond the standard HFS and A2DFP protocols and may require specialized receivers.

2. The audio coder. This includes coding for error recovery at both the bit and packet level.

10 The coder also handles bit rate synchronization due to the difference in clock signals of the Bluetooth link and the sampling rate of the signal chain.

3. Optimizing data transfer from the signal processing chain to the input buffer of the radio link.

15 4. Optimizing the latency in signal processing chain when it converts from an oversampled FFT domain to a critically sampled voice coder. This is the focus of the innovation discussed here.

The presented invention is in the audio coding and how it interfaces to the signal processing chain. One aspect of the invention is the efficiencies that are possible by more closely integrating the output of the signal processing chain and the sub band filters that are used in many audio coders. SBC, the Bluetooth default coder for music is of this type, for example.

20

In a filter-bank system, the delay is determined by the input and output buffers which, in turn is dependent upon the number of sub-bands. Half of the delay comes on the input and other half comes on the final output when the data is sent one sample at a time to the DAC. One key aspect of this approach is to make sure that additional delay is introduced only by the radio link and to minimize the delay due to serialization for the radio link.

25

Note that the SBC codec is a subband based algorithm with block based ADPCM coding of the subband outputs. By making an entire buffer available — the output of the IFFT — the SBC has enough data to begin processing. The normal delay of waiting for a sufficient number of samples is bypassed.

- 5 Processing efficiency is possible by converting from the oversampled complex frequency domain of the FFT to the subband filters of many coder algorithms. The A2DP SBC codec is one option

Adaptive Delta Pulse Coded Modulation (ADPCM), when implemented with backward prediction is a 0ms delay codec. Early versions were implemented for compressing telephone
10 calls from 64 kbps to 32 kbps. To achieve greater bandwidth than the 3.2 khz bandwidth of the phone network, filter banks were developed to break the desired frequency range into smaller bands and then using ADPCM to code the output of each of the bands. Note that there is no requirement that the same number of bits are required to code each sub band.

As digital signal processing chips become more powerful, several main techniques were used to
15 improve quality and compression. They included: 1) more sub bands, 2) use of psycho-acoustic models both to better match the bands to the critical bands of human hearing and to use masking principles to hide the noise, and 3) more sophisticated quantization, and 4) bit coding to whiten the output bit stream. The MPEG codecs, of which MP3 is the most popular, are well known examples of this type of codec.

20 **Delay Analysis**

The signal processing chain does analysis using the overlap-add method and processes 256
samples into 256 frequency bins. With 75% overlap, 64 of the output bits are valid after every overlap add execution (aka one cycle through the signal chain). The delay from input of 256
samples to 64 bit output samples, at 44.1Khz is 5.8ms plus the processing time. The processing
25 time is under 0.1ms, so the total processing delay is less than 5.9ms.

The total delay includes the input and output delay of the device. An iPod Touch Gen 4 has 5.2ms of delay for 256 samples at 44.1KHz. This is in addition to the 256 bit delay for the processing. Thus the total delay for the microphone input to the earpiece output is $5.2 + 5.9 = 11.1$ ms. If we look at the wireless case, we subtract the output delay (assume it to be $\frac{1}{2}$ of 5.2ms) and then add conversion delay, wireless transport delay and earpiece output delay. This is shown graphically in Figure x below. We can design the earpiece output delay to be under 1ms.

One approach is to convert from the complex frequency domain to a real subband domain by converting from the complex frequency domain to a Modified Discrete Cosine Transform (MDCT). The MDCT uses a 50% overlap and produces half the number of outputs as inputs, effectively reducing the sample rate by $\frac{1}{2}$. In order to satisfy the 50% overlap, the first 32 bits will need 80 bits, or a second 64bit output block. This adds a delay of ~ 1.45 ms at 44.1kHz.

This delay is less than the estimated system output delay. If we assume that the earpiece delay plus the MDCT delay is equal to $\frac{1}{2}$ the system delay, the total wireless delay will be 11.1ms plus the wireless transport delay.

The next question is to figure out how many valid bits (or bins if in the frequency domain) are needed for the coder to produce an output. There are two main types of coders:

1. Coders based on subband filtering and followed by a quantization and coding. The delay and most of the calculations are due to the subband.
2. Coders based on linear prediction followed by quantization and coding. The delay comes from the linear prediction. Some prediction coders, such as ADPCM can have zero delay. Pairing these coders with the signal processing chain, yields a total delay of 5.9ms + delay of the coder. If ADPCM is the coder, for example, the delay is under 6ms.

Bluetooth Radio Link Processing

The radio link audio processing includes coding and error prevention and recovery. The table below shows the types of packets and their corresponding delay and bit rates that are illustrative of the Bluetooth packets type that could be used for this application.

Packet Type	FEC	CRC	No. bytes/TS	No. Time Slots 1 TS = 625 usec	Max Bit Rate	Delay ⁱ
HV3 (SCO_)	N	N	30	1 every 6 (3.75 ms)	64 kbps	7.84 ms
EV3 (eSCO)	N	Y	30	1 every 4 (2.5 ms)	96 kbps	12.1 ms
DM3 (ACL)	2/3	Y	121	3 every 12 (3 ms)	129.06 kbps	17 ms (est)
2DHI (ACL)	N	N	54	1 every 4 (2.5 ms)	172.8 kbps	17 ms (est)

- 5 HV3 is an SCO packet. HV3 packets are sent without options for re-transmit. EV3 is an eSCO packet. eSCO packets have a re-transmission request if the CRC indicates a problem. EV3 may be a good choice because 1) it has a re-transmit capability, 2) if we put in redundancy for packet loss, the delay would be 5 ms if we repeated each packet (note this would require compression to 48kbps).
- 10 ACL provides several options, some of which include FEC and CRC at the expense of bandwidth. Error rates may indicate another choice.

TS	1	2	3	4	5	6	7	8	9	10	11	12
HV3	To S1	Fr S1	To S2	Fr S2	To D1	Fr D1	To S1	Fr S1	To S2	Fr S2	To D2	Fr D2
EV3	To S1	Fr S1	To S2	Fr S2	To S1	Fr S1	To S2	Fr S2	To S1	Fr S1	To S2	Fr S2
DM3	To S1	To S1	To S1	Fr S1	Fr S1	Fr S1	To S2	To S2	To S2	Fr S2	Fr S2	Fr S2
2DHI	To S1	Fr S1	To S2	Fr S2	To S1	Fr S1	To S2	Fr S2	To S1	Fr S1	To S2	Fr S2

For each case, there is a different approach for handling two slaves, and a small amount of data.

In the case of HV3, the data will be sent in DM1 ACL packets. In the case of EV3 and DM3, the data bits will be packed with the voice bits, so the effective data rate will be somewhat less than

ideal rate. The data rate is expected to be low enough that the audio bit rate for EV3 will be over 90 kbps.

Codec Requirements

1. Audio Quality

5 The audio that will be sent over the wireless link from the signal processing chain is bandwidth limited to 8 kHz. The dynamic range is muted by the broadband AGC. (Note we may want to move the Broadband AGC.)

2. Compression Ratio / Bit Rates

10 Delay can be added into the system when buffers are serialized. This implies that fitting a frame of data into one or, at most two, packets is advantageous. Based on the data above, a bit rate of about 95,500 bps would yield the lowest delay (including overhead for error recovery/mitigation).

3. Low Latency

15 We want something with essentially zero delay. Under 2 ms or so including error resilience is the target.

4. Bit Error Resilience

20 Wireless communications links have particular levels of susceptibility with respect to increasing range, interference from other devices and the effects of multipath propagation. The audio codec has a role to play in terms of tolerance to bit errors and recovery from longer-term data loss. For high-quality digital wireless microphones the maximum allowable time for the audio decoder to re-synchronize to the data stream after longer-term data loss is of the order of 3ms. Notes: AMR uses information about the channel to determine bit rate. See also paper to use jitter buffer info to adjust the encoding and decoding, loss concealment, etc.

The Design

This innovation leverages the sub-band structure of the signal processing chain to produce a sub-band coder for the wireless link. In particular the complex frequency representation is converted into a real modified cosine transform sub-band representation. Then the sub-band
5 outputs are quantized and coded.

From Complex Frequency to Modified Cosine

There are two things that need to be achieved.

1. Convert from Real and Imaginary representation to Real representation. Conversion from the complex FFT to the Modified Cosine Transform is well known in the art.
- 10 2. Sample rate conversion to convert the over-sampled filterbank to a rate that matches the Bluetooth link. The bit rate = frame rate * number of bands * number of bits/band. For 8 KHz at 44.1k samples/sec and 256 bin frames, only 46 bands have data worth sending. The frame rate, if critically samples is $44.1\text{KHz}/256 = 172$. 25% overlap/add means the frame rate is 689 frames/sec. 95500 bps at 689 frames/sec and 46
15 bins/frame is just over 3 bits per bin.

Quantization

There are a couple of different quantization approaches to choose from.

1. G.726 standard ADPCM quantizer.
2. Improved ADPCM quantizer with enhanced prediction and possibly psyc acoustic
20 enhancement.
3. Range based quantization with energy and shape quantization.

The initial implementation will use standard ADPCM

Bit-level coding

This has to do with how the quantized data is combined for transmission. SBC combines several output vectors which add latency, so the key is not to combine successive frames.

Claims

1. A hearing assistance system comprising:

a mobile platform executing a mobile platform operating system with a programmable microprocessor; and

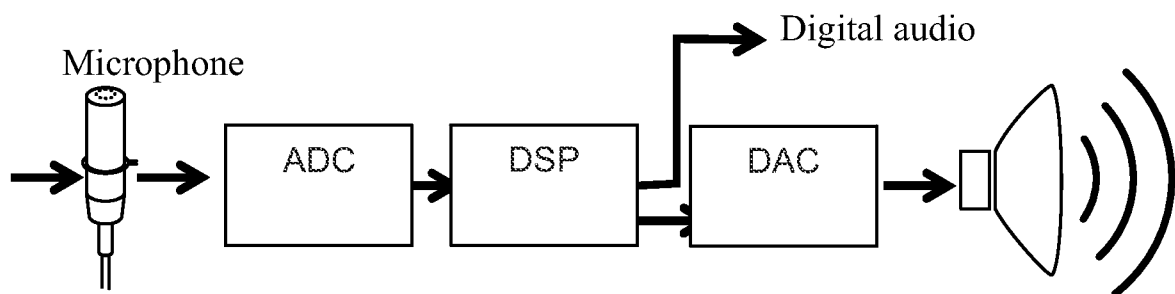
a hearing assistance software application having executable code for executing on the mobile platform using the mobile platform operating system and programmable microprocessor to process sound input received by the mobile platform to improve clarity of received speech.
2. The system of claim 1, wherein the mobile platform implements a Bluetooth-based transmission protocol for achieving transmission of the sound input, after processing thereof to improve the clarity of received speech, with a transmission latency of less than 20ms.
3. The system of claim 1, wherein the processing the received sound input is achieved with a processing latency of less than 20 ms.
4. The system of claim 1, wherein the mobile platform operating system is one of an IOS operating system, an Android operating system or a Windows operating system.
5. The system of claim 1, wherein the improving the clarity of received speech includes increasing a speech to noise ratio for the received sound input.
6. The system of claim 1, wherein the processing the received sound input includes pre-processing the received sound input to at least one of (i) amplify the received sound input or (ii) suppress consistent noise.
7. The system of claim 1, wherein the processing the received sound input includes utilizing digital signal processor (DSP) components of the microprocessor to process the received sound input.
8. The system of claim 7, wherein the processing the received sound input includes using DSP blocks for low latency signal processing.
9. The system of claim 1, wherein the processing the received sound input includes a primary processing in near-real time and a secondary processing of the received sound input to set or adjust one or more parameters applied during the primary processing.

10. The system of claim 9, wherein primary processing is achieved with a processing latency of less than 25ms.
11. The system of claim 9, wherein the secondary processing includes a dynamic adaptive control to automatically set or adjust the one or more parameters based on detected changes in a noise environment or in detected speech.
12. The system of claim 11, wherein the one or more parameters includes a frequency equalization profile for a targeted individual, group of individuals or sound type.
13. The system of claim 1, wherein the processing the received sound input includes applying speech detection to detect speech.
14. The system of claim 13, wherein the speech detection includes a combination of energy based speech detection and voiced speech detection.
15. The system of claim 14, wherein the energy based speech detection is used to determine start and end points of speech and the voiced speech detection is used to determine whether detected sound is voiced speech.
16. The system of claim 14, wherein the voiced speech detection is a pitch based voiced speech detection.
17. The system of claim 14, wherein the voiced speech detection is a spectral pattern based voiced speech detection.
18. The system of claim 14, wherein the energy based speech detection includes spectral band energy based speech detection.
19. The system of claim 14, wherein the voiced speech detection is used to detect non-voiced sound for use as a noise reference.
20. The system of claim 13, wherein the processing the received sound input further includes applying one or more algorithms to enhance speech or reduce noise when speech is detected.

21. The system of claim 13, wherein the processing the received sound input further includes applying noise reduction or noise suppression when speech is not detected.
22. The system of claim 1, wherein the processing the received sound input includes applying at least one of dynamic range compression, frequency based amplification or format boosting.
23. The system of claim 1, wherein the processing the received sound input includes applying one or more noise reduction algorithms.
24. The system of claim 1, wherein the processing the received sound input includes applying frequency equalization profile to improve targeted speech.
25. The system of claim 24, wherein the frequency equalization profile is selected by a user from a plurality of stored profiles.
26. The system of claim 25, wherein the frequency equalization profile is automatically selected from a plurality of stored profiles.
27. The system of claim 24, wherein the frequency equalization profile is a custom profile for a targeted individual, group of individuals or sound type.
28. The system of claim 27, wherein the frequency equalization profile is a custom profile for a targeted gender.
29. The system of claim 1, wherein the processing the received sound input includes applying a frequency transformation.
30. The system of claim 1, further comprising a speaker component in communication with the mobile platform, the speaker component including a speaker for outputting the processed sound input.
31. The system of claim 30, wherein the speaker component is in wireless communication with the mobile platform.
32. The system of claim 30, wherein the speaker component is an earpiece.

33. (The system of claim 1, wherein the processing the received sound input includes wireless transmission of the sound input after processing thereof to improve the clarity of received speech, wherein the processing the received sound input including the wireless transmission is achieved with a processing latency of less than 40 ms.
34. The system of claim 1, wherein the mobile platform operating system includes low level routines called by an operating environment including at least one of device drivers or firmware, wherein the processing the received sound input includes utilizing the low level routines to reduce latency.
35. The system of claim 1, wherein the improving speech clarity includes improving targeted speech clarity.
36. The system of claim 1, wherein the processing the received sound input includes suppression of noise in a selected range of frequencies and transformation of detected speech to the suppressed frequency range.
37. The system of claim 1, wherein the processing the received sound input includes selecting a sound input from a plurality of stored recordings and processing the selected sound input.
38. The system of claim 1, further comprising a database for storing a plurality of frequency equalization profiles based on targeted individuals, groups of individuals or sound types.
39. The system of claim 1, further comprising a database for storing a plurality of sets of one or more parameters for the processing the received sound input to improve the clarity of received speech.
40. The system of claim 1, wherein the processing the received sound input includes improving speech-to-noise ratio by at least one of lowering a gain of a non-speech range of frequencies and increasing a gain of speech range of frequencies.
41. The system of claim 1, wherein the processing the received sound input includes circumventing an operating system interface by including speech and sound processing directly using digital signal processor (DSP) blocks or low level routines thereby reducing latency.

1/22

Aggregate Delay – Wired Components**May be Less than 25ms****Fig. 1a****Aggregate Delay – Wireless Components****May be Less than 40ms****Fig. 1b****Fig. 2**

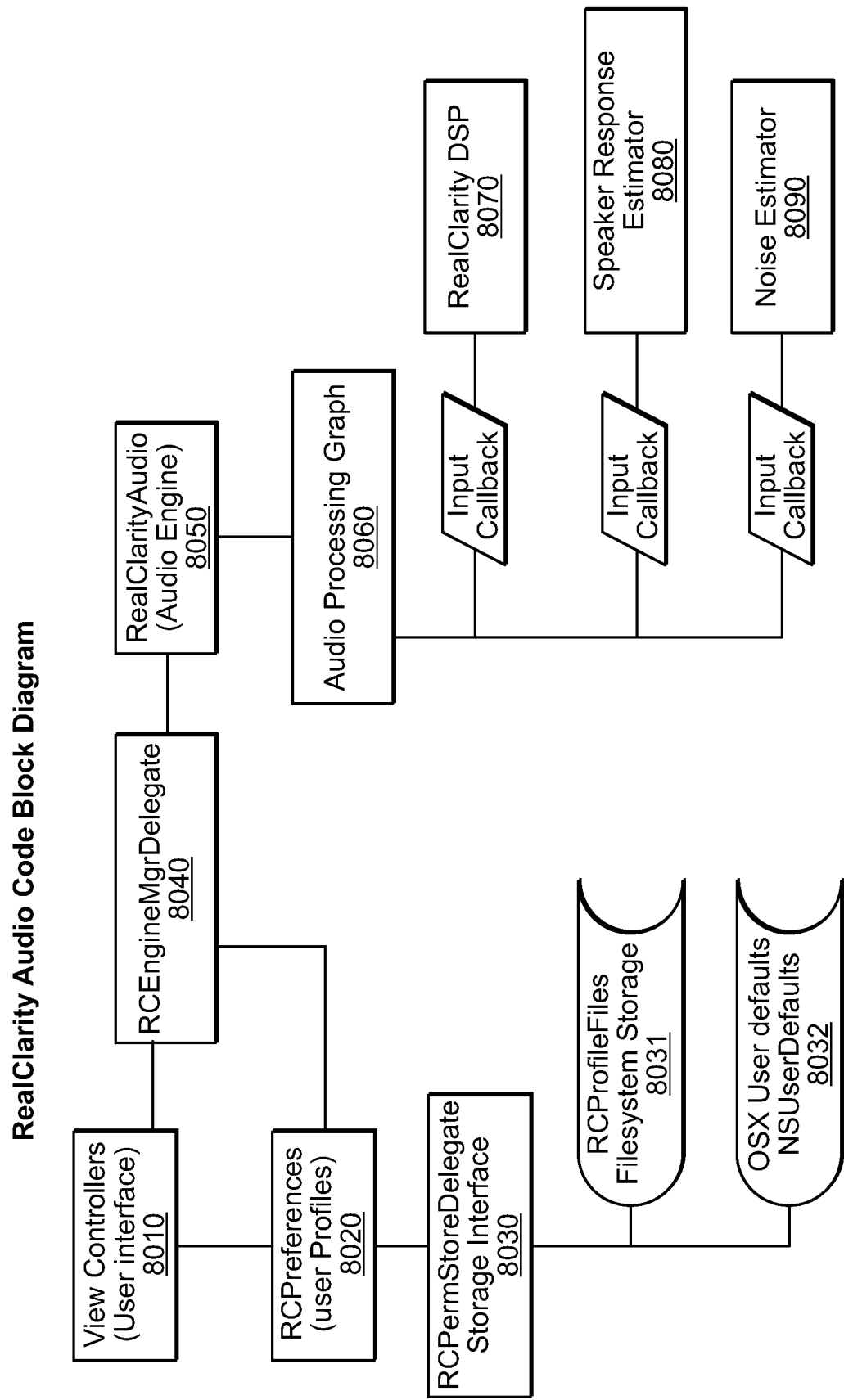
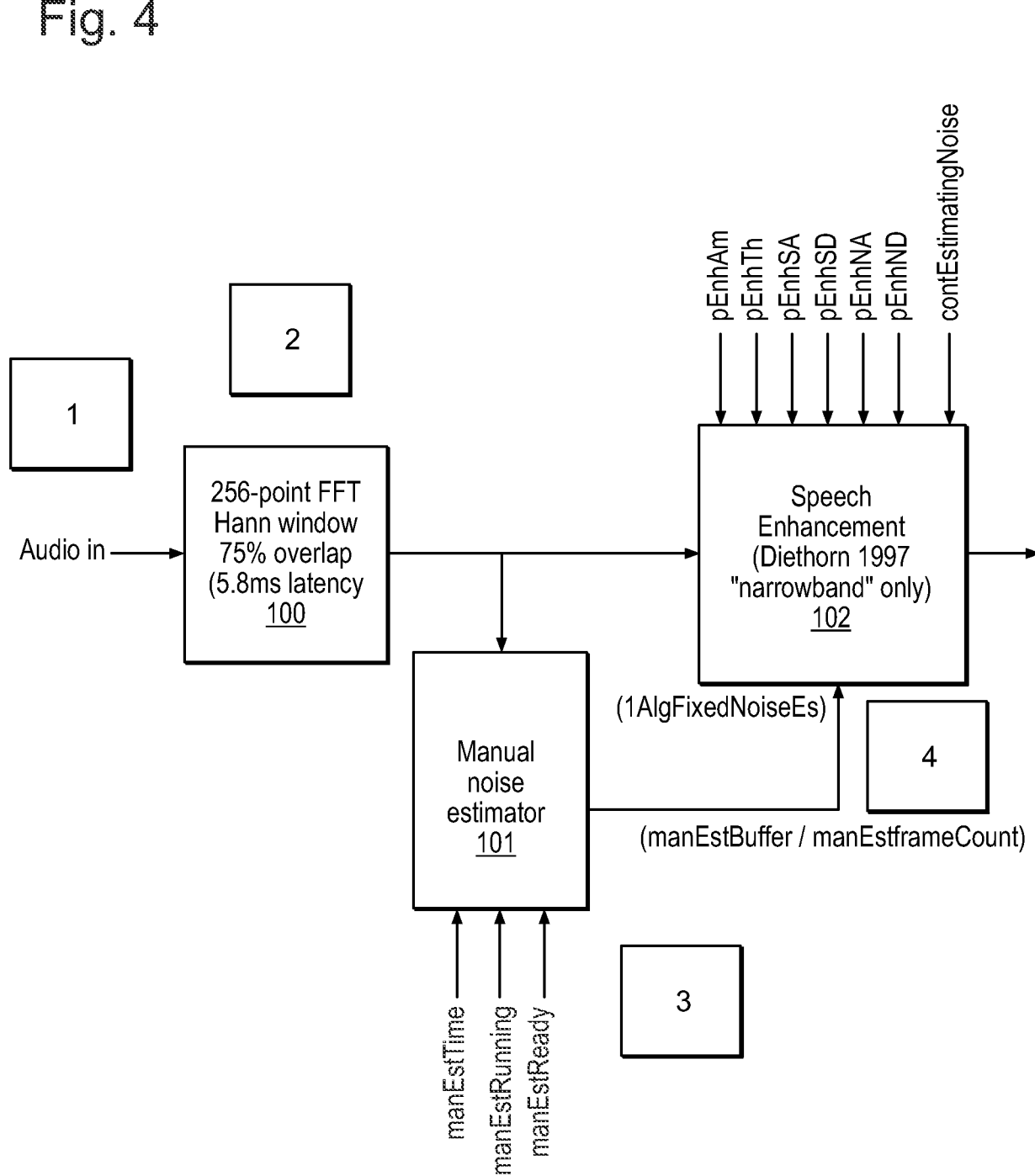


Fig. 3

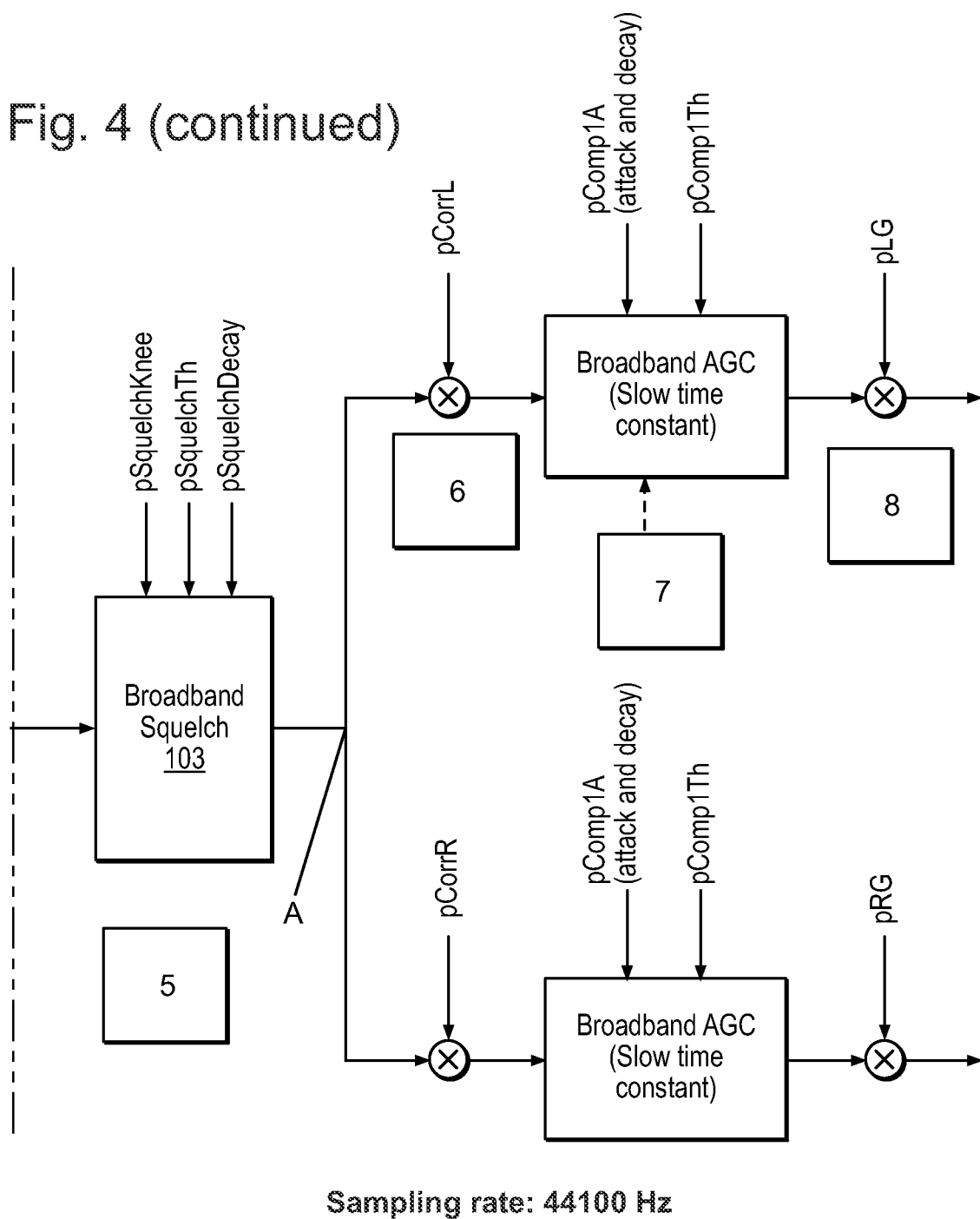
3/22

Fig. 4



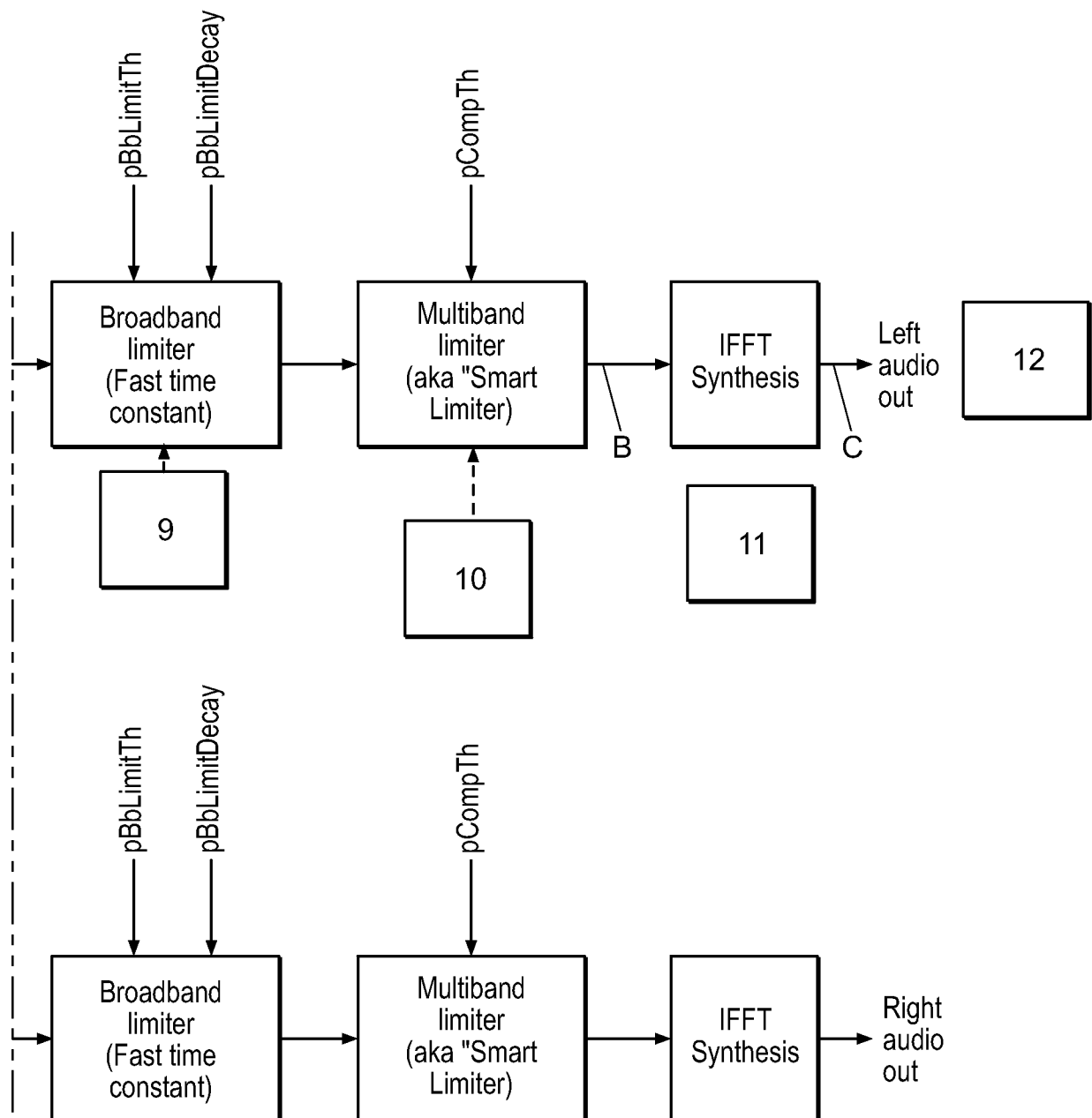
4/22

Fig. 4 (continued)



5/22

Fig. 4 (continued)



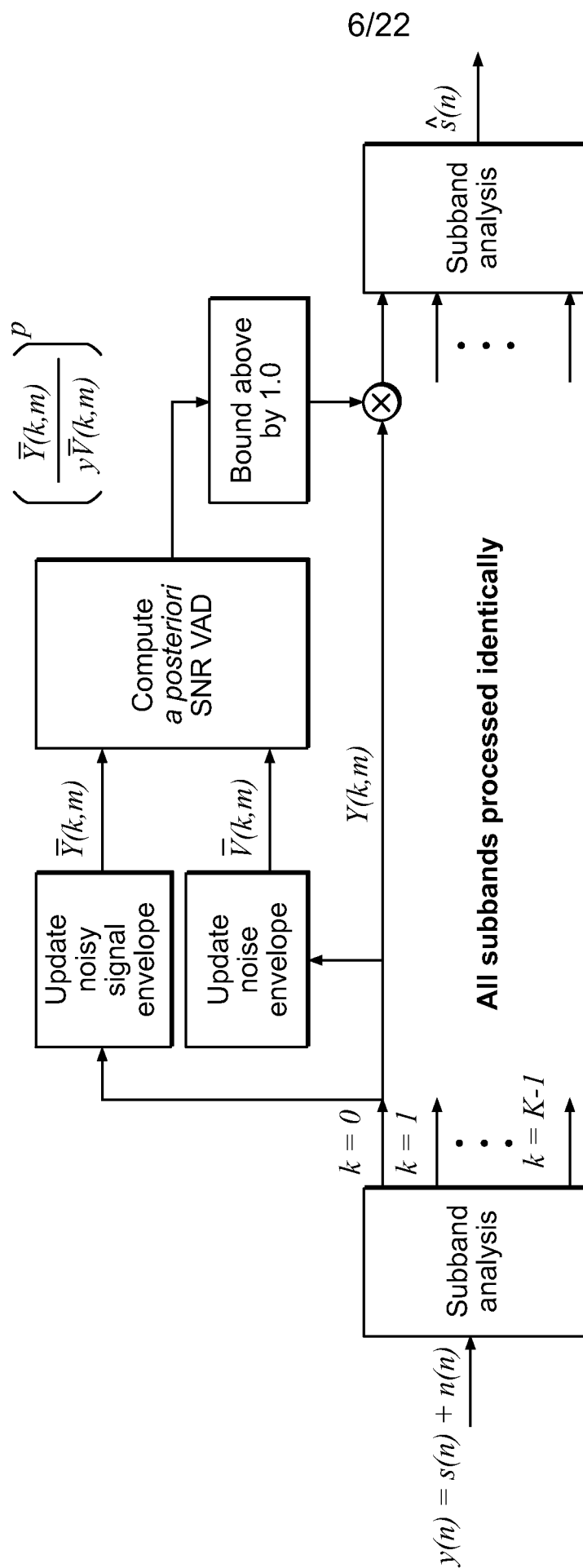


Fig. 5

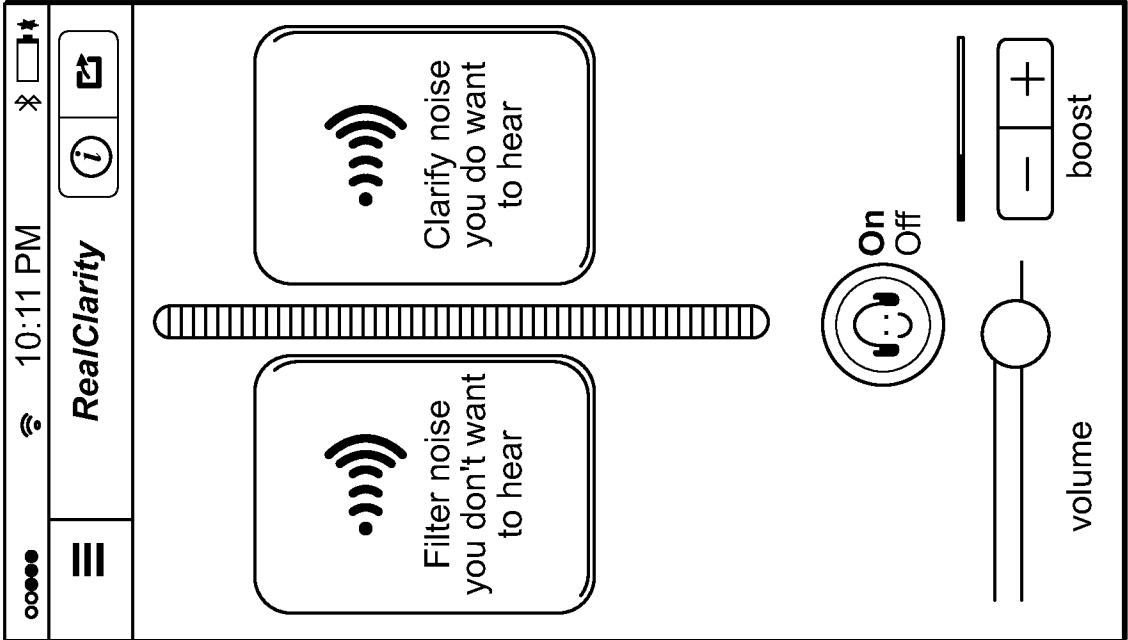


Fig. 6.2

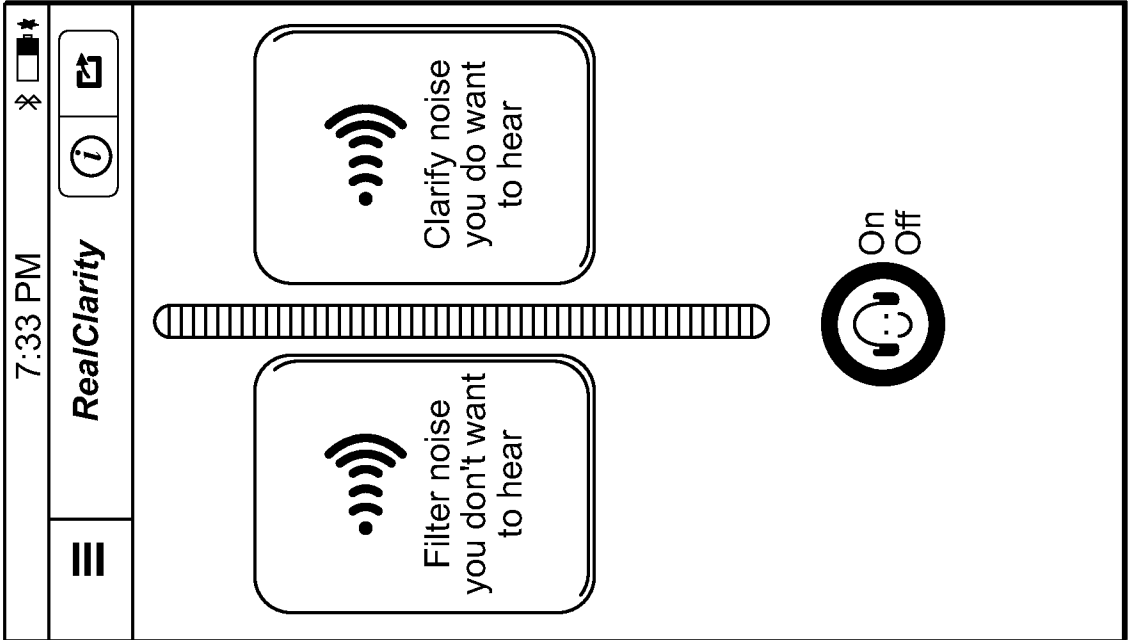


Fig. 6.1

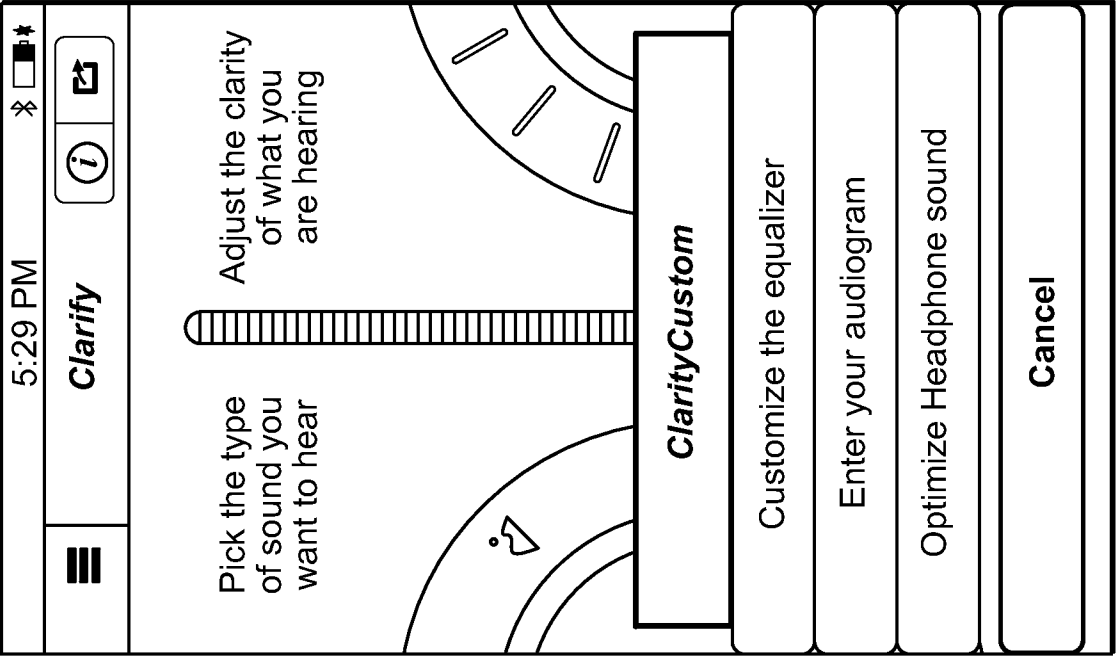


Fig. 6.4

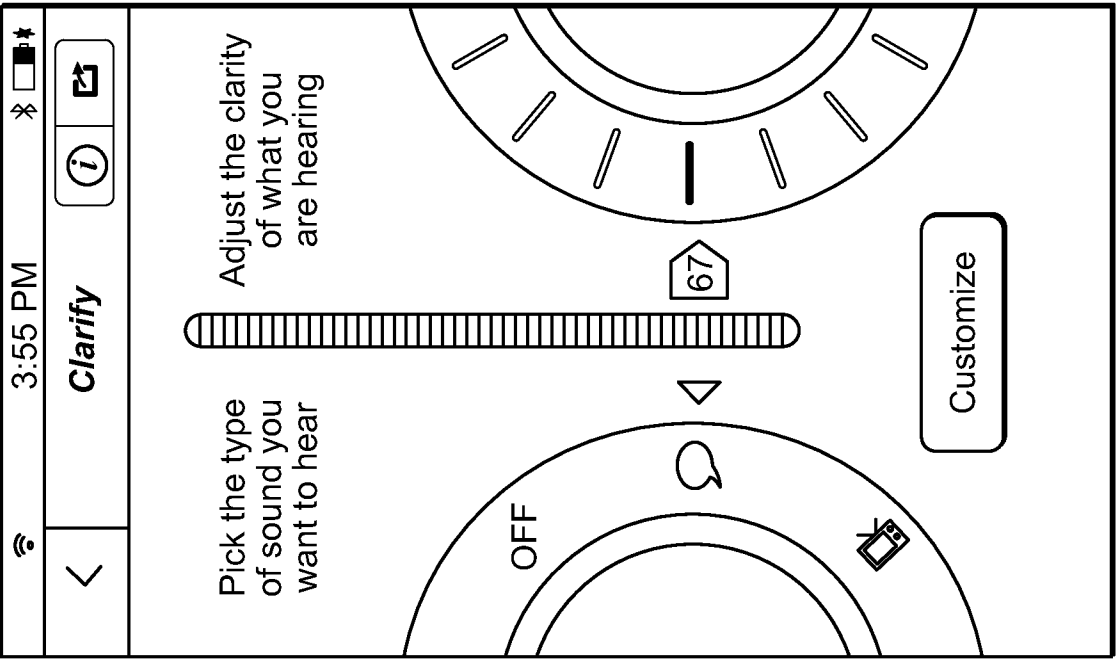


Fig. 6.3

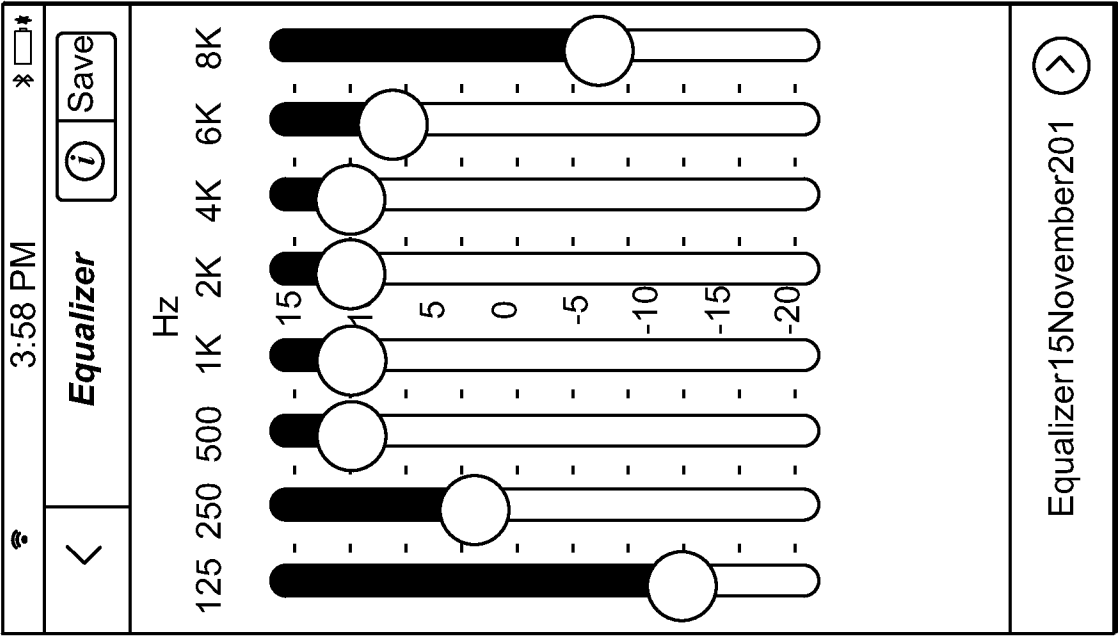


Fig. 6.5

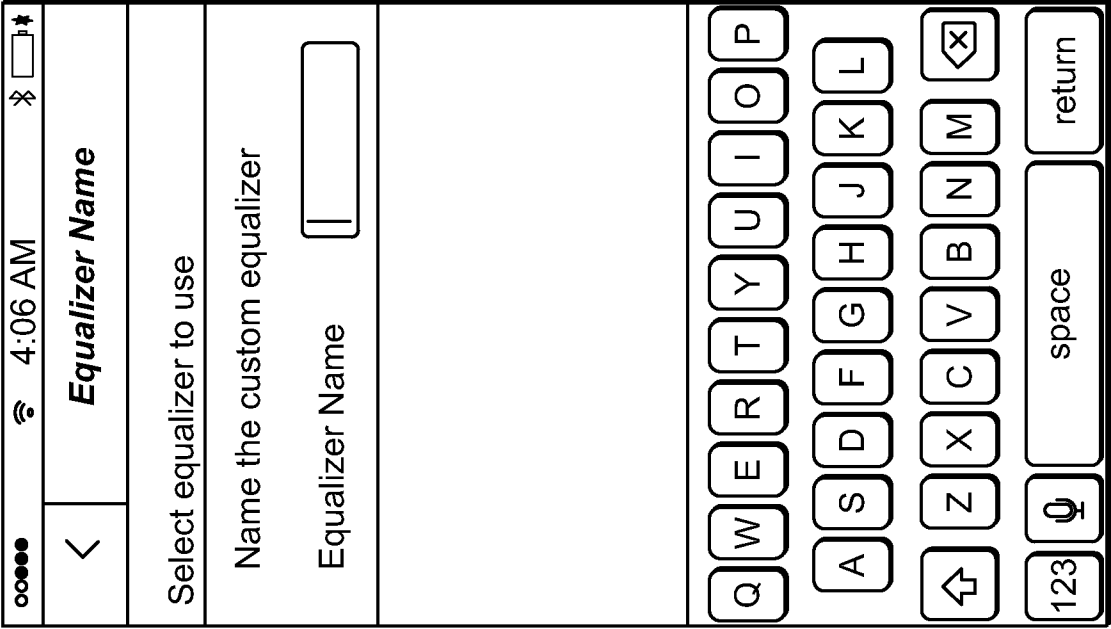


Fig. 6.6

10/22

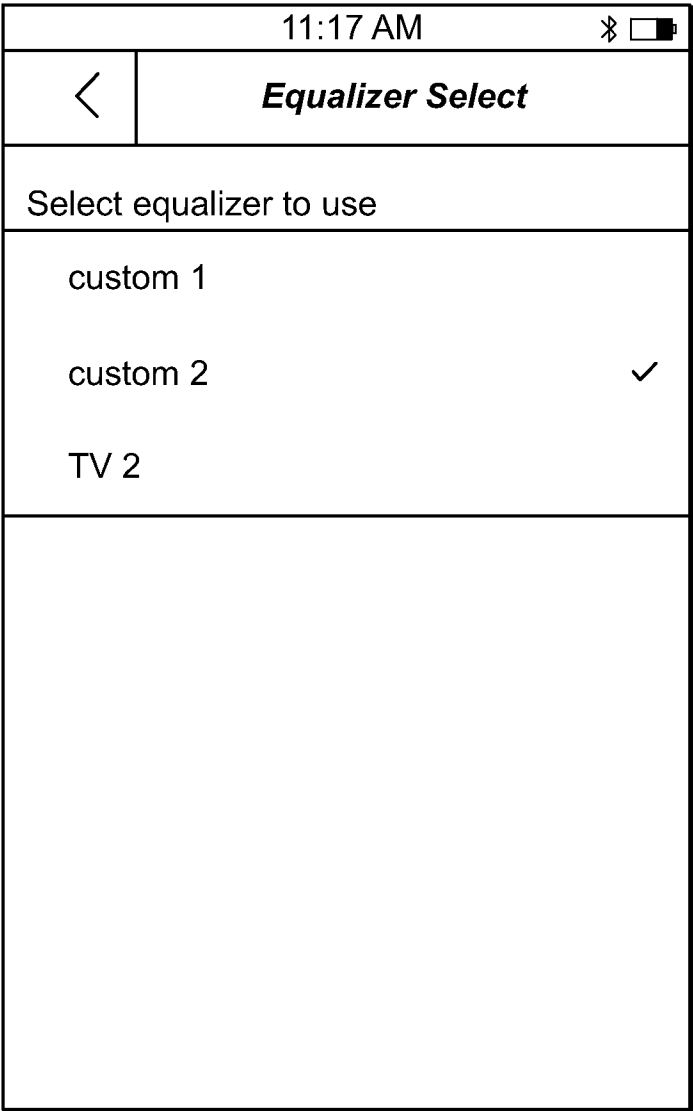


Fig. 6.7

11/22

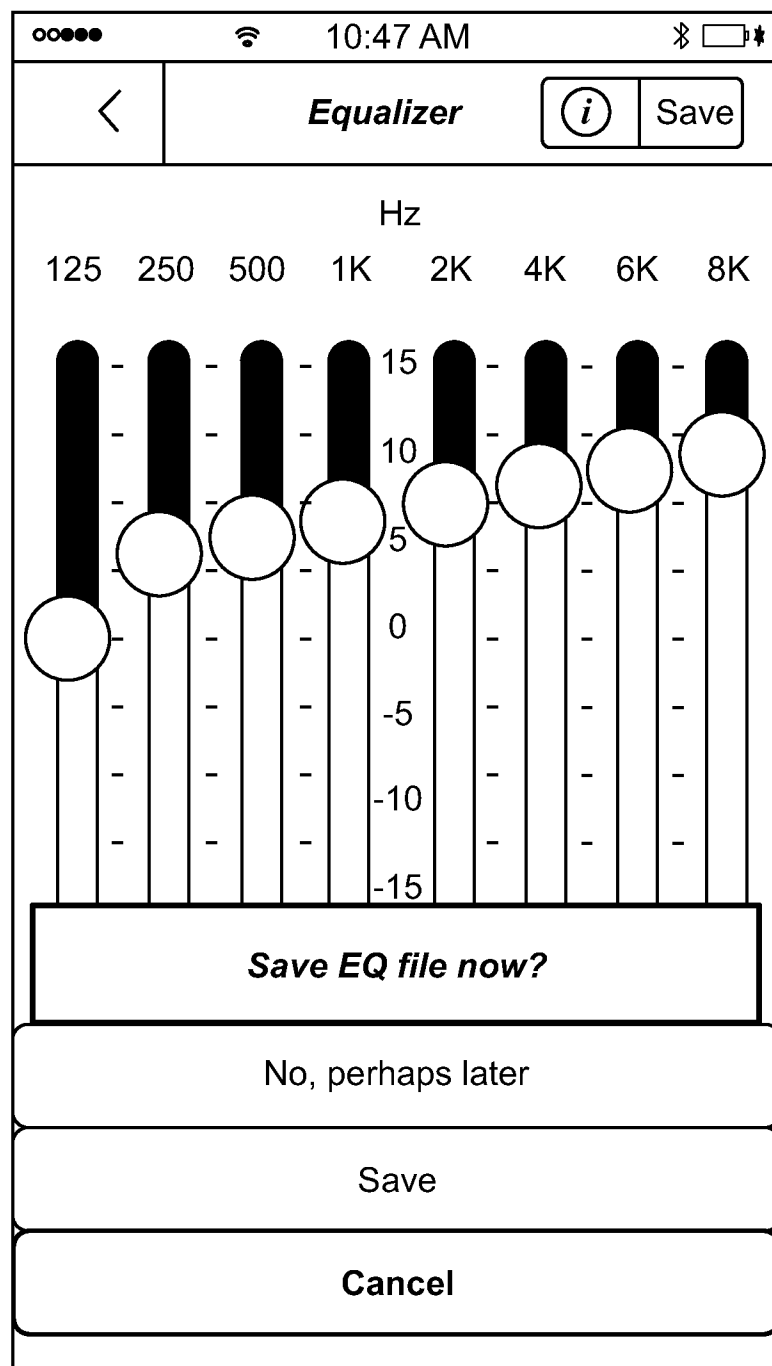


Fig. 6.8

12/22

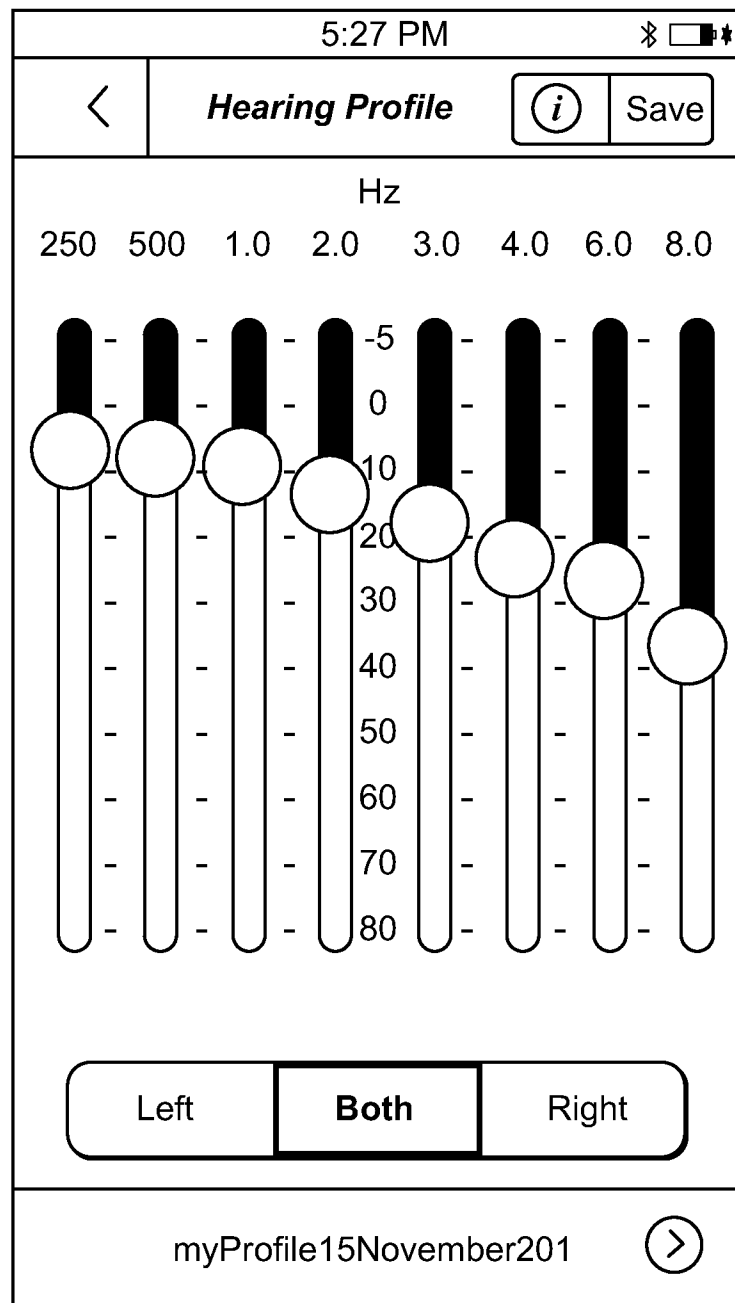


Fig. 6.9

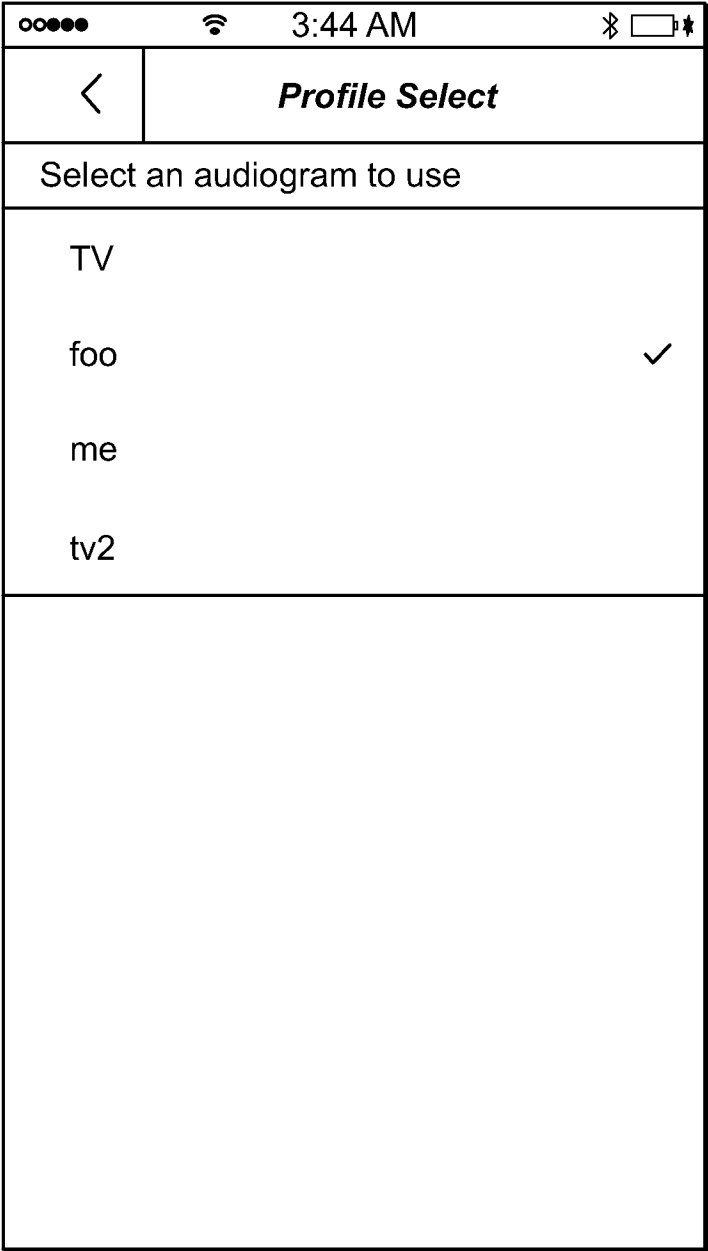


Fig. 6.10

14/22

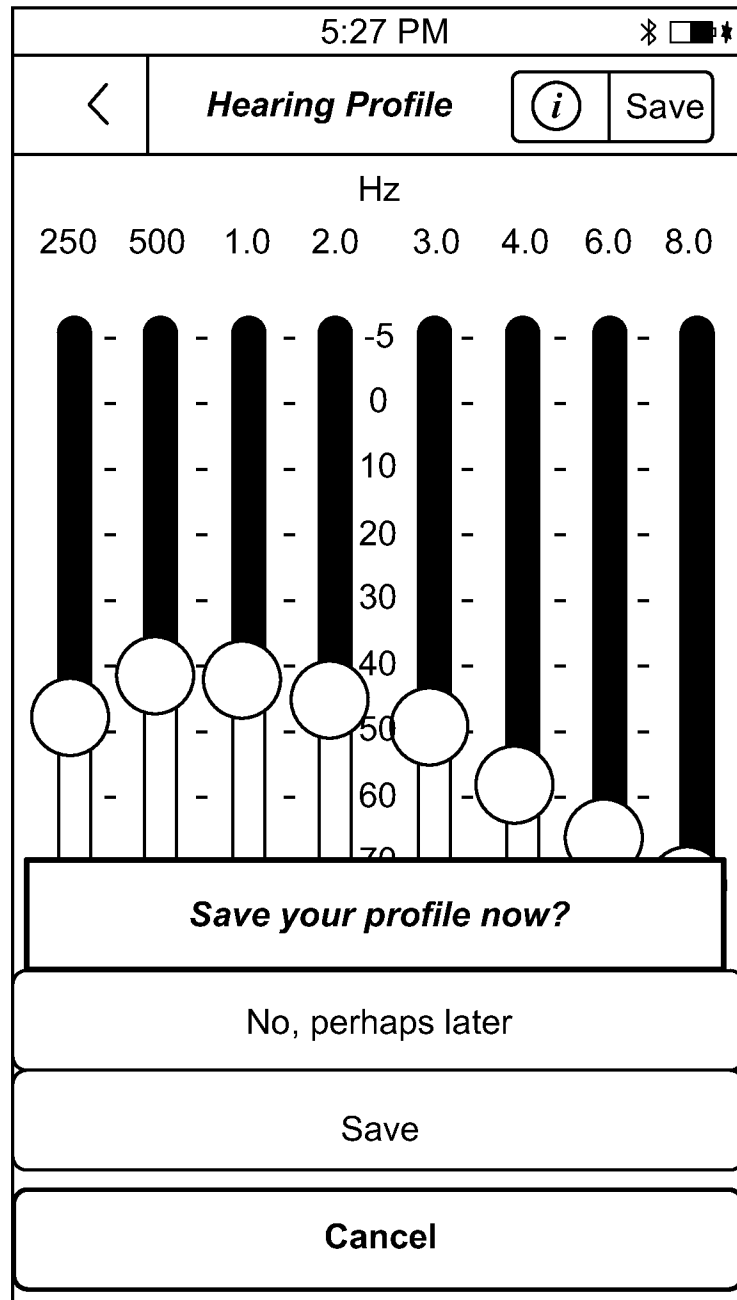


Fig. 6.11

15/22

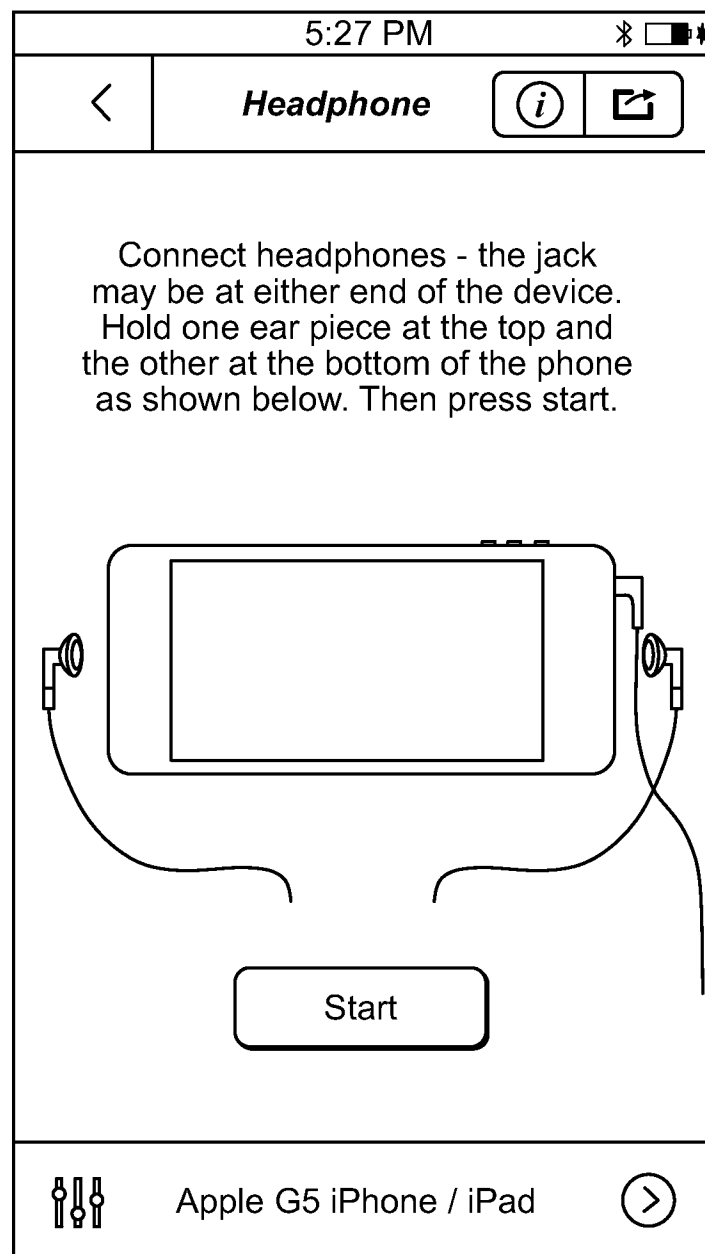


Fig. 6.12

16/22

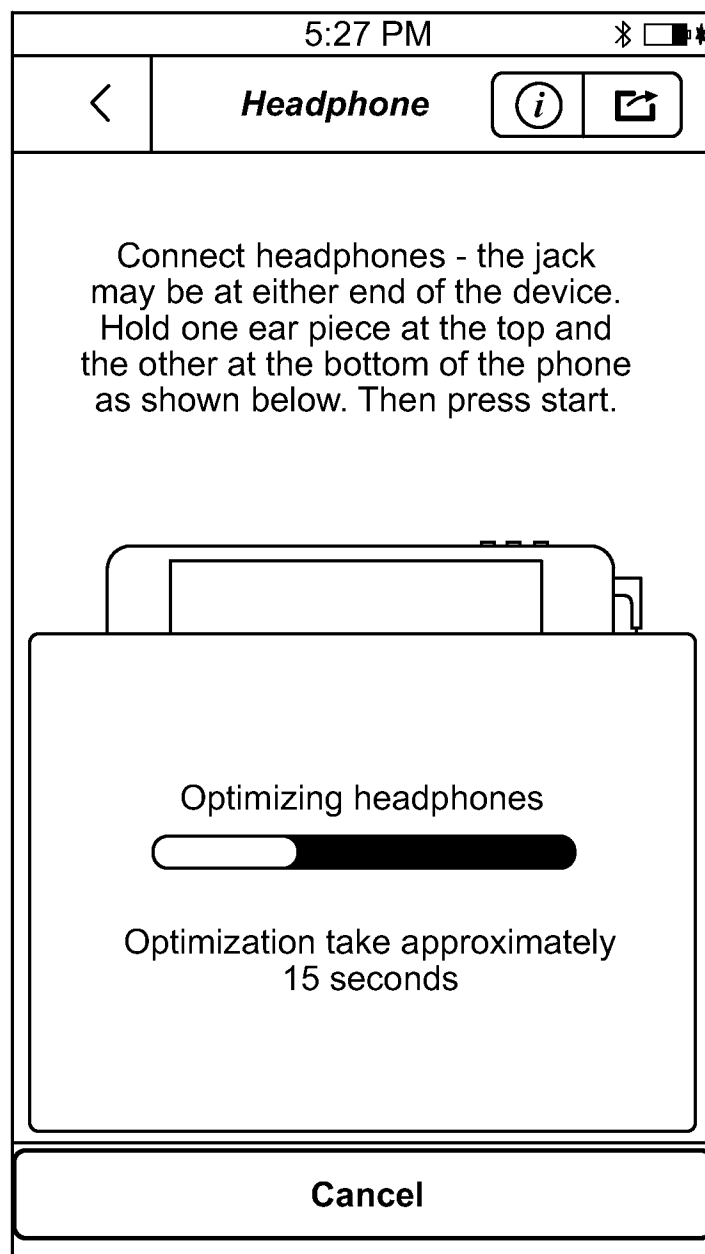


Fig. 6.13

17/22

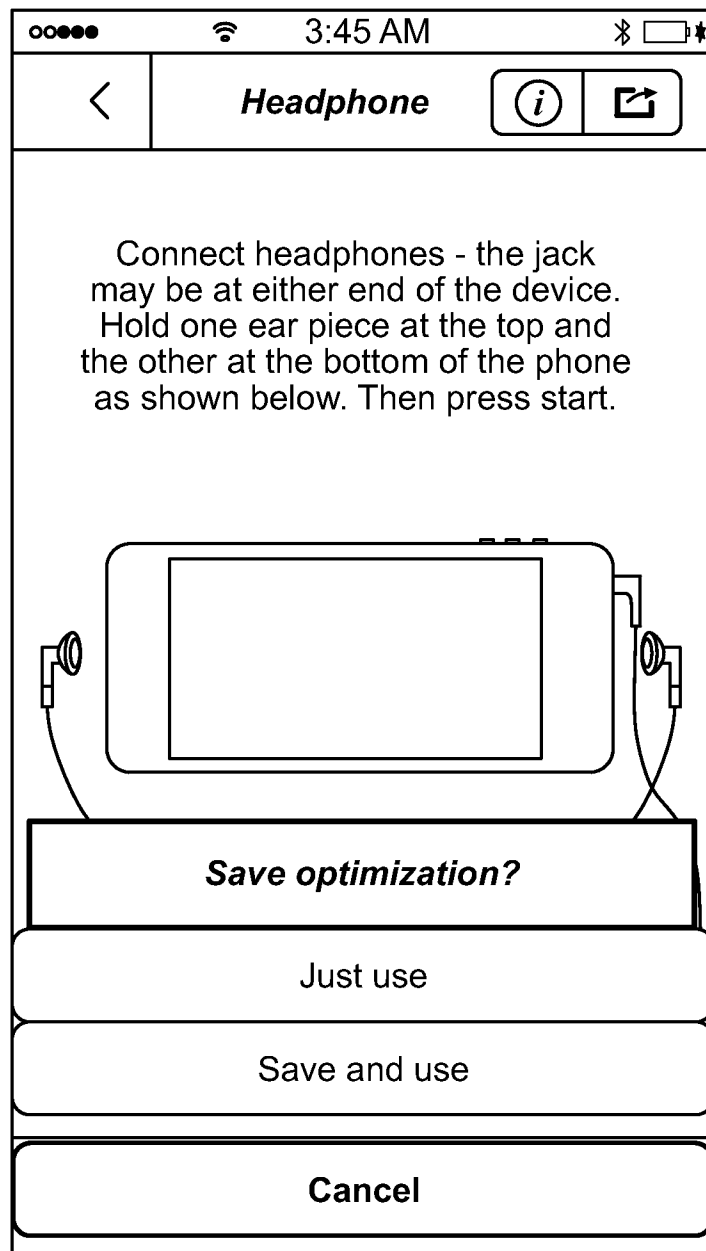


Fig. 6.14

18/22

A screenshot of an iPhone screen showing a "Headphone Select" dialog. The status bar at the top shows the time as 3:45 AM, signal strength, Wi-Fi, and battery levels. The dialog has a title bar with a back arrow on the left and the text "Headphone Select" in the center. Below the title bar is a subtitle "Select headphone to use". The main area contains a list of items: "Apple G4 iPhone / iPod Touch", "Apple G5 iPhone / iPod Touch", "Apple", and "foo". The "foo" item is selected, indicated by a checkmark on the right.

Fig. 6.15

19/22

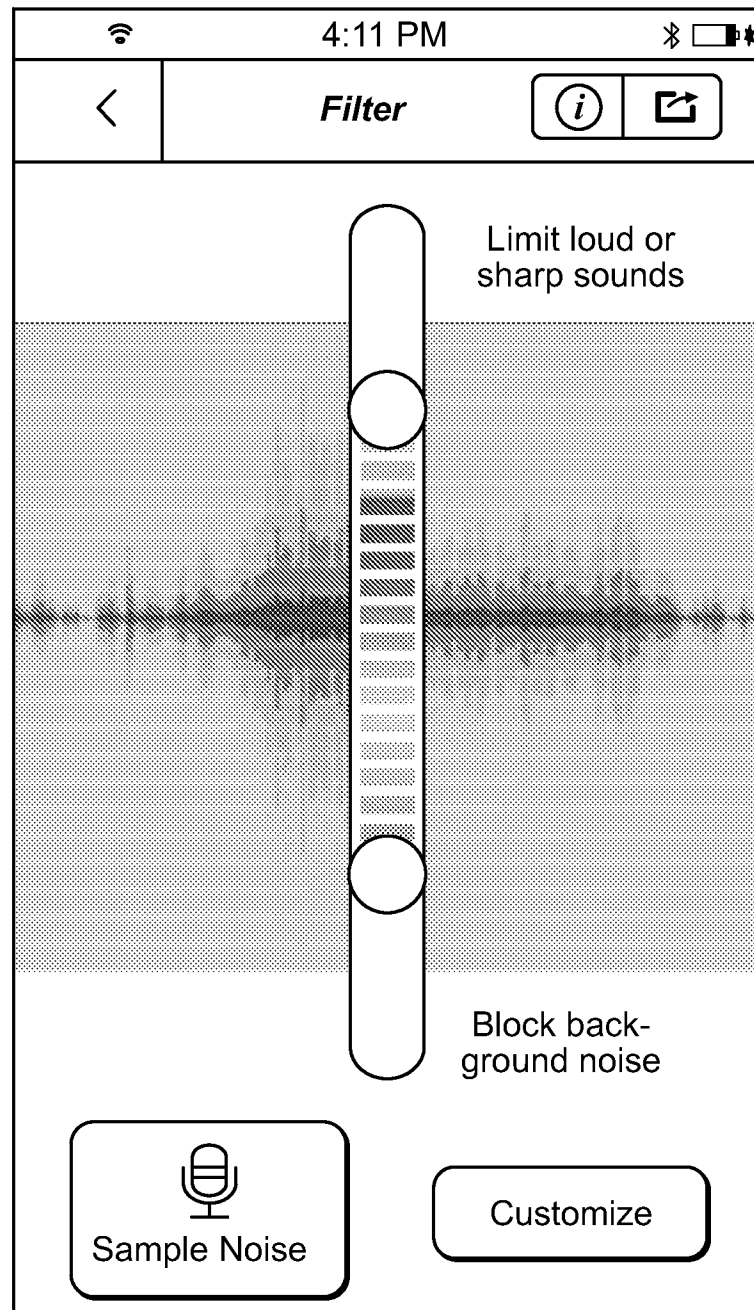


Fig. 6.16

20/22

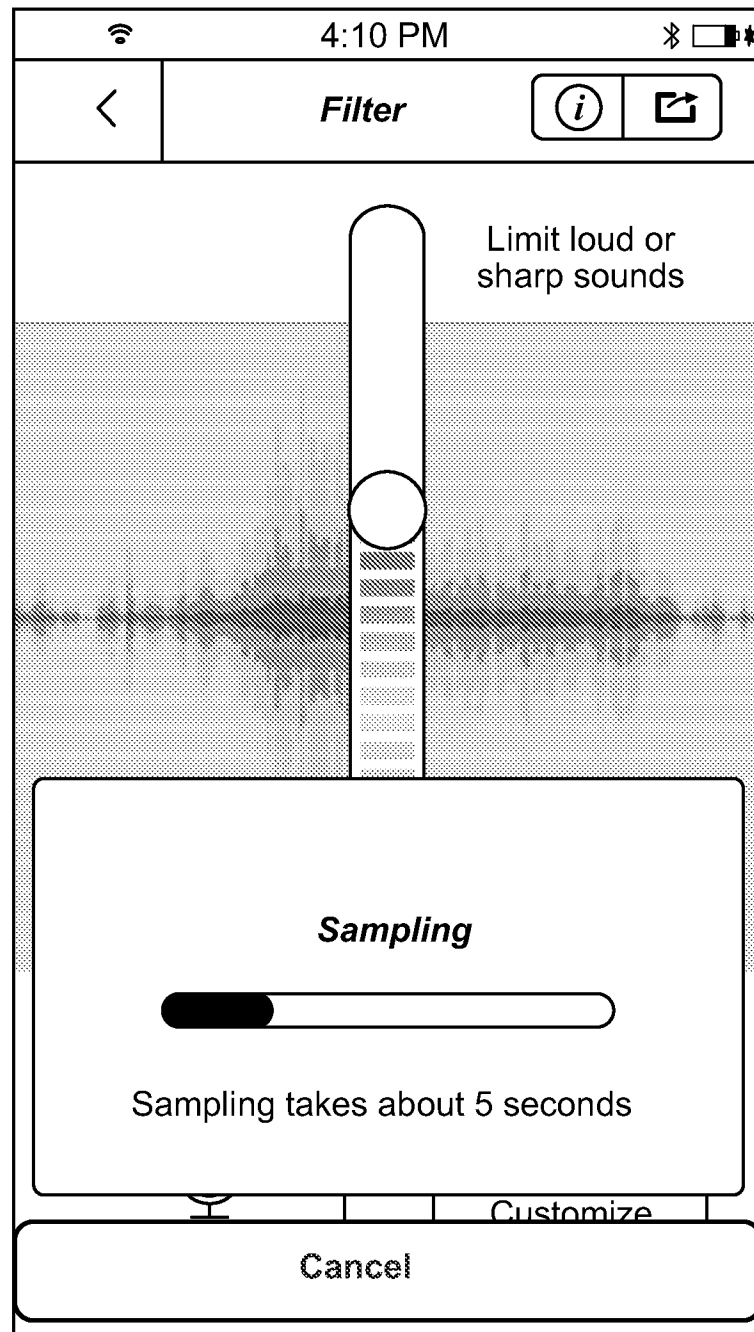


Fig. 6.17

21/22

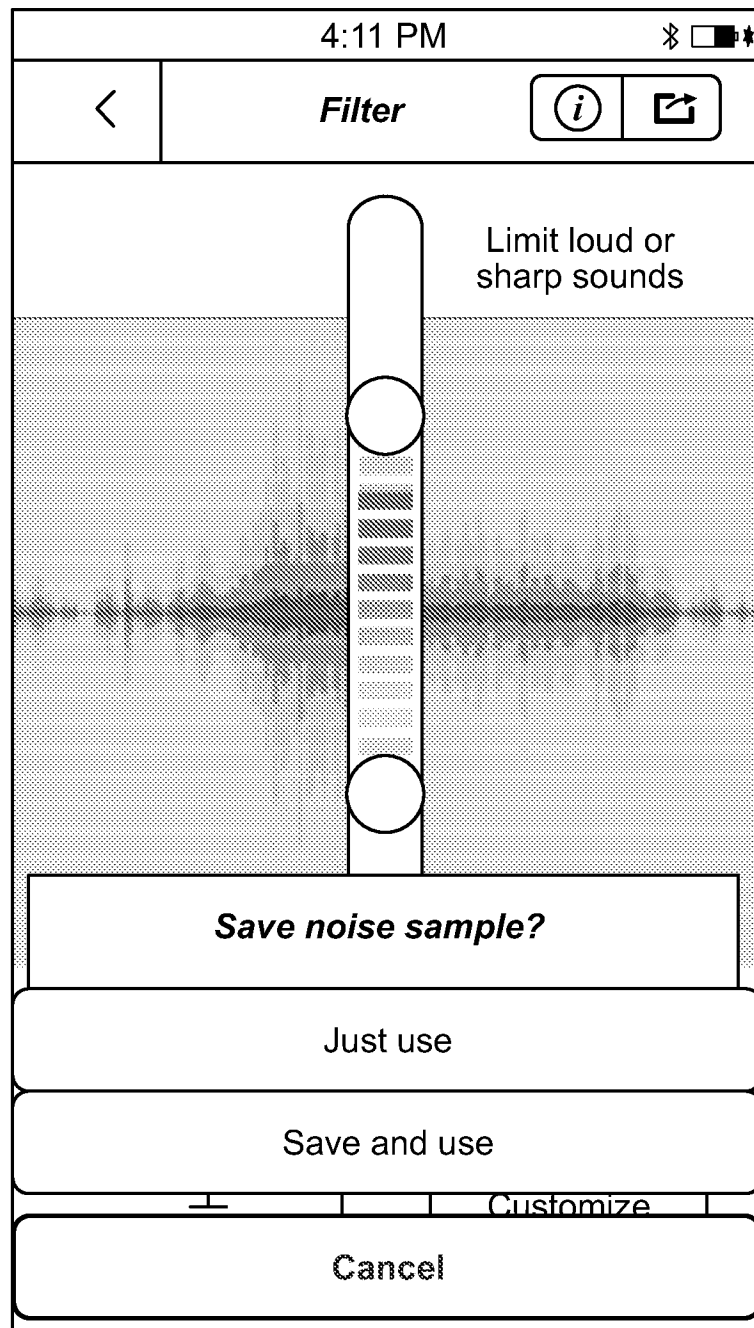


Fig. 6.18

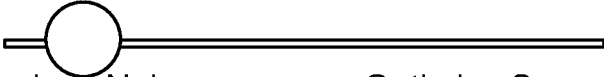
	<i>Advanced Filter</i>  
 Reduce Noise Optimize Speech Noise Reduction 	
SELECT A NOISE 'FILTER' TO USE	
Continuous adaptation	
Mac	

Fig. 6.19